Ilmu Big Data dan Mesin Cerdas

Dr. Budi Raharjo, S.Kom., M.Kom., MM.

YAYASAN PRIMA AGUS TEKNIK

Ilmu Big Data dan Mesin Cerdas

Penulis :
Dr. Budi Raharjo, S.Kom., M.Kom., MM.

ISBN :

Editor :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Penyunting :
Dr. Joseph Teguh Santoso, M.Kom.

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom

Penebit :
Yayasan Prima Agus Teknik

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Dekade terakhir, industri komputer dan informasi telah mengalami perubahan yang cepat baik dalam skala *platform* maupun cakupan aplikasi. Komputer, *Smart phone, cloud*, dan jejaring sosial tidak hanya menuntut kinerja tinggi, tetapi juga kecerdasan mesin tingkat tinggi. Faktanya, kita sedang memasuki era analisis *big data* dan komputasi kognitif. Modernisasi ini diamati dengan meluasnya penggunaan ponsel, penyimpanan dan komputasi cloud, penggunaan *artificial intelligence* (Kecerdasan buatan), aplikasi superkomputer yang diperluas, dan penyebaran luas platform *Internet of Things* (IoT). Untuk menghadapi paradigma komputasi dan komunikasi baru ini.

Di era *Big data*, sistem *cloud*, layanan web, dan pusat data yang sukses harus dirancang untuk menyimpan, memproses, mempelajari, dan menganalisis *Big Data* untuk menemukan pengetahuan baru atau membuat keputusan penting. Tujuannya adalah untuk membangun industri *Big Data* untuk menyediakan layanan kognitif untuk mengimbangi kekurangan manusia dalam menangani tugas padat karya dengan efisiensi tinggi. Tujuan ini dicapai melalui virtualisasi perangkat keras*, Smart Machine, Deep Learning,* penginderaan IoT, analitik data, dan komputasi kognitif. Misalnya, layanan *cloud* baru muncul sebagai *Learning as a Services (LaaS), Analytics as a Service (AaaS),* atau *Security as a Service (SaaS),* bersama dengan praktik pembelajaran mesin dan analitik data yang berkembang.

Buku ini terdiri dari delapan Bab, disajikan dalam tiga bagian teknis. Ketiga bagian tersebut harus dibaca atau diajarkan secara berurutan, seluruhnya atau selektif. Pada Bagian I memiliki tiga bab tentang ilmu data, peran *cloud*, dan perangkat atau kerangka IoT untuk komputasi data besar. Bab-bab ini mencakup teknologi yang memungkinkan untuk mengeksplorasi komputasi *smart cloud* dengan Analisis *Big Data* dan kemampuan pembelajaran mesin kognitif. Kami mencakup arsitektur *cloud, IoT* dan sistem kognitif, dan dukungan perangkat lunak. *Cloud* seluler dan kerangka kerja interaksi IoT diilustrasikan dengan desain sistem dan contoh aplikasi yang konkret.

Bagian II memiliki tiga bab yang dikhususkan untuk prinsip dan algoritma untuk pembelajaran mesin, analisis data, dan pembelajaran mendalam dalam aplikasi data besar. Buku ini menyajikan metode pembelajaran mesin terawasi dan tidak terawasi serta pembelajaran mendalam dengan jaringan saraf tiruan. Arsitektur komputer yang terinspirasi otak. Bab-bab ini meletakkan dasar yang diperlukan untuk metodologi desain dan implementasi algoritma.

Bagian III menyajikan dua bab tentang analitik *Big Data* untuk pembelajaran mesin di bidang kesehatan dan *Deep Learning* untuk aplikasi kognitif dan media sosial. Pembaca harus menguasai sistem, algoritma, dan perangkat lunak seperti proyek DeepMind Google dalam mempromosikan aplikasi *Big Data Artificial Intelligence* di *cloud* atau bahkan di perangkat seluler atau sistem komputer apa pun. Buku ini ditulis untuk memenuhi kurikulum yang diperbarui dalam pendidikan Ilmu Komputer dan Teknik Elektro. Akhir kata semoga buku ini berguna bagi para pembaca.

Semarang, April 2022
Penulis

Dr. Budi Raharjo, MM., M.Kom.

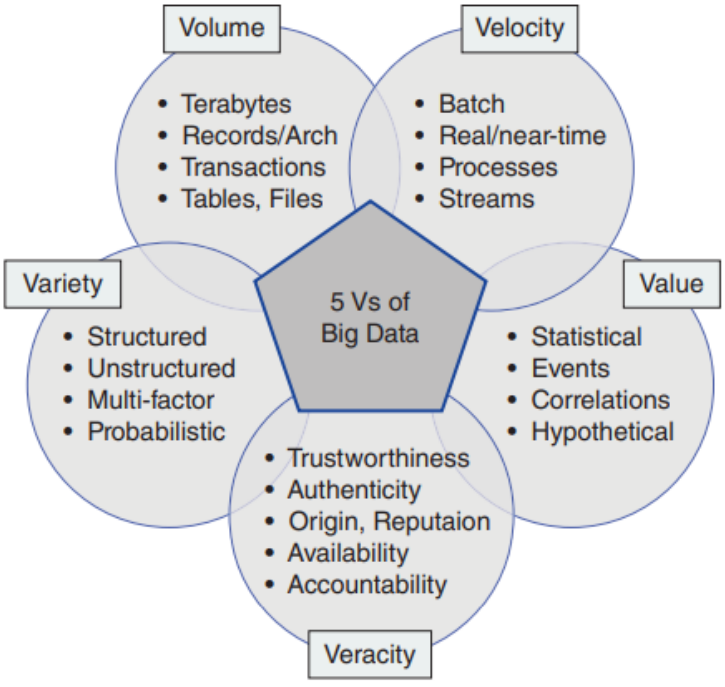
DAFTAR ISI

BAGIAN 1
BIG DATA, *CLOUDS*, DAN *INTERNET OF THINGS*
BAB 1
ILMU *BIG DATA* DAN *SMART MACHINE*

1.1 MENGAKTIFKAN TEKNOLOGI UNTUK COMPUTING *BIG DATA*

Selama tiga dekade terakhir, keadaan teknologi tinggi telah mengalami perubahan besar dalam platform computing dan komunikasi. Secara khusus, kami mendapat banyak manfaat dari peningkatan kinerja Internet dan *World Wide Web* (WWW). Kami memeriksa di sini perubahan evolusioner dalam arsitektur platform, infrastruktur yang digunakan, konektivitas jaringan dan variasi aplikasi. Alih-alih menggunakan desktop atau komputer pribadi untuk memecahkan masalah computing, cloud muncul sebagai platform hemat biaya untuk melakukan pencarian, penyimpanan, dan computing basis data skala besar melalui Internet.

Bab ini memperkenalkan konsep dasar ilmu data dan teknologi pendukungnya. Tujuan utamanya adalah untuk memadukan jaringan sensor, penandaan RFID (identifikasi frekuensi radio), layanan GPS, jaringan sosial, ponsel pintar, tablet, cloud dan *Mashup*, WiFi, Bluetooth, Internet nirkabel+, dan jaringan inti 4G/5G dengan jaringan inti yang sedang berkembang. Internet of Things (IoT) untuk membangun industri *big data* yang produktif di tahun-tahun mendatang. Secara khusus, kami akan memeriksa gagasan fusi teknologi di antara teknologi SMACT.

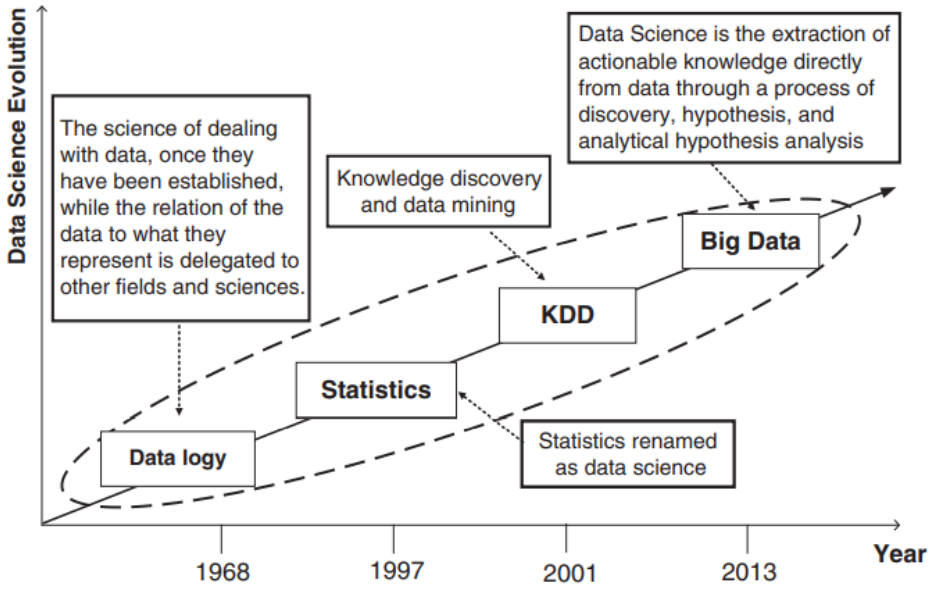


Gambar 1.1 Karakteristik *Big data*: Lima V dan tantangan terkait.

Ilmu Data dan Disiplin Terkait

Konsep ilmu data memiliki sejarah panjang, tetapi baru belakangan ini menjadi sangat populer karena meningkatnya penggunaan cloud dan IoT untuk membangun dunia yang cerdas. Seperti yang diilustrasikan pada Gambar 1.1, *Big data* saat ini memiliki tiga karakteristik penting: data dalam volume besar, menuntut kecepatan tinggi untuk memprosesnya, dan banyak jenis tipe data. Ini sering dikenal sebagai lima V *Big data*, karena beberapa orang menambahkan dua V *Big data* lagi: satu adalah kejujuran, yang mengacu pada kesulitan untuk melacak data atau memprediksi data. Yang lainnya adalah nilai data, yang dapat bervariasi secara drastis jika data ditangani secara berbeda. Menurut standar saat ini, satu Terabyte atau lebih besar dianggap sebagai *Big data*. IDC telah memperkirakan bahwa 40 ZB data akan diproses pada tahun 2030, yang berarti setiap orang mungkin memiliki

5.2 TB data yang akan diproses. Volume tinggi menuntut kapasitas penyimpanan yang besar dan kemampuan analitis untuk menangani volume data yang begitu besar. Keragaman yang tinggi menyiratkan bahwa data datang dalam berbagai format, yang bisa sangat sulit dan mahal untuk dikelola secara akurat. Kecepatan tinggi mengacu pada ketidakmampuan untuk memproses *Big data* secara *real time* untuk mengekstrak informasi atau pengetahuan yang berarti darinya. Kebenaran menyiratkan bahwa agak sulit untuk memverifikasi data. Nilai *Big data* bervariasi dengan domain aplikasinya. Kelima V membuat sulit untuk menangkap, mengelola, dan memproses *Big data* menggunakan infrastruktur perangkat keras/perangkat lunak yang ada. 5 V ini membenarkan panggilan untuk cloud yang lebih cerdas dan dukungan IoT.



Gambar 1.2 Evolusi ilmu data hingga era *big data*.

Forbes, Wikipedia dan NIST telah memberikan beberapa tinjauan sejarah bidang ini. Untuk mengilustrasikan evolusinya ke era *Big data*, kami membagi timeline menjadi empat tahap,

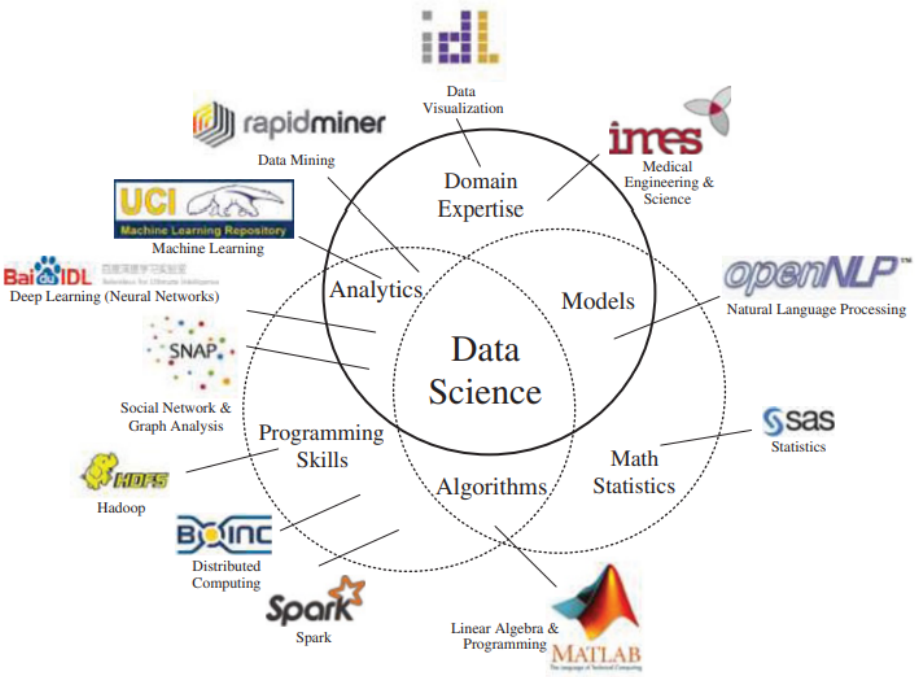
seperti yang ditunjukkan pada Gambar 1.2. Pada tahun 1970-an, beberapa orang menganggap ilmu data setara dengan logika data, seperti yang dicatat oleh Peter Naur: “Ilmu yang menangani data, setelah mereka ditetapkan, sedangkan hubungan data dengan apa yang mereka wakili didelegasikan ke bidang dan ilmu lain.” Pada suatu waktu, ilmu data dianggap sebagai bagian dari statistik dalam berbagai aplikasi. Sejak tahun 2000-an, ruang lingkup ilmu data telah diperluas. Ini menjadi kelanjutan dari bidang penambangan data dan analitik prediktif, juga dikenal sebagai bidang penemuan pengetahuan dan penambangan data (KDD).

Dalam konteks ini, pemrograman dipandang sebagai bagian dari ilmu data. Selama dua dekade terakhir, data telah meningkat dalam skala yang meningkat di berbagai bidang. Evolusi ilmu data memungkinkan ekstraksi pengetahuan dari volume besar data yang terstruktur atau tidak terstruktur. Data tidak terstruktur meliputi email, video, foto, media sosial, dan konten buatan pengguna lainnya. Pengelolaan *Big data* memerlukan skalabilitas di seluruh sumber daya penyimpanan, computing, dan komunikasi dalam jumlah besar.

Secara formal, kami mendefinisikan ilmu data sebagai proses ekstraksi pengetahuan yang dapat ditindaklanjuti langsung dari data melalui penemuan data, hipotesis, dan hipotesis analitik. Seorang ilmuwan data adalah seorang praktisi yang memiliki pengetahuan yang cukup tentang rezim keahlian yang tumpang tindih dalam kebutuhan bisnis, pengetahuan domain, keterampilan analitis, dan keahlian pemrograman untuk mengelola proses ilmiah ujung ke ujung melalui setiap tahap dalam siklus hidup *Big data*.

Ilmu data saat ini membutuhkan agregasi dan pemilahan melalui sejumlah besar informasi dan algoritma penulisan untuk mengekstrak wawasan dari elemen data skala besar. Ilmu data memiliki berbagai aplikasi, terutama dalam uji klinis, ilmu biologi, pertanian, perawatan medis dan jaringan sosial, dll. Kami membagi rantai nilai *Big data* menjadi empat fase: yaitu pembuatan data, akuisisi, penyimpanan, dan analisis. Jika kita mengambil data sebagai bahan mentah, pembuatan data dan akuisisi data adalah proses eksploitasi. Penyimpanan data dan analisis data membentuk proses produksi yang menambah nilai bahan baku.

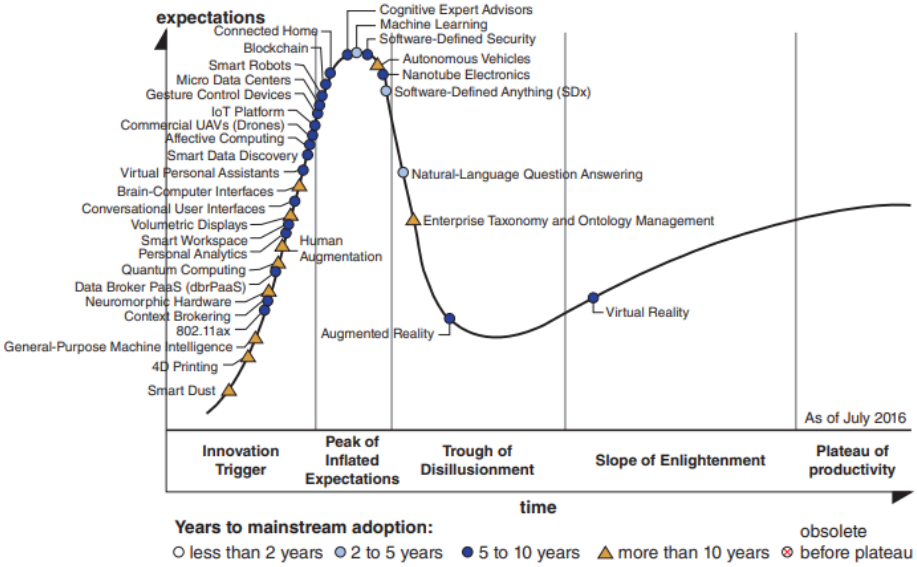
Pada Gambar 1.3, ilmu data dianggap sebagai persilangan dari tiga bidang interdisipliner: ilmu komputer atau keterampilan pemrograman, matematika dan statistik, dan keahlian domain aplikasi. Sebagian besar ilmuwan data dimulai sebagai pakar domain yang menguasai pemodelan matematika, teknik penambangan data, dan analitik data. Melalui kombinasi pengetahuan domain dan keterampilan matematika, model khusus dikembangkan sementara algoritma dirancang. Ilmu data berjalan di seluruh siklus hidup data. Ini menggabungkan prinsip, teknik, dan metode dari banyak disiplin ilmu dan domain, termasuk penambangan data dan analitik, terutama ketika *Machine learning* dan pengenalan pola diterapkan.



Gambar 1.3 Komponen fungsional ilmu data yang didukung oleh beberapa perpustakaan perangkat lunak di cloud pada tahun 2016

Statistik, riset operasi, visualisasi, dan pengetahuan domain juga tak tergantikan. Tim ilmu data memecahkan masalah data yang sangat kompleks. Seperti yang ditunjukkan pada Gambar 1.3, ketika dua area tumpang tindih, mereka menghasilkan tiga bidang minat khusus yang penting.Bidang pemodelan dibentuk dengan memotong keahlian domain dengan statistik matematika. Pengetahuan yang akan ditemukan sering digambarkan dengan bahasa matematika yang abstrak. Bidang lain adalah analitik data, yang dihasilkan dari persilangan keahlian domain dan keterampilan pemrograman. Pakar domain menerapkan alat pemrograman khusus untuk menemukan pengetahuan dengan memecahkan masalah praktis di domain mereka. Akhirnya, bidang algoritme adalah persilangan antara keterampilan pemrograman dan statistik matematika. Rangkuman di bawah ini adalah beberapa tantangan terbuka dalam penelitian, pengembangan, dan aplikasi *Big data*:

- Data terstruktur versus tidak terstruktur dengan pengindeksan yang efektif;
- Identifikasi, de-identifikasi dan re-identifikasi;
- Ontologi dan semantik *Big data*;
- Teknik introspeksi dan reduksi data;
- Desain, konstruksi, operasi dan deskripsi;
- Integrasi data dan interoperabilitas perangkat lunak;
- Kekekalan dan keabadian;
- metode pengukuran data;
- Rentang data, penyebut, tren dan estimasi.



Gambar 1.4 Siklus sensasi untuk munculnya teknologi tinggi untuk mencapai kedewasaan dan produktivitas industri dalam dekade berikutnya. (Sumber: Gartner Research, Juli 2016, dicetak ulang dengan izin.)

Teknologi yang Muncul dalam Dekade Berikutnya

Garnter Research adalah sumber resmi teknologi baru. Mereka mengidentifikasi teknologi baru yang muncul terpanas dalam siklus hype setiap tahun. Pada Gambar 1.4 kami memeriksa Siklus Hype Gartner untuk teknologi baru yang muncul di banyak bidang pada tahun 2016. Waktu yang dibutuhkan untuk teknologi baru untuk menjadi matang mungkin memerlukan 2 hingga 10 tahun untuk mencapai puncak produktivitasnya. Pada tahun 2016, teknologi yang paling diharapkan diidentifikasi pada puncak siklus hype. 12 teratas termasuk penasihat ahli kognitif, *Machine learning*, keamanan yang ditentukan perangkat lunak, rumah yang terhubung, kendaraan otonom, blockchain, elektronik nanotube, robot pintar, pusat data mikro, perangkat kontrol gerakan, platform IoT, dan drone (UAV komersial).

Seperti yang diidentifikasi oleh lingkaran padat gelap, sebagian besar teknologi membutuhkan waktu 5 hingga 10 tahun untuk matang. Lingkaran padat ringan, seperti *Machine learning*, apa pun yang ditentukan perangkat lunak (SDx), dan penjawab bahasa alami, adalah lingkaran yang mungkin menjadi matang dalam waktu 2 hingga 5 tahun. Pembaca harus memeriksa siklus hype yang dirilis pada tahun-tahun sebelumnya untuk menemukan lebih banyak teknologi panas. Segitiga mengidentifikasi mereka yang mungkin membutuhkan lebih dari 10 tahun pengembangan lebih lanjut. Mereka adalah pencetakan 4-D, Smart Machine tujuan umum, perangkat keras neuromorfik, computing kuantum dan kendaraan otonom, dll. Mobil self-driving adalah topik hangat pada tahun 2016, tetapi mungkin perlu lebih banyak waktu untuk diterima, baik secara teknis maupun hukum. Taksonomi perusahaan dan manajemen ontologi sedang memasuki tahap kekecewaan, tetapi masih perlu waktu lama untuk menjadi kenyataan.

Teknologi panas lainnya, seperti augmented reality dan virtual reality, menghasilkan kekecewaan, tetapi sekarang sedang menuju produktivitas industri. Pada tahap pemicu inovasi awal, kami mengamati bahwa Wifi 11.ac dan broker konteks sedang naik daun, bersama dengan PaaS broker data (dbrPaaS), analitik pribadi, tempat kerja cerdas, antarmuka pengguna percakapan, penemuan data cerdas, computing afektif , asisten pribadi virtual, keamanan digital, dan teknologi yang melek masyarakat. Banyak teknologi lain yang sedang naik daun dari kurva ekspektasi termasuk bio-printing 3-D, rumah yang terhubung, biochip, keamanan yang ditentukan perangkat lunak, dll. Siklus sensasi ini mencakup teknologi yang lebih matang seperti computing cloud hybrid, pertukaran mata uang kripto dan pencetakan 3-D perusahaan yang diidentifikasi pada tahun-tahun sebelumnya.

Beberapa teknologi yang lebih matang seperti computing cloud, jejaring sosial, komunikasi jarak dekat (NFC), pemindai 3-D, telematika konsumen dan pengenalan suara, yang telah muncul dalam siklus sensasi yang dirilis dari 2010 hingga 2015, tidak muncul pada Gambar 1.4. Kedalaman kekecewaan mungkin tidak buruk, karena ketika minat berkurang setelah eksperimen ekstensif, pelajaran yang berguna dipelajari untuk menghasilkan produk dengan lebih berhasil. Teknologi jangka panjang yang ditandai dengan segitiga dalam siklus hype juga tidak dapat diabaikan. Sebagian besar pengembang industri berpandangan dekat atau sangat konservatif dalam arti bahwa mereka hanya mengadopsi teknologi matang yang dapat menghasilkan produk yang menguntungkan dengan cepat. Secara tradisional, teknologi jangka panjang atau berisiko tinggi seperti computing kuantum, debu pintar, penginderaan bio-akustik, tampilan volumetrik, antarmuka otak-manusia, dan komputer saraf hanya banyak dikejar di dunia akademis.

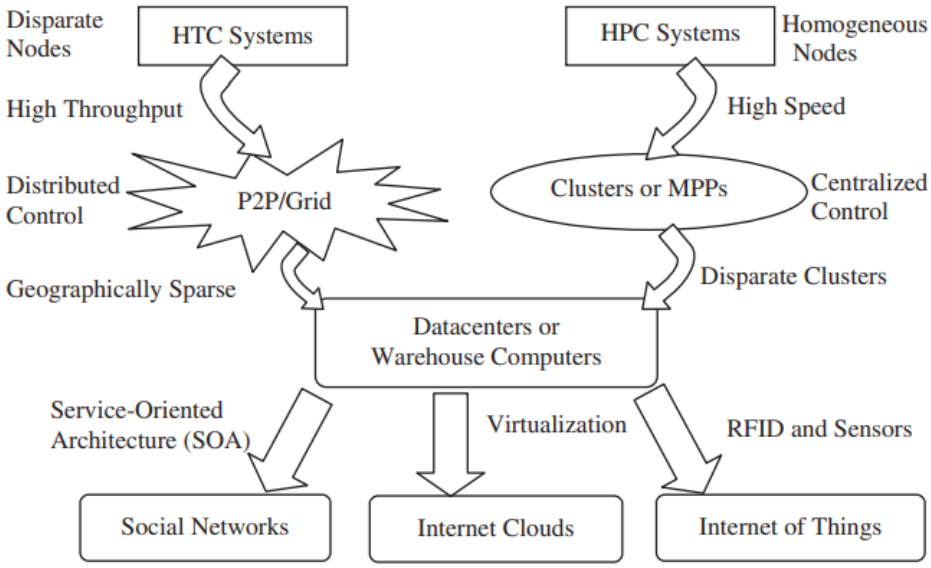
Telah diterima dengan baik bahwa teknologi akan terus menjadi lebih human-centric, ke titik di mana ia akan memperkenalkan transparansi antara orang, bisnis, dan hal-hal. Hubungan ini akan semakin mengemuka seiring dengan evolusi teknologi yang semakin adaptif, kontekstual dan lancar di tempat kerja, di rumah, dan berinteraksi dengan dunia bisnis. Seperti yang ditunjukkan di atas, kita melihat munculnya pencetakan 4-D, computing mirip otak, pembesaran manusia, tampilan volumetrik, computing afektif, rumah yang terhubung, elektronik nanotube, augmented reality, realitas virtual, dan perangkat kontrol gerakan. Beberapa di antaranya akan dibahas dalam bab-bab berikutnya.

Ada tren yang dapat diprediksi dalam teknologi yang mendorong aplikasi computing. Desainer dan pemrogram ingin memprediksi kemampuan teknologi sistem masa depan. Makalah Jim Gray “Rules of Thumb in Data Engineering” adalah contoh yang sangat baik tentang bagaimana teknologi memengaruhi aplikasi dan sebaliknya. Hukum Moore menunjukkan bahwa kecepatan prosesor berlipat ganda setiap 18 bulan. Ini memang benar selama 30 tahun terakhir.

Namun, sulit untuk mengatakan bahwa Hukum Moore akan bertahan lebih lama di masa depan. Hukum Gilder menunjukkan bahwa bandwidth jaringan berlipat ganda setiap tahun di masa lalu. Rasio harga/kinerja yang luar biasa dari komoditas perangkat keras didorong oleh pasar

ponsel pintar, tablet, dan notebook. Ini juga telah memperkaya teknologi komoditas dalam computing skala besar.

Sangat menarik untuk melihat ekspektasi yang tinggi dari IoT dalam beberapa tahun terakhir. Computing cloud dalam *mashup* atau aplikasi lain menuntut ekonomi computing, pengumpulan data skala web, keandalan sistem, dan kinerja yang dapat diskalakan. Misalnya, pemrosesan transaksi terdistribusi sering dipraktikkan di industri perbankan dan keuangan. Transaksi mewakili 90% dari pasar yang ada untuk sistem perbankan yang andal. Pengguna harus berurusan dengan beberapa server database dalam transaksi terdistribusi. Bagaimana menjaga konsistensi catatan transaksi yang direplikasi sangat penting dalam layanan perbankan real-time. Komplikasi lain termasuk kekurangan dukungan perangkat lunak, kejenuhan jaringan dan ancaman keamanan dalam aplikasi bisnis ini.



Gambar 1.5 Tren evolusi menuju computing paralel, terdistribusi, dan cloud menggunakan cluster, MPP, jaringan P2P, *grid computing, cloud Internet*, layanan web, dan *Internet of things*. (HPC: computing kinerja tinggi; HTC: computing throughput tinggi; P2P: peer-to-peer; MPP: prosesor paralel besar-besaran; RFID: Identifikasi Frekuensi Radio.)

Sejumlah teknologi yang lebih matang yang mungkin memakan waktu 2 sampai 5 tahun untuk mencapai dataran tinggi disorot oleh titik abu-abu terang pada Gambar 1.4. Ini termasuk biochip, analitik tingkat lanjut, terjemahan ucapan-ke-suara, *Machine learning*, *computing cloud hybrid*, pertukaran mata uang kripto, kendaraan lapangan otonom, kontrol gerakan, dan pencetakan 3-D perusahaan. Beberapa teknologi matang yang banyak dikejar oleh industri sekarang tidak ditampilkan dalam siklus sensasi 2016 sebagai teknologi baru. Ini mungkin termasuk *computing cloud*, jejaring sosial, komunikasi jarak dekat (NFC), pemindai 3-D, telematika konsumen, dan pengenalan suara yang muncul dalam siklus hype dalam beberapa tahun terakhir. Sangat menarik untuk melihat ekspektasi yang tinggi dari IoT dalam beberapa tahun terakhir. *Computing cloud* dalam *mashup* atau cloud hibrida telah diadopsi di arus utama.

Seiring berjalannya waktu, sebagian besar teknologi akan maju ke tahap harapan yang lebih baik. Seperti disebutkan di atas, kekecewaan mendalam mungkin tidak terlalu buruk, karena minat berkurang setelah eksperimen ekstensif, dan pelajaran berguna dipelajari untuk menghasilkan produk dengan sukses. Perlu dicatat bahwa teknologi jangka panjang yang ditandai dengan segitiga dalam siklus hype mungkin membutuhkan waktu lebih dari 10 tahun untuk menjadi kenyataan industri. Ini termasuk bidang computing kuantum yang meningkat, debu pintar, penginderaan akustik bio, tampilan volumetrik, augmentasi manusia, antarmuka brainhuman, dan bisnis saraf yang populer di komunitas akademisi dan penelitian.

Tren computing umum adalah untuk memanfaatkan lebih banyak dan lebih banyak sumber daya web bersama melalui Internet. Seperti yang diilustrasikan pada Gambar 1.5, kita melihat evolusi dari dua jalur pengembangan sistem: sistem HPC versus HTC. Di sisi HPC, superkomputer (prosesor paralel besar-besaran, MPP) secara bertahap digantikan oleh kelompok komputer kooperatif karena keinginan untuk berbagi sumber daya computing. Cluster sering merupakan kumpulan node computing homogen yang secara fisik terhubung dalam jarak dekat satu sama lain.

Di sisi HTC, jaringan *Peer-to-Peer* (P2P) dibentuk untuk berbagi file terdistribusi dan aplikasi pengiriman konten. Baik P2P, computing cloud, dan platform layanan web lebih menekankan pada HTC daripada aplikasi HPC. Selama bertahun-tahun, sistem HPC menekankan kinerja kecepatan mentah. Oleh karena itu, kami menghadapi perubahan strategis dari paradigma HPC ke HTC. Paradigma HTC ini lebih memperhatikan multi-computing fluks tinggi, di mana pencarian Internet dan layanan web diminta oleh jutaan atau lebih pengguna secara bersamaan. Tujuan kinerja dengan demikian digeser untuk mengukur throughput yang tinggi atau jumlah tugas yang diselesaikan per unit waktu.

Di era *Big data*, kita menghadapi masalah banjir data. Data berasal dari sensor IoT, eksperimen lab, simulasi, arsip masyarakat, dan web dalam semua skala dan format. Pelestarian, pergerakan, dan akses kumpulan *Big data* memerlukan alat umum yang mendukung sistem file, database, algoritme, alur kerja, dan visualisasi yang dapat diskalakan dengan kinerja tinggi. Dengan sains menjadi data centric, paradigma baru penemuan ilmiah didasarkan pada computing data intensif. Kita perlu mengembangkan alat untuk pengambilan data, pembuatan data, dan analisis data. Teknologi cloud dan IoT didorong oleh lonjakan minat terhadap banjir data.

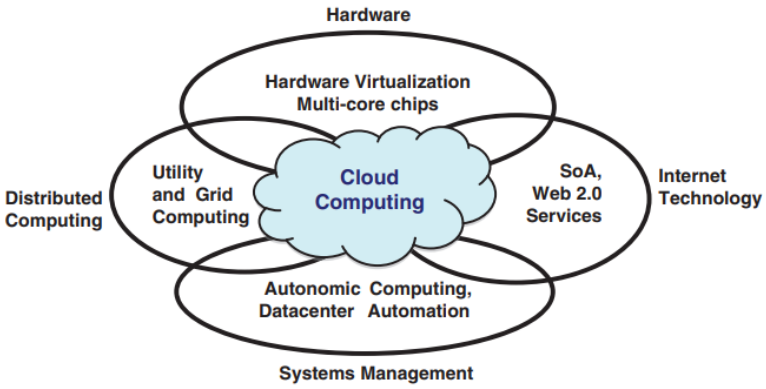
Internet dan WWW digunakan oleh miliaran orang setiap hari. Akibatnya, pusat data atau cloud yang besar harus dirancang untuk menyediakan tidak hanya penyimpanan besar tetapi juga daya computing terdistribusi untuk memenuhi permintaan sejumlah besar pengguna secara bersamaan. Munculnya cloud publik atau hybrid menuntut peningkatan banyak pusat data menggunakan cluster server yang lebih besar, sistem file terdistribusi, dan jaringan bandwidth tinggi. Dengan ponsel pintar dan tablet besar yang meminta layanan, mesin cloud, penyimpanan terdistribusi, dan jaringan seluler harus berinteraksi dengan Internet untuk memberikan layanan *mashup* dalam computing seluler skala web melalui jaringan sosial dan media secara dekat.

Baik P2P, computing cloud, dan platform layanan web menekankan throughput tinggi pada sejumlah besar tugas pengguna, daripada kinerja tinggi seperti yang sering ditargetkan dalam penggunaan superkomputer. Paradigma throughput tinggi ini lebih memperhatikan fluks tugas pengguna yang tinggi secara bersamaan atau bersamaan. Aplikasi utama dari sistem cloud fluks tinggi terletak pada pencarian Internet dan layanan web. Tujuan kinerja dengan demikian digeser untuk mengukur throughput yang tinggi atau jumlah tugas yang diselesaikan per unit waktu.

Hal ini tidak hanya menuntut peningkatan kecepatan pemrosesan batch yang tinggi, tetapi juga mengatasi masalah akut biaya, penghematan energi, keamanan, dan keandalan di cloud. Kemajuan dalam virtualisasi memungkinkan untuk menggunakan cloud Internet dalam layanan pengguna yang sangat besar. Faktanya, perbedaan antara cluster, sistem P2P, dan cloud mungkin menjadi kabur. Beberapa melihat cloud sebagai cluster computing dengan perubahan sederhana dalam virtualisasi. Yang lain mengantisipasi pemrosesan yang efektif dari kumpulan *Big data* yang dihasilkan oleh layanan web, jejaring sosial, dan IoT. Dalam pengertian ini, banyak pengguna menganggap platform cloud sebagai bentuk computing utilitas atau computing layanan.

Konvergensi Teknologi

Computing cloud diaktifkan oleh konvergensi dari empat teknologi yang diilustrasikan pada Gambar 1.6. Virtualisasi perangkat keras dan chip multicore memungkinkan konfigurasi dinamis di cloud. Utilitas dan teknologi computing grid meletakkan dasar yang diperlukan dari computing cloud. Kemajuan terbaru dalam arsitektur berorientasi layanan (SOA), Web 2.0 dan *mashup* platform mendorong cloud ke langkah maju lainnya. Computing otonom dan operasi pusat data otomatis telah memungkinkan computing cloud.



Gambar 1.6 Konvergensi teknologi memungkinkan computing cloud melalui Internet. (Courtesy of Buyya, Broberg dan Goscinski, dicetak ulang dengan izin [3])

Computing cloud mengeksplorasi teknologi computing multi-core dan paralel. Untuk mewujudkan visi sistem data-intensif, kita perlu melakukan konvergensi dari empat bidang: yaitu perangkat keras, teknologi Internet, computing terdistribusi, dan manajemen sistem, seperti yang diilustrasikan pada Gambar 1.6. Teknologi Internet saat ini menekankan pada layanan SOA dan Web 2.0. Utilitas dan computing grid meletakkan dasar computing terdistribusi yang dibutuhkan

untuk computing cloud. Terakhir, kita tidak dapat mengabaikan meluasnya penggunaan pusat data dengan teknik virtualisasi yang diterapkan untuk mengotomatiskan proses penyediaan sumber daya di cloud.

Computing Utilitas

Paradigma computing dikaitkan dengan karakteristik yang berbeda. Pertama, mereka semua ada di mana-mana dalam kehidupan kita sehari-hari. Keandalan dan skalabilitas adalah dua tujuan desain utama. Kedua, mereka ditujukan untuk operasi otonom yang dapat diatur sendiri untuk mendukung penemuan dinamis. Akhirnya, paradigma ini dapat dicampur dengan QoS (kualitas layanan) dan SLA (perjanjian tingkat layanan), dll. Paradigma ini dan atributnya mewujudkan visi utilitas komputer.

Computing utilitas didasarkan pada model bisnis, di mana pelanggan menerima sumber daya computing dari penyedia layanan cloud atau IoT. Ini menuntut beberapa tantangan teknologi, termasuk hampir semua aspek ilmu komputer dan teknik. Misalnya, pengguna mungkin menuntut prosesor baru yang efisien jaringan, memori yang dapat diskalakan dan skema penyimpanan, OS terdistribusi, middleware untuk virtualisasi mesin, model pemrograman baru, manajemen sumber daya yang efektif, dan pengembangan program aplikasi. Kemajuan perangkat keras dan perangkat lunak ini diperlukan untuk memfasilitasi computing cloud seluler di berbagai domain aplikasi IoT.

Tabel 1.1 Perbedaan tiga model layanan cloud dari computing di tempat dalam kendali sumber daya di bawah tanggung jawab pengguna, vendor, dan bersama.

Jenis Sumber Daya	Computing Lokal	Model IaaS	Model PaaS	Model SaaS
Perangkat Lunak Aplikasi	Pengguna	Pengguna	Bersama	Penjaja
Mesin virtual	Pengguna	Bersama	Bersama	Penjaja
Server	Pengguna	Penjaja	Penjaja	Penjaja
Penyimpanan	Pengguna	Penjaja	Penjaja	Penjaja
Jaringan	Bersama	Penjaja	Penjaja	Penjaja

Computing Cloud versus Computing Lokal

Aplikasi computing tambahan terutama dijalankan pada host lokal di tempat. Mereka muncul sebagai desktop, deskside, notebook atau tablet, dll. Computing di tempat berbeda dari computing cloud terutama dalam kontrol sumber daya dan manajemen infrastruktur. Pada Tabel 1.1, kami membandingkan tiga model layanan cloud dengan paradigma computing di tempat. Kami mempertimbangkan sumber daya perangkat keras dan perangkat lunak dalam lima jenis: penyimpanan, server, mesin virtual, jaringan dan perangkat lunak aplikasi, seperti yang tercantum di kolom kiri Tabel 1.1. Dalam kasus computing di tempat di host lokal, semua sumber daya harus diperoleh oleh pengguna kecuali jaringan, yang dibagi antara pengguna dan penyedia. Ini menyiratkan beban berat dan biaya operasional di pihak pengguna. Dalam hal menggunakan

cloud IaaS seperti AWS EC2, pengguna hanya perlu khawatir tentang penerapan perangkat lunak aplikasi. Mesin virtual digunakan bersama oleh pengguna dan penyedia. Vendor bertanggung jawab untuk menyediakan perangkat keras dan jaringan yang tersisa. Dalam menggunakan cloud PaaS, seperti Google AppEngine, kode aplikasi dan mesin virtual digunakan bersama oleh pengguna dan vendor dan sumber daya yang tersisa disediakan oleh vendor. Akhirnya, ketika model SaaS menggunakan *cloud Saleforce,* semuanya disediakan oleh vendor, bahkan termasuk perangkat lunak aplikasi. Sebagai kesimpulan, kami melihat bahwa *computing cloud* mengurangi beban manajemen infrastruktur pengguna dari dua sumber menjadi tidak ada sama sekali, saat kami beralih dari layanan IaaS ke PaaS dan SaaS. Ini jelas menunjukkan keuntungan bagi pengguna dalam memisahkan aplikasi dari investasi dan manajemen sumber daya.

Menuju Industri Big data

Seperti yang ditunjukkan pada Tabel 1.2, kami memiliki industri database pada tahun 1960 hingga 1990-an. Pada saat itu sebagian besar blok data diukur sebagai MB, GB, dan TB. Pusat data mulai digunakan secara luas dari tahun 1980 hingga 2010, dengan kumpulan data dengan mudah mulai dari TB hingga PB atau bahkan EB. Setelah 2010, kami melihat formasi bertahap dari industri baru yang disebut *big data*. Untuk mengolah *big data* kedepannya, kami harapkan EB sampai ZB atau YB. Ukuran pasar industri *Big data* mencapai 34 miliar pada tahun 2013. Melampaui 100 miliar aplikasi *Big data* dapat dicapai pada tahun 2020.

Tabel 1.2 Evolusi industri *big data* dalam tiga tahap pengembangan.

Panggung	**database**	**Pusat Data**	**Industri *Big data***
Jangka waktu	1960-1990	1980–2010	2010 dan Selanjutnya
Ukuran Data	MB – GB -TB	TB – PB-EB	EB – ZB- YB
Ukuran Pasar dan Tingkat Pertumbuhan	Pasar basis data, Rekayasa Data/Pengetahuan	Pasar Rp 339.000 M menurut IDC 2012, (pertumbuhan 21,5%)	Pengeluaran TI Rp 510.000 miliar (2013), 4,4 juta pekerjaan *Big data* baru (2015), Gartner memperkirakannya akan melebihi 100 miliar pada tahun 2020

Teknologi SMACT Interaktif

Hampir semua aplikasi menuntut ekonomi computing, pengumpulan data skala web, keandalan sistem, dan kinerja yang dapat diskalakan, seperti di industri perbankan dan keuangan yang dijelaskan di atas. Dalam beberapa tahun terakhir, lima teknologi informasi mutakhir: yaitu Sosial, Seluler, Analytics, Cloud, dan IoT, menjadi lebih menuntut, yang dikenal sebagai teknologi SMACT. Tabel 1.3 merangkum teori yang mendasari, perangkat keras, perangkat lunak dan

kemajuan jaringan, dan penyedia layanan perwakilan dari lima teknologi ini. Kami akan mempelajari kemajuan ini dalam bab-bab berikutnya.

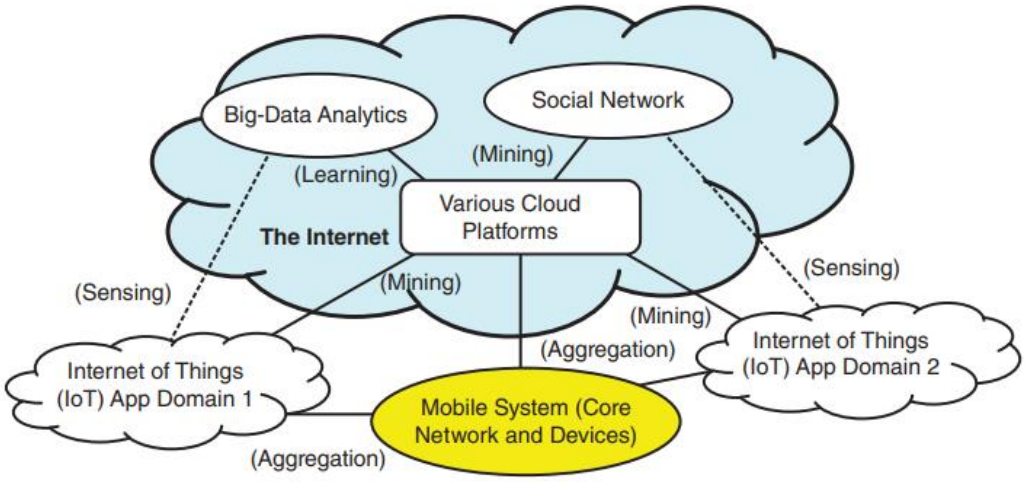
Internet Hal

Internet tradisional menghubungkan mesin ke mesin atau halaman web ke halaman web. IoT mengacu pada interkoneksi jaringan dari objek, alat, perangkat, atau komputer sehari-hari. Hal-hal (objek) kehidupan kita sehari-hari bisa besar atau kecil. Idenya adalah untuk menandai setiap objek menggunakan identifikasi frekuensi radio (RFID) atau sensor terkait atau teknologi elektronik seperti GPS (sistem penentuan posisi global). Dengan diperkenalkannya protokol IPv6, ada 2128 alamat IP yang tersedia untuk membedakan semua objek di Bumi, termasuk semua perangkat seluler, perangkat yang disematkan, komputer, dan bahkan beberapa objek biologis. Diperkirakan bahwa rata-rata orang dikelilingi oleh 1000 hingga 5000 objek setiap hari. IoT perlu dirancang untuk melacak 100 triliun objek statis atau bergerak secara bersamaan. Untuk alasan ini, IoT menuntut kemampuan beralamat yang unik dari semua objek di Bumi. Objek dikodekan atau diberi label dan dapat diidentifikasi dengan IP. Mereka diinstrumentasi dan saling terhubung oleh berbagai jenis jaringan kabel atau nirkabel. Dalam beberapa kasus mereka dapat berinteraksi satu sama lain secara cerdas melalui jaringan. Istilah *Internet of Things* (IoT) adalah konsep fisik. Ukuran IoT bisa besar atau kecil, mencakup wilayah lokal atau berbagai ruang fisik. IoT bukan sekadar jaringan virtual atau jaringan logis atau jaringan peer-to-peer (P2P) di ruang siber. Dengan kata lain, IoT dibangun di dunia fisik, meskipun secara logika dapat dialamatkan di dunia maya.

Komunikasi antar objek dapat dilakukan dengan berbagai cara: Misalnya, H2H mengacu pada manusia-ke-manusia, H2T untuk manusia-ke-benda, T2T untuk benda-benda, dll. Yang penting adalah menghubungkan segala sesuatu di kapan saja dan di mana saja dengan biaya rendah. Dengan koneksi apa pun, kami merujuk ke antara PC, H2H (tidak menggunakan PC tetapi menggunakan perangkat seluler), H2T (menggunakan peralatan generik) dan T2T. Dengan koneksi di mana saja, kami merujuk ke semua PC, di dalam ruangan, di luar ruangan, dan saat bepergian. Kapan saja, kami menyiratkan koneksi pada periode waktu apa pun: siang hari, malam hari, di luar ruangan dan di dalam ruangan, dan saat bepergian, dll. Koneksi dinamis akan tumbuh secara eksponensial menjadi jaringan jaringan universal baru, yang disebut IoT. IoT sangat terkait dengan domain aplikasi tertentu. Domain aplikasi yang berbeda dianut oleh lingkaran atau kelompok komunitas yang berbeda dalam masyarakat kita. Kami cukup menyebutnya sebagai domain IoT atau jaringan IoT.

Tabel 1.3 Teknologi SMACT yang dicirikan oleh teori dasar, perangkat keras tipikal, perangkat lunak, jaringan dan penyedia layanan yang dibutuhkan.

Teknologi SMACT	**Landasan Teoritis**	**Kemajuan Perangkat Keras**	**Perangkat Lunak dan Pustaka**	**Pengaktif Jaringan**	**Penyedia Layanan Perwakilan**
Sistem Seluler	Telekomunikasi, Teori Akses Radio, Computing Seluler	Perangkat Cerdas, Nirkabel, Infrastruktur Mobilitas	Android, iOS, Uber, WeChat, NFC, iCloud, Google Player	4G LTE, WiFi, Bluetooth, Jaringan Akses Radio	AT&T Nirkabel, T-Mobile, Verizon, Apple, Samsung, Huawei
Jaringan sosial	Ilmu Sosial, Teori Grafik, Statistika, Computing Sosial	Pusat Data, Mesin Pencari, dan Infrastruktur WWW	Browser, API, Web 2.0. YouTube, Whatsapp, WeChat, Pijat	Internet Broadband, Jaringan yang Ditentukan Perangkat Lunak	Facebook, Twitter, QQ, Linkedin, Baidu, Amazon, Taobao
Analisis *Big data*	Penambangan Data, *Machine learning*, Kecerdasan Buatan	Pusat Data, Cloud, Mesin Pencari, Danau *Big data*, Penyimpanan Data	Spark, Hama, DatTorrent, MLlib, Impala, GraphX, KFS, Hive, Hbase,	Co-Location Clouds, *Mashup*, Jaringan P2P, dll.	AMPLab, Apache, Cloudera, FICO, Databricks, eBay, Oracle
Computing cloud	Virtualisasi Computing Paralel dan Terdistribusi	Cluster server, Cloud, Mesin Virtual, Jaringan interkoneksi.	OpenStack, GFS, HDFS, MapReduce, Hadoop, Spark, Storm, Cassandra	Jaringan Virtual, Jaringan OpenFlow, Jaringan yang Ditentukan Perangkat Lunak	AWS, GAE, IBM, Salesforce, GoGrid Apache, Azure Rachspace, DropBox
Internet of Things (IoT)	Teori Penginderaan, Fisika Cyber, Navigasi, Computing Pervasif	Sensor, RFID, GPS, Robotika, Satelit, Zigbee, Giroskop	TyneOS, WAP, WTCP, IPv6, Mobile IP, Android, iOS, WPKI, UPnP, JVM	LAN Nirkabel, PAN, MANET, WLAN Mesh, VANet, Bluetooth	Dewan IoT, IBM, Layanan Kesehatan, SmartGrid, Media Sosial, Smart Earth, Google, Samsung



Gambar 1.7 Interaksi antara jejaring sosial, sistem seluler, analitik *Big data*, dan platform cloud melalui berbagai domain Internet of Things (IoT).

Interaksi antar Subsistem SMACT

Gambar 1.7 mengilustrasikan interaksi antara lima teknologi SMACT. Beberapa platform cloud bekerja sama dengan banyak jaringan seluler untuk menyediakan inti layanan secara interaktif. Jaringan IoT menghubungkan objek apa pun termasuk sensor, komputer, manusia, dan objek apa pun yang dapat diidentifikasi dengan IP di Bumi. Jaringan IoT muncul dalam berbagai bentuk di domain aplikasi yang berbeda. Jejaring sosial, seperti Facebook dan Twitter, dan sistem analitik *Big data* dibangun di dalam Internet. Semua jaringan sosial, analitik, dan IoT terhubung ke cloud melalui Internet dan jaringan seluler, termasuk beberapa jaringan tepi seperti WiFi, Ethernet, atau bahkan beberapa data GPS dan Bluetooth.

Kita perlu mengungkapkan interaksi di antara subsistem penghasil, transmisi, atau pemrosesan data ini dalam sistem Internet seluler. Pada Gambar 1.7, kami memberi label tepi antara subsistem dengan tindakan yang terjadi di antara mereka. Kami secara singkat memperkenalkan tindakan interaktif di bawah ini untuk lima tujuan: i) penginderaan sinyal data terkait dengan interaksi antara IoT dan jejaring sosial dengan platform cloud; ii) penambangan data melibatkan penggunaan kekuatan cloud untuk penggunaan data yang diambil secara efektif; iii) agregasi data terjadi antara sistem seluler; iv) domain IoT; dan vi) cloud pemrosesan. *Machine learning* membentuk dasar untuk analitik *Big data*.

Interaksi antar Teknologi

Sejumlah besar data sensor atau sinyal digital dihasilkan oleh sistem seluler, jejaring sosial, dan berbagai domain IoT. Penginderaan RFID, jaringan sensor dan data yang dihasilkan GPS diperlukan untuk menangkap data secara tepat waktu dan selektif, jika data tidak terstruktur terganggu oleh kebisingan atau kehilangan udara. Penginderaan IoT menuntut kualitas data yang tinggi, dan pemfilteran sering digunakan untuk meningkatkan kualitas data. Bab 3 didedikasikan untuk berbagai operasi penginderaan dalam sistem IoT:

- **Data Mining:** Data mining melibatkan penemuan, pengumpulan, agregasi, transformasi, pencocokan, dan pemrosesan kumpulan *Big data*. Penambangan data adalah operasi mendasar yang terjadi dengan sistem informasi *Big data*. Tujuan utamanya adalah penemuan pengetahuan dari data. Data numerik, tekstual, pola, gambar dan video dapat ditambang. Bab 2 akan membahas inti dari penambangan *Big data* pada khususnya.
- **Agregasi dan Integrasi Data:** Ini mengacu pada pra-pemrosesan data untuk meningkatkan kualitas data. Operasi penting termasuk pembersihan data, menghilangkan redundansi, memeriksa relevansi, reduksi data, transformasi dan diskritisasi, dll.
- ***Machine learning* dan Analisis *Big data*:** Ini adalah dasar untuk menggunakan kekuatan computing cloud untuk menganalisis kumpulan *Big data* secara ilmiah atau statistik. Program komputer khusus ditulis untuk secara otomatis belajar mengenali pola kompleks dan membuat keputusan cerdas berdasarkan data. Bab 4, 5, dan 8 akan membahas *Machine learning* dan analitik *Big data*.

Penggabungan Teknologi untuk Memenuhi Permintaan Masa Depan

IoT memperluas Internet komputer ke objek apa pun. Penggunaan bersama cloud, IoT, perangkat seluler, dan jejaring sosial sangat penting untuk menangkap *Big data* dari semua sumber. Sistem terintegrasi ini dibayangkan oleh para peneliti IBM sebagai "bumi pintar" [22], yang memungkinkan interaksi yang cepat, efisien, dan cerdas antara manusia, mesin, dan objek apa pun di sekitar kita. Bumi yang cerdas harus memiliki kota yang cerdas, air bersih, listrik yang efisien, transportasi yang nyaman, persediaan makanan yang aman, bank yang bertanggung jawab, telekomunikasi yang cepat, TI hijau, sekolah yang lebih baik, perawatan kesehatan, dan sumber daya yang melimpah untuk dibagikan. Ini terdengar seperti mimpi, yang belum menjadi kenyataan di tahun-tahun mendatang.

Secara umum, teknologi yang matang seharusnya diadopsi dengan cepat. Penggunaan gabungan dari dua atau lebih teknologi mungkin memerlukan upaya tambahan untuk mengintegrasikannya untuk tujuan bersama. Jadi integrasi mungkin menuntut beberapa perubahan transformasional. Untuk memungkinkan aplikasi baru yang inovatif, transformasi teknologi inti menghadirkan tantangan. Teknologi yang mengganggu bahkan lebih sulit untuk diintegrasikan karena risiko yang lebih tinggi. Mereka mungkin menuntut lebih banyak penelitian dan eksperimen atau upaya pembuatan prototipe. Ini membawa kita untuk mempertimbangkan perpaduan teknologi dengan memadukan teknologi yang berbeda untuk saling melengkapi.

Kelima teknologi SMACT digunakan dalam Internet seluler (juga dikenal sebagai Internet nirkabel). Jaringan IoT dapat muncul dalam berbagai bentuk di domain aplikasi yang berbeda. Misalnya, kami dapat membangun domain IoT untuk pertahanan nasional, perawatan kesehatan, energi hijau, media sosial dan kota pintar, dll. Jaringan sosial dan subsistem analisis *Big data* dibangun di Internet dengan pencarian basis data yang cepat dan fasilitas akses seluler.

Daya penyimpanan dan pemrosesan yang tinggi disediakan oleh layanan cloud khusus domain pada platform khusus. Kami masih memiliki jalan panjang sebelum kami melihat

meluasnya penggunaan platform cloud khusus domain untuk *Big data* atau aplikasi IoT di lingkungan Internet seluler.

1.2 MEDIA SOSIAL, JARINGAN SELULER, DAN COMPUTING CLOUD

Bagian ini memberikan gambaran umum tentang jaringan sosial, perangkat seluler, dan jaringan akses radio dari segala jenis untuk komunikasi dan pergerakan data jarak pendek dan luas. Computing cloud sosial dan seluler akan dinilai. Perlakuan lebih rinci dari topik ini dapat ditemukan di Bab 4, 7, 8 dan 9.

Tabel 1.4 Ringkasan jejaring sosial populer dan layanan web yang disediakan

Jejaring Sosial, Tahun dan Situs Web	Pengguna Aktif Terdaftar	Layanan Utama yang Disediakan
Facebook, 2004 http://www.facebook.com	1,65 miliar pengguna, 2016	Berbagi konten, membuat profil, iklan, acara, perbandingan sosial, komunikasi, bermain game sosial, dll.
Tencent QQ di Cina, 1999 http://www.qq.com	853 juta pengguna, 2016	Layanan pesan instan, game online, musik, ebQQ, belanja, microblogging, film, WeChat, QQ Player, dll.
Linkedin, 2002, http://www.linkedin.com	364 juta pengguna, 2015	Layanan profesional, perekrutan online, daftar pekerjaan, layanan grup, keterampilan, penerbitan, pengaruh, iklan, dll.
Twitter, 2006 http://www.twitter.com	320 juta pengguna, 2016	Microblogging, berita, peringatan, pesan singkat, peringkat, demografi, sumber pendapatan, berbagi foto, dll.

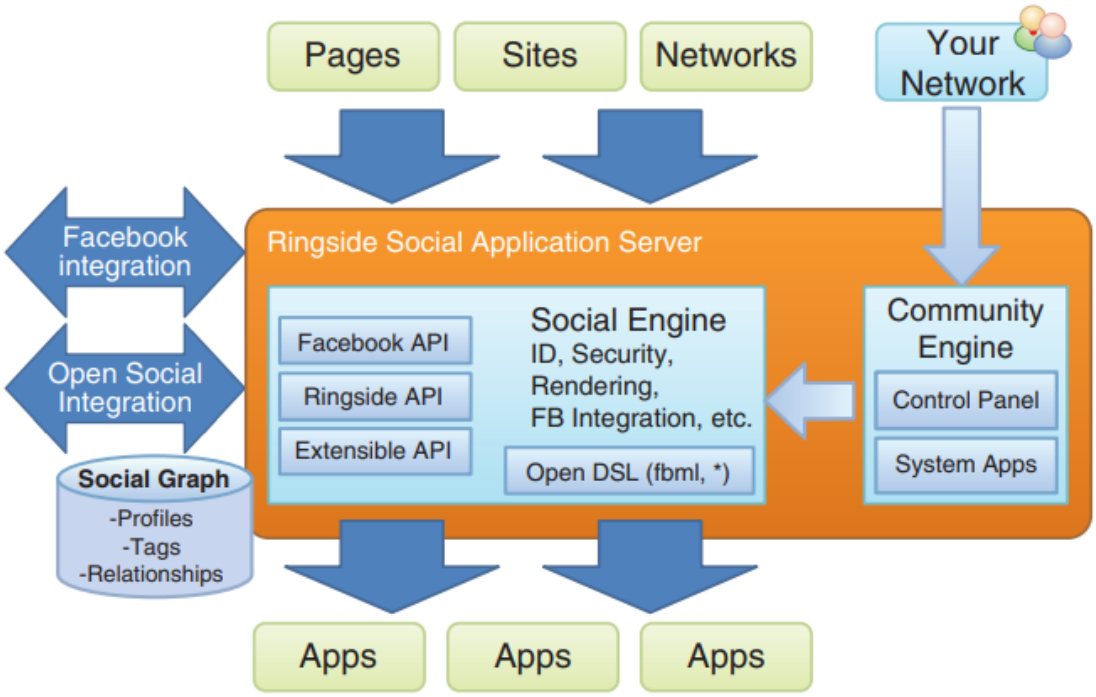
Jejaring Sosial dan Situs Layanan Web

Sebagian besar jejaring sosial menyediakan layanan manusia seperti koneksi pertemanan, profil pribadi, layanan profesional, hiburan, dll. Secara umum, pengguna harus mendaftar menjadi anggota untuk mengakses situs web. Pengguna dapat membuat profil pengguna, menambahkan pengguna lain sebagai “teman”, bertukar pesan, memposting pembaruan status dan foto, berbagi video, dan menerima pemberitahuan saat orang lain memperbarui profilnya. Selain itu, pengguna dapat bergabung dengan grup pengguna dengan minat yang sama, yang diatur berdasarkan tempat kerja, sekolah atau perguruan tinggi, atau karakteristik lainnya, dan mengkategorikan teman mereka ke dalam daftar seperti "Orang Dari Tempat Kerja" atau "Teman Dekat", dll. Pada Tabel 1.4, kami bandingkan beberapa jejaring sosial populer dan perkenalkan layanan mereka secara singkat.

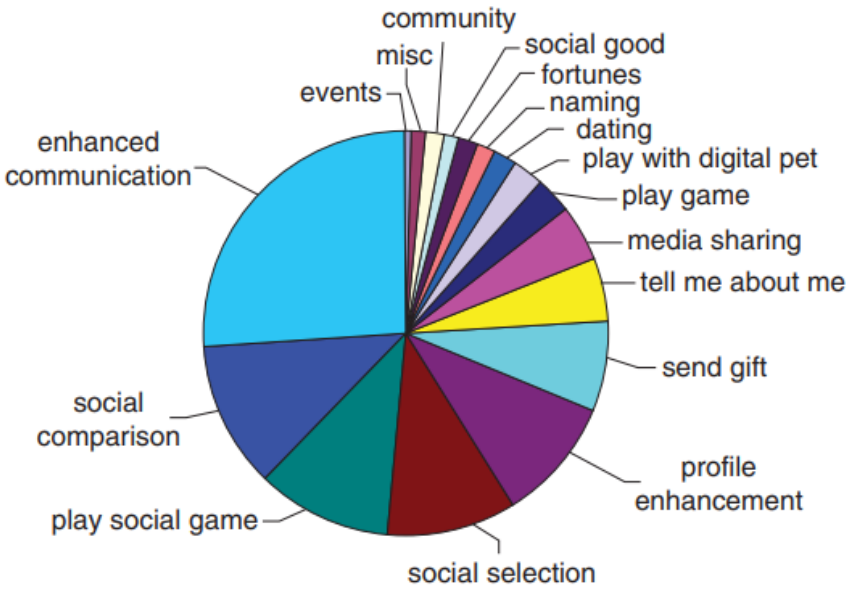
Facebook sejauh ini merupakan penyedia layanan jejaring sosial terbesar, dengan lebih dari 1,65 miliar pengguna. Jaringan Tencent QQ adalah jaringan sosial terbesar kedua yang berbasis di Cina. Jaringan QQ memiliki lebih dari 800 juta pengguna. Ini benar-benar Facebook di Cina, dengan layanan yang diperluas seperti akun email, hiburan, dan bahkan beberapa operasi bisnis web. Linkedin adalah jaringan sosial berorientasi bisnis yang menyediakan layanan profesional. Ini sangat digunakan oleh perusahaan bisnis besar dalam merekrut dan mencari bakat. Twitter menawarkan pesan teks pendek dan layanan blogging terbesar saat ini. Situs lain adalah jaringan belanja online atau terkait dengan kelompok minat khusus.

Contoh 1.1 Arsitektur Platform Facebook dan Layanan Sosial yang Disediakan

Dengan 1,65 miliar pengguna aktif di seluruh dunia pada tahun 2016, Facebook menyimpan profil, tag, dan hubungan pribadi yang besar sebagai grafik sosial. Sebagian besar pengguna berada di AS, Brasil, India, Indonesia, dll. Grafik sosial dibagikan oleh berbagai grup sosial di situs. Situs web ini telah menarik lebih dari 3 juta pengiklan aktif dengan pendapatan $12,5 miliar yang dilaporkan pada tahun 2014. Platform Facebook dibangun dengan kumpulan pusat *Big data* dengan kapasitas penyimpanan yang sangat besar, sistem file yang cerdas, dan kemampuan pencarian. Web harus mengatasi kemacetan lalu lintas dan tabrakan di antara semua penggunanya. Pada Gambar 1.8(a), infrastruktur platform Facebook ditampilkan.



(a) infrastruktur Facebook



(b) distribusi aplikasi Facebook

Gambar 1.8 Platform Facebook menawarkan lebih dari 2,4 juta aplikasi pengguna.

Platform ini dibentuk dengan sekelompok besar server. Permintaan ditampilkan sebagai halaman, situs, dan jaringan yang memasuki server Facebook dari atas. Mesin sosial adalah inti dari server aplikasi. Mesin sosial ini menangani operasi IS, keamanan, rendering, dan integrasi Facebook. Sejumlah besar API tersedia untuk memberi manfaat bagi pengguna yang menggunakan lebih dari 2,4 juta aplikasi. Facebook telah mengakuisisi aplikasi Insragram, WhatsApp, Qculus VR dan PrivateCore. Mesin sosial menjalankan semua aplikasi pengguna. Open DSL digunakan untuk mendukung eksekusi aplikasi. Fungsionalitas layanan Facebook mencakup enam item penting, seperti yang dirangkum dalam Tabel 1.5. Facebook menyediakan blog, obrolan, hadiah, pasar, panggilan suara/video, dll. Gambar 1.8(b) menunjukkan distribusi layanan Facebook. Ada mesin komunitas yang menyediakan layanan jaringan kepada pengguna. Sebagian besar aplikasi Facebook membantu pengguna untuk mencapai tujuan sosial mereka, seperti komunikasi yang lebih baik, belajar tentang diri sendiri, menemukan orang lain yang serupa, terlibat dalam permainan sosial dan pertukaran. Oleh karena itu, Facebook lebih menarik di domain pribadi dan pribadi.

Tabel 1.5 Fungsionalitas layanan platform Facebook.

Fungsi	**Deskripsi Singkat**
Halaman Profil	Gambar profil, informasi bio, daftar teman, log aktivitas pengguna, pesan publik

Grafik Traversal	Akses melalui daftar teman pengguna di halaman profil, dengan kontrol akses
Komunikasi	Kirim dan terima pesan di antara teman, pesan instan, dan blog mikro
Item yang Dibagikan	Album foto dengan kontrol akses bawaan, video luar yang disematkan di halaman profil
Kontrol akses	Level kontrol akses: Hanya saya, Hanya teman, Teman dari teman, dan Semua Orang
API khusus	Game, kalender, klien seluler, dll.

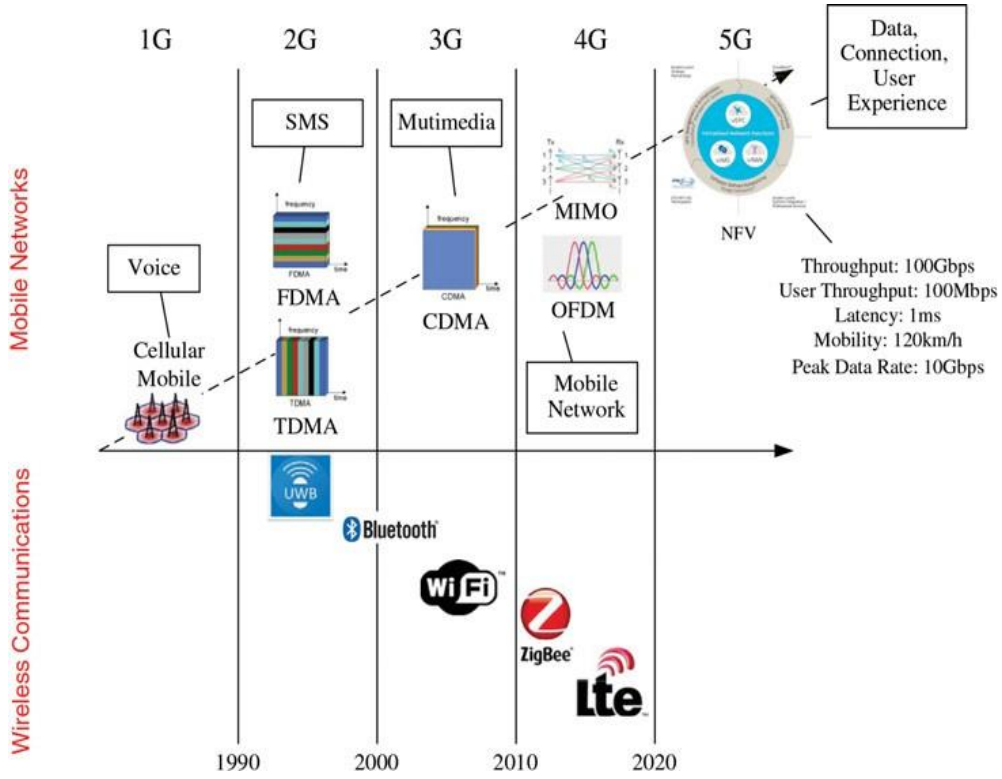
Jaringan Inti Seluler Seluler

Jaringan seluler atau jaringan seluler adalah jaringan nirkabel yang didistribusikan melalui area daratan yang disebut sel, masing-masing dilayani oleh setidaknya satu transceiver lokasi tetap, yang dikenal sebagai situs sel atau stasiun pangkalan. Dalam jaringan seluler, setiap sel menggunakan rangkaian frekuensi yang berbeda dari sel tetangga, untuk menghindari interferensi dan menyediakan bandwidth yang terjamin di dalam setiap sel. Sistem komunikasi seluler telah merevolusi cara orang berkomunikasi, menggabungkan komunikasi dan mobilitas. Gambar 1.9 menunjukkan kemajuan jaringan inti seluler untuk komunikasi jarak jauh, yang telah melalui lima generasi pengembangan, sementara komunikasi nirkabel jarak pendek juga telah ditingkatkan dalam kecepatan data, QoS, dan aplikasi selama periode yang sama.

Evolusi teknologi akses nirkabel baru saja memasuki generasi keempat (4G). Melihat masa lalu, teknologi akses nirkabel telah mengikuti jalur evolusi berbeda yang ditujukan untuk kinerja dan efisiensi di lingkungan seluler yang tinggi. Generasi pertama (1G) telah memenuhi kebutuhan dasar komunikasi suara bergerak, sedangkan generasi kedua (2G) telah memperkenalkan kapasitas dan jangkauan. Generasi ketiga (3G) adalah pencarian data dengan kecepatan lebih tinggi untuk membuka gerbang pengalaman “broadband seluler” yang sesungguhnya. Generasi keempat (4G) menyediakan akses ke berbagai layanan telekomunikasi, termasuk layanan seluler tingkat lanjut, didukung oleh jaringan seluler dan tetap, yang sepenuhnya bersifat packet-switched dengan mobilitas dan kecepatan data yang tinggi.

Saat industri komunikasi seluler melakukan perjalanan jauh dari 2G ke 4G, sekarang 5G bertujuan untuk mengubah dunia dengan menghubungkan apa pun ke apa pun. Berbeda dari versi sebelumnya, penelitian 5G tidak hanya berfokus pada pita spektrum baru, transmisi nirkabel, jaringan seluler, dll, untuk peningkatan kapasitas. Ini akan menjadi teknologi cerdas untuk menghubungkan dunia nirkabel tanpa hambatan. Untuk memenuhi persyaratan 5G untuk memungkinkan kapasitas yang lebih tinggi, kecepatan yang lebih tinggi, konektivitas yang lebih banyak, keandalan yang lebih tinggi, latensi yang lebih rendah, keserbagunaan yang lebih besar, dan topologi khusus domain aplikasi, diperlukan konsep dan pendekatan desain baru. Pekerjaan standarisasi saat ini untuk 4G dapat memengaruhi pengenalan fitur radio dan solusi jaringan yang menjanjikan untuk sistem 5G.

Arsitektur jaringan baru, melampaui jaringan heterogen dan mengeksploitasi spektrum frekuensi baru (misalnya mmWave), muncul dari laboratorium penelitian di seluruh dunia. Selain sisi jaringan, terminal dan receiver canggih sedang dikembangkan untuk mengoptimalkan kinerja jaringan. Memisahkan bidang kontrol dan data (saat ini dipelajari dalam 3GPP) adalah paradigma yang menarik untuk 5G, bersama-sama dengan multi-input multi-output (MIMO) besar-besaran, sistem antena canggih, jaringan yang ditentukan perangkat lunak (SDN), Virtualisasi Fungsi Jaringan (NFV), Internet of Things (IoT) dan computing cloud.



Gambar 1.9 Jaringan inti seluler untuk komunikasi jarak jauh telah melewati lima generasi, sementara jaringan nirkabel jarak pendek ditingkatkan dalam kecepatan data, QoS, dan aplikasi.

Perangkat Seluler dan Jaringan Tepi Internet

Perangkat seluler muncul sebagai ponsel pintar, komputer tablet, peralatan yang dapat dikenakan, dan alat industri. Pengguna global perangkat seluler melebihi 3 miliar pada tahun 2015. Perangkat 1G, digunakan pada 1980-an, kebanyakan telepon analog untuk komunikasi suara saja. Jaringan seluler 2G dimulai pada awal 1990-an. Telepon digital muncul sesuai untuk komunikasi suara dan data. Seperti yang ditunjukkan pada Gambar 1.9, jaringan seluler 2G muncul sebagai GSM, TDMA, FDMA dan CDMA, berdasarkan skema pembagian yang berbeda untuk memungkinkan beberapa penelepon mengakses sistem secara bersamaan. Jaringan 2G dasar mendukung data 9,6 Kbps dengan circuit switching. Kecepatan ditingkatkan menjadi 115

Kbps dengan layanan radio paket. Hingga tahun 2015, jaringan 2G masih digunakan di banyak negara berkembang.

Sejak tahun 2000, perangkat seluler 2G secara bertahap digantikan oleh produk 3G. Jaringan 3G dan telepon dirancang untuk memiliki kecepatan 2 Mbps untuk memenuhi kebutuhan komunikasi multimedia melalui sistem selular. Jaringan 4G LTE (Long Term Evolution) muncul pada tahun 2000-an. Mereka ditargetkan mencapai kecepatan unduh 100 Mbps, kecepatan unggah 50 Mbps, dan kecepatan statis 1 Gbps. Sistem 3G diaktifkan oleh teknologi radio yang lebih baik dengan antena pintar MIMO dan teknologi OFDM. Sistem 3G telah menerima penyebaran luas sekarang, tetapi dapat digantikan secara bertahap oleh jaringan 4G. Kami mengharapkan penggunaan campuran jaringan 3G dan 4G setidaknya untuk satu dekade lagi. Jaringan 5G mungkin muncul setelah tahun 2020 dengan kecepatan target minimal 100 Gbps.

Jaringan Inti Seluler

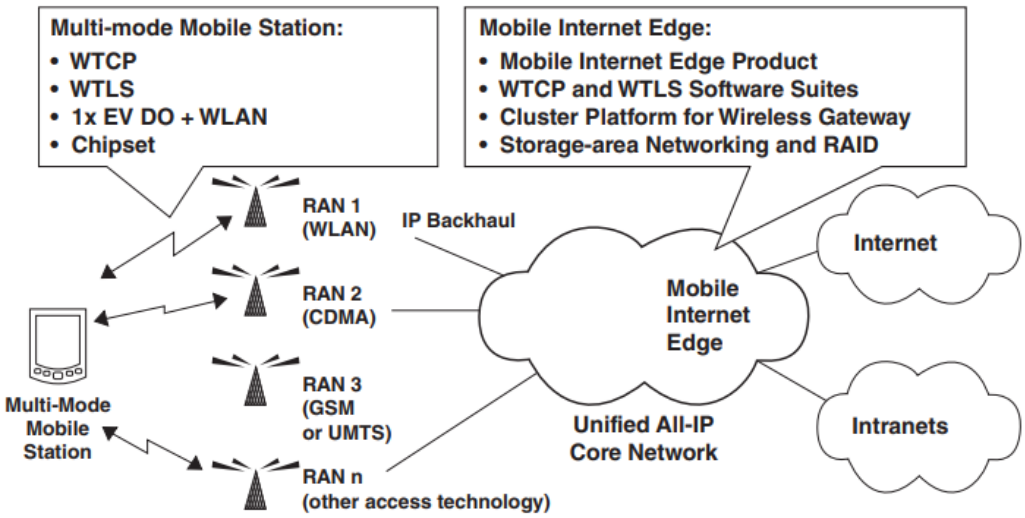
Jaringan akses radio seluler (RAN) terstruktur secara hierarkis. Jaringan inti seluler membentuk tulang punggung sistem telekomunikasi saat ini. Jaringan inti telah melalui empat generasi penyebaran dalam tiga dekade terakhir. Jaringan seluler 1G digunakan untuk komunikasi suara analog berdasarkan teknologi circuit switching. Jaringan seluler 2G dimulai pada awal 1990-an untuk mendukung penggunaan telepon digital di telekomunikasi suara dan data yang menjelajahi sirkuit switching paket. Sistem 2G yang terkenal adalah GSM (Global System for Mobile Communications) yang dikembangkan di Eropa dan sistem CDMA (Code Division Multiple Access) yang dikembangkan di AS. Baik sistem GSM maupun CDMA tersebar di berbagai negara.

Jaringan seluler 3G dikembangkan untuk komunikasi suara/data multimedia dengan layanan jelajah global. Sistem 4G dimulai pada awal 2000-an berdasarkan teknologi radio LTE dan MIMO. Jaringan seluler 5G masih dalam pengembangan besar, yang mungkin muncul pada tahun 2020. Teknologi, kecepatan data puncak, dan aplikasi yang digerakkan dari lima generasi jaringan seluler seluler dirangkum dalam Tabel 1.6. Secara kecepatan, sistem seluler meningkat dari 1 Kbps menjadi 10 Kbps, 2 Mbps, dan 100 Mbps dalam empat generasi. Diproyeksikan bahwa sistem 5G yang akan datang dapat mencapai peningkatan 1000 dalam kecepatan data hingga 100 Gbps atau lebih tinggi. Sistem 5G dapat dibangun dengan kepala radio jarak jauh (RRH) dan stasiun pangkalan virtual yang dipasang di CRAN (Jaringan Akses Radio Berbasis Cloud).

Tabel 1.6 Tonggak sejarah jaringan inti seluler untuk telekomunikasi seluler.

Generasi	1G	2G	3G	4G	5G
Teknologi Radio dan Jaringan	Telepon analog, AMPS, TDMA	Telepon Digital GSM, CDMA	CDMA2000, WCDMA, dan D-SCDMA	LTE, OFDM, MIMO, perangkat lunak - radio	LTE, RAN berbasis Cloud

				yang dikendalikan RAN	
Kecepatan Data Seluler Puncak	8 Kbps	9.6–344 Kbps	2 Mbps	100 Mbps	10 Gbps-1 Tbps
Aplikasi Mengemudi	Komunikasi Suara	Komunikasi Suara/Data	Komunikasi Multimedia	Komunikasi pita lebar	Komunikasi ultra-kecepatan



Gambar 1.10 Interaksi berbagai jaringan akses radio (RAN) dengan jaringan inti seluler berbasis IP terpadu, Intranet, dan Internet.

Jaringan Tepi Internet Seluler

Sebagian besar jaringan nirkabel dan seluler saat ini didasarkan pada transmisi dan penerimaan sinyal radio di berbagai rentang operasi. Kami menyebutnya jaringan akses radio (RAN). Gambar 1.10 mengilustrasikan bagaimana berbagai RAN digunakan untuk mengakses jaringan inti seluler, yang terhubung ke tulang punggung Internet dan banyak Intranet melalui jaringan tepi Internet seluler. Infrastruktur akses Internet semacam itu juga dikenal sebagai Internet nirkabel atau Internet seluler oleh komunitas computing yang meresap. Berikut ini, kami memperkenalkan beberapa kelas RAN yang dikenal sebagai jaringan WiFi, Bluetooth, WiMax dan Zigbee. Secara umum, kami mempertimbangkan beberapa jaringan nirkabel jarak pendek, seperti jaringan area lokal nirkabel (WLAN), jaringan area rumah nirkabel (WHAN), jaringan area pribadi (PAN) dan jaringan area tubuh (BAN), dll. Jaringan nirkabel ini memainkan peran kunci dalam computing seluler dan aplikasi IoT.

Perangkat dan Jaringan Bluetooth

Bluetooth adalah teknologi radio jarak pendek yang dinamai sesuai nama Raja Denmark yang berasal dari abad ke-9. Perangkat Bluetooth beroperasi pada pita medis ilmiah industri 2,45 GHz, sebagaimana ditentukan oleh Standar IEEE 802.15.1. Ini mentransmisikan sinyal omni-directional (360◦) tanpa batas pada garis pandang, yang berarti data atau suara dapat menembus objek non-logam padat. Mendukung hingga 8 perangkat (1 master dan 7 slave) dalam PAN yang disebut Piconet. Perangkat Bluetooth memiliki biaya rendah dan kebutuhan daya rendah. Perangkat ini menawarkan kecepatan data 1 Mbps dalam jaringan ad hoc dengan jangkauan 10 cm hingga 10 meter. Mendukung komunikasi suara atau data antara ponsel, komputer, dan perangkat wearable lainnya. Pada dasarnya, koneksi nirkabel Bluetooth menggantikan sebagian besar kabel kabel antara komputer dan periferalnya seperti mouse, keyboard, dan printer, dll.

Jaringan WiFi

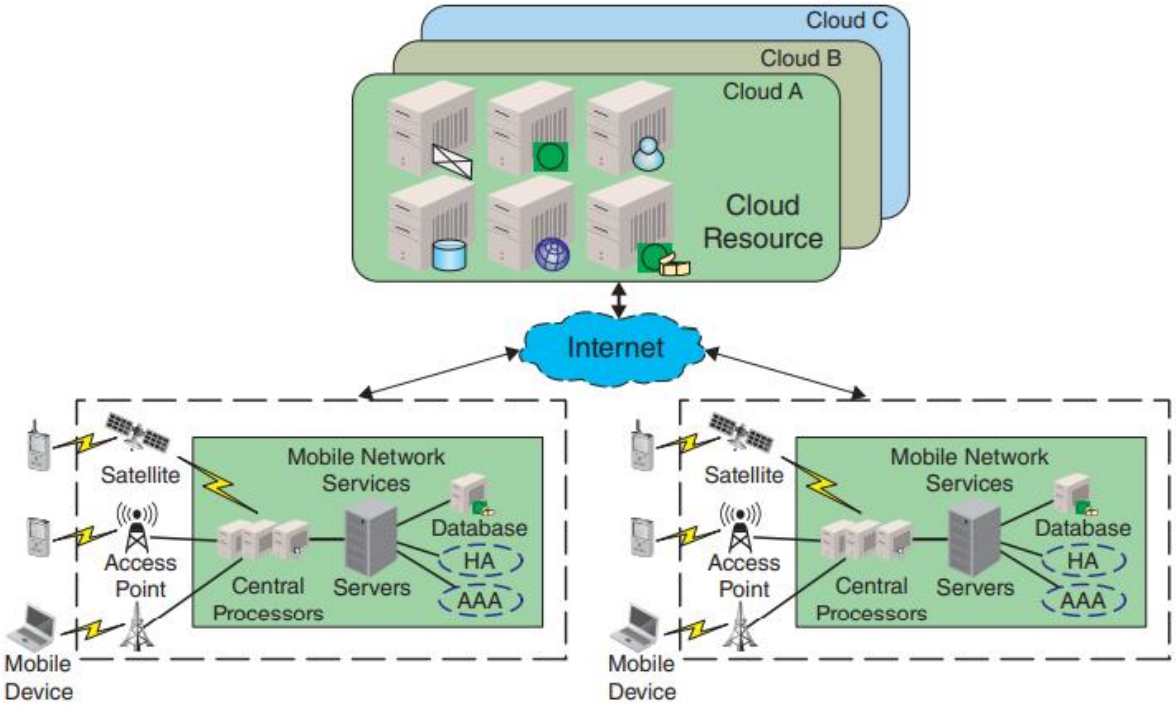
Titik akses WiFi atau jaringan WiFi ditentukan dalam Standar IEEE 802.11. Sejauh ini, mereka telah muncul sebagai rangkaian 11 a, b, g, n, ac dan jaringan ay. Titik akses memancarkan sinyalnya dalam radius kurang dari 300 kaki. Semakin dekat dengan titik akses, semakin cepat kecepatan data yang dialami. Kecepatan maksimum hanya dimungkinkan dalam jarak 50–175 kaki. Kecepatan data puncak jaringan WiFi telah meningkat dari kurang dari 11 Mbps pada jaringan 11b menjadi 54 Mbps pada jaringan 11g dan 300 Mbps pada jaringan 11n. Jaringan 11n dan 11ac menerapkan teknologi modulasi OFDG dengan penggunaan radio dan antena multiple input and multiple output (MIMO) untuk mencapai kecepatan tinggi. WiFi memungkinkan WLAN tercepat dalam jaringan titik akses atau router nirkabel. Mereka menawarkan akses hampir gratis ke Internet dalam jarak 300 kaki di banyak lokasi saat ini.

Infrastruktur Computing Cloud Seluler

Perangkat seluler dengan cepat menjadi peserta layanan utama saat ini. Ada pergeseran preferensi pengguna dari ponsel dan laptop tradisional ke ponsel pintar dan tablet. Kemajuan dalam portabilitas dan kemampuan perangkat seluler, bersama dengan jaringan 3G/4G LTE yang tersebar luas dan akses Wi-Fi, telah menghadirkan pengalaman aplikasi seluler yang kaya bagi pengguna akhir. *Computing cloud* seluler adalah model untuk peningkatan elastis kemampuan perangkat seluler melalui akses nirkabel di mana-mana ke penyimpanan cloud dan sumber daya computing. Ini lebih ditingkatkan dengan adaptasi dinamis konteks-sadar terhadap perubahan dalam lingkungan operasi. Gambar 1.11 menunjukkan lingkungan seluler yang khas untuk memindahkan pekerjaan besar ke cloud jarak jauh dari pemegang perangkat seluler.

Dengan dukungan *mobile cloud computing* (MCC), pengguna ponsel pada dasarnya memiliki opsi cloud baru untuk menjalankan aplikasinya. Pengguna mencoba untuk menurunkan beban computing melalui WiFi, jaringan seluler atau satelit ke cloud yang jauh. Perangkat terminal di ujung pengguna memiliki sumber daya yang terbatas, yaitu perangkat keras, energi, bandwidth, dll. Ponsel itu sendiri tidak layak untuk menyelesaikan beberapa tugas computing intensif. Sebagai gantinya, data yang terkait dengan tugas computing diturunkan ke cloud jarak

jauh. Cloudlet khusus diperkenalkan untuk berfungsi sebagai gateway nirkabel antara pengguna seluler dan Internet. Cloudlet ini dapat digunakan untuk membongkar computing atau layanan web ke cloud jarak jauh dengan aman. Detail cloudlet untuk *mobile cloud* akan dijelaskan di Bab 2.



Gambar 1.11 Arsitektur lingkungan computing cloud bergerak.

1.3 AKUISISI *BIG DATA* DAN EVOLUSI ANALISIS

Analisis *Big data* adalah proses pemeriksaan sejumlah besar data dari berbagai jenis (*Big data*) untuk mengungkap pola tersembunyi, korelasi yang tidak diketahui, dan informasi berguna lainnya. Informasi tersebut dapat memberikan keunggulan kompetitif atas organisasi saingan dan menghasilkan kecerdasan bisnis atau penemuan ilmiah yang lebih tinggi, seperti pemasaran yang lebih efektif, peningkatan pendapatan, dll. Tujuan utama analitik *Big data* adalah untuk membantu perusahaan membuat keputusan bisnis yang lebih baik dengan mengaktifkan data ilmuwan dan pengguna lain untuk menganalisis volume besar data transaksi yang mungkin tidak dimanfaatkan oleh program business intelligence (BI) konvensional.

Rantai Nilai *Big data* Diekstraksi dari Data Massif

Ilmu data, penambangan data, analitik data, dan penemuan pengetahuan adalah istilah yang terkait erat. Dalam banyak kasus, mereka digunakan secara bergantian. Komponen *Big data* ini membentuk rantai nilai *Big data* yang dibangun dari statistik, *Machine learning*, biologi, dan metode kernel. Statistik mencakup regresi linier dan logistik. Pohon keputusan adalah alat *Machine learning* yang khas. Biologi mengacu pada jaringan saraf tiruan, algoritma genetika, dan

kecerdasan swarm. Terakhir, metode kernel mencakup penggunaan mesin vektor dukungan. Teori dan model yang mendasari ini akan dipelajari dalam Bab 4, 5 dan 6. Penerapannya akan dibahas dalam Bab 7, 8 dan 9.

Dibandingkan dengan kumpulan data tradisional, *Big data* umumnya mencakup kumpulan data tidak terstruktur yang membutuhkan lebih banyak analisis waktu nyata. Selain itu, *big data* juga membawa peluang baru untuk menemukan nilai-nilai baru, membantu kita memperoleh pemahaman mendalam tentang nilai-nilai yang tersembunyi, dan menimbulkan tantangan baru, misalnya tentang bagaimana mengatur dan mengelola data tersebut secara efektif. Saat ini, *big data* telah menarik minat yang cukup besar dari industri, akademisi dan instansi pemerintah. Dewasa ini, pertumbuhan *big data* yang pesat terutama berasal dari kehidupan sehari-hari masyarakat, terutama yang terkait dengan Internet, Web, dan layanan cloud.

Contoh 1.2 Pertumbuhan *Big data* yang Diharapkan dari 2010 hingga 2020 dan Keuntungan Ekonomi Ukuran kumpulan *Big data* bervariasi dari waktu ke waktu. Tabel 1.7 menunjukkan beberapa ukuran data yang representatif dari 2010 hingga 2020. Angka-angka ini hanya memberi kesan kepada pembaca tentang peningkatan yang stabil seiring waktu. Menurut standar 2015, kumpulan data 1 TB diberi label sebagai *Big data*. Volume besar membawa nilai keuntungan ekonomi seperti yang terungkap di baris bawah. Aplikasi yang peka terhadap lokasi saja dapat menghasilkan pendapatan Rp 12.000.000 miliar dalam 10 tahun. *Big data* yang diterapkan untuk perawatan kesehatan dapat menghemat Rp 4.500.000 miliar dalam biaya pengobatan di AS. Industri *Big data* secara bertahap terbentuk dari waktu ke waktu.

Tabel 1.7 Pertumbuhan *big data* dari 2010 hingga 2020 dan nilai ekonomi yang diharapkan dalam dua aplikasi *big data* yang khas.

Sumber *Big data*, Pertumbuhan yang Diamati dalam Ukuran dan Tahun Data	**Ukuran Data**
Data global dihasilkan dalam 2 hari di tahun 2011	1,82 ZB
Gambar diunggah ke Facebook pada tahun 2014	759 juta keping
Kapasitas penyimpanan industri manufaktur Amerika pada tahun 2010	966 PB
Jumlah tag RFID yang dipindai pada 2011 hingga 2020	12 juta hingga 209 miliar
Data diambil selama 2,450 juta jam geek komputer	200 TB
Layanan data lokasi pribadi dapat mencapai Rp 12.000.000 miliar dolar dalam 10 tahun Penghematan dari analisis dan perawatan kesehatan dapat melebihi Rp 4.500.000 miliar dolar di AS	
Sumber *Big data*, Pertumbuhan yang Diamati dalam Ukuran dan Tahun Data	

Saat ini, data telah menjadi faktor produksi penting yang dapat disamakan dengan aset material dan modal manusia. Karena multimedia, media sosial, dan IoT berkembang pesat, perusahaan bisnis akan mengumpulkan lebih banyak informasi, yang mengarah ke pertumbuhan volume data yang eksponensial. *Big data* akan memiliki potensi yang besar dan terus meningkat dalam menciptakan nilai bagi bisnis dan konsumen. Aspek terpenting dari analitik *Big data* adalah nilai *Big data*. Kami membagi rantai nilai *Big data* menjadi empat fase: pembuatan data, akuisisi data, penyimpanan data, dan analisis data. Jika kita mengambil data sebagai bahan mentah, pembuatan data dan akuisisi data adalah proses eksploitasi, karena penyimpanan data harus menggunakan cloud atau pusat data. Analisis data adalah proses produksi yang memanfaatkan bahan mentah untuk menciptakan nilai baru.

Pesatnya pertumbuhan computing cloud dan IoT juga memicu pertumbuhan data yang tajam. Computing cloud menyediakan perlindungan, akses situs, dan saluran untuk aset data. Dalam paradigma IoT, sensor di seluruh dunia mengumpulkan dan mengirimkan data untuk disimpan dan diproses di cloud. Data tersebut dalam kuantitas dan hubungan timbal balik akan jauh melampaui kapasitas arsitektur TI dan infrastruktur perusahaan yang ada, dan persyaratan real-time akan sangat menekankan kapasitas computing yang tersedia. Contoh berikut menyoroti beberapa nilai *Big data* representatif yang didorong oleh volume *Big data* yang terlibat.

Generasi Big data

Jenis data utama termasuk data Internet, data sensorik, dll. Ini adalah langkah pertama dari *Big data*. Mengingat data Internet sebagai contoh, sejumlah besar data dalam hal entri pencarian, posting forum Internet, catatan obrolan dan pesan mikroblog, dihasilkan. Data tersebut terkait erat dengan kehidupan sehari-hari masyarakat, dan memiliki kesamaan fitur bernilai tinggi dan kepadatan rendah. Data Internet semacam itu mungkin tidak berharga secara individual tetapi, melalui eksploitasi akumulasi *Big data*, informasi yang berguna seperti kebiasaan dan hobi pengguna dapat diidentifikasi, dan bahkan memungkinkan untuk meramalkan perilaku dan suasana hati pengguna.

Selain itu, dihasilkan melalui sumber data longitudinal dan/atau terdistribusi, kumpulan data lebih berskala besar, sangat beragam, dan kompleks. Sumber data tersebut mencakup sensor, video, aliran klik, dan/atau semua sumber data lain yang tersedia. Saat ini, sumber utama *Big data* adalah informasi operasi dan perdagangan di perusahaan bisnis, logistik dan informasi penginderaan di IoT, informasi interaksi manusia dan informasi posisi di dunia Internet, dan data yang dihasilkan dalam penelitian ilmiah, dll.

Kontrol Kualitas Data, Representasi, dan Model Basis Data

Pada Tabel 1.8, kami merangkum properti dan atribut menarik yang memengaruhi kualitas data. Kami memperkenalkan metode, arsitektur, dan alat untuk analisis *Big data*. Studi kami tidak berarti mencakup semua kemajuan yang dibuat di bidang ini. Kami mengidentifikasi konsep kunci dan beberapa alat representatif atau model basis data yang digunakan dalam

konteks ini. Sumber *Big data* berasal dari transaksi bisnis, konten tekstual dan multimedia, data pengetahuan kualitatif, penemuan ilmiah, media sosial, dan data penginderaan dari IoT. Kualitas data seringkali buruk karena volume yang besar, variasi data karena tipe data yang tidak dapat diprediksi, dan kebenaran data karena kurangnya ketertelusuran.

Kontrol kualitas *Big data* melibatkan siklus melingkar dari empat tahap: i) kita harus mengidentifikasi atribut kualitas data yang penting; ii) untuk mengakses data bergantung pada kemampuan untuk mengukur atau menilai tingkat kualitas data; iii) maka kita harus mampu menganalisis kualitas data dan penyebab utamanya; dan terakhir iv) kita perlu meningkatkan kualitas data dengan menyarankan tindakan nyata yang harus diambil. Sayangnya, tidak satu pun dari tugas ini yang mudah diterapkan. Pada Tabel 1.8, kami mengidentifikasi atribut penting terhadap kontrol kualitas data. Di antara dimensi kontrol kualitas data ini, atribut intrinsik dan representasional dan mekanisme kontrol akses sama pentingnya.

Data dapat direpresentasikan dengan berbagai cara. Empat model representasi utama disarankan untuk *Big data*: i) pasangan <key, value> sering digunakan untuk mendistribusikan data dalam operasi MapReduce (akan disajikan di Bab 5). Dynamo Volldemort adalah contoh yang baik untuk menggunakan pasangan nilai kunci; ii) pencarian tabel atau database relasional seperti Perangkat lunak BigTable dan Cassandra Google; iii) alat grafik seperti GraphX yang digunakan di Spark untuk analisis grafik sosial, dan iv) sistem basis data khusus seperti MongoDB, SimpkleDB dan CouchDB yang biasa digunakan oleh komunitas *Big data*.

Tabel 1.8 Atribut untuk kontrol kualitas data, representasi dan operasi database.

Kategori	Atribut	Definisi Dasar dan Pertanyaan Untuk Ditanyakan
Intrinsik dan Kontekstual Perwakilan Kategori	Akurasi dan Kepercayaan	Kebenaran dan kredibilitas data: benar, palsu atau akurat?
	Integritas dan Reputasi	Data yang bias atau tidak memihak? Reputasi sumber data?
	Relevansi dan Nilai	Relevansi data dengan tugas yang ada dan nilai tambah atau tidak?
	Volume dan Kelengkapan	Volume data diuji dan nilai apa pun ada?
Intrinsik dan Kontekstual Perwakilan	Mudah Dipahami	Kejelasan data dan mudah dipahami tanpa ambiguitas?
	Interpretabilitas dan Visualisasi	Data terwakili dengan baik dalam angka, tekstual, grafik, gambar, video, profil atau metadata, dll.?
Aksesibilitas dan Keamanan	Kontrol akses	Ketersediaan data, protokol kontrol akses, mudah diambil?

	Tindakan Pencegahan Keamanan	Akses terbatas atau kontrol integritas dari perubahan atau penghapusan?

Akuisisi dan Pra-pemrosesan *Big data*

Pemuatan adalah prosedur yang paling kompleks di antara ketiganya, yang mencakup operasi seperti transformasi, penyalinan, pembersihan, standarisasi, penyaringan, dan pengorganisasian data. Sebuah database virtual dapat dibangun untuk query dan agregat data dari sumber data yang berbeda, tetapi database tersebut tidak berisi data. Sebaliknya, mencakup informasi atau metadata yang berkaitan dengan data aktual dan posisinya. Dua pendekatan "membaca-penyimpanan" seperti itu tidak memenuhi persyaratan kinerja tinggi dari aliran data atau program dan aplikasi pencarian. Dibandingkan dengan query, data dalam dua pendekatan tersebut lebih dinamis dan harus diproses selama transmisi data.

Umumnya, metode integrasi data disertai dengan mesin pemroses aliran dan mesin pencari:

1) Pemilihan Data: Pilih set data target atau subset sampel data tempat penemuan akan dilakukan.
2) Transformasi Data: Sederhanakan kumpulan data dengan menghapus variabel yang tidak diinginkan. Kemudian menganalisis fitur berguna yang dapat digunakan untuk mewakili data, tergantung pada tujuan atau tugas.
3) Data Mining: Mencari pola kepentingan dalam bentuk representasi tertentu atau sekumpulan representasi seperti aturan klasifikasi atau pohon, regresi, pengelompokan, dan sebagainya.
4) Evaluasi dan representasi pengetahuan: Mengevaluasi pola pengetahuan, dan memanfaatkan teknik visualisasi untuk mempresentasikan pengetahuan secara gamblang.

Akuisisi Big data

Ini termasuk pengumpulan data, transmisi data, dan pra-pemrosesan data. Sebagai fase kedua, akuisisi data juga mencakup pengumpulan data, transmisi data, dan pra-pemrosesan data. Selama akuisisi *Big data*, setelah kami mengumpulkan data mentah, kami menggunakan mekanisme transmisi yang efisien untuk mengirimkannya ke sistem manajemen penyimpanan yang tepat untuk mendukung berbagai aplikasi analitik. Kumpulan data yang dikumpulkan terkadang menyertakan banyak data yang berlebihan atau tidak berguna, yang secara tidak perlu menambah ruang penyimpanan dan memengaruhi analisis data selanjutnya. Tabel 1.9 merangkum metode akuisisi data utama dan operasi pra-pemrosesan.

Misalnya, redundansi tinggi sangat umum di antara kumpulan data yang dikumpulkan oleh sensor untuk pemantauan lingkungan. Teknologi kompresi data dapat diterapkan untuk mengurangi redundansi. Oleh karena itu, operasi pra-pemrosesan data sangat diperlukan untuk memastikan penyimpanan dan eksploitasi data yang efisien. Pengumpulan data adalah dengan memanfaatkan teknik pengumpulan data khusus untuk memperoleh data mentah dari

lingkungan pembuatan data tertentu. Banyak sumber pengumpulan dan pembangkitan data umum dan generator data diperkenalkan di bawah ini.

Tabel 1.9 Beberapa sumber akuisisi *Big data* dan operasi preprocessing utama.

Sumber Koleksi	Log, sensor, crawler, pengambilan paket, dan perangkat seluler, dll.
Langkah-Langkah Prapemrosesan	Penghapusan Integrasi, Pembersihan, dan Redundansi
Generator Data	Media Sosial, Perusahaan, Internet of Things, Internet, Bio-Medis, Pemerintah, penemuan ilmiah, lingkungan, dll.

File Log

CA adalah salah satu metode pengumpulan data yang banyak digunakan, dan file log adalah file rekaman yang dihasilkan secara otomatis oleh sistem sumber data, sehingga dapat merekam aktivitas dalam format file yang ditentukan untuk analisis selanjutnya. File log biasanya digunakan di hampir semua perangkat digital. Misalnya, server web mencatat dalam file log jumlah klik, rasio klik, kunjungan, dan catatan properti pengguna web lainnya. Untuk menangkap aktivitas pengguna di situs web, server web terutama menyertakan tiga format file log berikut: format file log publik (NCSA), format log yang diperluas (W3C) dan format log IIS (Microsoft).

Ketiga jenis file log berada dalam format teks ASCII. Basis data selain file teks terkadang dapat digunakan untuk menyimpan informasi log untuk meningkatkan efisiensi kueri penyimpanan log yang besar. Ada juga beberapa file log lain berdasarkan pengumpulan data, termasuk indikator stok dalam aplikasi keuangan dan penentuan status operasi dalam pemantauan jaringan dan manajemen lalu lintas.

Sensor

Sensor biasa digunakan dalam kehidupan sehari-hari untuk mengukur besaran fisis dan mengubah besaran fisis menjadi sinyal digital yang dapat dibaca untuk pemrosesan (dan penyimpanan) selanjutnya. Data sensorik dapat diklasifikasikan sebagai gelombang suara, suara, getaran, mobil, kimia, arus, cuaca, tekanan, suhu, dll. Informasi yang dirasakan ditransfer ke titik pengumpulan data melalui jaringan kabel atau nirkabel, untuk aplikasi yang mungkin mudah dikerahkan dan dikelola, misalnya sistem pengawasan video.

Metode untuk Memperoleh Data Jaringan

Saat ini, akuisisi data jaringan dilakukan dengan menggunakan kombinasi web crawler, sistem segmentasi kata, sistem tugas dan sistem indeks, dll. Web crawler adalah program yang digunakan oleh mesin pencari untuk mengunduh dan menyimpan halaman web. Secara umum, web crawler dimulai dari *Uniform Resource Locator* (URL) dari halaman web awal untuk mengakses halaman web terkait lainnya, di mana ia menyimpan dan mengurutkan semua URL yang diambil. Perayap web memperoleh URL dalam urutan prioritas melalui antrian URL dan

kemudian mengunduh halaman web, dan mengidentifikasi semua URL di halaman web yang diunduh, dan mengekstrak URL baru untuk dimasukkan ke dalam antrian.

Proses ini diulang sampai web crawler dihentikan. Akuisisi data melalui web crawler banyak diterapkan pada aplikasi berbasis halaman web, seperti mesin pencari atau web caching. Teknologi ekstraksi halaman web tradisional menampilkan beberapa solusi efisien dan banyak penelitian telah dilakukan di bidang ini. Saat aplikasi halaman web yang lebih maju muncul, beberapa strategi ekstraksi digunakan untuk mengatasi aplikasi Internet yang kaya. Teknologi akuisisi data jaringan saat ini terutama mencakup teknologi pengambilan paket berbasis Libpcap tradisional, teknologi pengambilan paket tanpa salinan, serta beberapa perangkat lunak pemantauan jaringan khusus seperti Wireshark, SmartSniff dan WinNetCap.

Penyimpanan Big data

Penyimpanan *Big data* mengacu pada penyimpanan dan pengelolaan kumpulan data skala besar sambil mencapai keandalan dan ketersediaan akses data. Pertumbuhan data yang eksplosif memiliki persyaratan yang lebih ketat pada penyimpanan dan pengelolaan data. Kami menganggap penyimpanan *Big data* sebagai komponen ketiga dari ilmu *Big data*. Infrastruktur penyimpanan perlu menyediakan layanan penyimpanan informasi dengan ruang penyimpanan yang andal, dan harus menyediakan antarmuka akses yang kuat untuk kueri dan analisis sejumlah besar data.

Penelitian yang cukup besar tentang *Big data* mendorong pengembangan mekanisme penyimpanan untuk *Big data*. Mekanisme penyimpanan *Big data* yang ada dapat diklasifikasikan ke dalam tiga level bottom-up: sistem file, database, dan model pemrograman. Sistem file adalah dasar dari aplikasi di tingkat atas. GFS Google adalah sistem file terdistribusi yang dapat diperluas untuk mendukung aplikasi skala besar, terdistribusi, dan intensif data. GFS menggunakan server komoditas murah untuk mencapai toleransi kesalahan dan menyediakan layanan berkinerja tinggi kepada pelanggan. GFS mendukung aplikasi file skala besar dengan lebih sering membaca daripada menulis. Namun, GFS juga memiliki beberapa keterbatasan, seperti satu titik kegagalan dan kinerja yang buruk untuk file kecil. Keterbatasan tersebut telah diatasi oleh Colossus, penerus GFS.

Selain itu, perusahaan dan peneliti lain juga memiliki solusi untuk memenuhi permintaan penyimpanan *Big data* yang berbeda. Misalnya, HDFS dan Kosmosfs adalah turunan dari kode sumber terbuka GFS. Microsoft mengembangkan Cosmos untuk mendukung bisnis pencarian dan periklanannya. Facebook memanfaatkan Haystack untuk menyimpan sejumlah besar foto berukuran kecil. Taobao juga mengembangkan TFS dan FastDFS. Kesimpulannya, sistem file terdistribusi telah menjadi relatif matang setelah bertahun-tahun pengembangan dan operasi bisnis. Oleh karena itu, kami akan fokus pada dua level lainnya di sisa bagian ini.

Pembersihan Data

Pembersihan data membersihkan dan memproses data sebelumnya dengan memutuskan strategi untuk menangani bidang yang hilang dan mengubah data sesuai persyaratan.

Pembersihan data adalah proses untuk mengidentifikasi data yang tidak akurat, tidak lengkap, atau tidak masuk akal, dan kemudian memodifikasi atau menghapus data tersebut untuk meningkatkan kualitas data. Umumnya, pembersihan data mencakup lima prosedur pelengkap: mendefinisikan dan menentukan jenis kesalahan, mencari dan mengidentifikasi kesalahan, mengoreksi kesalahan, mendokumentasikan contoh kesalahan dan jenis kesalahan, dan memodifikasi prosedur entri data untuk mengurangi kesalahan di masa mendatang.

Selama pembersihan, format data, kelengkapan, rasionalitas dan pembatasan harus diperiksa. Pembersihan data sangat penting untuk menjaga konsistensi data, yang banyak diterapkan di berbagai bidang, seperti perbankan, asuransi, industri ritel, telekomunikasi, dan kontrol lalu lintas. Dalam e-commerce, sebagian besar data dikumpulkan secara elektronik, yang mungkin memiliki masalah kualitas data yang serius. Masalah kualitas data klasik terutama berasal dari cacat perangkat lunak, kesalahan yang disesuaikan, atau salah konfigurasi sistem. Beberapa mempertimbangkan pembersihan data dalam e-commerce dengan menggunakan crawler dan secara teratur menyalin ulang informasi pelanggan dan akun.

Masalah pembersihan data RFID diperiksa selanjutnya. RFID banyak digunakan di banyak aplikasi, misalnya manajemen inventaris dan pelacakan target. Namun, fitur RFID asli berkualitas rendah, yang mencakup banyak data abnormal yang dibatasi oleh desain fisik dan dipengaruhi oleh kebisingan lingkungan. Model probabilistik dikembangkan untuk mengatasi kehilangan data di lingkungan seluler. Kita bisa membangun sistem untuk secara otomatis memperbaiki kesalahan data input dengan mendefinisikan batasan integritas global.

Integrasi data

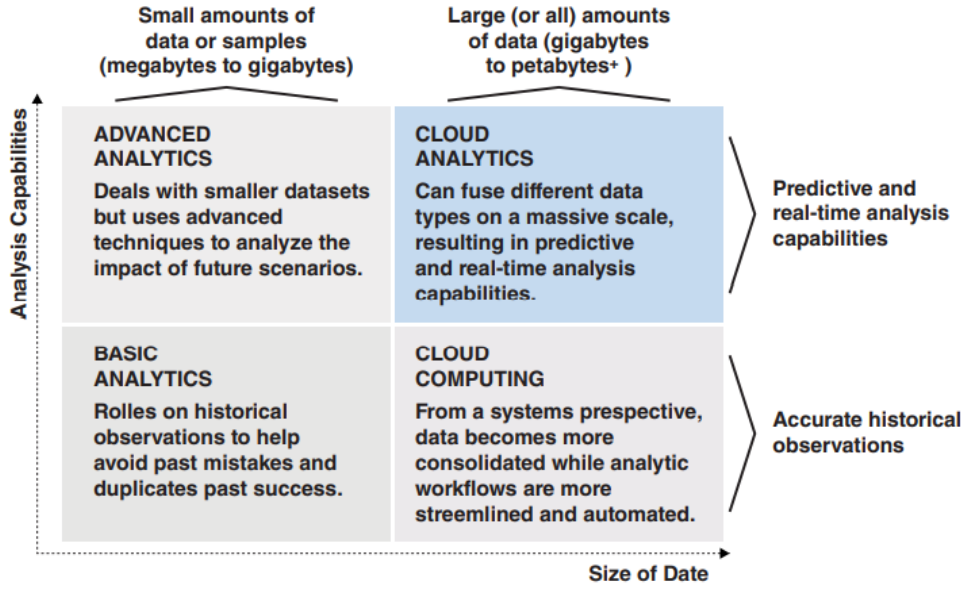
Integrasi data adalah landasan informatika komersial modern, yang melibatkan kombinasi data dari sumber yang berbeda dan memberikan tampilan data yang seragam kepada pengguna. Ini adalah bidang penelitian yang matang untuk database tradisional. Secara historis, dua metode telah dikenal secara luas: gudang data dan federasi data. Data warehousing mencakup proses bernama ETL (Extract, Transform and Load). Ekstraksi melibatkan menghubungkan sistem sumber, memilih, mengumpulkan, menganalisis dan memproses data yang diperlukan. Transformasi adalah eksekusi serangkaian aturan untuk mengubah data yang diekstraksi ke dalam format standar. Memuat berarti mengimpor data yang diekstraksi dan diubah ke dalam infrastruktur penyimpanan target.

Analisis Data yang Berkembang di Atas Cloud

Analisis *Big data* adalah proses pemeriksaan sejumlah besar data dari berbagai jenis (*Big data*) untuk mengungkap pola tersembunyi, korelasi yang tidak diketahui, dan informasi berguna lainnya. Informasi tersebut dapat memberikan keunggulan kompetitif atas organisasi saingan dan menghasilkan kecerdasan bisnis atau penemuan ilmiah yang lebih tinggi, seperti pemasaran yang lebih efektif, peningkatan pendapatan, dll.

Sumber *Big data* harus dilindungi dalam log server web dan data clickstream Internet, laporan aktivitas media sosial, catatan panggilan telepon seluler, dan informasi yang ditangkap

oleh sensor atau perangkat IoT. Analitik *Big data* dapat dilakukan dengan perangkat lunak yang biasa digunakan sebagai bagian dari disiplin analitik tingkat lanjut seperti analitik prediktif dan penambangan data.

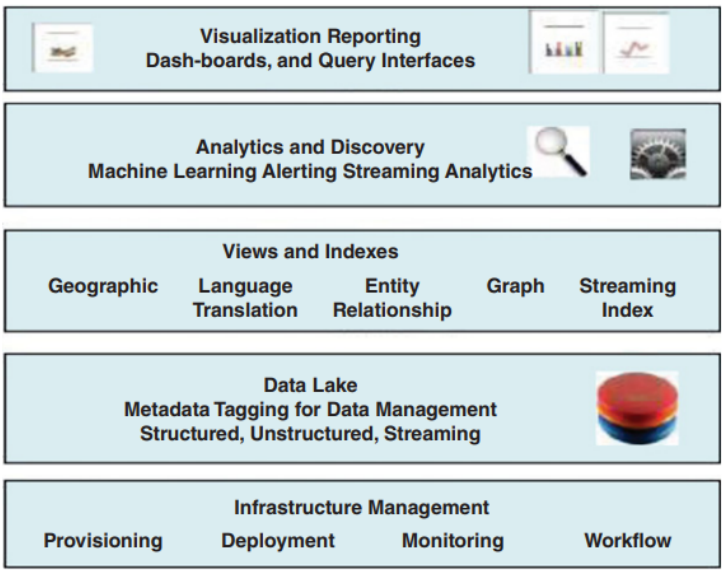


Gambar 1.12 Evolusi dari analisis dasar data kecil (MB ke GB) di masa lalu ke analisis cloud yang canggih pada kumpulan *Big data* (TB~PB) saat ini.

Pada Gambar 1.12, kami menentukan tujuan dan persyaratan analitik cloud saat ini yang berkembang dari analisis dasar yang digunakan dalam menangani data kecil di masa lalu. Di masa lalu, kami menangani objek "data kecil" dalam bentuk MB ke GB, seperti yang ditunjukkan di sisi kiri Gambar 1.12. Pada sumbu x, kami berevolusi dari data kecil menjadi "*Big data*", yang berkisar dari TB hingga PB berdasarkan standar 2015. Pada sumbu y, kami menunjukkan kemampuan analitik dalam dua tingkat menaik: pengamatan historis yang akurat versus kemampuan analisis prediktif dan real-time. Ruang pertunjukan dibagi menjadi empat subruang:

1) Analisis dasar data kecil bergantung pada pengamatan historis untuk membantu menghindari kesalahan masa lalu dan menduplikasi kesuksesan masa lalu.
2) Sistem analitik lanjutan pada data kecil ditingkatkan dari kemampuan dasar hingga menggunakan teknik lanjutan untuk menganalisis dampak skenario masa depan.
3) Saat kami beralih ke computing cloud, sebagian besar cloud yang ada menyediakan alur kerja analitik terkoordinasi yang lebih baik dengan cara yang efisien dan otomatis, tetapi masih kekurangan kemampuan prediktif atau waktu nyata.
4) Untuk sistem analitik cloud yang ideal, kami berharap dapat menangani *Big data* yang dapat diskalakan dalam mode streaming dengan kemampuan prediktif waktu nyata.

Analisis data tradisional berarti penggunaan metode statistik yang tepat untuk menganalisis data tangan pertama dan kedua yang besar, untuk memusatkan, mengekstrak dan memperbaiki data berguna yang tersembunyi dalam kumpulan data yang kacau, dan untuk mengidentifikasi hukum yang melekat pada subjek. masalah, sehingga dapat mengembangkan fungsi data semaksimal mungkin dan memaksimalkan nilai data. Analisis data memainkan peran panduan yang sangat besar dalam rencana pembangunan suatu negara, serta memahami permintaan pelanggan dan memprediksi tren pasar menurut perusahaan bisnis. Analisis *Big data* dapat dianggap sebagai analisis jenis data khusus. Oleh karena itu, banyak metode analisis data tradisional yang masih dapat digunakan untuk analisis *Big data*. Beberapa metode analisis data tradisional yang representatif diperiksa sebagai berikut, banyak di antaranya berasal dari statistik dan ilmu komputer.



Gambar 1.13 Pengembangan berlapis platform cloud untuk pemrosesan *Big data* dan aplikasi analitik.

Secara umum, kami membangun cloud untuk computing *Big data* dengan struktur berlapis, seperti yang diilustrasikan pada Gambar 1.13. Di lapisan bawah, kami memiliki kontrol manajemen infrastruktur cloud, yang menangani penyediaan sumber daya, penyebaran sumber daya yang disepakati, memantau kinerja sistem secara keseluruhan, dan mengatur alur kerja di cloud. Semua elemen *Big data* yang dikumpulkan dari semua sumber membentuk danau data. Data dapat terstruktur atau tidak terstruktur atau datang dan pergi dalam mode streaming. Danau ini tidak hanya menyimpan data mentah tetapi juga metadata untuk pengelolaan data.

Di lapisan tengah, kita perlu memberikan tampilan dan indeks untuk memvisualisasikan dan mengakses data dengan lancar. Ini mungkin termasuk data geografis, mekanisme terjemahan bahasa, hubungan entitas, analisis grafik dan indeks streaming, dll. Pada tingkat berikutnya yang lebih tinggi, kami memiliki mesin pemrosesan cloud yang mencakup

penambangan data, penemuan, dan mekanisme analitik untuk melakukan *Machine learning*, peringatan, dan operasi pemrosesan aliran data. Di tingkat atas, kita harus melaporkan atau menampilkan hasil analitik. Ini termasuk dukungan visualisasi untuk pelaporan dengan dasbor dan antarmuka kueri. Tampilan dapat berupa histogram, grafik batang, bagan, video, dll.

1.4 SMART MACHINE DAN APLIKASI *BIG DATA*

Di bagian ini, kami mencoba menghubungkan Smart Machine dengan aplikasi *Big data*. Smart Machine dikaitkan dengan penginderaan IoT yang diterapkan cloud pintar, dan kemampuan analitik data. Pertama, kami mengungkapkan hubungan antara penambangan data dan *Machine learning*. Kemudian kami memberikan gambaran tentang aplikasi *big data* yang penting. Akhirnya, kami menyajikan konsep kunci computing kognitif dan aplikasinya.

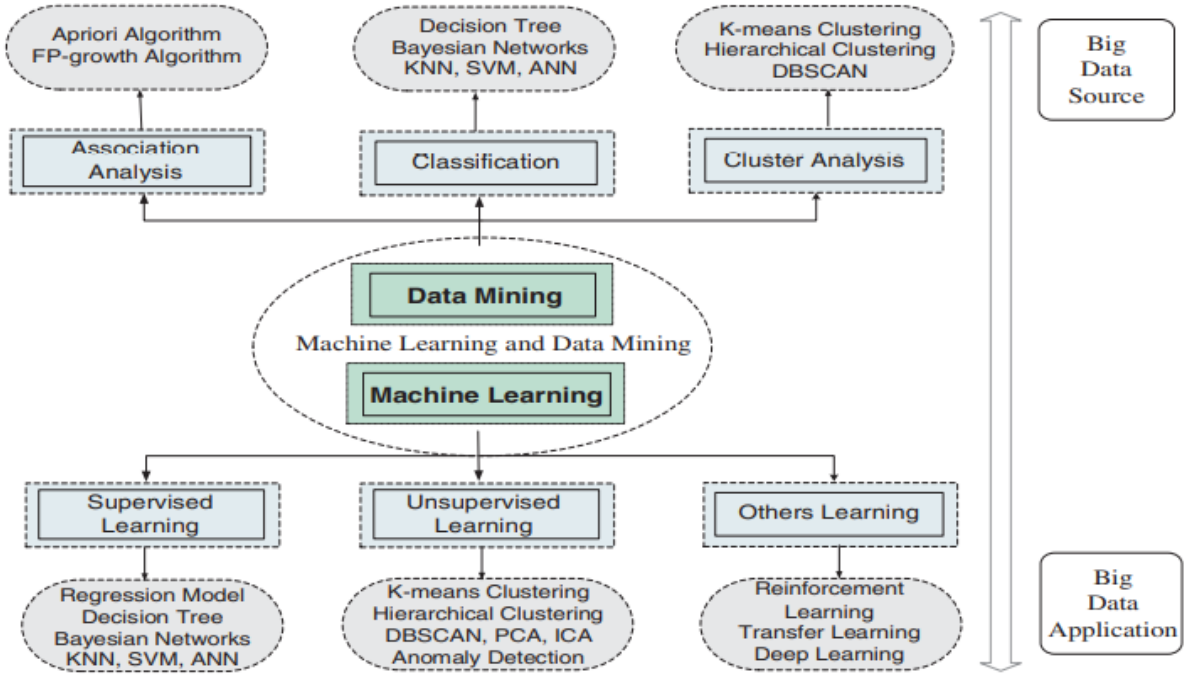
Data Mining dan *Machine learning*

Kami mengklasifikasikan data mining ke dalam tiga kategori: analisis asosiasi, klasifikasi dan analisis cluster. Teknik *Machine learning* dibagi menjadi tiga kategori: pembelajaran terawasi, pembelajaran tanpa pengawasan dan metode pembelajaran lainnya termasuk pembelajaran penguatan, pembelajaran aktif, pembelajaran transfer dan *Deep learning*, dll.

Penambangan Data versus Machine learning

Penambangan data dan *Machine learning* terkait erat satu sama lain. Data mining adalah proses computing untuk menemukan pola dalam kumpulan *Big data* yang melibatkan metode di persimpangan kecerdasan buatan, *Machine learning*, statistik, dan sistem basis data. Tujuan keseluruhan dari proses data mining adalah untuk mengekstrak informasi dari kumpulan data dan mengubahnya menjadi struktur yang dapat dimengerti untuk digunakan lebih lanjut. Selain langkah analisis mentah, ini melibatkan aspek database dan manajemen data, pra-pemrosesan data, pertimbangan model dan inferensi, metrik ketertarikan, pertimbangan kompleksitas, pasca-pemrosesan struktur yang ditemukan, visualisasi dan pembaruan online.

Machine learning mengeksplorasi konstruksi dan studi algoritme yang dapat belajar dari dan membuat prediksi pada data. Algoritme semacam itu beroperasi dengan membangun model dari input contoh untuk membuat prediksi atau keputusan berdasarkan data, daripada mengikuti instruksi program yang benar-benar statis. Kedua istilah ini biasanya membingungkan, karena mereka sering menggunakan metode yang sama dan tumpang tindih secara signifikan. *Machine learning* lebih dekat dengan aplikasi dan pengguna akhir. Ini berfokus pada prediksi, berdasarkan sifat yang diketahui yang dipelajari dari data pelatihan. Seperti yang ditunjukkan pada Gambar 1.14, kami membagi teknik *Machine learning* menjadi tiga kategori: i) pembelajaran terawasi seperti model regresi, pohon keputusan, dll.; ii) pembelajaran tanpa pengawasan, yang meliputi pengelompokan, deteksi anomali, dll.; dan iii) pembelajaran lainnya, seperti pembelajaran penguatan, pembelajaran transfer, pembelajaran aktif dan *Deep learning*, dll.



Gambar 1.14 Hubungan data mining dan *Machine learning*.

Data mining lebih dekat dengan sumber data. Ini berfokus pada penemuan properti yang tidak diketahui dari data, yang juga dianggap sebagai langkah analisis penemuan pengetahuan dalam database. Seperti yang ditunjukkan pada Gambar 1.14, teknik data mining tipikal diklasifikasikan ke dalam tiga kategori: i) analisis asosiasi termasuk algoritma Apriori dan algoritma FP-growing; ii) algoritma klasifikasi mencakup pohon keputusan, mesin vektor pendukung (SVM), k-nearest-neighbor, Na¨ıve Bayesian, jaringan kepercayaan Bayesian dan jaringan syaraf tiruan (JST), dll.; dan iii) algoritma pengelompokan termasuk K-means dan pengelompokan spasial berbasis kepadatan aplikasi dengan noise.

Analisis *Big data* dihadapkan pada banyak tantangan tetapi penelitian saat ini masih dalam tahap awal. Upaya penelitian yang cukup besar diperlukan untuk meningkatkan efisiensi tampilan data, penyimpanan data, dan analisis data. Komunitas riset menuntut definisi *Big data* yang lebih ketat. Kami menuntut model struktural *Big data*, deskripsi formal *Big data*, dan sistem teoritis ilmu data, dll. Sistem evaluasi kualitas data dan standar evaluasi efisiensi computing data harus dikembangkan.

Banyak solusi aplikasi *Big data* mengklaim bahwa mereka dapat meningkatkan pemrosesan data dan kapasitas analisis di semua aspek, tetapi tidak ada standar evaluasi dan tolok ukur terpadu untuk menyeimbangkan efisiensi computing *Big data* dengan metode matematis yang ketat. Kinerja hanya dapat dievaluasi oleh sistem yang diterapkan dan diterapkan, yang tidak dapat secara horizontal membandingkan keuntungan dan kerugian dari berbagai solusi. Sebelum dan sesudah penggunaan *big data*, efisiensinya juga sulit dibandingkan.

Selain itu, karena kualitas data merupakan dasar penting dari pra-pemrosesan, penyederhanaan, dan penyaringan data, maka merupakan masalah mendesak lainnya untuk mengevaluasi kualitas data secara efektif.

Kemunculan *big data* memicu perkembangan desain algoritma yang telah bertransformasi dari pendekatan computing-intensif menjadi pendekatan data-intensif. Transfer data telah menjadi hambatan utama computing *Big data*. Oleh karena itu, banyak model computing baru yang disesuaikan untuk *Big data* telah muncul dan lebih banyak model seperti itu ada di cakrawala. Smart Machine sangat penting untuk memecahkan masalah menantang yang ada dalam aplikasi *Big data*. Smart Machine diperoleh melalui *Machine learning*.

- *Machine learning* yang Diawasi: mencakup kategori berikut:
  a) Model Regresi: Pohon Keputusan, SVM;
  b) Pengklasifikasi Bayesian: Model Markov Tersembunyi;
  c) *Deep learning*: akan dijelaskan pada Bab 8.
- *Machine learning* Tanpa Pengawasan: termasuk:
  a) Pengurangan Dimensi: Analisis komponen utama (PCA);
  b) Clustering: menemukan partisi dari data yang diamati tanpa adanya label eksplisit yang menunjukkan partisi yang diinginkan. Bab 7 akan dikhususkan untuk ini tanpa pengawasan model.
- Teknik *Machine learning* lainnya:
  a) Pembelajaran Penguatan: Proses keputusan Markov (MDPs) menyediakan kerangka kerja matematis untuk pemodelan pengambilan keputusan dalam situasi di mana hasil sebagian acak dan sebagian di bawah kendali pembuat keputusan.
  b) Pembelajaran Transfer: Melalui pembelajaran transfer, biaya pemrosesan yang memakan waktu dan padat karya dapat dikurangi secara ekstensif. Setelah waktu tertentu pelabelan dan validasi melalui pembelajaran transfer, set pelatihan ditetapkan. Di antara berbagai teknologi *big data* utama, Smart Machine adalah komponen kuncinya. Teknik *Machine learning* untuk computing *Big data* akan dipelajari secara rinci di Bab 3, 4 dan 5.

Aplikasi *Big data* – Gambaran Umum

Sejumlah besar aplikasi *Big data* telah dilaporkan dalam literatur. Kami akan membahas aplikasi *big data* dan cloud di Bagian III buku ini. Di sini, di Tabel 1.10, kami hanya memberikan gambaran global tentang berbagai aplikasi *Big data*. Institut Standar dan Teknologi Nasional AS (NIST) telah mengidentifikasi 52 kasus aplikasi aplikasi *big data*. Tugas-tugas ini dikelompokkan ke dalam 9 kategori. Faktanya, banyak aplikasi berbasis data telah muncul dalam dua dekade terakhir. Misalnya, intelijen bisnis telah menjadi teknologi yang berlaku dalam aplikasi bisnis. Mesin pencari jaringan didasarkan pada proses penambangan *Big data*-besaran. Kami secara singkat memperkenalkan aplikasi ini sebagai berikut:

Aplikasi Komersial

Data bisnis paling awal umumnya adalah data terstruktur, yang dikumpulkan oleh perusahaan dari sistem lama dan kemudian disimpan dalam RDBMS. Teknologi analitik yang digunakan dalam sistem seperti itu berlaku pada 1990-an dan intuitif dan sederhana, misalnya laporan, panel instrumen, kueri khusus, intelijen bisnis berbasis pencarian, pemrosesan transaksi online, visualisasi interaktif, kartu skor, pemodelan prediktif, dan data min- ing. Sejak awal abad ke-21, jaringan dan situs web telah memberikan kesempatan unik bagi organisasi untuk memiliki tampilan online dan berinteraksi langsung dengan pelanggan.

Tabel 1.10 Kategori aplikasi *big data*: dari TB hingga PB (NIST 2013).

Kategori	**Deskripsi singkat**	**Contoh Aplikasi**
Pemerintah	Arsip dan Catatan Nasional,	CIA, FBI, Pasukan Polisi, dll.
Bisnis dan Komersial	Administrasi Federal/Negara Bagian, Biro Sensus, dll.	Netflix, Pengiriman kargo, Belanja online, P2P
Pertahanan dan Militer	Keuangan di Cloud, Cloud Backup, Mendeley (Kutipan), Pencarian Web, Materi Digital, dll.	Pentagon, Badan Keamanan Rumah
Perawatan Kesehatan dan Ilmu Kehidupan	Sensor, Pengawasan gambar, Penilaian Situasi, Kontrol Krisis, Manajemen pertempuran, dll.	Sensor Area Tubuh, Genomik, Kontrol Emosi
***Deep learning*, media sosial**	Rekam Medis, Analisis Grafik dan Probabilistik, Patologi, Bioimaging, Genomics, Epidemiologi, dll.	*Machine learning*, Pengenalan Pola, Persepsi, dll.
Penemuan Ilmiah	Mobil self-driving, geolokasi gambar/kamera, Crowd Sourcing, Network Science, set data benchmark NIST	Hadron Collider Besar di CERN dan Belle Accelerator di Jepang
Lingkungan Bumi	Survei Langit, Astronomi dan Fisika, ilmu kutub, Hamburan Radar di Atmosfer, Metadata, Kolaborasi, dll.	Sensor gas AmeriFlux dan FLUXNET, IoT untuk smart earth
Penelitian Energi	Gempa Bumi, Lautan, Pengamatan Bumi, Hamburan Radar Lapisan Es, Dataset simulasi iklim,	Proyek SmartGrid

Identifikasi turbulensi atmosfer, Biogeokimia.

Produk yang melimpah dan informasi pelanggan, termasuk log data aliran klik dan perilaku pengguna, dll., dapat diperoleh dari situs web. Optimalisasi tata letak produk, analisis perdagangan pelanggan, saran produk, dan analisis struktur pasar dapat dilakukan dengan analisis teks dan teknologi penambangan situs web. Jumlah ponsel dan tablet PC pertama kali melampaui laptop dan PC pada tahun 2011. Ponsel dan Internet of Things berbasis sensor membuka aplikasi inovasi generasi baru, dan mencari kapasitas yang lebih besar untuk mendukung penginderaan lokasi, berorientasi pada orang dan operasi konteks.

Aplikasi Jaringan

Internet awal terutama menyediakan layanan email dan halaman web. Analisis teks, penambangan data, dan teknologi analisis halaman web telah diterapkan pada penambangan konten email dan membangun mesin telusur. Saat ini, sebagian besar aplikasi berbasis web, terlepas dari bidang aplikasi dan tujuan desainnya. Data jaringan menyumbang persentase utama dari volume data global. Web telah menjadi platform umum untuk halaman yang saling berhubungan, penuh dengan berbagai jenis data, seperti teks, gambar, video, gambar dan konten interaktif, dll. Teknologi canggih sangat dibutuhkan dalam data semi terstruktur atau tidak terstruktur.

Misalnya, teknologi analisis gambar dapat mengekstrak informasi yang berguna dari gambar, misalnya pengenalan wajah. Teknologi analisis multimedia diterapkan pada sistem pengawasan video otomatis untuk aplikasi bisnis, penegakan hukum, dan militer. Aplikasi media sosial online, seperti forum Internet, komunitas online, blog, layanan jejaring sosial dan situs web multimedia sosial, dll., memberi pengguna peluang besar untuk membuat, mengunggah, dan berbagi konten. Kelompok pengguna yang berbeda dapat mencari berita harian dan mempublikasikan pendapat mereka dengan umpan balik yang tepat waktu.

Big data dalam Aplikasi Ilmiah

Penelitian ilmiah di banyak bidang memperoleh data yang sangat besar dengan sensor dan instrumen berkemampuan tinggi, seperti astrofisika, oseanologi, genomik, dan penelitian lingkungan. Yayasan Sains Nasional AS (NSF) baru-baru ini mengumumkan Inisiatif Penelitian BIGDATA untuk mempromosikan upaya penelitian untuk mengekstrak pengetahuan dan wawasan dari kumpulan data digital yang besar dan kompleks. Beberapa disiplin penelitian ilmiah telah mengembangkan platform data yang sangat besar dan memperoleh hasil yang bermanfaat.

Misalnya, dalam biologi, iPlant menerapkan infrastruktur jaringan, sumber daya computing fisik, lingkungan koordinasi, sumber daya mesin virtual, dan perangkat lunak analisis interoperatif dan layanan data untuk membantu penelitian, pendidik, dan siswa, dalam memperkaya semua ilmu tanaman. Dataset iPlant memiliki variasi bentuk yang tinggi, termasuk data spesifikasi atau referensi, data eksperimen, data analog atau model, data observasi dan data

turunan lainnya. *Big data* telah diterapkan dalam analisis data terstruktur, data teks, data situs web, data multimedia, data jaringan, dan data seluler.

Penerapan Big data di Perusahaan

Saat ini, *Big data* terutama berasal dari dan terutama digunakan di perusahaan bisnis, sementara BI dan OLAP dapat dianggap sebagai pendahulu aplikasi *Big data*. Penerapan *big data* di perusahaan bisnis dapat meningkatkan efisiensi produksi dan daya saing mereka dalam banyak aspek. Secara khusus, dalam pemasaran, dengan analisis korelasi *Big data*, perusahaan bisnis dapat secara akurat memprediksi perilaku konsumen.

Pada perencanaan penjualan, setelah membandingkan data yang sangat besar, perusahaan dapat mengoptimalkan harga komoditas mereka. Pada operasi, perusahaan tersebut dapat meningkatkan efisiensi operasi dan kepuasan operasi, mengoptimalkan input tenaga kerja, memperkirakan kebutuhan alokasi personel secara akurat, menghindari kelebihan kapasitas produksi dan mengurangi biaya tenaga kerja. Pada rantai pasokan, dengan menggunakan *Big data*, perusahaan bisnis dapat melakukan pengoptimalan inventaris, pengoptimalan logistik dan koordinasi pemasok, dll., untuk mengurangi kesenjangan antara pasokan dan permintaan, mengontrol anggaran, dan meningkatkan layanan.

Contoh 1.3 Penggunaan *Big data* di Perbankan dalam Aplikasi Pembiayaan dan e-Commerce

Dalam komunitas keuangan, aplikasi *big data* telah berkembang pesat dalam beberapa tahun terakhir. Misalnya, China Merchants Bank menggunakan analisis data untuk mengetahui bahwa aktivitas seperti “akumulasi skor berkali-kali” dan “pertukaran skor di toko”, efektif untuk menarik pelanggan berkualitas. Dengan membangun model peringatan dini kehilangan pelanggan, bank dapat menjual produk keuangan hasil tinggi kepada 20% pelanggan teratas dalam rasio kerugian untuk mempertahankannya. Akibatnya, rasio kerugian pelanggan dengan Kartu Emas dan Kartu Bunga Matahari telah berkurang masing-masing sebesar 15% dan 7%.

Dengan menganalisis catatan transaksi pelanggan, pelanggan perusahaan kecil dan mikro yang potensial dapat diidentifikasi secara efektif. Memanfaatkan perbankan jarak jauh, platform rujukan cloud dapat membantu menerapkan penjualan silang, dan peningkatan kinerja yang cukup besar telah diamati dalam beberapa tahun terakhir. Jelas, aplikasi paling klasik ada di e-commerce. Puluhan ribu transaksi dilakukan di Taobao dan waktu transaksi yang sesuai, harga komoditas, dan jumlah pembelian dicatat setiap hari.

Lebih penting lagi, informasi tersebut sesuai dengan usia, jenis kelamin, alamat dan bahkan hobi dan minat pembeli dan penjual. Data Cube of Taobao adalah aplikasi *Big data* pada platform Taobao, di mana pedagang dapat mengetahui status industri makroskopik platform Taobao, kondisi pasar merek dan perilaku konsumen mereka, dll., dan karenanya membuat produksi dan keputusan persediaan. Sementara itu, semakin banyak konsumen yang dapat membeli komoditas favorit mereka dengan harga yang lebih disukai.

Pinjaman kredit Alibaba secara otomatis menganalisis dan menilai apakah akan memberikan pinjaman kepada perusahaan bisnis melalui data transaksi perusahaan yang

diperoleh berdasarkan teknologi *Big data*, sementara intervensi manual tidak terjadi di seluruh proses. Diungkapkan bahwa, sejauh ini, Alibaba telah meminjamkan lebih dari RMB 30 miliar Yuan, dengan tingkat kredit macet hanya sekitar 0,3\%, yang jauh lebih rendah daripada bank komersial lainnya.

Aplikasi Perawatan Kesehatan dan Medis

Industri kesehatan berkembang pesat dan data medis merupakan data kompleks yang terus berkembang pesat, mengandung nilai informasi yang melimpah dan beragam. *Big data* memiliki potensi tak terbatas untuk menyimpan, memproses, menanyakan, dan menganalisis data medis secara efektif. Penerapan *big data* medis akan sangat mempengaruhi kesehatan manusia. IoT merevolusi industri perawatan kesehatan. Sensor mengumpulkan data pasien, kemudian mikrokontroler memproses, menganalisis, dan mengomunikasikan data melalui Internet nirkabel. Mikroprosesor memungkinkan antarmuka pengguna grafis yang kaya. Cloud dan gerbang layanan kesehatan membantu menganalisis data dengan akurasi statistik. Beberapa contoh sederhana diberikan di bawah ini. Lebih lanjut tentang IoT perawatan kesehatan dan aplikasi *Big data* akan dipelajari di Bab 4, 5, 8 dan 9.

Contoh 1.4 Aplikasi *Big data* di Industri Kesehatan

Perusahaan Asuransi Jiwa Aetna memilih 102 pasien dari 1000 pasien untuk menyelesaikan eksperimen guna membantu memprediksi pemulihan pasien dengan sindrom metabolik. Dalam sebuah eksperimen independen, ia memindai 600.000 hasil uji laboratorium dan 180.000 klaim melalui serangkaian hasil uji deteksi sindrom metabolik pasien dalam tiga tahun berturut-turut. Selain itu, hasil akhir diringkas menjadi rencana perawatan pribadi yang ekstrem untuk menilai faktor-faktor berbahaya dan rencana perawatan utama pasien.

Dengan cara ini, dokter dapat mengurangi morbiditas sebesar 50% dalam 10 tahun ke depan, dengan meresepkan statin dan membantu pasien untuk menurunkan berat badan sebanyak 5 lb, atau menyarankan pasien untuk mengurangi total trigliserida dalam tubuh mereka jika kandungan gula di dalam tubuh mereka. tubuh lebih dari 20%. Mount Sinai Medical Center di AS menggunakan teknologi Ayasdi, sebuah perusahaan *Big data*, untuk menganalisis semua urutan genetik Escherichia Coli, termasuk lebih dari 1 juta varian DNA, untuk mengetahui mengapa strain bakteri menolak antibiotik. Teknologi Ayasdi menggunakan analisis data Topologi, metode penelitian matematika baru, untuk memahami karakteristik data.

HealthVault dari Microsoft menawarkan aplikasi luar biasa dari *Big data* medis yang diluncurkan pada tahun 2007. Tujuannya adalah untuk mengelola informasi kesehatan individu dalam peralatan medis individu dan keluarga. Saat ini, informasi kesehatan dapat dimasukkan dan diunggah dengan perangkat pintar seluler dan diimpor ke rekam medis individu oleh agen pihak ketiga. Selain itu, dapat diintegrasikan dengan aplikasi pihak ketiga dengan perangkat pengembangan perangkat lunak (SDK) dan antarmuka terbuka.

Kecerdasan kolektif

Dengan perkembangan pesat teknologi komunikasi dan sensor nirkabel, ponsel dan komputer tablet telah mengintegrasikan lebih banyak sensor, dengan kapasitas computing dan penginderaan yang semakin kuat. Akibatnya, penginderaan kerumunan menjadi pusat perhatian computing seluler. Dalam penginderaan kerumunan, sejumlah besar pengguna umum menggunakan perangkat seluler sebagai unit penginderaan dasar untuk melakukan koordinasi dengan jaringan seluler untuk distribusi tugas penginderaan dan pengumpulan dan pemanfaatan data penginderaan. Tujuannya adalah untuk menyelesaikan tugas penginderaan sosial skala besar dan kompleks. Dalam penginderaan massa, peserta yang menyelesaikan tugas penginderaan kompleks tidak perlu memiliki keterampilan profesional.

Mode penginderaan kerumunan yang diwakili oleh Crowdsourcing telah berhasil diterapkan pada foto yang diberi tag geo, penentuan posisi dan navigasi, penginderaan lalu lintas jalan perkotaan, peramalan pasar, penggalian opini, dan aplikasi padat karya lainnya. Crowdsourcing, pendekatan baru untuk pemecahan masalah, mengambil sejumlah besar pengguna umum sebagai dasar dan mendistribusikan tugas secara bebas dan sukarela. Crowdsourcing dapat berguna untuk aplikasi padat karya, seperti penandaan gambar, terjemahan bahasa, dan pengenalan suara.

Ide utama dari Crowdsourcing adalah untuk mendistribusikan tugas kepada pengguna umum dan untuk menyelesaikan tugas yang tidak dapat diselesaikan oleh pengguna secara individu atau tidak diantisipasi untuk diselesaikan. Tanpa perlu secara sengaja menyebarkan modul penginderaan dan mempekerjakan profesional, Crowdsourcing dapat memperluas cakupan penginderaan sistem penginderaan untuk mencapai skala kota dan bahkan skala yang lebih besar. Crowdsourcing diterapkan oleh banyak perusahaan sebelum munculnya *big data*. Misalnya, P&G, BMW dan Audi meningkatkan kapasitas R&D dan desain mereka berdasarkan Crowdsourcing.

Di era *big data*, Spatial Crowdsourcing menjadi topik hangat. Kerangka operasi Spatial Crowdsourcing ditunjukkan sebagai berikut. Seorang pengguna dapat meminta layanan dan sumber daya yang terkait dengan lokasi tertentu. Kemudian pengguna ponsel yang bersedia berpartisipasi dalam tugas akan pindah ke lokasi yang ditentukan untuk memperoleh data terkait (yaitu video, audio atau gambar). Akhirnya, data yang diperoleh akan dikirim ke peminta layanan. Dengan pesatnya pertumbuhan perangkat seluler dan semakin kompleksnya fungsi yang disediakan oleh perangkat tersebut, diperkirakan bahwa Spatial Crowdsourcing akan lebih lazim daripada Crowdsourcing tradisional, misalnya Amazon Turk dan Crowdflower.

Computing Kognitif – Sebuah Pengantar

Istilah computing kognitif berasal dari ilmu kognitif dan kecerdasan buatan. Selama bertahun-tahun, kami ingin membangun "komputer" yang dapat menghitung serta belajar dengan pelatihan, untuk mencapai beberapa indera atau kecerdasan seperti manusia. Ini telah disebut "komputer yang terinspirasi otak" atau "komputer saraf". Komputer seperti itu akan

dibuat dengan perangkat keras dan/atau perangkat lunak khusus, yang dapat meniru fungsi dasar otak manusia seperti menangani informasi kabur dan melakukan respons afektif, dinamis, dan instan. Ini dapat menangani beberapa ambiguitas dan ketidakpastian di luar komputer tradisional.

Untuk tujuan ini, kami menginginkan mesin kognitif yang dapat memodelkan otak manusia dengan kekuatan kognitif untuk belajar, menghafal, menalar, dan merespons stimulus eksternal, secara mandiri dan tanpa lelah. Bidang ini juga disebut "neuroinformatika". Perangkat keras dan aplikasi computing kognitif bisa lebih efektif dan berpengaruh dengan pilihan desain untuk membuat kelas masalah baru dapat dihitung. Sistem seperti itu menawarkan sintesis, bukan hanya sumber informasi tetapi juga pengaruh, konteks, dan wawasan. IBM menjelaskan sistem yang belajar pada skala, alasan dengan tujuan dan berinteraksi dengan manusia. Perangkat keras dan aplikasi computing kognitif bisa lebih efektif dan berpengaruh dengan pilihan desain untuk membuat kelas masalah baru dapat dihitung. Sistem seperti itu menawarkan sintesis bukan hanya sumber informasi tetapi juga pengaruh, konteks, dan wawasan. Dengan kata lain, sistem computing kognitif membuat beberapa "konteks" yang terdefinisi dengan baik dapat dihitung. IBM menjelaskan sistem yang belajar pada skala, alasan dengan tujuan dan berinteraksi dengan manusia secara alami.

Fitur Sistem Computing Kognitif

Di satu sisi, sistem kognitif mendefinisikan kembali hubungan antara manusia dan lingkungan digital mereka yang meresap. Mereka mungkin memainkan peran sebagai asisten atau pelatih bagi pengguna, dan mereka dapat bertindak hampir secara mandiri dalam banyak situasi. Hasil computing dari sistem kognitif bisa sugestif, preskriptif atau instruktif di alam. Tercantum di bawah ini adalah beberapa karakteristik sistem computing kognitif:

Adaptif dalam pembelajaran: Mereka mungkin belajar saat informasi berubah, dan seiring dengan berkembangnya tujuan dan persyaratan. Mereka mungkin menyelesaikan ambiguitas dan mentolerir ketidakpastian. Mereka mungkin direkayasa untuk memberi makan pada data dinamis secara real time, atau mendekati real time.

Interaktif dengan pengguna: Pengguna dapat mendefinisikan kebutuhan mereka sebagai pelatih sistem kognitif. Mereka juga dapat berinteraksi dengan prosesor, perangkat, dan layanan cloud lainnya, serta dengan orang-orang.

Iterative and stateful: Mereka dapat mendefinisikan kembali suatu masalah dengan mengajukan pertanyaan atau menemukan masukan sumber tambahan jika pernyataan masalah tidak jelas atau tidak lengkap. Mereka mungkin "mengingat" interaksi sebelumnya secara berulang.

Kontekstual dalam penemuan informasi: Mereka dapat memahami, mengidentifikasi, dan mengekstrak elemen kontekstual seperti makna, sintaksis, waktu, lokasi, domain yang sesuai, peraturan, profil pengguna, proses, tugas, dan tujuan. Mereka dapat memanfaatkan berbagai

sumber informasi, termasuk informasi digital terstruktur dan tidak terstruktur, serta input sensorik seperti visual, gestural, pendengaran atau sensor yang disediakan.

Perbedaan dengan Komputer Saat Ini

Sistem kognitif berbeda dari aplikasi computing saat ini dalam hal mereka bergerak melampaui tabulasi dan penghitungan berdasarkan aturan dan program yang telah dikonfigurasi sebelumnya. Meskipun mereka mampu melakukan computing dasar, mereka juga dapat menyimpulkan dan bahkan bernalar berdasarkan tujuan yang luas. Sistem computing kognitif dapat diperluas untuk mengintegrasikan atau memanfaatkan sistem informasi yang ada dan menambahkan antarmuka dan alat khusus domain atau tugas. Sistem kognitif memanfaatkan sumber daya TI saat ini dan hidup berdampingan dengan sistem warisan ke masa depan. Tujuan utamanya adalah untuk membawa computing lebih dekat ke pemikiran manusia dan menjadi kemitraan mendasar dalam upaya manusia.

Tabel 1.11 Bidang terkait dengan neuroinformatika dan computing kognitif.

Area subjek	Deskripsi Singkat Bidang	Dukungan Teknologi
Kecerdasan buatan	Studi fenomena kognitif untuk menerapkan kecerdasan manusia di komputer	Pengenalan pola, robotika, visi komputer, pemrosesan ucapan, dll.
Belajar dan Memori	Mempelajari mekanisme pembelajaran dan memori manusia dan membangunnya di komputer masa depan	*Machine learning*, sistem database, peningkatan memori, dll.
Bahasa dan Linguistik	Studi tentang bagaimana linguistik dan bahasa dipelajari dan diperoleh, dan bagaimana memahami kalimat baru	Pemrosesan bahasa dan ucapan, terjemahan mesin, dll.
Persepsi dan Tindakan	Studi tentang kemampuan untuk menerima informasi melalui indera seperti penglihatan dan pendengaran, dll. Rangsangan haptic, olfactory dan gustatory termasuk dalam domain ini	Pengenalan dan pemahaman gambar, ilmu perilaku, pencitraan otak, psikologi dan antropologi

Neuro-informatika	Neuroinformatika berdiri di persimpangan ilmu saraf dan ilmu informasi	Neurokomputer, jaring saraf tiruan, *Deep learning*, penuaan, pengendalian penyakit, dll.
Rekayasa Pengetahuan	Studi tentang analisis *Big data*, penemuan pengetahuan, transformasi dan proses kreativitas	Pendataan, analitik data, penemuan pengetahuan, dan konstruksi sistem

Ilmu kognitif bersifat interdisipliner. Ini mencakup bidang psikologi kecerdasan buatan, ilmu saraf dan linguistik, dll. Ini mencakup banyak tingkat analisis dari *Machine learning* tingkat rendah dan mekanisme keputusan ke sirkuit saraf tingkat tinggi untuk membangun komputer model otak. Bidang terkait dengan neuroinformatika dan computing kognitif dirangkum dalam Tabel 1.11. Dalam Bab 3 dan bab-bab berikutnya, kita akan mengeksplorasi lebih jauh teknologi mutakhir ini.

Aplikasi Machine learning Kognitif

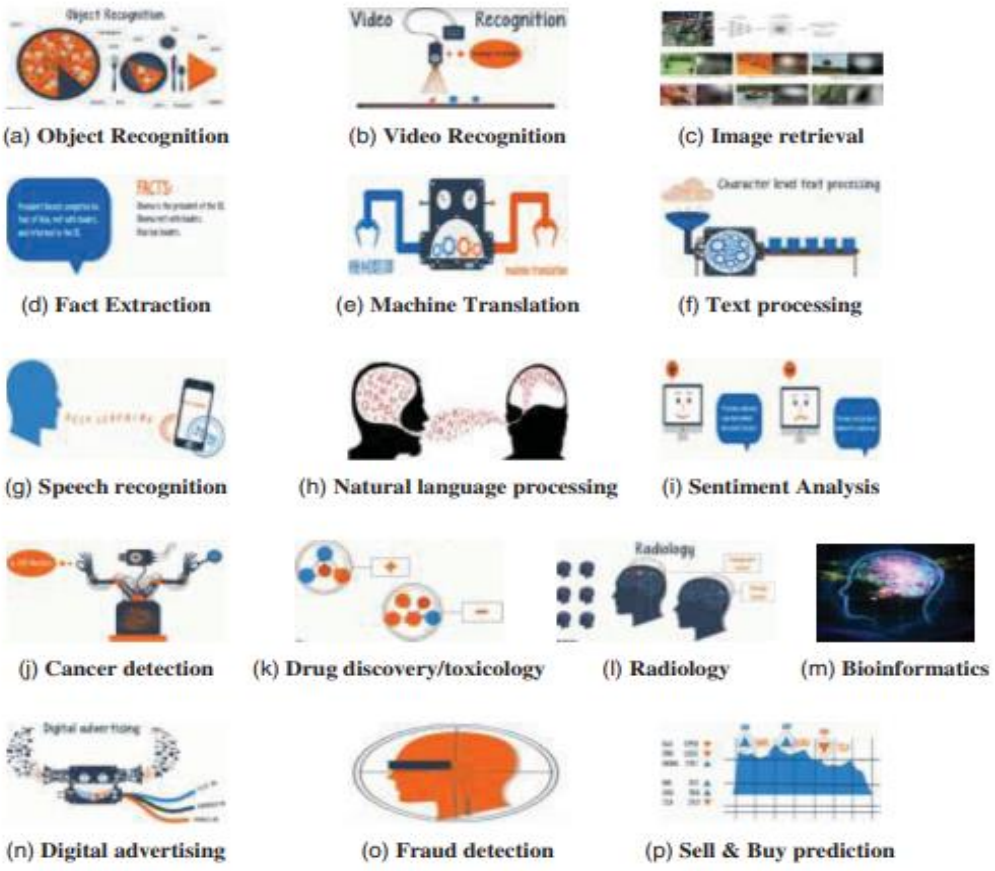
Platform computing kognitif telah muncul dan tersedia secara komersial, dan bukti aplikasi dunia nyata mulai muncul ke permukaan. Organisasi telah mengadopsi dan menggunakan platform computing kognitif ini untuk tujuan mengembangkan aplikasi untuk menangani kasus penggunaan tertentu, dengan setiap aplikasi memanfaatkan beberapa kombinasi fungsionalitas yang tersedia. Contoh kasus dunia nyata seperti itu meliputi: i) pemahaman ucapan; ii) analisis sentimen; iii) pengenalan wajah; iv) wawasan pemilu; v) mengemudi otonom; dan vi) aplikasi *Deep learning*. Banyak lagi contoh yang tersedia dalam layanan computing kognitif. Ini mengungkap kemungkinan ke dalam aplikasi dunia nyata. Gambar 1.15 mencantumkan semua aplikasi *Machine learning* kognitif yang penting.

Di antara aplikasi *Big data* ini:

a) pengenalan objek;
b) interpretasi video;
c) pengambilan gambar;

Terkait dengan aplikasi visi mesin. Tugas teks dan dokumen meliputi:

a) ekstraksi fakta;
b) terjemahan mesin; dan
c) pemahaman teks.



Gambar 1.15 Aplikasi mesin dan *Deep learning* yang diklasifikasikan dalam 16 kategori.

Di sisi deteksi audio dan emosi, kami memiliki:

a) pengenalan suara;
b) pemrosesan bahasa alami, dan
c) tugas analisis sentimen.

Dalam aplikasi medis atau perawatan kesehatan, kami memiliki:

a) deteksi kanker;
b) penemuan obat;
c) toksikologi dan radiologi; dan
d) bioinformatika.

Informasi tambahan tentang aplikasi *Machine learning* kognitif dapat ditemukan di situs web youtube: www.youtube.com/playlist?list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu

Dalam aplikasi bisnis dan keuangan, kami memiliki (n) periklanan digital, (o) deteksi penipuan, dan (p) prediksi jual dan beli dalam analisis pasar. Banyak dari tugas kognitif ini sedang menunggu otomatisasi. Beberapa aplikasi yang diidentifikasi melibatkan banyak data mentah dalam teks triliunan kata dalam berbagai bahasa, data visual dalam miliaran gambar dan video,

audio dalam 400 hari pidato, permintaan pengguna dan pesan pemasaran, ditambah grafik pengetahuan dan media sosial dalam miliaran. dari tupel berlabel.

1.5 KESIMPULAN

Bab ini memperkenalkan definisi dasar dan konsep kunci dari ilmu *Big data* dan computing kognitif. Tujuannya adalah untuk mempersiapkan pembaca kami untuk mempelajari perawatan mendalam di bab-bab berikutnya. Cloud pintar didukung oleh penginderaan IoT dan analitik *Big data*. Cakupan lebih dari cloud pintar diberikan dalam Bab 3, 4 dan 9. Kami menekankan interaksi atau perpaduan teknologi SMACT untuk pemrosesan *Big data*. Jaringan media sosial dan akses seluler dari layanan cloud diperkenalkan di Bagian 1.2. Topik-topik ini akan dipelajari lebih lanjut di Bab 2, 3, 8 dan 9. Kami memperkenalkan dasar-dasar penambangan data, *Machine learning*, analisis data, dan computing kognitif di Bagian 1.3 dan 1.4. Topik *Big data* ini dipelajari lebih lanjut dalam bab-bab berikutnya.

Tugas dan Latihan

1. Secara singkat cirikan perbedaan dalam paradigma dan teknologi computing berikut: Cloud, pusat data, virtualisasi, superkomputer, teknologi Internet, layanan web, computing utilitas, dan computing layanan:
   a) Computing cloud versus aplikasi superkomputer?
   b) Apa persamaan dan perbedaan antara cloud dan pusat data?
   c) Internet konvensional versus Internet benda?
   d) Apa itu computing utilitas versus computing layanan?
   e) Mengapa virtualisasi penting untuk penggunaan cloud saat ini?
2. Siklus hype diperbarui setiap tahun. Anda telah belajar dari Gambar 1.4 tentang kemajuan hingga Juli 2016. Periksa dengan Wikipedia dengan Laporan Gartner terbaru tentang Siklus Hype dan diskusikan perubahan baru dibandingkan dengan laporan 2016.
3. Pekerjaan rumah ini mengharuskan Anda untuk melakukan penelitian. Tulis laporan penilaian terbaru dari teknologi SMACT. Diskusikan kekuatan dan kelemahan serta pro/kontra masing-masing teknologi. Anda perlu menggali beberapa laporan teknis yang relevan atau kertas putih dari industri yang relevan, terutama dari pemain industri besar seperti Facebook, AT&T, Google, Amazon dan IBM, dll. Membaca beberapa makalah yang diterbitkan di Majalah atau Konferensi ACM/IEEE terkemuka akan menjadi berguna untuk membuat penilaian mendalam dengan bukti nyata.
4. Bandingkan computing desktop on-premise konvensional dengan tiga layanan cloud yang dibagi dalam lima kategori: aplikasi pengguna, mesin virtual, server, penyimpanan, dan antara pengguna dan vendor. Tunjukkan label kontrol yang sesuai sebagai Pengguna, Vendor, dan Dibagikan dalam empat model computing ini. Benarkan label Anda dengan alasan.

5. Setelah mempelajari konsep dasar cloud seluler di Bagian 1.2.4, coba jawab layanan cloud. Memeriksa Wikipedia di bawah cloud seluler atau computing cloud seluler dapat membantu dalam menemukan jawabannya. Informasi tambahan juga dapat ditemukan di IEEE MobileCloud Conferences atau edisi Khusus tentang *Mobile cloud computing* di IEEE Transactions on Could Computing atau Service Computing.
6. Jelaskan secara singkat masalah (tantangan) yang terkait dengan empat "V" karakteristik *Big data*. (1)Volume, (2)Velocity, (3)Variety, dan (4)Veracity. Diskusikan permintaan sumber daya dan persyaratan pemrosesan yang terkait dengan latar belakang matematika atau statistik.
7. Dalam ilmu data, apa yang merupakan persimpangan bidang keahlian domain aplikasi bidang keterampilan pemrograman dan latar belakang matematika atau statistik yang diperlukan.
8. Pertimbangkan dua aplikasi layanan cloud/IoT berikut, dan temukan layanan cloud lainnya di Contoh 1.4. Anda perlu melaporkan temuan Anda tentang bagaimana *Machine learning* dan analitik *Big data* dapat membantu dalam kisah sukses mereka.
9. Jelaskan mengapa *Big data* dapat ditangani dengan lebih hemat biaya oleh cloud daripada dengan menggunakan superkomputer. Mengapa ilmuwan *Big data* membutuhkan keahlian domain. Jelaskan juga perbedaan dalam teknik *Machine learning* yang diawasi dan tidak diawasi.
10. Pada Gambar 1.3, kami telah mengidentifikasi sejumlah perangkat lunak yang disediakan oleh berbagai perusahaan atau pusat penelitian. Pertimbangkan tiga paket perangkat lunak berikut: Pustaka MatLab untuk algoritma computing, gudang *Machine learning* UCI untuk analisis data, dan OpenNLP untuk pemrosesan bahasa alami. Cari tahu dari situs web mereka atau literatur tentang fungsionalitas dan persyaratan penggunaan mereka untuk computing *Big data*.

BAB 2
LAYANAN *SMART CLOUD*, VIRTUALISASI, DAN *MASHUP*

2.1 MODEL DAN LAYANAN *CLOUD COMPUTING*

Konsep computing cloud telah berkembang dari cluster, grid dan computing utilitas. Computing cluster dan grid memanfaatkan penggunaan banyak komputer secara paralel. Computing cloud memanfaatkan sumber daya elastis untuk memuaskan sejumlah besar pengguna. Kekuatan pendorong utama di balik computing cloud adalah keberadaan broadband dan jaringan nirkabel di mana-mana, penurunan biaya penyimpanan, dan peningkatan progresif dalam perangkat lunak computing Internet.

Pengguna cloud dapat menuntut lebih banyak sumber daya pada beban kerja puncak, mengurangi biaya mereka, bereksperimen dengan layanan baru, dan menghapus kapasitas yang tidak dibutuhkan. Penyedia layanan cloud dapat meningkatkan pemanfaatan sistem melalui multiplexing, virtualisasi dan penyediaan sumber daya dinamis. Cloud membebaskan pengguna untuk fokus pada aplikasi pengguna dan menciptakan nilai bisnis dengan mengalihdayakan pelaksanaan pekerjaan ke penyedia cloud.

Taksonomi Cloud berdasarkan Layanan yang Disediakan

Cloud diaktifkan oleh kemajuan dalam teknologi perangkat keras, perangkat lunak, dan jaringan yang dirangkum dalam Tabel 2.1. Teknologi ini memainkan peran penting dalam mewujudkan computing cloud. Sebagian besar teknologi ini telah matang untuk memenuhi permintaan yang meningkat. Di bidang perangkat keras, kemajuan pesat dalam CPU multi-core, chip memori, dan susunan disk telah memungkinkan untuk membangun pusat data yang lebih cepat dengan ruang penyimpanan yang besar. Virtualisasi sumber daya memungkinkan penyebaran cloud yang cepat dengan HTC dan kemampuan pemulihan bencana.

Kemajuan dalam menyediakan Software as a Service (SaaS), standar Wed 2.0 dan kinerja Internet semuanya berkontribusi pada munculnya layanan cloud. Cloud saat ini dirancang untuk melayani sejumlah besar penyewa dengan volume data yang sangat besar. Ketersediaan sistem penyimpanan terdistribusi berskala besar meletakkan dasar bagi pusat data saat ini.

Computing cloud telah dihasilkan dari kemajuan yang dibuat dalam manajemen lisensi dan teknik penagihan otomatis. Cloud pribadi lebih mudah diamankan dan lebih dapat dipercaya dalam suatu organisasi. Setelah cloud pribadi menjadi matang dan lebih aman, mereka dapat dibuka atau diubah menjadi cloud publik. Oleh karena itu, batas antara cloud publik dan pribadi dapat menjadi kabur di masa depan. Kemungkinan besar, sebagian besar cloud masa depan akan bersifat hibrida.

Tabel 2.1 Teknologi cloud dalam perangkat keras, perangkat lunak, dan jaringan.

Teknologi	Persyaratan dan Manfaat
Penerapan Platform Cepat	Penyebaran sumber daya cloud yang cepat, efisien, dan fleksibel untuk menyediakan lingkungan computing yang dinamis bagi pengguna
Cluster Virtual Sesuai Permintaan	Cluster VM tervirtualisasi yang disediakan untuk memenuhi permintaan pengguna saat beban kerja berubah
Teknik Multi-Penyewa	SaaS mendistribusikan perangkat lunak ke sejumlah besar pengguna untuk penggunaan simultan dan berbagi sumber daya jika diinginkan
Pemrosesan *Big data*-besaran	Pencarian internet dan layanan web seringkali membutuhkan pemrosesan data yang besar, terutama untuk mendukung layanan yang dipersonalisasi
Komunikasi Skala Web	Mendukung e-commerce, pendidikan jarak jauh, telemedicine, jejaring sosial, pemerintahan digital dan hiburan digital, dll.
Penyimpanan Terdistribusi	Penyimpanan catatan pribadi dan informasi arsip publik berskala besar menuntut penyimpanan terdistribusi di atas cloud
Layanan Perizinan dan Penagihan	Manajemen lisensi dan layanan penagihan sangat menguntungkan semua jenis layanan cloud dalam computing utilitas

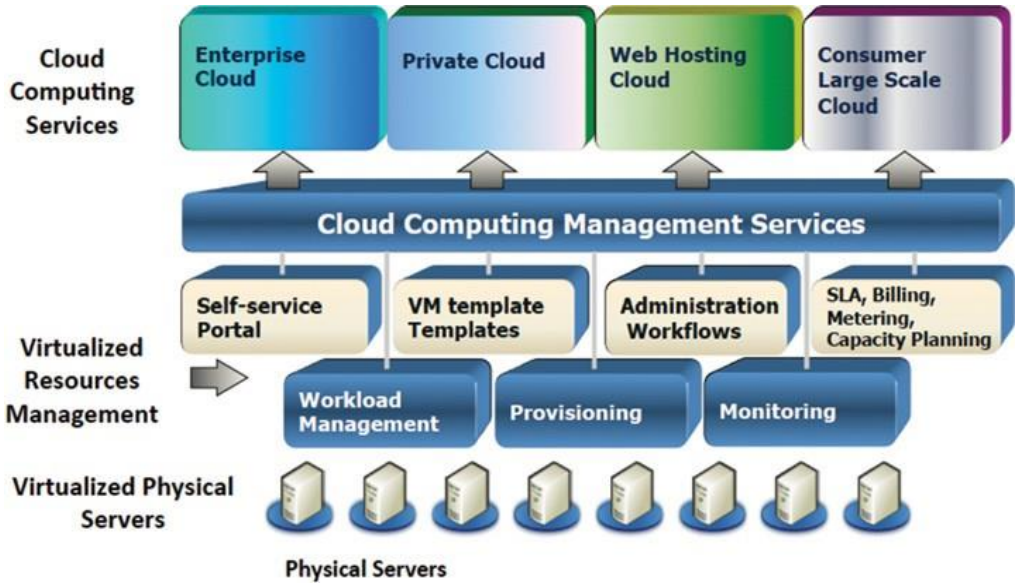
Tren Pengembangan Cloud

Banyak kode aplikasi yang dapat dieksekusi jauh lebih kecil daripada kumpulan data skala web yang mereka proses, dan computing cloud menghindari pergerakan data yang besar selama eksekusi. Ini akan menghasilkan lebih sedikit lalu lintas di Internet dan pemanfaatan jaringan yang lebih baik. Inti dari cloud adalah cluster server (atau cluster VM). Node cluster digunakan sebagai node computing. Beberapa node kontrol digunakan untuk mengelola dan memantau aktivitas cloud. Penjadwalan pekerjaan pengguna di cloud diperlukan untuk menetapkan pekerjaan ke cluster virtual yang dibuat untuk pengguna. Node gateway menyediakan titik akses layanan dari dunia luar. Node gateway ini juga dapat digunakan untuk kontrol keamanan seluruh platform cloud. Dalam cluster fisik, pengguna mengharapkan permintaan sumber daya yang statis. Cloud dirancang untuk menghadapi beban kerja yang berfluktuasi dan dengan demikian menuntut jumlah sumber daya yang bervariasi secara dinamis. Pusat data dan superkomputer memang memiliki beberapa persamaan dan perbedaan. Dalam kasus pusat data, penskalaan merupakan persyaratan mendasar. Cluster server pusat data dibangun dengan server berbiaya rendah. Misalnya, Microsoft memiliki pusat data di wilayah Chicago dengan 100.000 server 8-core yang ditempatkan di 50 kontainer. Dalam superkomputer, array disk penyimpanan terpisah digunakan, sementara pusat data menggunakan disk lokal yang terpasang ke node server.

Cloud menawarkan manfaat yang signifikan bagi perusahaan TI dengan membebaskan mereka dari tugas tingkat rendah dalam menyiapkan perangkat keras (server) dan mengelola perangkat lunak sistem. Computing cloud menerapkan platform virtual dengan sumber daya elastis yang disatukan oleh penyediaan perangkat keras, perangkat lunak, dan kumpulan data sesuai permintaan, secara dinamis. Ide utamanya adalah untuk memindahkan computing desktop ke platform berorientasi layanan menggunakan cluster server dan database besar di pusat data. Computing cloud memanfaatkan biaya rendah dan kesederhanaannya bagi penyedia dan pengguna.

Arsitektur Cloud Umum

Arsitektur cloud yang sadar akan keamanan ditunjukkan pada Gambar 2.1. Cloud Internet dibayangkan sebagai sekelompok besar server. Server ini disediakan berdasarkan permintaan untuk melakukan layanan web kolektif atau aplikasi terdistribusi menggunakan sumber daya pusat data. Platform cloud dibentuk secara dinamis dengan menyediakan atau membatalkan penyediaan server, perangkat lunak, dan sumber daya basis data. Server di cloud dapat berupa mesin fisik atau mesin virtual. Antarmuka pengguna diterapkan untuk meminta layanan. Alat penyediaan mengukir sistem cloud untuk memberikan layanan yang diminta.



Gambar 2.1 Arsitektur generik sistem computing cloud, di mana server fisik divirtualisasikan sebagai instans VM di bawah kendali sistem manajemen sumber daya.

Pada dasarnya, lapisan bawah server fisik adalah perangkat keras dan infrastruktur mesin hosting. Lapisan atas mencakup aplikasi cloud untuk layanan pengguna. Di lapisan tengah aplikasi sering disebut middleware untuk tujuan virtualisasi dan manajemen sumber daya. Selain membangun cluster server, platform cloud menuntut penyimpanan terdistribusi dan layanan yang menyertainya. Sumber daya cloud computing dibangun di pusat data, yang biasanya dimiliki dan dioperasikan oleh penyedia pihak ketiga.

Konsumen tidak perlu mengetahui teknologi yang mendasarinya. Di cloud, perangkat lunak menjadi layanan. Cloud menuntut tingkat kepercayaan yang tinggi atas *Big data* yang diambil dari pusat *Big data*. Kita perlu membangun kerangka kerja untuk memproses data

berskala besar yang disimpan dalam sistem penyimpanan. Ini menuntut sistem file terdistribusi melalui sistem database. Sumber daya cloud lain yang ditambahkan ke platform cloud termasuk jaringan area penyimpanan, sistem basis data, firewall, dan perangkat keamanan. Penyedia layanan web menawarkan API khusus yang memungkinkan pengembang mengeksploitasi cloud Internet. Unit pemantauan dan pengukuran digunakan untuk melacak penggunaan dan kinerja sumber daya yang disediakan.

Mesin virtual

Beberapa VM dapat dimulai dan dihentikan sesuai permintaan pada satu mesin fisik untuk memenuhi permintaan layanan yang diterima, sehingga memberikan fleksibilitas maksimum untuk mengonfigurasi berbagai partisi sumber daya pada mesin fisik yang sama untuk persyaratan spesifik permintaan layanan yang berbeda. Selain itu, beberapa VM dapat menjalankan aplikasi secara bersamaan berdasarkan lingkungan sistem operasi yang berbeda pada satu mesin fisik, karena setiap VM diisolasi satu sama lain pada mesin fisik yang sama. Perlakuan mendalam VM dan kontainer diberikan di Bagian 2.2 dan 2.3 selanjutnya.

Infrastruktur perangkat lunak platform cloud harus menangani semua manajemen sumber daya dan melakukan sebagian besar pemeliharaan, secara otomatis. Perangkat lunak harus mendeteksi status setiap node, bergabung dan keluarnya server. dan mengerjakan tugas sebagaimana mestinya. Penyedia computing cloud, seperti Google dan Microsoft, telah membangun sejumlah besar pusat data di seluruh dunia. Setiap pusat data mungkin memiliki ribuan server. Lokasi pusat data dipilih untuk mengurangi biaya daya dan pendinginan. Dengan demikian, pusat data sering dibangun dekat dengan pembangkit listrik tenaga air. Pembangun platform fisik cloud lebih peduli tentang kinerja/rasio harga dan masalah keandalan daripada kecepatan geser kinerja.

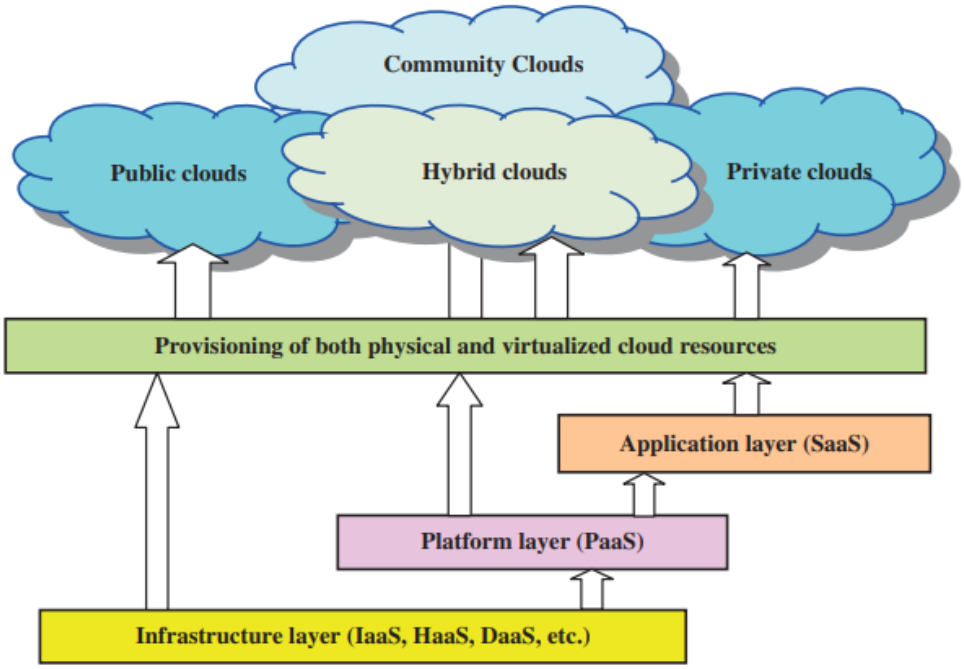
Beberapa karakteristik umum tercantum di bawah ini, berdasarkan apa yang ditentukan oleh NIST AS (Institut Standar dan Teknologi Nasional) untuk operasi cloud. Kami akan membahas persyaratan ini dalam bab-bab berikutnya:

- Skala besar dalam hal jumlah server yang digunakan, seringkali dalam puluhan ratusan atau hingga satu juta;
- Homogenitas dalam penggunaan server komponen, seringkali server ×-86 berbiaya rendah;
- Virtualisasi server banyak digunakan untuk menyediakan layanan pengguna multi-penyewa;
- Perangkat lunak komoditas dengan biaya yang hilang seperti menggunakan host berbasis Linux;
- Computing yang tangguh diperlukan dengan pemulihan bencana yang cepat dan toleransi kesalahan;
- Distribusi geografis dari beberapa pusat data untuk mengurangi latensi akses;
- Operasi cloud sering kali berorientasi pada layanan untuk menyediakan infrastruktur, platform, dan aplikasi;
- Keamanan tingkat lanjut dan perlindungan data diperlukan, diterapkan dalam perjanjian tingkat layanan.

Untuk mendukung hal di atas, sistem cloud saat ini harus dilengkapi dengan: i) pengumpulan sumber daya; ii) layanan terukur; iii) elastisitas yang cepat dalam konfigurasi ulang sistem; dan iv) akses jaringan pita lebar. Semua fitur di atas diperlukan untuk mendukung model layanan IaaS, PaaS, dan SaaS untuk semua cloud publik, provider, hybrid, dan komunitas. Secara keseluruhan, pengguna menuntut layanan mandiri sesuai permintaan dalam konstruksi cloud apa pun. Empat keluarga platform cloud dicirikan di bawah ini:

- **Cloud Publik:** Cloud publik dibangun melalui Internet, yang dapat diakses oleh pengguna mana pun yang telah membayar layanan tersebut. Cloud publik dimiliki oleh penyedia layanan. Mereka diakses dengan berlangganan. Cloud publik yang terkenal termasuk Google App Engine (GAE), Amazon Web Service (AWS), Microsoft Azure, IBM Blue Cloud, Sales-force Sales Clouds, dll. Penyedia ini menawarkan antarmuka jarak jauh yang dapat diakses publik untuk membuat dan mengelola instans VM dalam sistem.
- **Cloud Komunitas:** Ini adalah subkelas cloud publik yang sedang berkembang. Cloud ini muncul sebagai infrastruktur kolaboratif yang dimiliki oleh beberapa organisasi dengan beberapa kepentingan sosial atau bisnis yang sama, penemuan ilmiah, ketersediaan tinggi, dll. Cloud komunitas sering dibangun di atas beberapa pusat data. Dalam beberapa tahun terakhir, cloud komunitas telah berkembang pesat di sektor pendidikan, perusahaan bisnis, dan pemerintah untuk memenuhi pertumbuhan aplikasi *Big data*.
- **Private Clouds:** Private cloud dibangun di dalam domain intranet yang dimiliki oleh satu organisasi. Oleh karena itu, mereka dimiliki dan dikelola oleh klien. Cloud pribadi memberi pengguna lokal infrastruktur pribadi yang fleksibel dan gesit untuk menjalankan beban kerja layanan dalam domain administratif mereka. Private cloud seharusnya memberikan layanan cloud yang lebih efisien dan nyaman. Private cloud mungkin ingin mempertahankan kustomisasi yang lebih besar dan kontrol organisasi.
- **Cloud Hibrida:** Cloud hibrida dibangun dengan semua keluarga cloud. Private cloud mendukung model hybrid cloud dengan melengkapi infrastruktur lokal dengan kapasitas computing dari public cloud eksternal. Misalnya, *cloud computing* penelitian (RC2) adalah cloud pribadi yang dimiliki oleh IBM. RC2 menghubungkan sumber daya computing di 8 Pusat Penelitian IBM yang tersebar di AS, Eropa, dan Asia. Cloud hibrida menyediakan akses ke klien, jaringan mitra, dan pihak ketiga.

Singkatnya, cloud publik mempromosikan standardisasi, mempertahankan investigasi modal, dan menawarkan fleksibilitas aplikasi. Cloud pribadi berusaha mencapai penyesuaian dan menawarkan efisiensi, ketahanan, keamanan, dan privasi yang lebih tinggi. Cloud hibrida beroperasi di tengah dengan beberapa kompromi dalam berbagi sumber daya. Secara umum, private cloud lebih mudah dikelola, sedangkan public cloud lebih mudah diakses. Tren perkembangan cloud adalah semakin banyak cloud yang akan menjadi hybrid.



Gambar 2.2 Pengembangan arsitektur berlapis dari platform cloud untuk aplikasi IaaS, PaaS dan SaaS melalui Internet.

Platform Layanan Cloud Pengembangan Berlapis

Computing cloud menguntungkan industri jasa dan memajukan computing bisnis dengan paradigma baru. Telah diperkirakan bahwa pendapatan global dalam computing cloud dapat mencapai Rp 2.250.000 miliar pada tahun 2013 dari Rp 285.000 miliar yang dilaporkan pada tahun 2009. Keuntungan dasar computing cloud terletak pada penyediaan layanan di mana-mana, efisiensi sumber daya, dan fleksibilitas aplikasi. Pengguna dapat mengakses dan menyebarkan aplikasi dari mana saja di dunia dengan biaya yang sangat kompetitif.

Arsitektur cloud dikembangkan pada tiga lapisan: infrastruktur, platform dan aplikasi, seperti yang ditunjukkan pada Gambar 2.2. Ketiga lapisan pengembangan ini diimplementasikan dengan virtualisasi dan standarisasi sumber daya perangkat keras dan perangkat lunak yang disediakan di dalam cloud. Layanan ke cloud publik, pribadi dan hibrida disampaikan kepada pengguna melalui dukungan jaringan melalui Internet dan intranet yang terlibat. Jelas bahwa lapisan infrastruktur dikerahkan terlebih dahulu untuk mendukung jenis layanan IaaS. Lapisan infrastruktur ini berfungsi sebagai dasar untuk membangun lapisan platform cloud untuk mendukung layanan PaaS. Pada gilirannya, lapisan platform adalah dasar untuk mengimplementasikan lapisan aplikasi untuk aplikasi SaaS.

Lapisan infrastruktur dibangun dengan computing virtual, penyimpanan, dan sumber daya jaringan. Abstraksi dari sumber daya perangkat keras ini dimaksudkan untuk memberikan fleksibilitas yang diminta oleh pengguna. Secara internal, virtualisasi mewujudkan penyediaan sumber daya secara otomatis dan mengoptimalkan proses manajemen infrastruktur. Perlu dicatat bahwa tidak semua layanan cloud dibatasi untuk satu lapisan. Banyak aplikasi mungkin menerapkan sumber daya pada lapisan campuran. Lagi pula, ketiga lapisan itu dibangun dari bawah ke atas dengan hubungan ketergantungan.

Lapisan platform adalah untuk tujuan umum dan penggunaan berulang dari kumpulan sumber daya perangkat lunak. Lapisan ini menyediakan pengguna dengan lingkungan untuk mengembangkan aplikasi mereka, untuk teks aliran operasi dan untuk memantau hasil dan kinerja eksekusi. Platform harus dapat meyakinkan pengguna tentang skalabilitas, ketergantungan, dan perlindungan keamanan. Di satu sisi, platform cloud virtual berfungsi sebagai "sistem middleware" antara infrastruktur dan lapisan aplikasi cloud.

Lapisan aplikasi dibentuk dengan kumpulan semua modul perangkat lunak yang dibutuhkan untuk aplikasi SaaS. Aplikasi layanan di lapisan ini mencakup pekerjaan manajemen kantor sehari-hari, seperti pencarian informasi, pemrosesan dokumen, dan layanan kalender dan otentikasi, dll. Lapisan aplikasi juga banyak digunakan oleh perusahaan bisnis dalam pemasaran dan penjualan bisnis, manajemen hubungan konsumen (CRM), transaksi keuangan, manajemen rantai pasokan, dll.

Dari perspektif penyedia, layanan di berbagai lapisan menuntut jumlah dukungan fungsi dan manajemen sumber daya yang berbeda oleh penyedia. Secara umum, SaaS menuntut pekerjaan paling banyak dari penyedia, PaaS di tengah, dan IaaS paling sedikit. Misalnya, Amazon EC2 tidak hanya menyediakan sumber daya CPU tervirtualisasi kepada pengguna tetapi juga pengelolaan sumber daya yang disediakan ini. Layanan di lapisan aplikasi menuntut lebih banyak pekerjaan dari penyedia. Contoh terbaik adalah layanan CRM Salesforce di mana penyedia tidak hanya memasok perangkat keras di lapisan bawah dan perangkat lunak di lapisan atas, tetapi juga menyediakan platform dan alat perangkat lunak untuk pengembangan dan pemantauan aplikasi pengguna:

- **Infrastructure as a Service (IaaS):** Model ini menyatukan infrastruktur yang diminta oleh pengguna, yaitu server, penyimpanan, jaringan, dan fabric pusat data. Pengguna dapat menerapkan dan menjalankan beberapa VM yang menjalankan OS tamu pada aplikasi tertentu. Pengguna tidak mengelola atau mengontrol infrastruktur cloud yang mendasarinya, tetapi dapat menentukan kapan harus meminta dan merilis VM dan data yang diperlukan. Contoh IaaS terbaik adalah AWS, GoGrid, Rackspace, Eucalyptus, flexscale, RightScale, dll.
- **Platform as a Service (PaaS):** Model ini menyediakan pengguna untuk menyebarkan aplikasi yang dibuat pengguna ke platform cloud virtual. PaaS mencakup middleware, database, alat pengembangan dan beberapa dukungan runtime seperti Web 2.0 dan Java, dll. Platform ini mencakup perangkat keras dan perangkat lunak yang terintegrasi dengan antarmuka pemrograman tertentu. Penyedia menyediakan API dan perangkat lunak (misalnya Java, Python, Web2.0, .Net). Pengguna dibebaskan dari pengelolaan infrastruktur cloud. PaaS menyediakan lingkungan pemrograman untuk membangun dan mengelola aplikasi cloud. Contoh terbaik dari platform PaaS adalah Google AppEngine, Windows Azure, Force.com, dll.
- **Software as a Service (SaaS):** Ini mengacu pada perangkat lunak aplikasi yang diprakarsai oleh browser yang dikirimkan ke ribuan pelanggan cloud berbayar. Model SaaS berlaku untuk proses bisnis, aplikasi industri, CRM (manajemen hubungan konsumen), ERP (perencanaan sumber daya perusahaan bisnis), SDM (sumber daya manusia) dan aplikasi kolaboratif. Di sisi pelanggan, tidak ada investasi di muka untuk server atau lisensi

perangkat lunak. Di sisi penyedia, biayanya agak rendah, dibandingkan dengan hosting konvensional untuk aplikasi pengguna. Contoh SaaS terbaik adalah Cloudera, Hadoop, salesforce.com, .NETService, Google Docs, Layanan Microsoft Dynamic CRM, layanan Share-Point, dll.

Cloud Internet dibayangkan sebagai sekelompok server publik yang disediakan berdasarkan permintaan untuk melakukan layanan web kolektif atau aplikasi terdistribusi menggunakan sumber daya pusat data. Tujuan desain cloud ditentukan di bawah ini. Kemudian kami menyajikan prinsip dasar di balik desain arsitektur cloud.

Tujuan Desain Platform Cloud

Skalabilitas, virtualisasi, efisiensi, dan keandalan adalah empat tujuan desain utama dari platform computing cloud. Cloud mendukung aplikasi Web 2.0. Manajemen cloud menerima permintaan pengguna dan menemukan sumber daya yang benar, dan kemudian memanggil layanan penyediaan yang memanggil sumber daya di cloud. Perangkat lunak manajemen cloud perlu mendukung mesin fisik dan virtual. Keamanan dalam sumber daya bersama dan akses bersama pusat data juga menimbulkan tantangan desain lainnya.

Platform perlu membangun infrastruktur HPC berskala sangat besar. Sistem perangkat keras dan perangkat lunak digabungkan bersama untuk membuatnya mudah dan efisien untuk dioperasikan. Skalabilitas sistem dapat mengambil manfaat dari arsitektur cluster. Jika satu layanan membutuhkan banyak daya pemrosesan atau kapasitas penyimpanan atau lalu lintas jaringan, cukup mudah untuk menambahkan lebih banyak server dan bandwidth. Keandalan sistem dapat mengambil manfaat dari arsitektur ini. Data dapat dimasukkan ke beberapa lokasi. Misalnya, email pengguna dapat dimasukkan ke dalam tiga disk yang diperluas ke pusat data geografis yang berbeda. Dalam situasi seperti itu, bahkan jika salah satu pusat data mogok, data pengguna masih dapat diakses. Skala arsitektur cloud dapat dengan mudah diperluas dengan menambahkan lebih banyak server dan memperbesar konektivitas jaringan yang sesuai.

Model Cloud untuk Penyimpanan dan Pemrosesan *Big data*

Penggunaan cloud terbesar adalah di area layanan bisnis. Tabel 2.2 mengklasifikasikan berbagai jenis cloud komersial ke dalam lima kategori bisnis, yaitu aplikasi, platform, computing/penyimpanan, co-location, dan cloud jaringan. Penyedia layanan cloud perwakilan terdaftar. Kami telah memperkenalkan tiga lapisan layanan teratas sebagai SaaS, PaaS dan IaaS, masing-masing. Platform cloud menyediakan PaaS, yang berada di atas infrastruktur IaaS. Di kategori teratas, cloud menawarkan layanan aplikasi perangkat lunak (SaaS). Implikasinya adalah kami tidak dapat meluncurkan aplikasi SaaS tanpa platform cloud. Platform cloud tidak dapat dibangun jika infrastruktur computing dan penyimpanan tidak ada. Namun, pengembang dapat menyewa cloud tingkat yang lebih rendah untuk membangun platform atau portal aplikasi tingkat yang lebih tinggi.

Tabel 2.2 Lima layanan cloud bisnis dan penyedia perwakilan.

Kategori Cloud	Penyedia Layanan Cloud
Cloud Aplikasi	OpenTable, KeneXa, Netsuite, RightNow, Webex, Balckbaud, Consur Cloud, Telco, Omiture, Vocus, Microsoft OWA (office 365), Google Gmail, Yahoo! Hotmail
Cloud Platform	force,com, Google AppEngine, Facebook, IBM BlueCloud, postini, Layanan SQL, Twitter, postini, MicroSoft Azure, SGI Cyclone, Amazon EMR
Cloud computing dan Penyimpanan	Amazon AWS, Rackspace, OpSource, GoGrid, MeePo, Flexiscale, HP Cloud, Banknorth, VMware, XenEnterprise, iCloud
Cloud Lokasi Bersama	Savvis, Internap, Realitas Digital, Kepercayaan, 365 Utama
Cloud Jaringan	AboveNet, AT&T, Qwest, NTTCommunications

Dua contoh cloud konkret diberikan di bawah ini untuk menyoroti penggunaan komersial cloud publik. Lebih banyak studi kasus cloud besar dalam kategori yang berbeda akan dibahas di Bagian 2.4.

Contoh 2.1 Apple iCloud untuk Penyimpanan, Cadangan, dan Banyak Layanan Pribadi

Pada tahun 2011, Apple Inc. meluncurkan iCloud sebagai penyimpanan cloud dan layanan computing cloud. Salah satu pusat data iCloud Apple terletak di Maiden, Carolina Utara. Pada tahun 2015, layanan iCloud memiliki lebih dari 500 juta pengguna. iCloud menyediakan sarana bagi penggunanya untuk menyimpan dokumen, foto, dan musik di server jarak jauh di pusat data Apple untuk diunduh ke perangkat iOS, Macintosh, atau Windows. Cloud membagikan dan mengirim data ke pengguna lain, dan mengelola perangkat Apple mereka jika hilang atau dicuri.

Layanan iCloud juga menyediakan sarana untuk mencadangkan perangkat iOS secara nirkabel langsung ke iCloud, alih-alih bergantung pada pencadangan manual ke komputer host Mac atau Windows menggunakan layanan iTunes. Sistem ini juga memungkinkan pengguna menggunakan layanan nirkabel AirDrop untuk berbagi foto, musik, dan game secara instan dengan menautkan ke akun seluler mereka. Ini juga bertindak sebagai pusat sinkronisasi data untuk email, kontak, kalender, bookmark, catatan, pengingat (daftar tugas), dokumen iWork, foto, dan data lainnya. Perangkat iOS cadangan langsung ke iCloud, bertindak sebagai pusat sinkronisasi data dan mengelola perangkat Apple jika hilang atau dicuri, merupakan peningkatan besar dari masa lalu menggunakan iTunes secara tidak langsung.

Jenis *Big data* yang disimpan di iCloud termasuk kontak, kalender, penanda, pesan email, catatan, album foto bersama, perpustakaan foto iCloud, aliran foto saya, iMessages, pesan teks (SMS) dan MMS, dll. Dokumen yang disimpan di iCloud menggunakan iOS dan aplikasi Mac di situs web iCloud.com. Jenis data dan pengaturan yang disimpan di perangkat seluler Anda (iPhone, iPad, dll.) dicadangkan oleh iCloud setiap hari, bahkan termasuk riwayat pembelian untuk musik, film, acara TV, aplikasi, dan buku. iCloud juga menawarkan fitur menarik untuk mencari teman. Pengguna Cari Teman Saya membagikan lokasi mereka dengan

orang yang mereka pilih. Lokasi ditentukan menggunakan GPS di perangkat iOS saat Layanan Lokasi dihidupkan.

Notifikasi muncul ketika pengguna meminta pengguna lain untuk melihat di mana mereka berada. Lokasi Anda dikirim dari perangkat Anda saat seseorang meminta untuk melihat lokasi Anda. Fitur ini dapat dihidupkan dan dimatikan kapan saja. Untuk menemukan iPhone yang salah tempat atau dicuri, Anda dapat memutar suara pada volume maksimum, dan membuat kedipan di layar meskipun dimatikan. Fitur ini berguna jika perangkat salah letak. Anda juga dapat menandai perangkat dalam mode hilang: Pengguna dapat menguncinya dengan kode sandi. Orang yang menemukan telepon dapat menghubungi pemiliknya secara langsung di perangkat yang hilang. Sistem ini juga dapat menghapus semua catatan sensitif iPhone pada ponsel yang dicuri.

Contoh 2.2 Layanan Cloud Co-Location oleh Savvis

Layanan co-location berkaitan dengan pengelolaan pusat data, di mana peralatan, ruang, dan bandwidth tersedia untuk disewakan kepada pelanggan ritel. Fasilitas co-location menyediakan ruang, daya, pendinginan dan keamanan fisik untuk server, penyimpanan dan peralatan jaringan dari beberapa cloud yang berinteraksi satu sama lain melalui telekomunikasi dan penyedia layanan jaringan. Savvis adalah perusahaan semacam itu, didirikan pada tahun 1996. Mereka menyediakan hosting web dan layanan lokasi bersama, termasuk perumahan cloud dan catu daya, manajemen infrastruktur, jaringan, dan layanan keamanan untuk sumber daya fisik dan jaringan dari banyak pusat data.

Apple adalah pelanggan besar pertama mereka. Dengan memanfaatkan referensi dan testimonial pelanggan Apple Computer, Savvis menutup kontrak besar tambahan dengan penyedia cloud lainnya. Perusahaan ini menjual hosting terkelola dan layanan lokasi bersama dengan lebih dari 50 pusat data di seluruh Amerika Utara, Eropa, dan Asia, sistem manajemen dan penyediaan otomatis, serta konsultasi teknologi informasi. Pada tahun 2015, Savvis memiliki sekitar 2.500 pelanggan bisnis dan pemerintah. Savvis diposisikan bersama 19 penyedia hosting web lainnya, termasuk AT&T, Rackspace, Verizon Business, Terremark dan Sungard dalam menyediakan layanan cloud co-location.

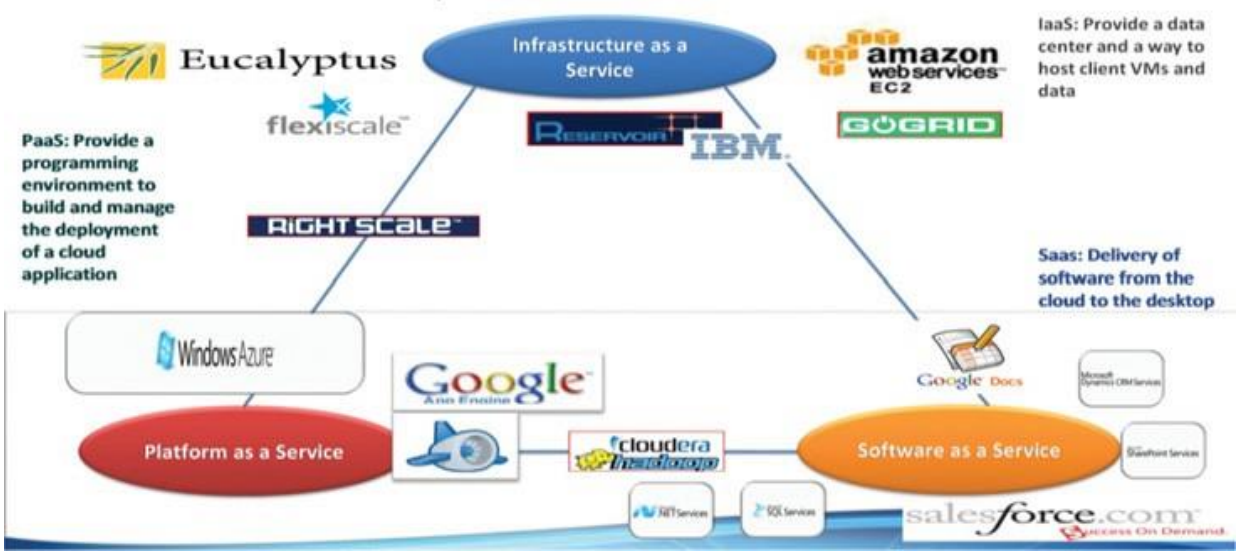
Di masa lalu, perusahaan mengalami pasang surut. Pada tahun 2006, layanan jaringan pengiriman konten (CDN) Savvis sedang booming. Mereka tumbuh pesat dengan aset jaringan, kontrak pelanggan, dan kekayaan intelektual yang digunakan dalam bisnis CDN Savvis. Beberapa pelajaran yang mereka pelajari adalah dalam tuduhan dukungan spam dan pelanggaran keamanan pada manajemen cloud IaaS. Salah satu insiden adalah tuduhan meminta bisnis dari spammer untuk keuntungan. Sebagai hasil dari perhatian negatif media, Savvis melanjutkan bisnisnya dengan menggunakan Spamhaus (organisasi penangkal spam di seluruh dunia) untuk mencegah serangan spam pelanggan.

Ian Foster mendefinisikan computing cloud sebagai berikut:

> *Paradigma computing terdistribusi skala besar yang didorong oleh skala ekonomi, di mana kumpulan daya computing, penyimpanan, platform, dan layanan terkelola yang tervirtualisasi, terukur secara dinamis, dan terkelola secara abstrak, dikirimkan sesuai permintaan kepada pelanggan eksternal melalui Internet.*

Computing cloud menawarkan paradigma computing berdasarkan permintaan. Tiga model layanan cloud dasar diberikan di bawah ini. Gambar 2.3 menunjukkan lanskap cloud dengan penyedia cloud utama. Ketiga model layanan cloud diterapkan. Cloud internet menawarkan empat mode penyebaran: pribadi, publik, komunitas, dan hibrida. Mode ini menuntut tingkat implikasi keamanan yang berbeda. SLA yang berbeda menyiratkan bahwa keamanan menjadi tanggung jawab bersama dari semua penyedia cloud, konsumen sumber daya cloud, dan penyedia perangkat lunak berkemampuan cloud pihak ketiga. Keuntungan computing cloud telah diadvokasi oleh banyak pakar TI, pemimpin industri, dan peneliti ilmu komputer.

Computing cloud menerapkan platform tervirtualisasi dengan sumber daya elastis sesuai permintaan dengan menyediakan perangkat keras, perangkat lunak, dan kumpulan data, secara dinamis. Idenya adalah untuk memindahkan computing desktop ke platform berorientasi layanan menggunakan cluster server dan database besar di pusat data. Computing cloud memanfaatkan biaya rendah dan kesederhanaannya yang menguntungkan pengguna dan penyedia. Virtualisasi mesin telah memungkinkan efektivitas biaya tersebut. Computing cloud bermaksud untuk memuaskan banyak aplikasi pengguna secara bersamaan. Ekosistem cloud harus dirancang agar aman, dapat dipercaya, dan dapat diandalkan. Jika tidak, ini dapat menghalangi pengguna untuk menerima layanan yang dialihdayakan.



Gambar 2.3 Tiga model layanan cloud yang digunakan oleh penyedia utama. (Dicetak ulang dengan izin dari Dennis Gannon, alamat Keynote di IEEE Cloudcom2010)

Persyaratan Penyimpanan Big data

Pada tahun 2015, total data yang tersimpan dalam segala bentuk di Bumi diperkirakan 300+ EB, dengan tingkat pertumbuhan tahunan sebesar 28%. Namun, total data yang dikirimkan di antara semua sumber yang mungkin adalah sekitar 1900+ EB per tahun (http://www.martinhilbert.net/WorldInfoCapacity.html).

Di masa lalu, sebagian besar item informasi diekspresikan dalam format analog. Perangkat penyimpanan digital menjadi populer sejak tahun 2002 dan menggantikan sebagian besar perangkat analog dengan cepat. Tabel 2.3 menunjukkan bahwa pada tahun 2007 hanya

6% (19 EB) yang menggunakan perangkat analog dan 94% (280 EB) adalah perangkat digital. Item analog terutama disimpan pada kaset audio/video (94%). Informasi digital tersebar di berbagai jenis perangkat penyimpanan. Hard drive PC/server memberikan kontribusi terbesar (44,5%), termasuk yang digunakan di pusat *Big data*. Berikutnya adalah perangkat DVD dan Blue Ray (22,8%). Jelas, perangkat penyimpanan sekunder masih pra-
mendominasi spektrum penyimpanan.

Sumber Daya Cloud untuk Mendukung Analisis *Big data*

Ekosistem cloud berubah menuju aplikasi *Big data*. Computing cloud, penginderaan IoT, basis data, dan teknologi visualisasi sangat diperlukan untuk menganalisis *Big data*. Teknologi ini memainkan peran mendasar dalam layanan kognitif, kecerdasan bisnis, *Machine learning*, pengenalan wajah, pemrosesan bahasa alami, dll. Array data multidimensi yang dikenal sebagai tensor, yang dapat ditangani oleh library TensorFlow, dipelajari di Bab 9 dan 10 Teknologi tambahan yang penting untuk manajemen *Big data* termasuk penambangan data, sistem file terdistribusi, jaringan seluler, dan infrastruktur berbasis cloud.

Tabel 2.3 Kapasitas penyimpanan informasi global dalam hal total byte pada tahun 2007

Teknologi	Perangkat penyimpanan	>Distribusi
Analog, 19 EB, 6% dari total	Kertas, film, pita audio, dan vinil	6%
	Kaset video analog	94%
Teknologi	Media portabel dan flash drive	2%
	Hard disk portabel	2.4%
	CD dan mini disk	6.8%
	kaset digital	11,8%
	DVD dan sinar biru	22,8%
	Hard disk PC/server	44,5%
	Lainnya (kartu memori, floppy disk, ponsel, PDS, kamera, video game, dll.)	> 1%

Di AS, program analisis data topologi yang didukung oleh DARPA (Defense Advanced Research Project Agency) mencari struktur dasar kumpulan *Big data*-besaran. Untuk menggunakan analitik *Big data*, sebagian besar pengguna lebih memilih penyimpanan yang terpasang langsung seperti solid state drive (SSD) dan disk terdistribusi di kluster cloud. Jaringan area penyimpanan tradisional (SAN) dan penyimpanan yang terpasang ke jaringan (NAS) terlalu lambat untuk memenuhi permintaan analisis *Big data*. Desainer cloud harus memperhatikan kinerja sistem, infrastruktur komoditas, biaya rendah, dan respons waktu nyata terhadap pertanyaan.

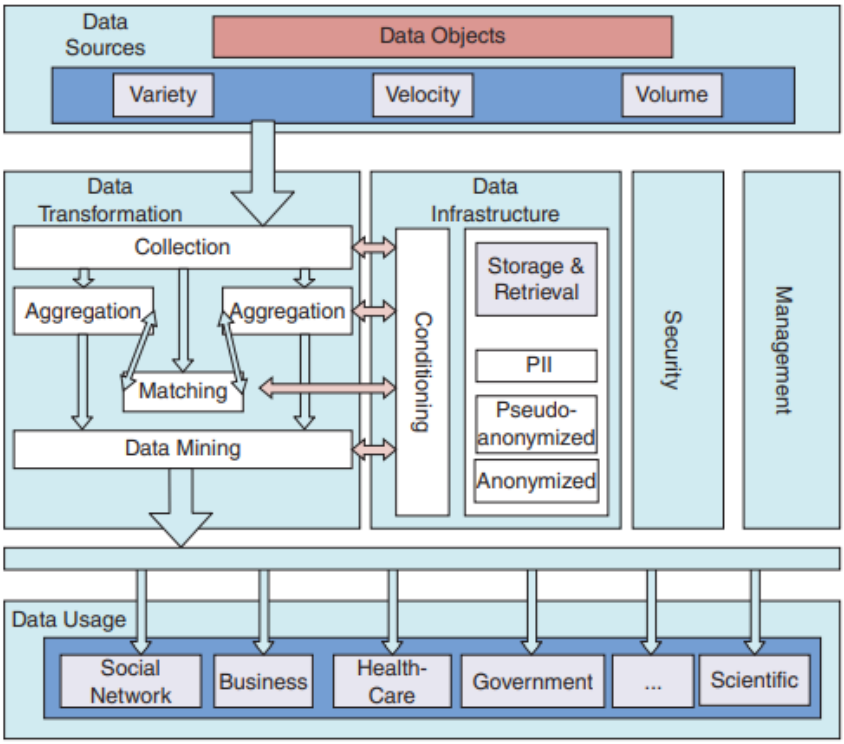
Masalah latensi akses cloud juga menjadi perhatian utama dalam penggunaan cloud. Kekhawatiran lain adalah masalah skalabilitas karena kumpulan data tumbuh secara dramatis. Penyimpanan bersama memiliki keuntungan lebih cepat, tetapi tidak memiliki skalabilitas. Praktisi analitik *Big data* lebih memilih penyimpanan terdistribusi dalam kelompok besar karena skalabilitas dan pertimbangan biaya rendah. *Big data* untuk manufaktur menuntut

infrastruktur untuk transparansi. Manufaktur prediktif menawarkan pendekatan yang menarik menuju waktu henti, ketersediaan, dan produktivitas yang mendekati nol.

Arsitektur Platform Cloud Big data

Tercantum di bawah ini adalah masalah penting yang harus ditangani untuk merancang cloud pintar untuk aplikasi *Big data*:

- **Pra-pemrosesan data tidak terstruktur:** Basis data relasional tradisional tidak dapat mendukung data tidak terstruktur, sehingga kami menuntut pemrosesan data tidak lengkap NoSQL dari sumber yang bising dan kotor. Data ini sering kali tidak memiliki kebenaran atau tidak dapat dilacak. Banyak blog atau pertukaran sosial tidak dapat diverifikasi dengan mudah, memerlukan penyaringan data dan kontrol integritas.
- **Grafik sosial, API, dan alat visualisasi** diperlukan untuk menangani data media sosial yang tidak terstruktur secara efektif. Ini menuntut cloud yang hemat biaya dan sistem file terdistribusi untuk mengumpulkan, menyimpan, memproses, dan menganalisis *Big data*. Teknik bottom-up diperlukan untuk mengungkap struktur dan pola yang tidak diketahui.
- **Alat perangkat lunak analitik data** diperlukan di cloud *Big data*. Dalam bab-bab berikutnya, kami akan membahas beberapa alat sumber terbuka atau komersial untuk analisis *Big data*. Alat-alat ini harus diintegrasikan untuk memaksimalkan efek kolaboratif mereka. Intelijen bisnis harus ditingkatkan ke statistik induktif atau mendukung analitik prediktif dalam pengambilan keputusan penting.
- ***Machine learning* dan algoritme analitik cloud** sangat diminati untuk mesin atau *Deep learning* yang diawasi atau tidak diawasi. Ini akan dipelajari di Bab 4, 5, dan 6. Ilmuwan data harus memiliki pengetahuan domain yang memadai, penambangan data statistik, ilmu sosial, dan keterampilan pemrograman. Oleh karena itu menuntut para ahli dari domain crossover untuk bekerja secara kooperatif.
- **Tata kelola data dan keamanan** menuntut privasi data, kontrol integritas, kepatuhan SLA, akuntabilitas, dan manajemen kepercayaan, dll. Kontrol keamanan harus diterapkan dalam skala global. Privasi data harus dijaga hingga ke tingkat kontrol akses yang halus.



Gambar 2.4 Arsitektur konseptual sistem cloud modern untuk aplikasi computing *Big data*.

Alur Kerja di Mesin Pemroses Big data

Pada Gambar 2.4, kami menunjukkan alur kerja konseptual di cloud analitik data yang khas. *Big data* datang sebagai blok data atau aliran data dari berbagai sumber dari atas. Sumber daya platform cloud dibagi menjadi empat bagian infrastruktur: terutama digunakan untuk menyimpan, mengambil, mengubah, dan memproses data yang mengalir melalui sekelompok besar server yang membentuk inti cloud. Unit manajemen sumber daya dan keamanan mengontrol keamanan dan dibagi menjadi dua bagian. Di sebelah kanan, mekanisme kontrol aliran data mengelola pergerakan data melalui mesin cloud di sebelah kiri. Mesin ini melakukan berbagai fungsi transformasi data, termasuk pengumpulan, agregasi, pencocokan, dan operasi penambangan, sebelum memasukkan data yang diekstraksi atau diurutkan ke berbagai aplikasi di kotak bawah.

Kami mempertimbangkan beberapa persyaratan utama yang harus dilakukan oleh *smart cloud* dalam aplikasi computing *Big data* yang khas. Layanan cloud ini dilakukan pada lima lapisan: sumber data, pemrosesan, kontrol akses, manajemen insiden, dan perlindungan privasi yang ditampilkan sebagai judul baris. Jelas, mekanisme, kebijakan, dan kemampuan analitik ini harus dibangun ke dalam sistem analitik cloud modern. Secara khusus, privasi data dan keamanan cloud menjadi sangat penting di antara kelima lapisan pemrosesan.

2.2 PEMBUATAN MESIN VIRTUAL DAN KONTAINER DOCKER

Pusat data tradisional dibangun dengan klaster server berskala besar. Cluster besar tersebut digunakan tidak hanya untuk menyimpan database yang besar tetapi juga untuk membangun mesin pencari yang cepat.

Tabel 2.4 Sumber daya virtualisasi dan produk perangkat lunak yang representatif.

Virtualisasi	Deskripsi singkat	Produk Perwakilan
Server	Beberapa VM dipasang di server untuk	XenServer, PowerVM, Hyper/V, VMware EXS Server, dll.
Desktop	tingkatkan tingkat pemanfaatan server rusak	VMware workstation, VMwarew ACE, XenDesktop, Virtual PC, dll.
jaringan	Tingkatkan fleksibilitas aplikasi pada PC dan workstation	Virtualisasi intranet, OpenStack, Eucalyptus, dll.
Penyimpanan	Jaringan pribadi virtual (VPN), jaringan area lokal virtual, kluster virtual di cloud	DropBox, Apple iCloud, AWS S3, MS One Drive, IBM Datastore, dll.
Aplikasi	Penyimpanan jaringan dan virtualisasi NAS untuk cluster bersama atau aplikasi cloud	Wadah dok, XenApp, MS CRM, berbagai cloud SaaS Salesforce, dll.

Sejak diperkenalkannya virtualisasi, semakin banyak cluster pusat data yang diubah menjadi cloud. Google, Amazon, dan Microsoft semuanya membangun platform cloud mereka dengan cara ini. Di bagian ini, kami membahas teknik virtualisasi sumber daya. Kedua hypervisor dan mesin Docker diperkenalkan. Virtualisasi dapat dilakukan pada tingkat proses perangkat lunak, tingkat sistem host atau pada berbagai tingkat yang diperluas.

Tabel 2.4 merangkum lima tingkat virtualisasi sumber daya. Beberapa produk representatif juga terdaftar. Di antaranya, virtualisasi server sangat diperlukan dalam mengubah pusat data menjadi cloud yang beroperasi untuk melayani sejumlah besar pengguna pada saat yang bersamaan. Tujuan utama dari virtualisasi server adalah untuk meningkatkan elastisitas cluster dan meningkatkan pemanfaatan server bersama. Virtualisasi desktop mencoba memberikan fleksibilitas aplikasi oleh pengguna individu. Penyimpanan virtual dan jaringan virtual membuat cloud lebih kuat untuk operasi lokasi bersama. Virtualisasi aplikasi mengacu pada virtualisasi tingkat proses perangkat lunak.

Virtualisasi Sumber Daya Mesin

Konsep virtualisasi komputer dimulai pada tahun 1960-an. Ini adalah teknik untuk mengabstraksi sumber daya mesin secara logis pada tingkat arsitektur yang berbeda. Memori virtual adalah contoh khas untuk memperluas memori fisik di luar kapasitas fisiknya dengan mengizinkan pertukaran halaman antara disk fisik dan ruang alamat virtual. Di bagian ini, kami memperkenalkan konsep kunci virtualisasi perangkat keras dan jenis virtualisasi lainnya. Adalah adil untuk mengamati bahwa tidak ada cloud elastis yang dapat dibangun untuk memenuhi operasi multi-penyewaan tanpa virtualisasi sumber daya.

Virtualisasi Perangkat Keras

Ini mengacu pada penggunaan perangkat lunak khusus untuk membuat mesin virtual (VM) pada mesin perangkat keras host. VM ini bertindak seperti komputer nyata dengan OS tamu. Mesin host adalah mesin sebenarnya di mana VM dijalankan, dengan VM dan mesin host mungkin berjalan dengan OS yang berbeda. Perangkat lunak yang membuat VM pada

perangkat keras host disebut hypervisor atau monitor mesin virtual (VMM). Tiga pendekatan virtualisasi perangkat keras ditentukan di bawah ini:

- **Virtualisasi penuh:** Ini mengacu pada simulasi lengkap atau terjemahan dari perangkat keras host ke beberapa jenis CPU virtual, memori virtual atau disk virtual untuk digunakan oleh VM menggunakan OS sendiri yang tidak dimodifikasi.
- **Virtualisasi parsial:** Ini mengacu pada fakta bahwa beberapa sumber daya yang dipilih divirtualisasikan dan beberapa tidak. Oleh karena itu, beberapa program tamu harus dimodifikasi untuk berjalan di lingkungan seperti itu.
- **Para-virtualisasi:** Dalam hal ini, lingkungan perangkat keras VM tidak divirtualisasikan. Aplikasi tamu dieksekusi dalam domain yang terisolasi atau terkadang wadah perangkat lunak pemanggil. OS tamu tidak lagi digunakan. Sebagai gantinya, VMM dipasang di ruang pengguna untuk memandu eksekusi program pengguna.

Virtualisasi di Berbagai Tingkat Abstraksi

Lima tingkat abstraksi ditentukan dalam Tabel 2.5 untuk mengimplementasikan VM. Pada tingkat ISA (arsitektur set instruksi), VM dibuat dengan meniru satu ISA yang diberikan oleh yang lain. Pendekatan ini memberikan kinerja terendah karena proses emulasi yang lambat. Namun, ini memberikan fleksibilitas aplikasi yang sangat tinggi. Beberapa penelitian akademis tentang VM menggunakan pendekatan ini seperti Dynamo, dll.

Performa VM tertinggi berasal dari virtualisasi pada level bare-metal atau OS. Hypervisor XEN yang terkenal menciptakan CPU virtual, memori virtual, dan disk virtual tepat di atas perangkat fisik bare-metal. Namun, virtualisasi tingkat perangkat keras menghasilkan kompleksitas yang paling tinggi. Contoh terbaik dari virtualisasi tingkat OS adalah wadah Docker. Virtualisasi di perpustakaan run-time dan tingkat aplikasi pengguna menghasilkan kinerja rata-rata. Membuat VM di tingkat aplikasi pengguna menyebabkan tingkat isolasi aplikasi yang tinggi dengan mengorbankan upaya implementasi yang sangat kompleks oleh pengguna. Kita harus menggunakan hypervisor untuk membuat VM di tingkat perangkat keras dan penggunaan wadah Docker di tingkat kernel Linux. Menerapkan VM di ISA, tingkat perpustakaan pengguna atau run-time sebagian besar dilakukan oleh akademisi, tetapi jarang dipraktikkan di industri karena kinerjanya yang rendah.

Sebagian besar virtualisasi menggunakan pendekatan perangkat lunak atau firmware untuk menghasilkan VM. Namun, kami juga dapat menggunakan pendekatan yang dibantu perangkat keras untuk membantu virtualisasi. Intel telah memproduksi VT-x untuk tujuan ini, untuk meningkatkan efisiensi prosesornya di lingkungan VM. Ini memerlukan modifikasi CPU untuk menyediakan dukungan perangkat keras untuk virtualisasi. Jenis virtualisasi lain juga muncul di virtualisasi desktop, virtualisasi memori dan penyimpanan, dan berbagai tingkat virtualisasi diperkenalkan pada Tabel 2.6. Bahkan ada yang mempertimbangkan virtualisasi data dan jaringan. Misalnya, jaringan pribadi virtual (VPN) memungkinkan jaringan virtual dibuat melalui Internet. Virtualisasi memungkinkan konsep computing cloud. Perbedaan utama antara computing grid tradisional dan cloud saat ini terletak pada penggunaan sumber daya virtual.

Tabel 2.5 Manfaat relatif virtualisasi pada lima tingkat abstraksi.

Tingkat Virtualisasi	**Deskripsi Fungsional**	**Contoh Paket**	**Keunggulan, Fleksibilitas/Isolasi Aplikasi, Kompleksitas Implementasi**
Arsitektur Set Instruksi	Emulasi ISA tamu oleh tuan rumah	Dinamo, Burung, Bochs, Crusoe	Performa rendah, fleksibilitas aplikasi tinggi, kompleksitas median dan isolasi
Virtualisasi tingkat perangkat keras	Virtualisasi di atas perangkat keras bare metal	XEN, VMWare, PC Virtual	Performa dan kompleksitas tinggi, fleksibilitas aplikasi rata-rata, dan isolasi aplikasi yang baik
Tingkat Sistem Operasi	Wadah aplikasi pengguna yang terisolasi dengan sumber daya yang terisolasi	Mesin Docker, Penjara, FVM,	Performa tertinggi, fleksibilitas Aplikasi rendah, dan isolasi terbaik serta kompleksitas rata-rata
Tingkat Perpustakaan Run-Time	Membuat VM melalui perpustakaan run-time melalui kait API	Anggur, cCUDA, WABI, LxRun	Performa rata-rata, fleksibilitas dan isolasi aplikasi rendah, serta kompleksitas rendah
Tingkat Aplikasi Pengguna	Terapkan VM HLL di tingkat aplikasi pengguna	JVM, .NET CLR, Panot	Performa rendah dan fleksibilitas aplikasi, kompleksitas sangat tinggi, dan isolasi aplikasi

Tabel 2.6 Hypervisor atau monitor mesin virtual untuk menghasilkan mesin virtual.

Hypervisor	**CPU tuan rumah**	**OS tuan rumah**	**OS tamu**	**Arsitektur, Aplikasi dan Komunitas Pengguna**
XEN	×-86, ×-86–64, IA-64	NetBSD, Linux	Linux, Windows, BSD, Linux, Solaris	Hypervisor Asli (Contoh 1.6) yang dikembangkan di Universitas Cambridge
KVM	×-86, ×-86–64, IA-64, S390, PowerPC	Linux	Linux, Windows, FreeBSD, Solaris	Hosted Hypervisor berdasarkan para-virtualisasi di ruang pengguna
Hyper V	×-86 berbasis	Server 2003	Server Windows	Hypervisor asli berbasis Windows, dipasarkan oleh Microsoft

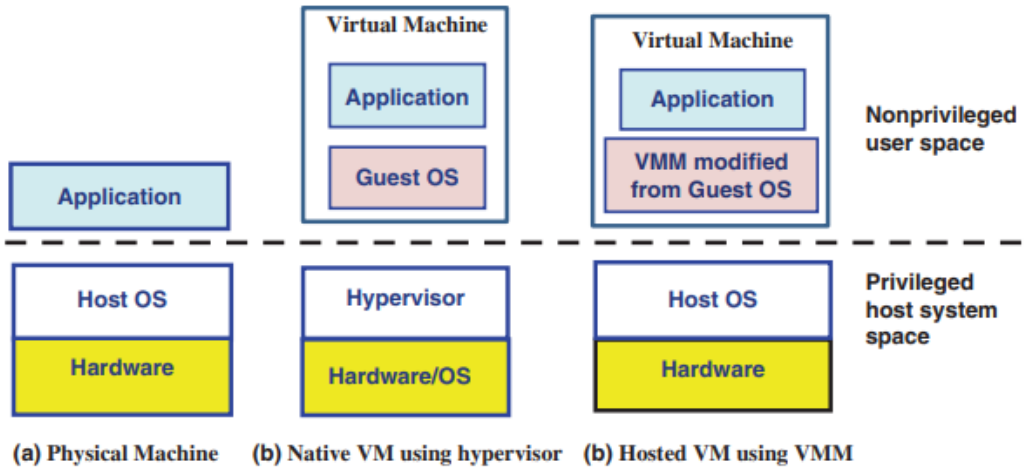
VMWare Player, Workstation, VirtualBox	×-86, ×-8-6–64	OS host apa pun	Windows, Linux, Darwin Solaris, OS/2, BSD Gratis	Hypervisor yang di-host dengan arsitektur para-virtualisasi ditunjukkan pada Gambar 1.20(c)

Hypervisor dan Mesin Virtual

Komputer tradisional disebut mesin fisik. Setiap host fisik berjalan dengan OS-nya sendiri. Sebaliknya, mesin virtual (VM) adalah mesin abstrak yang ditentukan perangkat lunak yang dibuat oleh proses virtualisasi. Dalam komputer fisik yang menjalankan OS, X menjalankan program aplikasi yang hanya dirancang khusus untuk platform X. Dalam program lain yang ditulis untuk OS yang berbeda, Y mungkin tidak dapat dieksekusi pada platform X. Dalam menggunakan VM, OS tamu bisa berbeda dari OS host. Misalnya, platform-X adalah OS Apple dan platform-Y bisa menjadi komputer berbasis Window. VM menawarkan solusi untuk melewati penghalang portabilitas perangkat lunak.

Arsitektur Mesin Virtual

Komputer konvensional memiliki arsitektur sederhana, seperti yang diilustrasikan pada Gambar 2.5(a), di mana OS mengelola semua sumber daya perangkat keras di ruang sistem yang diistimewakan dan semua aplikasi berjalan di ruang pengguna di bawah kendali OS. Pada VM asli, VM terdiri dari kontrol aplikasi pengguna oleh OS tamu. VM ini dibuat oleh hypervisor yang diinstal pada ruang sistem yang diistimewakan. Hypervisor ini berada tepat di atas bare metal, seperti yang ditunjukkan Gambar 2.5(b). Beberapa VM dapat di-porting ke satu komputer fisik. Pendekatan VM memperluas portabilitas perangkat lunak di luar batas platform. Hypervisor bare-metal berjalan langsung pada perangkat keras host untuk mengelola OS tamu.



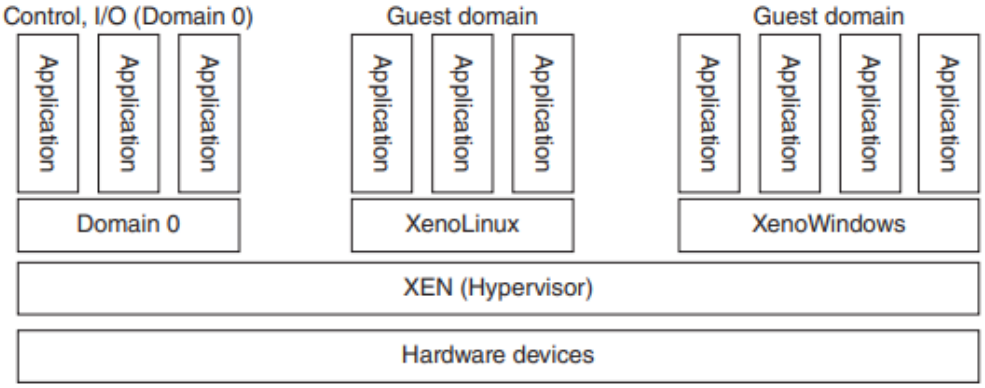
Gambar 2.5 Dua arsitektur VM dibandingkan dengan mesin fisik konvensional.

Arsitektur VM lainnya ditunjukkan pada Gambar 2.5(c), yang dikenal sebagai VM yang dihosting yang dibuat oleh VMM atau hypervisor yang dihosting yang diimplementasikan di atas OS host. VMM adalah middleware antara OS host dan aplikasi pengguna. Ini menggantikan OS tamu yang digunakan dalam VM asli. Jadi VMM mengabstraksi OS tamu dari OS host. VMware Workstations, VM player dan VirtualBox adalah contoh dari VM yang di-host

yang dikenal sebagai para-virtualisasi. Dalam hal ini, OS host dibiarkan tidak berubah dan VMM memantau eksekusi aplikasi pengguna secara langsung. Kecuali ditentukan lain, kami hanya akan mempertimbangkan VM asli yang diproduksi oleh hypervisor bare-metal.

Empat hypervisor atau VMM dirangkum dalam Tabel 2.6. XEN adalah yang paling populer yang digunakan di hampir semua PC, server, atau workstation berbasis ×-86. VM yang dibuat dengan hypervisor seringkali sangat berbobot, karena terdiri dari kode aplikasi pengguna (yang mungkin hanya KB) ditambah OS tamu, yang mungkin memerlukan memori GB. OS tamu mengawasi eksekusi aplikasi pengguna di VM. KVM adalah VM berbasis Kernel Linux. Microsoft Hyper-V digunakan untuk virtualisasi server Windows. Dengan kata lain, KVM banyak digunakan di host Linux, sedangkan Hyper V harus digunakan di host Windows. Hypervisor ini melibatkan integrasi OS pada level terendah. Malware dan rootkit dapat mengirimkan beberapa ancaman terhadap keamanan hypervisor. Peneliti dari Microsoft dan akademisi telah mengembangkan beberapa perangkat lunak anti-rootkit Hooksafe untuk melindungi hypervisor dari serangan malware dan rootkit.

VMware menawarkan tiga hypervisor yang dihosting yang tercantum dalam Tabel 2.6. Hypervisor ini lebih dikenal sebagai VMM yang melakukan para-virtualisasi yang diilustrasikan pada Gambar 2.5(c). VMware memelopori pengembangan perangkat lunak virtualisasi. Mereka mulai dengan versi workstation yang dapat dijalankan dengan host Windows dan Linux, yang benar-benar untuk virtualisasi penuh. Kemudian, VMware meluncurkan paket server ESX untuk penggunaan virtualisasi di server ×.86. Versi ini tidak memerlukan penggunaan OS host untuk memvirtualisasikan sumber daya. Sekarang OS cloud, vSphere, sepenuhnya didukung oleh paket virtualisasi VMware sendiri.



Gambar 2.6 Arsitektur XEN: Domain 0 untuk kontrol sumber daya dan I/O dan beberapa domain tamu (VM) dibuat untuk aplikasi pengguna perumahan.

Secara umum, beberapa paket VMware VMM (pemain atau VirtualBox) tidak bertanggung jawab atas alokasi sumber daya untuk semua program pengguna. Mereka digunakan untuk mengalokasikan hanya sumber daya terbatas untuk aplikasi yang dipilih. VMM mengontrol sumber daya yang secara eksplisit dialokasikan ke aplikasi khusus yang dipilih ini. Dengan kata lain, VMM terikat dengan sumber daya prosesor yang dipilih. Tidak semua prosesor memenuhi persyaratan VMM. Keterbatasan khusus termasuk

ketidakmampuan untuk menjebak beberapa instruksi istimewa. Ini adalah semangat virtualisasi berbantuan perangkat keras.

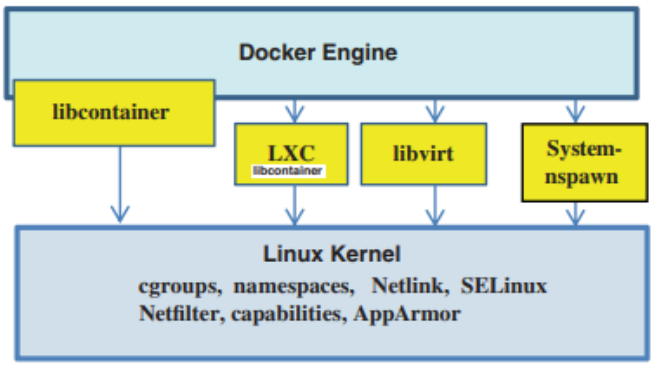
Contoh 2.3 Arsitektur Hypervisor XEN dan Kontrol Sumber Daya

XEN adalah hypervisor mikro-kernel open-source yang dikembangkan di Universitas Cambridge. Hypervisor XEN mengimplementasikan semua mekanisme, membiarkan kebijakan ditangani oleh Domain 0, seperti yang ditunjukkan pada Gambar 2.6. XEN tidak menyertakan driver perangkat apa pun secara asli. Komponen inti dari sistem XEN adalah hypervisor, kernel dan aplikasi. OS tamu, yang memiliki kemampuan kontrol, disebut Domain0 dan yang lainnya disebut DomainU. Domain0 adalah OS tamu istimewa dari XEN. Ini pertama kali dimuat ketika XEN melakukan booting tanpa driver sistem file apa pun.

Domain0 dirancang untuk mengakses perangkat keras secara langsung dan mengelola perangkat. Oleh karena itu, salah satu fungsi Domain0 adalah mengalokasikan dan memetakan sumber daya perangkat keras ke domain tamu (DomainUs). Misalnya, XEN berbasis Linux dan tingkat keamanannya lebih tinggi. VM manajemennya bernama domain 0, yang memiliki hak istimewa untuk mengelola VM lain yang diimplementasikan pada host yang sama. Jika domain 0 dikompromikan, peretas dapat mengontrol seluruh sistem. Kebijakan keamanan khusus diterapkan untuk mengamankan domain 0. Domain 0, berperilaku seperti hypervisor, memungkinkan pengguna untuk membuat, menyalin, menyimpan, membaca, memodifikasi, berbagi, bermigrasi, dan mengembalikan VM semudah memanipulasi file.

Mesin Docker dan Wadah Aplikasi

Docker menyediakan virtualisasi tingkat OS pada mesin host yang menjalankan Linux, Mac OS, dan Windows. Di bagian ini, kami memperkenalkan mesin Docker dan container Docker. Kemudian kami membandingkan perbedaan implementasi dan mendiskusikan kekuatan dan kelemahan relatif antara VM yang dibuat dengan hypervisor dan container Docker. Sebagian besar pusat data dibangun dengan server ×-86 berbiaya rendah dalam skala besar, sehingga mudah untuk melihat minat yang meningkat dari pembuat dan penyedia cloud untuk beralih ke wadah Docker untuk aplikasi pengguna yang skalabel. Namun, VM masih berguna dalam berbagai jenis aplikasi. Mereka mungkin hidup berdampingan untuk waktu yang lama.



Gambar 2.7 Mesin Docker mengakses fitur kernel Linux untuk virtualisasi terisolasi dari wadah aplikasi yang berbeda.

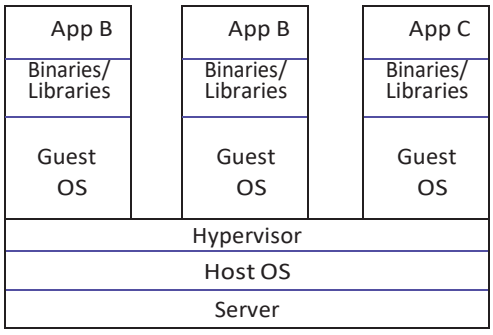
Mesin buruh pelabuhan

Ini adalah perangkat lunak virtualisasi yang berjalan antara OS host dan kode aplikasi pengguna serta binari dan pustakanya. Docker mengimplementasikan API tingkat tinggi untuk menyediakan container ringan yang menjalankan proses perangkat lunak secara terpisah. Konsep virtualisasi Docker diilustrasikan pada Gambar 2.7. Mesin Docker menggunakan fitur isolasi sumber daya dari kernel Linux. Ruang nama cgroups dan kernel memungkinkan container independen berjalan dalam instance Linux yang terpisah. Kontainer terisolasi ini menghindari overhead pembuatan VM.

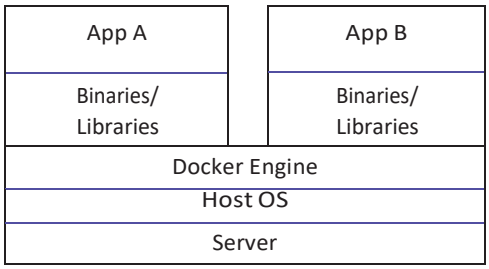
Kontainer Docker

Docker adalah proyek sumber terbuka yang mengotomatiskan pengembangan aplikasi pengguna sebagai wadah perangkat lunak. Mesin Docker menyediakan lapisan tambahan abstraksi dan otomatisasi virtualisasi tingkat OS di platform host berbasis Linux. Mesin Docker ditulis dalam bahasa Go, berjalan di platform Linux. Docker berbeda dari VM tradisional karena terdiri dari aplikasi, ditambah binari dan pustaka yang diperlukan. Setiap wadah aplikasi membutuhkan sekitar 10 detik memori MB.

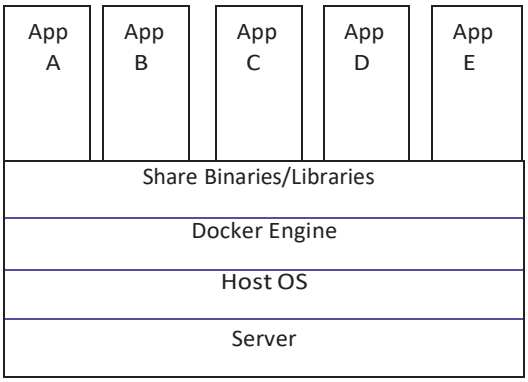
Sebaliknya, VM yang dibuat oleh hypervisor dapat meminta 10 detik GB untuk menghosting OS tamu selain kode aplikasi. Kontainer Docker diisolasi, tetapi dapat berbagi OS dan binari dan pustaka, seperti yang ditunjukkan pada Gambar 2.8(c). Keuntungan yang jelas adalah container ringan versus VM tugas berat, dan masing-masing harus memiliki OS tamu dan bi/pustaka sendiri. Kontras ditunjukkan oleh perbedaan ketinggian antara VM dan kontainer pada Gambar 2.8 (a b). Mungkin lebih murah untuk membangun dan menggunakan container daripada membuat dan menggunakan VM. Karena alasan ini, container Docker menggantikan VM tradisional di cloud utama. Tidak diperlukan OS tamu untuk menjalankan aplikasi pengguna. Wadah menerapkan fungsi kernel. Isolasi sumber daya termasuk CPU, memori dan blok I/O, jaringan, dll. Dilakukan dengan menggunakan ruang nama terpisah untuk aplikasi yang berbeda.



(a) Tiga mesin virtual (VM) yang dibuat oleh hypervisor pada host perangkat keras yang sama, setiap VM banyak dimuat dengan OS tamunya sendiri dan binari serta pustaka tertentu.



(b) Setiap kontainer dimuat dengan binari/pustakanya sendiri yang tidak dibagikan.



(c) Mesin Docker membuat banyak wadah aplikasi ringan, yang terisolasi, tetapi berbagi OS

Gambar 2.8 Mesin Hypervisor versus Docker untuk membuat mesin virtual dan wadah aplikasi, masing-masing.

Docker menerapkan fitur virtualisasi kernel secara langsung menggunakan pustaka libcontainer. Antarmuka ini tersedia sejak Docker 0.9. Mesin Doctor juga dapat mengakses kernel Linux secara tidak langsung, melalui penggunaan antarmuka: LXC (Linux Containers), libvirt atau system-spawn. Docker adalah proyek sumber terbuka. Kode sumber dipercayakan ke situs web GitHub. Sistem ini dalam alat dengan perjanjian Apache 2.0. Mesin Docker menghasilkan container virtual yang ringan di platform Linux. Sistem ini pada dasarnya adalah mesin pembangkit dan manajemen kontainer. Kode sumber Docker kecil agar sesuai dengan kebanyakan komputer. Ini diimplementasikan dengan bahasa Go. Untuk klien, Docker mengasumsikan arsitektur klien/server.

Peluang Penerapan VM/Kontainer

Pada Tabel 2.7, kami telah merangkum properti dari beberapa hypervisor (XEN, KVM, Hyper V, dan VMware). VM yang dibuat dengan hypervisor ini dibandingkan dengan container Docker. Mungkin lebih murah untuk membangun dan menggunakan container daripada membuat dan menggunakan VM. Karena alasan ini, container Docker dapat menggantikan VM tradisional di beberapa cloud. Misalnya, AWS EC2 telah menawarkan layanan ECS (Elastic Container Service).

Hal ini memungkinkan wadah pengguna untuk mengimplementasikan aplikasi mereka dengan permintaan dan kompleksitas memori yang diturunkan secara signifikan. Menggunakan wadah, sumber daya diisolasi, layanan dibatasi, dan proses disediakan untuk memiliki tampilan pribadi lengkap OS di dalam ruang ID proses, sistem file, dan antarmuka jaringan mereka sendiri. Untuk membangun sistem yang sangat terdistribusi, penggunaan wadah aplikasi dapat menyederhanakan masalah pembuatan, keamanan, dan manajemen secara signifikan, dibandingkan dengan penggunaan VM yang dibuat dengan hypervisor.

Tabel 2.7 Perbandingan antara VM yang dibuat hypervisor dan container Docker.

Jenis VM	Kekuatan dan kelemahan	Aplikasi yang Cocok
Mesin Virtual yang dibuat oleh Hypervisor	Fleksibilitas aplikasi yang lebih tinggi dalam meluncurkan aplikasi OS yang berbeda, tetapi menuntut lebih banyak memori dan overhead untuk membuat dan meluncurkan VM	Lebih cocok untuk digunakan di banyak aplikasi tanpa orkestrasi. VM menarik untuk dijalankan di berbagai sistem operasi
Kontainer Docker	Wadah aplikasi ringan dengan overhead rendah untuk dibuat dan dijalankan dengan perlindungan yang lebih baik di bawah lingkungan eksekusi yang terisolasi	Lebih cocok untuk penggunaan skalabel dari aplikasi yang sama dalam banyak salinan di bawah orkestrasi. Bekerja lebih baik dengan versi OS tertentu. Ini dapat menghemat biaya operasi cloud

Misalnya, wadah dapat di-boot dan siap untuk aplikasi dalam 500 ms, sedangkan hypervisor dapat boot dalam 20 detik sesuai dengan OS yang digunakan. Secara umum, kita dapat menyimpulkan bahwa container yang ringan cocok untuk penggunaan skalabel dengan banyak salinan untuk orkestrasi cloud. Ini menyiratkan bahwa wadah mendukung pengelompokan dan kelipatan. Misalnya, container bertindak untuk menjalankan banyak salinan dari satu aplikasi, katakanlah MySQL. Hypervisor seringkali lebih cocok untuk aplikasi tugas berat yang memiliki permintaan orkestrasi cloud yang terbatas. Jika Anda ingin fleksibilitas menjalankan beberapa aplikasi, Anda menggunakan VM.

Daemon berinteraksi dengan tiga driver di mesin Docker. Driver mengontrol pembuatan lingkungan eksekusi container. Graphdriver adalah pengelola gambar wadah, yang berkomunikasi dengan file akar berlapis yang terkait dengan wadah yang dibuat di kotak bawah. Networkdriver menyelesaikan penyebaran kontainer. Execud-river bertanggung jawab untuk mengarahkan eksekusi container dengan bekerja dengan namespace dan cgroup di libcontainer, yang ditulis dalam bahasa Go dan digunakan sebagai basis untuk mengontrol semua container yang dibuat.

Akhirnya, wadah dibuat di kotak bawah. Docker menggunakan Daemon sebagai manager dan libcontainer sebagai algojo dalam menghasilkan container. Kontainer memiliki fungsi yang serupa dengan VM di bawah lingkungan eksekusi yang terisolasi. Dalam proses ini, container Docker dibuat dengan overhead rendah, membutuhkan memori minimum, dan terlindungi dengan baik dengan isolasi kernel.

2.3 ARSITEKTUR CLOUD DAN MANAJEMEN SUMBER DAYA

Pada bagian ini, pertama-tama kita mempelajari arsitektur berorientasi layanan (SoA) untuk membangun cloud publik, pribadi, dan hibrida. Kemudian kami mempelajari masalah manajemen dalam menggunakan VM dan container untuk membangun kluster virtual untuk

digunakan dalam layanan cloud. Untuk tujuan ini, kami menghadirkan tiga arsitektur cloud paling populer, yaitu sistem cloud AWS, OpenStack, dan VMWare.

Arsitektur Cloud Platform

Sebagian besar cloud saat ini mengikuti organisasi SoA. Secara umum, arsitektur cloud dapat digambarkan dengan dua lapisan sumber daya. Lapisan bawah adalah infrastruktur statis, batas sistem dan antarmuka pengguna dengan dunia luar. Lapisan atas dibentuk oleh sumber daya dinamis seperti VM atau wadah di bawah pengelolaan OS cloud atau pusat kendali. Pada Tabel 2.8, kami membandingkan tiga SoA untuk membangun tiga jenis cloud. AWS cloud mewakili cloud publik paling populer. OpenStack adalah digunakan untuk konstruksi cloud pribadi dalam bisnis kecil dan komunitas yang dilindungi. Paket perangkat lunak VMWare komersial adalah untuk membangun cloud hybrid yang digunakan oleh perusahaan dan organisasi bisnis besar.

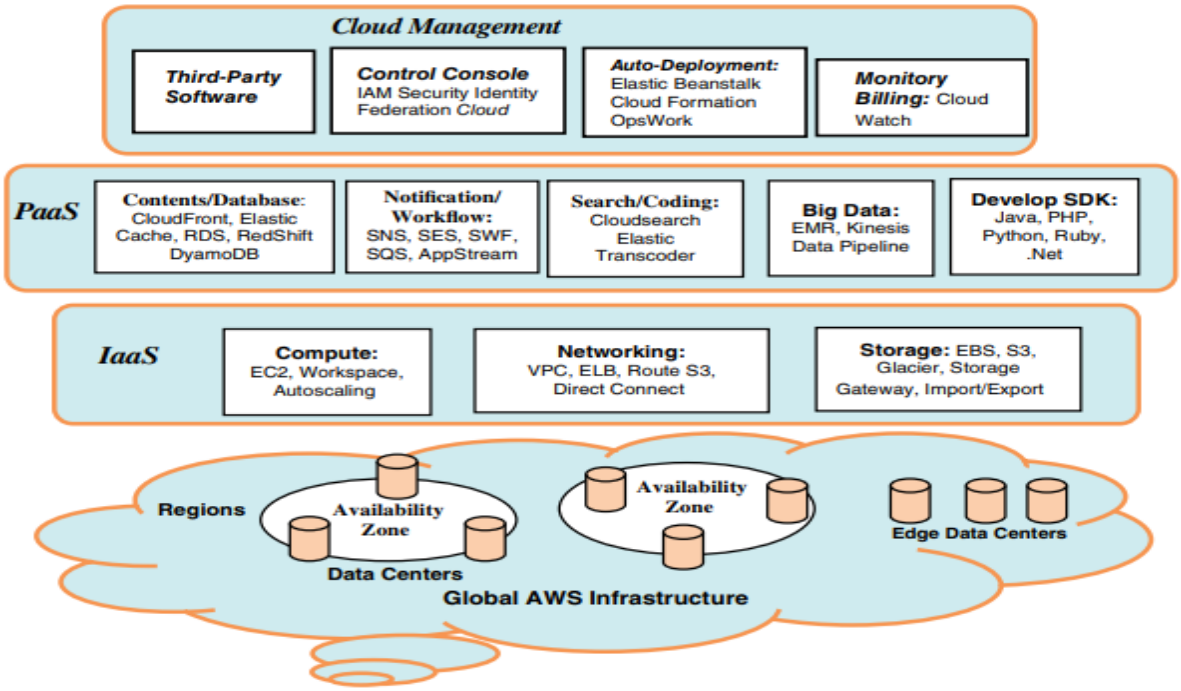
Tabel 2.8 Perbandingan tiga arsitektur platform cloud.

Fitur Sistem Cloud	**Amazon Web Service (AWS): Cloud Publik**	**Sistem OpenStack: Private Cloud**	**Sistem VMWare: Cloud Hibrida**
Model Layanan	IaaS, PaaS	IaaS	IaaS, PaaS
Pengembang/Penyedia dan Desain	Amazon (Bag. 2.3.1 dan 2.4.1–2.4.2)	Rackspace/NASA dan Apache (2.3.4)	VMWare (Bag. 2.3.5) Kepemilikan
Paket dan Skala Arsitektur	Pusat data didistribusikan sebagai Availability Zone di berbagai Wilayah global (Gambar 2.11)	Cloud kecil di situs pemilik, dilisensikan melalui Apache (Gambar 2.14)	Cloud pribadi berinteraksi dengan cloud publik (Gambar 2.17)
Dukungan Cloud OS/Perangkat Lunak	Mendukung instans mesin Linux dan Windows dengan penskalaan dan penagihan otomatis	Sumber terbuka, terbentang dari Eucalyptus dan OpenNebula	vSphere dan vCenter, mendukung server ×-86 dengan NSX dan vSAN
Spektrum Pengguna	Masyarakat Umum: Badan Usaha Bisnis dan Pengguna Perorangan	Pusat Penelitian atau Usaha Kecil	Perusahaan Bisnis dan Organisasi Besar

Arsitektur SoA berbeda dari arsitektur komputer tradisional dalam banyak hal. Komponen-komponen dalam sistem komputer tradisional sangat erat digabungkan. Ini membatasi fleksibilitas aplikasi dan mempersulit pemeliharaan sistem. Konsep SoA dimulai dengan IBM, HP dan Microsoft pada awal tahun 2000. Prinsip SoA bergantung pada sambungan longgar di antara blok layanan suatu sistem. Antarmuka layanan dirancang untuk menghubungkan berbagai modul layanan. Ini akan membebaskan efek mengikat dari sistem, memungkinkan skalabilitas yang lebih tinggi dan pertumbuhan dan pemeliharaan modular. Itulah yang seharusnya dimiliki sistem cloud. CEO Amazon Bezzop telah mendorong ide SoA

ke dalam pengembangan cloud AWS, yang telah terbukti berhasil di antara semua cloud publik.

AWS cloud dibangun dengan infrastruktur global dari banyak pusat data yang berlokasi di berbagai wilayah di dunia. Misalnya, inti AWS, EC2 (*cloud computing* elastis), memiliki sembilan situs regional di seluruh dunia. Di setiap wilayah, mereka mengelompokkan pusat data ke dalam zona ketersediaan (AZ). Setiap AZ dibangun dengan setidaknya tiga pusat data, jarak 50 KM satu sama lain. Pendekatan multi-pusat data sangat meningkatkan kinerja, keandalan, dan toleransi kesalahan dari AWS cloud. Infrastruktur global Amazon mendistribusikan sumber daya, seperti yang ditunjukkan pada Gambar 2.9. Ada banyak pusat data edge yang dapat ditambahkan ke dalam operasi cloud AWS. AWS cloud telah mulai menyediakan layanan IaaS di komputer dan fungsi penyimpanan.



Gambar 2.9 AWS publik cloud yang terdiri dari lapisan manajemen teratas, platform PaaS dan IaaS, dan infrastruktur global yang dibangun di atas pusat data di zona ketersediaan yang terletak di berbagai wilayah secara global.

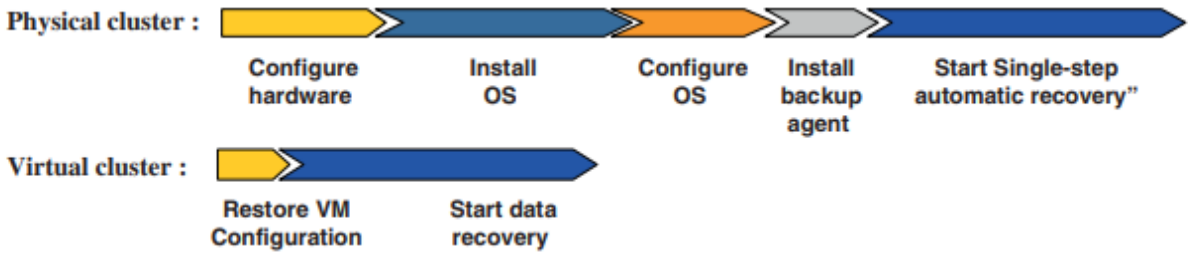
Sekarang layanan juga telah diperluas ke tingkat PaaS. Layanan PaaS dimaksudkan untuk mendukung operasi *Big data*, basis data, dan analitik data. Sejumlah besar modul layanan dibangun ke dalam platform IaaS dan PaaS dari AWS. Rincian layanan ini akan disajikan di Bagian 2.4.2 dalam beberapa kategori. Antarmuka layanan khusus dirancang untuk menyediakan komunikasi di antara modul layanan. Gabungan layanan IaaS+PasS juga didukung oleh Microsoft Azure dan Google cloud. Di bagian atas Gambar 2.9, manajemen cloud mengoordinasikan seluruh operasi cloud dalam hal pemantauan, keamanan, penagihan, dan penggunaan perangkat lunak pihak ketiga. Kami akan mempelajari lebih lanjut komponen AWS cloud di Bagian 2.4.

Manajemen VM dan Pemulihan Bencana

Manajemen infrastruktur cloud melibatkan beberapa masalah. Pertama, kami mempertimbangkan manajemen VM dari pekerjaan layanan independen. Kemudian kami mempertimbangkan cara menjalankan aplikasi cloud pihak ketiga:

- **Manajemen Layanan Independen:** Layanan independen meminta fasilitas untuk menjalankan banyak tugas yang tidak terkait. Umumnya, API yang disediakan adalah layanan web yang dapat digunakan pengembang dengan nyaman. Di AWS EC2, SQS (Simple Queue Service) dibuat untuk menyediakan layanan komunikasi yang andal antara penyedia yang berbeda. Bahkan titik akhir tidak berjalan saat entitas lain telah memposting pesan di SQS. Dengan menggunakan penyedia layanan independen, aplikasi cloud dapat menjalankan layanan yang berbeda secara bersamaan.
- **Menjalankan Aplikasi Pihak Ketiga:** Platform cloud sering digunakan untuk menjalankan aplikasi pihak ketiga. Karena aplikasi web saat ini sering disediakan dengan menggunakan formulir Web 2.0, antarmuka pemrograman berbeda dari yang digunakan di perpustakaan runtime. API bertindak sebagai layanan. Mesin aplikasi layanan web digunakan oleh pemrogram untuk membangun aplikasi pihak ketiga. Browser web adalah antarmuka pengguna untuk pengguna akhir.
- **Virtualisasi Perangkat Keras:** Dalam sistem cloud, hypervisor sering digunakan untuk memvirtualisasikan sumber daya perangkat keras untuk membuat VM. Virtualisasi tingkat sistem menuntut jenis perangkat lunak khusus yang mensimulasikan eksekusi perangkat keras dan bahkan menjalankan sistem operasi yang tidak dimodifikasi. Server, penyimpanan, dan jaringan tervirtualisasi disatukan untuk menghasilkan platform computing cloud. Lingkungan pengembangan dan penyebaran cloud harus konsisten untuk menghilangkan masalah runtime. VM yang diinstal pada platform computing cloud terutama digunakan untuk menghosting aplikasi pihak ketiga. Mesin virtual menyediakan layanan runtime yang fleksibel untuk membebaskan pengguna dari kekhawatiran tentang lingkungan sistem.

Dengan menggunakan VM, fleksibilitas aplikasi yang tinggi seringkali menjadi keunggulan utama dibandingkan sistem komputer tradisional. Karena sumber daya VM digunakan bersama oleh banyak pengguna, kami memerlukan metode untuk memaksimalkan hak istimewa pengguna dan menjaga VM yang disediakan dalam lingkungan eksekusi yang terisolasi. Pembagian tradisional sumber daya cluster sering diatur secara statis sebelum waktu berjalan, tetapi pembagian seperti itu tidak fleksibel. Pengguna tidak dapat menyesuaikan sistem untuk aplikasi interaktif dan OS sering menjadi penghalang pada portabilitas perangkat lunak. Virtualisasi memungkinkan pengguna untuk memiliki hak istimewa penuh sambil menjaga mereka sepenuhnya terpisah. Dalam hal ini, wadah Docker lebih baik diisolasi daripada menggunakan VM yang dibuat oleh hypervisor. Virtualisasi dapat menguntungkan sistem cloud dengan mencapai ketersediaan tinggi, pemulihan bencana, perataan beban dinamis, penyediaan sumber daya yang fleksibel, dan lingkungan computing yang dapat diskalakan.



Gambar 2.10 Overhead pemulihan pada cluster fisik dibandingkan dengan cluster virtual.

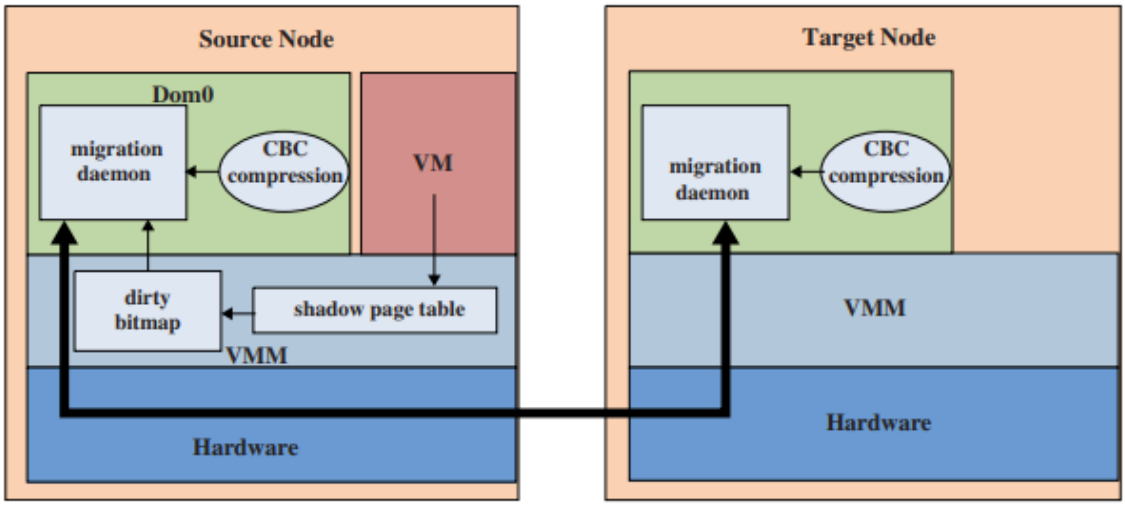
Kloning VM untuk Pemulihan Bencana

Teknologi mesin virtual (VM) memerlukan skema pemulihan bencana tingkat lanjut. Salah satu skema adalah untuk memulihkan mesin fisik (PM) oleh PM lain. Skema kedua adalah memulihkan VM oleh VM lain. Seperti yang ditunjukkan pada garis waktu atas Gambar 2.10, pemulihan bencana tradisional dari PM ke PM lambat, kompleks dan mahal. Total waktu pemulihan dikaitkan dengan konfigurasi perangkat keras, pemasangan dan konfigurasi OS, pemasangan agen pencadangan, dan waktu lama yang diperlukan untuk memulai ulang PM. Untuk memulihkan platform VM, waktu penginstalan dan konfigurasi untuk OS dan agen pencadangan dihilangkan. Oleh karena itu, kami berakhir dengan waktu pemulihan bencana yang jauh lebih singkat, sekitar 40% dari itu untuk memulihkan PM.

Kloning VM menawarkan solusi yang efektif. Idenya adalah membuat VM kloning di server jauh untuk setiap VM yang berjalan di server lokal. Di antara semua VM kloning, hanya satu yang perlu aktif. VM jarak jauh harus dalam mode ditangguhkan. Pusat kendali cloud harus dapat mengaktifkan VM kloning ini jika terjadi kegagalan VM asli, mengambil snapshot VM untuk mengaktifkan migrasi langsung dengan waktu minimum. VM yang dimigrasikan berjalan pada koneksi Internet bersama. Hanya data yang diperbarui dan status yang dimodifikasi yang dikirim ke VM yang ditangguhkan untuk memperbarui statusnya. Keamanan VM harus ditegakkan selama migrasi langsung VM.

Langkah-Langkah Migrasi VM Langsung

Dalam sebuah cluster yang dibangun dengan node campuran dari sistem host dan guest, cara operasi yang normal adalah dengan menjalankan semuanya pada mesin fisik. Ketika VM gagal, perannya dapat digantikan oleh VM lain pada node yang berbeda, selama keduanya berjalan dengan OS tamu yang sama. Dengan kata lain, node fisik dapat gagal ke VM di host lain. Ini berbeda dari failover fisik-ke-fisik dalam cluster fisik tradisional. Keuntungannya adalah meningkatkan fleksibilitas failover. Kelemahan potensial adalah bahwa VM harus berhenti memainkan perannya pada node host yang gagal. Namun, masalah ini dapat diatasi dengan migrasi VM langsung. Gambar 2.11 menunjukkan proses migrasi kehidupan VM dari host A ke host B. Migrasi dilakukan dengan menyalin file status VM dari area penyimpanan ke mesin host.



Gambar 2.11 Migrasi langsung VM dari domain Dom0 ke host target berkemampuan XEN.

Contoh 2.4 Migrasi Langsung VM antara Dua Host Berkemampuan Xen

Xen mendukung migrasi langsung. Ini adalah fitur yang berguna dan ekstensi alami untuk platform virtualisasi yang memungkinkan transfer VM dari satu mesin fisik ke mesin fisik lainnya, dengan sedikit waktu henti layanan yang dihosting oleh VM. Migrasi langsung mentransfer status kerja dan memori VM di seluruh jaringan saat sedang berjalan. Xen juga mendukung migrasi VM dengan menggunakan mekanisme: Remote Direct Memory Access (RDMA).

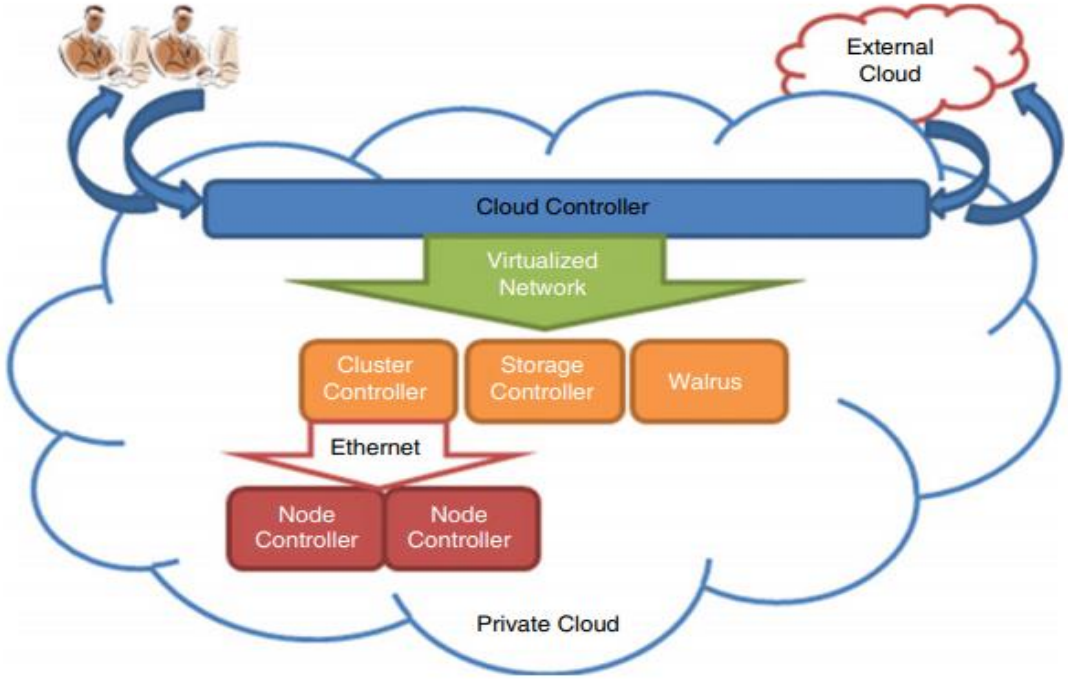
Ini mempercepat migrasi VM dengan menghindari overhead pemrosesan tumpukan TCP/IP. RDMA mengimplementasikan protokol transfer yang berbeda di mana buffer VM asal dan tujuan harus didaftarkan sebelum operasi transfer apa pun menguranginya menjadi antarmuka "satu sisi". Komunikasi data melalui RDMA tidak perlu melibatkan CPU, cache, atau sakelar konteks. Ini memungkinkan migrasi dengan dampak minimal pada sistem operasi tamu dan aplikasi yang dihosting. Gambar 2.11 menunjukkan desain algoritma kompresi untuk migrasi VM.

Daemon migrasi bertanggung jawab untuk melakukan migrasi. Tabel halaman bayangan di lapisan VMM melacak modifikasi ke halaman memori di mesin virtual yang dimigrasikan selama fase pra-salin. Bendera yang sesuai diatur dalam bitmap kotor. Pada awal setiap putaran pra-penyalinan, bitmap dikirim ke daemon migrasi. Kemudian, bitmap dihapus dan tabel halaman bayangan dihancurkan dan dibuat ulang di babak berikutnya. Sistem berada di VM manajemen Xen. Halaman memori yang dilambangkan dengan bitmap diekstraksi dan dikompresi sebelum dikirim ke tujuan. Data terkompresi kemudian dikompresi pada target.

OpenStack untuk Membangun Private Cloud

Untuk mengubah cluster server atau pusat data menjadi cloud pribadi, Eucalyptus tentu saja merupakan pionir dari University of Santa Barbara. Ini adalah perangkat lunak open-source untuk membangun cloud pada cluster server skala besar. Versi stabil dirilis pada tahun 2010 dan tersedia untuk masyarakat umum. OpenStack diperluas dari Eucalyptus dengan lebih

banyak dukungan perangkat lunak. Mari kita periksa fungsi Eucalyptus terlebih dahulu. Kemudian kami menyajikan kemajuan dengan OpenStack.



Gambar 2.12 Eucalyptus untuk membangun private cloud dengan membangun jaringan virtual melalui VM yang terhubung melalui Ethernet dan Internet.

Contoh 2.5 Eucalyptus untuk Jaringan Virtual Private Cloud

Ini adalah sistem perangkat lunak sumber terbuka (Gambar 2.12) untuk mendukung cloud IaaS. Sistem ini terutama mendukung jaringan virtual dan manajemen mesin virtual. Penyimpanan virtual tidak didukung. Ini telah digunakan secara luas dalam membangun cloud pribadi yang dapat berinteraksi dengan pengguna akhir melalui Ethernet atau Internet. Sistem ini juga mendukung interaksi dengan private cloud atau public cloud lainnya melalui Internet. Sistem ini kekurangan keamanan dan fitur untuk aplikasi grid atau cloud tujuan umum.

Perancang mengklaim bahwa "Eucalyptus" adalah singkatan dari "Arsitektur Computing Utilitas Elastis untuk Menghubungkan Program Anda ke Sistem yang Berguna". Dalam hal fungsionalitas, Eucalyptus berfungsi seperti AWS API, sehingga dapat berinteraksi dengan EC2. Ini menyediakan API penyimpanan untuk meniru API Amazon S3 untuk menyimpan data pengguna dan gambar VM. Eucalyptus diinstal pada platform berbasis Linux. Ini kompatibel dengan EC2 dan S3 dalam layanan SOAP, REST, dan Query. Layanan CLI dan portal Web dapat diterapkan dengan Eucalyptus.

Tabel 2.9 mencantumkan sejumlah paket perangkat lunak sumber terbuka untuk membangun sebagian besar platform cloud IaaS. eCloud Foundry dan ApplScale juga mendukung cloud PaaS. Sebagian besar paket dapat membuat VM melalui host Linux. Hampir semua paket kompatibel dengan layanan EC2 dan S3 yang ditawarkan oleh AWS. Mereka semua menggunakan Xen dan KVM. Hypervisor VMware digunakan di Eucalyptus, Cloud Foundry, ApplScale, dan vSphere/4. Di antaranya, kami memilih untuk meninjau OpenStack untuk menilai kemampuan perangkat lunak konstruksi cloud.

Open Stack diperkenalkan oleh Rackspace dan NASA pada Juli 2010. Tujuan utamanya adalah membuat perpustakaan perangkat lunak cloud yang skalabel dan aman untuk membangun cloud. Sejauh ini, saat ini, 200+ perusahaan telah bergabung dengan Proyek OpenStack. Proyek ini menawarkan perangkat lunak sumber terbuka gratis di bawah lisensi Apache. Perangkat lunak cloud OpenStack ditulis dengan Python. Sistem diperbarui setiap enam bulan.

Tabel 2.9 Sistem perangkat lunak manajemen cluster open source.

Perangkat lunak	**Jenis Cloud, Lisensi**	**Bahasa yang digunakan**	**Linux/ Windows**	**Kompatibilitas EC2/S3**	**XEN/KVM/VMWare**
kayu putih	iaaS, Rackspace	Jawa, C	Ya/ya	Ya/ya	Ya/ya/ya
Nimbus	laaS, Apache	Jawa, Python	Tidak diketahui	Ya/Tidak	Ya/Ya/Tidak Diketahui
pengecoran cloud	PaaS, Apache	Ruby, C	Ya/Tidak	Ya/Tidak	Ya/ya/ya
OpenStack	laaS, Apache	Python	Ya/Tidak diketahui	Ya/ya	Ya/Ya/Tidak Diketahui
BukaNebua	laaS, Apache	C, C++, Ruby, Java, lex, yacc, Shellscript	Ya/Tidak diketahui	Ya/Tidak diketahui	Ya/Ya/Tidak Diketahui
skala aplikasi	Paas, BSD	Python, Ruby, Go	Tidak dikenal	Ya/ya	Ya/ya/ya

OpenStack Compute (Nova)

Ini adalah modul OpenStack Compute. Nova adalah pengontrol untuk menyiapkan struktur internal cloud IaaS apa pun dengan membuat dan mengelola kluster besar server virtual. Sistem ini berlaku dengan konfigurasi KVM, VMware, Xen, Hyper-v, Linux container LXC dan bare-metal HPC. Arsitektur untuk Nova didasarkan pada konsep shared-nothing dan pertukaran informasi berbasis pesan. Oleh karena itu sebagian besar komunikasi di Nova difasilitasi oleh antrian pesan. Untuk mencegah pemblokiran, seperti beberapa komponen menunggu respons dari yang lain, objek yang ditangguhkan diperkenalkan untuk mengaktifkan panggilan balik saat respons diterima. AMQP menawarkan protokol antrian pesan yang canggih. Pengontrol cloud menerapkan protokol http dan AMQP untuk berinteraksi dengan node Nova lainnya atau AWS S3.

Nova diimplementasikan dengan Python sambil memanfaatkan sejumlah perpustakaan dan komponen yang didukung secara eksternal. Ini termasuk Boto, API Amazon yang disediakan dengan Python, dan Tornado, server HTTP cepat yang digunakan untuk mengimplementasikan kemampuan S3 di OpenStack. Server API menerima permintaan http dari Boto, mengonversi perintah ke dan dari format API, sambil meneruskan permintaan ke

pengontrol cloud. Pengontrol cloud mempertahankan status global sistem, memastikan otorisasi saat berinteraksi dengan Manajer Pengguna melalui LDAP. Sistem Nova berinteraksi dengan layanan S3 dan mengelola node dan pekerja penyimpanan yang berpartisipasi. Selain itu, Nova mengintegrasikan komponen jaringan untuk mengelola jaringan pribadi, pengalamatan IP publik, konektivitas VPN, dan aturan firewall.

Penyimpanan OpenStack (Swift)

Ini adalah sistem penyimpanan redundan yang dapat diskalakan pada banyak disk yang tersebar di server pusat data yang besar. Solusi Swift adalah membangun sejumlah komponen yang berinteraksi, termasuk server proxy, cincin, server objek, server wadah, server akun, replikasi, pembaru, dan auditor. Server proxy memungkinkan pencarian lokasi akun, wadah, atau objek di cincin penyimpanan Swift dan merutekan permintaan. Jadi objek apa pun dialirkan ke atau dari server objek melalui server proxy.

Sebuah cincin mewakili pemetaan antara nama-nama entitas yang disimpan di disk dan lokasi fisiknya. Cincin terpisah dibuat untuk akun, wadah, dan objek yang berbeda. Objek disimpan sebagai file biner dengan metadata yang disimpan dalam atribut file yang diperluas. Ini memerlukan pilihan sistem file yang mendasari untuk dukungan server objek, yang seringkali tidak berlaku untuk instalasi Linux standar. Untuk membuat daftar objek, server kontainer digunakan. Daftar kontainer ditangani oleh server akun. Redundansi (dengan demikian toleransi kesalahan) dicapai melalui replikasi data melalui disk yang didistribusikan.

Modul Fungsional OpenStack Lainnya

Block Storage (Cinder) menyediakan perangkat penyimpanan tingkat blok persisten untuk digunakan dengan instance computing OpenStack yang dikelola oleh Dasbor. Networking (Neutron) menawarkan sistem untuk mengelola jaringan dan alamat IP dalam penerapan cloud, dan memberi pengguna kemampuan layanan mandiri melalui konfigurasi jaringan. Dashboard (Horizon) menyediakan antarmuka grafis bagi administrator dan pengguna untuk mengakses, menyediakan, dan mengotomatisasi sumber daya berbasis cloud. Layanan Identitas (Keystone) menyediakan direktori pusat pengguna yang dipetakan ke layanan OpenStack. Ini bertindak sebagai sistem otentikasi umum di seluruh OS cloud dan dapat berintegrasi dengan direktori backend yang ada seperti LDAP.

Penjadwalan dan Orkestrasi Kontainer

Pengguna Docker ingin menskalakan sejumlah besar container di banyak host. Clustered host menghadirkan beberapa tantangan manajemen. Ini menuntut penggunaan penjadwal Docker dan alat orkestrasi. Pertama, kami mengidentifikasi tantangan dan kemudian memeriksa OpenStack Magnum, salah satu alat container yang dapat membantu mengelola container Docker untuk menghasilkan kinerja yang terukur. Orkestrasi adalah konsep luas yang melibatkan penjadwalan kontainer, manajemen cluster, dan bahkan penyediaan host tambahan.

Penjadwalan Kontainer

Kontainer Docker perlu dimuat ke host untuk memenuhi permintaan layanan. Penjadwalan adalah kemampuan administrator Docker untuk memuat file layanan ke dalam host yang menetapkan cara menjalankan wadah tertentu. Manajemen cluster diperlukan untuk mengontrol sekelompok host, yang mencakup penambahan atau penghapusan host

dari sebuah cluster. Manajer cluster pertama-tama harus mendapatkan informasi pemuatan tentang status host saat ini dan wadah yang dimuatnya. Penjadwal container harus memiliki akses ke setiap host di cluster. Pemilihan host merupakan masalah besar bagi penjadwal container dan informasi host. Proses seleksi ini harus seotomatis mungkin. Fungsi container dan beban kerja host harus disesuaikan dengan load balancing dalam sebuah cluster.

Alat Orkestrasi Kontainer

Perangkat lunak manajemen cluster seperti OpenStack dimaksudkan untuk mendukung penjadwalan kontainer. Penjadwalan lanjutan menuntut pengelompokan dan pengoptimalan container. Administrator harus mengelola wadah grup sebagai aplikasi tunggal. Pengelompokan kontainer mungkin menuntut sinkronisasi waktu mulai dan berhenti. Masalah lainnya adalah penyediaan host, yang mengacu pada penyatuan host baru secara tepat waktu dan lancar ke cluster yang ada. Enam alat populer untuk penjadwalan kontainer dan manajemen cluster diringkas dalam Tabel 2.10. Swarm dan compose dikembangkan oleh tim Docker. Kubernetes dikembangkan oleh Google untuk memberi label, mengelompokkan, dan menyetel grup penampung.

Tabel 2.10 Penyediaan host dan alat penjadwalan kontainer.

Nama Alat	**Deskripsi Singkat Fungsi Alat**
armada kapal	Penjadwalan dan komponen manajemen klaster dari CoreOS
maraton	Penjadwalan dan komponen manajemen layanan dari instalasi Mesosphere
Kcloudan	Penjadwal kuat Docker untuk menjalankan container pada host yang disediakan
meso	Apache mesos mengabstraksi dan mengelola sumber daya semua host dalam sebuah cluster
Kubernetes	Penjadwal Google atas container yang berjalan di infrastruktur cloud Anda
menyusun	Alat Docker yang memungkinkan manajemen grup wadah, secara deklaratif

Orkestrasi OpenStack (Magnum)

Ini adalah layanan OpenStack API yang dikembangkan oleh tim kontainer OpenStack. Tujuannya adalah untuk membuat mesin orkestrasi kontainer seperti Docker dan Kubernetes tersedia sebagai sumber daya kelas satu di OpenStack. Magnum menerapkan Docker Heat untuk mengatur image OS yang berisi Docker dan Kubernetes. Magnum menjalankan image baik dalam VM atau bare metal dalam konfigurasi cluster. Detail lebih lanjut dapat ditemukan di https://github.com/stackforge/magnum/release/tag/2015.1.0b2 OpenStack container ditinjau di http://eavesdrop.openstack.org/irclogs/%23openstack-containers/

Magnum dirancang untuk digunakan oleh operator cloud OpenSrack. Tujuannya adalah untuk menawarkan solusi layanan mandiri untuk menyediakan wadah bagi pengguna cloud sebagai layanan host yang dikelola. Magnum seharusnya membuat wadah aplikasi untuk dijalankan dengan instans Nova, Volume Cinder, dan Basis Data Trove yang ada. Inovasi

utama adalah kemampuan untuk menskalakan aplikasi ke sejumlah instans, menyebabkan aplikasi memunculkan kembali sebuah instans jika terjadi kegagalan, secara otomatis, dan mengemas aplikasi bersama-sama secara lebih efektif daripada menggunakan VM tugas berat. Rincian lebih lanjut dari operasi Magnum dapat ditemukan di https://wiki.openstack.org/wiki/Magnum

Beberapa instance Nova digunakan. Docker Heat, Kubernetes/Swarm, OpenStack Heat dan Micro OS (Fedora Atomic, Core OS) digunakan sebagai komponen. Docker Heat tidak menyediakan penjadwal sumber daya, tetapi khusus untuk Docker yang menggunakan Glance untuk menyimpan gambar container. Gambar berlapis didukung oleh Heat. Komponen utama dalam node pengontrol Magnum adalah Magnum API dan Konduktor dan OpenStack Heat, yang mengontrol Unit Cloud, Kunernetes/Swarm, dan Docker di instans Nova agar berfungsi secara terkoordinasi.

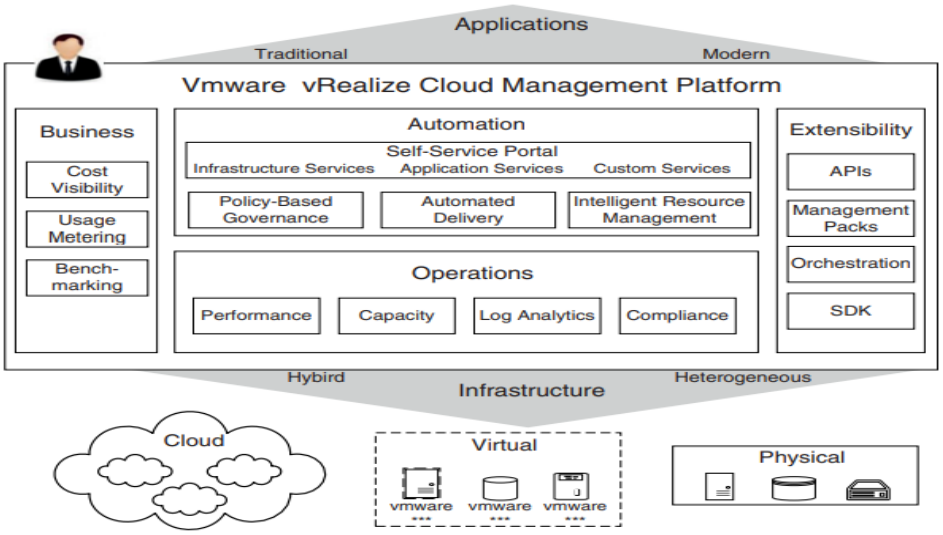
Paket VMWare untuk Membangun Hybrid Clouds

VMWare adalah perusahaan pertama yang mendukung virtualisasi server ×-86, dengan produk VMWare terutama digunakan untuk mendukung lebih dari 80% pangsa pasar cloud enterprise atau cloud hybrid. Produk cloud OS mereka muncul sebagai kernel vSphere dan antarmuka vCenter. Gambar 2.13 menunjukkan platform manajemen VMWare vRealize untuk mendukung konstruksi hybrid.

Lingkungan virtual yang didukung termasuk vSphere untuk tujuan computing, NSX untuk SDN (nama domain server) dan vSAN untuk aplikasi penyimpanan terdistribusi. Lingkungan virtual ini dikelola dalam empat subsistem: bisnis, otomatisasi, operasi, dan perluasan cloud hybrid. Sejumlah besar modul layanan dibangun ke dalam subsistem ini. Tujuan utamanya adalah untuk membangun private cloud berbasis vSphere atau vCenter yang dapat bekerja bersama-sama dengan public cloud eksternal seperti yang akan dilakukan oleh kebanyakan hybrid cloud. Berikut ini, kami menyajikan fungsi vSphere/4 sebagai OS berpemilik yang dirilis oleh VMware. OS ini digunakan untuk membuat VM dan menggabungkannya ke dalam kluster virtual sebagai sumber daya elastis. vSphere/4 menggunakan hypervisors ESX dan ESXi dari VMware. Selanjutnya, vSphere/6 mendukung penyimpanan virtual selain jaringan virtual dan melindungi data. VSphere berpemilik dibandingkan dengan Eucalyptus open-source pada Tabel 2.11. Eucalyptus mendukung virtualisasi XEN dan KVM dan terutama jaringan virtual VM atau wadah.

Contoh 2.6 VMware vSphere 6: A Cloud OS Komersial untuk Hybrid Clouds

vSphere 4 adalah ekosistem perangkat keras dan perangkat lunak yang dikembangkan oleh VMware, dirilis pada April 2009. vSphere dikembangkan dari produk perangkat lunak virtualisasi sebelumnya oleh VMWare, yaitu virtualisasi workstation, ESX untuk virtualisasi server, dan infrastruktur virtual untuk cluster server. Sistem berinteraksi dengan aplikasi pengguna melalui lapisan antarmuka, yang disebut vCenter yang dikelola oleh VMware. Penggunaan utama vSphere adalah untuk menawarkan dukungan virtualisasi dan pengelolaan sumber daya pusat data dalam membangun cloud perusahaan. VMware mengklaim bahwa sistem tersebut adalah OS cloud pertama yang mendukung ketersediaan, keamanan, dan skalabilitas dalam layanan cloud untuk keperluan umum.



Gambar 2.13 Platform cloud VMware yang dibuat dengan vSphere, NSX, dan vSAN, berfungsi sebagai cloud hybrid dengan AWS.

Tabel 2.11 Eucalyptus dan vSphere/6 untuk manajemen sumber daya cloud.

Platform OS	**Sumber daya sedang divirtualisasikan, tautan web**	**API Klien**	**Hypervisor**	**Antarmuka Cloud**	**Fitur spesial**
Kayu Putih, Linux, BSD	Jaringan virtual, (Contoh 2.7) http://www. kayu putih.com/	EC2 WS, CLI	XEN, KVM	EC2	Pengelompokan virtual dengan kontrol hierarkis
vSphere/6, Linux, Windows, Hak Milik	Virtualisasi OS untuk pusat data (Contoh 2.8), http://www. vmware.com/ produk/ vsphere/ produk/vsphere/	CLI, GUI, Portal, WS	VMware ESX, ESXi	Mitra VMware vCloud	Perlindungan data, vStorage, VMFS, DRM, Ketersediaan tinggi

vSphere 4 dibangun dengan dua rangkaian perangkat lunak fungsional: layanan infrastruktur tepat di atas perangkat keras dan layanan aplikasi terhadap aplikasi pengguna. Layanan infrastruktur ditampilkan di bagian bawah. Suite ini memiliki tiga paket komponen, terutama untuk tujuan virtualisasi. vCompute didukung oleh perpustakaan virtualisasi ESX, ESX1 dan DRS dari VMWare. vStorage didukung oleh VMS dan library thin provisioning. vNetwork menawarkan fungsi switching dan jaringan terdistribusi. Paket-paket ini berinteraksi dengan server perangkat keras, disk dan jaringan di pusat data. Fungsi infrastruktur ini juga berkomunikasi dengan cloud eksternal lainnya.

Layanan aplikasi juga dibagi menjadi tiga kelompok: ketersediaan, keamanan, dan skalabilitas. Dukungan ketersediaan termasuk VMotion, Storage VMotion, HA (ketersediaan tinggi), Fault Tolerance dan Data Recovery dari VMWare.

Paket keamanan mendukung vShield Zones dan VMSafe. Paket skalabilitas telah dibangun dengan DRS dan Hot Add. Pembaca yang tertarik dirujuk ke situs web vSphere 4 untuk detail lebih lanjut tentang fungsi perangkat lunak komponen ini. Untuk memahami sepenuhnya penggunaan VSphere 4, pengguna juga harus mempelajari cara menggunakan antarmuka vCenter agar dapat terhubung dengan aplikasi yang ada atau untuk mengembangkan aplikasi baru.

2.4 STUDI KASUS DARI IAAS, PAAS DAN SAAS CLOUDS

Computing cloud memberikan infrastruktur, platform, dan perangkat lunak (aplikasi) sebagai layanan, yang tersedia sebagai layanan berbasis langganan dalam model bayar sesuai pemakaian kepada konsumen. Model IaaS, PaaS dan SaaS membentuk tiga pilar di atas solusi *Cloud computing* yang dikirimkan ke pengguna akhir. Ketiga model tersebut memungkinkan pengguna untuk mengakses layanan melalui Internet, bergantung sepenuhnya pada infrastruktur penyedia layanan cloud.

Cloud IaaS memungkinkan pengguna untuk menggunakan sumber daya TI virtual untuk computing, penyimpanan, dan jaringan. Singkatnya, layanan ini dilakukan oleh infrastruktur cloud yang disewa. Pengguna dapat menyebarkan dan menjalankan aplikasinya sendiri di lingkungan OS yang dipilihnya. Pengguna tidak mengelola atau mengontrol infrastruktur cloud yang mendasarinya tetapi memiliki kontrol atas OS, penyimpanan, aplikasi yang diterapkan, dan mungkin memilih komponen jaringan. Model IaaS ini mencakup penyimpanan sebagai layanan, computing instance sebagai layanan, dan komunikasi sebagai layanan. Beberapa penyedia IaaS yang representatif tercantum dalam Tabel 2.12. Rincian lebih lanjut tentang layanan Amazon EC2 dan S3 diberikan dalam Contoh 2.7 dan 2.8.

Model ini ditawarkan berdasarkan berbagai perjanjian tingkat layanan (SLA) antara penyedia dan pengguna. Dalam arti luas, SLA untuk computing cloud ditujukan dalam hal kinerja ketersediaan layanan dan perlindungan data serta aspek keamanan. SaaS diterapkan pada aplikasi akhir menggunakan antarmuka khusus oleh pengguna atau klien. Pada lapisan PaaS, platform cloud harus melakukan layanan penagihan dan penanganan antrian pekerjaan, meluncurkan dan memantau layanan. Di lapisan bawah layanan IaaS, database, instans computing, sistem file, dan penyimpanan harus disediakan untuk memenuhi permintaan pengguna.

Tabel 2.12 Cloud IaaS dan infrastruktur serta penawaran layanannya (Agustus 2015).

Nama Cloud	Konfigurasi Instans Mesin Virtual	API dan Alat Akses
Amazon EC2	Setiap instans memiliki 1-20 prosesor EC2,	Portal CLI atau Layanan Web (WS)
GoGrid	Memori 1,7–15 GB dan penyimpanan 160 TB	REST, Java, PHP, Python, Ruby

Rackspace Cloud	Setiap instans memiliki 1–6 CPU, memori 0,5–8 GB, dan penyimpanan 30–480 GB	REST, Python, PHP, Java, C#, .NET
Skala Fleksibel di Inggris	Setiap instans memiliki CPU 4-inti, 0.25–16	Konsol web

Arsitektur AWS melalui Pusat Data Terdistribusi

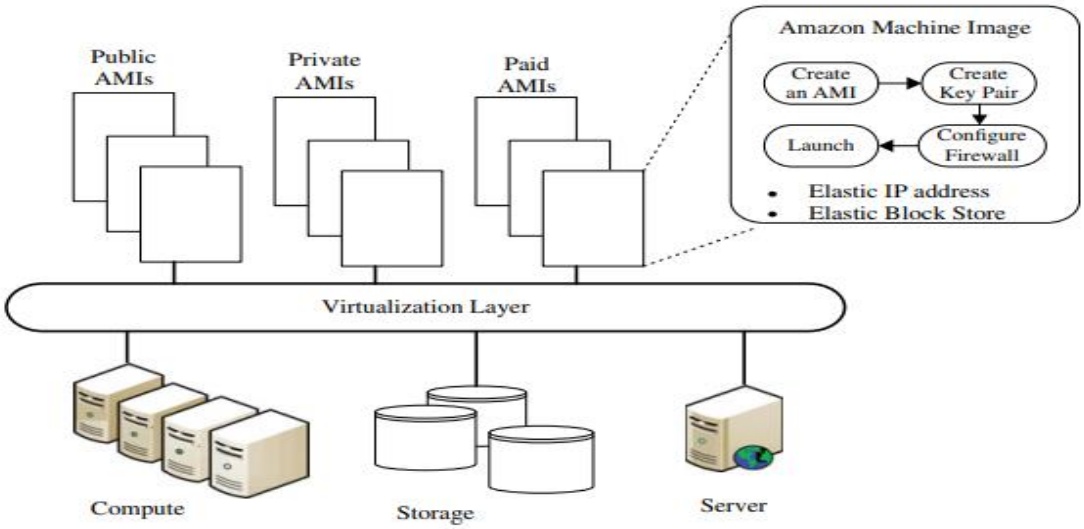
Arsitektur cloud AWS ditunjukkan pada Gambar 2.9. AWS memberikan fleksibilitas ekstrim (mesin virtual) bagi pengguna untuk menjalankan aplikasi mereka sendiri. Penyeimbangan beban elastis secara otomatis mendistribusikan lalu lintas aplikasi yang masuk ke beberapa instans Amazon EC2 dan memungkinkan pengguna untuk menghindari node yang tidak beroperasi dan untuk menyamakan beban pada gambar yang berfungsi. Penskalaan Otomatis dan Penyeimbangan Beban Elastis diaktifkan oleh CloudWatch, yang memantau instans yang sedang berjalan. CloudWatch adalah layanan web yang menyediakan pemantauan untuk sumber daya cloud AWS, dimulai dengan Amazon EC2. Ini memberi pelanggan visibilitas ke dalam pemanfaatan sumber daya, kinerja operasional dan pola permintaan keseluruhan - termasuk metrik seperti pemanfaatan CPU, membaca dan menulis disk, dan lalu lintas jaringan.

Amazon menawarkan RDS Layanan Database Relasional dengan antarmuka perpesanan. Kemampuan Elastic MapReduce setara dengan Hadoop yang berjalan pada penawaran EC2 dasar. Impor/Ekspor AWS memungkinkan kami mengirimkan data dalam jumlah besar ke dan dari EC2 dengan mengirimkan disk fisik; diketahui bahwa ini sering kali merupakan koneksi bandwidth tertinggi antara sistem yang jauh secara geografis. CloudFront mengimplementasikan jaringan distribusi konten. Amazon DevPay adalah layanan manajemen akun dan penagihan online yang mudah digunakan.

Layanan Pembayaran Fleksibel (FPS) memberi pengembang sistem komersial di AWS cara mudah untuk menagih pelanggan Amazon yang menggunakan layanan semacam itu yang dibangun di AWS. Pelanggan dapat membayar menggunakan kredensial login, alamat pengiriman, dan informasi pembayaran yang sama yang telah mereka miliki di Amazon. Layanan Web Pemenuhan memungkinkan pedagang mengakses cloud Amazon melalui layanan web sederhana. Banyak layanan AWS dipelajari di bagian selanjutnya.

Contoh 2.7 AWS Elastic Compute Cloud (EC2) untuk IaaS

Struktur Elastic Compute Cloud (EC2) ditunjukkan pada Gambar 2.14. EC2 mendukung banyak layanan cloud, dengan sekelompok instans mesin. Baik instans Linux dan Windows tersedia. Amazon Machine Images (AMI) menawarkan template untuk membuat instans mesin dari berbagai jenis. AMI publik dapat digunakan secara bebas oleh pengguna mana pun.



Gambar 2.14 Lingkungan eksekusi EC2 tempat Amazon Machine Image (AMI) dapat dibuat dari kumpulan publik, pribadi, atau berbayar dengan perlindungan keamanan.

AMI pribadi dibuat hanya untuk penggunaan pribadi pemilik. AMI berbayar dapat dibagikan di antara pengguna dengan sejumlah pembayaran antara pengguna dan pemilik. Siklus peluncuran AMI ditampilkan di kotak ledakan, di mana keamanan ditegakkan dengan firewall akses instan.

Penskalaan otomatis dan penyeimbangan beban di antara instans didukung di EC2. Instans mesin yang disediakan dalam klaster EC2 dipilih berdasarkan permintaan pengguna. Konfigurasi cluster harus sesuai dengan beban kerja yang diantisipasi. Kami akan mempelajari strategi scale-out dan scale-up untuk kontrol konfigurasi EC2 di Bab 6. Auto Scaling memungkinkan Anda menaikkan atau menurunkan ukuran EC2 secara otomatis, menurut beberapa kondisi ambang. Jumlah instans EC2 dalam sebuah cluster didorong oleh permintaan beban kerja. Auto Scaling sangat cocok untuk aplikasi yang sering mengalami variabilitas beban kerja. Teknik penskalaan secara otomatis dipicu oleh Amazon Cloud- Watch dan tersedia tanpa biaya tambahan bagi pengguna di luar penggunaan Cloud- Watch.

Penawaran Layanan Cloud AWS

Tiga tabel diberikan di bawah ini untuk meringkas penawaran oleh AWS di tiga area layanan utama. Tabel 2.13 menentukan layanan computing, penyimpanan, dan jaringan pita basis data (IaaS). Tabel 2.14 menetapkan aplikasi, seluler, dan layanan analitik yang ditawarkan oleh AWS cloud. Ini adalah penawaran PaaS terkait. Dalam hal layanan yang diberikan, AWS bukan lagi sekadar cloud IaaS murni. Kami secara singkat memperkenalkan layanan AWS di bawah ini. Sejauh ini, EC2 dan S3 adalah layanan IaaS paling populer yang disediakan oleh AWS. Banyak cloud IaaS lainnya, swasta atau publik, juga mencoba membuat sistem cloud mereka kompatibel dengan EC2 dan S3. Layanan RDS mendukung layanan SQL relasional. DB Dinamis mendukung operasi NoSQL melalui *Big data* yang tidak terstruktur. Layanan jaringan mendukung pengelompokan virtual dari sumber daya jaringan.

Tabel 2.13 Layanan computing, penyimpanan, database, dan jaringan di AWS cloud.

Kategori	Menawarkan	Modul Layanan atau Deskripsi Singkat
Menghitung	EC2	Server virtual di cloud AWS
	lambda	Jalankan kode sebagai respons terhadap acara
	Layanan Kontainer EC2	Jalankan dan kelola container Docker
Penyimpanan dan Pengiriman Konten	S3	Penyimpanan yang dapat diskalakan di cloud AWS
	Sistem File Elastis	Sistem file manajemen penuh untuk EC2
	Gerbang Penyimpanan	Integrasikan fasilitas TI lokal dengan penyimpanan cloud
	Gletser	Penyimpanan arsip di cloud AWS
	CloudFront	Jaringan pengiriman konten global
Basis Data	RDS	MySQL, Postgres, Oracle, SQL server
	DynamicDB	Penyimpanan data NoSQL yang dapat diprediksi dan terukur
	ElastiCache	Cache dalam memori
	pergeseran merah	Layanan gudang skala petabyte terkelola
Jaringan	VPC	Cloud pribadi virtual sebagai sumber daya cloud yang terisolasi
	Koneksi langsung	Koneksi Jaringan Khusus ke AWS
	Rute S3	DNS dan pendaftaran nama domain yang dapat diskalakan

Tabel 2.14 Layanan aplikasi, seluler, dan analitik di cloud AWS.

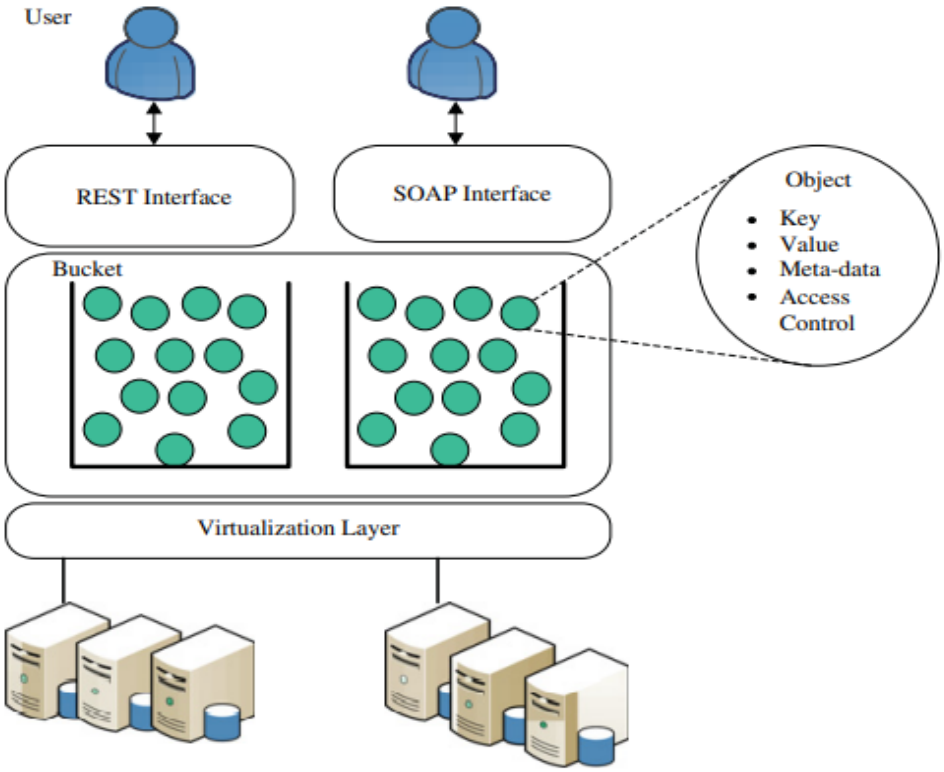
Kategori	Menawarkan	Modul Layanan atau Deskripsi Singkat
Layanan Aplikasi	SQS	Layanan antrian pesan
	SWF	Layanan alur kerja untuk mengoordinasikan komponen aplikasi
	AppStream	Streaming aplikasi dengan latensi rendah
	Transcoder elastis	Transcoding media terukur yang mudah digunakan
	SES	Layanan pengiriman dan penerimaan email
	CloudSearch	Layanan pencarian terkelola
	Gerbang API	Bangun, terapkan, dan kelola API
Layanan Seluler	kesadaran	Identitas pengguna dan sinkronisasi data aplikasi
	Peternakan Perangkat	Uji aplikasi Android, Fire OS, dan iOS di perangkat di cloud
	Analisis Seluler	Kumpulkan, lihat, dan ekspor analisis aplikasi
	SNS	Layanan pemberitahuan push sederhana
Layanan Analisis	ESDM	Kerangka kerja Hadoop (MapReduce) elastis terkelola
	Kinesis	Pemrosesan data streaming secara real-time

	Pipa Data	Orkestrasi untuk alur kerja berbasis data
	Machine learning	Bangun solusi prediksi *Machine learning*

Contoh 2.8 Arsitektur AWS S3 dengan Operasi Data Berorientasi Blok Amazon S3 menyediakan layanan penyimpanan sederhana yang dapat digunakan untuk menyimpan dan mengambil data dalam jumlah berapa pun, kapan pun, dari mana pun di web. S3 menyediakan layanan penyimpanan berorientasi objek untuk pengguna. Pengguna dapat mengakses objek mereka melalui protokol SOAP dengan browser apa pun. Lingkungan eksekusi S3 ditunjukkan pada Gambar 2.15. Unit operasi dasar S3 adalah objek, yang dikaitkan dengan nilai, meta, dan kontrol akses. Setiap objek disimpan dalam ember dan diambil melalui kunci yang unik dan ditetapkan oleh pengembang. Ember adalah wadah objek.

Mekanisme otentikasi disediakan untuk memastikan bahwa data tetap aman dari akses yang tidak sah. Objek dapat dibuat pribadi atau publik, dan hak dapat diberikan kepada pengguna tertentu. Protokol unduhan default adalah http. Tidak ada biaya transfer data antara Amazon EC2 dan S3 dalam Wilayah yang sama. Langkah-langkah untuk menggunakan S3 adalah: i) membuat bucket di region tempat bucket dan objek Anda berada untuk mengoptimalkan latensi, meminimalkan biaya, atau memenuhi persyaratan peraturan; ii) mengunggah Objek ke keranjang Anda, tempat data Anda didukung oleh Amazon SLA; dan iii) kontrol akses bersifat opsional untuk memberi orang lain akses ke data Anda dari mana saja di dunia.

Tabel 2.14 mencantumkan 15 penawaran layanan berorientasi aplikasi oleh AWS. Banyak di antaranya adalah tipe SaaS, kecuali pengguna dapat meminta kluster server mereka sendiri yang disesuaikan untuk menjalankan aplikasi ini. Layanan aplikasi mencakup antrian pesan, streaming waktu nyata, pengiriman email, pencarian, sinkronisasi, operasi pengaturan alur kerja seluler dan analitik. Sebagian besar adalah fitur baru yang ditambahkan ke cloud AWS. Layanan seluler membantu pengguna menyinkronkan data seluler mereka dengan penyimpanan S3 yang disewa. Analisis seluler disediakan untuk menganalisis data ini untuk pengambilan keputusan atau tanggapan. Layanan SNS menangani server pemberitahuan push antara ponsel dan layanan S3.



Gambar 2.15 Layanan penyimpanan Amazon S3 untuk menyimpan objek data tak terbatas.

Platform AWS mendukung banyak klien kecil dan perusahaan besar untuk membangun cloud sewaan mereka guna menjalankan bisnis mereka guna memperoleh keuntungan dari sejumlah besar pengguna Internet. Salah satu contoh yang baik adalah DropBox, yang menerapkan S3 untuk waktu yang lama untuk menyediakan operasi penyimpanan data cadangan mereka sebelum mereka membangun penyimpanan pusat data mereka sendiri. Layanan analitik baru ditambahkan. Mereka menerapkan EMR menggunakan Hadoop atau Spark. Streaming real-time dan dukungan orkestrasi kontainer juga disediakan.

Machine learning Amazon

Amazon *Machine learning* (ML) menawarkan layanan yang memungkinkan ilmuwan data menggunakan teknologi *Machine learning*. Amazon ML menyediakan alat visualisasi dan wizard yang memandu pengguna melalui proses pembuatan model prediksi. Ini membebaskan pengembang dari mempelajari algoritme ML dan alat perangkat lunak yang kompleks. Amazon ML memudahkan untuk mendapatkan prediksi menggunakan API sederhana. Pengguna tidak harus mengimplementasikan kode pembuatan prediksi kustom, atau mengelola infrastruktur apa pun.

Amazon ML didasarkan pada teknologi yang sangat skalabel yang telah digunakan oleh Amazon selama bertahun-tahun. Layanan ini menerapkan algoritme yang kuat untuk membuat model ML dengan menemukan pola dalam set data pelatihan pengguna. Kemudian pengguna menerapkan model prediksi ini ke kumpulan *Big data* yang sedang diuji untuk menghasilkan hasil prediksi. AWS mengklaim bahwa layanan ini dapat menghasilkan miliaran prediksi setiap hari. Prediksi ini disajikan secara real time dan pada throughput yang tinggi. Dengan Amazon ML, analitik *Big data* dilakukan tanpa investasi perangkat keras atau

perangkat lunak di muka. Pengguna cukup membayar sambil berjalan. Pengguna dapat memulai dari yang kecil dan meningkatkannya seiring dengan meningkatnya alur kerja aplikasi mereka.

Di bawah ini tercantum beberapa layanan analitik prediktif yang disediakan oleh AWS: Deteksi penipuan, prediksi churn pelanggan, personalisasi konten, pemodelan kecenderungan untuk kampanye pemasaran, klasifikasi dokumen, dan rekomendasi solusi otomatis untuk dukungan pelanggan. Pembaca yang tertarik mungkin ingin mengakses situs web mereka untuk detailnya: https://aws.amazon.com/machinbe-learng/ Contoh berikut menyajikan salah satu aplikasi layanan Amazon ML dalam aplikasi perumahan komersial.

Sejumlah besar pengguna individu, bisnis kecil, atau sementara menggunakan instans EC2 berdasarkan kebutuhan mereka, secara dinamis. Faktanya, AWS memberikan hibah promosi kepada universitas yang berorientasi pada penelitian bagi siswa untuk belajar dari praktik di AWS cloud. Cloud publik lainnya, seperti mesin komputer Azure dan Google, menyediakan layanan yang sama untuk pemula. Menurut penulis, dengan pengajaran computing cloud di University of Southern California, pengalaman langsung memainkan peran penting bagi siswa untuk mempelajari operasi computing cloud, lingkungan perangkat lunak, dan keterampilan pemrograman. Sejumlah masalah pekerjaan rumah disertakan dalam buku ini, yang dirancang untuk memenuhi tujuan itu.

Platform PaaS Clouds – Google AppEngine

Untuk mengembangkan, menyebarkan, dan mengelola eksekusi aplikasi menggunakan sumber daya yang disediakan, menuntut platform cloud dengan semua lingkungan perangkat lunak yang diperlukan. Platform cloud semacam itu mencakup sistem operasi dan dukungan perpustakaan run-time. Ini telah memicu pembuatan model PaaS untuk memungkinkan pengguna mengembangkan dan menyebarkan aplikasi penggunanya. Layanan platform cloud yang ditawarkan oleh lima penyedia disajikan pada Tabel 2.15. Penyedia layanan PaaS ini termasuk Google AppEngine, Microsoft Azure, Force.com, Amazon Elastic MapReduce, dan Aneka di Australia.

Platform cloud adalah sistem komputer terintegrasi yang terdiri dari infrastruktur perangkat keras dan perangkat lunak. Aplikasi pengguna dapat dikembangkan pada platform cloud virtual ini menggunakan beberapa bahasa pemrograman dan perangkat lunak yang didukung oleh penyedia (misalnya Java, Python, .Net). Pengguna tidak mengelola infrastruktur cloud yang mendasarinya. Penyedia cloud mendukung pengembangan aplikasi pengguna dan pengujian pada platform layanan yang terdefinisi dengan baik. Model PaaS ini memungkinkan sarana untuk memiliki platform pengembangan perangkat lunak yang dikolaborasikan untuk pengguna dari berbagai belahan dunia. Model ini juga mendorong pihak ketiga untuk menyediakan solusi manajemen perangkat lunak, integrasi, dan pemantauan layanan.

Mesin Aplikasi Google

Google adalah salah satu penyedia aplikasi cloud yang lebih besar, meskipun program layanan dasarnya bersifat pribadi dan orang luar tidak dapat menggunakan infrastruktur Google untuk membangun layanan mereka sendiri. Blok bangunan aplikasi computing cloud Google termasuk Sistem File Google untuk menyimpan data dalam jumlah besar, kerangka kerja pemrograman MapReduce untuk pengembang aplikasi, Chubby untuk layanan kunci

aplikasi terdistribusi dan BigTable sebagai layanan penyimpanan untuk mengakses struktur atau semi- data struktural. Dengan blok bangunan ini, Google telah membangun banyak aplikasi cloud.

Aplikasi GAE yang terkenal termasuk Google Search Engine, Google Docs, Google Earth, gmail, dll. Aplikasi ini dapat mendukung sejumlah besar pengguna secara bersamaan. Pengguna dapat berinteraksi dengan aplikasi Google dengan antarmuka web yang disediakan oleh setiap aplikasi. Penyedia aplikasi pihak ketiga dapat menggunakan App Engine untuk membangun aplikasi cloud untuk menyediakan layanan. Semua aplikasi dijalankan di pusat data Google. Di dalam setiap pusat data, mungkin ada ribuan node server untuk membentuk cluster yang berbeda (lihat bagian sebelumnya). Setiap cluster dapat menjalankan beberapa server tujuan. Konfigurasi khas dari sebuah cluster dapat menjalankan sistem file Google, pekerjaan MapReduce, serta server BigTable untuk data struktur. Layanan ekstra seperti Chubby untuk kunci terdistribusi juga dapat berjalan di cluster.

Tabel 2.15 Cloud publik yang menawarkan layanan *platform-as-a-service* (PaaS) (Agustus 2015).

Nama Cloud	**Bahasa dan Alat Pengembang**	**Model Pemrograman yang didukung oleh penyedia**	**Aplikasi Target dan Opsi Penyimpanan**
Google App Engine	IDE berbasis Python, Java, dan Eclipse	MapReduce, pemrograman Web sesuai permintaan	Aplikasi web dan Penyimpanan BigTable
Salesforce.com Force.com	Apex, IDE berbasis Eclipse, Wizard berbasis web	Alur kerja, seperti Excel, pemrograman Web sesuai permintaan	CRM dan pengembangan Aplikasi tambahan untuk Bisnis
Microsoft Azure	NET, alat Azure untuk MS Visual Studio	Dryad, Twister, .NET Framework	Aplikasi Perusahaan dan Web
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, dan C++	MapReduce, Hadoop, Spark,	Pemrosesan Data, eMail, dan e-Commerce, S3 dan WorkDocs

Google App Engine menjalankan program pengguna pada infrastruktur Google. Sebagai platform yang menjalankan program pihak ketiga, pengembang aplikasi tidak perlu lagi khawatir dengan pemeliharaan server. Google App Engine dapat dipahami sebagai kombinasi dari beberapa komponen perangkat lunak. Front end adalah framework aplikasi yang mirip dengan framework aplikasi web lainnya seperti ASP, J2EE atau JSP. Saat ini, Google App Engine mendukung lingkungan pemrograman Python dan Java. Aplikasi dapat berjalan

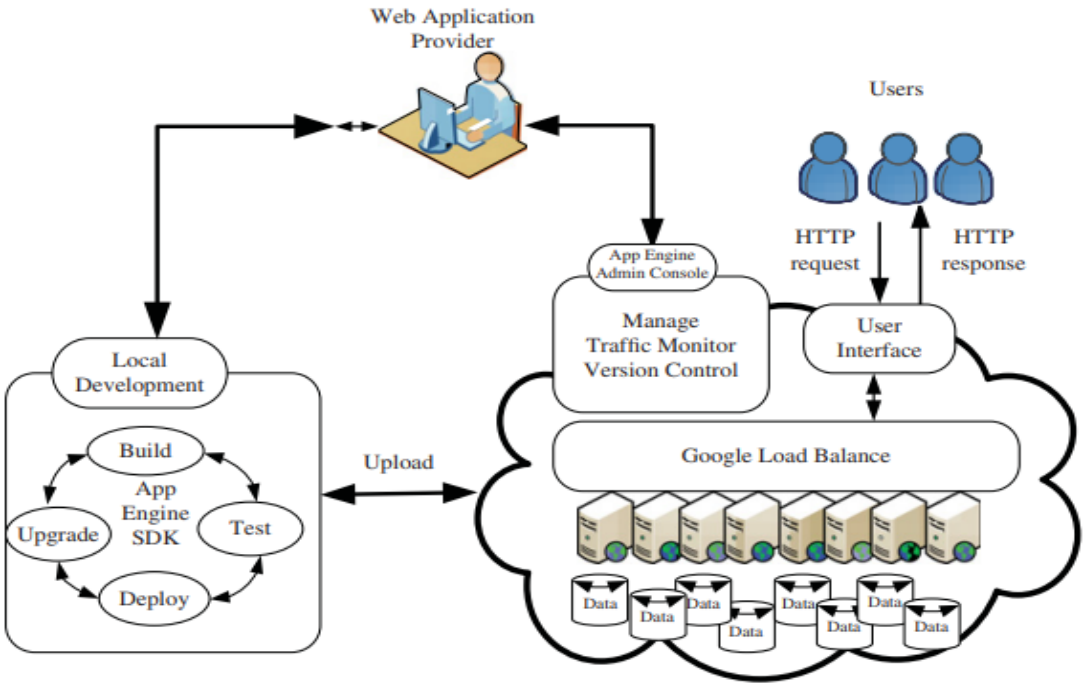
sama seperti dalam wadah aplikasi web. Frontend dapat digunakan sebagai infrastruktur layanan web dinamis, yang dapat memberikan dukungan penuh untuk teknologi umum.

Google memiliki fasilitas mesin pencari terbesar di dunia. Mereka memiliki pengalaman luas dalam pemrosesan *Big data*-besaran yang telah menghasilkan wawasan baru tentang desain pusat data dan model pemrograman baru yang berskala hingga ukuran luar biasa. Pengguna dapat berinteraksi dengan aplikasi Google melalui antarmuka web yang disediakan oleh setiap aplikasi. Pihak ketiga tetapi seperti yang telah dibahas sebelumnya dengan MapReduce, infrastruktur ini dapat diterapkan ke banyak area lain. Google memiliki ratusan pusat data dan telah memasang lebih dari 460.000 server di seluruh dunia. Misalnya, 200 pusat data Google digunakan pada satu waktu untuk sejumlah aplikasi cloud. Item data disimpan dalam teks, gambar dan video dan direplikasi untuk menoleransi kesalahan atau kegagalan. Di sini kita membahas Google App Engine (GAE), yang menawarkan platform PaaS yang mendukung berbagai aplikasi cloud dan web.

Contoh 2.9 Google AppEngine untuk Layanan PaaS dengan Load Balancing

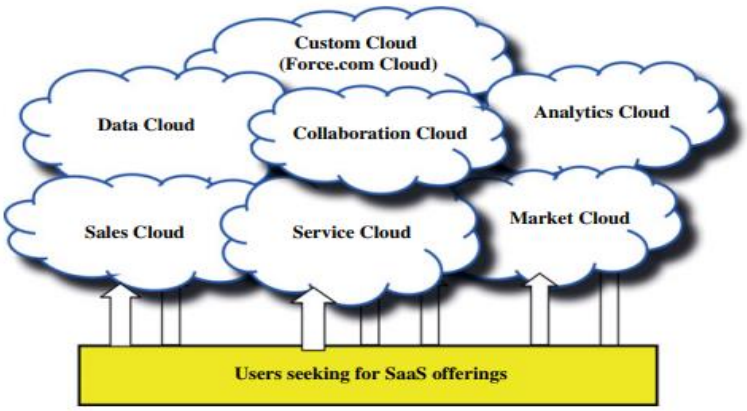
Google telah memelopori pengembangan cloud dengan memanfaatkan sejumlah besar pusat data yang mereka operasikan. Misalnya, Google memelopori layanan cloud di gmail, Google Documents, Google Earth, dll. Aplikasi ini dapat mendukung sejumlah besar pengguna secara bersamaan dengan ketersediaan tinggi. Pencapaian teknologi yang menonjol termasuk Google File System (GFS), MapReduce, Bigtable, Chubby, dll. GAE menjadi platform umum bagi banyak penyedia layanan cloud kecil. Platform ini berspesialisasi dalam mendukung aplikasi web yang dapat diskalakan (elastis). GAE memungkinkan pengguna untuk menjalankan aplikasi mereka di sejumlah besar pusat data yang terkait dengan operasi mesin pencari Google. Angka 2.16 menunjukkan blok bangunan utama platform cloud Google, yang telah digunakan untuk menghadirkan layanan cloud yang disorot di atas.

GFS digunakan untuk menyimpan data dalam jumlah besar. MapReduce digunakan dalam pengembangan program aplikasi. Chubby digunakan untuk layanan kunci aplikasi terdistribusi. BigTable memungkinkan penyedia aplikasi menggunakan App Engine untuk membangun aplikasi cloud. Semua aplikasi berjalan di pusat data di bawah manajemen yang ketat oleh para insinyur Google. Di dalam setiap pusat data, ada ribuan server yang membentuk cluster yang berbeda. Google App Engine mendukung banyak aplikasi web. Salah satunya adalah layanan penyimpanan untuk menyimpan data spesifik aplikasi di infrastruktur Google.



Gambar 2.16 Platform Google AppEngine untuk operasi PaaS dengan penyeimbangan beban.

Data dapat disimpan secara terus-menerus di server penyimpanan backend sambil tetap menyediakan fasilitas untuk kueri, pengurutan, dan bahkan transaksi yang serupa dengan sistem basis data tradisional. Google App Engine menyediakan layanan khusus Google seperti layanan akun gmail. Padahal, layanan tersebut adalah layanan login, yaitu aplikasi dapat menggunakan akun gmail secara langsung. Ini dapat menghilangkan pekerjaan yang membosankan dalam membangun komponen manajemen pengguna yang disesuaikan dalam aplikasi web. Dengan demikian, aplikasi web yang dibangun di atas Google App Engine dapat menggunakan API yang mengautentikasi pengguna dan mengirim email menggunakan Akun Google.



Gambar 2.17 Tujuh penawaran layanan cloud Salesforce: semua untuk aplikasi SaaS kecuali aplikasi PaaS yang menawarkan cloud kustom.

Fungsionalitas Mesin Aplikasi Google

Platform GAE dibangun dengan lima komponen utama. GAE bukanlah platform infrastruktur melainkan platform pengembangan aplikasi untuk pengguna. Kami memperkenalkan di bawah fungsi komponen, secara terpisah:

a) Datastore menawarkan layanan penyimpanan data terstruktur, terdistribusi, dan berorientasi objek berdasarkan teknik Bigtable. Datastore mengamankan operasi manajemen data.
b) Lingkungan runtime aplikasi menawarkan platform untuk pemrograman dan eksekusi web yang dapat diskalakan. Ini mendukung dua bahasa pengembangan: Python dan Java.
c) Software development kit (SDK) digunakan untuk pengembangan aplikasi lokal. SDK memungkinkan pengguna untuk menjalankan uji coba aplikasi lokal dan mengunggah kode aplikasi.
d) Konsol administrasi digunakan untuk memudahkan pengelolaan siklus pengembangan aplikasi pengguna, daripada digunakan untuk pengelolaan sumber daya fisik.
e) Infrastruktur layanan Web GAE menyediakan antarmuka khusus untuk menjamin penggunaan dan pengelolaan penyimpanan dan sumber daya jaringan yang fleksibel oleh GAE.

Google pada dasarnya menawarkan layanan GAE gratis untuk semua pemilik akun gmail. Anda dapat mendaftar untuk akun GAE atau menggunakan nama akun gmail Anda untuk mendaftar ke layanan ini. Layanan ini gratis dalam kuota. Jika Anda melebihi kuota, halaman akan menginstruksikan Anda cara membayar layanan. Kemudian, Anda mengunduh SDK mereka dan membaca Python atau Panduan Java untuk memulai. GAE menerima bahasa pemrograman Python, Ruby dan Java. Platform tidak menyediakan layanan IaaS apa pun. Model ini memungkinkan pengguna untuk menyebarkan aplikasi buatan pengguna di atas infrastruktur cloud, yang dibangun menggunakan bahasa pemrograman dan alat perangkat lunak yang didukung oleh Google (misalnya Java, Python). Azure melakukan hal yang sama untuk platform .Net dan Azure. Pengguna tidak mengelola infrastruktur cloud yang mendasarinya. Penyedia cloud mendukung semua pengembangan, pengujian, dan operasi aplikasi.

Cloud SaaS Aplikasi – Cloud Salesforce

SaaS ini mengacu pada perangkat lunak aplikasi yang diprakarsai oleh browser pada ribuan pelanggan cloud. Layanan dan alat yang ditawarkan oleh PaaS digunakan dalam pembangunan aplikasi dan manajemen penyebarannya pada sumber daya yang ditawarkan oleh penyedia IaaS. Model SaaS menyediakan aplikasi perangkat lunak sebagai layanan. Akibatnya, di sisi pelanggan, tidak ada investasi di muka untuk server atau lisensi perangkat lunak. Di sisi penyedia, biaya tetap rendah, dibandingkan dengan hosting konvensional untuk aplikasi pengguna. Data pelanggan disimpan di cloud yang dimiliki vendor atau dihosting secara publik. Tabel 2.16 merangkum empat platform cloud SaaS.

Contoh terbaik layanan SaaS termasuk Google gmail dan dokumen, Microsoft SharePoint, dan perangkat lunak CRM dari Salesforce.com. Mereka semua sangat sukses dalam mempromosikan bisnis mereka sendiri dan digunakan oleh ribuan bisnis kecil dalam operasi mereka sehari-hari. Penyedia seperti Google dan Microsoft menawarkan layanan IaaS

dan PaaS terintegrasi, sedangkan yang lain seperti Amazon dan GoGrid menawarkan layanan IaaS murni. Penyedia pihak ketiga, seperti Manjrasoft, menawarkan layanan pengembangan dan penerapan aplikasi di atas cloud komersial. SaaS cloud terkenal lainnya adalah Outlook Web Access (OWA), atau dikenal sebagai Office 365, yang ditawarkan oleh Microsoft untuk layanan email yang dihosting di cloud.

Tabel 2.16 Empat platform cloud SaaS dan penawaran layanannya (Agustus 2015).

Model	**Amazon AWS**	**Google App Engine**	**Microsoft Azure**	**Tenaga penjualan**
Dukungan Platform	AWS EC2, S3, EMR, SNS, dll.	GAE, GFS, BigTable, MapReduce, dll.	Azure, layanan .NET, CRM Dinamis,	Salesforce.com, Force.com, CRM Online, Gifttag
Penawaran SaaS	Pohon Kacang Elastis, Code-Deploy, OpsWorks, Code-Commit, Code Pipeline, Mobile Analytics	Gmail, Dokumen, YouTube, WhatsApp	Langsung, SQL, Office”365 (OWA), Hotmail	Penjualan, Layanan, Pasar, Data, Kolaborasi, Analisis
Fitur keamanan	CloudWatch, Penasihat Tepercaya, Kontrol Identitas/Akses	Kunci gemuk untuk penegakan keamanan	Data yang Direplikasi, Kontrol akses berbasis aturan	Keamanan Adm./Rekam, Gunakan Metadata API
API dan Bahasa	API Gateway, LatinPig	Konsol Adm. Berbasis Web, Python	Portal Azure, .net Framework	Apex, Visualforce, AppExchange, SOSL, SOQL

Untuk menemukan obat baru melalui analisis urutan DNA, Eli Lily Company telah menggunakan platform EC2 dan S3 Amazon dengan server dan cluster penyimpanan yang disediakan. Tujuannya adalah untuk melakukan analisis urutan biologis kinerja tinggi tanpa menggunakan superkomputer yang mahal. Manfaat dari aplikasi IaaS ini adalah pengurangan waktu penyebaran obat dengan biaya yang jauh lebih rendah. Contoh bagus lainnya adalah New York Times yang menerapkan layanan Amazon, EC2 dan S3 untuk mengambil informasi bergambar yang berguna dengan cepat dari jutaan artikel arsip dan surat kabar. The New York Times telah secara signifikan mengurangi waktu dan biaya mereka dalam menyelesaikan pekerjaan. Banyak perusahaan cloud startup menyediakan beberapa layanan SaaS dengan platform sewaan seperti AWS.

Berikut ini adalah review layanan SaaS dan PaaS yang ditawarkan oleh Salesforce.com. Perusahaan ini didirikan pada tahun 1999 untuk menyediakan solusi on-line untuk SaaS, terutama dalam aplikasi CRM. Awalnya, mereka menggunakan platform cloud pihak ketiga untuk menjalankan layanan perangkat lunak mereka. Secara bertahap, perusahaan meluncurkan Force.com sendiri sebagai platform PaaS yang dapat menjalankan banyak

aplikasi SaaS atau membantu pengguna mengembangkan aplikasi tambahan di bawah dukungan PaaS.

Contoh 2.10 Layanan Cloud SaaS yang Ditawarkan oleh Perusahaan Salesforce

Baru-baru ini, Salesforce telah membagi layanan CRM-nya ke dalam tujuh kategori layanan cloud spesifik: yaitu Sales Cloud, Service Cloud, Data Cloud, Market Cloud and Collaboration Cloud, Analytics Cloud dan Custom Cloud, seperti yang diilustrasikan pada Gambar 2.17. Di antaranya, semuanya menyediakan aplikasi SaaS, kecuali PaaS Custom Cloud, juga dikenal sebagai Force.com.

Kami secara singkat memperkenalkan fungsinya sebagai berikut:

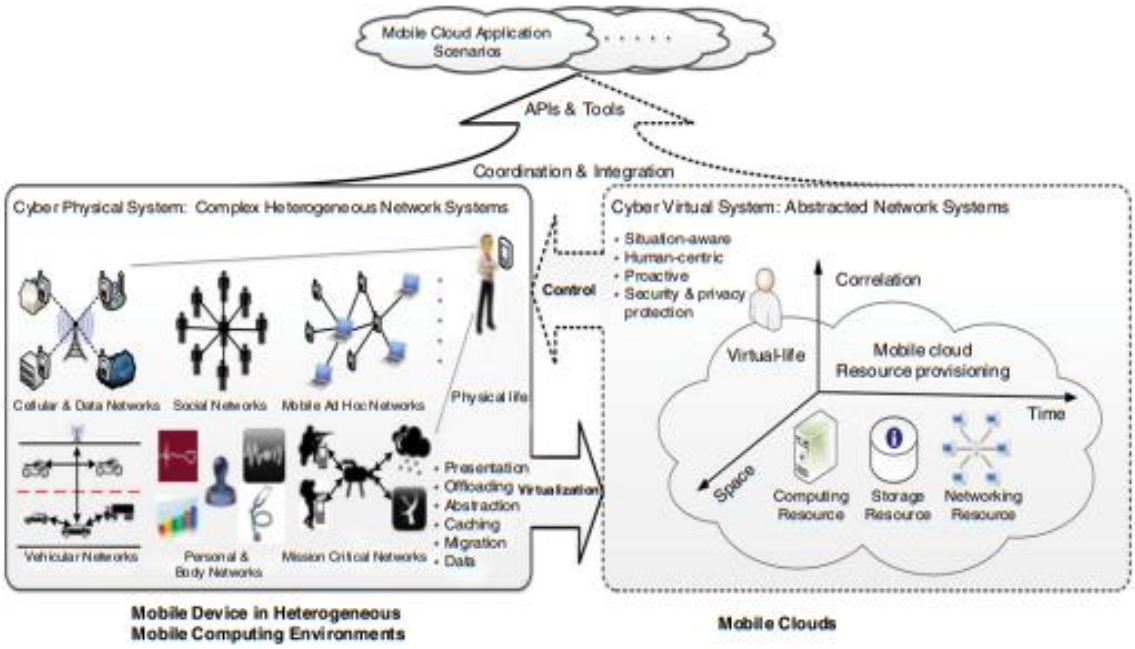
- **Sales Cloud:** untuk aplikasi CRM SaaS untuk mengelola profil pelanggan, pelacakan peluang, mengoptimalkan kampanye, dll.;
- **Service Cloud:** SaaS layanan pelanggan berbasis cloud. Mengizinkan perusahaan membuat, melacak, dan merutekan kasus layanan, termasuk layanan jejaring media sosial;
- **Market Cloud:** menyediakan aplikasi SaaS pemasaran sosial. Memungkinkan perusahaan mengidentifikasi prospek penjualan dari media sosial, menemukan pendukung, dll.;
- **Data Cloud:** untuk memperoleh dan mengelola catatan CRM;
- **Kolaborasi Cloud:** untuk digunakan oleh kolaborator bisnis;
- **Analytics Cloud:** untuk analisis kinerja penjualan berdasarkan *Machine learning*;
- **Cloud Kustom:** platform PaaS untuk membuat aplikasi tambahan di atas aplikasi CRM standar.

2.5 LAYANAN *MOBILE CLOUDS* DAN INTER-CLOUD *MASHUP*

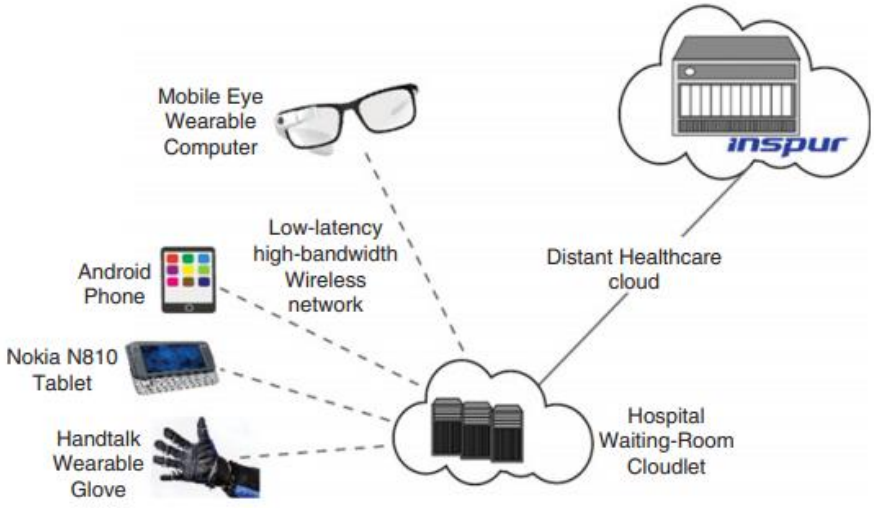
Bagian ini dikhususkan untuk memahami state-of-the-art di *mobile cloud* dan aplikasinya. Materi yang disajikan sangat relevan dengan perangkat seluler, Internet nirkabel, dan teknologi serta platform penginderaan IoT.

Cloud Seluler dan Gateway Cloudlet

Seperti yang ditunjukkan pada Gambar 2.18, pengguna yang membawa perangkat seluler bergerak melintasi lingkungan computing seluler yang heterogen, seperti jaringan seluler, jaringan ad hoc seluler, jaringan area tubuh, jaringan kendaraan, dll. Namun, sifat perangkat seluler yang dibatasi sumber daya, terutama masa pakai baterai yang terbatas, telah menjadi batu sandungan bagi pengguna untuk menikmati peningkatan lebih lanjut dari aplikasi dan layanan seluler. Cloudlet khusus [21] diperkenalkan untuk melayani sebagai gateway nirkabel antara pengguna ponsel dan Internet. Cloudlet ini dapat digunakan untuk membongkar computing atau layanan web ke cloud jarak jauh dengan aman.



Gambar 2.18 Kemampuan perangkat seluler ditingkatkan oleh cloud seluler di lingkungan computing seluler yang heterogen.



Gambar 2.19 Cloudlet berbasis mesin virtual untuk aplikasi computing cloud seluler.

Kombinasi komunikasi seluler dan cloud seluler membuka jalan ke banyak aplikasi yang lebih berguna dalam kehidupan kita sehari-hari. Dengan kata lain, computing tugas berat yang diprakarsai oleh perangkat seluler "kecil" dapat dilakukan oleh cloud "besar". Misalnya, pengguna bergerak di dunia fisik. Sementara itu, data yang melimpah dikumpulkan melalui penginderaan IoT di berbagai lingkungan seluler. Sinyal penginderaan ini harus diarahkan ke cloud untuk penyimpanan data. Objek data virtual dibuat di cloud untuk pengguna. Dengan memanfaatkan sumber daya yang melimpah di platform cloud, penambangan data dan algoritme *Machine learning* sering dikembangkan untuk menganalisis situasi pengguna seluler dan untuk mengambil tindakan tepat waktu secara proaktif. Di bagian bawah Gambar 2.19,

sistem fisik cyber (CPS) dikerahkan untuk melakukan eksekusi terintegrasi dari banyak aplikasi seluler.

Baru-baru ini, para peneliti di Universitas Carnegie-Mellon, Microsoft, AT&T dan Universitas Lancaster telah mengusulkan infrastruktur berbiaya rendah untuk memungkinkan computing cloud menggunakan perangkat seluler. Idenya disebut Cloudlet, yang menawarkan portal kaya sumber daya untuk meningkatkan perangkat seluler dengan kemampuan kognitif untuk mengakses cloud yang jauh, seperti yang ditunjukkan pada Gambar 2.19. Portal ini harus dirancang sebagai dapat dipercaya dan menggunakan mesin virtual untuk menjelajahi aplikasi cloud yang sadar lokasi. Idenya dapat diterapkan untuk penemuan peluang, pemrosesan informasi yang cepat, dan pengambilan keputusan yang cerdas di jalan, dll. Cloudlet memungkinkan perangkat seluler untuk mengakses cloud Internet dengan mudah dalam layanan computing seluler yang hemat biaya.

Ide penggunaan cloudlet untuk *mobile cloud computing* diilustrasikan pada Gambar 2.20. Baik perangkat seluler maupun cloud atau pusat data terpusat memiliki kekurangan dalam mendukung computing seluler. Handset seluler menghadapi masalah properti sumber daya dengan daya CPU yang terbatas, kapasitas penyimpanan, dan bandwidth jaringan pada ponsel pintar atau komputer tablet. Perangkat seluler tidak dapat digunakan untuk menangani kumpulan data yang besar. Di sisi lain, cloud jauh di Internet menghadapi masalah latensi WAN.

Bagaimana mengatasi masalah dua sisi ini adalah tantangan cloudlet untuk disebarkan di tempat-tempat umum seperti kedai kopi, toko buku, dan ruang tunggu rumah sakit untuk akses yang mudah, seperti halnya kenyamanan yang diberikan oleh titik akses untuk layanan WiFi untuk terhubung ke Internet. Cloudlet yang disebarkan secara luas memungkinkan computing cloud terdistribusi dan penanganan sumber daya yang diperluas di toko serba ada, ruang kelas, atau pengguna saat bepergian. Idenya adalah menggunakan cloudlet sebagai gateway atau portal fleksibel untuk mengakses cloud yang jauh. Cloudlet dapat diimplementasikan pada PC, workstation, atau server berbiaya rendah. Inovasi utama terletak pada penggunaan fleksibilitas berbasis VM untuk menangani permintaan dari perangkat seluler yang berbeda.

Sintesis VM Cepat di Cloudlets

Sebuah prototipe cloudlet dengan nama Kimberley dibangun di CMU. Prototipe ini mensintesis overlay VM di host cloudlet. Mereka telah melaporkan waktu sintesis VM yang cepat kurang dari 100 detik. Dengan kata lain, mereka membuat overlay VM di cloudlet sementara, yang disesuaikan untuk mengikat sumber daya cloud di kejauhan untuk memenuhi kebutuhan pengguna. Hamparan VM kecil dikirimkan oleh perangkat seluler ke cloudlet yang sudah memiliki VM dasar. Overlay VM ditambah VM dasar menciptakan lingkungan eksekusi khusus untuk perangkat seluler untuk meluncurkan aplikasi cloud-nya melalui portal cloudlet. Masalah kepercayaan dan keamanan juga merupakan faktor utama dalam penyebaran cloudlet.

Perlindungan data mencakup kontrol akses file/log, pewarnaan data, dan kepatuhan hak cipta. Pemulihan bencana juga diperlukan untuk mengamankan agar tidak hilang karena kegagalan perangkat keras/perangkat lunak. Keamanan cloud dapat ditegakkan dengan

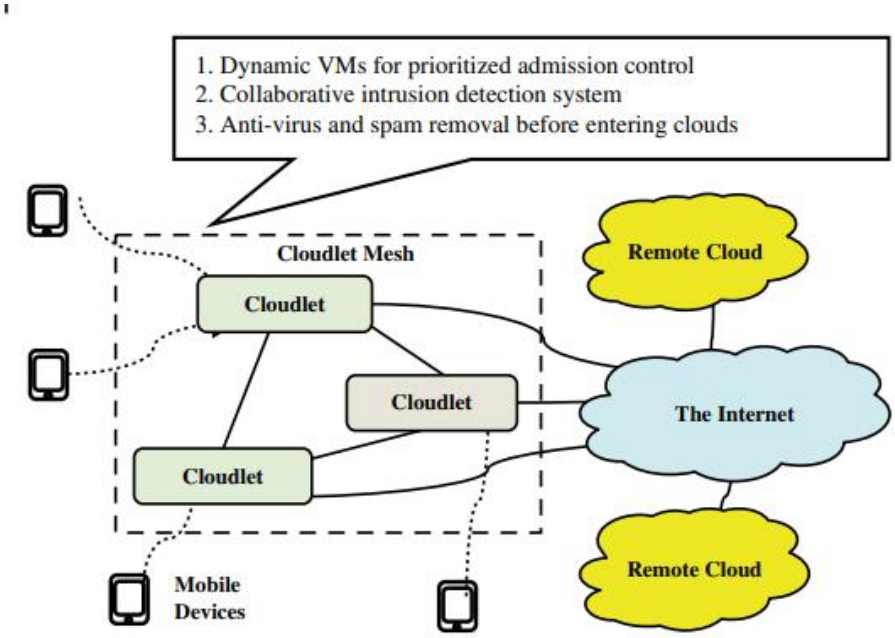
membangun akar kepercayaan, mengamankan proses penyediaan VM, watermarking perangkat lunak, dan penggunaan firewall dan IDS di tingkat host dan jaringan. Baru-baru ini, jaringan overlay kepercayaan dan sistem reputasi telah disarankan untuk melindungi pusat data dalam computing cloud tepercaya.

Arsitektur mesh cloudlet ditunjukkan pada Gambar 2.20. Semua cloudlet berkemampuan WiFi. Setiap server cloudlet memiliki titik akses WiFi tertanam. Setiap cloudlet menghubungkan banyak perangkat seluler dalam jangkauan WiFi. Cloudlet saling berhubungan melalui tautan nirkabel untuk membentuk mesh. Semua cloudlet pada dasarnya beroperasi sebagai gateway di jaringan tepi Internet. Beberapa cloudlet dapat memperluas jangkauan jangkauan nirkabel untuk melayani lebih banyak perangkat seluler. Pertahanan kolaboratif disarankan untuk menggunakan banyak cloudlet secara kolektif untuk membangun perisai untuk mencegah penyusup dan penyerang. Terakhir, caching dan load balancing dipraktikkan untuk meningkatkan QoS dan throughput selama pembongkaran multi-tugas ke cloud jarak jauh.

Perangkat seluler rentan terhadap serangan virus atau worm jaringan. Enkripsi mungkin bukan solusi terbaik untuk perangkat seluler, karena daya computing yang terbatas dan kendala konsumsi energi. Beberapa perangkat lunak khusus tersedia untuk menahan serangan virus atau worm pada perangkat seluler. Ini mungkin melibatkan otentikasi, pemeriksaan URL dan penyaringan spam. Dengan penyimpanan besar dan layanan pencadangan, pengguna seluler akan memindahkan tugas ini ke cloud.

Layanan *Mashup* Multi-Cloud

Cloud *mashup* terdiri dari beberapa layanan dengan kumpulan data bersama dan fungsionalitas terintegrasi. Misalnya, EC2 yang disediakan oleh Amazon Web Service (AWS), layanan otentikasi dan otorisasi yang disediakan oleh Facebook, dan layanan MapReduce yang disediakan oleh Google, semuanya dapat digabungkan untuk memberikan layanan rekomendasi rute mengemudi yang dipersonalisasi secara real-time. Untuk menemukan layanan yang memenuhi syarat dan menyusunnya dengan Quality of Service (QoS) yang terjamin, kami mengusulkan metode pemrosesan kueri skyline terintegrasi untuk membangun aplikasi cloud *mashup*. Kami menggunakan uji kesamaan untuk mencapai cakrawala lokal yang optimal.



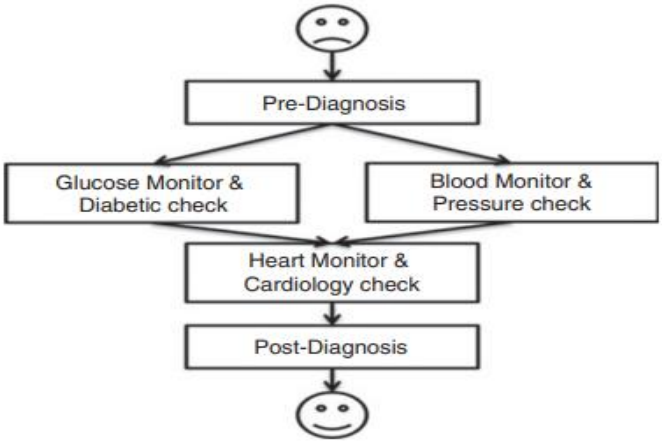
Gambar 2.20 Arsitektur Cloudlet mesh untuk mengamankan computing cloud seluler.

Metode *mashup* ini berskala baik dengan semakin banyaknya situs cloud yang terlibat dalam aplikasi *mashup*. Pemilihan cakrawala yang lebih cepat, pengurangan waktu komposisi, pembagian dataset, dan integrasi sumber daya memastikan QoS melalui banyak cloud. Kami bereksperimen dengan tolok ukur Kualitas Layanan Web (QWS) lebih dari 10.000 layanan Web sepanjang enam dimensi QoS. Dengan memanfaatkan eliminasi blok, partisi ruang data dan pemangkasan kesamaan layanan, proses skyline dipersingkat sepertiga, jika dibandingkan dengan dua metode canggih.

Mashup cloud telah berkembang pesat dengan diperkenalkannya Web 2.0, arsitektur berorientasi layanan, dan manajemen *Big data*. Kumpulan data cloud besar tunduk pada *mashup* dalam layanan antar-cloud. Aplikasi *mashup* menghadapi peningkatan permintaan untuk layanan web/cloud yang dipersonalisasi. Banyak penyedia cloud publik atau komersial bersaing untuk memenuhi permintaan layanan *mashup*. Kesulitan utama berasal dari kenyataan bahwa mungkin ada sejumlah kemungkinan kombinasi. Akibatnya, pemilihan layanan komponen yang optimal merupakan masalah NP-hard dan hanya beberapa layanan komposit sub-optimal yang dapat dihasilkan.

Untuk tujuan ini, operator skyline dan paradigma MapReduce telah disarankan untuk mendukung pemilihan dan komposisi *mashup* antar-cloud. Penelitian sebelumnya mengintegrasikan dua alat canggih yang disebutkan di atas untuk mempercepat proses komposisi layanan dan untuk mencapai QoS yang tinggi. Tujuannya adalah untuk meningkatkan layanan cloud *mashup* dan mempromosikan penggunaan analitik *Big data*. Metode skyline sangat menarik untuk menemukan layanan Web yang berkualitas dalam proses pengambilan keputusan multi-atribut. Kualitas pembuatan layanan Web di cloud *mashup* dapat sangat ditingkatkan dengan pemrosesan kueri skyline MapReduce yang lebih cepat.

Cloud *mashup* dibangun di atas beberapa penyedia layanan web, cloud, dan *Big data*. Istilah ini mengacu pada aplikasi cloud komposit yang menerapkan dan menggabungkan kumpulan data atau fungsi dari lebih dari satu sumber atau penyedia. Motivasinya adalah untuk memberikan lebih banyak kelincahan dan skalabilitas aplikasi dengan memperluas computing cloud dengan aplikasi Internet atau layanan web lainnya. Tujuan desain adalah untuk menawarkan layanan terintegrasi dengan menggabungkan beberapa layanan cloud dengan layanan web terkait yang ditawarkan oleh jejaring sosial dan platform seluler. Misalnya, cloud *mashup* dapat diintegrasikan untuk membentuk alur kerja menggunakan layanan Amazon AWS, DropBox, Twitter, dan Facebook, secara bersama-sama. *Mashup* cloud dibuat dengan memilih API dan tipe data tertentu yang diatur oleh beberapa fungsi layanan yang diinginkan.



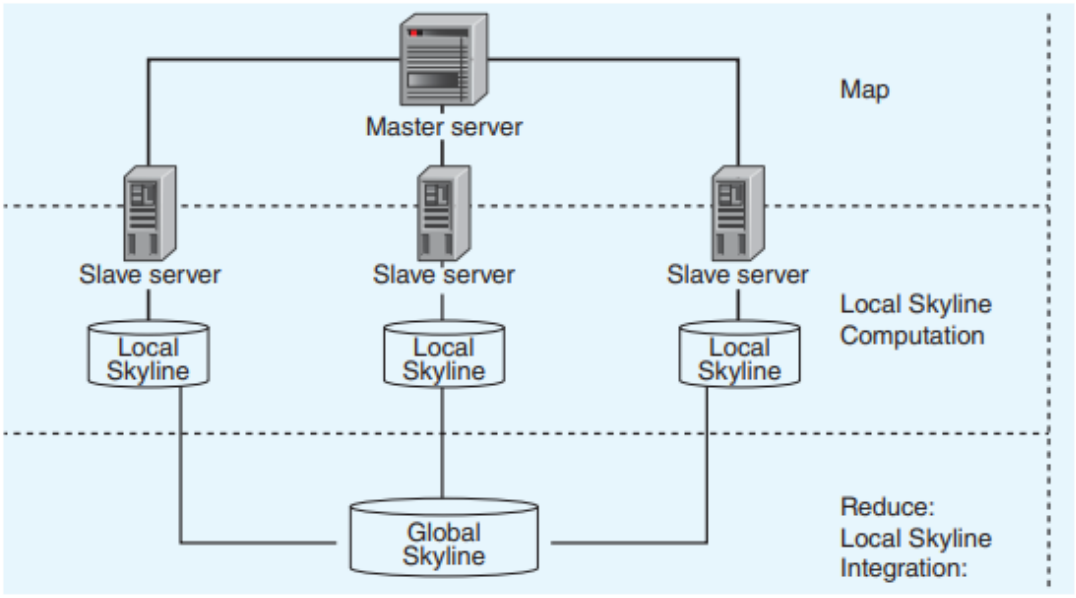
Gambar 2.21 Alur kerja dalam gabungan lima layanan cloud untuk memecahkan masalah perawatan kesehatan pasien.

Contoh 2.11 *Mashup* Beberapa Layanan Cloud dalam Aplikasi Perawatan Kesehatan Misalkan setiap tugas ditangani oleh layanan rumah sakit yang disebarkan di cloud terpisah. Lima layanan cloud membentuk *mashup* yang menyediakan layanan terintegrasi sebagai alur kerja yang dihubungkan oleh tepi terarah dalam DAG (grafik asiklik terarah) pada Gambar 2.21. Permintaan layanan gabungan dikirim ke cloud *mashup* untuk membuat rencana perawatan kesehatan online semacam itu. Output dari alur kerja adalah proses lengkap hingga kepuasan pasien.

Setiap tugas dapat berupa satu layanan yang disediakan oleh satu atau lebih platform berbasis web/cloud. Setiap kandidat layanan dipilih dari ruang layanan besar yang didukung oleh berbagai fungsi cloud. Misalnya, beberapa layanan rumah sakit disebarkan di cloud dengan waktu respons yang cepat dan hasil diagnosis yang memuaskan tetapi dengan biaya yang tinggi. Pasien dengan gejala yang sama perlu memilih kombinasi dari kelima layanan rumah sakit tersebut dengan mempertimbangkan total waktu tunggu dan biaya yang terlibat dalam kelima tugas tersebut.

Dalam aplikasi kehidupan nyata, kesulitannya terletak pada kumpulan layanan cloud yang tersedia yang sangat besar. Lebih buruk lagi, pemeriksaan medis mungkin melibatkan lebih banyak tugas dibandingkan dengan lima tugas. Jika beberapa pasien memilih layanan yang sama dari penyedia cloud yang sama, daftar tunggu akan panjang. Sejalan dengan itu, waktu dan biaya menunggu menjadi tidak dapat diprediksi. Oleh karena itu, metode yang

efisien harus dikembangkan untuk membantu pengguna memilih alur kerja yang tersusun dari layanan cloud sebagai QoS yang dijamin.



Gambar 2.22 Model MapReduce untuk pemilihan layanan skyline untuk mengoptimalkan QoS.

Gambar 2.22 mengilustrasikan ide pemrosesan kueri skyline yang dioptimalkan dalam tiga bagian: pemilihan skyline, uji kesamaan, dan komposisi layanan. Kami memilih layanan skyline berdasarkan partisi eliminasi blok dari ruang data. Cakrawala dapat menghasilkan sejumlah besar layanan kandidat. Untuk menemukan pilihan terbaik di setiap subruang skyline, metode relaksasi skyline dapat digunakan untuk mempertimbangkan hanya perwakilan di setiap subruang. Tujuannya adalah untuk mempercepat komposisi layanan yang dijamin QoS selanjutnya.

Ketiga kelas layanan komponen ini membentuk layanan cloud komposit. Untuk mengurangi waktu komposisi, pengujian kesamaan di antara pilihan skyline yang kompatibel dapat digunakan di berbagai sektor skyline. Tujuannya adalah untuk menghilangkan redundansi menggunakan perwakilan skyline. Terakhir, kami menyusun layanan *mashup* sebagai paket terintegrasi untuk pengguna. QoS dan QoE menentukan persyaratan kinerja yang diinginkan dalam layanan *mashup*.

Kualitas Layanan Mashup (QoMS)

QoS secara langsung mengevaluasi atribut metrik kinerja yang berbeda dari layanan *mashup* komposit. Ambil "perencana kesehatan online" sebagai contoh. Kita dapat mempertimbangkan waktu tunggu, waktu layanan, biaya, reputasi, keandalan, dan ketersediaan untuk setiap tugas. Waktu respon merupakan faktor utama QoS, karena memperhitungkan lalu lintas komunikasi ketika pengguna mengakses layanan dan memiliki dampak besar pada kualitas layanan. Durasi layanan komposit yang dihitung oleh CPA bukanlah durasi optimal maupun durasi aktual, tetapi merupakan metode evaluasi terbaik sejauh ini selama prosedur komposisi. Tiga atribut pertama dari layanan gabungan, waktu

tunggu, waktu layanan, dan biaya, tidak hanya bergantung pada tugas-tugas dasarnya tetapi juga operasi di antaranya; ketika tiga yang terakhir, yaitu reputasi, keandalan, dan ketersediaan, diturunkan dari atribut-atribut dasarnya.

Kualitas Pengalaman (QoE)

Bagaimana pelanggan puas dengan solusi yang diberikan oleh layanan komposit adalah evaluasi penting dari QoE. Misalnya, seluruh rencana medis yang dibuat oleh perencana adalah solusi dari layanan gabungan, dan kualitas rencana medis tergantung pada solusi dari setiap tugas, yaitu, aplikasi perawatan medis, penyedia layanan cloud, dll. tidak memiliki, sejauh pengetahuan kami, penggabungan kualitas solusi ke dalam prosedur komposisi layanan. Orang mungkin berpendapat bahwa "reputasi" dapat digunakan untuk memasukkan bagaimana pengguna puas, tetapi dalam hal layanan daripada melihat ke bawah ke dalam solusinya.

Kami mendefinisikan kriteria QoE sebagai tingkat kepuasan persentil untuk solusi solusi layanan. Seperti yang diberi label ke dalam simpul, setiap solusi diberi skor yang menunjukkan kualitas solusi yang ditentukan pelanggan. Metode penilaian kualitas solusi terbagi dalam dua kategori: berbasis statistik atau berbasis profil. Metode berbasis statistik mencetak solusi dari pemungutan suara pelanggan atau meninjau komentar. Metode berbasis profil memperkirakan tingkat kepuasan pelanggan secara dinamis, misalnya menggunakan perbandingan pasangan, dan mempertahankan profil khusus pengguna. Menggunakan metode berbasis statistik dan berbasis profil, pelabelan skor layanan dapat ditawarkan baik offline atau online. Skor dapat diberikan terlebih dahulu atau dihasilkan secara dinamis selama proses komposisi.

Penemuan Skyline dari Layanan *Mashup*

Diberikan satu set Q titik data dalam ruang QoS dimensi-d, setiap dimensi mewakili atribut kinerja dengan nilai yang diurutkan dengan benar. Misalkan poin yang bernilai lebih rendah lebih baik daripada yang bernilai lebih tinggi. Sebuah titik data Pj didominasi oleh Pi, jika Pi lebih baik dari atau sama dengan Pj di semua dimensi. Selanjutnya, Pi harus lebih baik dari Pj setidaknya dalam satu dimensi. Semua titik data yang tidak didominasi oleh titik lain dari subset disebut skyline. Misalnya, mari kita pilih dua titik dimensi ganda (10, 20) dan (20, 10). Karena titik-titik tersebut tidak saling mendominasi, kedua titik tersebut merupakan bagian dari skyline.

Di ruang d-dimensi, kaki langit benar-benar merupakan permukaan yang paling dekat dengan asalnya dari ruang terkoordinasi. Secara intuitif, semua titik di kaki langit lebih diinginkan daripada semua titik data di luar kaki langit. Kueri cakrawala memilih titik terbaik atau paling menarik di semua dimensi. Ada beberapa pekerjaan yang menerapkan MapReduce untuk meningkatkan efisiensi computing dengan kinerja terukur dalam pemrosesan kueri skyline skala besar. Pendekatan kami didasarkan pada metode eliminasi blok baru. Selanjutnya, kami mengusulkan varian dari metode MapReduce dengan menambahkan proses antara Map dan Reduce. Idenya diilustrasikan pada Gambar 2.22 dalam tiga langkah:

1) Proses Peta: Titik data layanan dipartisi oleh server master (misalnya UDDI) menjadi beberapa blok data berdasarkan permintaan QoS. Blok data dikirim ke server budak untuk pemrosesan paralel.
2) Perhitungan Skyline Lokal: Dalam proses ini, setiap server slave menghasilkan skyline lokal dari titik data layanan pada blok data yang dibaginya sendiri.
3) Proses Reduce: Dalam proses ini, skyline lokal yang dihasilkan oleh semua server slave digabungkan dan diintegrasikan ke dalam skyline global, yang berlaku untuk semua layanan yang dievaluasi.

Kualitas layanan skyline yang dipilih tergantung pada efisiensi computing skyline lokal dan kinerja proses integrasi. Dengan demikian, efisiensi dan QoS dari proses skyline MapReduce tergantung terutama pada bagaimana mengeksplorasi paralelisme terdistribusi untuk mempercepat tahap Map. Efisiensi pemetaan tergantung pada partisi ruang data. Titik data layanan dipartisi menjadi wilayah yang dibagi. Tujuannya adalah untuk mencapai penyeimbangan beban, agar sesuai dengan memori lokal, dan untuk menghindari computing berulang ketika layanan lama dihentikan dan layanan baru ditambahkan secara dinamis. Sebelum proses Reduce, kami memperkenalkan proses tengah (computing skyline lokal) pada Langkah 2.

Alasannya adalah bahwa computing layanan skyline mahal jika jumlah kandidat layanan sangat besar. Dengan memperkenalkan proses tengah, hanya layanan skyline lokal yang dikirimkan ke proses Reduce pada langkah 3. Ini akan mengurangi sebagian besar jumlah layanan yang akan diproses pada tahap Reduce. MapReduce efektif untuk mempercepat proses pemrosesan query skyline. Kita perlu membandingkan layanan berpasangan secara paralel. Dengan MapReduce, layanan baru pertama kali dipetakan ke dalam grup dan ditambahkan ke dalam perhitungan skyline lokal. Kemudian semua skyline lokal diintegrasikan ke dalam skyline global pada tahap Reduce. Kami telah mengadopsi metode skyline untuk memecahkan masalah QoS dalam dua karya sebelumnya. Kami mengevaluasi tiga versi MapReduce dari algoritma skyline BNL (Block Name Label) berdasarkan tiga skema partisi ruang data yang berbeda.

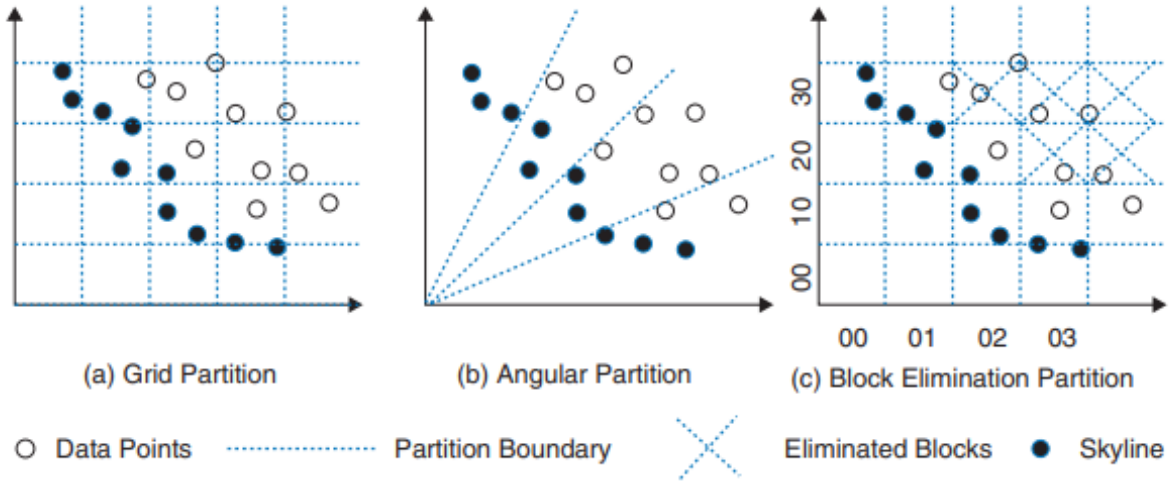
Pertimbangkan dua titik data layanan s1, s2, di ruang QoMS Q. Layanan s1 mendominasi layanan s2, jika s1 lebih baik dari atau sama dengan s2 di semua dimensi atribut Q. Selanjutnya, s1 harus lebih baik dari s2 di setidaknya satu dimensi atribut. Subset S dari layanan membentuk kaki langit di ruang Q, jika semua titik layanan di kaki langit lebih baik dari atau sama dengan layanan lain di sepanjang semua dimensi atribut. Dengan kata lain, semua layanan skyline tidak didominasi oleh layanan lain di ruang Q. Kami mengevaluasi tiga metode sky-line MapReduce, dilambangkan sebagai MR-grid, MR-angular dan MR-block, di mana MR adalah singkatan dari MapReduce di semua gambar label dan badan teks. Tiga algoritma skyline MapReduce ditentukan berdasarkan tiga skema partisi data yang ditunjukkan pada Gambar 2.23(a,b,c). Sumbu x dan sumbu y adalah dua dimensi atribut yang mendukung nilai yang lebih rendah.

Algoritma MR-grid berisi dua tahap: i) Partitioning Job, di mana kita membagi ruang data menjadi beberapa subruang yang terpisah dan menghitung skyline lokal dari setiap subruang; dan ii) Merging Job, di mana kita menggabungkan semua skyline lokal untuk

menghitung skyline global. Secara empiris, jumlah partisi ditetapkan sebagai (2 kali node) dalam algoritma MR-grid. Pada MR-grid, nilai parameter QoS di semua dimensi digunakan untuk melakukan partisi. Misalnya, kami memisahkan ruang data dua dimensi menjadi 16 blok sesuai dengan waktu respons setiap layanan pada Gambar 2.23(a). Metode ini mudah diimplementasikan, sementara banyak perhitungan redundan ada dalam metode ini. Metode ini perlu menyeimbangkan beban kerja dalam proses Reduce.

Komposisi Dinamis Layanan *Mashup*

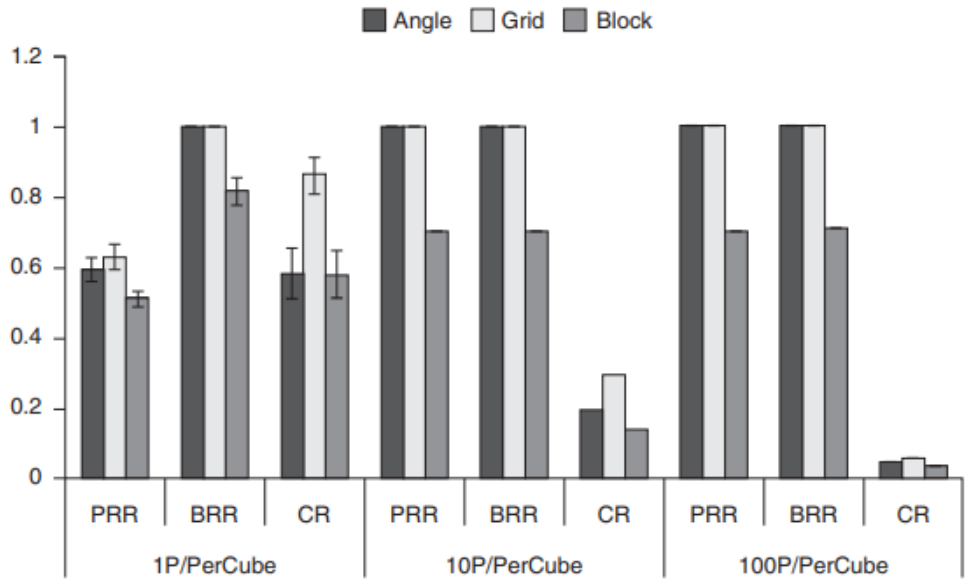
MR-grid diperkenalkan dalam Algoritma 2.1 di bawah ini. Seluruh ruang layanan pertama-tama dipartisi menjadi N bagian yang terpisah-pisah. Poin dalam satu partisi dikirim ke satu Tugas Peta, dan setiap Tugas Peta dapat memproses satu atau lebih partisi. Map Task menampilkan nomor partisi sebagai kunci, dan daftar skyline lokal dari partisi tertentu sebagai nilai. Pada fase reduce, semua skyline lokal diproses melalui Reduce Task, dan oleh karena itu, skyline Global dihasilkan. Metode partisi sudut awalnya diusulkan oleh desain DK. Kami menekankan komposisi layanan pilihan skyline, yang bertujuan untuk mencapai QoMS yang dioptimalkan sehubungan dengan serangkaian sumber daya dan kendala biaya tertentu.



Gambar 2.23 Tiga metode partisi data untuk pemrosesan kueri skyline MapReduce. (Dicetak ulang dengan izin dari F. Zhang, K. Hwang, dkk. 2016 [23]).

Algoritma eliminasi blok ditingkatkan dari algoritma Partisi Grid di bagian C. Pertimbangkan daftar kotak diagonal, di mana layanan di dalam sel grid kiri-bawah mendominasi layanan di dalam sel grid kanan-atasnya. Ambil Gambar 2.23(c) sebagai contoh. Sel (1, 2) memiliki dua titik, oleh karena itu semua sisa titik di sepanjang garis diagonal, misalnya Sel (2, 3) dan Sel (3, 4), dapat dihilangkan tanpa pemrosesan lebih lanjut. Algoritma memeriksa semua sel dengan setidaknya satu nilai koordinat sama dengan nol. Metode eliminasi berbasis blok mengurangi beberapa blok pada langkah 1, dan dua metode lainnya (Sudut dan Kisi) tidak mengurangi blok apa pun. Point Reduction Rate (PPR) diukur dengan agregat titik skyline lokal yang digulirkan ke langkah 2, di atas jumlah total poin. Block Reduction Rate (BRR) didefinisikan sebagai blok yang berisi titik skyline lokal yang digulirkan ke langkah 2, di atas jumlah total blok. Pada Gambar 2.29, kami memilih tiga kelompok data berdasarkan kepadatan data yang didistribusikan secara acak.

Kami mendefinisikan Rasio Skyline (CR) sebagai jumlah pasangan dari semua blok yang perlu berpasangan dibandingkan dengan jumlah total pasangan yang harus dihitung pada langkah 2. Ketiga metrik kinerja lebih memilih nilai rendah untuk menghasilkan hasil yang dipersempit kumpulan poin untuk analisis lebih lanjut. Misalnya, 100 P/Per Cube mewakili kepadatan tinggi dari setiap ruang, yang berarti setiap partisi kecil memiliki sekitar 100 titik di dalamnya. Gambar 2.24 memplot hasil untuk data yang terdistribusi secara acak dalam ruang 4-D. Untuk mengevaluasi efisiensi berbagai metode pemilihan skyline MapReduce, kami menggunakan metrik dasar waktu pemrosesan, yang terdiri dari waktu reduksi dan waktu peta. Singkatnya, dengan kardinalitas layanan yang sangat besar dari 10.000 titik data lebih dari 10 atribut, metode MR-block kami mengungguli metode MR-grid dan MR-angular dengan faktor masing-masing sekitar 3 dan 1,5.



Gambar 2.24 Kinerja relatif dari tiga metode MapReduce untuk kinerja cloud *mashup* (Dicetak ulang dengan izin dari F. Zhang, K. Hwang, et al., 2016. [23])

2.6 KESIMPULAN

Arsitektur cloud dipelajari dalam bab ini. Kami mempertimbangkan cloud yang dapat diterapkan pada penyimpanan dan pemrosesan *Big data* dalam aplikasi analitik. Kemudian kami mendedikasikan Bagian 2.2 dan 2.3 untuk teknik virtualisasi. Konsep virtualisasi diperkenalkan dengan hypervisor dan wadah Docker, yang meletakkan dasar untuk konstruksi cloud dan manajemen elastis. Studi kasus cloud AWS, GAE dan Salesforce diberikan di Bagian 2.4. Kemudian kami memperkenalkan kemajuan terbaru dalam layanan cloud *mashup* dan alat perangkat lunak untuk aplikasi.

Layanan cloud *mashup* diproyeksikan akan tumbuh pesat dalam dekade mendatang karena pertumbuhan populasi cloud publik. Kami menyajikan penemuan cakrawala dan komposisi layanan cloud *mashup*. Secara umum, kinerja tinggi mendorong produktivitas cloud. QoS di cloud didasarkan pada preferensi pengguna. Untuk berbagai tolok ukur cloud dan pemodelan kinerja cloud, pembaca dirujuk ke buku baru Hwang tentang Cloud and

Cognitive Computing: Principles, Architecture, Programming, yang akan terbit pada tahun 2017.

Tugas dan Latihan

1. Pilih tiga sistem cloud IaaS dari berikut ini: Amazon AWS, GoGrid, OpSource Cloud, Rackspace, HP Cloud, Banknorth, dan Fleiscale. Lakukan studi mendalam di luar apa yang telah Anda baca dalam buku ini. Anda perlu menggali informasi teknis yang berguna dengan mengunjungi situs web penyedia atau dengan mencari melalui Google, Wikipedia, dan literatur terbuka apa pun. Tujuannya adalah untuk melaporkan kemajuan terbaru mereka dalam teknologi cloud, penawaran layanan, aplikasi perangkat lunak yang baru-baru ini dikembangkan, model bisnis yang diterapkan, dan pelajaran keberhasilan/kegagalan yang diperoleh. Pastikan laporan studi Anda secara teknis kaya akan konten dan hindari promosi penjualan.
2. Dari cloud PaaS berikut: Google Compute Engine, Force.com, postini, MS azure, NetSuite, IBM RC2, IBM Blue Cloud, SGIN Cyclone, dan Amazon Elastic MapReduce, dll., pilih tiga untuk terlibat dalam studi yang lebih mendalam di luar apa telah Anda baca dalam buku ini. Anda perlu menggali informasi teknis yang berguna dengan mengunjungi situs web penyedia atau dengan mencari melalui Google, Wikipedia, dan literatur apa pun. Tujuannya adalah untuk melaporkan kemajuan terbaru mereka dalam teknologi cloud, penawaran layanan, aplikasi perangkat lunak yang dikembangkan, model bisnis yang diterapkan, dan pembelajaran tentang keberhasilan/kegagalan. Pastikan laporan studi Anda secara teknis kaya akan konten dan hindari promosi penjualan.
3. Pilih tiga sistem cloud SaaS dari berikut ini: Consur, RightNow, Salesforce, Kenexa, Webex, Balckbaud, Netsuite, Omniture, Kenexa, Vocus, Google, dan Microsoft Azure. Lakukan studi mendalam di luar apa yang telah Anda baca dalam buku ini. Anda perlu menggali informasi teknis yang berguna dengan mengunjungi situs web penyedia atau dengan mencari melalui Google, Wikipedia, dan literatur terbuka apa pun. Tujuannya adalah untuk melaporkan kemajuan terbaru mereka dalam teknologi cloud, penawaran layanan, aplikasi perangkat lunak yang baru-baru ini dikembangkan, model bisnis yang diterapkan, dan pembelajaran tentang keberhasilan/kegagalan. Pastikan laporan studi Anda secara teknis kaya akan konten dan hindari promosi penjualan.
4. Pilih dua Layanan Cloud Co-Location dari berikut ini: Savvis, Internap, NTTCommunications, Digital Realty, Trust, dan 365 main. Lakukan studi mendalam di luar apa yang telah Anda baca dalam buku ini. Anda perlu menggali informasi teknis yang berguna dengan mengunjungi situs web penyedia atau dengan mencari melalui Google, Wikipedia, dan literatur terbuka lainnya. Tujuannya adalah untuk melaporkan kemajuan terbaru mereka dalam teknologi cloud, penawaran layanan, aplikasi perangkat lunak yang baru-baru ini dikembangkan, model bisnis yang diterapkan, dan pelajaran keberhasilan/kegagalan yang diperoleh. Pastikan laporan studi Anda secara teknis kaya akan konten dan hindari promosi penjualan.
5. Periksa situs web cloud AWS. Rencanakan aplikasi computing nyata menggunakan cloud komputer Elastis (EC2), Layanan Penyimpanan Sederhana (S3), atau Layanan antrian Sederhana (SQS), secara terpisah. Anda harus menentukan sumber daya yang diminta dan

menghitung biaya yang dibebankan oleh Amazon. Lakukan eksperimen EC2, S3 dan SQS pada platform AWS dan laporkan serta analisis hasil kinerja yang diukur. Pekerjaan rumah ini dapat diperluas menjadi proyek jangka yang dilakukan oleh tim siswa.

6. Dalam Contoh 2.7 dan 2.8, Anda telah mempelajari bagaimana layanan EC2 dan S3 ditawarkan oleh AWS. Kunjungi situs web: https://www.aws.com untuk pembaruan tentang layanan dan produk terbaru dari AWS. Gali fungsionalitas dan aplikasi layanan tambahan yang disediakan oleh AWS. Laporan Anda harus seteknis mungkin. Jangan berspekulasi, karena semua yang Anda laporkan harus dibuktikan dengan bukti yang meyakinkan.

   Pemrosesan kueri skyline yang dioptimalkan:

   a) Apa itu layanan Simple Notification Service (SNS) di cloud AWS? Jelaskan cara kerjanya dan antarmuka pengguna untuk SNS dalam menggunakan ponsel untuk mengirim dan menyimpan aliran foto di S3.
   b) Apa itu Elastic MapReduce (EMR) di AWS? Bagaimana implementasinya? Apa bahasa yang diterapkan untuk menggunakan EMR dan bagaimana cara kerjanya dengan sistem Hadoop?
7. Layanan AWS baru ditawarkan untuk virtualisasi menggunakan Docker Engine untuk membuat wadah perangkat lunak aplikasi. Layanan ini dikenal sebagai Amazon EC2 Container Service (ECS). Jelaskan bagaimana hal itu dilakukan di cloud AWS. Laporkan penerapan ECS dan diskusikan pengalaman dalam menggunakan container dan instans VM pada instans EC2.
8. VM dan manajemen orkestrasi cluster container adalah topik hangat di antara penyedia cloud dan klien cloud. Pilih satu atau lebih VM/penjadwalan container dan alat orkestrasi dari daftar berikut: Armada CorOS, Mesosphere Marathon, Docker Swarm, Apache mesos, Google Kubernetes, dan penulisan Docker. Alat yang Anda pilih harus saling mendukung dalam pengelolaan dan orkestrasi VM/cluster container. Lakukan studi mendalam dengan mengunjungi situs web perusahaan untuk menemukan pengalaman mereka dalam menggunakan perangkat lunak ini. Tulis laporan teknis singkat berdasarkan temuan penelitian Anda.
9. vSphere/4 yang tercakup dalam Contoh 2.6 adalah OS cloud solid yang tersedia secara komersial dari VMware. Cari literatur untuk laporan porting dan pengalaman aplikasi dan kinerja yang diukur oleh klien atau grup pengguna. Tulis laporan teknis singkat untuk meringkas temuan penelitian Anda.
10. Kunjungi situs web iCloud https://www.icloud.com atau Wikipedia untuk mengetahui fungsionalitas dan layanan aplikasi yang disediakan oleh Apple iCloud. Secara khusus, jawab pertanyaan berikut di iCloud:

    a) Secara singkat, tentukan layanan utama yang disediakan oleh iCloud. Berapa banyak pengguna? dilaporkan sampai sekarang?
    b) Apa saja jenis data atau item informasi yang ditangani oleh iCloud?
    c) Jelaskan prosedur menemukan teman lama menggunakan layanan Temukan Teman Saya di iCloud.

d) Jelaskan fitur iCloud Temukan iPhone Saya untuk menemukan i-phone Anda yang hilang atau dicuri.

11. Membandingkan kekuatan, kelemahan, dan aplikasi VM yang sesuai yang dibuat oleh hypervisor di atas bare metal dengan wadah aplikasi yang dibuat oleh mesin Docker pada host Linux. Anda harus membandingkannya berdasarkan tuntutan sumber daya, overhead pembuatan, mode eksekusi, kompleksitas implementasi, lingkungan eksekusi, isolasi aplikasi, fleksibilitas OS, dan platform host.
12. Pilih dua cloud Network-as-a-Service (NaaS) dari berikut ini: Owest, AT&T, dan Abovenet. Lakukan studi mendalam di luar apa yang telah Anda baca di buku teks. Anda perlu menggali informasi teknis yang berguna dengan mengunjungi situs web penyedia atau dengan mencari melalui Google, Wikipedia dan literatur lainnya. Tujuannya adalah untuk melaporkan kemajuan terbaru mereka dalam teknologi cloud, penawaran layanan, aplikasi perangkat lunak yang baru-baru ini dikembangkan, model bisnis yang diterapkan, dan pembelajaran tentang keberhasilan/kegagalan. Pastikan laporan studi Anda secara teknis kaya akan konten dan hindari promosi penjualan.
13. Pertimbangkan Ketersediaan Sistem (A) dari cluster server dalam tiga parameter: yaitu waktu rata-rata untuk kegagalan (MTTF), waktu rata-rata untuk perbaikan (MTTR) dan waktu pemeliharaan reguler (RMT). MTTF mencerminkan waktu rata-rata antara dua kegagalan alam yang berdekatan. MTTR adalah waktu henti karena kegagalan alam. RMT mengacu pada waktu henti yang dijadwalkan untuk pemeliharaan atau pembaruan perangkat keras/perangkat lunak.
    a) Diberikan sistem cloud dengan ketersediaan yang diminta A = 98%. Jika MTTF adalah diketahui 2 tahun (atau 365 × 24 × 2 = 17.520 jam) dan MTTR dikenal sebagai 24 jam. Berapa nilai RMT dalam jam per bulan yang dapat Anda jadwalkan untuk sistem cloud ini?
    b) Pertimbangkan cluster cloud dari 3 server. Cluster dianggap tersedia (atau dapat diterima dengan tingkat kinerja yang memuaskan), jika setidaknya k server beroperasi secara normal di mana k 3. Asumsikan bahwa setiap server memiliki tingkat ketersediaan p (atau tingkat kegagalan 1 – p). Turunkan formula untuk menghitung total ketersediaan cluster A (yaitu probabilitas bahwa cluster tersedia dengan memuaskan). Perhatikan bahwa A adalah fungsi dari k dan p.
    c) Mengingat bahwa setiap server memiliki ketersediaan p = 0,98. Berapa jumlah server minimum terbesar yang harus tersedia untuk mencapai total ketersediaan cluster A, yang lebih tinggi dari 96%? Anda harus memeriksa efek dari semua kemungkinan nilai k di bagian (b) untuk menjawab pertanyaan ini dengan benar.
14. Kami telah mempelajari layanan cloud AWS dan Salesforce. Kunjungi situs web mereka untuk menggali fungsionalitas terperinci dan fitur layanan dalam penawaran layanan berikut oleh AWS, Google, Saleforce, Savvis, dan Apple icloud:
    a) AWS Glacier, CloudFront, RDS, VPC, Direct Coonect, SQS, Elastic Transcoder, Cloud Search, API Gateway, Mobile Analytics, Data Pipeline, Kinesis, *Machine learning*, Trusted Advisor, CloudWatch, WorkMail, Elastic Beanstalk, CodeCommit, dan Code Pipline;

b) Layanan cloud Salesforce: Penjualan, Data, Pasar, Layanan, Kolaborator, dan Anlytics, dan cloud kustom.

15. Pelajari tentang layanan kontainer AWS dan jalankan kode kontainer sampel Amazon ECS. Anda perlu mengambil beberapa tangkapan layar untuk membuktikan bahwa Anda telah melakukannya dengan benar. Anda perlu melaporkan apa yang telah Anda pelajari dari pengujian ini.

Langkah 1: Pelajari tentang Amazon EC2 Container Service, tonton videonya di sini: https://aws.amazon.com/ecs/
Lihat panduan pengembang: http://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html

Langkah 2: Sebelum menggunakan layanan, siapkan lingkungan eksekusi dengan Amazon ECS: http://docs.aws.amazon.com/AmazonECS/latest/developerguide/get-set-up-for-amazon-ecs.html

Langkah 3: Mulai Amazon EC2 Container Service dengan membuat definisi tugas, menjadwalkan tugas, mengonfigurasi klaster di konsol ECS melalui tautan: http://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_GetStarted.html

Langkah 4: Matikan wadah dan instance EC 2 hostnya: http://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_CleaningUp.html

16. Magnum dalam Contoh 6.11 adalah proyek perangkat lunak yang baik untuk mewujudkan orkestrasi container dan pengelompokan host pada instance mesin OpenStack Nova. Periksa dengan situs web OpenStack untuk menindaklanjuti dengan rilis terbaru dari kode sumber Mag-num. Tulis laporan teknis singkat untuk meringkas temuan penelitian Anda.
17. Eucalyptus yang disajikan dalam Contoh 2.5 terus ditingkatkan untuk mendukung pengelolaan sumber daya cloud IaaS yang efisien. Periksa situs web Eucalyptus untuk menindaklanjuti perkembangan terbaru dan pengalaman porting yang dirilis oleh grup pengguna terdaftar mereka. Tulis laporan teknis singkat untuk meringkas temuan penelitian Anda.
18. Masalah ini meminta Anda untuk berlatih mengunggah foto seluler ke Amazon S3. Jelajahi beberapa alat SDK di AWS untuk menggunakan ponsel iOS atau ponsel Android apa pun untuk menyimpan foto di cloud Amazon S3 dan untuk memberi tahu pengguna AWS menggunakan layanan SNS. Laporkan fitur layanan penyimpanan/pemberitahuan, hasil pengujian, dan pengalaman aplikasi Anda. Periksa situs web untuk alat Android SDK. Sumber: http://aws.amazon.com/sdkforandroid/ Anda dapat menemukan alat SDK iOS dan Android dengan memeriksa /sdk-for-ios/ dan /sdk-for-android/ dengan cara yang sama: Ikuti tiga langkah berikut untuk menjalankan eksperimen:

Langkah 1: Unduh Amazon AWS SDK untuk Android (atau iOS) dari URL sumber.

Langkah 2: Periksa kode sampel yang diberikan di aws-android-sdk-1.6/samples/ S3 Uploader, yang membuat aplikasi sederhana yang memungkinkan pengguna mengunggah gambar dari ponsel ke bucket S3 di akun pengguna.

Langkah 3: Gambar-gambar ini dapat dilihat oleh siapa saja yang memiliki akses ke URL yang dibagikan oleh pengguna.

Anda perlu melakukan operasi berikut dan melaporkan hasilnya dalam snapshot atau menggunakan metrik kinerja apa pun yang Anda pilih untuk ditampilkan saat menggunakan ponsel Android. Demikian pula, untuk siswa yang menggunakan ponsel Apple iOS:

1) Coba unggah data (gambar) yang dipilih ke bucket AWS S3, menggunakan Access kunci dan kredensial Kunci Keamanan yang disediakan untuk pengguna. Ini akan memungkinkan Anda sebagai klien AWS.
2) Periksa apakah ember S3 ada dengan nama yang sama dan buat ember dan menempatkan gambar dalam ember S3.
3) Tampilkan di tombol Browser dan tampilkan gambar di browser.
4) Pastikan gambar diperlakukan sebagai file gambar di browser web.
5) Buat URL, untuk gambar di ember, sehingga dapat dibagikan dan dilihat oleh orang lain.
6) Mengomentari aplikasi yang diperluas di luar eksperimen ini.

19. Jelaskan perbedaan dalam dua skema pemulihan mesin berikut. Komentar tentang persyaratan implementasinya, kelebihan dan kekurangannya serta potensi aplikasinya:
    a) Pemulihan kegagalan mesin fisik oleh mesin fisik lain;
    b) Pemulihan kegagalan mesin virtual oleh mesin virtual lain;
    c) Sarankan metode untuk memulihkan VM dari mesin fisik yang gagal.
20. vSphere/6 yang tercakup dalam Contoh 2.6 adalah OS cloud solid yang tersedia secara komersial dari VMware. Gali dari laporan literatur tentang porting dan pengalaman aplikasi dan kinerja yang diukur oleh klien atau grup penggunanya. Tulis laporan teknis singkat untuk meringkas temuan penelitian Anda.

BAB 3
SISTEM PENGINDERAAN, SELULER, DAN KOGNITIF IOT

3.1 TEKNOLOGI PENGINDERAAN UNTUK INTERNET OF THINGS

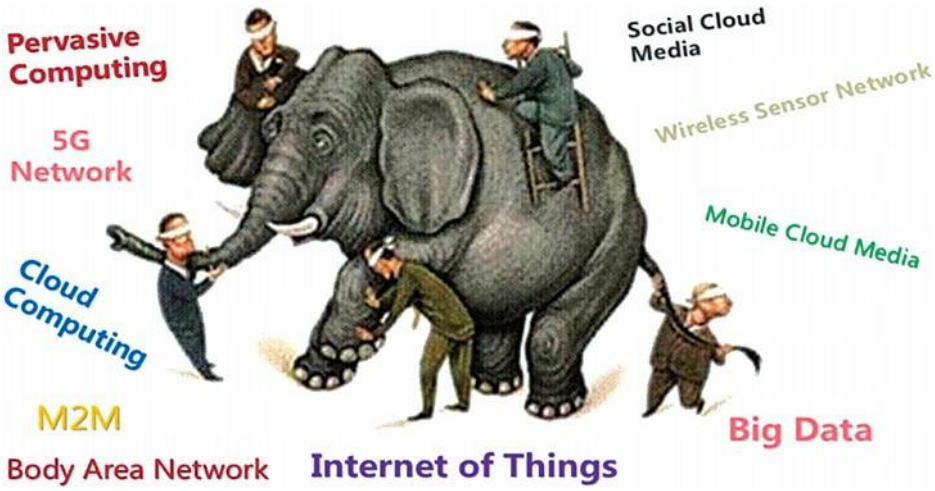
Mengintegrasikan dunia digital dan dunia fisik adalah tujuan akhir dari IoT. Ini bisa dianggap sebagai evolusi ketiga dari industri informasi. Pertama, skala jaringan menjadi sangat besar untuk menghubungkan banyak hal di dunia fisik. Kedua, mobilitas jaringan meningkat pesat karena meluasnya penggunaan perangkat seluler dan kendaraan. Ketiga, perpaduan jaringan heterogen menjadi lebih dalam dengan berbagai jenis perangkat yang terhubung ke Internet. Selain itu, Internet seluler, computing cloud, *Big data*, jaringan yang ditentukan perangkat lunak, dan 5G semuanya berdampak pada pengembangan IoT.

Mengaktifkan Teknologi dan Evolusi IoT

Pada Gambar 3.1, kami telah mengidentifikasi banyak teknologi yang memungkinkan pengembangan infrastruktur IoT untuk berbagai aplikasi. Teknologi pendukung dibagi menjadi dua kategori: i) teknologi yang memungkinkan membangun fondasi IoT. Di antara teknologi yang memungkinkan, pelacakan (RFID), jaringan sensor dan GPS sangat penting; ii) teknologi sinergis memainkan peran pendukung. Misalnya, biometrik dapat diterapkan secara luas untuk mempersonalisasi interaksi antara manusia dan mesin dan objek. Kecerdasan buatan, visi komputer, robotika, dan telepresence dapat membuat hidup kita lebih otomatis di masa depan.

Pada tahun 2005, konsep IoT menjadi pusat perhatian. IoT harus dirancang untuk menghubungkan objek dunia secara sensorik. Pendekatannya adalah untuk menandai sesuatu melalui RFID, merasakan sesuatu melalui sensor dan jaringan nirkabel, dan memikirkan sesuatu dengan membangun sistem tertanam yang berinteraksi dengan aktivitas manusia. IoT sekarang menjadi pendorong utama, tidak hanya di komunitas riset tetapi juga di industri besar seperti IBM dan Google. IoT benar-benar diaktifkan oleh banyak teknologi terkait. Untuk menyebutkan beberapa saja, computing pervasif, cloud media sosial, jaringan sensor nirkabel, computing cloud, *Big data*, komunikasi mesin ke mesin dan computing yang dapat dipakai, dll.

Pada tahun 2008, Dewan Intelijen Nasional AS menerbitkan sebuah laporan tentang “Teknologi Sipil yang Mengganggu”, yang juga mengidentifikasi IoT sebagai teknologi penting untuk kepentingan AS hingga tahun 2025. Secara kuantitatif, IoT harus dirancang untuk mengkodekan 50 hingga 100 triliun objek. Selain itu, IoT harus dirancang untuk mengikuti pergerakan objek-objek tersebut. Dengan lebih dari 6 miliar populasi manusia, berarti setiap orang dikelilingi oleh 1000 hingga 5000 objek setiap hari. Bayangkan bagaimana IoT dapat meningkatkan interaksi atau kenyamanan kita dengan segala sesuatu (objek) di sekitar kita.



Gambar 3.1 Teknologi yang memungkinkan dan sinergis IoT.

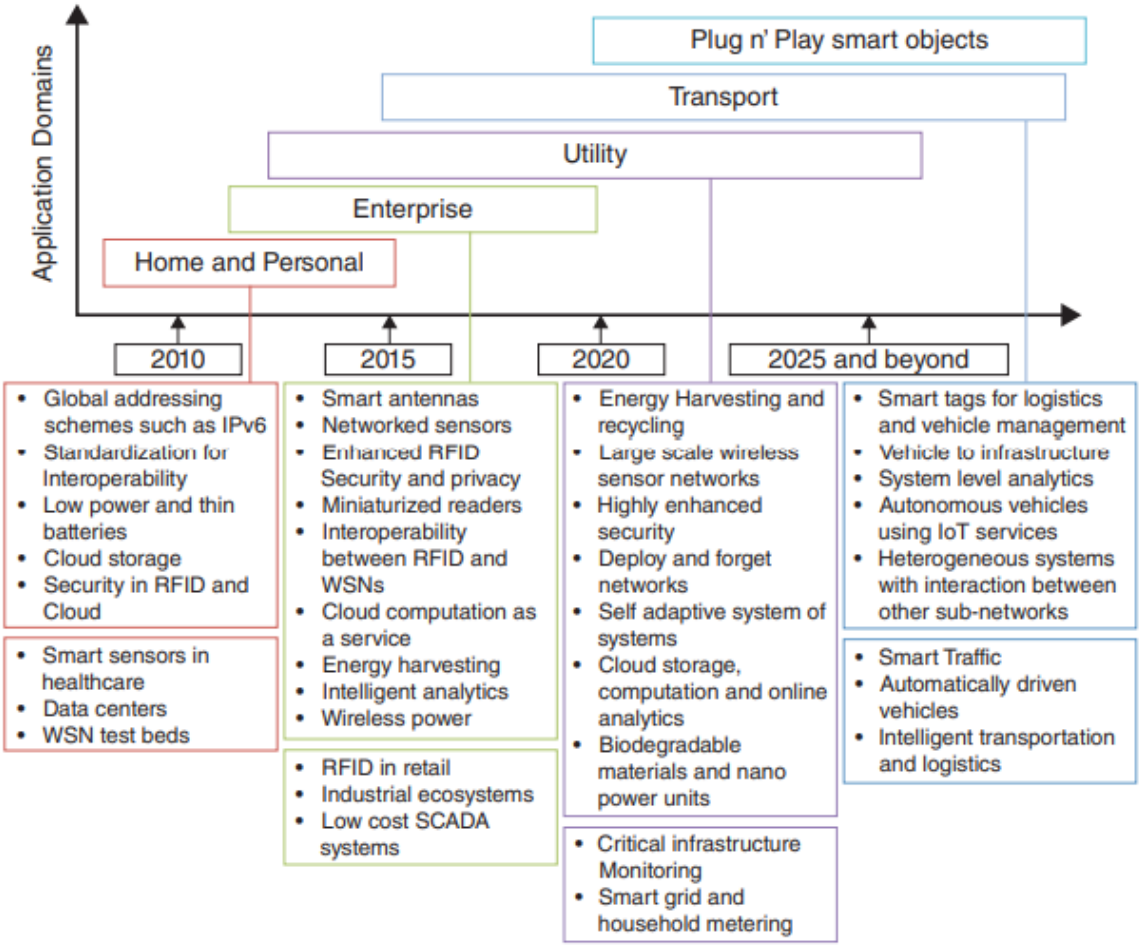
Mengaktifkan dan Teknologi Sinergis

Selama periode 25 tahun, pengembangan IoT bisa menjadi lebih matang dan lebih canggih. Misalnya, rantai pasokan bisa lebih disempurnakan dalam 10 tahun pertama (2000–2010). Aplikasi pasar vertikal mungkin merupakan gelombang kemajuan berikutnya. Pemosisian di mana-mana diharapkan menjadi kenyataan saat kita bergerak menuju tahun 2020. Di luar itu, web dunia fisik mungkin muncul untuk mencapai tujuan akhir IoT. Tujuannya adalah untuk mencapai peningkatan luar biasa dalam kemampuan manusia, hasil sosial, produktivitas bangsa, dan kualitas hidup secara umum.

Dengan jumlah perangkat seluler yang terus meningkat dan lalu lintas seluler yang eksplosif, jaringan 5G membutuhkan berbagai kemajuan teknologi untuk mentransmisikan lalu lintas secara lebih efektif sambil mengubah dunia dengan menghubungkan sejumlah besar perangkat seluler. Namun, perangkat seluler memiliki kemampuan komunikasi dan computing yang terbatas dalam hal daya computing, memori, penyimpanan, dan energi. Selain dukungan bandwidth pita lebar dari 5G, computing cloud perlu dimanfaatkan untuk memungkinkan perangkat seluler memperoleh sumber daya dinamis yang hampir tak terbatas untuk computing, penyimpanan, dan penyediaan layanan yang akan mengatasi kendala pada perangkat seluler pintar. Dengan demikian, kombinasi teknologi 5G dan computing cloud membuka jalan bagi aplikasi menarik lainnya.

Dengan dukungan *mobile cloud computing* (MCC), pengguna ponsel pada dasarnya memiliki satu opsi lagi untuk mengeksekusi computing aplikasinya, yaitu memindahkan computing ke cloud. Jadi, satu masalah utama adalah dalam kondisi apa pengguna seluler harus memindahkan computingnya ke cloud. Skenario pembongkaran computing di cloud jarak jauh mengharuskan pengguna untuk dilindungi oleh WiFi. Karena perangkat terminal di ujung pengguna memiliki sumber daya yang terbatas, yaitu perangkat keras, energi, bandwidth, dll., ponsel itu sendiri tidak dapat melakukan beberapa tugas computing intensif. Sebagai gantinya, data yang terkait dengan tugas computing dapat diturunkan ke cloud jarak jauh melalui WiFi atau saluran bandwidth tinggi lainnya.

Banyak tantangan IoT terbuka lebar, namun belum terpecahkan. Tantangan khusus mencakup privasi, penginderaan partisipatif, analisis data, visualisasi berbasis GIS (sistem informasi geografis), dan computing cloud. Area lain terkait dengan standarisasi arsitektur IoT, efisiensi energi, keamanan, protokol, dan Kualitas Layanan. Standarisasi pita frekuensi dan protokol memainkan peran penting dalam mencapai tujuan ini. Peta jalan pengembangan kunci untuk penelitian IoT dalam konteks aplikasi yang meluas ditunjukkan pada Gambar 3.2. Diagram ini menunjukkan pertumbuhan lima domain aplikasi IoT utama dari 2010 hingga 2025.



Gambar 3.2 Proyeksi peningkatan IoT di lima domain aplikasi IoT dari 2010 hingga 2015. (Courtesy of Gubbi et al. 2013. [10]) Direproduksi dengan izin Elsevier.

Pada awal 2000-an, IoT terutama diterapkan untuk mempercepat manajemen rantai pasokan. Pasar vertikal dan aplikasi pemosisian di mana-mana telah mendominasi aplikasi IoT sejak 2010. Akhirnya, kita akan melihat meluasnya penggunaan IoT di web dunia fisik, di mana teleoperasi dan telepresence akan memungkinkan pemantauan dan kontrol objek jarak jauh. Pada akhirnya, IoT akan memungkinkan pembuatan web dunia fisik yang terhubung ke segala sesuatu di Bumi. Ini akan membuat kehidupan sehari-hari kita nyaman dan mendapat informasi yang baik dengan seluruh dunia. Ini akan membuat keputusan yang lebih cerdas, menyelamatkan nyawa manusia, menghindari bencana dan mengurangi beban manusia secara signifikan. Di sisi

lain, munculnya IoT membawa beberapa dampak negatif. Misalnya, kami mungkin kehilangan privasi. Penjahat atau kekuatan musuh dapat menggunakan IoT untuk melakukan lebih banyak kehancuran. Sistem hukum perlu dibentuk untuk mencegah atau menghindari dampak negatif IoT ini.

Memperkenalkan Teknologi RFID dan Sensor

Bagian ini secara singkat memperkenalkan Radio Frequency Identification (RFID) dan sensor. Rincian lebih lanjut diberikan dalam Bagian 3.3 dan 3.4. Dengan kemajuan pesat dalam elektronik, elektromekanik dan nanoteknologi, perangkat di mana-mana tumbuh pesat dalam kuantitas dan lebih kecil dalam ukuran fisik. Benda-benda ini disebut sebagai "benda", seperti komputer, sensor, orang, aktuator, lemari es, TV, kendaraan, ponsel, pakaian, makanan, obat-obatan, buku, paspor, koper, dll. peserta aktif dalam bisnis, informasi dan proses sosial. Para peserta ini dapat bereaksi secara mandiri di dunia fisik. Mereka mempengaruhi atau memicu tindakan dan menciptakan layanan dengan atau tanpa campur tangan manusia secara langsung. Ada banyak perangkat sensor untuk penginderaan dan pengumpulan data. Setiap node senor dapat menggabungkan fungsi penginderaan, komunikasi dan pemrosesan lokal.

Teknologi RFID

Langkah pertama untuk mengaktifkan layanan cerdas adalah mengumpulkan informasi kontekstual tentang lingkungan, "sesuatu" dan objek yang menarik. Misalnya, sensor dapat digunakan untuk terus memantau aktivitas dan tindakan fisiologis manusia seperti status kesehatan dan pola gerak. Teknologi RFID dapat digunakan untuk mengumpulkan informasi pribadi yang penting dan menyimpannya pada chip murah yang melekat pada individu setiap saat. RFID adalah teknologi elektronik frekuensi radio (RF) yang memungkinkan identifikasi otomatis atau lokasi objek, orang, dan hewan dalam berbagai pengaturan penyebaran. Dalam dekade terakhir, sistem RFID telah dimasukkan ke dalam berbagai sistem industri dan komersial, termasuk manufaktur dan logistik, ritel, pelacakan dan penelusuran barang, pemantauan inventaris, manajemen aset, anti-pencurian, pembayaran elektronik, anti-gangguan, transportasi. tiket, manajemen rantai pasokan, dll.

Aplikasi RFID tipikal terdiri dari tag RFID, pembaca RFID, dan sistem backend. Dengan chip RF sederhana dan antena, tag RFID dapat menyimpan informasi yang mengidentifikasi objek yang dilampirkan. Ada tiga jenis tag RFID, yaitu tag pasif, tag aktif, dan tag semi aktif. Tag pasif memperoleh energi melalui sinyal RF dari pembaca, sementara tag aktif ditenagai oleh baterai tertanam, yang memungkinkan memori lebih besar atau fungsionalitas lebih. Meskipun tag semi-aktif berkomunikasi dengan pembaca RFID seperti tag pasif, modul tambahan dapat didukung melalui baterai internal. Ketika datang dalam jarak dekat pembaca RFID, informasi yang disimpan dalam tag ditransfer ke pembaca, dan ke sistem backend, yang dapat berupa komputer yang digunakan untuk memproses informasi ini dan mengendalikan operasi sub-sistem lain.).

Sensor dan Jaringan Sensor

Dalam dekade terakhir, kami telah menyaksikan minat yang tumbuh dalam menyebarkan sejumlah besar sensor mikro yang berkolaborasi secara terdistribusi dalam pengumpulan dan pemrosesan data. Node sensor diharapkan tidak mahal dan dapat digunakan di berbagai

lingkungan. Jaringan sensor nirkabel (WSN) terdiri dari sensor otonom yang terdistribusi secara spasial untuk memantau kondisi fisik atau lingkungan, seperti suhu, suara, tekanan, dll. dan untuk secara kooperatif meneruskan datanya melalui jaringan ke lokasi utama. Node sensor membentuk jaringan nirkabel ad hoc multi-hop. Perbedaan terbesar antara WSN dan jaringan seluler adalah bahwa WSN tidak memerlukan stasiun pangkalan dan setiap node sensor berfungsi sebagai pemancar dan penerima. Karena sumber daya yang terbatas pada node sensor, perutean di WSN adalah tugas yang menantang, sambil meminimalkan konsumsi energi selama penyebaran data.

Jaringan Sensor Nirkabel

WSN adalah sekelompok transduser khusus dengan infrastruktur komunikasi yang dimaksudkan untuk memantau dan merekam kondisi di berbagai lokasi. Parameter yang biasanya dipantau adalah suhu, kelembapan, tekanan, arah dan kecepatan angin, intensitas penerangan, intensitas getaran, intensitas suara, tegangan saluran listrik, konsentrasi bahan kimia, tingkat polutan, dan fungsi vital tubuh. Jaringan sensor terdiri dari beberapa stasiun deteksi yang disebut node sensor, yang masing-masing berukuran kecil, ringan, dan portabel. Setiap node sensor dilengkapi dengan transduser, mikrokomputer, transceiver, dan sumber daya. Transduser menghasilkan sinyal listrik berdasarkan data yang dirasakan.

Prosesor sensor menangani sinyal input dan menyimpan atau mengirimkan output. Transceiver dapat berupa kabel atau nirkabel. Daya untuk setiap node sensor berasal dari utilitas listrik atau dari baterai. Node sensor dapat bervariasi dalam ukuran dari kotak sepatu hingga ukuran butiran debu. Biaya node sensor juga sangat bervariasi, mulai dari ratusan dolar hingga beberapa sen, tergantung pada ukuran jaringan sensor dan kompleksitas yang diperlukan dari masing-masing node sensor. Batasan ukuran dan biaya pada node sensor seringkali ditentukan oleh energi, memori, kecepatan computing dan bandwidth dari sensor yang digunakan.

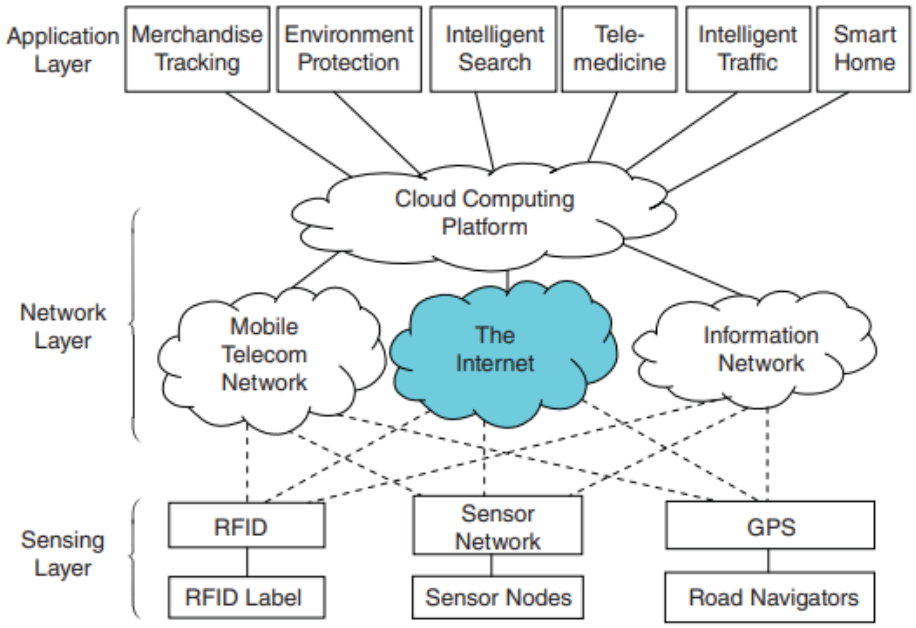
Teknologi sensor yang banyak digunakan adalah perangkat Zigbee yang ditentukan dalam Standar IEEE 802.15.4. Frekuensi radio yang diterapkan di Zigbee menghasilkan kecepatan data yang rendah, masa pakai baterai yang lama, dan jaringan yang aman. Mereka digunakan terutama di IoT monitor dan remote control atau aplikasi seluler. Banyak supermarket, department store, dan rumah sakit dipasangi jaringan Zigbee. Kecepatan data berkisar antara 20-250 Kbps. Mereka dapat beroperasi hingga 100 meter. Namun, perangkat Zigbee dapat terhubung ke jaringan bersama untuk mencakup area yang jauh lebih luas. Jaringan Zigbee sangat skalabel. Jaringan Zigbee digunakan dalam jaringan area rumah nirkabel (WHAN). Teknologi ini lebih sederhana untuk digunakan dan lebih murah daripada Bluetooth atau WiFi.

Dukungan Arsitektur dan Nirkabel IoT

Arsitektur dasar IoT diperkenalkan di bawah ini dalam tiga lapisan, yaitu lapisan penginderaan, jaringan, dan aplikasi. Sistem IoT cenderung memiliki arsitektur yang digerakkan oleh peristiwa. Pada Gambar 3.3, pengembangan IoT ditunjukkan dengan arsitektur tiga lapis. Lapisan atas dibentuk oleh aplikasi yang digerakkan. Aplikasi IoT untuk perawatan kesehatan akan disajikan di Bab 8. Lapisan bawah terdiri dari berbagai jenis penginderaan dan perangkat generasi informasi otomatis: yaitu sensor, perangkat ZigBee, tag RFID dan navigator GPS

pemetaan jalan, dll. Perangkat penginderaan lokal atau area luas yang terhubung dalam bentuk jaringan sensor, jaringan RFID dan sistem GPS, dll. Sinyal atau informasi yang dikumpulkan pada perangkat penginderaan ini dihubungkan ke aplikasi melalui platform computing cloud di lapisan tengah.

Cloud pemrosesan sinyal dibangun di atas jaringan seluler, tulang punggung Internet, dan berbagai jaringan informasi di lapisan tengah. Dalam IoT, makna peristiwa penginderaan tidak mengikuti model deterministik atau sintaksis. Bahkan, model arsitektur berorientasi layanan (SoA) dapat diadopsi di sini. Sejumlah besar sensor dan filter digunakan untuk mengumpulkan data mentah. Berbagai computing dan penyimpanan cloud dan grid digunakan untuk memproses data dan mengubahnya menjadi format informasi dan pengetahuan. Data sensorik digunakan untuk menyusun sistem pengambilan keputusan untuk aplikasi intelijen. Lapisan tengah juga dianggap sebagai web atau kisi semantik. Beberapa aktor (layanan, komponen, avatar) direferensikan sendiri.



Gambar 3.3 Arsitektur Internet of Things (IoT) dan teknologi yang mendasarinya.

3.2 INTERAKSI IOT DENGAN GPS, CLOUD, DAN MESIN CERDAS

Ini mencakup persyaratan jaringan, termasuk jaringan inti nirkabel, kabel, dan seluler. Kami akan memeriksa sistem penentuan posisi lokal dan global serta jaringan akses radio berbasis cloud yang memungkinkan untuk sistem seluler 5G. Terakhir, kita akan mempelajari empat kerangka kerja untuk interaksi IoT dengan seluruh dunia.

Teknologi Pemosisian Lokal versus Global

Persyaratan untuk mengintegrasikan dunia cyber dan dunia fisik lebih tinggi dari sebelumnya. Lokalisasi menjadi jembatan untuk menghubungkan dua dunia ini. Mengingat WSN sebagai contoh, bersama dengan informasi lokasi, data sensorik menjadi bermakna. Ada proyek WSN nyata bernama ZebraNet, untuk penggunaan ahli biologi yang ingin melacak dan

mempelajari hewan. Namun, tanpa lokasi, hewan tidak dapat dilacak, sehingga tidak memungkinkan penelitian lebih lanjut. Sebagai contoh lain dari computing di mana-mana, lokasi sangat penting untuk membedakan berbagai skenario untuk menyediakan layanan yang dipersonalisasi bagi pengguna.

Menurut kemampuan perangkat keras yang beragam, kami mengklasifikasikan teknik pengukuran ke dalam enam kategori (dari halus hingga kasar): lokasi, jarak, sudut, area, jumlah hop, dan lingkungan. Di antara mereka, pengukuran fisik yang paling kuat adalah langsung mendapatkan posisi tanpa perhitungan lebih lanjut. GPS adalah infrastruktur semacam itu. Kami membahas lima pengukuran lainnya dalam bab ini, dengan penekanan pada prinsip-prinsip dasar teknik pengukuran. Pada dasarnya, informasi terkait jarak dapat diperoleh dengan kekuatan sinyal radio atau waktu propagasi radio, informasi sudut dengan susunan antena, dan informasi area, jumlah hop dan lingkungan dengan fakta bahwa radio hanya ada untuk node di sekitarnya.

Teknologi Pemosisian Lokal

Salah satu metode untuk menentukan lokasi perangkat adalah melalui konfigurasi manual, yang seringkali tidak layak untuk penerapan skala besar atau sistem seluler. Sebagai sistem yang populer, GPS tidak cocok untuk lingkungan dalam ruangan atau bawah tanah dan membutuhkan biaya perangkat keras yang tinggi. Sistem pemosisian lokal bergantung pada stasiun pangkalan kepadatan tinggi yang digunakan, beban yang mahal untuk sebagian besar jaringan ad hoc nirkabel dengan sumber daya terbatas.

Keterbatasan sistem penentuan posisi yang ada memotivasi skema baru lokalisasi jaringan, di mana beberapa node khusus (alias jangkar atau beacon) mengetahui lokasi global mereka dan sisanya menentukan lokasi mereka dengan mengukur informasi geografis dari node tetangga lokal mereka. Skema lokalisasi seperti itu untuk jaringan multihop nirkabel secara alternatif digambarkan sebagai "kooperatif", "ad hoc", "lokalisasi dalam jaringan" atau "lokalisasi mandiri".

Dengan demikian, node jaringan secara kooperatif menentukan lokasi mereka dengan berbagi informasi. Istilah node "diketahui" dan "tidak diketahui" mengacu pada node yang sadar dan tidak menyadari lokasinya masing-masing. Misalkan proses penentuan posisi tertentu adalah salah satu di mana node yang tidak diketahui menentukan lokasinya berdasarkan informasi yang diberikan oleh sejumlah node yang dikenal. Node yang tidak diketahui juga dikenal sebagai node target atau node yang akan ditempatkan, sedangkan node yang dikenal sebagai node referensi.
Solusi pelokalan terdiri dari dua tahap dasar: i) mengukur informasi geografis dari kebenaran dasar penyebaran jaringan; dan ii) menghitung lokasi node sesuai dengan data yang diukur. Informasi geografis mencakup berbagai hubungan geometris dari kesadaran tetangga berbutir kasar hingga rentang ruas berbutir halus (misalnya jarak atau sudut). Berdasarkan pengukuran fisik, algoritme lokalisasi memecahkan masalah: bagaimana menyebarkan informasi lokasi dari node suar melalui jangkauan jaringan yang luas.

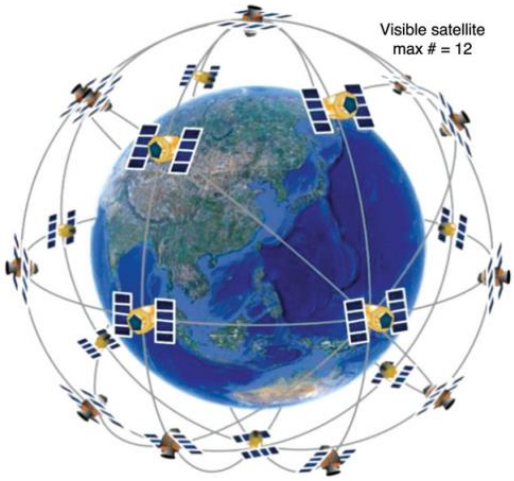
Umumnya, desain algoritme lokalisasi sangat bergantung pada berbagai faktor, termasuk ketersediaan sumber daya, persyaratan akurasi, dan pembatasan penerapan, dan tidak ada algoritme tertentu yang menjadi favorit mutlak di seluruh spektrum. Karena keterbatasan

perangkat keras, jangkauan tidak selalu tersedia untuk perangkat nirkabel. Dalam situasi seperti itu, pendekatan bebas jangkauan adalah alternatif hemat biaya, di mana node hanya mengetahui tetangga mereka.

Tanpa rentang jarak langsung, jarak fisik dari sepasang node diperkirakan dengan jumlah hop atau kedekatan. Ide dasar dari lokalisasi berbasis jumlah hop adalah dengan menggunakan pengiriman pesan hop-by-hop untuk menghitung jumlah hop dari node ke jangkar. Informasi hop-count selanjutnya dikonversi ke perkiraan jarak. Akhirnya, setiap node mengadopsi trilateration atau metode lain untuk menentukan lokasinya sesuai dengan perkiraan jarak. Kemungkinan lain adalah untuk mengeksplorasi kedekatan relatif dari node. Ketika jarak mulai tidak tersedia, fakta bahwa satu node lebih dekat ke beberapa node lain dapat membantu proses lokalisasi.

Teknologi Satelit untuk Penentuan Posisi Global

Penentuan posisi global dilakukan dengan beberapa satelit yang ditempatkan di luar angkasa.



Gambar 3.4 Arsitektur GPS 24-satelit: satelit mengelilingi Bumi dua kali sehari dalam beberapa lapisan orbit tetap tanpa gangguan satu sama lain.

Penyebaran satelit ditunjukkan pada Gambar 3.4. Setiap satelit terus mengirimkan pesan yang mencakup waktu transmisi dan posisi satelit. Sebuah penerima GPS menghitung posisinya dengan tepat waktu sinyal yang dikirim oleh satelit. Penerima menggunakan pesan yang diterimanya untuk menentukan waktu transit dan menghitung jarak ke setiap satelit menggunakan kecepatan cahaya. Masing-masing jarak dan lokasi satelit menentukan bola sinyal. Penerima terletak di bidang sinyal persimpangan dari beberapa satelit.

Contoh 3.1 Sistem GPS Dikembangkan di AS

GPS AS dibangun dengan tiga segmen. Segmen antariksa adalah satelit yang beredar di luar angkasa. Segmen pengguna mencakup objek bergerak atau tidak bergerak seperti pesawat terbang, kapal laut, dan kendaraan bergerak di permukaan bumi. Segmen kontrol mencakup beberapa antena bumi dan stasiun master dan monitor di permukaan bumi yang tersebar secara global. Tipe data uplink dan downlink berbeda. Waktu tempuh sinyal yang dihitung digunakan

untuk menampilkan lokasi penerima. Sejumlah aplikasi untuk GPS memang menggunakan pengaturan waktu yang murah dan sangat akurat ini, termasuk transfer waktu, pengaturan waktu sinyal lalu lintas, dan sinkronisasi dengan BTS ponsel.

Angkatan Udara AS mengembangkan, memelihara, dan mengoperasikan segmen ruang dan kendali dari sistem GPS. Ada 24 satelit yang ditempatkan di sekitar Bumi (Gambar 3.4) dalam orbit tetap. Satelit mengorbit pada ketinggian sekitar 20.200 km. Satelit GPS memancarkan sinyal dari luar angkasa, dimana setiap penerima GPS menghitung lokasi tiga dimensinya (lintang, bujur dan ketinggian) ditambah waktu saat ini. Segmen luar angkasa terdiri dari 24 satelit di orbit Bumi sedang dan juga termasuk booster yang diperlukan untuk meluncurkannya ke orbit.

Satelit GPS mengelilingi Bumi dua kali sehari dalam orbit yang tepat dan mengirimkan sinyal ke Bumi. Perangkat GPS di darat menerima sinyal ini dan menggunakan triangulasi untuk menghitung lokasi persis pengguna. Dalam kasus umum, empat satelit diperlukan untuk menemukan satu titik di permukaan bumi. Sistem ini awalnya dikembangkan untuk penggunaan militer pada tahun 1975. Sekarang sistem, di bawah peraturan ketat, terbuka untuk penggunaan sipil dan komersial, terutama dalam pelacakan kendaraan dan aplikasi navigasi.

Aplikasi IoT Mandiri versus Cloud-Centric

Biasanya, IoT mandiri berfokus pada lingkungan yang stabil di mana aplikasi baru kemungkinan akan meningkatkan kualitas hidup kita: di rumah, saat bepergian, saat sakit, di tempat kerja, saat jogging, dan di gym, hanya untuk menyebutkan beberapa. Lingkungan ini sekarang dilengkapi dengan objek dengan hanya kecerdasan primitif, sebagian besar tanpa kemampuan komunikasi apa pun. Memberikan objek-objek ini kemampuan untuk berkomunikasi satu sama lain dan untuk menguraikan informasi yang dirasakan dari lingkungan mereka menyiratkan memiliki lingkungan yang berbeda di mana berbagai aplikasi yang sangat luas dapat digunakan. Ini dapat dikelompokkan ke dalam domain berikut.

Domain transportasi dan logistik, domain perawatan kesehatan, domain lingkungan cerdas (rumah, kantor, pabrik), domain pribadi dan sosial; di antara aplikasi yang mungkin, kita dapat membedakan antara yang dapat diterapkan secara langsung atau lebih dekat dengan kebiasaan hidup kita saat ini dan yang futuristik, yang hanya dapat kita bayangkan saat ini, karena teknologi dan/atau masyarakat kita belum siap untuk penerapannya. Dalam subbagian berikut kami memberikan ulasan tentang aplikasi jangka pendek-menengah untuk masing-masing kategori ini dan berbagai aplikasi futuristik.

Contoh 3.2 Smart Power Grid yang didukung oleh Internet of Things

Jaringan pintar mencakup sistem pemantauan cerdas yang melacak semua aliran listrik dalam sistem. Meter dan sensor pintar, peningkatan digital pengukur utilitas saat ini, melacak penggunaan energi secara real time sehingga pelanggan dan perusahaan utilitas mengetahui berapa banyak yang digunakan pada waktu tertentu. Energi dibayar untuk menggunakan harga "waktu", yang berarti listrik akan lebih mahal pada waktu puncak penggunaan.

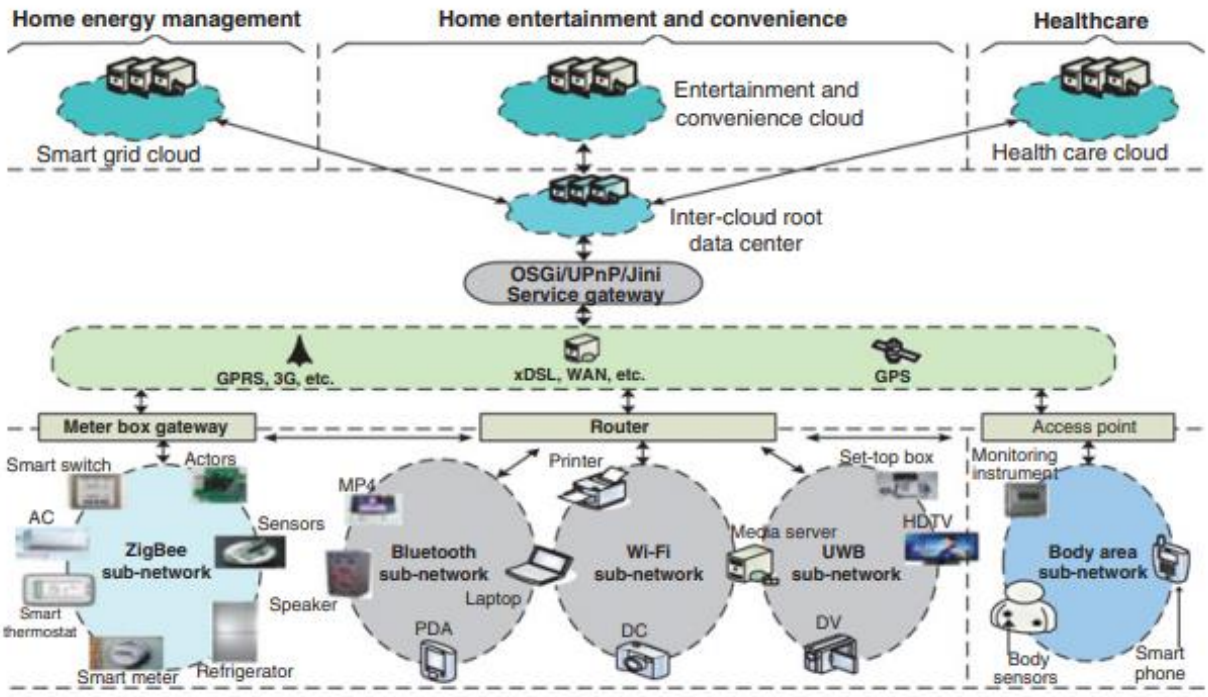
Misalnya, ketika daya paling murah, pengguna dapat mengizinkan smart grid untuk menyalakan peralatan rumah tangga tertentu seperti mesin cuci atau proses pabrik yang dapat berjalan pada jam yang berubah-ubah. Pada waktu puncak, alat ini dapat mematikan peralatan

tertentu untuk mengurangi permintaan. Pengguna yang lebih terlibat akan dapat menggunakan meteran pintar untuk melihat penggunaan energi dari jarak jauh dan membuat keputusan waktu nyata tentang konsumsi energi. Kulkas atau sistem pendingin udara dapat dimatikan dari jarak jauh saat penghuni pergi.

Dengan pengembangan WSN, serta sistem tertanam berdaya rendah dan computing cloud, sistem IoT yang dibantu cloud secara bertahap matang untuk mendukung aplikasi IoT yang cerdas dan intensif computing yang melibatkan sejumlah besar data. Aplikasi IoT rumahan dapat ditingkatkan oleh lingkungan computing cloud yang muncul. Kerangka kerja berbantuan cloud yang terukur dan elastis menggeser computing dan penyimpanan ke dalam jaringan untuk mengurangi biaya operasional dan pemeliharaan.

Aplikasi Sistem IoT Cloud-Centric

Informasi yang disampaikan dari domain yang berbeda (misalnya smart grid dan layanan kesehatan) sulit dipahami dan ditangani oleh komputer di layanan cloud. Dengan dukungan model semantik, pendekatan berbasis ontologi dapat digunakan untuk mengimplementasikan interaksi dan berbagi informasi di IoT rumahan yang dibantu cloud. Seperti yang ditunjukkan pada Gambar 3.5, sistem cloud terpisah dapat saling beroperasi, dengan root cloud tambahan yang menyediakan layanan berbeda untuk perawatan kesehatan, manajemen energi, kenyamanan dan hiburan, dll.



Gambar 3.5 Sistem IoT Cloud-centric untuk lingkungan rumah pintar.

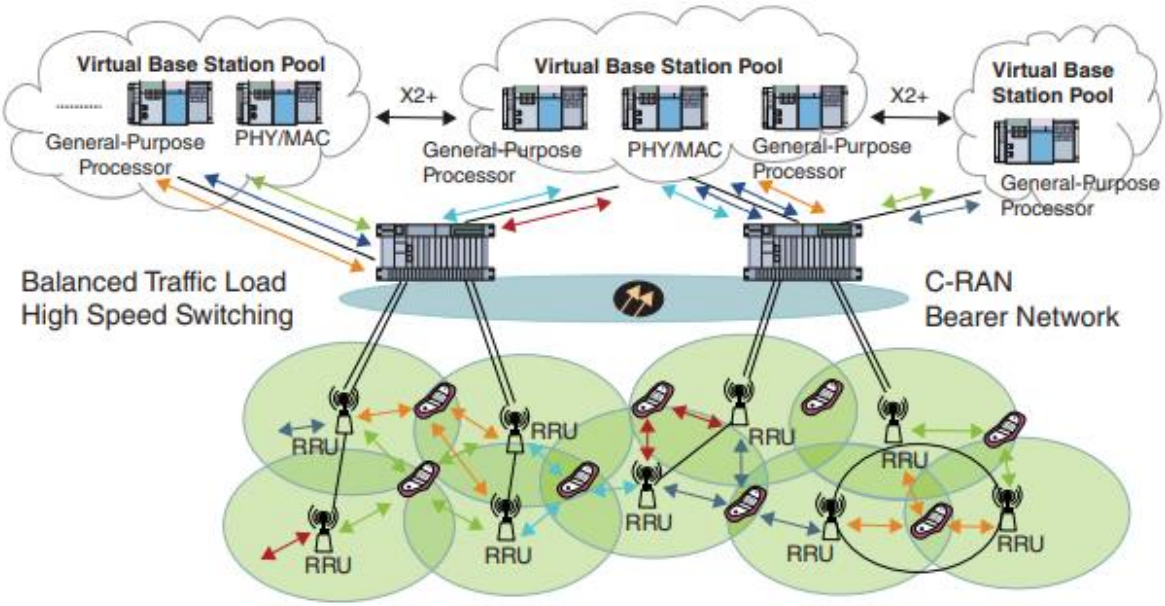
Gateway layanan mengimplementasikan berbagai teknologi, protokol, standar, dan layanan untuk mendiversifikasi kemampuan komunikasi dan mengintegrasikan perangkat. Saat ini, sebagian besar gateway layanan menerapkan mode dan sistem perangkat lunak yang

terdefinisi dengan baik, seperti Jini, UPnP, dan OSGi. Selain itu, komunikasi objek heterogen di IoT adalah masalah utama, karena objek yang berbeda memberikan informasi yang berbeda dalam format yang berbeda untuk tujuan yang berbeda. Teknologi dan model web semantik juga dapat digunakan untuk membantu memecahkan masalah ini. Teknologi web semantik dapat diterapkan untuk memfasilitasi komunikasi dalam aplikasi IoT rumahan.

Dalam beberapa tahun terakhir, computing cloud telah memberikan perspektif baru dalam teknologi yang dibantu cloud untuk tujuan yang berbeda. Sistem komunikasi berbantuan cloud dapat mencakup beberapa sistem cloud yang beroperasi dengan kebijakan berbeda untuk berbagi sumber daya, sehingga QoS end-to-end kepada pengguna dapat dipertahankan, bahkan jika terjadi fluktuasi besar dalam beban computing yang tidak dapat ditangani oleh sistem cloud tunggal. Diketahui bahwa arsitektur sebelumnya untuk IoT belum memperhitungkan kemampuan yang dibantu cloud ini. Dalam pandangan kami, ini adalah faktor penting bagi IoT untuk mencapai kelengkapan fungsionalitas. Oleh karena itu, dibandingkan dengan literatur survei sebelumnya, kami mengusulkan lapisan yang dibantu cloud untuk kemajuan arsitektur IoT. Contoh berikut menunjukkan upaya bersama antara Intel dan China Mobile menuju pengembangan jaringan inti seluler 5G.

Contoh 3.3 Jaringan Akses Radio Berbasis Cloud (C-RAN) untuk Sistem Seluler 5G

Sejumlah besar BTS digunakan dalam jaringan inti seluler 3G atau 4G saat ini. Mereka menghadapi serangkaian masalah: yaitu ukuran fisik yang besar, kecepatan data yang lambat, kehilangan udara selama serah terima antar sel, dan membutuhkan daya yang cukup besar agar tetap berjalan lancar tanpa gangguan. C-RAN adalah proyek bersama antara Intel dan China Mobile menuju solusi efisien untuk masalah ini. Idenya diilustrasikan pada Gambar 3.6. Detail arsitektur C-RAN ini dapat ditemukan di kertas putih oleh Chen dan Ran, 2011.



Gambar 3.6 Arsitektur konseptual jaringan akses radio berbasis cloud (C-RAN). (Courtesy of China Mobile Research Institute, 2009.)

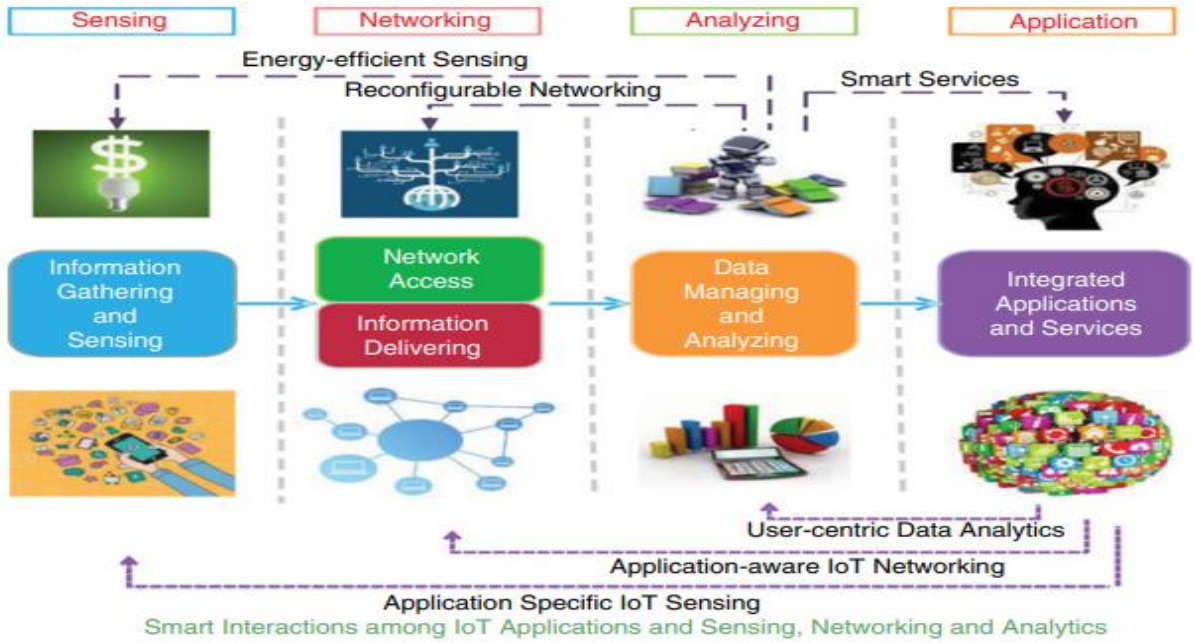
Menara antena besar yang digunakan di stasiun berbasis konvensional digantikan oleh sejumlah besar kepala radio jarak jauh kecil (RRH), yang beroperasi dengan daya kecil (bahkan energi matahari dapat melakukan pekerjaan itu) dan mudah didistribusikan dengan kepadatan tinggi di area pengguna yang padat. . Kontrol dan pemrosesan di stasiun berbasis fisik diganti dengan menggunakan kumpulan stasiun pangkalan virtual (VBS) yang ditempatkan dalam hierarki pusat switching berbasis cloud. Beban lalu lintas yang seimbang antara RRH dan kumpulan VBS diaktifkan dengan menggunakan jaringan transportasi optik berkecepatan tinggi dan sakelar dengan kabel serat dan tautan gelombang mikro.

Keuntungan menggunakan C-RAN diringkas dalam empat aspek: i) kumpulan sumber daya pemrosesan terpusat dapat mendukung 10-1000 sel dengan efisiensi tinggi; ii) radio kooperatif digunakan dalam penjadwalan dan pemrosesan gabungan multi-sel, yang memecahkan masalah kehilangan udara dan serah terima; iii) C-RAN menawarkan layanan real-time dengan menargetkan untuk membuka platform TI, konsolidasi sumber daya dan operasi dan migrasi multi-standar yang fleksibel; dan iv) telekomunikasi seluler yang hijau dan bersih diwujudkan dengan asumsi daya yang jauh lebih sedikit, biaya operasional yang lebih rendah, dan peluncuran sistem yang cepat. Banyak perusahaan lain juga membangun sistem C-RAN serupa, termasuk CISCO dan Korean Telecommunication.

Kerangka Kerja Interaksi IoT dengan Lingkungan

Seperti halnya jenis "jaringan" baru, IoT tidak hanya menghubungkan terminal jaringan seperti ponsel, komputer, dan perangkat pintar, tetapi juga objek kehidupan sehari-hari yang sampai sekarang bagi kita hanyalah "benda yang tidak memiliki jaringan" atau "objek lembam" . Kami pertama menjelaskan arsitektur berbasis lapisan untuk IoT, seperti yang ditunjukkan pada Gambar 3.7.

- **Penginderaan objek dan pengumpulan informasi:** Langkah pertama untuk mengaktifkan layanan cerdas adalah mengumpulkan informasi kontekstual tentang lingkungan, “benda”, dan objek yang menarik. Misalnya, sensor dapat digunakan untuk terus memantau aktivitas dan tindakan fisiologis manusia seperti status kesehatan dan pola gerak. Teknik RFID dapat digunakan untuk mengumpulkan informasi pribadi yang penting dan menyimpannya pada chip murah yang melekat pada individu setiap saat.
- **Penyampaian informasi:** Berbagai teknologi nirkabel dapat digunakan untuk menyampaikan informasi, seperti jaringan sensor nirkabel (WSN), jaringan area tubuh (BAN), WiFi, Bluetooth, Zigbee, GPRS, GSM, seluler dan 3G, dll. Komunikasi yang beragam tersebut teknik tion dapat menampung lebih banyak aplikasi ke dalam sistem.
- **Pemrosesan informasi untuk layanan cerdas:** Mesin di mana-mana harus memproses informasi dengan cara "otonom" dan "pintar" untuk menyediakan layanan yang luas dan otonom. Misalnya, informasi yang tidak berarti dapat disaring sesuai dengan minat pengguna di jejaring sosial.



Gambar 3.7 Interaksi antara penginderaan IoT, pemantauan seluler, dan analitik cloud.

Penginderaan IoT dapat berinteraksi dengan banyak sistem cyber lainnya, seperti yang diilustrasikan pada Gambar 3.7. Misalnya, pengguna dapat meminta kinerja analitik yang dipersonalisasi berdasarkan data tertentu. Profil data penginderaan dan strategi jaringan dapat disesuaikan berdasarkan persyaratan khusus aplikasi. Berdasarkan kecerdasan yang diperoleh dari analisis data, penginderaan hemat energi dapat dicapai melalui interaksi antara lapisan penginderaan dan lapisan analisis; analitik data juga dapat bermanfaat bagi layanan cerdas. Adapun lapisan jaringan, fungsi manajemen (misalnya virtualisasi fungsi jaringan bersama-sama dengan jaringan yang ditentukan perangkat lunak) yang diwujudkan dalam lapisan analisis juga berpotensi membantu operator memenuhi perjanjian tingkat layanan yang ketat, memantau dan memanipulasi lalu lintas jaringan secara akurat, dan meminimalkan biaya operasi.

Tabel 3.1 Persyaratan empat kerangka kerja computing dan komunikasi IoT.

Kerangka	**WSN**	**M2M**	**BAN**	**CPS**
Persyaratan Penginderaan	XXXX	XX	XXX	XXX
Permintaan Jaringan	XX	XXXX	XX	XXXX
Menganalisis Kompleksitas	XX	XX	XXX	XXXX
Industrialisasi Aplikasi	XXXX	XXX	XX	X
Permintaan Keamanan	X	XX	XXX	XXXX

Melalui antarmuka dengan WSN, berbagai informasi dapat dikumpulkan oleh sensor untuk sistem M2M. Jadi, selain komunikasi M2M, mesin juga dapat bertindak melalui informasi yang dikumpulkan dengan berintegrasi dengan WSN. Dengan kemampuan pengambilan keputusan dan kontrol otonom, sistem M2M dapat ditingkatkan ke CPS. Dengan demikian, CPS

merupakan evolusi dari M2M dengan pengenalan operasi yang lebih cerdas dan interaktif, di bawah arsitektur IoT. Berfokus pada berbagai jenis aplikasi, IoT memiliki inkarnasi yang berbeda seperti WSN, M2M, BAN dan CPS.

Pada Tabel 3.1, kami menandai dari satu X hingga empat X (yaitu XXXX) untuk menunjukkan permintaan kerangka kerja IoT yang berbeda pada fitur relevan yang tercantum dalam judul baris. Lebih banyak bintang mengacu pada permintaan yang lebih tinggi dari fitur tertentu di bawah kerangka kolom. Aplikasi CPS memiliki potensi untuk mendapatkan keuntungan dari jaringan nirkabel besar dan perangkat pintar, yang memungkinkan aplikasi CPS menyediakan layanan cerdas berdasarkan pengetahuan dari dunia fisik di sekitarnya. Kami mengamati bahwa WSN adalah skenario paling dasar dari IoT. Ini dianggap sebagai suplemen M2M sebagai dasar CPS. CPS berevolusi dari M2M dalam pemrosesan informasi cerdas.

Berikut ini, kami menentukan empat kerangka kerja nirkabel untuk penyebaran aplikasi IoT. Ini muncul sebagai WSN, M2M, BAN dan CPS, seperti yang dijelaskan di bawah ini:

- **Wireless Sensor Network (WSN):** Ini terdiri dari sensor otonom yang didistribusikan secara spasial untuk memantau kondisi fisik atau lingkungan, dan untuk secara kooperatif meneruskan data mereka melalui jaringan ke lokasi utama. WSN, menekankan persepsi informasi melalui semua jenis node sensor, adalah skenario paling dasar dari IoT.
- **Komunikasi Mesin ke Mesin (M2M):** Biasanya, M2M mengacu pada komunikasi data tanpa atau dengan campur tangan manusia yang terbatas, di antara berbagai perangkat terminal seperti komputer, prosesor tertanam, sensor/aktuator pintar, dan perangkat seluler, dll. Komunikasi M2M didasarkan pada tiga pengamatan: i) mesin jaringan lebih berharga daripada yang terisolasi; ii) ketika beberapa mesin saling berhubungan, aplikasi yang lebih otonom dapat dicapai; dan iii) layanan cerdas dan ada di mana-mana dapat diaktifkan oleh perangkat tipe mesin yang berkomunikasi secara cerdas dengan perangkat lain kapan saja dan di mana saja.
- **Body-Area Network (BAN):** Jenis arsitektur jaringan baru yang diwarisi dari jaringan sensor dengan menggunakan kemajuan baru dalam sensor wearable pemantauan yang ringan, berukuran kecil, berdaya sangat rendah, dan cerdas, yang terus memantau fisiologis manusia. aktivitas dan tindakan, seperti status kesehatan dan pola gerak.
- **Cyber Physical System (CPS):** Ini adalah sistem kolaborasi elemen computing yang mengendalikan entitas fisik.

3.3 IDENTIFIKASI FREKUENSI RADIO (RFID)

Teknologi RFID adalah semacam mode transfer informasi non-kontak yang diwujudkan oleh sinyal frekuensi radio melalui sambungan ruang (medan magnet bolak-balik atau medan elektromagnetik), dan mencapai tujuan identifikasi otomatis melalui informasi yang ditransfer. Pada akhir 1990-an, sebuah kelompok MIT datang dengan istilah IoT. Kemajuan RFID telah memicu pengembangan IoT. Pusat ID Otomatis AS adalah yang pertama mengusulkan konsep pelacakan ID otomatis, yang menjadi salah satu bentuk awal penyebaran IoT. Pada tahun 2008,

Dewan Intelijen Nasional AS menerbitkan sebuah laporan tentang "Teknologi Sipil yang Mengganggu", yang mengidentifikasi IoT sebagai teknologi penting untuk kepentingan nasional.

Teknologi RFID dan Perangkat Penandaan

Perangkat RFID memiliki berbagai ukuran, kebutuhan daya, frekuensi operasi, jumlah penyimpanan yang dapat ditulis ulang dan tidak mudah menguap, dan kecerdasan perangkat lunak. Mereka beroperasi dari beberapa sentimeter hingga ratusan meter. Namun, sumber daya internal diperlukan untuk memungkinkan perangkat RFID besar beroperasi pada jarak yang jauh. Sebaliknya, perangkat RIFD yang lebih kecil tidak memerlukan catu daya apa pun. RFID bekerja melalui kombinasi tiga komponen fungsional, yaitu tag RFID, pembaca RFID, dan antena pembaca. Mari kita mulai dengan contoh RFID.

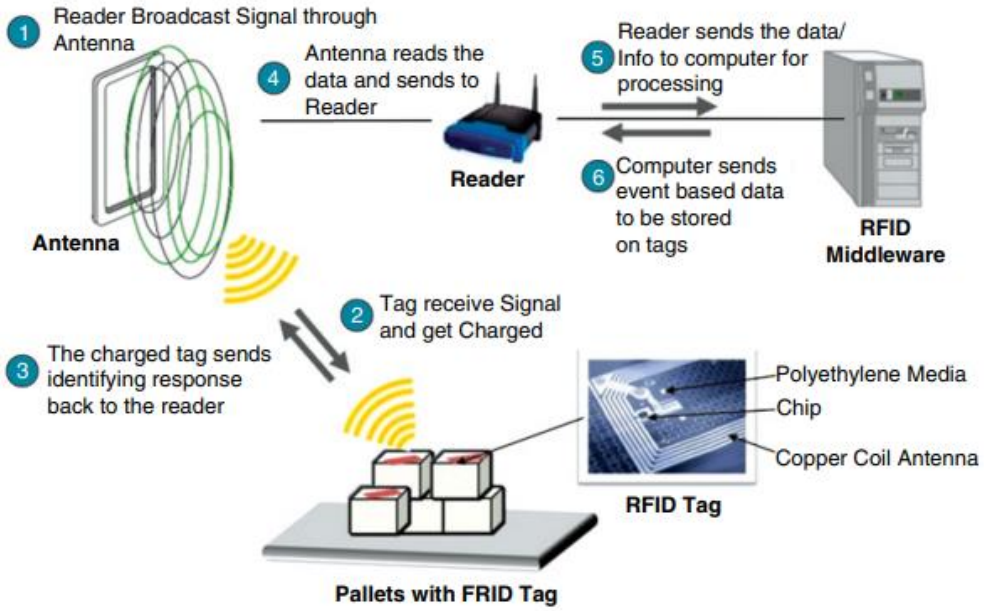
Tag RFID

Ini terdiri dari chip silikon kecil dan antena kecil. Komponen tag tertutup dalam plastik, silikon atau kadang-kadang kaca. Data yang disimpan dalam microchip menunggu untuk dibaca. Biasanya, antena tag menerima energi elektromagnetik dari pembaca RFID. Menggunakan daya yang diambil dari medan elektromagnetik pembaca, tag mengembalikan sinyal radio kembali ke pembaca. Pembaca mengambil sinyal radio tag dan menafsirkan frekuensi menjadi data yang berarti.

Ada tiga jenis tag RFID, yaitu aktif, semi aktif, dan pasif. Tag RFID aktif berisi baterai dan dapat mengirimkan sinyal secara mandiri. Tag RFID pasif tidak memiliki baterai dan memerlukan sumber eksternal untuk memicu transmisi sinyal. Tag RFID semi-aktif sebenarnya adalah tag RFID pasif yang dibantu baterai. Namun, baterai hanya diaktifkan ketika pembaca RFID mengirimkan sinyal energi. Berdasarkan frekuensi radio yang digunakan, tag RFID pasif beroperasi dari frekuensi rendah (LF) ke frekuensi tinggi (HF), dan ke frekuensi ultra tinggi (UHF).

Pembaca dan Antena RFID

Ini adalah stasiun perangkat yang berbicara dengan tag. Pembaca dapat mendukung satu atau lebih antena. Dibandingkan dengan barcode, RFID memiliki keunggulan penting, yaitu perangkat pembaca dapat mendeteksi objek tanpa saling berhadapan. Beberapa pembaca RFID dapat mengidentifikasi beberapa objek secara bersamaan. Antena digunakan untuk memancarkan energi dan kemudian menangkap sinyal kembali yang dikirim kembali dari tag. Hal ini dapat diintegrasikan dengan perangkat pembaca genggam atau dihubungkan ke pembaca dengan kabel. Antena pembaca RFID mendeteksi tag RFID yang mirip dengan radar yang menerangi target. Namun, RFID beroperasi pada rentang yang lebih pendek.



Gambar 3.8 Pembaca RFID mengambil data produk pada e-label (tag RFID) yang ditempatkan pada kotak paket.

Contoh 3.4 Teknologi RFID untuk Tag Barang Dagangan atau e-Labeling Label elektronik atau tag RFID muncul di kotak barang dagangan atau pengiriman. Label elektronik dibuat dari media polietilen yang menampung chip IC kecil dan sirkuit cetak yang digerakkan oleh antena kumparan tembaga. Tag itu sendiri tidak memiliki catu daya yang terpasang padanya. Tag diberi energi oleh gelombang sinyal yang dipancarkan dari antena pembaca. Gambar 3.8 menunjukkan urutan enam peristiwa. Acara 1 hingga 3 menunjukkan energi dan jabat tangan antara pembaca dan tag. Peristiwa 4 sampai 6 menunjukkan bagaimana antena membaca data pada label ke komputer backend untuk diproses.

Komputer mengirimkan data berbasis peristiwa yang diperbarui untuk disimpan pada tag untuk penggunaan di masa mendatang. RFID di tengah dijalankan oleh komputer backend untuk menyelesaikan proses membaca dan memperbarui. Tentu saja, ide tersebut dapat dimodifikasi untuk melayani tujuan identifikasi jarak jauh lainnya. Misalnya, label RFID juga digunakan di department store, supermarket, pencarian inventaris, industri perkapalan, dll.

Arsitektur Sistem RFID

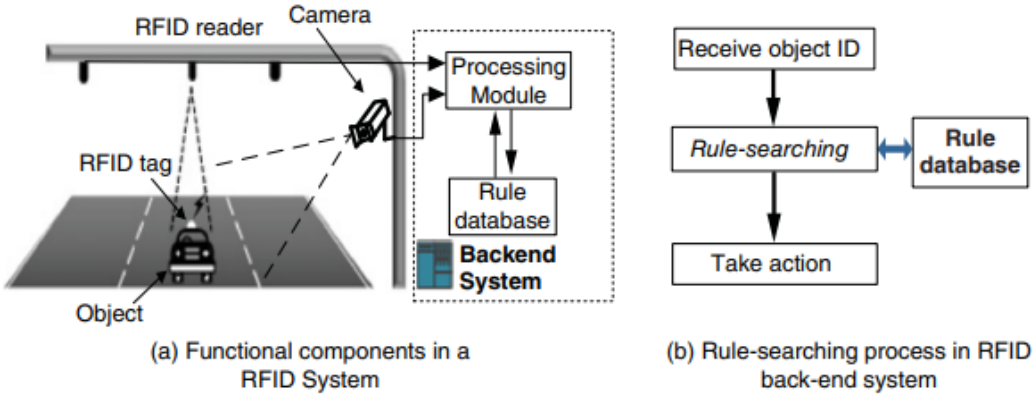
Teknologi dan produk RFID memasuki aplikasi bisnis pada 1980-an. Sekarang, kategori produk RFID meningkat pesat, berbagai tag telah berkembang pesat dengan biaya yang terus menurun, dan industri dengan aplikasi skala besar mulai berkembang. Berasal dari teknologi radar, RFID memiliki prinsip operasi yang mirip dengan radar. Pertama, pembaca mengirimkan sinyal elektronik melalui antena, tag memancarkan informasi identifikasi yang disimpan secara internal setelah menerima sinyal, kemudian pembaca menerima dan mengidentifikasi informasi yang dikirim kembali oleh tag identifikasi melalui antena, dan akhirnya pembaca mengirimkan hasil identifikasi ke host.

Sistem RFID tipikal terdiri dari tag RFID, pembaca RFID, dan sistem backend. Dengan chip RF sederhana dan antena, tag RFID dapat menyimpan informasi yang mengidentifikasi objek yang dilampirkan. Ada tiga jenis tag RFID, yaitu tag pasif, tag aktif, dan tag semi aktif. Tag pasif memperoleh energi melalui sinyal RF dari pembaca, sedangkan tag aktif ditenagai oleh baterai tertanam, yang memungkinkan memori lebih besar atau fungsionalitas lebih. Meskipun tag semi-aktif berkomunikasi dengan pembaca RFID seperti tag pasif, modul tambahan dapat didukung melalui baterai internal. Ketika datang dalam jarak dekat pembaca RFID, informasi yang disimpan dalam tag ditransfer ke pembaca, dan ke sistem backend, yang dapat menjadi komputer yang digunakan untuk memproses informasi ini dan mengendalikan operasi sub-sistem lainnya.).

Contoh 3.5 Jam Tangan Mengebut Mobil dalam Sistem RFID Pencarian Aturan Biasa

Ketika kendaraan yang membawa tag RFID melaju kencang di jalan raya, ia melewati titik pemeriksaan yang dilengkapi dengan pembaca RFID, dan data identifikasi kendaraan dikirimkan ke pembaca RFID dan sistem backend. ID kendaraan diperiksa terhadap database di backend. Sistem memeriksa database untuk mengeluarkan kutipan. Pada Gambar 3.9(a), kendaraan sedang dipantau oleh kamera untuk memeriksa kecepatannya. Jika batas kecepatan terlampaui, berdasarkan aturan pemantauan lalu lintas, beberapa tindakan dipicu, seperti mengeluarkan tilang kepada pengemudi kendaraan. Alternatifnya, kendaraan dapat dikejar oleh mobil polisi untuk menghindari membahayakan pengemudi lain yang berbagi jalan.

Pada Gambar 3.9(b), proses pencarian aturan dijalankan oleh sistem RFID pemeriksaan kecepatan tersebut. Format dasar aturan terdiri dari pernyataan kondisional sederhana dan serangkaian kode tindakan: jika {kondisi (parameter lingkungan), maka {<tindakan1 (parameter1)>, <aksi2 (parameter2)>, di mana parameter lingkungan (misalnya suhu) perature atau kelembaban yang dirasakan oleh beberapa sensor) digunakan untuk menentukan apakah kondisi aturan terpenuhi. Suatu tindakan mewakili operasi yang dimiliki sistem terhadap kendaraan yang sedang berjalan. Diberikan contoh pada Gambar 3.9(a), hasil pencarian aturan dapat berupa: jika {Kecepatan > 120 km/jam} maka {beritahu polisi()}.



Gambar 3.9 Contoh tipikal sistem RFID yang diterapkan pada pemeriksaan kecepatan mobil.

Dukungan IoT dari Manajemen Rantai Pasokan

Teknologi RFID memainkan peran penting dalam bisnis dan pasar. Banyak industri, pemerintah dan layanan masyarakat dapat mengambil manfaat dari aplikasi ini. Ini termasuk

kegiatan atau inisiatif untuk mempromosikan pengembangan masyarakat, kota, dan pemerintah yang lebih baik dan lebih efisien. Aplikasi IoT RFID pada umumnya mencakup layanan ritel dan logistik serta manajemen rantai pasokan.

Layanan Ritel dan Logistik

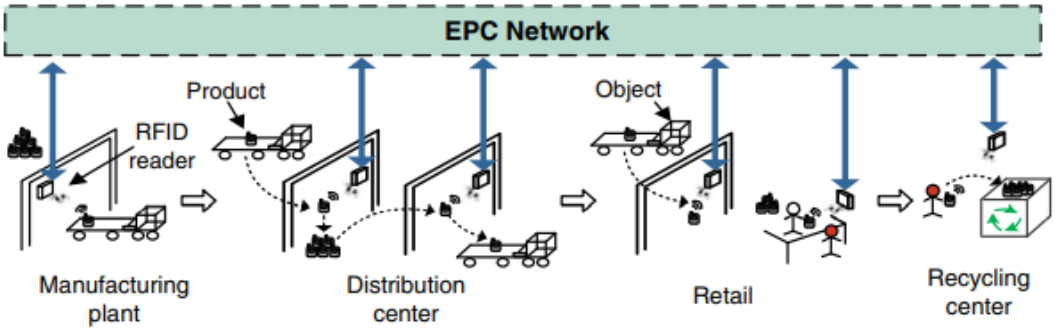
Munculnya aplikasi RFID sangat bergantung pada adopsi oleh pengecer, organisasi logistik dan perusahaan pengiriman paket. Secara khusus, pengecer dapat menandai objek individu untuk memecahkan sejumlah masalah sekaligus: inventaris yang akurat, pengendalian kerugian, dan kemampuan untuk mendukung terminal titik penjualan tanpa pengawasan (yang menjanjikan kecepatan checkout, sekaligus mengurangi pengutilan dan biaya tenaga kerja). Audit dan jaminan rantai dingin dapat memerlukan penandaan makanan dan obat-obatan dengan bahan dan/atau elektronik yang peka terhadap suhu. Memastikan atau memantau apakah bahan yang mudah rusak masih utuh dan/atau memerlukan perhatian mungkin memerlukan komunikasi antara lain seperti sistem pendingin, sistem pencatatan data otomatis, dan teknisi manusia.

Misalnya, di toko kelontong, Anda membeli sekotak susu. Wadah susu akan memiliki tag RFID yang menyimpan tanggal kedaluwarsa dan harga susu. Saat Anda mengangkat susu dari rak, rak mungkin menampilkan tanggal kedaluwarsa spesifik susu, atau informasinya dapat dikirim secara nirkabel ke asisten digital pribadi atau ponsel Anda. Saat Anda keluar dari toko, Anda melewati pintu dengan pembaca tag tertanam. Pembaca ini mentabulasi biaya semua item di keranjang belanja Anda dan mengirimkan tagihan belanjaan ke bank Anda. Produsen produk tahu apa yang telah Anda beli dan komputer toko tahu persis berapa banyak dari setiap produk yang perlu dipesan ulang.

Setelah sampai di rumah, Anda memasukkan susu ke dalam lemari es yang dilengkapi dengan tag reader. Kulkas pintar ini mampu melacak semua bahan makanan yang tersimpan di dalamnya. Ini dapat melacak makanan yang Anda gunakan, seberapa sering Anda mengisi kembali lemari es Anda dan dapat memberi tahu Anda ketika susu dan makanan lainnya rusak. Produk juga dilacak saat dibuang ke tempat sampah atau tempat sampah daur ulang. Berdasarkan produk yang Anda beli, toko kelontong Anda akan mengetahui preferensi unik Anda. Alih-alih menerima buletin generik dengan penawaran belanja mingguan, Anda mungkin menerima buletin yang dibuat khusus untuk Anda.

Manajemen Rantai Pasokan

Supply Chain Management dapat dibantu dengan sistem RFID. Idenya adalah untuk mengelola seluruh jaringan bisnis terkait atau mitra yang terlibat dalam pembuatan produk, pengiriman dan layanan seperti yang dipersyaratkan oleh pelanggan akhir. Pada waktu tertentu, kekuatan pasar dapat menuntut perubahan dari pemasok, penyedia logistik, lokasi dan pelanggan, dan dari sejumlah peserta khusus dalam rantai pasokan. Variabilitas ini memiliki efek signifikan pada infrastruktur rantai pasokan, mulai dari lapisan dasar untuk membangun komunikasi elektronik antara mitra dagang hingga konfigurasi proses yang lebih kompleks, dan pengaturan alur kerja yang penting untuk proses produksi yang cepat.



Gambar 3.10 Manajemen rantai pasokan dalam saluran bisnis multi-mitra.

Jalur pasokan menggabungkan proses, metodologi, alat, dan opsi pengiriman untuk memandu mitra kolaboratif agar bekerja secara berurutan untuk menjalankan bisnis dengan efisiensi tinggi dan kecepatan pengiriman. Perusahaan koperasi harus bekerja dengan cepat karena kompleksitas dan kecepatan rantai pasokan meningkat karena efek persaingan global, fluktuasi harga yang cepat, lonjakan harga minyak, siklus hidup produk yang pendek, spesialisasi yang diperluas, dan kelangkaan bakat. Rantai pasokan adalah jaringan fasilitas yang efisien yang mendapatkan bahan, mengubah bahan ini menjadi produk jadi dan akhirnya mendistribusikan produk jadi ke pelanggan. Contoh berikut dapat menjelaskan bagaimana rantai pasokan dapat dibantu oleh IoT, yang secara khusus dirancang untuk mendorong efisiensi bisnis dan pertumbuhan yang cepat.

Contoh 3.6 Manajemen Rantai Pasokan Dibantu oleh Internet of Things

Manajemen rantai pasokan adalah proses yang digunakan oleh perusahaan untuk memastikan bahwa rantai pasokan mereka efisien dan hemat biaya. Pada Gambar 3.10, rantai pasokan untuk produksi dan penjualan produk konsumen diilustrasikan. Rantai pasokan melibatkan pemasok bahan atau komponen, pusat distribusi, hubungan komunikasi, pusat data cloud, sejumlah besar toko ritel, kantor pusat perusahaan (seperti Wal-Mart) dan pembayaran bank, dll. Ini adalah mitra bisnis yang dihubungkan oleh satelit, Internet, jaringan kabel dan nirkabel, truk, kereta api atau perusahaan pengiriman, perbankan elektronik, penyedia cloud, dll.

Sensor, tag RFID, dan perangkat GPS dapat ditempatkan di mana saja di sepanjang rantai pasokan. Seluruh idenya adalah untuk mempromosikan bisnis online, e-commerce, atau transaksi seluler. Manajemen rantai pasokan terdiri dari lima tahap utama operasi:

1) **Perencanaan dan Koordinasi:** Sebuah rencana atau strategi harus dikembangkan untuk mengatasi bagaimana barang atau jasa dapat memuaskan kebutuhan pelanggan.
2) **Pasokan Bahan dan Peralatan:** Fase ini melibatkan membangun hubungan yang kuat dengan pemasok bahan baku dan juga merencanakan metode pengiriman, pengiriman, dan pembayaran.
3) **Manufaktur dan Pengujian:** Produk diuji, diproduksi dan dijadwalkan untuk pengiriman.
4) **Pengiriman Produk:** Pesanan pelanggan diambil dan pengiriman barang direncanakan.

5) **Layanan Purna Jual dan Pengembalian:** Pada tahap ini pelanggan dapat mengembalikan produk yang cacat dan perusahaan juga memenuhi permintaan pelanggan. Perangkat lunak rantai pasokan digunakan oleh banyak perusahaan untuk manajemen rantai pasokan yang efisien.

3.4 SENSOR, JARINGAN SENSOR NIRKABEL, DAN SISTEM GPS

Dengan berkembangnya desain sirkuit, pemrosesan sinyal dan Micro-Electro Mechanical Systems (MEMS), berbagai jenis sensor diproduksi, dari sensor optik sederhana dan sensor suhu hingga sensor rumit seperti sensor karbon dioksida, dll. sensor biasanya ditentukan oleh persyaratan aplikasi tertentu. Secara umum, prosesor berinteraksi dengan sensor baik melalui sinyal analog atau sinyal digital. Output sensor berbasis sinyal analog secara fisik diukur kuantitas analog seperti tegangan. Besaran analog harus didigitalkan sebelum digunakan. Oleh karena itu, sensor ini membutuhkan konverter analog-digital eksternal serta teknik kalibrasi tambahan. Dalam sensor berbasis sinyal digital, interaksi antara prosesor dan sensor disederhanakan, karena kuantitas digital disediakan oleh sensor secara langsung.

Perangkat Keras Sensor dan Sistem Operasi

Sensor menjembatani dunia fisik dan sistem elektronik. Sensor menghasilkan data dalam bentuk sinyal analog atau digital yang diumpankan ke node sensor untuk diproses segera. Namun, tergantung pada situasinya, beberapa bentuk pra-pemrosesan atau penyaringan khusus juga dapat dilakukan sebelumnya, baik sebagai bagian dari algoritma yang diterapkan di node sensor, atau sebagai bagian dari komponen perangkat keras perantara, meskipun kasus sebelumnya telah menjadi lazim.

Kami mempertimbangkan dua kategori sensor: satu untuk pengawasan lingkungan dan yang lainnya untuk penginderaan tubuh. Sensor pengawasan lingkungan digunakan untuk mengumpulkan informasi lingkungan. Sensor tubuh dikerahkan untuk mengumpulkan data tubuh vital. Seperti yang ditunjukkan pada Tabel 3.2, sensor pengawasan lingkungan tipikal termasuk sensor untuk cahaya tampak, suhu, kelembaban, tekanan, magnet, akselerasi, giroskopik, suara, asap, optik RF pasif, cahaya terstruktur, kelembaban tanah, gas karbon dioksida (CO2), dll.

Sensor Gerak Inersia

Dalam kategori ini, akselerometer dan giroskop sejauh ini merupakan perangkat yang paling umum digunakan untuk memperkirakan dan memantau postur tubuh, dan berbagai pola gerak manusia. Kemampuan ini sangat diperlukan untuk berbagai jenis aplikasi, terutama di bidang perawatan kesehatan, olahraga, dan game konsol. Untuk tujuan ini, akselerometer mengukur tarikan gravitasi dan kemiringan, sedangkan giroskop mengukur perpindahan sudut. Secara umum, penggunaan gabungannya menghasilkan informasi orientasi dan pola gerakan pengguna yang beragam.

Tabel 3.2 Karakteristik sensor pengawasan lingkungan.

Pabrikan	Sensor	Tegangan (V)	Kekuatan	Waktu Pengambilan Sampel
Taos	Cahaya tampak	2.7–5.5	1.9 mA	330 us
Dallas Semiconductor	Suhu	2.5–5.5	1 mA	400 ms
Sensirion	Kelembaban	2.4–5.5	550 uA	300 ms
Intersema	Tekanan	2.2–3.6	1 mA	35 ms
Honeywell	Daya tarik	Any	4 mA	30 us
Analog Devices	Percepatan	2.5–3.3	2 mA	10 ms
Panasonic	Suara	2–10	0.5 mA	1 ms
Motorola	Merokok	6–12	5 uA	–
Melixis	RF-Optik	Any	0 mA	1 ms
Li-Cor	Cahaya terstruktur	Any	0 mA	1 ms
Ech2o	Kelembaban tanah	2–5	2 mA	10 ms

Sensor Bioelektrik

Jenis sensor khusus ini digunakan untuk mengukur variasi listrik pada kulit pengguna/pasien yang dapat secara langsung atau tidak langsung berkorelasi dengan aktivitas atau kondisi organ tubuh saat ini. Sensor elektrokardiografi adalah contoh khasnya, yang biasanya berbentuk bantalan melingkar yang ditempatkan secara strategis di sekitar tubuh dan ekstremitas manusia untuk memantau aktivitas jantung (EKG). Jenis sensor serupa yang ditempatkan di atas kulit digunakan untuk mengukur aktivitas listrik otot rangka (EMG) untuk membantu dalam diagnosis gangguan saraf dan otot.

Node sensor tubuh terutama terdiri dari dua bagian: sensor sinyal fisiologis dan platform radio, di mana beberapa sensor tubuh dapat dihubungkan. Fungsi umum sensor tubuh adalah untuk mengumpulkan sinyal analog yang sesuai dengan aktivitas fisiologis manusia atau tindakan tubuh. Sinyal analog semacam itu dapat diperoleh oleh papan yang dilengkapi radio terkait dengan cara kabel, di mana sinyal analog didigitalkan. Akhirnya, sinyal digital diteruskan oleh radio transceiver. Jenis sensor tubuh yang tersedia secara komersial terdaftar sebagai berikut:

- **Sensor elektrokimia:** Jenis sensor ini menghasilkan keluaran listrik yang digerakkan oleh reaksi kimia kecil antara bahan kimia sensor dan zat tubuh. Contoh yang baik adalah sensor glukosa darah, yang mengukur jumlah glukosa yang beredar dalam darah. Contoh lain adalah pemantauan tingkat konsentrasi karbon dioksida (CO_2) dalam respirasi manusia.
- **Sensor optik:** Perangkat yang memancarkan dan menerima cahaya baik dalam pita cahaya tampak maupun inframerah biasanya digunakan dalam pengukuran saturasi oksigen non-invasif dalam darah yang bersirkulasi dalam tubuh manusia. Untuk tujuan ini,

oksimeter denyut mengukur tingkat penyerapan cahaya saat cahaya melewati pembuluh darah dan arteri pengguna/pasien.

- **Sensor suhu:** Jenis sensor yang populer ini ditempatkan di atas kulit di berbagai tempat di sekitar tubuh manusia, dan secara rutin digunakan selama penilaian fisiologis pasien.

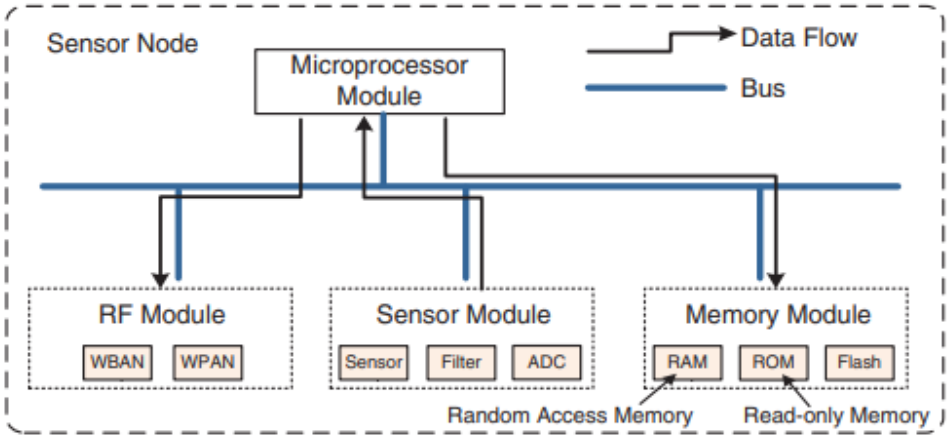
Dalam lingkungan IoT yang cerdas, kita dapat memantau dan menjelajahi dunia fisik secara ekstrem. Misalnya, manusia tidak dapat mentolerir suhu lebih dari 1000 C atau membedakan perubahan suhu yang tidak kentara. Oleh karena itu, IoT untuk perlindungan lingkungan telah menghadirkan persyaratan yang lebih tinggi pada pengukuran suhu dengan jangkauan luas menggunakan sensor tahan panas. Kehidupan kita sehari-hari telah dipengaruhi oleh penggunaan sensor yang ekstensif, seperti sensor suhu dan kelembaban di AC, pengontrol suhu di pemanas air, pengontrol suara untuk lampu di koridor, dan remote control untuk perangkat TV, dll.

Selanjutnya, sensor telah banyak diterapkan di bidang-bidang seperti perlindungan lingkungan, kesehatan medis, industri dan pertanian serta militer dan pertahanan nasional. Sensor adalah peralatan atau perangkat yang dapat merasakan ukuran yang ditentukan dan mengubahnya menjadi sinyal keluaran yang dapat digunakan sesuai dengan aturan tertentu, dan umumnya terdiri dari elemen penginderaan, elemen transduksi, dan rangkaian dasar. Elemen penginderaan mengacu pada bagian dalam sensor yang dapat langsung merasakan kuantitas fisik; elemen transduksi mengubah output elemen penginderaan menjadi parameter rangkaian (yaitu tegangan dan induktansi); dan akhirnya rangkaian dasar mengubah parameter rangkaian menjadi keluaran listrik.

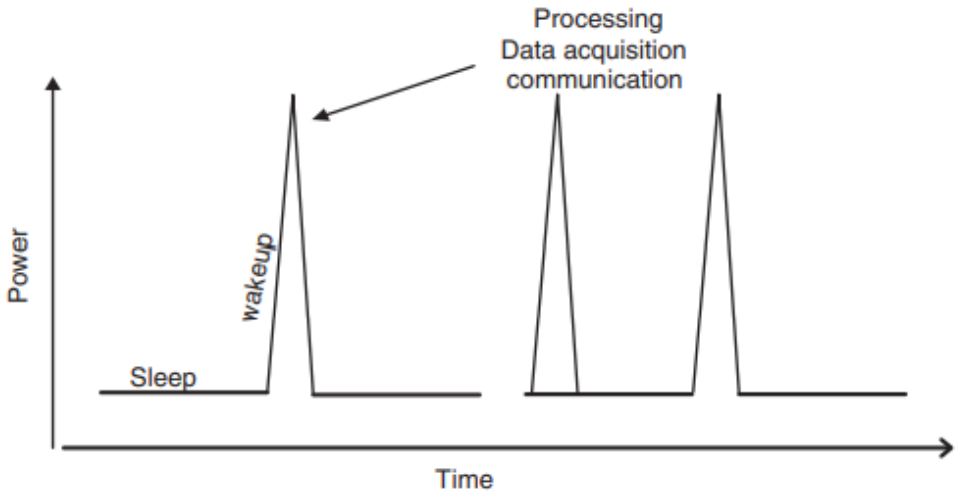
Desain Arsitektur Sensor

Gambar 3.11(a) menunjukkan node sensor tipikal dengan modul sensor, radio dan memori. Modul sensor terdiri dari sensor, filter dan analog-to-digital converter (ADC). Sensor mengubah beberapa bentuk energi menjadi sinyal listrik analog, yang difilter dan didigitalkan oleh ADC untuk diproses lebih lanjut. Kita akan membahas sistem radio untuk BAN dan WPAN yang digunakan untuk transmisi data yang dirasakan di bagian selanjutnya. Sebuah sensor mudah dikacaukan dengan beberapa konsep, seperti sensor node, wireless sensor node dan wireless sensor network. Sensor bertindak di sini sebagai konverter sinyal.

Node sensor biasanya disebut sensor plus mikroprosesor, yang fungsinya untuk lebih jauh mengubah sinyal analog menjadi sinyal digital. Node sensor nirkabel selanjutnya mengintegrasikan chip komunikasi nirkabel, dll. berdasarkan node sensor tradisional: sistem operasi mikro seperti TinyOS dan Contiki diinstal dengan beberapa program tertanam untuk menganalisis dan memproses informasi yang diterima dan mengirimkannya melalui jaringan . Jika beberapa node sensor nirkabel ditempatkan untuk membuat mereka saling terhubung dan membentuk satu jaringan ad hoc, jaringan seperti itu disebut jaringan sensor atau jaringan sensor nirkabel.



(a) Modul fungsional dalam node sensor pintar.



(b) Manajemen daya di node sensor

Gambar 3.11 Manajemen daya dalam operasi sensor biasa.

Konsumsi Daya

Node sensor nirkabel umumnya ditempatkan di udara terbuka, sehingga mereka tidak dapat memperoleh catu daya melalui kabel. Oleh karena itu, desain perangkat keras mereka harus mempertimbangkan penghematan energi sebagai tujuan desain yang penting. Misalnya, dalam mode operasi normal, kekuatan prosesor sensor tipikal adalah antara 3 dan 15 mW. Manajemen daya periodik dari sensor tipikal diilustrasikan pada Gambar 3.11(b). Tidur, bangun dan lonjakan pemrosesan mengambil bentuk siklus periodik, berulang dalam interval waktu yang tetap. Sebagian besar waktu, sensor dalam mode tidur, dengan perangkat bangun secara berkala. Lonjakan kekuatan memungkinkan pengumpulan data dan operasi komunikasi. Jarang, peristiwa pemicu terdeteksi dan mereka mengaktifkan perangkat sensor. Masa pakai operasi sensor yang lama dapat berlangsung selama berbulan-bulan hingga bertahun-tahun, tergantung pada apakah perangkat ditenagai oleh energi matahari atau oleh sumber energi berkelanjutan lainnya.

Harga dan Ukuran

Umumnya, penyebaran jaringan skala besar membutuhkan jumlah node sensor yang lebih besar untuk menyelesaikan tugas yang rumit. Ada tradeoff antara harga node dan jumlah node sensor dengan anggaran tetap. Oleh karena itu, desain perangkat keras mereka harus mempertimbangkan efektivitas biaya sebagai tujuan desain kritisnya. Biasanya, node sensor nirkabel harus mudah diangkut dan digunakan, sehingga desain perangkat kerasnya harus menganggap mikrominiaturisasi sebagai tujuan desain yang penting. Namun, batasan ukuran node juga membatasi fungsi node sensor.

Fleksibilitas dan Perluasan

Node sensor diterapkan pada berbagai jenis aplikasi, sehingga desain perangkat keras dan perangkat lunaknya harus fleksibel dan dapat diperluas. Selain itu, fleksibilitas dan perluasan merupakan pengamanan penting untuk mewujudkan penyebaran skala besar jaringan sensor. Desain perangkat keras node harus memenuhi antarmuka standar tertentu; misalnya, antarmuka node dan pelat sensor membuatnya bermanfaat untuk memasang sensor dengan fungsi berbeda pada node.

Selain itu, desain perangkat lunak harus dapat disesuaikan dan dapat menginstal modul perangkat lunak dengan fungsi yang berbeda sesuai dengan tuntutan aplikasi yang berbeda. Sementara itu, desain perangkat lunak juga harus mempertimbangkan perluasan sistem dalam domain waktu. Misalnya, jaringan sensor harus dapat menambahkan node baru secara terus menerus, dengan proses ini tidak mempengaruhi kinerja jaringan yang ada. Contoh lain adalah bahwa perangkat lunak node harus dapat memperbarui program secara otomatis melalui jaringan daripada menyebarkan kembali setiap kali setelah node yang disebarkan diambil kembali dan dibakar.

Tabel 3.3 membandingkan fitur platform sensor representatif dalam hal dukungan OS, standar nirkabel dan kecepatan data, dll. Kami berfokus pada dukungan sistem operasi, standar nirkabel yang digunakan, kecepatan data maksimum, jangkauan luar ruangan, dan tingkat daya. Fitur sistem ini mengungkapkan karakteristik utama sensor dari perspektif desainer aplikasi umum. Kita dapat melihat bahwa semua sensor mencapai konsumsi daya yang rendah, tetapi memiliki kecepatan data yang rendah berkisar antara 38,4 hingga 720 kbps, yang tidak cukup untuk jaringan sensor tubuh skala besar atau aplikasi yang melibatkan lalu lintas data multimedia seperti streaming video. Paket yang berjalan pada sensor ZigBee IEEE 802.15.4 telah diadopsi secara luas. Bluetooth ternyata kurang hemat energi. Interferensi dari perangkat radio lain yang berbagi pita ISM 2,4 GHz dapat menimbulkan masalah lain saat menggunakannya untuk membangun jaringan area tubuh.

Tabel 3.3 Perbandingan node sensor tipikal dalam aplikasi kehidupan sehari-hari.

Nama	Dukungan OS	Standar Nirkabel	Tarif Tanggal	Diluar Ruangan Rentang (m)
BAN node	TinyOS	IEEE 802.15.4	250 kbps	50

BTNode	TinyOS	Bluetooth	24 Mbps	100
eyesIFX	TinyOS	TDA5250	64 kbps	–
iMote	TinyOS	Bluetooth	720 kbps	30
iMote2	TinyOS or .NET	IEEE 802.15.4	250 kbps	30
IRIS	TinyOS	IEEE 802.15.4	250 kbps	300
Micaz	TinyOS	IEEE 802.15.4	250 kbps	75 to 100
Mica2	TinyOS	IEEE 802.15.4	38.4 kbps	>100
Mulle	TCP/IP or TinyOS	Any	250 kbps	>10
TelOS	TinyOS	Bluetooth or IEEE	250 kbps	75 to 100
802.15.4				
ZigBit	ZDK	IEEE 802.15.4	250 kbps	3700

Kekokohan

Kekokohan adalah perlindungan penting untuk mewujudkan penerapan jaringan sensor dalam waktu lama. Untuk komputer umum, jika sistem macet, orang dapat mem-boot ulangnya untuk memulihkan sistem; namun, ini tidak berguna untuk node sensor. Oleh karena itu, desain program node harus kuat untuk menjamin bahwa node dapat bekerja secara efisien untuk waktu yang lama. Misalnya, jika biaya desain perangkat keras memungkinkan, kita dapat mengadopsi sensor multi-bentuk, sehingga bahkan jika satu jenis sensor rusak, yang lain dapat digunakan untuk keseluruhan sistem. Saat merancang perangkat lunak, kita biasanya perlu memodulasi fungsi dan melakukan tes total setiap modul fungsi sebelum penerapan sistem. Bagian selanjutnya dikhususkan untuk pemilihan komponen perangkat keras, antarmuka komunikasi, catu daya, dan sistem operasi untuk desain dan aplikasi modul sensor.

Perangkat Penyedia Energi

Umumnya, node sensor bertenaga baterai, yang membuat node lebih mudah digunakan. Secara teori, baterai dengan kapasitas 2000 mAh dapat terus mengeluarkan arus 10 mA selama 200 jam. Namun, karena berbagai faktor seperti perubahan tegangan dan perubahan lingkungan, kapasitas baterai tidak dapat digunakan secara total. Selain bertenaga baterai, node juga dapat menggunakan sumber energi terbarukan seperti energi matahari dan energi angin. Misalnya, di bawah sinar matahari langsung, panel surya satu inci persegi dapat menyediakan 10 mW energi listrik; sementara di bawah pencahayaan dalam ruangan, panel yang sama dapat menyediakan energi listrik 10–100 uW.

Energi listrik yang terkumpul pada siang hari dapat digunakan oleh node pada malam hari. Teknologi kunci untuk memanfaatkan energi terbarukan adalah cara menyimpannya, dengan dua jenis teknologi yang saat ini digunakan. Salah satunya adalah dengan menggunakan baterai yang dapat diisi ulang, keuntungan utama mereka adalah bahwa ada self-discharge yang relatif lebih sedikit dan tingkat pemanfaatan energi listrik yang lebih tinggi, dan kelemahan utama adalah efisiensi pengisian yang relatif lebih rendah dan waktu pengisian yang terbatas. Teknologi lain yang relatif baru adalah penggunaan ultra-kapasitor, dengan keunggulan utamanya adalah

efisiensi pengisian yang tinggi dan waktu pengisian dapat mencapai 1 juta. Juga, mereka tidak mudah dipengaruhi oleh faktor-faktor seperti suhu dan getaran.

Contoh 3.7 Aplikasi Sensor Jaringan untuk Perlindungan Lingkungan:

Banyak skema perlindungan lingkungan yang penting dapat didukung oleh jaringan sensor nirkabel. Skenario perlindungan lingkungan alam dapat mencakup: i) respon struktur seismik terhadap penilaian kerusakan setelah gempa bumi; ii) transportasi kontaminasi untuk pengendalian polusi; iii) pengendalian pencemaran laut dengan memantau mikroorganisme laut; dan iv) analisis biokompleksitas ekosistem melalui pemantauan sensor. Skenario perlindungan lingkungan ini memerlukan penggunaan sejumlah besar sensor mikro, pemrosesan on-board dan antarmuka nirkabel yang layak pada skala yang sangat kecil, yang dapat memantau fenomena dari dekat. Skema ini memungkinkan pemantauan lingkungan yang padat secara spasial dan temporal. Penginderaan tertanam terdistribusi skala besar sekarang dapat mengungkapkan beberapa fenomena, yang sebelumnya tidak mudah diamati.

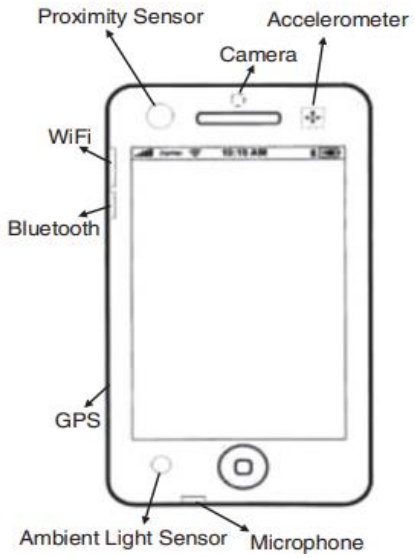
Mikroprosesor

Mikroprosesor adalah inti yang bertanggung jawab untuk computing dalam node penginderaan nirkabel. Chip mikroprosesor saat ini juga mengintegrasikan penyimpanan internal, memori flash, konverter modul, Io digital, dll. Jenis karakteristik yang sangat terintegrasi ini membuatnya sangat cocok untuk digunakan dalam jaringan penginderaan nirkabel. Sekarang kita akan menganalisis beberapa karakteristik kunci prosesor, yang mempengaruhi kinerja operasi keseluruhan dari node.

Chip Komunikasi

Chip komunikasi adalah komponen penting dari node penginderaan nirkabel. Ada dua karakteristik utama dari konsumsi energi chip komunikasi: satu adalah bahwa chip tersebut mengkonsumsi sebagian besar energi dalam satu node penginderaan nirkabel. Sebagai contoh, pada Node TelosB yang sering digunakan, saat ini arus CPU hanya 50 0uA dalam keadaan normal, sedangkan arus mencapai hampir 20 mA ketika chip komunikasi mengirim dan menerima data.

Saat chip komunikasi dengan konsumsi daya rendah mengirim dan menerima data, konsumsi energinya menunjukkan sedikit perbedaan. Ini berarti bahwa selama chip komunikasi terbuka, chip tersebut mengkonsumsi jumlah energi yang hampir sama baik untuk mengirim dan menerima data atau tidak. Umumnya, jarak transmisi dari chip komunikasi merupakan indeks penting bagi kita untuk memilih node penginderaan. Jarak transmisi mereka diatur oleh daya yang ditransmisikan dari chip.



Gambar 2.12 Perangkat sensor yang dibangun di dalam ponsel pintar pada tahun 2016.

Sistem Operasi di Node Sensor

Sebagai inti dari sistem perangkat lunak node sensor, sistem operasi node menyediakan drive perangkat keras pada aplikasi atas, manajemen sumber daya, penjadwalan tugas, antarmuka program, dll. Karakteristik utama yang membuat sistem operasi node berbeda dari yang tradisional adalah bahwa sumber daya platform perangkat keras sangat terbatas. Sistem operasi node sensor tipikal termasuk TinyOS dan MOS, dll. Sensor OS sangat mini dan TinyOS adalah yang paling banyak digunakan dalam jaringan penginderaan nirkabel.

Penginderaan melalui Ponsel Pintar

Mengenai penginderaan IoT, perubahan besar terjadi di sekitar kita, seperti telepon seluler sekarang yang secara langsung mengintegrasikan berbagai sensor untuk penginderaan dunia fisik. Seperti yang ditunjukkan pada Gambar 3.12, semakin banyak ponsel pintar yang dilengkapi dengan perangkat seperti penerima GPS, kamera, perekam suara, termometer, altimeter dan barometer, dll. Karena kemampuan intrinsik koneksi ke Internet di lingkungan seluler, ponsel pintar memungkinkan perangkat sensor dimigrasikan ke grup pengguna global. Dengan demikian, ponsel pintar tidak hanya membangun jembatan antara perangkat sensor dan Internet, tetapi juga menghubungkan manusia dengan dunia sosial dan dunia maya, sehingga menjadi elemen penting untuk IoT.

Saat ini, penilaian status kesehatan pengguna adalah topik penelitian yang menjanjikan di bidang IoT kesehatan. Ada berbagai faktor yang dapat digunakan untuk membuat penilaian, seperti suasana hati, interaksi sosial, kebiasaan tidur, tingkat aktivitas, tingkat kepuasan hidup yang dirasakan, dll. Namun, penelitian sebelumnya memerlukan interaksi tatap muka antara koordinator studi dan subjek, sehingga membatasi jangkauan geografis dan skala studi ini.

Tabel 3.4 Sensor dan penggunaan yang dipasang pada ponsel pintar biasa.

Sensor/Nama Data	Jenis	Periode Pengambilan Sampel
Memori, beban CPU, pemanfaatan CPU, Baterai, Lalu lintas jaringan, Status konektivitas	Sistem	1 detik
Geo-lokasi	Sensor	10 detik
Akselerometer, Magnetometer, Giroskop	Sensor	100 ms
Kedekatan, Tekanan, Cahaya, Kelembaban, Suhu	Sensor	1 detik
Aktivitas Telepon, SMS, MMS	Aktivitas pengguna	1 detik
Status layar, Bluetooth, WiFi	Aktivitas pengguna	3 menit

Tabel 3.4 menunjukkan beberapa contoh data yang dapat ditangkap oleh ponsel pintar. Sensor yang terpasang pada ponsel pintar modern memungkinkan berbagai pengumpulan informasi, seperti aktivitas fisik, lokasi, pola mobilitas, interaksi sosial (misalnya menggunakan sensor jarak), dan detak jantung. Selain itu, pola dan tren penggunaan telepon juga dapat memberikan informasi konteks yang sangat berharga. Data tersebut mencakup riwayat penelusuran, kebiasaan komunikasi (panggilan, SMS), aktivitas jejaring sosial, dan penggunaan aplikasi.

Contoh 3.8 Fitur Smart Phone atau Smart Watch untuk Aplikasi Kesehatan Berbagai data sensorik, seperti akselerometer, GPS, SMS, kamera, perekam suara, termometer, altimeter dan barometer, dll., dapat memberikan informasi kontekstual tambahan yang penting untuk kehidupan yang lebih baik. pemahaman tentang tren dan outlier dalam respons survei (misalnya respons survei terkait suasana hati dapat berbeda saat dikirimkan dari rumah, tempat kerja, saat berlibur, dll.). Selain itu, ponsel pintar memudahkan pengumpulan data dalam jangka waktu yang lama.

Dengan digitalisasi informasi medis dan distribusi perangkat pintar yang cepat, saat ini layanan kesehatan secara aktif direncanakan dan dikembangkan berdasarkan perangkat pintar. Pada tahun 2015, 500 juta pengguna ponsel pintar diharapkan menerapkan aplikasi kesehatan seluler, terutama untuk olahraga, diet, dan manajemen penyakit kronis. Tidak seperti kebanyakan penyakit kronis lainnya, diabetes dapat dikelola oleh pasien. Oleh karena itu, perangkat seluler pintar dapat menjadi alat universal untuk manajemen diabetes mandiri karena penetrasi dan fungsinya yang tinggi.

Aplikasi perawatan kesehatan seluler untuk OS Android dikembangkan untuk menyediakan manajemen diabetes mandiri. Aplikasi ini terdiri dari manajemen Diabetes, manajemen Berat Badan, evaluasi risiko Kardio-serebrovaskular, Evaluasi stres dan depresi dan manajemen Latihan. Dengan dukungan ponsel pintar atau jam tangan pintar, berbagai data

terkait perawatan kesehatan dapat dikumpulkan, seperti Detak jantung, Laju Pernapasan, Suhu kulit, Durasi waktu tidur, Tingkat aktivitas (misalnya Statis, Berjalan, Lari), Ekspresi wajah video, dll.

Jaringan Sensor Nirkabel dan Jaringan Area Tubuh

Seperti yang ditunjukkan pada Tabel 3.5, WSNs telah diklasifikasikan menjadi 3 generasi selama 30 tahun terakhir. Sensor yang digunakan pada generasi pertama sebagian besar adalah sensor tunggal yang ditempatkan di kendaraan atau dijatuhkan dari udara. Mereka besar, seperti kotak sepatu, dan beratnya beberapa kilogram. Jaringan hanya mengasumsikan topologi bintang atau titik ke titik dan ditenagai oleh baterai besar yang dapat bertahan selama berjam-jam atau berhari-hari. Pada generasi kedua, sensor menjadi lebih kecil, seperti sebungkus kartu remi, dan beratnya beberapa gram, dan baterai AA bertahan selama berhari-hari dan berminggu-minggu. Mereka muncul dalam konfigurasi client-server atau P2P. Generasi saat ini berukuran partikel debu, dengan berat yang dapat diabaikan, dan muncul di jaringan P2P untuk aplikasi tertanam dan jarak jauh.

Tabel 3.5 Tiga generasi jaringan sensor nirkabel.

Fitur WSN	**Generasi Pertama (1990-an)**	**Detik. Jenderal (2000-an)**	**Generasi Ketiga (2010-an)**
Produsen	Konstruktor khusus, mis. untuk TRSS	Crossbow Technology, Inc. Sensoria Corp, Ember Corp.	Dust, Inc. dan lainnya
Ukuran fisik	Kotak sepatu besar dan lebih tinggi	Paket kartu ke kotak sepatu	Partikel debu
Bobot	Kilogram	gram	diabaikan
Arsitektur simpul	Penginderaan, pemrosesan, dan komunikasi terpisah	Penginderaan, pemrosesan, dan komunikasi terintegrasi	Penginderaan, pemrosesan, dan komunikasi terintegrasi
Topologi	Titik-ke-Titik, bintang	Server klien, peer-to-peer	Rekan ke rekan
Seumur hidup catu daya	baterai besar; jam, hari, dan lebih lama	baterai AA; berhari-hari hingga berminggu-minggu	Tenaga surya; bulan hingga tahun
Penyebaran	Sensor tunggal yang ditempatkan di kendaraan atau air-drop	Ditempatkan dengan tangan	Tertanam, ditaburkan, tertinggal

Jaringan sensor ad hoc nirkabel menerapkan sejumlah besar sensor (kebanyakan stasioner). Selain penyebaran sensor di permukaan laut atau penggunaan sensor robot bergerak,

tak berawak, dalam operasi militer, sebagian besar node dalam jaringan sensor pintar tidak bergerak. Jaringan 10.000 atau bahkan 100.000 node dibayangkan di masa depan dan skalabilitas menjadi tuntutan. Penggunaan energi yang rendah diharapkan pada sensor modern. Karena dalam banyak aplikasi node sensor akan ditempatkan di area yang jauh, layanan node mungkin tidak dapat dilakukan. Dalam hal ini, masa pakai node ditentukan oleh masa pakai baterai, sehingga memerlukan pengurangan pengeluaran energi atau penggunaan energi matahari untuk memberi daya pada perangkat.

Kemajuan dalam teknologi komunikasi nirkabel, seperti biosensor yang dapat dipakai dan ditanamkan, bersama dengan perkembangan terkini di area computing tertanam, memungkinkan desain, pengembangan, dan implementasi jaringan area tubuh. Kelas jaringan ini membuka jalan bagi penerapan aplikasi pemantauan perawatan kesehatan yang inovatif. Perbedaan antara BAN dan WSN adalah sebagai berikut:

Penerapan dan Kepadatan

Jumlah node sensor/aktuator yang digunakan oleh pengguna bergantung pada berbagai faktor. Biasanya, simpul BAN ditempatkan secara strategis di tubuh manusia, atau disembunyikan di bawah pakaian. Selain itu, BAN tidak menggunakan node yang berlebihan untuk mengatasi berbagai jenis kegagalan. Ketentuan desain lain yang umum adalah WSN konvensional. Akibatnya, BAN tidak padat simpul. Namun, WSN sering digunakan di tempat-tempat yang mungkin tidak mudah diakses oleh operator, yang mengharuskan lebih banyak node ditempatkan untuk mengkompensasi kegagalan node.

Kecepatan Data

Sebagian besar WSN digunakan untuk pemantauan berbasis peristiwa, di mana peristiwa dapat terjadi pada interval yang tidak teratur. Sebagai perbandingan, BAN digunakan untuk mendaftarkan aktivitas dan tindakan fisiologis manusia, yang mungkin terjadi secara lebih berkala, dan dapat mengakibatkan aliran data aplikasi menunjukkan kecepatan yang relatif stabil.

Latensi

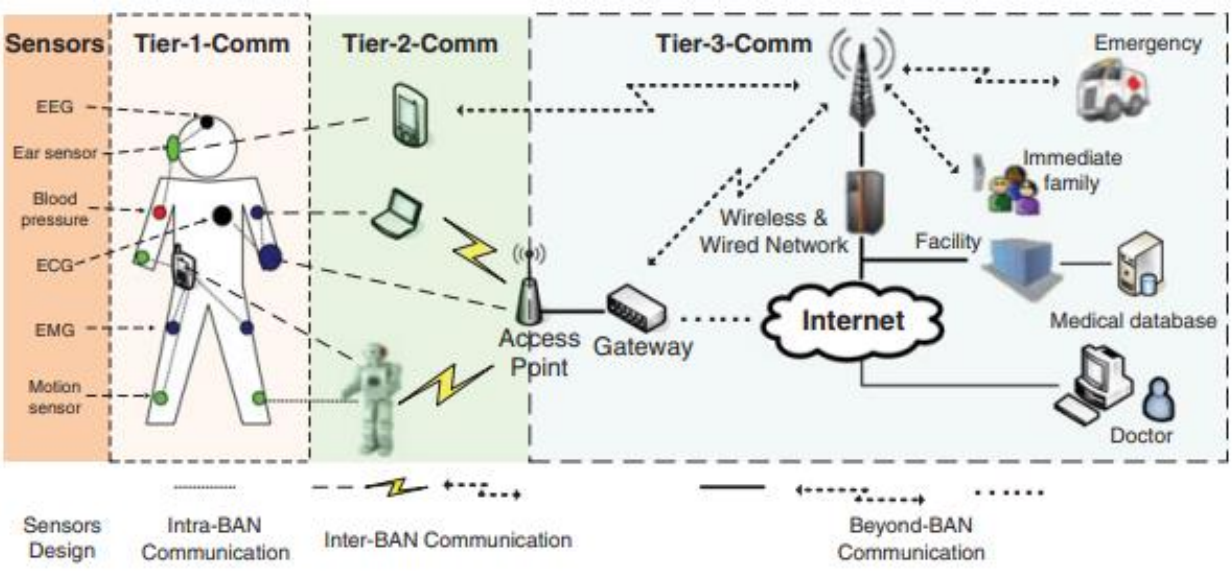
Persyaratan ini ditentukan oleh aplikasi, dan dapat ditukar dengan peningkatan keandalan dan konsumsi energi. Namun, meskipun penghematan energi jelas bermanfaat, penggantian baterai di node BAN jauh lebih mudah dilakukan daripada di WSN, yang nodenya secara fisik tidak dapat dijangkau setelah penerapan. Oleh karena itu, mungkin perlu untuk memaksimalkan masa pakai baterai di WSN dengan mengorbankan latensi yang lebih tinggi.

Mobilitas

Pengguna BAN dapat berpindah-pindah. Oleh karena itu, node BAN memiliki pola mobilitas yang sama, tidak seperti node WSN yang biasanya dianggap stasioner. BAN umumnya dianggap sebagai teknologi yang memungkinkan untuk berbagai aplikasi, termasuk pemantauan kesehatan dan kebugaran, tanggap darurat, dan kontrol perangkat. Terobosan terbaru dalam elektronik solid-state memungkinkan pembuatan perangkat low-profile berdaya rendah yang dapat saling berhubungan secara modular untuk menciptakan apa yang disebut node sensor yang terdiri dari satu atau lebih perangkat sensor, MCU, dan radio transceiver. yang menghilangkan kebutuhan akan kabel untuk berkomunikasi dengan node untuk mentransfer data yang

dikumpulkan. BAN hanya pada tahap awal. Gambar 3.13 mengilustrasikan arsitektur umum sistem pemantauan kesehatan berbasis BAN.

Dalam bentuknya yang paling dasar, perangkat sensor beroperasi dengan memuat MCU dengan program biner yang mengakses antarmuka perangkat keras tingkat rendah, yang pada gilirannya memperoleh data dari perangkat sensor yang sebenarnya. Program berisi instruksi yang diperlukan untuk perangkat sensor untuk mengumpulkan satu atau lebih bacaan dalam periode waktu tertentu. Data sensor mentah selanjutnya dapat diproses untuk mengubahnya menjadi informasi bermakna yang dapat ditafsirkan setelah dikirimkan oleh chip radio ke perangkat atau sistem eksternal untuk analisis lebih lanjut. Sesuai dengan namanya, node sensor dimaksudkan untuk dipakai atau ditanamkan di tubuh manusia.



Gambar 3.13 Arsitektur tiga tingkat berdasarkan sistem komunikasi BAN.

Selain itu, dua atau lebih perangkat sensor di sekitar mereka dapat membangun hubungan nirkabel untuk mengoordinasikan operasi bersama mereka, sehingga menciptakan sistem jaringan. Oleh karena itu, literatur yang ada sering menyebut BAN sebagai Wireless BAN (WBAN) atau Wireless Body Area Sensor Network (WBASN). Bagian selanjutnya memperkenalkan beberapa kemajuan paling relevan dalam teknologi BAN, diikuti dengan deskripsi tantangan teknis penting yang harus diatasi oleh para peneliti agar BAN menjadi efisien, andal, dan ekonomis.

Sistem ini memonitor EKG: (electroencephalography) EEG, (electromyography) EMG, sensor gerak dan sensor tekanan darah mengirim data ke perangkat server pribadi (PS) terdekat. Kemudian, melalui koneksi Bluetooth/WLAN, data ini dialirkan dari jarak jauh ke situs dokter medis untuk diagnosis waktu nyata, ke database medis untuk penyimpanan catatan, atau ke peralatan terkait yang mengeluarkan peringatan darurat. Pada artikel ini, kami memisahkan arsitektur komunikasi BAN menjadi tiga komponen. Desain Tier-1-Comm (yaitu komunikasi intra-

BAN), desain Tier-2-Comm (yaitu komunikasi antar-BAN) dan desain Tier-3-Comm (yaitu komunikasi di luar BAN), seperti yang ditunjukkan pada Gambar 3.13. Komponen-komponen ini mencakup berbagai aspek yang berkisar dari masalah desain tingkat rendah hingga tingkat tinggi, dan memfasilitasi pembuatan sistem BAN berbasis komponen yang efisien untuk berbagai aplikasi. Dengan menyesuaikan setiap komponen desain, misalnya biaya, cakupan, efisiensi, bandwidth, QoS, dll., persyaratan khusus dapat dicapai sesuai dengan konteks aplikasi dan permintaan pasar tertentu.

Sistem Pemosisian Global

Layanan berbasis lokasi (LBS) adalah teknologi pendukung utama untuk aplikasi IoT. Sebagian besar "hal" dalam domain IoT saling berhubungan melalui berbagai jaringan komunikasi nirkabel. Ketika data sensorik yang dikumpulkan oleh teknologi penginderaan IoT dikirim ke cloud, data lokasi terkait menjadi penting. Tanpa informasi lokasi, pengetahuan dasar tentang keadaan dan pengguna tidak dapat diperoleh, yang mengakibatkan kegagalan aplikasi IoT. Dengan demikian, lokalisasi adalah proses kritis.

Salah satu metode populer untuk menentukan lokasi perangkat adalah melalui Global Positioning System (GPS). Penerima GPS (aGPS) aktif dapat menerima sinyal satelit dan mengirimkan informasi posisi ke pusat kendali aGPS. AGPS menjadi standar bagi perusahaan yang ingin memantau armada kendaraan serta alat berat lainnya. Pelacakan GPS real-time praktis untuk memperoleh informasi langsung dan terperinci tentang sejumlah besar kendaraan atau objek yang sedang dilacak. Ini bisa menjadi bisnis rental mobil yang menyediakan mobil untuk banyak pelanggan. Proses pelacakan kendaraan waktu nyata dibagi menjadi empat langkah berikut:

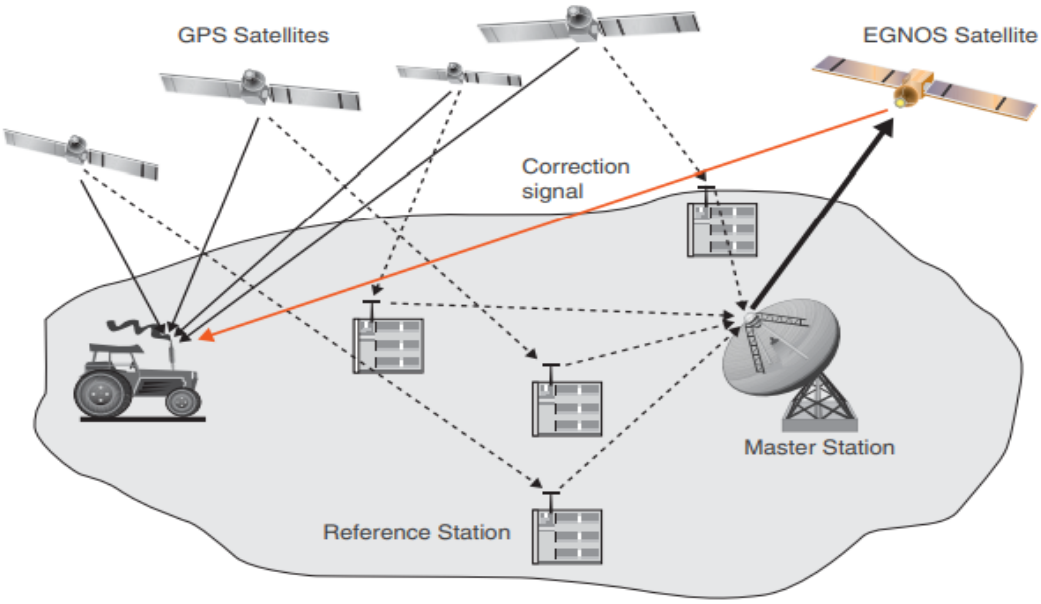
1) Penerima GPS di setiap mobil mengakses sinyal penerima dari jaringan satelit.
2) Informasi satelit yang dikumpulkan dikirim ke pusat GPS melalui jaringan seluler.
3) Pusat kendali memasukkan informasi lokasi yang dihitung melalui peta global.
4) Pusat kendali mengirimkan perintah ke setiap unit untuk memicu alarm, menghentikan mesin, mengubah arah atau beberapa pesan pribadi, dll. Lokalisasi otonom penerima GPS sangat penting, karena lokasi membuat data sensorik bermakna secara geografis.

Banyak aplikasi dan layanan jaringan nirkabel secara langsung atau tidak langsung bergantung pada informasi lokasi. Untuk aplikasi IoT, pelokalan penting karena data yang melimpah yang dirasakan melalui IoT tidak ada artinya tanpa lokasi untuk layanan berbasis lokasi. Meskipun GPS adalah solusi langsung, GPS memiliki dua keterbatasan untuk penggunaan praktis. Pertama, sinyal GPS sangat dinamis dan tidak stabil di lingkungan dalam ruangan atau dinamis, menghasilkan akurasi lokasi yang buruk. Kedua, mahalnya biaya untuk melengkapi setiap node sensor dengan penerima GPS untuk sistem IoT skala besar. Direkomendasikan skema gabungan yang menggunakan GPS dan lokalisasi jaringan secara bersamaan.

Setiap satelit hanya terlihat oleh penerima GPS di bagian tertentu dari permukaan bumi. Pada waktu yang berbeda, setiap receiver hanya dapat melihat sebagian (sekitar 6) satelit. Empat satelit cukup untuk menemukan posisi penerimaan secara akurat. Pada waktu yang berbeda, subset satelit yang berbeda akan terlihat oleh penerima. Segmen kontrol terdiri dari stasiun

kontrol utama dan sejumlah antena dan stasiun pemantau tanah khusus dan bersama. Segmen pengguna terdiri dari ratusan ribu pengguna militer AS dan sekutu dari layanan penentuan posisi akurat GPS yang aman.

Penerima GPS tanah menghitung lokasi 3-D dari empat atau lebih satelit dengan bantuan beberapa stasiun referensi bumi dan stasiun induk, seperti yang ditunjukkan pada Gambar 3.14. Pada dasarnya, penerima GPS membandingkan waktu sinyal ditransmisikan oleh satelit dengan waktu diterimanya. Perbedaan waktu memberitahu penerima GPS seberapa jauh jarak satelit. Dengan jarak yang diukur dari beberapa satelit, penerima dapat menentukan posisi pengguna dan menampilkannya di peta elektronik unit. Puluhan juta pengguna sipil, komersial, dan ilmiah hanya diizinkan untuk menggunakan fungsionalitas terdegradasi dari apa yang disebut layanan penentuan posisi standar yang tidak dapat digunakan untuk tujuan serangan musuh.



Gambar 3.14 Penerima GPS tanah menghitung lokasi 3-D dari empat atau lebih satelit dengan bantuan dari beberapa stasiun referensi bumi dan stasiun induk.

GPS Pasif versus Aktif

Perangkat pelacak GPS memungkinkan untuk melacak orang, kendaraan, dan aset lainnya, di mana pun di Bumi. Ada dua jenis sistem pelacakan GPS, pasif versus aktif. Dalam pelacakan pasif, GPS hanyalah penerima, bukan pemancar. Perangkat pelacakan GPS pasif tidak memiliki kemampuan transmisi untuk mengirim data GPS dari kendaraan. Oleh karena itu, GPS pasif juga dikenal sebagai pencatat data yang digunakan terutama sebagai alat perekam. Unit pelacakan GPS aktif menggabungkan metode untuk mengirimkan informasi pengguna dari kendaraan. Meskipun uplink data satelit tersedia, komunikasi data seluler adalah yang paling umum dan hemat biaya. Pembaruan inkremental otomatis menyediakan sumber pelacakan yang berkelanjutan selama periode perekaman. Ini memberikan posisi logging saat ini dan juga historis.

Perangkat pelacakan GPS pasif menyimpan data lokasi GPS di memori internalnya, yang kemudian dapat diunduh ke komputer untuk dilihat di lain waktu, sementara sistem pelacakan GPS aktif mengirim data secara berkala untuk dilihat secara real time. Ketika data real-time tidak diperlukan, perangkat pelacakan GPS pasif cenderung lebih disukai oleh konsumen individu karena kenyamanan dan keterjangkauannya yang ringkas. Orang tua yang peduli dapat memasang unit pelacakan GPS di mana saja di kendaraan remaja mereka untuk memantau kebiasaan mengemudi mereka dan mengetahui ke mana mereka pergi, dan bahkan petugas penegak hukum sekarang mengandalkan pelacakan GPS pasif untuk melacak tersangka kriminal dan meningkatkan keselamatan sipil melalui elektronik pengawasan terhadap pembebasan bersyarat. Unit pelacakan GPS pasif juga berfungsi sebagai pencegahan pencurian dan bantuan pengambilan di kendaraan konsumen serta komersial.

Prinsip Operasi GPS

Mengetahui jarak dari penerima ke satelit posisi tetap menyiratkan bahwa penerima berada di permukaan bola yang berpusat di satelit. Dengan empat satelit, lokasi penerima terdeteksi di persimpangan empat permukaan bola. Perpotongan dua bola satelit umumnya berbentuk lingkaran. Lingkaran ini dapat direduksi menjadi satu titik, jika kedua bola hanya menyentuh permukaannya. Setelah menemukan dua permukaan bola yang berpotongan, sekarang kami mempertimbangkan bagaimana lingkaran yang berpotongan itu berpotongan dengan bola satelit ketiga. Lingkaran dan permukaan bola berpotongan di nol, satu atau dua titik. Dengan penerima di permukaan bumi, penerima perlu memilih titik yang paling dekat dengan penerima dari dua titik yang berpotongan.

Jelas, metode triangulasi di atas dapat mengakibatkan beberapa kesalahan dalam mempersempit ke tepat satu titik dengan ketidakakuratan minimum. Untuk menemukan titik secara akurat, penerima harus menggunakan satelit keempat ke rumah dengan lebih tepat. Bola satelit keempat akan datang sangat dekat dengan dua titik perpotongan terakhir dari tiga bola satelit. Dengan demikian, lokasi penerima akhir ditentukan dengan mencatat titik terdekat yang dihitung dari dua titik akhir ke permukaan bola satelit keempat. Dalam kasus tidak ada kesalahan, posisi yang tepat berada. Jika tidak, beberapa offset, katakanlah 10 meter dari lokasi yang tepat, dapat dihasilkan dari kesalahan yang diperkenalkan. Untuk lebih mengurangi kesalahan, lebih banyak satelit dapat dilibatkan, tetapi ini akan memakan biaya.

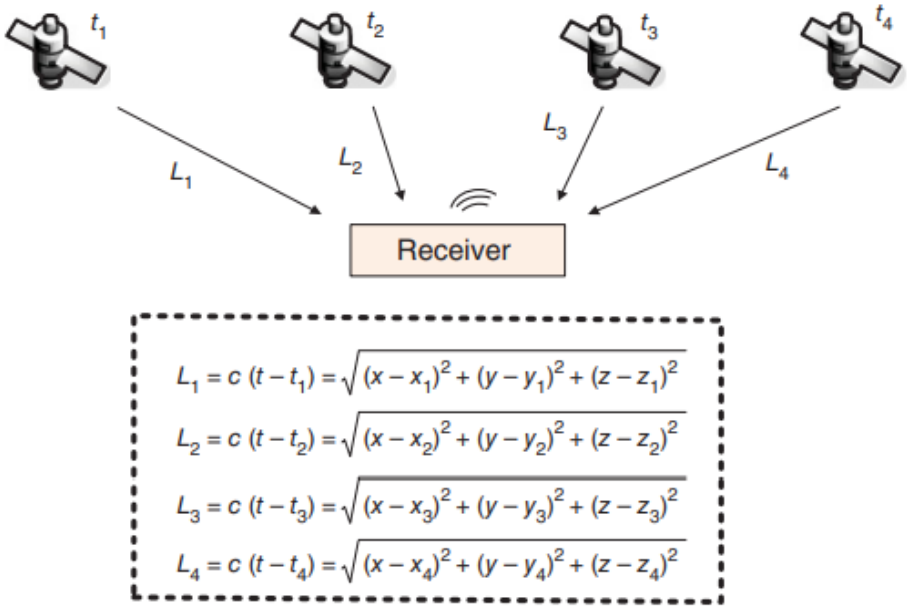
Setiap satelit secara terus menerus mentransmisikan pesan yang mencakup: i) waktu pesan ditransmisikan; ii) informasi orbit yang tepat (ephemeris); dan iii) kesehatan sistem umum dan orbit kasar semua satelit GPS (almanak). Sebuah penerima GPS menghitung posisinya dengan tepat waktu sinyal yang dikirim oleh satelit GPS tinggi di atas permukaan bumi. Penerima GPS harus terkunci pada sinyal setidaknya tiga satelit untuk menghitung posisi 2-D (lintang dan bujur) dan melacak pergerakan. Dengan empat atau lebih satelit dalam pandangan, penerima dapat menentukan posisi 3-D pengguna (lintang, bujur dan ketinggian).

Setelah posisi pengguna ditentukan, unit GPS dapat menghitung informasi lain, seperti kecepatan, arah, lintasan, jarak perjalanan, jarak ke tujuan, waktu matahari terbit dan terbenam, dan banyak lagi. Penerima menggunakan pesan yang diterimanya untuk menentukan waktu

transit setiap pesan dan menghitung jarak ke setiap satelit. Jarak ini bersama dengan lokasi satelit digunakan untuk menghitung posisi penerima. Posisi ini ditampilkan mungkin dengan tampilan peta bergerak atau informasi garis lintang, garis bujur dan ketinggian. Banyak unit GPS menunjukkan informasi turunan seperti arah dan kecepatan yang dihitung dari perubahan posisi.

Perhitungan Lokasi Triangulasi

Metode perhitungan lokasi ini diilustrasikan pada Gambar 3.15. Penerima menggunakan pesan yang diterima dari empat satelit untuk menentukan posisi satelit dan waktu pengiriman. Komponen x, y dan z dari posisi dan waktu pengiriman dilambangkan sebagai [xi, yi, zi, ti], di mana indeks i = 1, 2, 3 atau 4 menunjukkan satelit. Mengetahui kapan pesan diterima tri, penerima menghitung waktu transit pesan sebagai tri – ti. Dengan asumsi pesan menempuh kecepatan cahaya c, jarak yang ditempuh dihitung dengan di = (tri – ti) × c. Setelah membahas bagaimana permukaan bola berpotongan, sekarang kita merumuskan persamaan untuk kasus ketika kesalahan muncul. Biarkan b menunjukkan kesalahan jam atau bias, jumlah jam penerima mati. Penerima memiliki empat yang tidak diketahui, tiga komponen posisi penerima GPS dan bias jam [x, y, z, b].



Gambar 3.15 Metode triangulasi untuk menghitung sinyal lokasi tertunda dari empat satelit.

Persamaan permukaan bola dihitung dengan persamaan berikut untuk i = 1, 2, 3, dan 4:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = ([tr_i + b - t_i]c)^2 \qquad (3.1)$$

Metode pencarian akar multidimensi seperti metode Newton-Raphson dapat digunakan. Pendekatan ini adalah untuk linierisasi di sekitar solusi perkiraan, katakanlah [$x^{(k)}$, $y^{(k)}$, $z^{(k)}$, $b^{(k)}$] pada iterasi k, kemudian selesaikan empat persamaan linier yang diturunkan dari persamaan kuadrat di atas menjadi dapatkan nilai yang sesuai pada contoh waktu k + 1. Metode Newton–

Raphson konvergen lebih cepat daripada metode posisi lainnya. Jika tersedia lebih dari empat satelit, penghitungan dapat memilih dari empat hasil yang ditunjukkan pada Gambar 3.15. Kesalahan perhitungan posisi sangat sensitif terhadap kesalahan jam. Oleh karena itu, dalam sistem navigasi berbasis satelit, sinkronisasi jam sangat penting untuk meminimalkan kesalahan lokasi.

Tiga satelit cukup untuk menentukan posisi penerima, karena ruang memiliki tiga dimensi dan posisi dekat permukaan bumi selalu diasumsikan. Namun, bahkan kesalahan jam yang sangat kecil dikalikan dengan kecepatan cahaya sinyal satelit dapat mengakibatkan kesalahan posisi yang besar. Oleh karena itu, kebanyakan receiver menggunakan empat atau lebih satelit untuk mengetahui lokasi dan waktu receiver. Waktu yang dihitung seringkali disembunyikan oleh sebagian besar aplikasi GPS, yang hanya menggunakan informasi lokasi. Beberapa aplikasi GPS khusus memang menggunakan waktu untuk transfer waktu, pengaturan waktu sinyal lalu lintas, dan sinkronisasi stasiun pangkalan ponsel.

Meskipun empat satelit diperlukan untuk operasi normal, jika variabel 1-D sudah diketahui, penerima dapat menentukan posisinya hanya dengan menggunakan tiga satelit. Misalnya, sebuah kapal atau pesawat mungkin memiliki ketinggian yang diketahui. Beberapa penerima GPS mungkin menggunakan petunjuk atau asumsi tambahan (yaitu menggunakan kembali ketinggian terakhir yang diketahui, perhitungan mati, navigasi inersia atau termasuk informasi dari komputer kendaraan) untuk memberikan posisi yang kurang akurat (terdegradasi) ketika kurang dari empat satelit tersedia.

Status Penerapan di Seluruh Dunia

Tabel 3.6 merangkum empat sistem penentuan posisi global yang digunakan saat ini. Selain GPS yang digunakan oleh AS, yang sekarang terbuka untuk aplikasi sipil global oleh banyak negara, Rusia telah menggunakan GLONASS (Global Navigation Satellite System) untuk penggunaan militer Rusia secara eksklusif. Di Uni Eropa, ada posisi Galileo sistem. Pada tahun 2015, China telah meluncurkan 20 satelit menuju sistem yang lengkap, dengan 31 satelit direncanakan untuk tahun 2020-an.

Tabel 3.6 Empat Sistem Pemosisian Global di AS, UE, Rusia, dan Cina.

Fitur	**GPS**	**GLONASS**	**Beidou**	**Galileo**
Entitas politik	Amerika Serikat	Rusia	Cina	Uni Eropa
pengkodean	CDMA	FDMA/CDMA	CDMA	CDMA
Ketinggian orbit	20.180 km (12.540 mi)	19.130 km (11.890 mil)	21.150 km (13.140 mil)	23.220 km (14.430 mil)
Periode	11,97 jam (11 jam 58m)	11,26 jam (11 jam 16m)	12,63 jam (12 jam 38m)	14.08 jam
Jumlah satelit	Setidaknya 24	31 (24 operasional)	5 GEO, 30 satelit MEO	(14 jam 5m)

3.5 TEKNOLOGI COMPUTING KOGNITIF DAN SISTEM PROTOTIPE

Bagian ini dikhususkan untuk mempelajari teknologi computing kognitif dan sistem prototipe. Kami memeriksa proses pengembangan tiga sistem kognitif eksperimental di Pusat Penelitian Almaden IBM, Proyek Tim Google Brain, dan Akademi Ilmu Pengetahuan China. Terakhir, kami menunjukkan bagaimana konteks IoT bermanfaat bagi layanan kognitif, dan menyajikan konteks IoT dalam kognisi dan perangkat kognitif terkini seperti kacamata AR dan headset VR.

Ilmu Kognitif dan Neuroinformatika

Ilmu kognitif bersifat interdisipliner. Ini mencakup bidang psikologi, kecerdasan buatan, ilmu saraf dan linguistik, dll. Ini mencakup banyak tingkat analisis dari *Machine learning* tingkat rendah dan mekanisme keputusan ke sirkuit saraf tingkat tinggi untuk membangun komputer model otak. Pada tahun 2008, konsep dasar ilmu kognitif diberikan oleh Paul Thagrad: “Berpikir paling baik dipahami dalam kerangka representasi struktur dalam pikiran dan prosedur computing yang beroperasi pada struktur tersebut.” Secara umum, tiga pendekatan diadopsi dalam aplikasi computing kognitif:

1) Menerapkan pustaka perangkat lunak di cloud atau superkomputer untuk *Machine learning* dan studi neuroinformatika.
2) Menggunakan representasi dan algoritma untuk menghubungkan input dan output dari komputer syaraf tiruan.
3) Gunakan chip saraf untuk mengimplementasikan komputer mirip otak untuk *Machine learning* dan kecerdasan.

Neuroinformatika mencoba menggabungkan penelitian informatika dan pemodelan otak untuk memberi manfaat bagi kedua bidang sains. Informatika berbasis komputer tradisional memfasilitasi pemrosesan dan penanganan data otak. Melalui teknologi perangkat keras dan perangkat lunak, kita dapat mengatur basis data, pemodelan, dan komunikasi dalam penelitian otak. Atau sebaliknya, penemuan yang disempurnakan dalam ilmu saraf dapat memicu pengembangan model baru komputer mirip otak.

Pada Tabel 1.12, kami telah mengidentifikasi bidang terkait dengan computing kognitif dan teknologi neuroinformatika. Perhatian utama penelitian AI adalah untuk mengetahui bagaimana pembelajaran manusia, memori, bahasa, persepsi, tindakan, dan penemuan pengetahuan ditangani. Kami berharap dapat menerapkan Smart Machine untuk membantu pengambilan keputusan manusia atau membuat aktivitas kehidupan sehari-hari kita lebih aman, efektif, efisien, dan nyaman, dll.

Contoh 3.9 Ilmu Kognitif dan Neuroinformatika di Academia dan IBM Labs McCulloch dan Pitts mengembangkan model computing jaringan saraf tiruan (JST) pertama yang terinspirasi oleh struktur jaringan saraf biologis. Contoh pertama eksperimen sains kognitif dilakukan di Departemen Psikologi Sosial MIT menggunakan memori komputer sebagai model untuk kognisi manusia. Sebagian besar upaya penelitian tentang ilmu kognitif didukung oleh National Institute of Health di AS.

IBM mendirikan Blue Brain Project pada Mei 2005. Proyek ini dilakukan pada superkomputer 8000-prosesor Blue Gene/L yang dibuat oleh IBM. Pada saat itu, ini adalah salah satu superkomputer tercepat di dunia. Misi Blue Brain Project adalah untuk memahami fungsi dan disfungsi otak mamalia melalui simulasi terperinci. Proyek IBM mencakup aspek-aspek berikut:

- **Basis data:** neuron model yang direkonstruksi 3-D, sinapsis, jalur sinaptik, statistik sirkuit mikro, neuron model komputer, dan neuron virtual.
- **Visualisasi:** pembuat sirkuit mikro dan visualisator hasil simulasi, 2-D, 3-D dan sistem visualisasi imersif sedang dikembangkan.
- **Lingkungan simulasi:** lingkungan simulasi untuk simulasi skala besar neuron kompleks secara morfologis pada superkomputer Blue Gene IBMl.
- **Simulasi dan eksperimen:** iterasi antara simulasi skala besar dari sirkuit mikro neokorteks dan eksperimen.

Ditelusuri lebih jauh, IBM Watson Center memiliki proyek Deep Blue yang mengarah pada kompetisi bermain catur yang mengalahkan Juara Catur Dunia Garry Kasparov pada tahun 1997. Itu adalah sistem kognitif pertama di dunia yang dibangun untuk bekerja dengan perangkat keras komputer tradisional. Baru-baru ini, IBM mengumumkan bahwa mereka akan mendorong layanan kognitif sebagai upaya utama dalam dekade berikutnya. Tujuannya adalah untuk mengembangkan industri kognitif untuk melayani masyarakat manusia dan untuk mempromosikan ekonomi global. Kami akan membahas pengembangan chip sinaptik IBM dalam membangun komputer mirip otak. Kemudian, kami akan memeriksa upaya Tim Otak Google. Teknologi cloud dan IoT memainkan peran penting dalam transformasi industri ini.

Chip dan Sistem Computing yang Terinspirasi dari Otak

Di bagian ini, kami mempelajari beberapa chip prosesor baru, arsitektur non-von Neumann, dan pengembangan ekosistem di IBM, Nividia, Intel, dan Institut Teknologi Computing China untuk computing kognitif. Meskipun proyek-proyek ini masih dalam tahap penelitian, mereka mewakili teknologi baru yang menggabungkan computing dengan kognisi untuk meningkatkan kapasitas manusia dan pemahaman tentang semua jenis lingkungan di sekitar kita.

Program SyNapse IBM

IBM memiliki program penelitian SyNapse, yang ditujukan untuk pengembangan perangkat keras dan perangkat lunak baru untuk computing kognitif. Proyek ini telah didukung oleh DARPA (Defense Advanced Research Projects Agency) di AS. Pada tahun 2014, IBM meluncurkan desain chip komputer neurosinaptik di Science Magazine, yang dikenal sebagai prosesor TruthNorth. Prosesor ini dapat meniru kemampuan computing otak manusia dan efisiensi daya. Desain chip dapat memungkinkan berbagai aplikasi, seperti membantu orang dengan gangguan penglihatan untuk menavigasi dengan aman di lingkungan mereka.

Chip ini bisa menjejalkan kekuatan superkomputer ke dalam mikroprosesor seukuran prangko. Daripada memecahkan masalah melalui perhitungan matematis brute-force, chip dirancang untuk memahami lingkungannya, menangani ambiguitas dan mengambil tindakan secara real time dan dalam konteks. Diperkirakan rata-rata otak manusia memiliki 100 miliar

neuron dan 100 hingga 150 triliun sinapsis. Dimodelkan pada otak manusia, chip TrueNorth menggabungkan 5,4 miliar transistor, paling banyak yang pernah dipasang IBM pada sebuah chip. Chip ini memiliki 1 juta neuron yang dapat diprogram dan 256 juta sinapsis yang dapat diprogram.

Diperkirakan bahwa chip sinaptik ini dapat diterapkan pada robot penyelamat kecil, secara otomatis membedakan suara dalam rapat dan membuat transkrip yang akurat untuk setiap pembicara. Bahkan berpotensi untuk mengeluarkan peringatan tsunami, memantau tumpahan minyak atau menegakkan aturan jalur pelayaran. Apa yang menakjubkan adalah bahwa chip tersebut hanya mengkonsumsi daya 70 miliwatt untuk melakukan fungsi-fungsi di atas, hampir sama dengan tingkat yang dikonsumsi oleh alat bantu dengar. Chip tersebut masih dalam tahap prototipe.

Diumumkan di sebuah konferensi bahwa IBM mungkin menghabiskan $3 miliar untuk mendorong masa depan chip komputer semacam itu dan mengeksplorasi potensi layanan kognitifnya. Itu tidak memerlukan beban computing yang berat untuk operasi kompleks seperti dalam sistem kognitif biologis. Misalnya, jika robot yang berjalan dengan mikroprosesor saat ini berjalan menuju pilar, itu akan bergantung pada pemrosesan gambar dan sumber daya dan daya computing yang besar untuk menghindari tabrakan. Sebagai perbandingan, robot yang menggunakan chip sinaptik akan menghindari bahaya dengan merasakan pilar, seperti yang dilakukan seseorang dengan konsumsi daya yang kecil.

Para ahli percaya bahwa inovasi seperti SyNapse's TrueNorth dapat membantu mengatasi batas kinerja arsitektur von Neumann, sistem berbasis matematika yang menjadi inti dari hampir setiap komputer yang dibangun sejak 1948. “Ini merupakan pencapaian luar biasa dalam hal skalabilitas dan daya rendah. konsumsi,” kata Horst Simon, direktur Laboratorium Berkeley Departemen Energi AS dan pakar ilmu komputer. IBM mengharapkan chip tersebut membantu mengubah ilmu pengetahuan, teknologi, bisnis, pemerintah, dan masyarakat dengan mengaktifkan aplikasi visi, audisi, dan multi-indera. Ini bisa menjadi langkah pertama dalam merancang komputer masa depan berdasarkan model otak manusia. Demikian pula, grafik Nividia juga condong ke arah ini untuk menggerakkan otak superkomputer.

Proyek Cambricon China

Proyek ini dimulai sebagai program penelitian bersama antara Dr Tianshi Chen di Institut Teknologi Computing, Akademi Ilmu Pengetahuan China dan Prof Oliver Tenam di INRIA Prancis. Tim peneliti gabungan telah mengembangkan serangkaian akselerator perangkat keras, yang dikenal sebagai Cambricon, untuk jaringan saraf dan aplikasi *Deep learning*. Chip tersebut dibuat sebagai prosesor sinaptik untuk menggerakkan computing saraf tiruan dalam operasi *Machine learning*. Cambricon menjauh dari arsitektur klasik von Neumann untuk mencocokkan jenis operasi khusus yang dilakukan dalam perhitungan jaringan saraf tiruan.

Tugas *Machine learning* menjadi meresap dalam berbagai sistem, dari sistem tertanam hingga pusat data. Misalnya, algoritme *Deep learning*, menggunakan jaringan saraf convolutional dan mendalam (CNN dan DNN akan dibahas di Bab 6), membutuhkan siklus pembelajaran yang panjang untuk dilatih secara berguna di komputer konvensional. Akselerator Cambricon dirancang untuk fokus pada solusi CNN dan DNN skala besar. Tim gabungan ICT-INRIA

membuktikan bahwa adalah mungkin untuk membangun akselerator dengan throughput tinggi, yang mampu melakukan 452 GOP/dtk (operasi jaringan saraf kunci dalam perkalian bobot sinaptik).

Chip ini dibuat pada tapak kecil 3,02 mm2 dan teknologi silikon 485 mW. Tim juga telah menyusun arsitektur set instruksi (ISA) baru untuk penggunaan yang efisien dari prosesor mirip otak tersebut. ISA baru ini dirancang khusus untuk jaringan saraf atau computing kognitif. Dibandingkan dengan akselerator GPU SIMD 128-bit 2 GHz, chip akselerator mencapai kecepatan 117 lebih cepat dengan 21 kali pengurangan konsumsi daya. Dengan arsitektur *Machine learning* 64-chip yang diperluas, tim telah menunjukkan kecepatan 450 kali lebih cepat dari serangkaian chip GPU dengan pengurangan daya 150 kali.

Melalui upaya multi-nasional, computing kognitif yang berorientasi pada manusia menjadi lebih dekat dengan kenyataan. Sangat menarik untuk mengamati perkembangan prosesor sinaptik berbasis neuron tujuan khusus dalam chip multi-core atau multi-core multi-inti dan penggunaan sejumlah besar chip tersebut untuk membangun superkomputer kognitif masa depan. Kami akan mempelajari *Machine learning* dan algoritma *Deep learning* di Bab 4 hingga 6. Meskipun algoritma ML/DL ini diterapkan pada cloud atau klaster server saat ini, seperti yang tercakup dalam Bab 7 hingga 9, algoritma tersebut dapat ditargetkan untuk memandu desain masa depan. sistem computing kognitif.

Proyek Tim Otak Google

Dalam penelitian dan industri, pengenalan suara selalu diminati. Akan menyenangkan memiliki mesin perekam cerdas yang dapat mendengarkan ucapan manusia dan menghasilkan laporan tekstual yang terdokumentasi. Demikian pula, Perserikatan Bangsa-Bangsa perlu memiliki sistem penerjemahan bahasa otomatis, tidak hanya penerjemahan antar dokumen teks, tetapi juga antara pidato dan laporan terdokumentasi dalam bahasa yang berbeda.

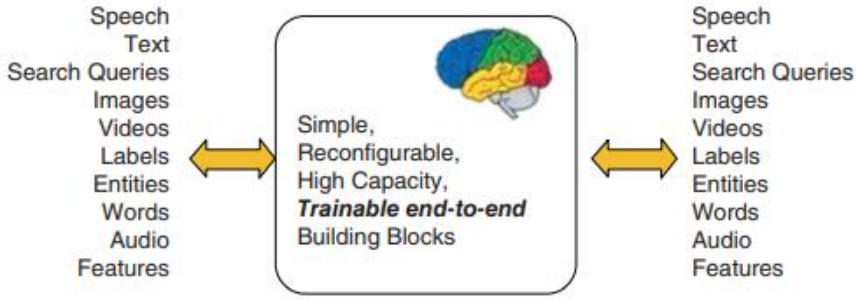
Contoh 3.10 Proyek Otak Google dalam Mengembangkan Produk ML/DL Baru Proyek

Google Brain dimulai pada tahun 2011 sebagai upaya bersama antara Jeff Dean, Greg Corrado dan Andrew Ng. Ng tertarik menggunakan teknik *Deep learning* untuk memecahkan masalah AI. Mereka telah membangun sistem perangkat lunak *Deep learning* skala besar, yang disebut DistBelief di atas infrastruktur computing cloud Google. Pada 2012, New York Times melaporkan bahwa sekelompok 16.000 komputer digunakan untuk meniru beberapa aspek aktivitas otak manusia dan telah berhasil melatih dirinya sendiri untuk mengenali kucing berdasarkan 10 juta gambar digital yang diambil dari video YouTube.

Pada tahun 2013, Geoffrey Hinton bergabung dengan Google, seorang peneliti terkemuka di bidang *Deep learning*. Selanjutnya, Google menggabungkan DeepMind Technologies dan merilis TensorFlow. Produk Google terkemuka yang dikembangkan dari Brain Team termasuk sistem pengenalan suara Android, pencarian foto Google, dan rekomendasi video di YouTube. Gambar 3.16 mencantumkan beberapa produk layanan yang dikembangkan di Google. Tim juga telah bekerja mengembangkan aplikasi Smart Machine seluler dan tertanam, dan mulai dengan layanan Android dan kemudian iOS. Selain itu, tim otak telah bekerja dengan Google X dan Quantum Artificial Intelligence Lab di NASA dalam program luar angkasa bersama. Pada Mei

2016, Google mengumumkan chip ASIC (sirkuit terintegrasi khusus aplikasi) khusus yang mereka buat khusus untuk *Machine learning* yang disesuaikan untuk pemrograman TensorFlow.

Mereka memasang TPU di dalam pusat data mereka selama lebih dari satu tahun, dan mencapai peningkatan kecepatan sepuluh kali lipat dalam operasi *Machine learning*. TPU adalah papan sirkuit kecil yang dimasukkan ke dalam hard drive yang terpasang ke server Google. TPU adalah chip yang telah dilatih sebelumnya untuk mendukung computing TensorFlow dengan aritmatika presisi rendah (misalnya 8 bit) volume tinggi. TPU memang dapat diprogram saat program TensorFlow berubah. Beberapa TPU digunakan dalam pertandingan AlphaGo yang akan dibahas di Bagian 9.4. TensorFlow menawarkan platform perangkat lunak untuk aplikasi *Deep learning* yang akan dibahas di Bagian 7.5. TPU mempercepat computing TensorFlow. Dukungan Android untuk TensorFlow tersedia untuk eksekusi seluler. Dukungan iOS akan segera hadir. Intel juga telah mengoptimalkan prosesor server kelas atas mereka untuk computing saraf. Kami akan menilai proyek Google DeepMind di Bab 9.



Gambar 3.16 Janji *Deep learning* di Proyek Otak Google (Dicetak ulang dengan izin dari presentasi slide publik oleh Jeff Dean, 2016).

Adalah adil untuk mengatakan bahwa jaringan saraf dalam memainkan peran penting dalam memahami ucapan, gambar, bahasa, dan aplikasi penglihatan. Model atau API yang telah dilatih sebelumnya harus memiliki overhead yang rendah dan mudah digunakan dalam pengembangan sistem ML. Berikut ini, kami memeriksa beberapa sistem *Deep learning* yang dikembangkan oleh Tim Google Brain untuk berbagai aplikasi *Big data*, seperti yang ditunjukkan pada Gambar 3.17. Di antaranya, kami melihat aplikasi DL di aplikasi Android, penemuan obat, gmail, pemahaman gambar, peta, pemahaman bahasa alami, foto, penelitian robotika, pidato, dan YouTube di antara banyak lainnya. Di antara 50 tim pengembangan produk internal di Google, minat menggunakan *Deep learning* telah diukur dengan jumlah direktori proyek unik yang berisi file deskripsi model.

Di banyak produk/area:

Android

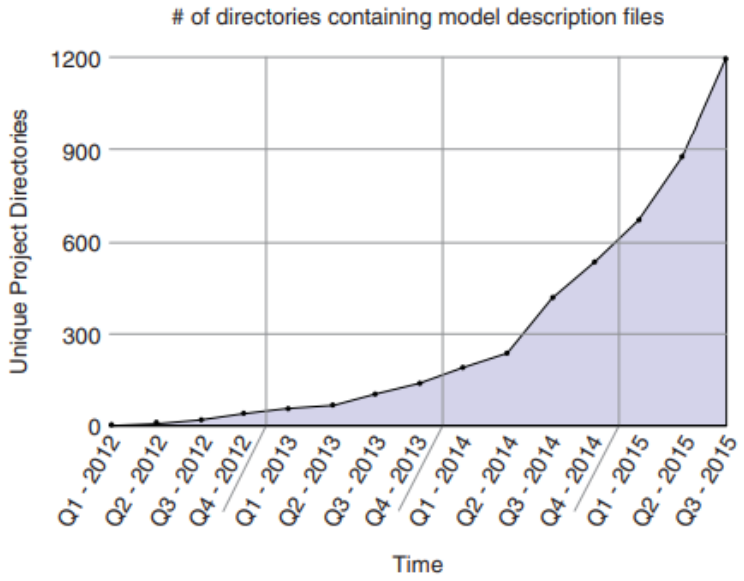
Aplikasi penemuan obat

Gmail

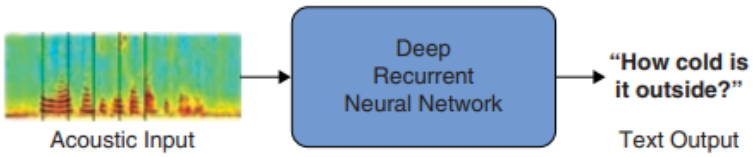
Pemahaman gambar

peta

Pemahaman bahasa alami
Foto
Riset robotika
Pidato
Terjemahan
Youtube
... banyak lainnya ...



Gambar 3.17 Meningkatnya penggunaan *Deep learning* di tim Google (Dicetak ulang dengan izin dari presentasi publik oleh Jeff Dean, 2016).



Gambar 3.18 Konsep sistem pengenalan suara Google yang dibangun dengan jaringan saraf berulang yang dalam.

Secara umum, tiga pendekatan diadopsi dalam aplikasi computing kognitif:

1) Menerapkan pustaka perangkat lunak di cloud atau superkomputer untuk *Machine learning* dan studi neuroinformatika.
2) Menggunakan representasi dan algoritma untuk menghubungkan input dan output dari komputer syaraf tiruan.
3) Merancang chip saraf perangkat keras untuk mengimplementasikan komputer mirip otak untuk *Machine learning* dan kecerdasan.

Pada Gambar 3.18, kami mendemonstrasikan ide membangun sistem pengenalan suara Google dengan jaringan saraf berulang yang dalam (DRNN).

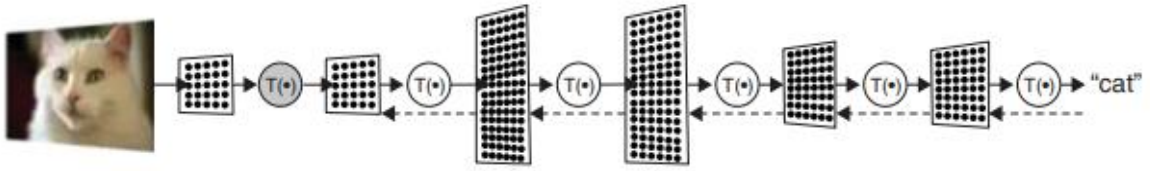
Di sini, sinyal suara akustik dimasukkan ke dalam sistem sebagai input. Melalui pembelajaran berulang dari sistem DRNN, output teks berupa pertanyaan seperti: “Seberapa dingin di luar?” dihasilkan secara otomatis. Sistem Siri Apple juga telah membangun kemampuan percakapan seperti itu. Jaringan saraf konvolusi dalam telah terbukti sangat berguna untuk tujuan ini juga. Selain itu, pengenalan dan deteksi objek sama pentingnya. Ini adalah bagian dari bidang tradisional pengenalan pola dan domain pemrosesan gambar. Konvolusi yang lebih dalam dan deteksi objek yang skalabel menawarkan pendekatan yang layak untuk memecahkan masalah di cloud modern.

Terjemahan mesin dapat dilakukan dengan proses pembelajaran sequence-to-sequence dengan jaringan syaraf tiruan. Terjemahan mesin saraf telah berhasil di Google. Pemodelan bahasa juga dilakukan dengan benchmark 1 miliar kata. Area menarik lainnya adalah tata bahasa parsing otomatis sebagai bahasa asing. Seluruh tujuan ANN adalah untuk mempelajari fungsi yang rumit dari data. JST telah menjadi bidang penelitian panas selama setidaknya 30 tahun.

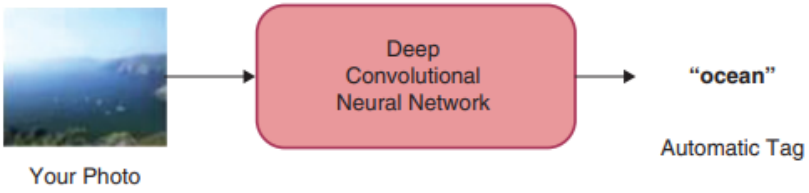
Contoh 3.11 Penggunaan ImageNet Google untuk Pemahaman Gambar

Model ANN, terutama jaringan saraf convolutional dalam (DCNN), telah bereinkarnasi. Mereka menawarkan kumpulan fungsi matematika sederhana yang dapat dilatih, yang kompatibel dengan banyak varian *Machine learning*. Gambar 3.19 menunjukkan ide di balik penggunaan DCNN untuk mengenali "kucing" atau "laut" dari ribuan kelas melalui jutaan gambar foto. Keterangan gambar sangat diminati dalam permintaan pencarian Google.

Mencari foto pribadi tanpa tag sama dengan tugas mengidentifikasi 1 gambar dari 1000 kelas yang berbeda. Pekerjaan itu dilakukan di Google menggunakan ImageNet. Proyek lain yang menggunakan GoogleNet juga menekankan pendekatan konvolusi yang lebih dalam di area awal. Jaringan saraf telah membuat kemajuan pesat dalam pengenalan gambar. Proyek ImageNet menantang banyak tugas klasifikasi. Tim Inception yang menggunakan GoogLeNet hanya mencapai kesalahan 6,66% pada tahun 2014.



(a) Mengenali kucing (bukan anjing)



(b) Membedakan pemandangan laut dari banyak pemandangan lainnya

Gambar 3.19 Menggunakan jaringan saraf konvolusi dalam untuk memahami gambar tertentu dari jutaan foto milik kelas yang berbeda atau serupa.

Bidang menarik lainnya adalah menggabungkan visi dengan terjemahan atau menggabungkan visi dengan kecerdasan robotika. Dengan kata lain, kami ingin robot melalui proses pembelajaran yang mendalam dengan interaksi skala besar. Ini sangat penting untuk proyek mengemudi kendaraan otonom, yang secara aktif dikejar di Google, Baidu, dan pusat penelitian lainnya. Misalnya, kami ingin robot mempelajari koordinasi tangan-mata melalui sistem *Deep learning* di atas mobil. Beberapa kemajuan telah ditunjukkan dalam beberapa proyek yang sedang berlangsung di AS dan Cina.

Konteks IoT untuk Layanan Kognitif

Dalam praktiknya, penyebaran layanan kognitif bergantung pada konteks yang berbeda, seperti informasi lokasi yang digunakan oleh layanan yang ditawarkan melalui Internet untuk memberikan penyesuaian yang sadar lokasi kepada pengguna. Setelah perangkat seluler (ponsel dan tablet) menjadi bagian populer dan integral dari kehidupan sehari-hari, informasi konteks (gravitasi, vektor rotasi, orientasi, medan geomagnetik, kedekatan, cahaya, tekanan, kelembaban dan suhu, dll.) dikumpulkan dari sensor dibangun ke dalam perangkat (misalnya accelerometer, giroskop, GPS, dan oksimetri nadi, dll) digunakan untuk menyediakan fungsionalitas konteks-sadar. Misalnya, sensor bawaan digunakan untuk menentukan aktivitas pengguna, pemantauan lingkungan, kesehatan dan kesejahteraan, lokasi, dan sebagainya.

Informasi konteks hari ini dikumpulkan melalui layanan jejaring sosial (misalnya Facebook, Myspace, Twitter dan WeChat, dll.) menggunakan perangkat seluler. Beberapa aplikasi kontekstual dikembangkan untuk prediksi aktivitas, rekomendasi dan bantuan pribadi. Misalnya, aplikasi seluler mungkin menawarkan informasi lokasi yang diambil dari ponsel untuk merekomendasikan restoran terdekat yang mungkin disukai pelanggan potensial. Contoh lain adalah pendingin yang terhubung ke Internet. Pengguna dapat memeriksa makanan di lemari es dari jarak jauh dan memutuskan apa yang akan dibeli dalam perjalanan pulang.

Ketika pengguna meninggalkan tempat kerjanya, aplikasi secara mandiri melakukan belanja dan memandu pengguna ke pasar belanja tertentu sehingga ia dapat mengambil barang yang telah dipesannya. Untuk melakukan tugas tersebut, aplikasi harus menggabungkan data lokasi, preferensi pengguna, prediksi aktivitas, jadwal pengguna, informasi yang diambil melalui lemari es (yaitu daftar belanja) dan banyak lagi. Berdasarkan contoh di atas, jelaslah bahwa kompleksitas pengumpulan, pemrosesan, dan penggabungan informasi telah meningkat dari waktu ke waktu. Jumlah informasi yang dikumpulkan untuk membantu pengambilan keputusan juga meningkat secara signifikan.

Di era IoT, akan ada banyak sekali sensor yang menempel pada benda sehari-hari. Objek-objek ini akan menghasilkan sejumlah besar data sensorik yang harus dikumpulkan, dianalisis, digabungkan, dan diinterpretasikan. Data sensorik yang dihasilkan oleh sensor tunggal tidak akan memberikan informasi yang diperlukan yang dapat digunakan untuk memahami situasi sepenuhnya. Oleh karena itu, data yang dikumpulkan melalui beberapa sensor perlu digabungkan. Untuk mencapai fusi data sensor, konteks perlu diberi tag bersama dengan data

sensorik untuk diproses dan dipahami nanti. Oleh karena itu, anotasi konteks memainkan peran penting dalam penelitian computing yang sadar konteks.

Konteks IoT

Teknologi kontekstual menyediakan metodologi untuk mengevaluasi kinerja solusi IoT. Evaluasi ini terutama didasarkan pada tiga fitur konteks-sadar di tingkat tinggi: i) pemilihan dan presentasi konteks-sadar; ii) eksekusi kontekstual; dan iii) penandaan kontekstual. Namun, kami juga telah memperkaya kerangka evaluasi dengan mengidentifikasi sub-fitur di bawah tiga fitur yang disebutkan di atas. Pada Tabel 3.7 kami memberikan contoh untuk mengevaluasi solusi IoT dalam domain aplikasi kota pintar.

Data konteks utama yang ditangkap oleh solusi IoT tercantum di bawah ini: W menunjukkan berbasis Web; M menunjukkan berbasis Seluler; D menunjukkan berbasis Desktop; dan O menunjukkan Object-based.

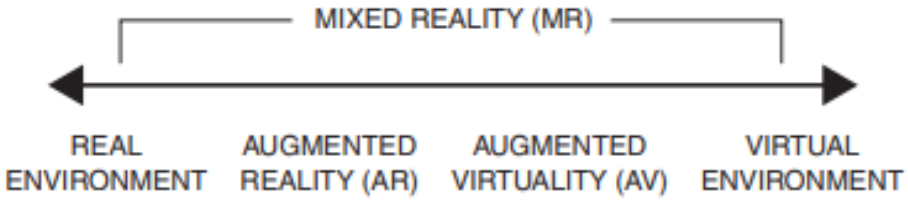
Tabel 3.7 Representatif konteks IoT dalam aplikasi kota pintar.

Proyek IoT, Pembangun dan (Situs Web)	**Konteks Utama**	**Konteks Sekunder**	**Saluran Presentasi**	**Interaksi pengguna**	**Arsip Waktu Nyata**	**Mekanisme Notifikasi**	**Kemampuan belajar**	**Eksekusi Pemberitahuan**
Pengelolaan Sampah Evevo (enevo.com)	Tingkat pengisian limbah	Rute yang efisien untuk mengambil sampah	W	M	RT, A	N,R	ML,UD	E
Lokalisasi Dalam Ruangan, Estimote (estimote.com)	Kekuatan sinyal Bluetooth, ID Beacon	Lokasi, Jarak	M	M	RT	N,R	UD	T,S,E
Manajemen Slot Parkir, ParkSight (streetline.com)	Tingkat suara, Suhu permukaan jalan	Rute untuk slot parkir gratis	M, W	M	RT, A	N,R	ML,UD	T,S,E
Penerangan Jalan, Tvilight (tvilight.com)	Cahaya, kehadiran, cuaca, acara	Penggunaan energi, pola, lampu, dll	W	M	RT, A	N,A	ML,UD	T,S,E
Analisis Gerakan, Ketuk Adegan (scenetap.com)	GPS, Video	Profil kerumunan berdasarkan lokasi	M,W,D	M	RT	N,A	ML	T, S
Pemantauan Lalu Lintas Kaki (scanalyticsinc.com)	tingkat lantai	Gerakan pelacakan peta panas	W	T, M	RT, A	n	ML,UD	S,E
Analisis Ramai, Penghidupan (livehoods.org)	Layanan cloud check-in Foursquare	Dinamika sosial, kota-kota besar	W	M	RT, A	-	ML	E

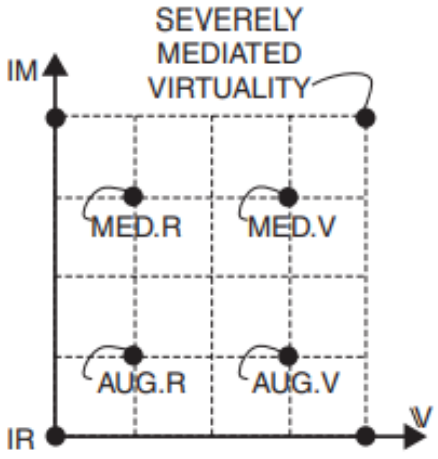
Kami mengidentifikasi Touch (T), Gesture (G) dan Voice (V) sebagai tiga mekanisme umum. M artinya interaksi dilakukan melalui PC atau telepon pintar. RT menyatakan bahwa solusi IoT memproses data secara real time, sedangkan A berarti solusi IoT memproses data arsip. Notasi lain pada Tabel 3.7 kami memiliki S untuk penginderaan IoT, E untuk energi, UD untuk perangkat pengguna, R untuk teknologi radio dan N untuk notifikasi. Kami memperkenalkan nama proyek IoT di kolom paling kiri. Tautan halaman web adalah referensi paling andal untuk solusi IoT yang diberikan. Tautan semacam itu memungkinkan pembaca untuk menyelidiki lebih jauh dan menjelajahi teknologi IoT dengan lebih bermakna.

Aplikasi Augmented dan Virtual Reality

Virtual reality (VR) adalah teknologi komputer yang mereplikasi lingkungan, nyata atau imajiner, dan mensimulasikan kehadiran fisik dan lingkungan pengguna untuk memungkinkan interaksi pengguna. Realitas virtual secara artifisial menciptakan pengalaman sensorik, yang dapat mencakup penglihatan, sentuhan, pendengaran, dan penciuman. Lingkungan imersif harus serupa dengan dunia nyata untuk menciptakan pengalaman yang nyata – misalnya, dalam simulasi untuk pelatihan pilot atau tempur, dapat berbeda secara signifikan dari kenyataan, seperti dalam game VR (Gambar 3.20).



(a) Spektrum realitas campuran



(b) Empat lingkungan

Gambar 3.20 Spektrum dari lingkungan nyata ke AR, AV dan VR.

Augmented reality (AR) adalah tampilan langsung dari fisik, lingkungan dunia nyata yang elemennya ditambah (atau ditambah) oleh input sensorik yang dihasilkan komputer seperti suara, video, grafik, atau data GPS. Hal ini terkait dengan konsep yang lebih umum yang disebut realitas termediasi. Akibatnya, teknologi berfungsi dengan meningkatkan persepsi kita saat ini tentang realitas. Sebaliknya, realitas virtual menggantikan dunia nyata dengan dunia simulasi. Augmentasi secara konvensional dalam waktu nyata dan dalam konteks semantik dengan elemen lingkungan. Realitas campuran berada di mana saja di antara ekstrem kontinuitas virtualitas, membentang dari realitas lengkap hingga lingkungan virtual lengkap dengan augmented reality dan augmented virtuality bercampur menjadi satu.

Pada Gambar 3.20, empat contoh poin ditampilkan: augmented reality, augmented virtuality, mediated reality dan mediated virtuality pada sumbu virtuality dan mediality. Ini termasuk, misalnya, realitas yang berkurang (misalnya helm las terkomputerisasi yang menyaring dan mengurangi bagian tertentu dari suatu pemandangan), akselerometer, girometer, sensor jarak, dan sensor cahaya adalah headset VR bawaan, termasuk HTC Vive, Paystation VR dan Samsung Gear VR, dll.

Video Game

Penggunaan grafis, suara dan teknologi input dalam video game dapat dimasukkan ke dalam Virtual Reality. Beberapa VR head mount display (HMD) telah dirilis untuk bermain game, termasuk Virtual Boy yang dikembangkan oleh Nintendo dan iGlasses yang dikembangkan oleh Virtual I-O. Beberapa perusahaan sedang mengerjakan generasi baru headset VR: Oculus Rift adalah layar yang dipasang di kepala untuk tujuan permainan, yang diakuisisi oleh Facebook pada tahun 2014. Salah satu pesaingnya dinamai oleh Sony sebagai PlayStation VR (dengan nama kode Morpheus). Valve Corporation mengumumkan kemitraan mereka dengan HTC Vive untuk membuat headset VR yang mampu melacak posisi tepat penggunanya. Produk AR/VR lainnya dapat dilihat pada Tabel 3.8.

Tabel 3.8 Produk AR/VR terbaru yang dikembangkan oleh perusahaan teknologi tinggi.

Perusahaan	Produk	Pengantar
Microsoft	HoloLens	Sepasang kacamata pintar yang dipasang di kepala realitas campuran oleh Microsoft. HoloLens mendapatkan popularitas karena menjadi komputer pertama yang menjalankan platform Windows Holographic.
Google	Google Karton	Ini adalah platform VR oleh Google untuk digunakan dengan head mount untuk ponsel pintar. Dinamakan karena penampil karton lipatnya, ini adalah sistem berbiaya rendah untuk mendorong aplikasi VR.
Facebook	Oculus Rift	Oculus Rift adalah headset realitas virtual yang dikembangkan dan diproduksi oleh Oculus VR, dirilis 28 Maret 2016.

Samsung	Gear VR	Samsung Gear VR adalah headset realitas virtual seluler yang dikembangkan oleh Samsung Electronics, bekerja sama dengan Oculus, dan diproduksi oleh Samsung.
Sony	PlayStation VR	Dikenal dengan nama kode Project Morpheus selama pengembangan, adalah tampilan headmount gaming VR yang dikembangkan oleh Sony Interactive Entertainment dan diproduksi oleh Sony.
HTC	HTC VIVE	Ini adalah headset realitas virtual yang dikembangkan oleh HTC dan Valve Corporation pada tahun 2016. Ini dirancang untuk memanfaatkan teknologi "skala ruangan" untuk mengubah ruangan menjadi ruang 3-D melalui sensor
Huawei	Huawei VR	Huawei honor VR dirilis pada 10 Mei 2016 untuk menandingi ponsel pintar honor V8.
Alibaba	Beli + Paket	Program Beli+ menggunakan teknologi VR untuk menghasilkan lingkungan belanja 3-D interaktif dengan sistem grafis komputer dan sensor tambahan.

Pendidikan dan Pelatihan

Langkah-langkah sedang dibuat di bidang pendidikan, meskipun masih banyak yang harus dilakukan. Kemungkinan VR dan pendidikan tidak terbatas dan membawa banyak keuntungan bagi siswa dari segala usia. Beberapa membuat konten yang dapat digunakan untuk tujuan pendidikan, dengan sebagian besar kemajuan dibuat di industri hiburan, tetapi banyak yang memahami dan menyadari masa depan dan pentingnya pendidikan dan VR. Personel Angkatan Laut AS menggunakan simulator pelatihan parasut VR. Penggunaan VR dalam perspektif pelatihan adalah untuk memungkinkan para profesional melakukan pelatihan di lingkungan virtual di mana mereka dapat meningkatkan keterampilan mereka.

3.6 KESIMPULAN

IoT telah muncul dengan cepat untuk mempengaruhi aktivitas kehidupan kita sehari-hari. Tujuan utamanya adalah untuk membangun web fisik di seluruh dunia yang menghubungkan semuanya bersama-sama. IoT mengarah ke ruang kelas pintar, rumah sakit, pasar, department store, jalan raya, jalan raya, kota, dan Bumi. Dengan kata lain, kami ingin menerapkan Smart Machine di mana pun untuk membantu atau mempromosikan keselamatan, kenyamanan, kemudahan, dan produktivitas manusia. Ilmu dan layanan kognitif juga muncul. Baik augmented reality (AR) dan virtual reality (VR) dapat muncul sebagai perangkat komersial untuk memperluas pengalaman manusia dalam bermain game, relaksasi, dan kreativitas.

Bab ini membahas kemajuan teknologi utama yang memungkinkan penggunaan perangkat pintar, sensor, tag, ponsel, tablet, GPS, dll. di mana pun dan kapan pun. Itulah impian surga di Bumi. Kami telah menunjukkan cara menerapkan kemampuan penginderaan IoT dalam computing cloud dengan *Big data*. Chip CPU berbasis saraf muncul di laboratorium penelitian IBM dan di Institute of Computing Technology di China Academy of Sciences. Penggunaan komputer model neuron untuk *Machine learning* masa depan dan *Deep learning* bukan lagi mimpi. Sangat masuk akal untuk memiliki aplikasi inovatif yang melibatkan penginderaan IoT dan kognisi mesin dalam waktu dekat.

Tugas dan Latihan

1. Jawab dua pertanyaan berikut tentang perkembangan IoT dalam beberapa tahun terakhir:
   a) Teknologi IoT awal meliputi 4 T yaitu Telemetri, Telemetering, Telenet dan Telematika. Lakukan survei literatur tentang kemajuan terkini dalam IoT dan laporkan temuan Anda.
   b) Bagaimana perangkat atau teknik berikut: sensor, ponsel pintar, RFID label/pembaca, kode batang, kode QR 2D, atau jam tangan pintar digunakan untuk pengumpulan dan penginderaan data dalam aplikasi IoT?
2. Jawab dua penilaian terbaru tentang teknologi GPS dan empat sistem yang dibangun di AS, Rusia, UE, dan China berikut ini. Wikepedia mungkin merupakan sumber informasi cepat yang bagus:
   a) Apa perbedaan aplikasi sipil dan militer dari berbagai sistem GPS?
   b) Laporkan beberapa aplikasi sipil yang menarik dari layanan GPS.
   c) Apa potensi aplikasi militer dari kemampuan GPS?
3. Merancang sistem perawatan kesehatan yang terdiri dari sensor tubuh dan perangkat yang dapat dipakai untuk mengumpulkan sinyal fisiologis manusia. Sistem ini harus memiliki fungsi sebagai berikut: pemantauan waktu nyata, prediksi penyakit, dan deteksi dini penyakit kronis. Selain itu, Anda mungkin memerlukan sistem pemantauan dan manajemen yang dapat mengoptimalkan distribusi sumber daya medis dan memfasilitasi pembagian data untuk sumber daya tersebut.
4. Dalam beberapa tahun terakhir, analisis video menjadi topik hangat, terutama untuk pemeriksaan keamanan melalui pelacakan video, yang berguna untuk melindungi keselamatan pribadi dan properti. Teknologi keamanan tradisional menekankan respons waktu nyata dan efektivitas verifikasi. Jadi presentasi video dengan resolusi tinggi, tanpa kehilangan dan penundaan rendah telah menjadi arah pengembangan utama industri keamanan selama beberapa tahun terakhir. Saat ini, kita dapat melihat kamera untuk pengawasan kota di mana-mana.

   Dengan meningkatnya penggunaan kamera definisi tinggi, cara mengirimkan data video dalam jumlah besar secara efektif telah menjadi isu utama. Selain itu, melacak penjahat

untuk mendapatkan informasi lokasi mereka memakan waktu dan tenaga. Jelaskan cara menggunakan kecerdasan buatan dan teknologi *Machine learning* untuk menganalisis sampel video besar-besaran, melacak target secara otomatis, dan menemukan jalur bergerak.

5. Penyakit Parkinson (PD) adalah penyakit kronis yang disebabkan oleh gangguan gerak pada sistem saraf pusat. Biasanya, gaya berjalan merupakan indikator penting untuk mengidentifikasi dan mengevaluasi PD. Untuk mengevaluasi perubahan gaya berjalan pada lansia dengan PD secara terus menerus tanpa campur tangan manusia, tekanan langkah kaki dapat diukur saat pasien PD berjalan, dan mode pusat tekanan (CoP) dapat diperoleh. Coba cari tahu perbedaan CoP antara orang normal dan pasien PD.
   Pernyataan mana yang benar?
   a) Sensor tekanan ditempatkan di bawah kaki pasien PD.
   b) Sensor tekanan dipasang di tanah.
   c) Untuk mendapatkan CoP, tekanan bagian depan, tengah atau belakang kaki harus dikumpulkan.
   d) Ukur data tekanan saat pasien PD berdiri atau berjalan.
6. Insiden leukemia di kalangan anak muda meningkat, yang membutuhkan transplantasi sel punca sebagai pengobatan wajib. Setelah transplantasi, pasien harus tinggal di rumah selama 12 hingga 24 bulan. Untuk menghindari perasaan sulit dan tidak menyenangkan pasien selama rehabilitasi, sistem video dirancang untuk membantu komunikasi antara pasien dan tim medis melalui ponsel pintar, tablet, atau komputer pribadi. Sedangkan data pribadi pasien dapat dengan mudah diakses melalui sistem berbasis web. Terutama, jika elemen permainan ditambahkan dalam sistem pengambilan data jarak jauh seperti itu, suasana hati pasien dapat ditingkatkan selama laporan harian.
   Dengan data kesehatan yang lebih praktis dan sering, tim medis dapat memantau status kesehatan pasien secara lebih akurat dan tepat waktu serta memberikan pengobatan yang lebih efektif.
   1) Tentang sistem video, pernyataan mana yang benar?
   a. Kami membutuhkan kerangka data yang sangat fleksibel, untuk memenuhi persyaratan tentang parameter kesehatan kustom.
   b. Sumber data eksternal ditransfer melalui bus layanan data e-health ke database.
   c. Data hanya bisa sulit untuk didefinisikan, bukan definisi lunak.
   d. Game diprioritaskan dengan penggunaan ponsel pintar dan tablet, tetapi juga dapat dilakukan di browser web.
   2) Alur kerja sistem video game mencakup tiga langkah: definisi data, buat file konfigurasi, dan rencanakan tugas permainan. Saat membagikan permainan kecil kepada pasien, tugas dapat menentukan sekelompok pasien dengan praktik terapi

fisik sesuai dengan status kesehatan mereka yang dievaluasi oleh tim medis. Tuliskan pendapat Anda tentang tiga langkah di atas.

7. Merancang sistem manajemen kendaraan cerdas berbasis IoT, khususnya teknologi RFID. Sistem harus memungkinkan pembayaran otomatis, sehingga kendaraan dapat melewati persimpangan tanpa berhenti. Ketika kendaraan keluar, biaya parkir dipotong secara otomatis.
8. Masalah ini terkait dengan penggunaan IoT untuk mempromosikan pertanian hijau.
   1) Berdasarkan studi penelitian kepustakaan, uraikan masing-masing persyaratan sebagai berikut:
      a. Pengumpulan parameter lingkungan pertanian secara real-time seperti suhu, kelembapan, penerangan, suhu tanah, kelembapan tanah, dan kadar oksigen di rumah kaca atau hamparan air, dll.
      b. Keputusan cerdas waktu nyata tentang pertumbuhan tanaman, dan otomatis membuka atau menutup peralatan kontrol lingkungan. Penyebaran sistem memberikan dasar ilmiah dan sarana yang efektif untuk pemantauan pertanian, kontrol otomatis dan manajemen cerdas.
      c. Sistem akan menyimpan dan menganalisis data pemantauan real-time di server untuk secara otomatis membuka atau menutup perangkat yang ditentukan, seperti penyiraman remote control, sakelar rana, penambahan oksigen atau CO2, dll.
   2) Tentang solusi membangun sistem pertanian cerdas, diskusikan caranya untuk mengimplementasikan setiap solusi dengan teknologi nirkabel, sensor, dan GPS terkini:
      a. Teknologi jaringan sensor nirkabel diterapkan dalam sistem pertanian cerdas untuk mencapai pengumpulan dan kontrol data.
      b. Rumah kaca pertanian cerdas yang dilengkapi dengan sensor nirkabel untuk memantau parameter lingkungan seperti suhu udara/tanah, kelembapan, kelembapan, cahaya, dan konsentrasi CO2.
9. Jawab dua penilaian terbaru tentang teknologi GPS dan empat sistem yang dibangun di AS, Rusia, UE, dan Cina berikut ini. Wikepedia mungkin merupakan sumber informasi cepat yang bagus:
   a. Apa perbedaan antara aplikasi sipil dan militer dari berbagai sistem GPS?
   b. Laporkan beberapa aplikasi sipil yang menarik dari layanan GPS.
   c. Apa potensi aplikasi militer dari kemampuan GPS?
10. Kami telah mempelajari tiga aplikasi IoT dalam Contoh 3.1 hingga 3.8. Lakukan investigasi dan cari aplikasi IoT lain yang bermakna. Kirimkan laporan yang diselidiki dengan kedalaman yang sama seperti pada contoh. Gali sebanyak mungkin informasi teknis dari literatur atau sumber lain. Laporkan fitur IoT yang menarik, kemajuan perangkat keras

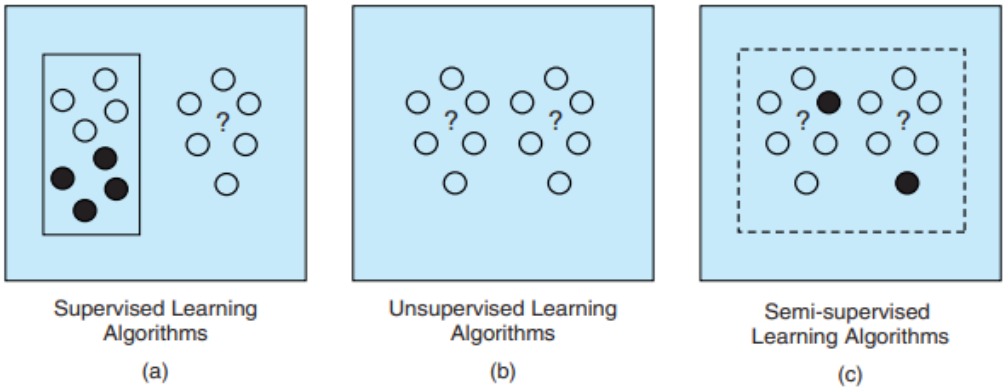
dan perangkat lunak, model interaksi yang diterapkan, dan hasil kinerja yang tersedia baik secara kuantitatif maupun kualitatif. Jangan membuat laporan melambaikan tangan. Segala sesuatu yang Anda laporkan harus dibuktikan dengan bukti dan analisis.

BAGIAN 2
MACHINE LEARNING DAN ALGORITMA *DEEP LEARNING*
BAB 4
ALGORITMA *MACHINE LEARNING* YANG DIAWASI

4.1 TAKSONOMI ALGORITMA *MACHINE LEARNING*

Machine learning (ML) adalah disiplin yang dapat ditindaklanjuti yang diperluas dari studi pengenalan pola dan teori pembelajaran computing dalam kecerdasan buatan (AI). Bidang ini sangat relevan dengan pengambilan keputusan statistik dan penambangan data dalam membangun AI atau sistem pakar. Ide utamanya adalah menggunakan komputer untuk belajar dari data. Untuk data yang membosankan atau tidak terstruktur, mesin sering kali dapat membuat keputusan yang lebih baik dan lebih tidak memihak daripada pembelajar manusia. Untuk tujuan ini, kita perlu menulis program komputer berdasarkan model algoritma. Belajar dari objek data yang diberikan, kita dapat mengungkapkan kelas kategoris atau afiliasi pengalaman dari data masa depan yang akan diuji. Konsep ini pada dasarnya mendefinisikan ML sebagai istilah operasional daripada istilah kognitif.

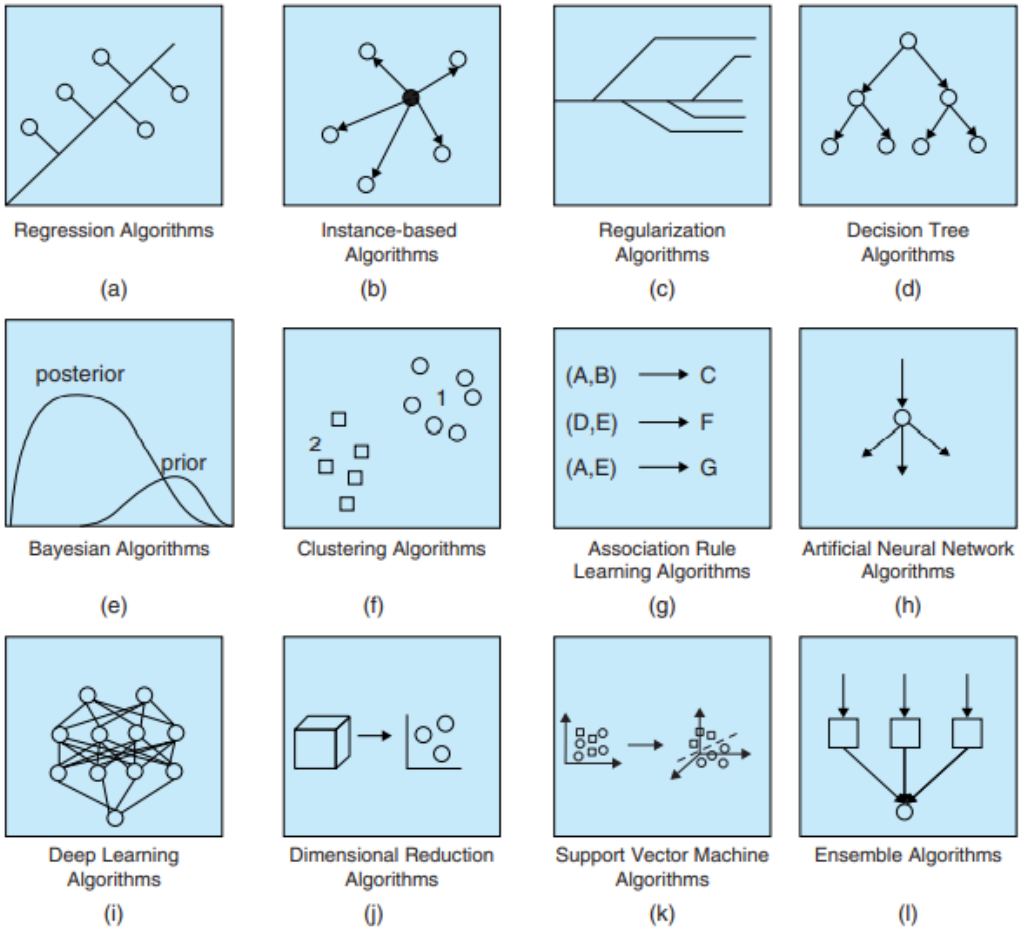
Untuk mengimplementasikan tugas ML, kita perlu menjelajahi atau membangun algoritme komputer untuk belajar dari data dan membuat prediksi pada data berdasarkan fitur, kesamaan, atau korelasi spesifiknya. Algoritma ML dioperasikan dengan membangun model pengambilan keputusan dari input data sampel. Keluaran dari model ML adalah prediksi atau keputusan berbasis data. Di Bagian 4.1.1, kami mengklasifikasikan algoritme ML berdasarkan gaya belajarnya. Gaya dapat berupa pendekatan yang diawasi menggunakan beberapa data pelatihan, atau pendekatan tanpa pengawasan yang mengeksplorasi struktur tersembunyi dalam data tanpa data pelatihan. Di Bagian 4.1.2, kami mengelompokkan algoritme ML berdasarkan kesamaannya dalam bentuk dan fungsionalitas. Baik metode ML terawasi maupun tak terawasi masuk akal dalam aplikasi kehidupan nyata.



Gambar 4.1 Algoritma *Machine learning* yang dikelompokkan berdasarkan gaya belajar yang berbeda.

***Machine learning* Berdasarkan Gaya Belajar**

Algoritma ML dapat dibangun dengan gaya yang berbeda untuk memodelkan suatu masalah. Gaya ditentukan oleh interaksi dengan lingkungan data yang dinyatakan sebagai input ke model. Gaya interaksi data menentukan model pembelajaran yang dapat dihasilkan oleh algoritma ML. Pengguna harus memahami peran data input dan proses konstruksi model. Tujuannya adalah untuk memilih model ML yang dapat menyelesaikan masalah dengan hasil prediksi terbaik. Dalam hal ini, ML terkadang tumpang tindih dengan tujuan penambangan data. Pada Gambar 4.1, kami menunjukkan tiga kelas algoritma ML berdasarkan gaya belajar yang berbeda: terawasi, tidak terawasi, dan semi terawasi.



Gambar 4.2 Algoritma *Machine learning* yang dikelompokkan berdasarkan pengujian kesamaan.

Gaya ini bergantung pada bagaimana data pelatihan digunakan dalam proses pembelajaran:

- **Pembelajaran yang diawasi:** Data input disebut data pelatihan dengan label atau hasil yang diketahui, yang digambarkan oleh dua jenis lingkaran pada Gambar 4.1(a) di dalam kotak data pelatihan. Sebuah model dibangun melalui pelatihan dengan menggunakan

dataset pelatihan. Model ditingkatkan dengan menerima prediksi umpan balik. Proses pembelajaran berlanjut hingga model mencapai tingkat akurasi yang diinginkan pada data pelatihan. Data yang masuk di masa depan (tanpa label yang diketahui) diuji pada model yang dibangun. Kami akan memperkenalkan berbagai algoritma ML yang diawasi di Bagian 4.1.3. Empat algoritma utama yang diawasi ada di bagian selanjutnya.

- **Pembelajaran tanpa pengawasan:** Semua data input tidak diberi label dengan hasil yang diketahui, seperti yang ditunjukkan pada Gambar 4.1(b). Sebuah model dihasilkan dengan menjelajahi struktur yang disajikan dalam data input. Ini dapat dicapai dengan mengekstraksi aturan umum, melalui proses matematis untuk mengurangi redundansi, atau mengatur data dengan pengujian kesamaan. Contoh masalah yang akan dipelajari dalam Bab 5 adalah pengelompokan, pengurangan dimensi dan aturan asosiasi.
- **Pembelajaran semi-diawasi:** Dalam hal ini, data input adalah campuran dari contoh berlabel dan tidak berlabel, seperti yang ditunjukkan pada Gambar 4.1(c). Model harus mempelajari struktur untuk mengatur data untuk membuat prediksi menjadi mungkin. Masalah tersebut dan algoritma ML lainnya akan dijelaskan dalam Bab 5 di bawah asumsi yang berbeda tentang bagaimana memodelkan data yang tidak berlabel.

***Machine learning* Berdasarkan Pengujian Kemiripan**

Algoritma ML dapat dibedakan dengan menerapkan fungsi pengujian kesamaan yang berbeda dalam proses pembelajaran. Misalnya, metode berbasis pohon menerapkan pohon keputusan. Jaringan saraf terinspirasi oleh neuron buatan dalam model otak koneksionis. Kami dapat menangani proses ML secara subyektif dengan menemukan yang paling cocok untuk menyelesaikan masalah keputusan berdasarkan karakteristik dalam kumpulan data yang diproses. Dua belas kategori algoritma ML diperkenalkan secara singkat di bawah ini. Konsep kunci yang mendasari diilustrasikan pada Gambar 4.2.

Beberapa algoritme ML menerapkan data pelatihan, termasuk regresi, pohon keputusan, jaringan Bayesian, dan mesin vektor dukungan. Algoritme tanpa pengawasan lainnya tidak menerapkan kumpulan data pelatihan. Sebagai gantinya, mereka berusaha menemukan struktur atau properti tersembunyi di seluruh kumpulan data input. Ini termasuk metode pengelompokan, analisis asosiasi, pengurangan dimensi dan jaringan saraf tiruan, dll. Beberapa algoritma ML dibahas dalam bab-bab berikutnya, secara selektif.

Banyak ekstensi dari algoritme ML ini akan diringkas di Bagian 4.1.3 dan 4.1.4:

- **Regresi:** ini menawarkan pendekatan terawasi menggunakan pembelajaran statistik, seperti yang diilustrasikan pada Gambar 4.2(a). Regresi memodelkan hubungan antara karakteristik data input. Proses regresi secara iteratif disempurnakan menggunakan kriteria kesalahan untuk membuat prediksi yang lebih baik. Metode ini meminimalkan kesalahan antara nilai prediksi dan pengalaman aktual dalam input data.
- **Pembelajaran berbasis instans:** ini memodelkan masalah keputusan dengan instans atau data pelatihan kritis, seperti yang disorot oleh titik padat pada Gambar 4.2(b). Instance

data dibangun dengan database contoh yang andal. Uji kesamaan dilakukan untuk menemukan kecocokan terbaik untuk membuat prediksi. Metode ini juga dikenal sebagai pembelajaran berbasis memori, karena contoh data yang representatif dan ukuran kesamaan disimpan dalam database.

- **Algoritma regularisasi:** metode ini meluas dari metode regresi yang mengatur model untuk mengurangi kompleksitas. Proses regularisasi ini mendukung model yang lebih sederhana yang juga lebih baik untuk generalisasi. Gambar 4.2(c) menunjukkan bagaimana mengurutkan model prediksi terbaik di antara berbagai pilihan desain.
- **Metode pohon keputusan:** ini menawarkan model keputusan yang ditunjukkan pada Gambar 4.2(d). Model ini didasarkan pada pengamatan nilai target data di sepanjang berbagai node fitur dalam proses keputusan terstruktur pohon. Berbagai jalur keputusan bercabang dalam struktur pohon sampai keputusan prediksi dibuat pada node meninggalkan, secara hierarkis. Pohon keputusan dilatih pada data yang diberikan untuk akurasi yang lebih baik dalam memecahkan masalah klasifikasi dan regresi.
- **Metode Bayesian:** ini didasarkan pada teori keputusan statistik. Hal ini sering diterapkan dalam pengenalan pola, ekstraksi fitur dan aplikasi regresi. Sebuah jaringan Bayesian ditunjukkan pada Gambar 4.2(e), yang menawarkan model grafik asiklik terarah (DAG) yang diwakili oleh satu set variabel acak independen secara statistik. Probabilitas sebelumnya dan probabilitas posterior diterapkan dalam membuat prediksi. Sekali lagi model dapat ditingkatkan dengan penyediaan dataset pelatihan yang lebih baik.
- **Analisis Clustering:** ini adalah metode yang didasarkan pada pengelompokan objek data yang serupa sebagai cluster. Dua cluster ditunjukkan pada Gambar 4.2(f). Seperti regresi, metode ini tidak diawasi dan dimodelkan dengan menggunakan pengelompokan berbasis centroid dan/atau pengelompokan hierarki. Semua metode pengelompokan didasarkan pada pengujian kesamaan.
- **Pembelajaran aturan asosiasi:** ini tidak diawasi dengan data pelatihan. Sebaliknya, metode menghasilkan aturan inferensi yang paling baik menjelaskan hubungan yang diamati antara variabel dalam data. Aturan-aturan ini, seperti yang ditunjukkan pada Gambar 4.2(g), digunakan untuk menemukan asosiasi yang berguna dalam kumpulan data multidimensi yang besar. Pola asosiasi ini sering dimanfaatkan oleh perusahaan bisnis atau organisasi besar.
- **Jaringan Syaraf Tiruan (JST):** ini adalah model kognitif yang terinspirasi oleh struktur dan fungsi neuron biologis, seperti yang ditunjukkan pada Gambar 4.2(h). JST mencoba untuk memodelkan hubungan yang kompleks antara input dan output. Mereka membentuk kelas algoritma pencocokan pola yang digunakan untuk memecahkan masalah *Deep learning*, regresi dan klasifikasi.
- **Metode *Deep learning*:** ini meluas dari jaringan saraf tiruan dengan membangun jaringan saraf yang jauh lebih dalam dan kompleks, seperti yang ditunjukkan pada Gambar 4.2(i).

Jaringan *Deep learning* dibangun dari beberapa lapisan neuron buatan yang saling berhubungan. Mereka sering digunakan untuk meniru proses otak manusia dalam menanggapi sinyal cahaya, suara dan visual. *Deep learning* akan dipelajari di Bab 5. Metode ini sering diterapkan pada masalah pembelajaran semi-terawasi, di mana kumpulan *Big data* berisi sangat sedikit data berlabel.

- **Pengurangan dimensi:** ini mengeksploitasi struktur yang melekat dalam data dengan cara yang tidak diawasi. Tujuannya adalah untuk meringkas atau mendeskripsikan data dengan menggunakan lebih sedikit informasi. Hal ini dilakukan dengan memvisualisasikan data multidimensi dengan komponen atau dimensi utama. Gambar 4.2(j) menunjukkan reduksi dari ruang 3-D e ke ruang data 2-D. Data yang disederhanakan kemudian dapat diterapkan dalam metode pembelajaran terawasi.
- **Support Vector Machines (SVM):** ini sering digunakan dalam metode pembelajaran terawasi untuk aplikasi regresi dan klasifikasi. Gambar 4.2(k) menunjukkan bagaimana hyperplane (permukaan dalam ruang 3-D) dihasilkan untuk memisahkan ruang data sampel pelatihan ke dalam subruang atau kategori yang berbeda. Algoritma pelatihan SVM membangun model untuk memprediksi apakah sampel baru termasuk dalam satu kategori atau yang lain.
- **Metode ensemble:** ini adalah model yang terdiri dari beberapa model yang lebih lemah yang dilatih secara independen. Hasil prediksi dari model-model ini digabungkan pada Gambar 4.2(l), yang membuat prediksi kolektif lebih akurat. Banyak upaya dilakukan untuk menggabungkan tipe pelajar yang lemah dan cara menggabungkan mereka secara efektif. Model ansambel terdiri dari pelajar campuran yang menerapkan algoritma terawasi, tidak terawasi, atau semi terawasi.

Tabel 4.1 merangkum enam kategori algoritma *Machine learning* berdasarkan fungsinya. Regresi polinomial adalah jenis metode regresi yang fungsi pasnya adalah polinomial. Ide dasar dari regresi bertahap adalah untuk membawa variabel dalam model langkah demi langkah, kemudian kita harus melakukan uji-F sebelum memasukkan setiap variabel penjelas untuk mengkonfirmasi apakah variabel ini akan dimasukkan ke dalam model atau tidak. Learning Vector Quantization (LVQ) adalah metode jaringan saraf tiruan berdasarkan model. Ide dari Self-Organizing Map (SOM) adalah bahwa jaringan saraf akan dibagi ke dalam area yang berbeda saat menerima input eksternal, di mana setiap area memiliki refleksi yang berbeda terhadap input dan prosedur ini diselesaikan secara otomatis.

Tabel 4.1 Klasifikasi algoritma *Machine learning* berdasarkan fungsionalitas.

Kategori Fungsional	**Deskripsi Singkat, Algoritma Boldfaced tercakup dalam buku ini**	**Bagian Terkait**
Regresi	Linear, Polinomial, Logistik, Bertahap, Eksponensial, MARS (Multivariate Adaptive Regression Splines)	4.2.1, 4.2.2

Berbasis Instans	KNN (k-nearest Neighbor), Neighboring, LVQ (Learning Vector Quantization), SOM (Self-Organizing Map), LWL (Locally Weighted Learning)	4.3.3, 4.3.4
Jaringan Bayesian	Na¨ıve Bayesian, Gaussian, Multinomial, AODE (Averaged One-Dependence Estimator), BBN (Bayesian Belief Network), BN (Bayesian Network)	4.4.1, 4.3
Kekelompokan	Analisis Clustering, k-Means, Hierarchical Clustering, DBSCAN (Database Scan), Web-Cluster, COBWEB	5.2
Pengurangan Dimensi	PCA (Principal Component Analysis), MDS (Multi-Dimensional Scaling), SVD (Singular Value Decomposition), PCR (Principal Component Regression), dan PLSR (Partial Least Squares Regression)	5.3
Ansambel	Jaringan Neural, Bagging, Adaboost, Hutan Acak	4.4.3, 6.2.1

Gaussian Bayesian adalah sejenis klasifikasi Bayesian yang atributnya kontinu dan mengikuti distribusi Gaussian. Multinomial Bayesian adalah Na¨ıve Bayesian khusus, yang menggunakan distribusi polinomial untuk menghitung probabilitas. Averaged one-dependency estimators (AODE) merupakan salah satu teknik pembelajaran klasifikasi probabilistik. Ini dikembangkan untuk mengatasi masalah independensi atribut dari classifier Bayesian naif yang populer. Metode clustering berbasis grid pertama-tama membagi ruang objek menjadi unit-unit hingga untuk membangun struktur grid, kemudian menyelesaikan pengelompokan dengan struktur grid ini. Metode ini banyak digunakan karena implementasi inkrementalnya mudah dan dapat mengolah data berdimensi tinggi.

COBWEB adalah metode clustering inkremental yang sering digunakan dan sederhana berdasarkan model. Singular Value Decomposition (SVD) adalah metode yang menggunakan dekomposisi nilai singular untuk memperkecil dimensi ruang sampel. Bagging menggunakan metode yang diberikan untuk memilih kombinasi (juga disebut pengklasifikasi lemah) untuk mendapatkan solusi optimal (atau label kelas) untuk meningkatkan akurasi klasifikasi. Adaboost adalah algoritma iteratif yang ide intinya adalah melatih pengklasifikasi lemah yang berbeda untuk set pelatihan yang sama, dan kemudian menggabungkan pengklasifikasi lemah ini untuk membentuk pengklasifikasi akhir yang lebih kuat.

Algoritma *Machine learning* yang Diawasi

Dalam sistem ML yang diawasi, komputer belajar dari set data pelatihan pasangan {input, output}. Masukan berasal dari sampel data yang diberikan dalam format tertentu seperti laporan kredit peminjam. Keluarannya mungkin berbeda, seperti “ya” atau “tidak” untuk aplikasi pinjaman. Outputnya bisa juga berkelanjutan, seperti distribusi probabilitas bahwa pinjaman dapat dilunasi tepat waktu. Tujuan utamanya adalah membuat model ML yang andal yang dapat

memetakan atau menghasilkan output yang benar dari input baru yang tidak terlihat sebelumnya. Sistem ML bertindak seperti fungsi prediktor yang disetel dengan baik g(x). Sistem "pembelajaran" dibangun dengan algoritme canggih untuk mengoptimalkan fungsi ini. Mengingat input data x dalam laporan kredit peminjam, sistem akan secara akurat membuat keputusan pinjaman untuk bank.

Di bagian ini, kami menyajikan empat kelompok algoritme ML terawasi yang penting, seperti yang tercantum dalam Tabel 4.2, di mana yang dicetak tebal akan dipelajari di bagian selanjutnya. Algoritme yang tersisa hanya terdaftar untuk dijelajahi pembaca lebih lanjut. Dalam klasifikasi, input dibagi menjadi dua atau lebih kelas, dan pelajar harus menghasilkan model yang memberikan input tak terlihat ke satu atau lebih kelas ini. Ini biasanya ditangani dengan cara yang diawasi. Pemfilteran spam adalah contoh klasifikasi yang baik, di mana inputnya adalah pesan email (atau lainnya) dan kelasnya adalah "spam" dan "bukan spam". Dalam regresi, juga merupakan masalah yang diawasi, keluarannya kontinu daripada diskrit.

Tabel 4.2 Algoritme *Machine learning* yang diawasi.

Kelas Algoritma ML	**Nama Algoritma, yang dicetak tebal adalah yang tercakup dalam buku ini**	**Bagian Terkait**
Regresi	Linear, Polinomial, Logistik, Stepwise, OLSR (Regresi Kuadrat Terkecil Biasa), LOESS (Penghalusan Scatterplot yang Diperkirakan Secara Lokal), MARS (Multivariate Adaptive Regression Splines)	4.2.2, 4.2.3
Klasifikasi	KNN (k-nearest Neighbor), Trees, Na¨ıve Bayesian, SVM (Support Vector Machine), LVQ (Learning Vector Quantization), SOM (Self-Organizing Map), LWL (Locally Weighted Learning)	4.3.3, 4.4.3
Pohon Keputusan	Pohon keputusan, Hutan Acak, CART (Pohon Klasifikasi dan Regresi), ID3 (Dikotomiser Iteratif 3), CHAID (Deteksi Interaksi Otomatis Chi-kuadrat), ID3 (Dikotomiser Iteratif 3), CHAID (Deteksi Interaksi Otomatis Chi-kuadrat)	4.3.1, 4.4.3
Jaringan Bayesian	Na¨ıve Bayesian, Gaussian, Multinomial, AODE (Averaged One-Dependence Estimator), BBN (Bayesian Belief Network), BN (Bayesian Network)	4.3.3, 4.3

Pohon keputusan digunakan sebagai model prediktif, yang memetakan pengamatan tentang suatu item ke kesimpulan tentang nilai target item tersebut. Mesin vektor pendukung (SVM) dibangun dengan seperangkat metode pembelajaran yang diawasi, juga sering digunakan dalam klasifikasi dan regresi. Jaringan Bayesian adalah model keputusan statistik yang mewakili

satu set variabel acak dan independensi bersyaratnya melalui grafik asiklik terarah (DAG). Misalnya, jaringan Bayesian dapat mewakili hubungan probabilistik antara penyakit dan gejala. Mengingat gejala, jaringan dapat digunakan untuk menghitung probabilitas adanya berbagai penyakit. Banyak algoritma efisien yang melakukan diagnosis medis ada di industri perawatan kesehatan.

Algoritma *Machine learning* Tanpa Pengawasan

Pembelajaran tanpa pengawasan biasanya digunakan dalam menemukan hubungan khusus dalam kumpulan data. Tidak ada contoh pelatihan yang digunakan dalam proses ini. Sebagai gantinya, sistem diberikan satu set data untuk menemukan pola dan korelasi di dalamnya. Tabel 4.3 mencantumkan beberapa algoritme ML yang dilaporkan yang beroperasi tanpa pengawasan. Misalnya, aturan asosiasi dihasilkan dari data input untuk mengidentifikasi kelompok teman yang erat dalam database jaringan sosial.

Tabel 4.3 Beberapa algoritma *Machine learning* tanpa pengawasan.

Kelas Algoritma	**Nama Algoritma ML tanpa pengawasan, algoritma huruf tebal tercakup dalam buku ini.**	**Bagian Terkait**
Analisis Asosiasi	Apriori, Aturan Asosiasi, Eclat, FP-Growth	5.1
Kekelompokan	Analisis Clustering, k-means, Hierarchical Clustering, Ekspektasi Maksimalisasi (EM), Clustering Berbasis Kepadatan	5.2
Pengurangan Dimensi	PCA (Analisis Komponen Utama), Analisis Diskriminan, MDS (Penskalaan Multi-Dimensi)	5.3
Jaringan Syaraf Tiruan (JST)	Perceptron, Back propagation, RBFN (Radial Basis Function Network)	6.2.1

Dalam pengelompokan, satu set input harus dibagi menjadi beberapa kelompok. Tidak seperti klasifikasi terawasi, kelompok tidak diketahui sebelumnya, sehingga ini menjadi tugas yang tidak terawasi. Estimasi densitas menemukan distribusi input di beberapa ruang. Pengurangan dimensi menyederhanakan input dengan memetakannya ke dalam ruang berdimensi lebih rendah. JST muncul di perceptron dan sistem prediksi propagasi balik.

4.2 METODE REGRESI UNTUK *MACHINE LEARNING*

Metode analisis regresi diperkenalkan di bawah ini untuk *Machine learning*. Pertama, kami menyajikan konsep dasar dan asumsi yang mendasarinya. Kemudian kami mempelajari metode regresi linier dan logistik yang telah sering diterapkan dalam *Machine learning*. Baik model matematika maupun contoh numerik diberikan untuk memperjelas ide dan proses pembelajaran yang terlibat.

Konsep Dasar Analisis Regresi

Metode analisis regresi menerapkan statistik matematis untuk menetapkan variabel dependen dan variabel independen dalam proses *Machine learning*. Hal ini pada dasarnya untuk melakukan urutan estimasi parametrik atau non-parametrik. Dengan kata lain, metode menemukan hubungan sebab akibat antara variabel input dan output. Biasanya, fungsi estimasi dapat ditentukan dengan pengalaman menggunakan pengetahuan apriori atau pengamatan visual dari data. Kita perlu menghitung koefisien fungsi yang tidak ditentukan dengan menggunakan beberapa kriteria kesalahan. Selanjutnya, metode regresi dapat diterapkan untuk mengklasifikasikan data dengan memprediksi tag kategori data.

Variabel independen merupakan input dari proses regresi, yang juga dikenal sebagai prediktor. Variabel terikat adalah output dari proses. Tujuan dari analisis regresi adalah untuk memahami bagaimana nilai khas dari variabel dependen berubah ketika variabel independen bervariasi, sedangkan variabel independen lainnya dibiarkan tidak berubah. Dengan demikian, analisis regresi memperkirakan nilai rata-rata variabel dependen ketika variabel independen adalah tetap. Nilai taksiran merupakan fungsi dari variabel bebas yang dikenal sebagai fungsi regresi, yang dapat digambarkan dengan distribusi probabilitas.

Analisis regresi banyak digunakan dalam *Machine learning* untuk prediksi dan peramalan. Hal ini pada dasarnya untuk mengungkapkan hubungan kausal antara variabel independen dan dependen. Kita harus sangat berhati-hati untuk membuat prediksi seperti itu, karena kausalitas dapat menyebabkan ilusi atau hubungan palsu yang pada gilirannya dapat menyesatkan pengguna. Sebagian besar metode regresi bersifat parametrik dan memiliki dimensi terbatas dalam ruang analisis. Dalam buku ini, kita tidak akan membahas analisis regresi nonparametrik, yang mungkin berdimensi tak hingga. Seperti banyak metode *Machine learning* lainnya, akurasi atau kinerja bergantung pada kualitas set data yang digunakan. Hal ini terkait dengan proses pembuatan data dan asumsi yang mendasari yang dibuat. Di satu sisi, regresi menawarkan estimasi variabel respons berkelanjutan, sebagai lcloud dari variabel respons diskrit yang digunakan dalam klasifikasi yang menuntut akurasi yang lebih tinggi.

Dalam perumusan proses regresi, parameter yang tidak diketahui sering dilambangkan sebagai , yang mungkin muncul sebagai skalar atau vektor. Variabel independen dilambangkan sebagai X dan variabel dependen sebagai Y. Ketika beberapa dimensi terlibat, parameter ini berbentuk vektor. Model regresi menetapkan perkiraan hubungan antara X, β dan Y sebagai berikut:

$$Y \approx f(X, \beta) \qquad (4.1)$$

Fungsi f (X, β) biasanya didekati dengan nilai harapan E (Y | X). Fungsi regresi f didasarkan pada pengetahuan tentang hubungan antara Y dan X. Jika tidak ada pengetahuan seperti itu yang tersedia, bentuk praktis yang didekati dipilih untuk f.

Pertimbangkan vektor parameter β yang tidak diketahui memiliki k komponen. Kami memiliki tiga model untuk menghubungkan input ke output, tergantung pada besaran relatif antara jumlah N titik data yang diamati dari formulir (X, Y) dan dimensi k dari ruang sampel:

- Ketika N < k, sebagian besar metode analisis regresi klasik dapat diterapkan. Karena persamaan pendefinisian kurang ditentukan, tidak ada cukup data untuk memulihkan parameter yang tidak diketahui .
- Ketika N = k dan fungsi f linier, persamaan Y = f(X, β) dapat diselesaikan secara eksak tanpa pendekatan, karena terdapat N persamaan untuk menyelesaikan N komponen dalam . Solusinya unik selama komponen X bebas linier. Jika f nonlinier, banyak solusi mungkin ada atau tidak ada solusi sama sekali.
- Secara umum, kita memiliki situasi bahwa N > k titik data. Ini menyiratkan bahwa ada informasi yang cukup dalam data untuk memperkirakan nilai unik untuk dalam situasi yang terlalu ditentukan.

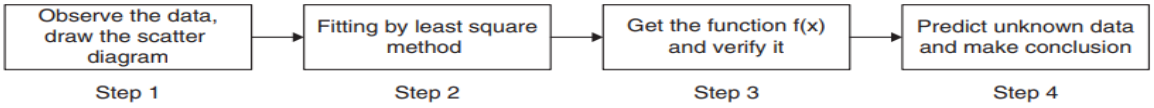
Contoh 4.1 Regresi dengan Serangkaian Pengukuran Independen yang Diperlukan

Contoh ini membantu pembaca untuk memahami jumlah data independen yang diperlukan untuk melakukan analisis regresi pengukuran data kontinu. Pertimbangkan model regresi yang memiliki empat parameter yang tidak diketahui, β_0, β_1, β_3 dan β_4. Misalkan seorang eksperimen melakukan 10 pengukuran semuanya pada nilai yang sama persis dari vektor variabel bebas X = (X_1, X_2, X_3, X_4). Analisis regresi gagal memberikan satu set nilai estimasi yang unik untuk empat parameter yang tidak diketahui. Dengan kata lain, peneliti tidak mendapatkan informasi yang cukup untuk melakukan prediksi regresi.

Dalam keadaan ini, yang terbaik yang dapat kita lakukan adalah memperkirakan nilai rata-rata dan standar deviasi dari variabel dependen Y. Demikian pula, mengukur pada dua nilai X yang berbeda akan memberikan data yang cukup untuk regresi dengan dua yang tidak diketahui, tetapi tidak untuk tiga atau lebih tidak dikenal. Jika eksperimenter telah melakukan pengukuran pada empat nilai yang berbeda dari vektor variabel independen X, maka analisis regresi akan memberikan satu set perkiraan yang unik untuk empat parameter yang tidak diketahui dalam β.

Dalam kasus N > k, kesalahan pengukuran i terdistribusi normal. Terdapat kelebihan informasi yang terkandung dalam pengukuran (N k), yang dikenal sebagai derajat kebebasan regresi. Tercantum di bawah ini adalah tiga asumsi dasar untuk analisis regresi:

1) Sampel mewakili ruang data yang terlibat. Kesalahan adalah variabel acak dengan rata-rata nol bersyarat pada variabel penjelas.
2) Variabel bebas diukur tanpa kesalahan. Prediktor adalah independen linier.
3) Kesalahan tidak berkorelasi dan varians kesalahan konstan di seluruh pengamatan. Jika tidak, metode kuadrat terkecil tertimbang dapat digunakan.



Gambar 4.3 Langkah-langkah utama dalam regresi linier

Regresi Linier untuk Prediksi dan Prakiraan

Analisis regresi adalah metode statistik yang menentukan hubungan kuantitatif di mana dua atau lebih variabel saling bergantung, termasuk regresi linier dan regresi nonlinier. Jika hanya satu variabel independen dan satu variabel dependen yang dimasukkan dalam analisis regresi, dan jika representasi perkiraan untuk hubungan antara keduanya dapat dilakukan dengan garis lurus, analisis regresi semacam ini disebut analisis regresi linier kesatuan. Jika dua atau lebih variabel bebas dimasukkan dalam analisis regresi, dan jika ada hubungan linier antara variabel terikat dan variabel bebas, maka ini disebut analisis regresi linier multivariat. Model regresi linier didefinisikan oleh y = f (X), dimana X = (x_1, x_2, , x_n) adalah vektor multidimensi dan y adalah variabel skalar. Regresi linier ditentukan pada Gambar 4.3 dalam empat langkah.

Analisis Regresi Linier Kesatuan

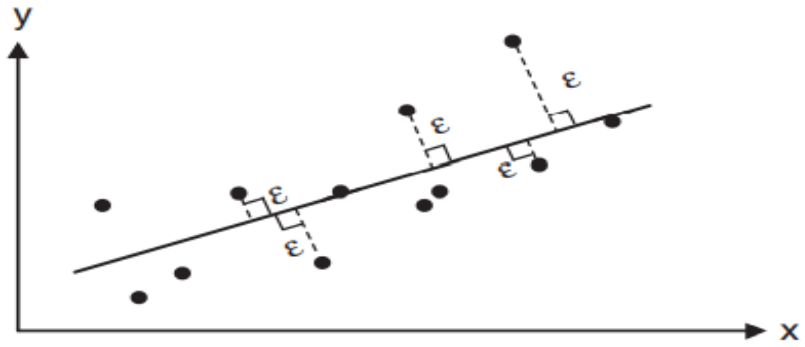
Pertimbangkan satu set elemen data dalam ruang sampel 2-D, (x_1, y_1), (x_2, y_2), , (x_n, y_n). Semua titik dipetakan ke dalam diagram pencar. Jika mereka dapat ditutupi, kira-kira, dengan garis lurus, maka kita memperoleh ekspresi regresi linier berikut:

$$y = ax + b + \varepsilon \tag{4.2}$$

di mana x adalah variabel penjelas, y adalah variabel yang dijelaskan, a dan b adalah koefisien yang sesuai, dan adalah kesalahan acak, yang mengikuti distribusi normal independen dengan distribusi yang sama dengan mean E(ε)dan varians Var(ε) . Kemudian kita perlu menghitung ekspektasi dengan menggunakan ekspresi regresi linier:

$$y = ax + b \tag{4.3}$$

Gambar 4.4 menunjukkan kesalahan residual dari model regresi kesatuan. Tugas utama analisis regresi adalah melakukan estimasi koefisien a dan b melalui pengamatan terhadap n kelompok sampel.



Gambar 4.4 Analisis Regresi Linier Kesatuan.

Metode umum adalah metode kuadrat terkecil, dan fungsi tujuannya diberikan oleh:

$$\min\ Q(\hat{a}, \hat{b}) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} [y_i - E(y_i)]^2 = \sum_{i=1}^{n} (y_i - \hat{a}x_i - \hat{b})^2 \tag{4.4}$$

Untuk meminimalkan jumlah kuadrat, kita perlu menghitung turunan parsial Q untuk â, b̂, dan menjadikannya nol, seperti yang ditunjukkan di bawah ini:

$$\begin{cases} \frac{\partial Q}{\partial \hat{b}} = \sum_{i=1}^{n} (y_i - \hat{a}x_i - \hat{b}) = 0 \\ \frac{\partial Q}{\partial \hat{a}} = \sum_{i=1}^{n} (y_i - \hat{a}x_i - \hat{b})x_i = 0 \end{cases} \xrightarrow{solve} \begin{cases} \hat{a} = \dfrac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \\ \hat{b} = \bar{y} - \hat{a}\bar{x} \end{cases} \quad (4.5)$$

di mana $\bar{x}$, $\bar{y}$ masing-masing adalah nilai rata-rata untuk variabel bebas dan variabel terikat. Jadi ekspresi spesifik untuk analisis regresi linier kesatuan dapat dikerjakan. Setelah mengerjakan ekspresi spesifik untuk model, kami ingin mengetahui tingkat kesesuaian ekspresi tersebut dengan dataset, apakah ekspresi tersebut dapat mengekspresikan hubungan antara dua variabel, dan apakah ekspresi tersebut dapat digunakan dalam prediksi yang sebenarnya. Semakin dekat R^2 (koefisien determinasi) ke 1, semakin baik derajat kesesuaiannya, dan semakin jauh R^2 dari 1, semakin buruk derajat kesesuaiannya:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2} , \; 0 \geq R^2 \geq 1 \quad (4.6)$$

Perlu dicatat bahwa regresi linier dapat digunakan tidak hanya untuk prediksi tetapi juga untuk klasifikasi, tetapi klasifikasi hanya digunakan dalam kiasan untuk masalah klasifikasi biner. Saat kita menghitung persamaan regresi y = âx + b̂ , kita dapat menghitung nilai estimasi variabel dependen untuk setiap sampel dalam kumpulan data pelatihan; rumusnya adalah = âx + b̂, sehingga mengasumsikan dua nilai yang mungkin:

$$class = \begin{cases} 1 & y_i > \hat{y}_i \\ 0 & y_i < \hat{y}_i \end{cases} \quad i = 1, 2, \cdots, n \quad (4.6)$$

Data awal (x_0, y_0) digunakan untuk klasifikasi. Pertama, kita tentukan 0 dengan menggunakan variabel dependen x_0, kemudian kita bandingkan y_0, $\hat{y}_0$ untuk menentukan termasuk kelas yang mana. Untuk regresi linier multivariat, seperti yang dipelajari di bawah ini, metode ini juga diterapkan untuk mengklasifikasikan kumpulan data.

Analisis Regresi Linier Multivariat

Selama menyelesaikan masalah yang sebenarnya, kita akan sering menemukan banyak variabel. Misalnya, nilai seorang siswa dapat dipengaruhi oleh faktor-faktor seperti kesungguhannya di kelas, persiapan sebelum kelas dan ulasan setelah kelas; Kesehatan seorang pria tidak hanya dipengaruhi oleh lingkungannya, tetapi juga terkait dengan kebiasaan makannya

pada waktu-waktu biasa. Semua ini menunjukkan bahwa model regresi linier kesatuan tidak disesuaikan dengan banyak kondisi, sehingga kami melakukan perbaikan dan mengajukan model analisis regresi linier multivariat, yang strukturnya diberikan di bawah ini:

$$\begin{cases} y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m + \varepsilon \\ \varepsilon - N(0, \sigma^2) \end{cases} \tag{4.7}$$

Oleh karena itu, β_0, β_1, … , β_m, σ^2 adalah parameter yang tidak diketahui, dan memenuhi distribusi normal dimana nilai rata-rata adalah 0 dan varians sama dengan 2. Dengan mengerjakan ekspektasi untuk struktur di atas, dan kita mendapatkan persamaan regresi linier multivariat (diganti y untuk E(y)) sebagai:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m \tag{4.8}$$

Bentuk matriksnya diberikan sebagai y = Xβ, di mana X = [1, x_1, … , x_m] , β = [β_0, β_1, … , m]^T . Demikian pula, kita perlu mengestimasi parameter . Cari tahu dengan metode kuadrat terkecil. Tujuannya diberikan sebagai

$$\min \; Q = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im})^2 \tag{4.9}$$

Untuk membuat jumlah kuadrat, kita perlu membuat turunan parsial untuk β. Proses ini memberikan

$$\begin{cases} \dfrac{\partial Q}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im}) = 0 \\ \dfrac{\partial Q}{\partial \beta_j} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im}) x_{ij} = 0 \\ j = 1, 2, \cdots, m \end{cases} \xrightarrow{solve} \hat{\beta} = (X^T X)^{-1} X^T Y \tag{4.10}$$

Oleh karena itu persamaan regresi akhir yang diperoleh adalah y = Xβ = 0 + 1x1 + + mxm. Sebenarnya, regresi multivariat adalah perluasan dan perluasan dari regresi kesatuan; mereka identik di alam, tetapi jangkauan aplikasi mereka berbeda. Regresi kesatuan adalah kiasan untuk masalah dengan satu variabel independen dan satu variabel dependen, sedangkan regresi multivariat berlaku untuk masalah dengan beberapa variabel independen dan satu variabel dependen.

Contoh 4.2 Analisis Data Kesehatan dengan Regresi Linier

Dengan membaiknya perekonomian, semakin banyak orang yang peduli dengan kondisi kesehatan mereka. Sebagai contoh, obesitas dicerminkan oleh indeks berat badan. Orang gemuk

lebih mungkin untuk memiliki tekanan darah tinggi atau diabetes. Menggunakan model regresi linier, kami memprediksi hubungan antara obesitas dan tekanan darah tinggi.

Tabel 4.4 menunjukkan dataset indeks berat badan dan tekanan darah beberapa orang yang menjalani pemeriksaan kesehatan di sebuah rumah sakit di Wuhan, China. Kami melakukan penilaian awal tentang apa datum tekanan darah seseorang dengan indeks berat badan 24.

Ini adalah model prediksi dengan dua variabel, sehingga regresi linier kesatuan dapat dipertimbangkan. Pertama, tentukan distribusi titik data, dan gambar diagram pencar untuk indeks berat badan-tekanan darah dengan MATLAB, seperti yang ditunjukkan pada Gambar 4.5.

Tabel 4.4 Lembar data indeks berat badan dan tekanan darah.

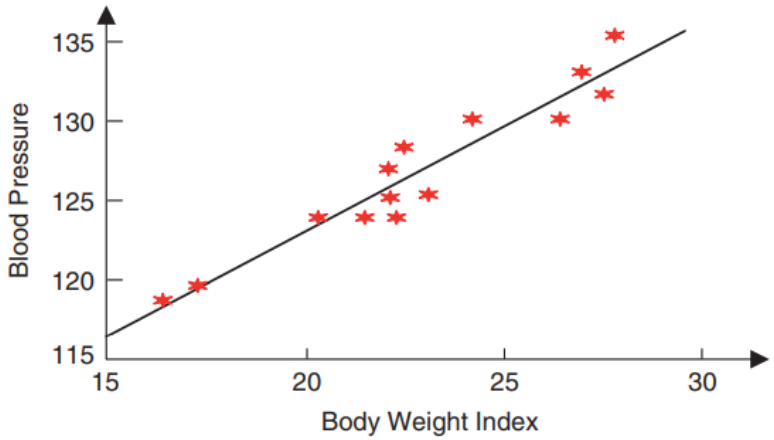
Id	Indeks Berat Badan	Tekanan Darah (mmHg)	Id	Indeks Berat Badan	Tekanan Darah (mmHg)
1	20.9	123	8	21.4	126
2	21.5	123	9	21.4	124
3	19.6	123	10	25.3	129
4	26	130	11	22.4	124
5	16.8	119	12	26.1	133
6	25.9	131	13	23	129
7	21.6	127	14	16	118

Semua titik data hampir berada pada atau di bawah garis lurus, dan terdistribusi secara linier. Oleh karena itu, ruang data dimodelkan dengan proses regresi linier kesatuan. Dengan metode kuadrat terkecil, kita mendapatkan a = 1,32 dan b = 96,58. Oleh karena itu kami memiliki y = 1,32x + 96,58. Uji signifikansi diperlukan untuk memverifikasi apakah model akan cocok dengan data saat ini. Kemudian dilakukan prediksi melalui perhitungan, sehingga mean residual dan koefisien determinasi model adalah: rata-rata error 1,17 dan R^2 = 0,90.

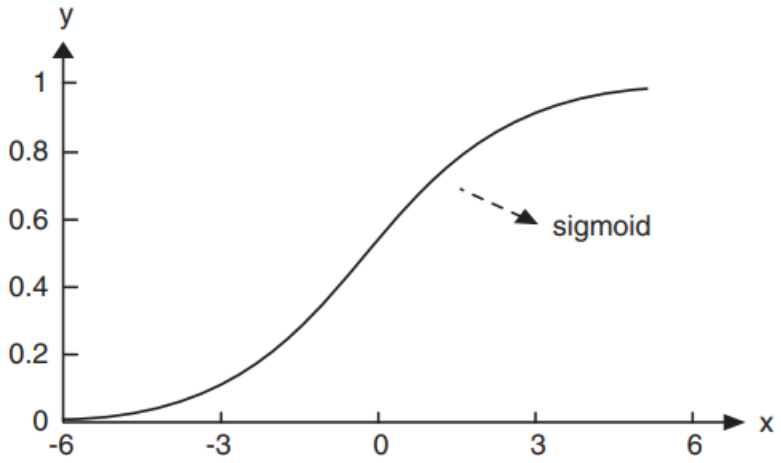
Residu rata-rata jauh lebih kecil daripada nilai rata-rata 125,6 tekanan darah, dan koefisien determinasi mendekati 1. Oleh karena itu, dapat disimpulkan bahwa persamaan regresi ini signifikan, dan dapat cocok dengan baik ke dalam kumpulan data, dan bahwa prediksi dapat dilakukan untuk data yang tidak diketahui atas dasar ini. Hasil model regresi ditunjukkan, seperti pada Gambar 4.5. Nilai tekanan darah seseorang dapat ditentukan dengan model yang diperoleh dan indeks berat badan yang diberikan. Substitusikan 24 untuk x, dan kita dapat memperoleh nilai tekanan darah orang tersebut sebagai y = 1,32 × 24 + 96,58 = 128.

Regresi Logistik untuk Klasifikasi

Regresi logistik adalah model analisis regresi linier dalam arti luas, dan dapat digunakan untuk prediksi dan klasifikasi.



Gambar 4.5 Hubungan antara berat badan dan tekanan darah.



Gambar 4.6 Kurva fungsi sigmoid yang diterapkan dalam metode regresi.

Ini biasanya digunakan di bidang-bidang seperti penambangan data, diagnosis otomatis untuk penyakit, dan prediksi ekonomis. Namun, yang perlu diperhatikan adalah Model Logistik hanya dapat digunakan untuk menyelesaikan masalah dikotomi. Adapun klasifikasi regresi logistik (singkatnya LR classifier), prinsipnya adalah melakukan klasifikasi terhadap data sampel dengan fungsi logistik; ekspresi untuk fungsi logistik (umumnya dikenal sebagai fungsi sigmoid) dinyatakan oleh:

$$f(x) = \frac{1}{1 + e^{-z}} \tag{4.11}$$

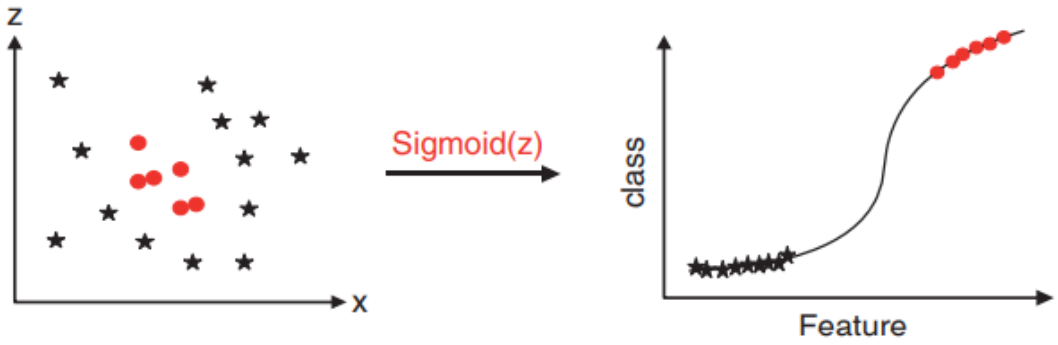
Untuk fungsi tersebut, domain definisinya adalah (−∞, +∞), dan range nilainya adalah (0, 1); oleh karena itu, kita dapat menganggap fungsi sigmoid sebagai fungsi kepadatan probabilitas untuk data sampel. Gambar fungsinya adalah seperti Gambar 4.6.

Citra sensitif jika z = 0, dan tidak sensitif jika $z \gg 0$ atau $z \ll 0$; oleh karena itu, data sampel dapat dikonsentrasikan pada kedua ujung fungsi sigmoid dengan menggunakan fitur perantara z dari sampel, sehingga dapat dibagi menjadi dua kelas. Ini adalah ide dasar untuk regresi logistik. Perhatikan vektor x dengan m variabel bebas $x = (x_1, x_2, x_3, \dots, x_m)$. Setiap dimensi x mewakili satu atribut (fitur) dari data sampel (data pelatihan). Dalam regresi logistik, beberapa fitur dari data sampel digabungkan menjadi satu fitur dengan menggunakan fungsi linier:

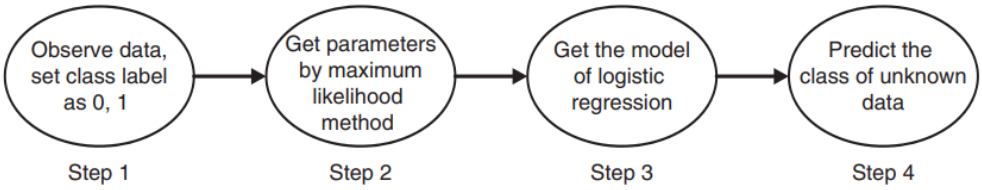
$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m \tag{4.12}$$

Dengan mencari tahu probabilitas fitur itu dalam data yang ditentukan, dan memanfaatkan fungsi sigmoid untuk bertindak pada fitur itu, kami memperoleh ekspresi untuk regresi logistik yang didefinisikan di bawah ini. Hasilnya diplot pada Gambar 4.7.

$$\begin{cases} P(Y=1|x) = \pi(x) = \dfrac{1}{1+e^{-z}} \\ z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m \end{cases} \rightarrow \begin{cases} x \in 1\ ,\ if\ P(Y=1|x) > 0.5 \\ x \in 0\ ,\ if\ P(Y=0|x) < 0.5 \end{cases} \tag{4.13}$$



Gambar 4.7 Prinsip penggunaan regresi logistik untuk tujuan klasifikasi.



Gambar 4.8 Empat langkah untuk proses regresi logistik.

Selama menggabungkan beberapa fitur menjadi satu fitur, kami menggunakan fungsi linier, tetapi kami tidak mengetahui koefisien fungsi linier (yaitu bobot fitur dari data sampel), sehingga bobot perlu ditentukan. Umumnya, estimasi kemungkinan maksimum diadopsi untuk mengubahnya menjadi masalah optimasi, dan koefisien ditentukan melalui metode optimasi. Kesimpulannya, langkah-langkah umum untuk regresi logistik adalah seperti pada Gambar 4.8.

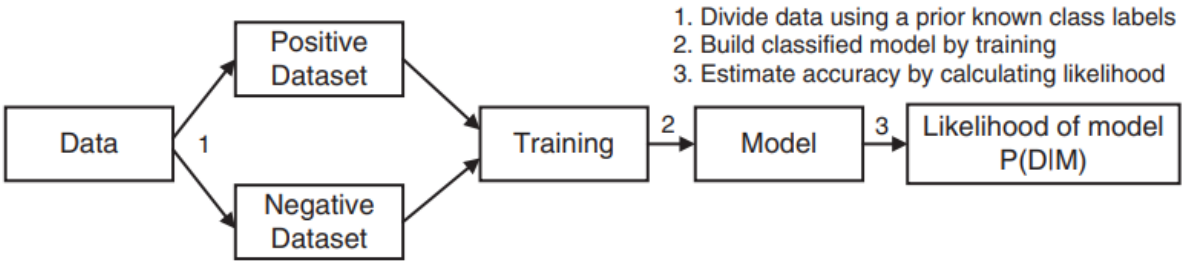
4.3 METODE KLASIFIKASI TERBIMBING

Algoritma klasifikasi sering digunakan dalam *Machine learning* yang diawasi. Data masukan adalah data pelatihan. Setiap data pelatihan diberi label tertentu. Misalnya, label "spam" atau "surat resmi" dapat diberikan untuk melabeli setiap contoh surat dalam rangkaian pelatihan sistem penyaringan spam. Pembelajaran terawasi perlu membangun model prediksi dengan tingkat akurasi yang dapat diterima. Model terus meningkatkan akurasinya dengan membandingkan hasil prediksi dengan hasil berlabel dari set pelatihan. Model secara konstan menyesuaikan mekanisme prediksinya hingga hasil prediksi mencapai tingkat akurasi tertentu.

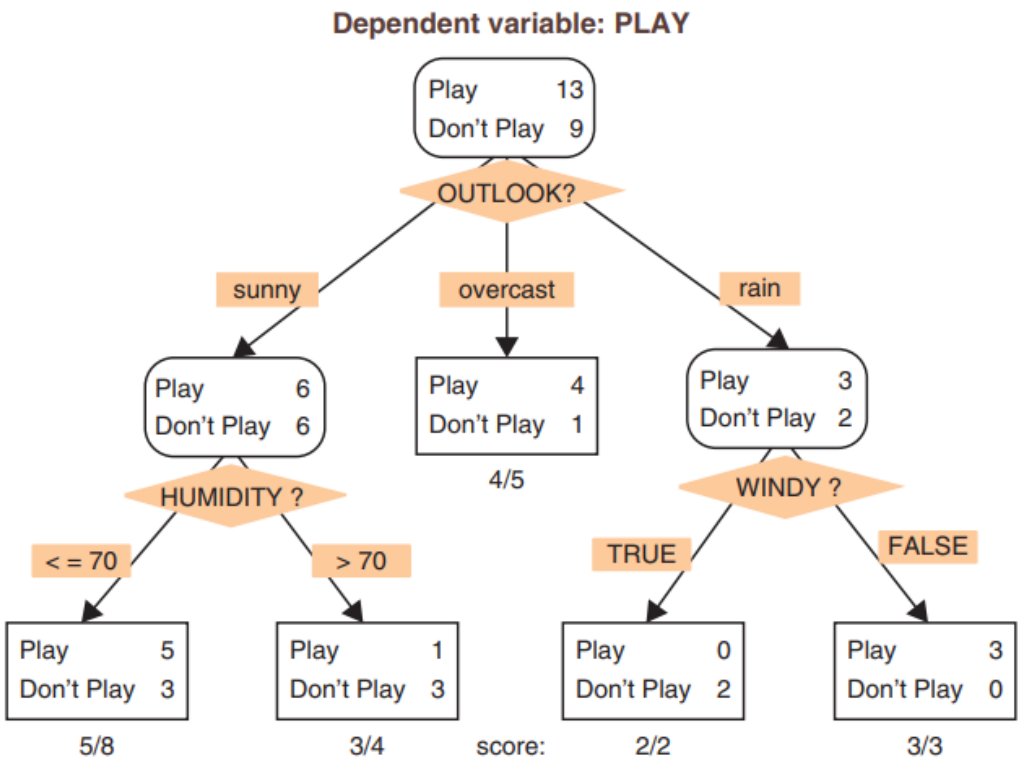
Biasanya, generasi model classifier melewati tiga langkah, seperti yang ditunjukkan pada Gambar 4.9, untuk masalah dua kelas. Langkah 1 membagi kumpulan data sampel menjadi dua himpunan bagian (positif versus negatif). Model terklasifikasi dibangun dengan pelatihan pada langkah 2. Akhirnya, akurasi model ditentukan dengan menggunakan probabilitas kemungkinan. Pada bagian ini, kita akan mempelajari empat keluarga metode klasifikasi terawasi: yaitu pohon keputusan, pengklasifikasi berbasis aturan, pengklasifikasi tetangga terdekat dan mesin support-vektor.

Pohon Keputusan untuk *Machine learning*

Pohon keputusan menawarkan model prediktif baik dalam penambangan data maupun *Machine learning*. Kami akan berkonsentrasi pada *Machine learning* menggunakan pohon keputusan. Tujuannya adalah untuk membuat model yang memprediksi nilai variabel target keluaran pada simpul daun pohon, berdasarkan beberapa variabel masukan atau atribut pada simpul akar dan simpul interior pohon itu. Pohon keputusan untuk klasifikasi dikenal sebagai pohon klasifikasi.



Gambar 4.9 Tiga langkah dalam membangun model klasifikasi melalui pelatihan data sampel.



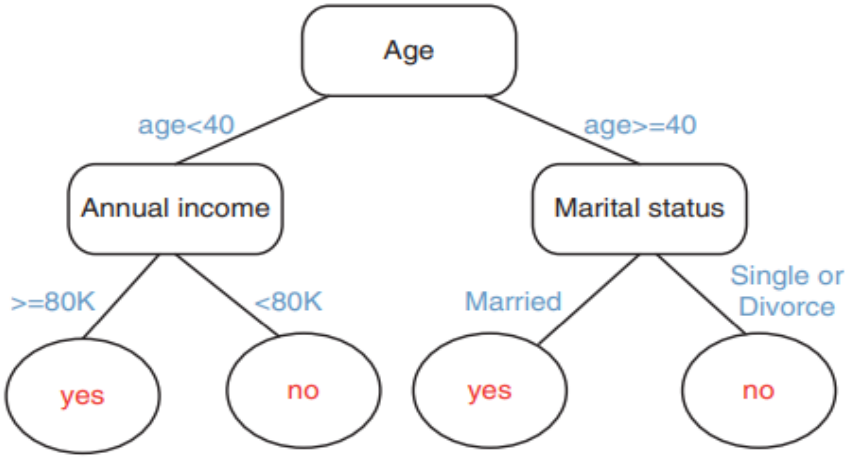
Gambar 4.10 Pohon keputusan untuk pengambilan keputusan bermain tenis atau tidak dengan probabilitas yang diberikan pada simpul daun.

Dalam pohon klasifikasi, daun mewakili label kelas dan cabang mewakili konjungsi atribut yang mengarah ke label kelas. Variabel target (output) dapat mengambil dua nilai (seperti ya atau tidak) atau beberapa nilai diskrit (seperti hasil 1, 2, 3 atau 4 dari suatu peristiwa). Busur dari sebuah simpul diberi label dengan masing-masing nilai atribut yang mungkin. Setiap daun pohon diberi label dengan kelas atau distribusi probabilitas di atas kelas. Pohon keputusan, di mana variabel target mengasumsikan nilai kontinu (seperti bilangan real), disebut pohon regresi.

Pohon keputusan mengikuti struktur pohon multi-level untuk membuat keputusan di simpul daun pohon. Konsep tersebut diilustrasikan dengan contoh pada Gambar 4.10. Di pohon ini, kita perlu memutuskan apakah akan bermain tenis dalam berbagai kondisi cuaca. Kondisi cuaca ditunjukkan oleh tiga atribut: pandangan, kelembaban dan angin. Prospek diperiksa di root, yang memiliki tiga kemungkinan busur keluar yang ditandai sebagai cerah, mendung, atau hujan. Kelembaban menyebar ke dua busur berlabel lebih dari 70 atau tidak. Nilai angin hanya benar atau salah.

Untuk melintasi pohon ini, kita mulai dari akar ke daun sepanjang jalan satu atau dua tingkat. Di dalam setiap simpul pohon, jumlah target diberikan untuk menentukan probabilitas jika simpul daun tercapai. Misalnya, jika nilai pandangan mendung, kita mencapai simpul daun dengan probabilitas 4/5 untuk bermain tenis. Di sisi lain, jika prospek cerah dan kelembaban di

atas 70, kita mencapai simpul daun paling kiri dengan probabilitas 5/8 untuk bermain tenis. Demikian pula, kita juga dapat menjangkau simpul daun lain dengan probabilitas berbeda.



Gambar 4.11 Pohon keputusan untuk menyetujui permohonan pinjaman kepada nasabah bank.

Dalam kasus keputusan prediksi sederhana, nilai target dapat berupa label kelas saja (seperti ya atau tidak) tanpa indikasi probabilitas, seperti terlihat pada Contoh 4.3.

Contoh 4.3 Persetujuan Pinjaman Bank menggunakan Pohon Keputusan dengan Data Pelatihan

Pertimbangkan untuk menggunakan pohon keputusan untuk membuat keputusan apakah bank akan menyetujui permohonan pinjaman dari nasabahnya. Dataset diklasifikasikan dengan tiga atribut: usia, pendapatan tahunan dan status perkawinan. Node internal di setiap level menguji satu atribut. Setiap simpul daun mewakili satu keputusan kelas: "ya" berarti persetujuan dan "tidak" untuk sebaliknya. Gambar 4.11 menunjukkan pohon keputusan yang sudah dibangun oleh bank.

Selama membangun pohon keputusan, usia pelamar dianggap pertama di simpul akar. Partisi usia sampel pelatihan menjadi dua kategori: mereka yang kurang dari 40 versus lainnya. Atribut pendapatan tahunan kemudian diuji pada tingkat kedua. Terakhir, status perkawinan digunakan untuk mengambil keputusan menyetujui pinjaman atau tidak.

Sekarang, gunakan pohon keputusan untuk menguji penerimaan pelamar yang lebih muda dari 40 dan dengan pendapatan tahunan lebih rendah dari Rp 1.200.000.000. Dengan melintasi pohon, kami memperoleh keputusan untuk menolak aplikasi pinjaman. Jelas, bank bertindak mendukung pelamar muda dengan pendapatan lebih tinggi dan untuk pelamar yang lebih tua yang tidak lajang atau bercerai.

Ada metode, ID3, C4.5 dan CART, untuk memilih pendekatan top-down untuk membangun pohon keputusan dari set sampel pelatihan. Secara umum, set sampel pelatihan dipartisi secara rekursif menjadi subset yang lebih kecil. Kami hanya memperkenalkan algoritma ID3 (Iterative Dichotomiser 3) di bawah ini. C4.5 adalah penerus yang ditingkatkan dari ID3 dan metode CART menggabungkan klasifikasi dan regresi dalam konstruksi pohon.

Penandaan Algoritma ID3

Ide inti dari algoritma ID3 mengambil keuntungan informasi dari atribut sebagai ukuran, dan membagi atribut dengan keuntungan informasi terbesar setelah pemisahan, untuk membuat partisi output pada setiap cabang milik kelas yang sama sejauh mungkin. Standar ukuran perolehan informasi adalah entropi, yang menggambarkan kemurnian set contoh apa pun. Diberikan satu set pelatihan S dari contoh positif dan negatif, fungsi entropi S didefinisikan sebagai:

$$Entropy(S) = -p_+ \log_2^{p_+} - p_- \log_2^{p_-} \quad (4.14)$$

Tabel 4.5 Dataset Pelatihan yang digunakan pada Contoh 4.4.

RID	Pendapatan tahunan (Rp)	Usia	Status pernikahan	Kelas: memuat
1	70 K	18	Lajang	Tidak
2	230 K	35	Perceraian	Ya
3	120 K	28	Telah menikah	Ya
4	200 K	30	Telah menikah	Ya

di mana p+ mewakili contoh positif dan p- mewakili contoh negatif. Jika atribut target memiliki m nilai yang berbeda, maka entropi S relatif terhadap klasifikasi m kelas ditentukan oleh:

$$Entropy(S) = \sum_{i=1}^{m} -p_i \log_2^{p_i} \quad (4.15)$$

Standar ukuran keefektifan data pelatihan didefinisikan sebagai entropi, yang merupakan standar untuk mengukur kemurnian rangkaian contoh pelatihan, dan standar pengukuran di atas disebut “perolehan informasi”. Perolehan informasi dari suatu atribut menunjukkan penurunan entropi yang diharapkan yang disebabkan oleh contoh tersegmentasi. Kami mendefinisikan gain Gain(S, A) dari atribut A di set S sebagai:

$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (4.16)$$

di mana V (A) adalah jangkauan A, S adalah himpunan sampel dan S_v adalah himpunan sampel dengan nilai A sama dengan v.

Contoh 4.4 Prediksi Pohon Keputusan menggunakan Algoritma ID3

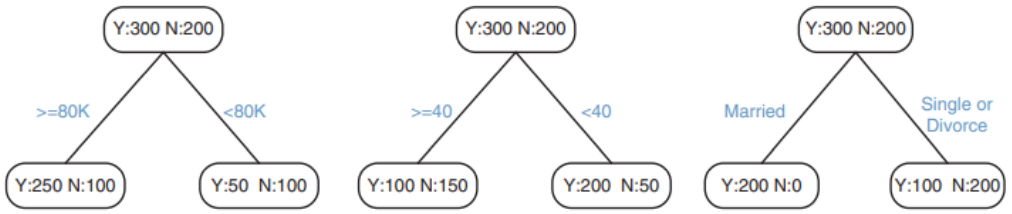
Diberikan training set D dengan 500 sampel, dimana format data ditunjukkan pada Tabel 4.5, dan atribut label kelas “load” memiliki dua nilai yang berbeda (yaitu {yes, no}), oleh karena itu terdapat dua kategori yang berbeda (yaitu m = 2) . Misalkan kategori C1 sesuai dengan "ya",

dan kategori C2 sesuai dengan "tidak". Ada 300 tupel dalam kategori "ya", dan 200 tupel dalam kategori "tidak". Dan (root) node N dibuat untuk tupel di D. Perolehan informasi dari setiap atribut harus dihitung untuk menemukan kriteria split dari tupel tersebut.
Nilai entropi digunakan untuk mengklasifikasikan tupel di D sebagai:

$$Entropy(D) = -\frac{2}{5}\log_2^{\frac{2}{5}} - \frac{3}{5}\log_2^{\frac{3}{5}} = 0.971 \quad (4.17)$$

Kemudian kami menghitung permintaan informasi yang diharapkan dari setiap atribut. Untuk atribut pendapatan sama atau lebih besar dari 80 K, terdapat 250 tupel “ya” dan 100 tupel “tidak”. Untuk atribut pendapatan kurang dari 80 K, terdapat 50 tupel “ya” dan 100 tupel “tidak”. Ketika keuntungan informasi digunakan, jika tupel dipartisi dengan pendapatan tahunan, entropi yang diharapkan untuk mengklasifikasikan tupel di D adalah:

$$Entropy_{income}(D) = \frac{7}{10} \times \left(-\frac{5}{7}\log_2^{\frac{5}{7}} - \frac{2}{7}\log_2^{\frac{2}{7}}\right) + \frac{3}{10} \times \left(-\frac{1}{3}\log_2^{\frac{1}{3}} - \frac{2}{3}\log_2^{\frac{2}{3}}\right) = 0.8797 \quad (4.18)$$



Gambar 4.12 Partisi pohon keputusan untuk tiga atribut pada Contoh 4.4.

Oleh karena itu, perolehan informasi dari partisi tersebut dinyatakan dengan

$$Gain(D, income) = Entropy(D) - Entropy_{income}(D) = 0.9710 - 0.8797 = 0.0913 \quad (4.19)$$

Representasi grafis ditunjukkan pada Gambar 4.12(a), situasi usia dan status perkawinan seperti yang ditunjukkan pada Gambar 4.12(b) dan (c); demikian pula, informasi usia dan status perkawinan dapat dihitung. Atribut dengan perolehan informasi terbesar dipilih untuk membangun pohon. Dari perhitungan di atas didapatkan information gain yang paling besar bila menggunakan atribut umur, oleh karena itu atribut ini dipilih menjadi kriteria klasifikasi.

Klasifikasi Berbasis Aturan

Teknik klasifikasi telah diterapkan di banyak bidang, seperti teknik klasifikasi teks otomatis dalam pencarian informasi dan mesin pencari, dan deteksi intrusi di bidang keamanan. Para peneliti di bidang *Machine learning*, sistem pakar, statistik, dan jaringan saraf telah

mengajukan sejumlah metode peramalan klasifikasi khusus. Bagian ini akan fokus pada teknik klasifikasi berbasis aturan.

Pengklasifikasi berbasis aturan adalah teknik untuk menggunakan seperangkat aturan "jika kemudian ..." untuk mengklasifikasikan catatan, biasanya mewakili aturan model dalam bentuk normal disjungtif seperti yang diberikan oleh R = ($r_1 \vee r_2 \vee r_k$), di mana R berarti himpunan aturan, sedangkan ri adalah aturan klasifikasi atau disjungsi.

Pertimbangkan penggunaan tiga aturan prediksi:

1) r_1: (Suhu tubuh = Darah dingin) → Non-mamalia
2) r_2: (Suhu tubuh = Suhu konstan) (Viviparitas = Ya) → Mamalia
3) r_3: (Suhu tubuh = Konstan) (Viviparitas = Tidak) → Non-mamalia

Setiap aturan klasifikasi diwakili oleh r_i : (Conditioni) → y_i. Sisi kiri aturan disebut premis atau anteseden aturan, dan sisi kanan aturan disebut kesimpulan atau akibat aturan. Jika catatan memenuhi aturan, kami mengatakan bahwa itu diaktifkan atau dipicu; atau catatan dicakup oleh aturan. Secara umum, rule antecedent direpresentasikan dengan $\text{Condition}_i = (A_1 \text{ op } v_1) \wedge (A_2 \text{ op } v_2) \wedge \cdots (A_k \text{ op } v_k)$, dimana masing-masing ($A_i$ op v_i) disebut conjunct dan terdiri dari pasangan atribut-nilai dan a operator logika (op), dan umumnya op $\{=, \neq, <, >, \leq, \geq\}$.

Untuk setiap kelas, mungkin ada lebih dari satu aturan yang dapat diterapkan. Jadi, aturan mana yang lebih unggul? Untuk menentukan kualitas aturan klasifikasi, kami mendefinisikan fungsi presisi cakupan. Untuk dataset D dan aturan klasifikasi: r : A → y, cakupan aturan didefinisikan sebagai proporsi record yang memicu aturan r dalam D. Ketepatan aturan atau faktor kepercayaan didefinisikan sebagai proporsi record yang label kelasnya sama dengan y dalam record memicu r. Rumus matematika diberikan sebagai

$$Coverage(r) = \frac{|A|}{|D|}$$
$$Accuracy(r) = \frac{|A \cap y|}{|A|} \qquad (4.20)$$

dimana |A| adalah jumlah record yang memenuhi aturan anteseden, |A ∩ y| adalah jumlah record yang memenuhi aturan anteseden dan aturan konsekuen dan |D| adalah jumlah total record.

Terkadang, kita tidak yakin apakah beberapa aturan dalam kumpulan aturan tertentu tidak efektif. Karena beberapa catatan dapat dipicu oleh lebih dari satu aturan, ini akan menyebabkan duplikasi aturan, sementara yang lain mungkin tidak tercakup oleh aturan apa pun. Oleh karena itu, kami mempertimbangkan dua properti penting untuk meningkatkan penerapan aturan:

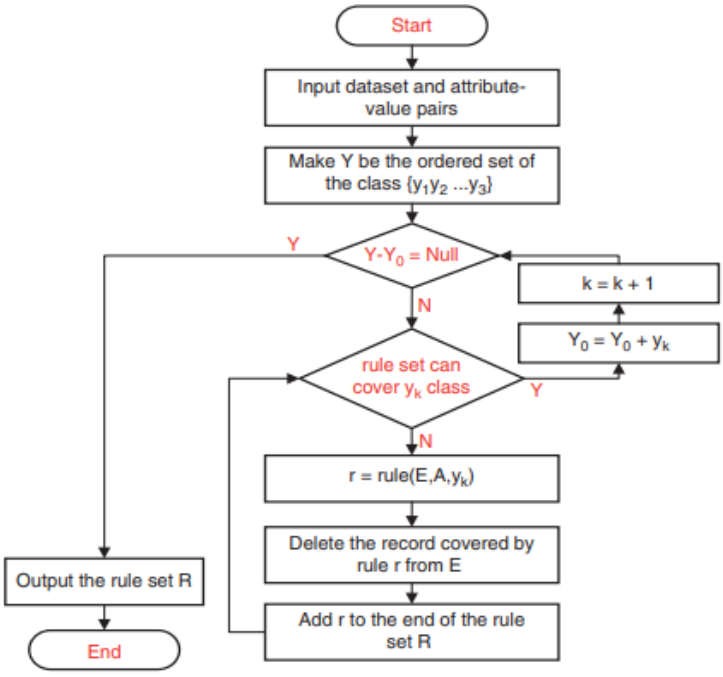
1) **Aturan pengecualian bersama:** Jika tidak ada aturan yang dipicu oleh record yang sama dalam kumpulan aturan R, dikatakan bahwa aturan dalam kumpulan aturan R saling

eksklusif. Properti ini memastikan bahwa catatan harian dicakup oleh satu aturan paling banyak di R. Kumpulan aturan di atas adalah aturan yang saling eksklusif.

2) **Aturan Exhaustive:** jika untuk setiap kombinasi nilai properti, ada aturan di R untuk menutupinya, dikatakan bahwa himpunan aturan R adalah dengan cakupan lengkap. Properti ini memastikan bahwa catatan harian dicakup oleh satu aturan setidaknya di R.

Aturan yang ditetapkan dengan properti yang saling eksklusif dan lengkap memastikan catatan dapat dicakup oleh satu dan hanya satu aturan. Namun, banyak kumpulan aturan tidak dapat memenuhi kedua properti ini. Jika kumpulan aturan tidak dapat memenuhi properti lengkap, aturan default rd : () → y_d harus ditambahkan untuk menutupi catatan yang tidak ditemukan. Jika anteseden dari aturan default kosong, pemicuan akan terjadi jika semua aturan gagal dan y_d adalah kelas default. Seringkali nilai sebagian besar kelas catatan tidak tercakup oleh aturan. Jika seperangkat aturan tidak memenuhi properti yang saling eksklusif, catatan dapat dicakup oleh lebih dari satu aturan, dan klasifikasi aturan ini mungkin bertentangan; lalu bagaimana cara menentukan hasil klasifikasi record tersebut? Dua solusi berikut diberikan:

1) **Aturan berurutan:** kumpulan aturan semacam ini diurutkan dari besar ke kecil sesuai dengan prioritas aturan, yang didefinisikan secara umum dengan presisi, cakupan, dan sebagainya. Saat mengklasifikasikan, aturan dipindai secara berurutan hingga ditemukan aturan yang menutupi catatan, dan aturan ini akan menjadi hasil klasifikasi catatan ini. Pengklasifikasi berbasis aturan umum mengadopsi metode ini.
2) **Aturan tidak berurutan:** dalam hal ini, semua aturan sama satu sama lain. Aturan dipindai secara berurutan, dan setelah rekor terjadi, masing-masing akan dipilih, dan yang mendapatkan suara terbanyak akan menjadi hasil klasifikasi akhir dari catatan.



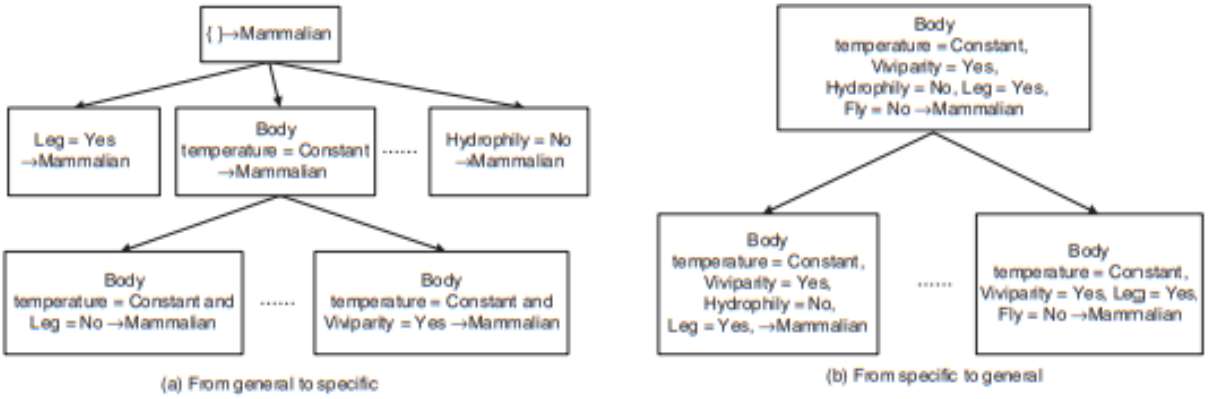
Gambar 4.13 Cakupan berurutan dan aliran data untuk ekstraksi aturan.

Ekstraksi Aturan dengan Aturan Langsung

Algoritma cakupan sekuensial sering digunakan untuk secara langsung mengekstrak aturan dari data, dan pertumbuhan aturan biasanya dengan cara yang rakus berdasarkan beberapa jenis ukuran evaluasi. Algoritme mengekstrak kelas aturan pada satu waktu dari catatan yang berisi lebih dari satu data pelatihan. Diagram alir untuk mengilustrasikan aliran data disajikan pada Gambar 4.13. Di sini, E mewakili kumpulan data pelatihan, A adalah pasangan nilai atribut $\{(A_j, v_j)\}$, dan R adalah kumpulan aturan. Pertama kita memasukkan kumpulan A dari dataset pelatihan E dan pasangan nilai atribut, kemudian membuat Y himpunan terurut dari kelas $\{y_1, y_2, \cdots, y_k\}$, dan R = {} adalah himpunan aturan awal. Untuk setiap kelas y di Y, sedangkan set aturan dapat mencakup kelas y data pelatihan, menggunakan fungsi Rule() menghasilkan aturan r, menghapus catatan yang dicakup oleh aturan r dari E dan menambahkan r ke akhir set aturan, yaitu R = R ∨ r . Jika tidak, akhiri sirkulasi. Terakhir, tambahkan aturan default () → y_d ke akhir set aturan.

Fungsi aturan adalah untuk mengekstrak aturan klasifikasi, yang mencakup lebih banyak contoh positif dengan pelatihan terkonsentrasi, dan tidak mencakup tidak ada atau hanya beberapa contoh penghitung. Untuk menghindari ledakan eksponensial, fungsi meningkatkan aturan dengan cara yang rakus. Ini pertama-tama membuat aturan r, dan kemudian terus-menerus melakukan perbaikan pada aturan sampai memenuhi kondisi tertentu; dan kemudian memangkas aturan untuk memperbaiki kesalahan generalisasinya. Gambar 4.14 menunjukkan kasus strategi pembuatan aturan dari umum ke khusus dan dari sifat sampel khusus ke umum.

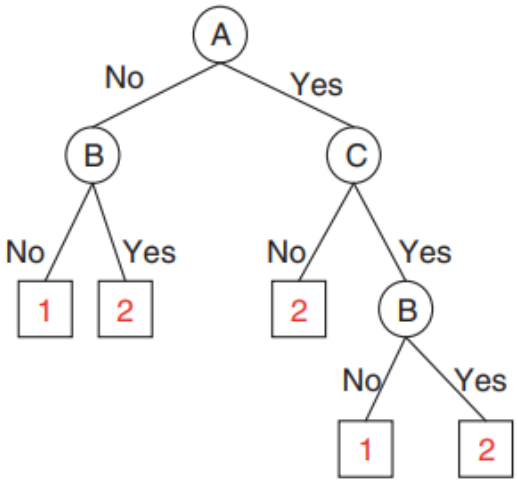
Secara umum, siapkan aturan awal r : {} → y, di mana anteseden aturan adalah himpunan kosong, dan aturan konsekuen berisi kelas target. Aturan ini mencakup semua rekaman set pelatihan, sehingga kualitasnya sangat buruk. Kita dapat menambahkan konjungsi baru untuk meningkatkan kualitas aturan, yang akan dilanjutkan hingga kondisi akhir yang memuaskan, seperti konjungsi yang ditambahkan tidak dapat meningkatkan kualitas aturan. Kami ingin meningkatkan aturan dalam strategi dari umum ke khusus, yang dapat dikonversi. Artinya, kita dapat secara acak memilih contoh positif sebagai benih awal untuk aturan yang meningkat, dan kemudian menghapus konjungsi aturan untuk mencakup lebih banyak contoh positif untuk menggeneralisasi aturan hingga memenuhi kondisi akhir, seperti aturan yang memunculkan contoh penghitung.



Gambar 4.14 Strategi pembangkitan aturan antara properti umum dan khusus.

Ekstraksi Aturan dari Pohon Keputusan

Ekstraksi aturan dari pemodelan pohon keputusan adalah metode tidak langsung yang umum untuk ekstraksi aturan. Pada prinsipnya, setiap jalur pohon keputusan dari simpul akar ke simpul daunnya dapat mengekspresikan aturan klasifikasi. Kondisi untuk setiap jalur merupakan anteseden aturan, sedangkan label kelas dari simpul daun merupakan aturan konsekuen. Dari pemodelan pohon keputusan pada Gambar 4.15, kumpulan aturan berikut dihasilkan:



Rule set:

r_1: $(A = No, B = No) \rightarrow 1$

r_2: $(A = No, B = Yes) \rightarrow 2$

r_3: $(A = Yes, C = No) \rightarrow 2$

r_4: $(A = Yes, C = Yes, B = No) \rightarrow 1$

r_5: $(A = Yes, C = Yes, B = Yes) \rightarrow 2$

Gambar 4.15 Kumpulan aturan yang dihasilkan dari penggunaan pohon keputusan.

Dengan mempertimbangkan r_2, r_3, r_5 dan dua aturan berikut:

$$r_6 : (B = Yes) \rightarrow 2$$
$$r_7 : (A = Yes) \wedge (C = No) \rightarrow 2$$

kami menemukan bahwa r_2, r_3, r_5 dapat digantikan oleh r_6, r_7. Dengan cara ini, akan lebih mudah untuk menggambarkan pemodelan pohon keputusan dengan aturan yang terdiri dari r_1, r_4, r_6, r_7.

Ini adalah konten yang dijelaskan oleh algoritma aturan C4.5: pertama, gunakan pohon keputusan untuk menghasilkan kumpulan aturan, lalu sederhanakan kumpulan aturan dan terakhir urutkan aturan.

Contoh 4.5 Prediksi Diabetes Menggunakan Rule Based Classification

Tabel 4.6 menunjukkan kumpulan data glukosa darah (tinggi, rendah), berat badan (kelebihan berat badan, normal), kadar lipid dan diabetes (ya, tidak) dari pemeriksaan fisik beberapa orang di Wuhan, yang menjadi dasar kumpulan aturan terkait , dan akan lebih mudah untuk mengklasifikasikan orang ke dalam dua kategori, yaitu diabetes dan normal.

Tabel 4.6 Dataset pemeriksaan fisik diabetes.

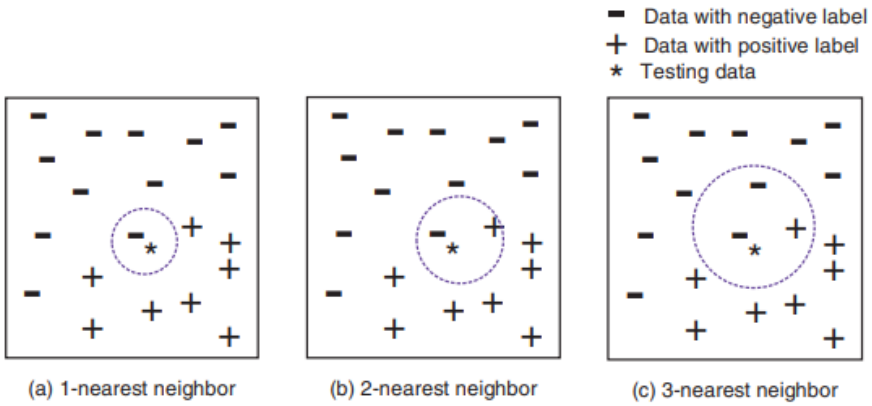
ID	Gula darah	Bobot	Kandungan lipid darah (mmol/L)	Diabetes (Ya atau Tidak)
1	Rendah	Kegemukan	2.54	No
2	Tinggi	Normal	1.31	No
3	Tinggi	Kegemukan	1.13	No
4	Rendah	Normal	2.07	No
5	Tinggi	Kegemukan	2.34	Yes
6	Tinggi	Normal	0.55	No
7	Rendah	Kegemukan	2.48	No
8	Tinggi	Kegemukan	3.12	Yes
9	Tinggi	Normal	1.14	No
10	Tinggi	Kegemukan	8.29	Yes

Pertama kita perlu menentukan set aturan dan mengklasifikasikan orang ke dalam dua kategori, yaitu pelarut dan pelarut, yang aturan akibatnya adalah diabetes (dinyatakan dengan Ya) dan normal (dinyatakan dengan Tidak). Gunakan algoritma cakupan sekuensial untuk menghasilkan aturan.

1) Tentukan kelas {Ya, Tidak}; dan kelas orang normal menjadi (Tidak ada kelas);
2) Gunakan strategi dari umum ke khusus untuk menghasilkan aturan {} → Tidak;
3) Tambahkan sifat glukosa darah (A), dan buat aturan berikut: r_1 : {A = L} → Tidak;
4) Hapus record dengan id 1, 4 dan 7, dan tambahkan aturan di atas ke himpunan aturan R, lalu R = {r_1};
5) Lanjutkan menjumlahkan sifat bobot (B), dan hasilkan r_2 : {A = H, B = Normal} → Tidak;
6) Hapus record dengan id 2, 6 dan 9, dan tambahkan aturan ke himpunan aturan R, maka R = {r_1, r_2};

7) Pertimbangkan lipid darah (C) dan kita mendapatkan aturan: r_3 : {A = H, B = Overweight, C < 1.8} → Tidak;
8) Hapus record dengan id 3, dan tambahkan aturan ke himpunan aturan R, lalu R = {r_1, r_2, r_3};
9) Periksa kelas diabetes (kelas Ya);
10) Analisis, dan buat aturan berikut: r_4 : {A = H, B = Overweight, C > 1.8} → Ya;
11) Hapus record dengan id 5, 8 dan 10, dan tambahkan aturan ke set aturan R, lalu R = {r_1, r_2, r_3, r_4};
12) Sekarang semua set data pelatihan telah dihapus, jadi hentikan sirkulasi;
13) Terakhir, keluarkan set aturan R sebagai berikut:
14) Dari uraian di atas, kami mendapatkan kumpulan aturan berikut:

$$r_1 : \{A = L\} \rightarrow No$$
$$r_2 : \{A = H, B = Normal\} \rightarrow No$$
$$r_3 : \{A = H, B = Overweight, C < 1.8\} \rightarrow No$$
$$r_4 : \{A = H, B = Overweight, C > 1.8\} \rightarrow Yes$$



Gambar 4.16 Contoh dari tiga jenis tetangga terdekat.

Pengklasifikasi Tetangga Terdekat

Pembelajaran dimulai dengan pengetahuan bahwa pohon keputusan dan klasifikasi berbasis aturan memiliki dataset pelatihan dengan model pemetaan yang dibangun dari properti input ke label kelas. Kami menyebutnya metode pembelajaran aktif. Namun, kami menyebutnya pembelajaran, ketika pemodelan dataset pelatihan ditunda hingga dataset uji dapat digunakan, metode pembelajaran pasif. Pengklasifikasi Rote, semacam metode pembelajaran pasif, tidak akan mengklasifikasikan data pengujian sampai data tersebut benar-benar cocok dengan instance set data pelatihan tertentu. Tetapi metode ini memiliki kelemahan yang jelas bahwa sebagian besar contoh data uji tidak dapat diklasifikasikan, karena tidak ada set data pelatihan yang cocok dengannya. Jadi model yang ditingkatkan yang disebut pengklasifikasi tetangga terdekat muncul dan kami akan memberikan detailnya di bawah ini.

Pengklasifikasi tetangga terdekat digunakan untuk menemukan semua contoh set data pelatihan yang memiliki properti paling mirip dengan sampel uji. Kumpulan instance dataset pelatihan ini disebut tetangga terdekat dari sampel uji dan label kelas ditentukan menurut instance ini. Jadi, pengklasifikasi tetangga terdekat menganggap setiap sampel sebagai dimensi titik n-dimensi (jumlah total properti), dan menentukan tetangga terdekat antara dua titik yang diberikan. Secara umum, kita menggunakan jarak Euclidean :

$$d(x, y) = \sum_{k=1}^{n} |x_k - y_k|$$

Sebuah contoh dari tiga jenis tetangga terdekat adalah seperti yang diberikan pada Gambar 4.16.

Sangat penting untuk memilih ambang jarak yang tepat k. Jika k terlalu kecil, pengklasifikasi tetangga terdekat cenderung terpengaruh oleh overfitting karena noise dari data pelatihan; jika k terlalu besar, pengklasifikasi tetangga terdekat mungkin salah mengklasifikasikan sampel uji karena berisi data yang jauh dari tetangga terdekat. Kita dapat menentukan label kelas dari sampel uji sesuai dengan label kelas di tetangga terdekat setelah tetangga terdekat dari sampel uji ditentukan.

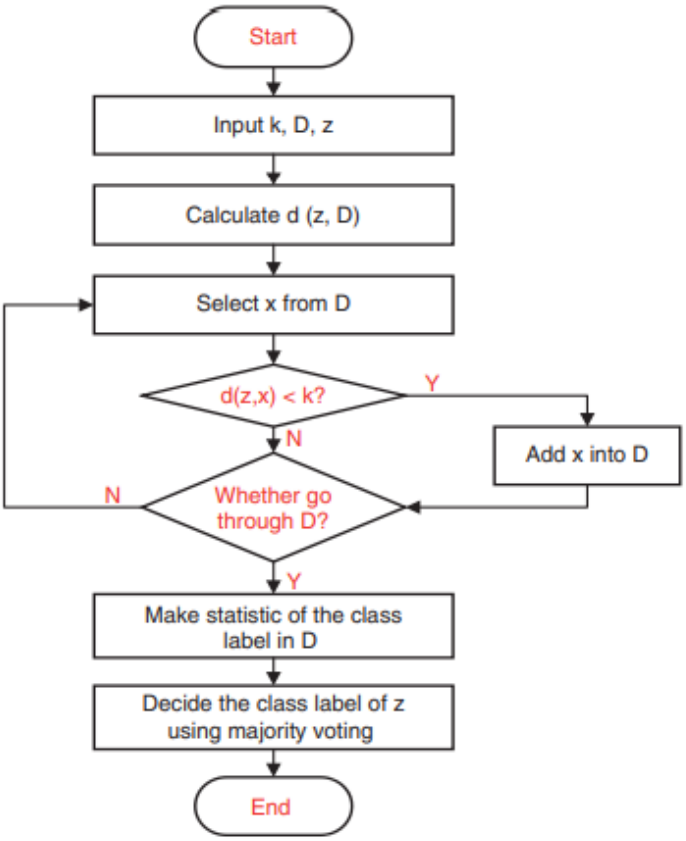
Dalam hal label kelas dari sampel uji tidak sesuai dengan rekanan di tetangga terdekat, label kelas di tetangga terdekat harus diambil sebagai label kelas dari sampel uji. Dalam hal beberapa sampel tetangga terdekat sangat penting (misalnya tetangga terdekat dengan jarak terkecil), pemilihan label kelas dapat dilakukan dengan metode pemberian koefisien bobot. Dua metode pemilihan label kelas sampel uji disebut voting mayoritas dan voting jarak tertimbang masing-masing, yang rumus matematikanya diberikan oleh:

$$\begin{cases} y = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} I(v = y_i) \\ y = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} w_i \times I(v = y_i) \end{cases} \quad (4.21)$$

V berarti label kelas, D_Z berarti tetangga terdekat dari sampel uji, Vi adalah label kelas dari tetangga terdekat dan I(·) adalah fungsi indikator yang didefinisikan sebagai:

$$I(y_i) = \begin{cases} 1 & y_i = v \\ 0 & y_i \neq v \end{cases} \quad (4.22)$$

Diagram alir diberikan pada Gambar 4.17. Variabel k mewakili ambang jarak, D adalah dataset pelatihan dan z adalah instance uji. Pertama, kita memasukkan k, D, z, kemudian menghitung jarak antara instance uji dan sampel dataset pelatihan. Sampel yang d (z, D) lebih rendah dari k dikumpulkan ke dalam himpunan D z. Kemudian, dengan menggunakan statistik label kelas di D z, kami akhirnya memutuskan label kelas dari contoh uji dengan metode pemungutan suara mayoritas.



Gambar 4.17 Diagram alir untuk algoritma klasifikasi tetangga terdekat.

Tabel 4.7 Dataset pemeriksaan fisik untuk diabetes.

Id	Trigliserida (mmol/L)	Kolesterol total (mmol/L)	Hiperlipemia atau tidak
1	1.33	4.19	No
2	1.31	4.32	No
3	1.95	5.02	Yes
4	1.86	5.17	Yes
5	1.30	537	Yes
6	1.30	4.36	No
7	2.04	4.42	Yes
8	1.45	4.68	No
9	1.35	4.41	Yes

Contoh 4.6 Prediksi Hiperlipemia Menggunakan Algoritma Nearest Neighbor

Penyakit hiperlipemia dikaitkan dengan dua indeks medis: yaitu trigliserida dan kolesterol total. Umumnya, orang yang memiliki indeks serupa mungkin memiliki masalah kesehatan yang

serupa. Kami menggunakan pengklasifikasi terdekat untuk membuat penilaian apakah pasien telah memperoleh hiperlipemia atau tidak. Tabel 4.7 menunjukkan dataset kadar trigliserida, kadar kolesterol total dan apakah akan mengalami hiperlipemia (Ya, Tidak) dari sembilan pasien potensial. Kami menganggap orang-orang tersebut mengalami hiperlipemia, jika mereka memiliki kadar trigliserida di atas 1,33 dan jumlah kolesterol total di atas 4,32.

Berdasarkan masalah tersebut, kita dapat mengklasifikasikannya dengan classifier tetangga terdekat dengan sampel uji (1.33, 4.32). Kami menetapkan ambang sebagai 0,2 untuk kasus ini dan kemudian sampel dataset pelatihan dalam kasus id = 1 dihitung sebagai:

$$\sqrt{(1.33-1.33)^2+(4.32-4.19)^2} = \ 0.13 < 0.2\ .$$

Sampel dataset pelatihan harus ditambahkan ke Dz. Sampel dataset pelatihan dalam kasus id:

$$\sqrt{(1.33-1.31)^2+(4.32-4.32)^2} = 0.02 < 0.2.$$

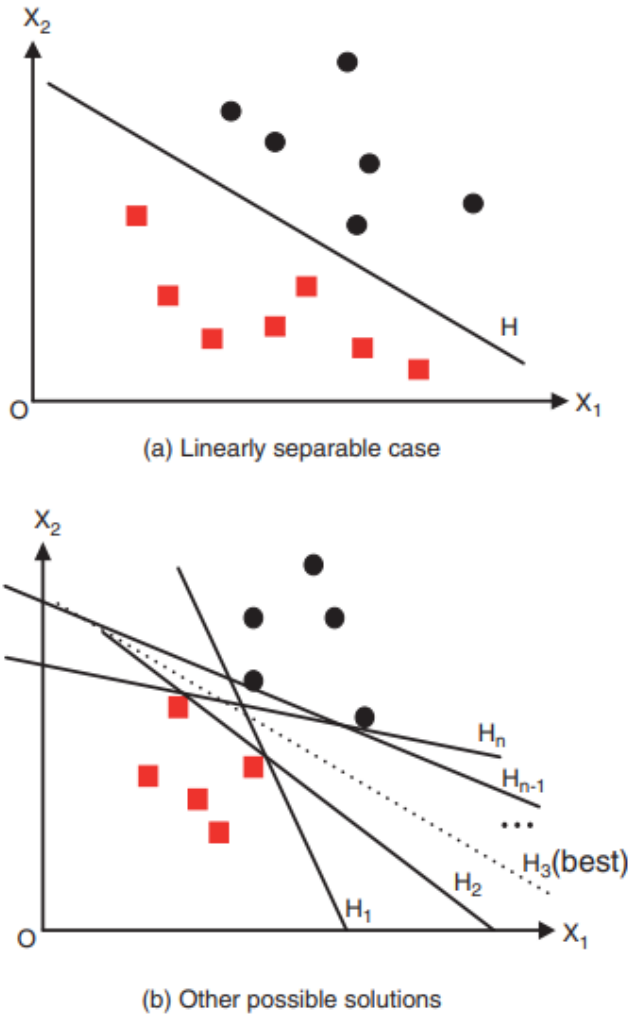
Dengan demikian, sampel dataset pelatihan harus ditambahkan ke Dz. Sampel dataset pelatihan untuk id = 3 dihitung sebagai:

$$\sqrt{(1.33-1.95)^2+(4.32-5.02)^2} = 0.94 > 0.2.$$

Kemudian, sampel dataset pelatihan harus dibuang. Dengan sisa yang ditangani dengan cara yang sama, kita memperoleh himpunan tetangga terdekat Dz sebagai Dz = {x | id = 1, 2, 6, 9}. Ada Ya dan Tidak dalam statistik koleksi tetangga terdekat. Akhirnya, kami mengumpulkan statistik dari label kelas di atas dengan metode pemungutan suara mayoritas, mengklasifikasikannya sebagai Tidak dalam kasus id = 1, 2 dan 6 dan sebagai Ya dalam kasus id = 9. Hasil pemungutan suara adalah: Ya = 1 , Tidak = 3 yang berarti orang yang diperiksa secara fisik tidak menderita hiperlipemia bila kadar trigliseridanya 1,33 dan kolesterol totalnya 4,32.

Mendukung Mesin Vektor

Vektor dukungan menawarkan pendekatan lain untuk mengklasifikasikan kumpulan data multi-dimensi. Sampel pada margin disebut vektor pendukung. Kita dapat menggunakan garis lurus untuk memisahkan titik-titik dalam ruang 2-D, dan menggunakan bidang untuk memisahkan titik-titik dalam ruang 3-D. Demikian pula, kami menggunakan hyperplane untuk memisahkan titik-titik dalam ruang dimensi tinggi. Kami menganggap titik-titik di area yang sama sebagai satu kelas, sehingga kami dapat menggunakan SVM untuk menyelesaikan masalah klasifikasi. Sedangkan masalah asli dapat dinyatakan dalam ruang berdimensi hingga, sering terjadi bahwa himpunan untuk diskriminasi tidak dapat dipisahkan secara linier dalam ruang itu. Untuk alasan ini, diusulkan agar ruang berdimensi-hingga asli dipetakan ke dalam ruang berdimensi-terbatas yang jauh lebih tinggi, mungkin membuat pemisahan lebih mudah dalam ruang itu. Dengan demikian kita dapat menggunakan hyperplane untuk mengelompokkan titik-titik ini dalam ruang dimensi tinggi.



Gambar 4.18 Konsep penggunaan SVM untuk mengklasifikasikan antara dua kelas data sampel.

Batas Keputusan Linier

Pertimbangkan bidang 2-D dengan dua jenis data, diwakili oleh titik merah dan titik biru, seperti Gambar 4.18(a). Data ini dapat dipisahkan secara linier, oleh karena itu satu garis lurus ditarik di antara mereka. Namun, garis lurus tak terbatas dapat ditarik, seperti yang ditunjukkan pada Gambar 4.18(b). Bagaimana cara mengetahui garis "terbaik", yaitu yang memiliki kesalahan klasifikasi minimum? Misalnya, pertimbangkan masalah dua kelas dalam ruang n-dimensi.

Kedua kelas dipisahkan oleh hyperplane berdimensi (n – 1), Anggap titik data D adalah (X_1, y_1) ($X_{|D|}$, $y_{|D|}$), dimana, Xi adalah sampel pelatihan dari n-dimensi, dengan label kelas yi . Setiap yi dapat mengasumsikan nilai +1 untuk satu kelas dan/atau -1 untuk kelas lainnya. Dalam hal ini, hyperplane berdimensi (n – 1) diwakili oleh:

$$w^T x + b = 0 \qquad (4.23)$$

Tabel 4.8 Dataset untuk menggunakan SVM pada Contoh 4.8.

x_1	x_2	y
1	0.5	−1
0.5	1	−1
1	2	+1
3	1	+1
0.25	2	−1

di mana w dan b adalah parameter, dan sesuai dengan garis lurus pada bidang 2-D. Tentu juga diharapkan hyperplane dapat memisahkan dua jenis data, yaitu semua yi yang berkorespondensi dengan titik-titik data pada satu sisi hyperplane adalah -1, dan 1 pada sisi lainnya. Buat f (x) = w^T x + b , gunakan f (x) > 0 pasangan titik untuk titik data dengan y = 1, dan f (x) < 0 pasangan titik untuk titik data dengan y = -1.

Contoh 4.7 Klasifikasi menggunakan Support Vector Machine dengan Training Samples

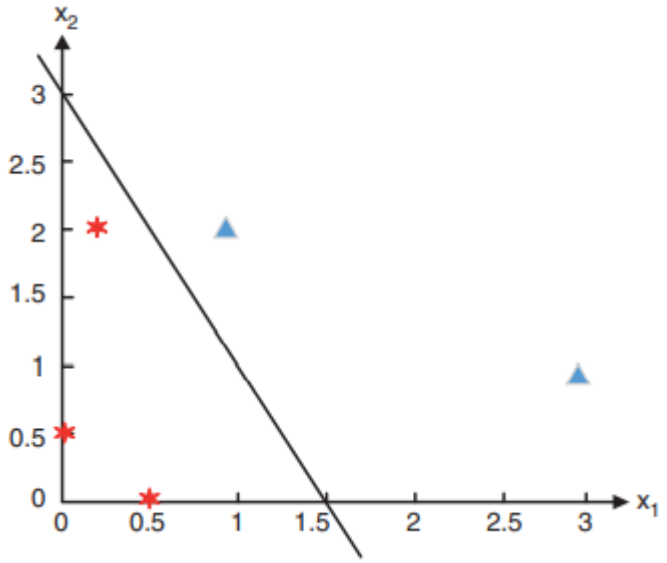
Data 2-D yang diberikan seperti pada Tabel 4.8. Maka satu garis lurus $2x_1 + x2_3 = 0$ dapat ditemukan untuk memisahkan data dalam tabel. Garis pemisah diplot pada Gambar 4.19.

Definisi Hyperplane Margin Maksimal

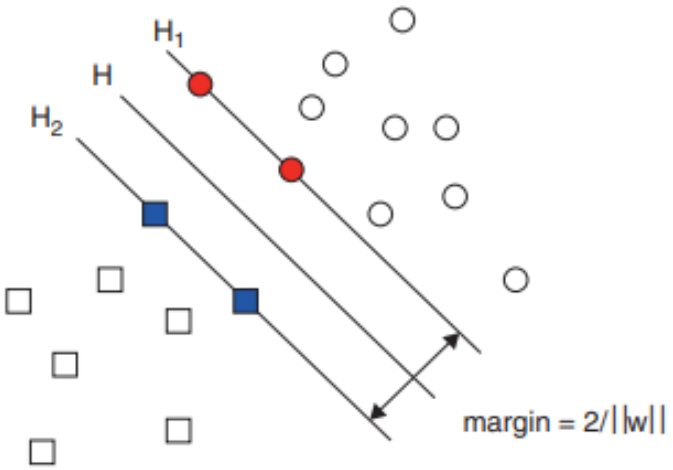
Pertimbangkan kotak dan lingkaran yang paling dekat dengan batas keputusan, seperti yang ditunjukkan pada Gambar 4.20; menyesuaikan parameter w dan b, dan dua hyperplanes paralel H1 dan H2 dapat diwakili oleh

$$H_1 : \ w^T x + b = 1 \quad H_2 \ : \ w^T x + b = -1 \tag{4.24}$$

Margin batas keputusan diberikan oleh jarak antara dua hyperplane tersebut.



Gambar 4.19 Ruang sampel dan solusi hyperplane untuk Contoh 4.7



Gambar 2.20 Memisahkan hyperplane secara linear dengan margin maksimum dari setiap kelas.

Untuk menghitung margin, buat x1 titik data pada H_1, dan x_2 titik data pada H_2, dan masukkan x_1 dan x_2 ke dalam rumus di atas, maka margin d dapat diperoleh dengan mengurangkan rumus: $w^T (x_1 - x_2) = 2$, oleh karena itu: kita memiliki $d = \frac{2}{||w||}$

Model SVM Formal

Tahap pelatihan SVM meliputi estimasi parameter w dan b dari data pelatihan, dan parameter yang dipilih harus memenuhi dua kondisi berikut dengan

$$\begin{cases} w^T x_i + b \geq 1 & y_i = 1 \\ w^T x_i + b \leq -1 & y_i = -1 \end{cases} \quad (4.25)$$

Kedua pertidaksamaan tersebut dapat dituliskan sebagai bentuk yang lebih kompak sebagai berikut:

$$y_i(w^T x_i + b) \geq 1 \quad i = 1, 2, \cdots, N \quad (4.26)$$

Maksimalisasi margin setara dengan minimalisasi fungsi tujuan berikut:

$$f(w) = \frac{||w||^2}{2} \quad (4.27)$$

Oleh karena itu, SVM diperoleh dengan mencari fungsi tujuan minimum:

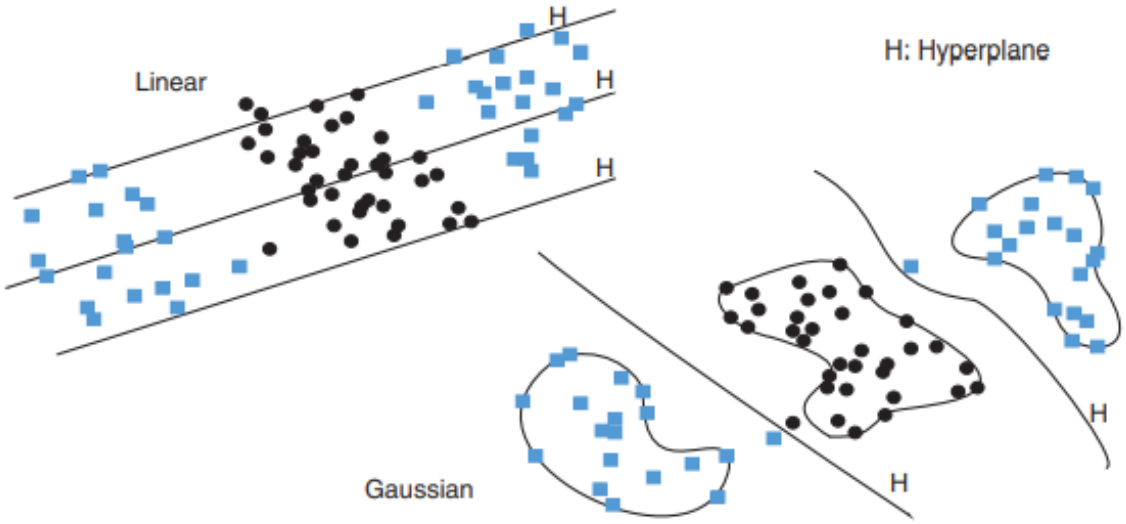
$$\min \frac{||w||^2}{2}, \quad subject\ to : y_i \left(w^T x_i + b\right) \geq 1 \quad i = 1, 2, \ldots, N \quad (4.28)$$

Ini adalah masalah optimasi cembung karena fungsi tujuannya adalah kuadrat dan kondisi kendalanya linier, dan dapat diselesaikan melalui pengali Lagrange standar. Kita mungkin perlu menyesuaikan model ketika sampel tidak dapat dipisahkan secara linier. Situasi ini ditunjukkan pada Gambar 4.21.

Hyperplanes Non-linear

Seperti yang ditunjukkan pada Gambar 4.21, terdapat outlier (atau noise), yang membuat ruang sampel tidak dapat dipisahkan secara linier. Beberapa variabel slack diperkenalkan untuk menghindari kasus ini:

$$\begin{cases} w^T x_i + b \geq 1 - \xi_i & y_i = 1 \\ w^T x_i + b \leq -1 + \xi_i & y_i = -1 \end{cases} \tag{4.29}$$



Gambar 4.21 Mesin vektor pendukung nonlinier.

Tanpa batasan pada sampel yang salah diklasifikasikan pada batas, algoritma pembelajaran dapat menemukan batas seperti itu dengan margin yang lebih luas dengan memungkinkan banyak sampel pelatihan yang salah diklasifikasikan. Fungsi tujuan dapat dimodifikasi sebagai aliran untuk menghindari batasan dengan nilai variabel slack yang besar:

$$f(w) = \frac{||w||^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)^k \tag{4.30}$$

di mana C dan k adalah parameter yang ditentukan oleh pengguna, yang berarti hukuman untuk instance pelatihan yang salah diklasifikasikan. Artinya, semakin banyak outlier, semakin besar nilai fungsi tujuan. C berarti bobot outlier, oleh karena itu model akhir diberikan sebagai

$$\min\left\{\frac{||w||^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)^k\right\} \tag{4.31}$$

$$\textit{Subject to}: \quad y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \quad i = 1, 2, \ldots, N, \xi_i \geq 0$$

Jika kita tidak dapat menemukan hyperplane untuk memisahkan data, yaitu SVM linier di atas tidak dapat menemukan solusi yang layak, kita perlu memperluas metode linier: SVM linier umumnya diperluas ke SVM nonlinier melalui dua langkah berikut: i) mengubah data input ke ruang dimensi yang lebih tinggi melalui pemetaan nonlinier; dan ii) mencari hyperplane pemisah di ruang baru. Misalnya, ketika data linier berdimensi rendah tidak dapat dipisahkan, data tersebut dapat dipetakan ke dimensi yang lebih tinggi untuk dapat dipisahkan setelah menggunakan fungsi Gaussian.

4.4 JARINGAN BAYESIAN DAN METODE ENSEMBLE

Thomas Bayes menemukan metode Bayesian untuk klasifikasi berdasarkan teori keputusan statistik. Kami memperkenalkan Naive Bayes dan Bayesian Network di bagian ini. Pengklasifikasi ini meningkatkan akurasi klasifikasi ketika digunakan dalam bidang medis, keuangan, dan banyak bidang lainnya. Kami akan memperkenalkan dasar-dasar pengklasifikasi Bayesian dan jaringan Bayesian Belief di Bagian 4.4.1 dan 4.4.2. Dalam beberapa kasus, satu algoritma *Machine learning* tidak dapat mencapai akurasi yang diinginkan. Akurasi dapat ditingkatkan dengan menggabungkan beberapa pengklasifikasi. Itu dikenal sebagai metode ensemble di Bagian 4.4.3. Untuk membuat keputusan yang andal, kami dapat menggabungkan beberapa metode *Machine learning* sederhana yang lemah dengan akurasi lebih besar dari 50%.

Pengklasifikasi Bayesian

Pertimbangkan sepasang variabel acak, X dan Y. Peluang gabungannya P(X = x, Y = y) dinyatakan oleh P(X, Y) = P(Y|X) × P(X) = P(X|Y) × P (Y), sehingga kita memiliki probabilitas kondisi terbalik:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{4.32}$$

Ini adalah Teorema Bayesian yang terkenal. Selama klasifikasi, variabel acak adalah kelas yang akan ditentukan, dan X adalah himpunan atribut. Kita perlu menghitung probabilitas kelas P(Y X_0), mengingat vektor atribut X_0 untuk item data pengujian. Nilai maksimum Y sesuai dengan kelas untuk data pengujian X_0. Pertimbangkan vektor atribut X = {X_1, X_2,...,X_k}, dan l nilai yang mungkin (atau kelas) untuk variabel acak Y = {Y_1, Y_2,...,Y_l}. Kami menyebut P(Y | X) probabilitas posterior dan P(Y) probabilitas sebelumnya dari Y. Kami berasumsi bahwa semua atribut secara statistik independen. Jadi kita dapat menghitung probabilitas bersyarat sebagai:

$$P(X|Y = y) = \prod_{j=1}^{k} P(X_j|Y = y) \tag{4.33}$$

Pengklasifikasi Bayesian naif menghitung probabilitas posterior untuk setiap kelas Y dengan:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} = \frac{P(Y)\prod_{j=1}^{k} P(X_j|Y)}{P(X)} \quad (4.34)$$

Metode klasifikasi Bayesian memprediksi X ke kelas dengan probabilitas posterior tertinggi. Probabilitas posterior $P(Y_i \mid X)$, i= 1, 2, ..., l untuk setiap kombinasi X dan Y, kemudian tentukan Yr dengan mencari $\max_{i = 1,2,\ldots,l}$ $P(Y_i \mid X)$, dan mengklasifikasikan X ke kelas Y_r. Karena P(X) adalah sama untuk semua kelas, maka cukup untuk menemukan pembilang maksimum:

$$P(Y)\prod_{j=1}^{k} P(X_j|Y)$$

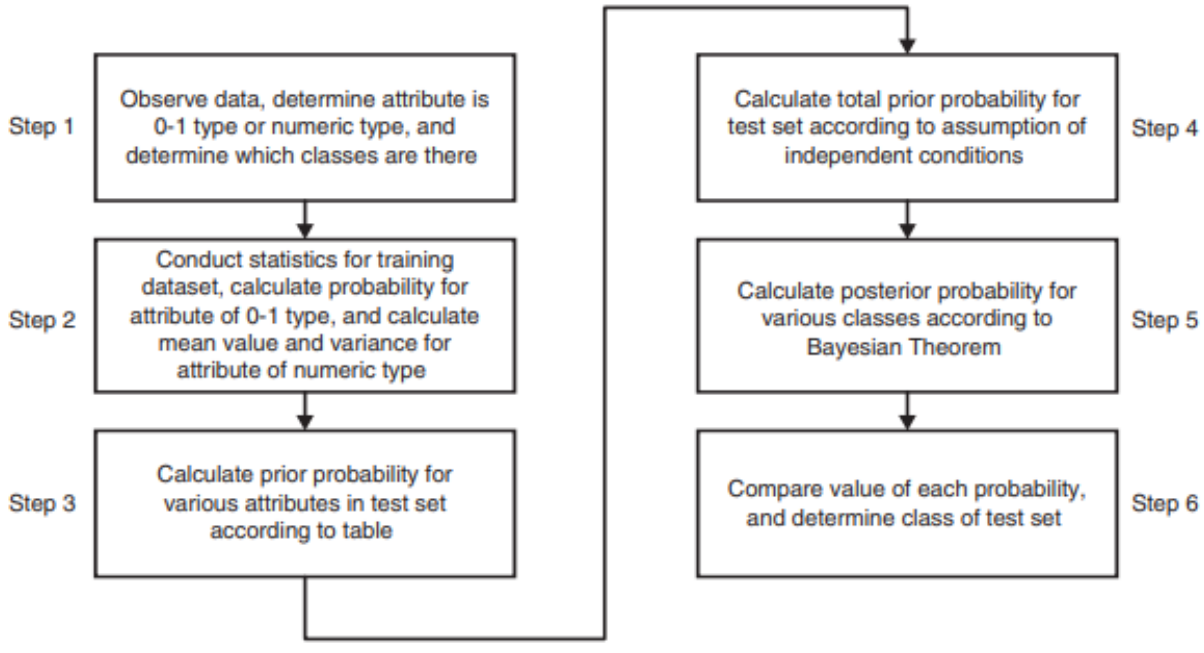
dalam Persamaan 4.34. Jadi, kita hanya menghitung sebagai berikut:

$$\max_{Y} \; P(Y)\prod_{j=1}^{k} P(X_j|Y) \quad (4.35)$$

Klasifikasi Naive Bayesian mengikuti enam langkah pada Gambar 4.22.

Contoh 4.8 Pengklasifikasi Bayesian dan Analisis Hasil Deteksi

Mengingat dataset pelatihan hewan berikut, setiap item data dapat diberi label sebagai mamalia atau non-mamalia, tetapi tidak keduanya. Setiap item data dicirikan oleh empat atribut independen A = <A1, A2, A3, A4 > = <Melahirkan, Bisa Terbang, Hidup di Air, Memiliki Kaki>. Kita perlu membangun model classifier Bayesian dari set pelatihan. Model akan diterapkan:



Gambar 4.22 Langkah-langkah computing dalam proses Klasifikasi Naive Bayesian.

Mengingat dataset pelatihan hewan berikut, setiap item data dapat diberi label sebagai mamalia atau non-mamalia, tetapi tidak keduanya. Setiap item data dicirikan oleh empat atribut independen A = <A1, A2, A3, A4 > = <Melahirkan, Bisa Terbang, Hidup di Air, Memiliki Kaki>. Kita perlu membangun model classifier Bayesian dari set pelatihan. Model akan diterapkan untuk mengklasifikasikan hewan yang tidak berlabel sebagai mamalia (M) atau non-mamalia (N). Perhatikan bahwa atribut A3: "hidup di air" berarti hewan itu terutama hidup di air, tidak hanya sesekali berenang di air. Nilai “kadang-kadang dalam air” dianggap sebagai entri “tidak” pada Tabel 4.9.

Tabel 4.9 Contoh data dalam dataset pelatihan untuk Contoh 4.8.

Nama	Melahirkan	Bisa terbang	Hidup di Air	punya kaki	Kelas
manusia	Ya	tidak	tidak	Ya	mamalia
ular piton	tidak	tidak	tidak	tidak	non-mamalia
ikan salmon	tidak	tidak	Ya	tidak	non-mamalia
Paus	Ya	tidak	Ya	tidak	mamalia
katak	tidak	tidak	kadang-kadang	Ya	non-mamalia
komodo	tidak	tidak	tidak	Ya	non-mamalia
kelelawar	Ya	Ya	tidak	Ya	mamalia
merpati	tidak	Ya	tidak	Ya	non-mamalia
kucing	Ya	tidak	tidak	Ya	mamalia
hiu macan tutul	Ya	tidak	Ya	tidak	non-mamalia
penyu	tidak	tidak	kadang-kadang	Ya	non-mamalia
pinguin	tidak	tidak	kadang-kadang	Ya	non-mamalia
landak	Ya	tidak	tidak	Ya	mamalia
belut	tidak	tidak	Ya	tidak	non-mamalia
salamander	tidak	tidak	kadang-kadang	Ya	non-mamalia
monster gila	tidak	tidak	tidak	Ya	non-mamalia
platipus	tidak	tidak	tidak	Ya	mamalia
burung hantu	tidak	Ya	tidak	Ya	non-mamalia
lumba-lumba	Ya	tidak	Ya	tidak	mamalia
burung rajawali	tidak	Ya	tidak	Ya	non-mamalia

Dengan menggunakan sampel pada Tabel 4.10, kami menghitung probabilitas sebelumnya: P (M)= 7/20 dan P (N) = 13/20. Pertimbangkan item data pengujian tidak berlabel yang dicirikan oleh vektor atribut: A* = < A1, A2, A3, A4> = <ya, tidak, ya, tidak>. Pertama, kami menghitung nilai probabilitas pengujian sebagai berikut:

Tabel .4.10 Probabilitas atribut pre-test untuk dataset sampel pada Tabel 4.9.

Fitur / Kemungkinan		Melahirkan		Bisa terbang		Hidup di air		Punya kaki	
		Yes	No	Yes	No	Yes	No	Yes	No
Menghitung	M	6	1	6	1	2	5	2	5
	N	1	12	10	3	10	3	4	9
Kemungkinan	M	6/7	1/7	6/7	1/7	2/7	5/7	2/7	5/7
	N	1/13	12/13	10/13	3/13	10/13	3/10	4/13	9/13

Karena P (M|A*) > P (N|A*), makhluk dengan vektor atribut A* ini dideteksi sebagai mamalia. Dengan kata lain, makhluk yang melahirkan, tidak bisa terbang, hidup di air, dan tidak berkaki itu tergolong mamalia. Sekarang, mari kita menganalisis akurasi penggunaan classifier Bayesian dengan menguji empat makhluk menggunakan metode di atas. Kami daftar hasil berikut seperti yang tercantum Tabel 4.11.

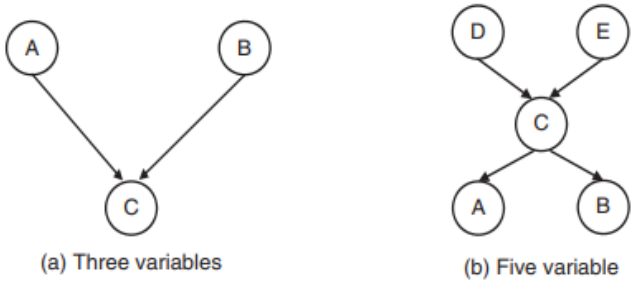
Kami memperoleh probabilitas posterior P(M|A1, A2, A3, A4) dan P(N|A1, A2.A3, A4) untuk masing-masing dari empat hewan uji. Pilih kelas dengan probabilitas tertinggi sebagai kelas prediksi. Membandingkan hasil prediksi dengan kelas makhluk sebenarnya pada Tabel 4.11, kami menemukan empat kemungkinan status prediksi di kolom sebelah kanan.

TP (benar positif) mengacu pada kasus yang benar diprediksi dengan benar, TN (Benar Negatif) untuk kasus yang benar salah diprediksi, FP (False Positive) berarti kasus salah diprediksi dengan benar dan FN (False negatif) untuk kasus palsu yang diprediksi secara tidak benar. Dengan mendasarkan hasil perbandingan pada Tabel 4.11, diperoleh hasil kinerja sebagai berikut: TP = 2/4 = 0,5, TN = $\frac{1}{4}$ = 0,25, FP = 0, dan FN = $\frac{1}{4}$ = 0,25. Kemudian kami menggunakan dua metrik kinerja untuk menilai akurasi pengklasifikasi Baysian:

$$\text{Prediction accuracy} = (TP + TN) / (TP + TN + FP + FN) = 0.75$$
$$\text{Prediction error} = (FP + FN) / (TP + TN + FP + FN) = 0.25$$

Tabel 4.11 Hasil prediksi empat hewan dibandingkan dengan kelas sebenarnya.

Nama Hewan	Melahirkan	Bisa terbang	Hidup di Air	punya kaki	Kelas yang Diprediksi	Kelas Sebenarnya	Status Prediksi
Anjing	Ya	tidak	tidak	Ya	M	M	Tp
aliran tunggal	tidak	tidak	tidak	Ya	n	M	FN
Buaya	tidak	tidak	Ya	Ya	n	n	TN
Kuda	Ya	tidak	tidak	Ya	M	M	Tp



Gambar 4.23 Dua jaringan kepercayaan Bayesian dengan dua jumlah variabel yang berbeda.

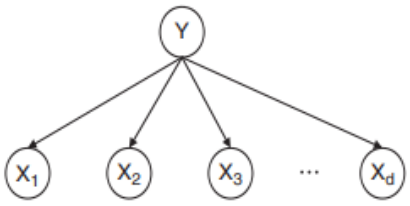
Akurasi atau kesalahan berasal dari asumsi lemah bahwa semua atribut independen. Secara umum, semakin besar set pelatihan untuk mencakup semua vektor atribut yang mungkin, semakin tinggi akurasi prediksi. Selanjutnya, jika salah satu dari probabilitas bersyarat individu $P(Ai \mid C) = N_{ic} / N_c = 0$ karena kasus $N_{ic} = 0$ dari dataset pelatihan (Tabel 4.11), seluruh probabilitas posterior dalam Persamaan (4.34) menjadi nol. Hal ini dapat dihindari dengan mengasumsikan nilai offset $P(Ai|C) = (N_{ic}+1)/(N_c + c) = 1/(N_c + c)$, di mana c adalah jumlah kelas yang dipertimbangkan. Metrik kinerja akan dipelajari lebih lanjut di Bab 8, ketika penerapan pengklasifikasi Baysian digunakan untuk prediksi penyakit kronis.

Jaringan Kepercayaan Bayesian

Jaringan Naive Bayesian mengasumsikan bahwa semua atribut secara statistik independen. Asumsi ini terlalu ketat dalam beberapa kasus. Untuk melonggarkan asumsi ini, jaringan kepercayaan Bayesian diperkenalkan dengan probabilitas bersyarat kelas. Jaringan kepercayaan Bayesian adalah representasi grafis dari hubungan antar atribut. Ada dua komponen utama: i) grafik asiklik berarah, yang mewakili ketergantungan antar variabel; dan ii) tabel probabilitas, menghubungkan setiap node dan node induknya secara langsung.

Pertimbangkan tiga variabel acak, A, B dan C, di mana A dan B independen dari masing-masing lainnya dan keduanya memiliki dampak langsung pada variabel acak ketiga. Jadi hubungan mereka diwakili oleh dua DAG pada Gambar 4.23(a). Pertimbangkan lima variabel, A, B, C, D dan E. Variabel A dan B, D dan E saling bebas, D dan E memiliki dampak langsung pada C, dan variabel C mempengaruhi A dan B. Jadi, kita dapatkan grafik DAG pada Gambar 4.23(b).

Pertimbangkan situasi di mana variabel independen, yaitu Nave Bayesian Network, yang merupakan jenis khusus dari jaringan kepercayaan Bayesian. Kami menggunakan Y untuk menunjukkan kelas target, $\{X_1, X_2, , X_d\}$ adalah himpunan atributnya, dan jaringan kepercayaan Bayesian ditunjukkan pada Gambar 4.24.



Gambar 2.24 Asumsi independensi bersyarat dari pengklasifikasi Naive Bayesian.

Untuk menunjukkan hubungan antar variabel secara lebih jelas, kami menetapkan aturan sebagai berikut: jika ada busur dari X ke Y, maka X adalah induk Y dan Y adalah anak dari X; jika ada busur dalam jaringan dari Y ke Z, maka X adalah nenek moyang dari Z, dan Z adalah keturunan dari X. Sebagai contoh, pada Gambar 4.23 (b), D adalah induk C dan nenek moyang dari A dan B, sedangkan C adalah anak dari D, dan A dan B adalah keturunan dari D. Selain independensi yang dibutuhkan oleh topologi jaringan, setiap node dikaitkan dengan tabel probabilitas:

1) Jika simpul X tidak memiliki simpul induk, tabel ini hanya berisi probabilitas apriori P(X).
2) Jika simpul X hanya memiliki satu simpul induk Y, tabel ini berisi probabilitas bersyarat P(X | Y).
3) Jika simpul X memiliki beberapa simpul induk $\{Y_1, Y_2, ..., Y_m\}$, tabel ini berisi probabilitas bersyarat $P(X \mid Y_1, Y_2, ..., Y_m)$.

Singkatnya, pemodelan jaringan kepercayaan Bayesian terdiri dari dua langkah. Pertama, buat struktur jaringan dan kemudian perkirakan probabilitas di busur. Topologi jaringan diperoleh melalui pengkodean yang didukung oleh pengetahuan domain subjektif. Nilai-nilai probabilitas diperoleh dengan probabilitas bersyarat. Algoritma 4.1 menetapkan prosedur sistematis dalam menggunakan jaringan kepercayaan Bayesian.

Algoritma 4.1 Penggunaan Bayesian Belief Network untuk Analisis Prediktif

Input: d Jumlah variabel, T Urutan umum variabel.

Output: topologi jaringan kepercayaan Bayesian

Prosedur:

1) Pertimbangkan $T = (X_1, X_2, , X_d)$ sebagai satu orde umum variable
2) untuk I = 1 tod
3) Jadikan sebagai $XT_{(i)}$ variabel tertinggi ke-i dalam T
4) Jadikan $C(XT_{(i)})$ sebagai himpunan variabel sebelum $XT_{(i)}$
5) Hilangkan semua variabel dalam $C(XT_{(i)})$ tanpa berdampak pada Xi dengan pengetahuan masa depan
6) Gambarlah busur antara variabel yang tersisa dari $C(XT_{(i)})$ dan $XT_{(i)}$
7) akhir
8) Keluarkan grafik topologi yang digambar, yaitu topologi jaringan kepercayaan Bayesian.

Contoh 4.9 Penggunaan Bayesian Belief Network dalam Prediksi Diabetes

Secara umum, diabetes dikaitkan dengan banyak faktor seperti obesitas, riwayat keluarga, glukosa darah, gula darah dan lipid darah, dll. Ini juga merupakan anteseden seperti obesitas dan riwayat keluarga. Gejala lain dapat disebabkan oleh glukosa darah dan gula darah, juga dikenal sebagai konsekuensi. Jaringan kepercayaan Bayesian dapat memodelkan anteseden dan konsekuensi pada saat yang sama, sedangkan jaringan Naive Bayesian hanya dapat memodelkan anteseden. Sebagian data pemeriksaan fisik pasien disajikan pada Tabel 4.12, dimana 1 menunjukkan gejala dan 0 sebaliknya.

Menurut pengalaman dan akal sehat, obesitas dan riwayat keluarga sama-sama berhubungan dengan diabetes. Glukosa darah dan lipid darah juga berhubungan dengan diabetes. Atribut diurutkan sebagai T = {Obesitas, riwayat keluarga, gula darah, lipid darah, diabetes} Berdasarkan analisis di atas, kami memperoleh jaringan kepercayaan Bayesian pada Gambar 4.25.

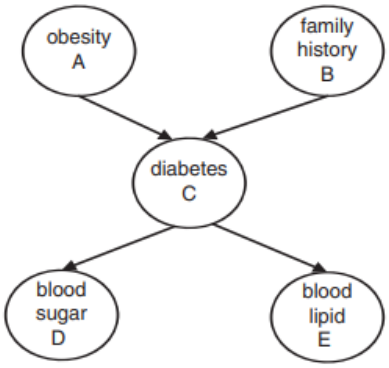
Tabel 4.12 Data pemeriksaan suspek pasien diabetes.

ID pasien	Kegemukan	Sejarah keluarga	Gula darah	lipid darah	Diabetes
1	1	0	1	1	1
2	0	1	1	0	0
3	1	1	0	1	0
4	0	0	0	0	1
5	0	0	0	0	0
6	1	0	1	0	1
7	1	1	1	1	1
8	0	1	0	1	0
9	0	0	1	0	0
10	1	1	1	1	1
10	0	1	0	0	0
12	1	0	0	1	0
13	1	0	1	0	0
14	0	0	0	0	0
15	0	1	1	1	1
16	0	0	0	0	0
17	0	0	0	0	0

Berdasarkan peta topologi di atas, kita dapat menyederhanakan probabilitas bersyarat, misalnya P (A |B) = P(A) karena obesitas (A) dan riwayat keluarga (B) adalah independen satu sama lain, dan dalam kombinasi dengan tabel yang diberikan, kita bisa mendapatkan probabilitas diabetes. Misalnya P(C | A, B), hasil perhitungan tabel probabilitas ini disajikan pada Tabel 4.13.

Untuk mengetahui apakah diabetes berpengaruh terhadap glukosa dan lipid darah, misalnya hasil perhitungan tabel probabilitas untuk P(D,E | C) disajikan sebagai Tabel 4.14. Menggunakan probabilitas bersyarat Bayesian, kami menghitung apakah ada kemungkinan diabetes dalam beberapa kasus lain. Sebagai contoh, misalkan kita mendeteksi beberapa orang dengan glukosa darah tinggi dan lemak darah tinggi, lalu berapakah probabilitas risiko diabetes? Yaitu P(C = ya | D = ya, E = ya). Jelas bahwa hasil yang diketahui membuat kita melepaskan probabilitas, yang sesuai dengan teorema probabilitas bersyarat Bayesian, maka:

$$\begin{cases} P(C = yes|D = yes, E = yes) = \dfrac{P(D = yes, E = yes|C = yes)}{P(D = yes, E = yes)} \cdot P(C) = \dfrac{2/3}{4/17} \times \dfrac{6}{17} = 1 \\ P(C = yes|D = yes, E = no) = \dfrac{P(D = yes, E = yes|C = no)}{P(D = yes, E = yes)} \cdot P(C) = \dfrac{0}{4/17} \times \dfrac{6}{17} = 0 \end{cases} \tag{4.36}$$



Gambar 4.25 Jaringan kepercayaan Bayesian untuk penderita diabetes pada Contoh 4.9.

Tabel 4.13 Probabilitas bersyarat untuk pasien diabetes.

Obesitas (A)	Riwayat keluarga (B)	penderita diabetes	Tidak Ada Diabetes
YA	YA	2/3	1/3
	TIDAK	1/2	1/2
TIDAK	YA	1/4	3/4
	TIDAK	1/6	5/6

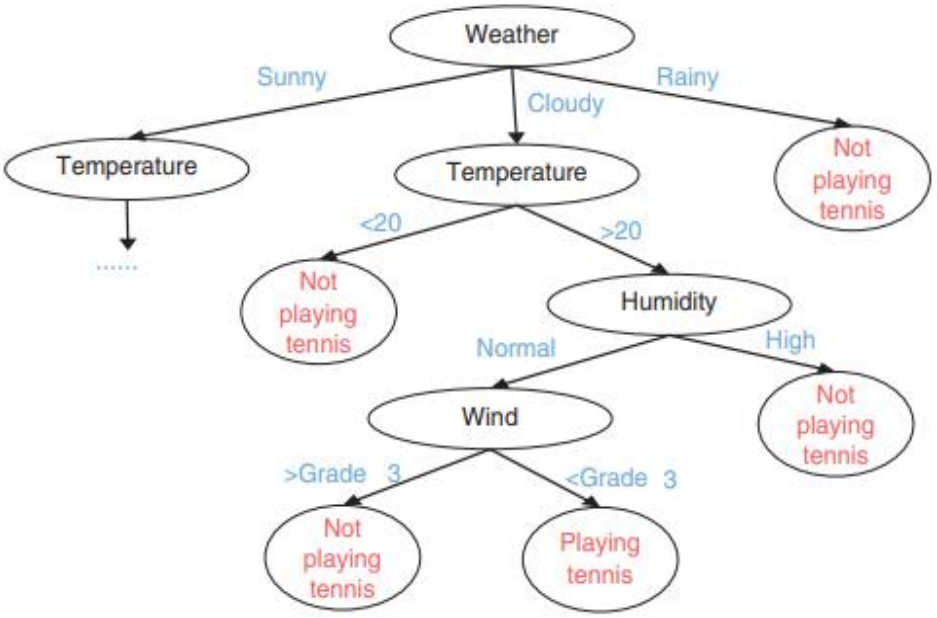
Jaringan kepercayaan Bayesian menunjukkan bahwa pasien menderita diabetes. Tabel 4.14 menunjukkan bahwa pasien obesitas dengan riwayat keluarga diabetes, memiliki probabilitas diabetes 2/3. Selanjutnya, dengan glukosa darah tinggi, kemungkinan pasien menderita diabetes dihitung dengan:

$$\begin{aligned} &P(C = yes|A = yes, B = yes, D = yes) \\ &= \frac{P(D = yes \mid C = yes, A = yes, B = yes)}{P(D = yes \mid A = yes, B = yes)} \times P(C = yes|A = yes, B = yes) \\ &= \frac{P(D = yes \mid C = yes)P(C = yes|A = yes, B = yes)}{\sum_i P(D = yes \mid C = i)P(C = i|A = yes, B = yes)} \qquad i = yes, no \\ &= \frac{5/6 \times 2/3}{5/6 \times 2/3 + 1/6 \times 1/3} = \frac{10}{11} \end{aligned} \tag{4.37}$$

Kami menyimpulkan bahwa pasien tersebut memiliki probabilitas tinggi 10/11 menjadi diabetes.

Tabel 4.14 Probabilitas glukosa darah dan lipid untuk pasien diabetes.

Penderita diabetes (C)	Glukosa darah (D)	Lipid darah (E)	Probabilitas (P)
YA	Tinggi	Tinggi	4/6
	Tinggi	Normal	1/6
	Normal	Tinggi	0/6
	Normal	Normal	1/6
TIDAK	Tinggi	Tinggi	0/11
	Tinggi	Normal	3/11
	Normal	Tinggi	3/11
	Normal	Normal	5/11



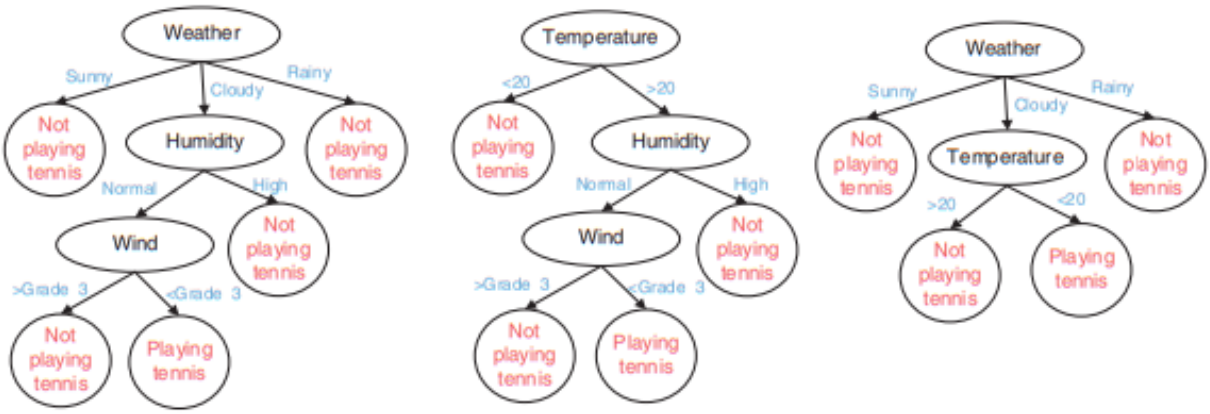
Gambar4.26 Pohon keputusan untuk turnamen tenis.

Hutan Acak dan Metode Ensemble

Teknik klasifikasi umum seperti jaringan Bayesian, pohon keputusan dan mesin vektor pendukung, menggunakan pengklasifikasi tunggal yang diperoleh dari data pelatihan untuk memprediksi label kelas yang tidak diketahui. Melalui agregasi beberapa pengklasifikasi, akurasi klasifikasi ditingkatkan, dan kami menyebut teknik ini sebagai model ensemble atau kombinasi pengklasifikasi. Random forest sebagai salah satu metode klasifikasi kombinasi adalah jenis metode kombinasi khusus yang ditujukan untuk klasifikasi pohon keputusan.

Melalui kombinasi beberapa pohon keputusan, prediksi dibuat, di mana setiap pohon dihasilkan oleh nilai himpunan independen berdasarkan vektor acak. Misalnya, jika kita ingin memutuskan apakah hari tertentu cocok untuk bermain tenis menurut cuaca, suhu, kelembapan, dan kondisi angin, Anda memiliki pohon keputusan yang ditunjukkan pada Gambar 4.26. Sekarang, dengan menggunakan beberapa pohon keputusan untuk meningkatkan akurasi, kita dapat membagi empat atribut menjadi beberapa kelompok atribut, seperti {cuaca, kelembaban, angin}, {suhu, kelembaban, angin}, {cuaca, suhu} dan seterusnya, seperti yang ditunjukkan pada Gambar 4.27.

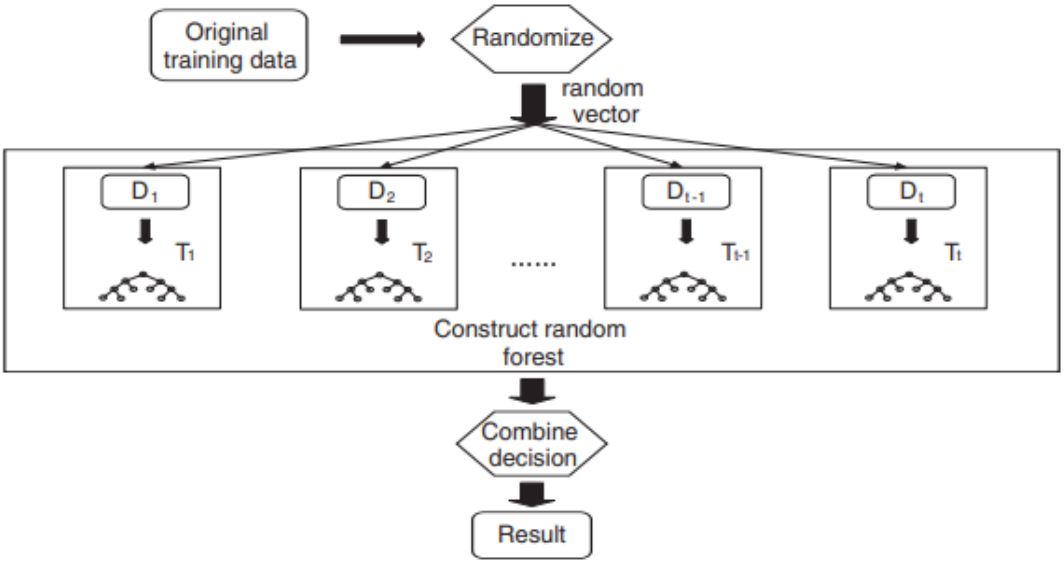
Dengan cara ini, satu pohon keputusan dapat dibagi menjadi tiga pohon keputusan, ketika pengambilan keputusan setiap pohon keputusan akan sesuai dengan hasil: bermain tenis, atau tidak bermain tenis. Dengan cara ini, kita mendapatkan tiga hasil pengambilan keputusan, kemudian mengetahui hasil mana yang memiliki suara terbanyak, dan hasil akhirnya adalah yang memiliki suara terbanyak. Misalnya, dalam kasus {cerah, lebih besar dari 20 derajat, kelembaban udara tinggi, tidak ada angin} akankah kita bermain tenis? Dengan pohon keputusan pada Gambar 4.26, kita mengetahui hasil yang pertama, metode pohon keputusan ketiga tidak bermain tenis dan hasil pohon keputusan kedua adalah bermain tenis. Karena bermain tenis mendapat 1 suara dan tidak bermain tenis mendapat 2 suara, maka hasil akhirnya bukan bermain tenis.



Gambar 4.27 Keputusan hutan acak untuk bermain tenis dalam berbagai kondisi cuaca.

Metode di mana vektor acak diperoleh dari atribut acak serupa seperti dijelaskan di atas, kemudian vektor acak digunakan untuk membangun pohon keputusan, dan setelah pohon keputusan dibangun, hasil pemungutan suara mayoritas digunakan untuk menggabungkan prediksi, yang dikenal sebagai Forest-RI, di antaranya RI mengacu pada pemilihan input t-random. Kekuatan keputusan hutan acak yang diperoleh dengan metode ini tergantung pada dimensi vektor acak, yaitu jumlah bilangan karakteristik yang diperoleh setiap pohon, F, biasanya F = log2d + 1, di mana d adalah jumlah total pohon. atribut.

Jika jumlah atribut asli, d, terlalu kecil, sulit untuk memilih sekumpulan atribut independen acak untuk membangun pohon keputusan. Sebuah metode untuk meningkatkan ruang atribut adalah dengan membuat kombinasi linier fitur, menggunakan atribut input L dari kombinasi linier untuk membuat atribut baru, dan kemudian menggunakan atribut baru yang dibuat untuk membentuk vektor acak, dan akhirnya membangun banyak bagian. dari pohon keputusan. Metode pengambilan keputusan hutan secara acak ini disebut sebagai Forest-RC. Proses umum pengambilan keputusan hutan secara acak adalah seperti pada Algoritma 4.2. Proses pembentukan hutan secara acak ditunjukkan pada Gambar 4.28.



Gambar 4.28 Proses penggunaan random forest untuk pengambilan keputusan ensemble.

Algoritma 4.2 Penggunaan Hutan Acak untuk Pengambilan Keputusan dalam Klasifikasi
Input: R: sampel ramalan, L: matriks atribut
Output: Hasil keputusan
Prosedur:

1) Hitung dimensi vektor F
2) Buat vektor atribut acak dimensi-F untuk membentuk koleksi, C
3) Pohon keputusan dibangun sesuai dengan elemen C, dan hutan acak dibuat
4) Membuat keputusan di setiap pohon keputusan
5) Hitung dan keluarkan hasil akhir dengan suara terbanyak
6) Akhir

Contoh 4.10 Prediksi Penderita Diabetes dengan Model Hutan Acak

Bagian dari data pemeriksaan fisik rumah sakit kehidupan nyata dari Cina tengah disajikan pada Tabel 4.15, dan pengumpulan data meliputi berat badan, gula darah, kandungan lipid dan apakah pasien menderita diabetes (1: pasien, 0 : biasa). Data indeks fisik seseorang adalah {berat:

60, gula darah: 6,8, lipid darah: 1,5}, apakah dapat diketahui bahwa orang tersebut menderita diabetes.

Tabel 4.51 Bagian dari data pemeriksaan fisik dari Rumah Sakit Wuhan.

No.	Berat (A) (kg)	Gula darah (B) (mmol/L)	Lipid darah (C) (mmol/L)	Diabetes (D)
1	68.4	17.5	7.7	1
2	64.3	4.7	1.33	0
3	65.4	8.6	2.48	0
4	62.0	14.3	4.67	1
5	81.5	8.5	0.82	1
6	58.3	5.0	1.99	0
7	55.2	4.6	0.86	0
8	84.3	5.8	2.34	1
9	85.6	5.9	2.54	0
10	54.7	5.7	2.63	1

Untuk meningkatkan akurasi prediksi, kita dapat mempertimbangkan untuk membuat prediksi dengan metode random forest. Kita perlu menentukan dimensi vektor acak F = log2d + 1 = 2. Mengingat atribut dalam contoh sedikit, dan untuk membuat korelasi antara vektor acak lebih rendah, kita dapat menentukan tiga vektor acak berikut: {bobot , gula darah}, {berat badan, lipid darah} dan {gula darah, lipid darah}. Kemudian ditentukan urutan beberapa properti seperti di bawah ini. Hal ini ditentukan oleh entropi informasi. Properti yang entropi informasinya meningkat paling banyak berada di bagian atas pohon keputusan, dan sejenisnya (ada pengenalan rinci tentang ini di bagian pohon keputusan).

$$Entropy(D) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$Entropy(A) = \frac{1}{2} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) + \frac{1}{2} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.9710$$

$$Entropy(B) = \frac{4}{10} \times \left(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right) + \frac{6}{10} \times \left(-\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}\right) = 0.8755$$

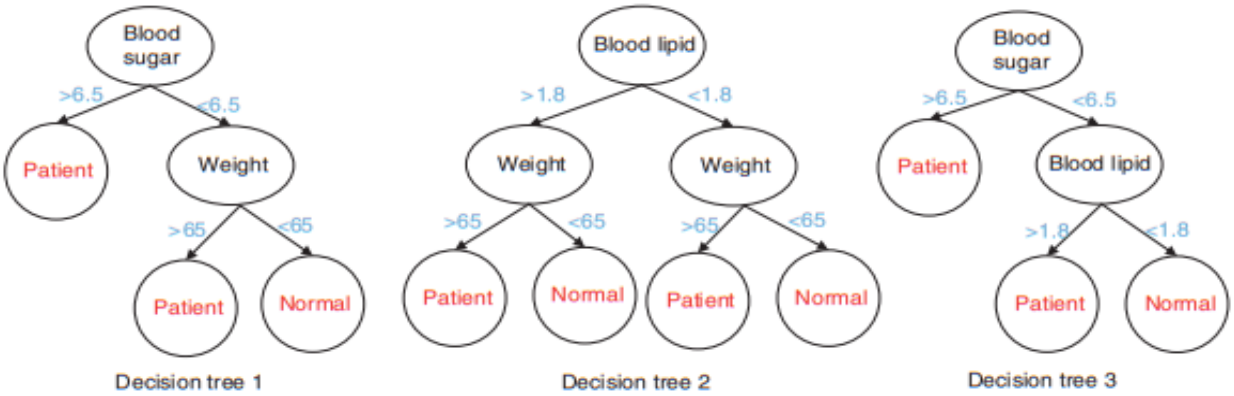
$$Entropy(C) = \frac{7}{10} \times \left(-\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}\right) + \frac{3}{10} \times \left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right) = 0.9651$$

Kenaikan entropi gula darah, lipid darah dan berat badan masing-masing adalah:

$$\Delta Entropy(A) = Entropy(D) - Entropy(A) = 0.0290$$
$$\Delta Entropy(B) = Entropy(D) - Entropy(B) = 0.1245$$
$$\Delta Entropy(C) = Entropy(D) - Entropy(C) = 0.0349$$

Oleh karena itu, kandungan gula darah dan lipid darah lebih penting, dan harus ditempatkan lebih dekat ke akar pohon keputusan. Urutannya adalah gula darah, lipid darah dan berat badan. Kita dapat membangun hutan acak berikut, seperti yang ditunjukkan pada Gambar 4.29.

Indeks pemeriksaan adalah {berat: 60, gula darah 6,8, lipid darah: 1,5}, dan sesuai dengan pohon keputusan 1 untuk pasien, hasilnya adalah YA, pohon keputusan 2 adalah TIDAK, dan pohon keputusan 3 adalah YA. Situasi terakhir adalah 2 suara untuk ya dan 1 suara untuk tidak, sehingga hasil awal menunjukkan bahwa orang tersebut menderita diabetes.



Gambar 4.29 Representasi hutan acak diabetes.

4.5 KESIMPULAN

Machine learning menjadi sangat diminati dengan munculnya ilmu data dan industri *Big data*. Taksonomi algoritma ML diperkenalkan untuk membedakan antara keluarga yang diawasi, tidak diawasi dan semi-diawasi. Algoritme ML terawasi yang dipelajari dalam bab ini lebih sering digunakan daripada yang tidak terawasi untuk dipelajari di Bab 5. Metode *Machine learning* yang dipelajari dalam dua bab ini diterapkan untuk analitik *Big data*. *Deep learning* dibahas di Bab 6 ketika jaringan saraf tiruan diperkenalkan. Algoritma regresi dan klasifikasi diawasi oleh data pelatihan. Demikian pula, pelatihan dipraktikkan dalam menggunakan pohon keputusan, mesin vektor pendukung, dan jaringan Bayesian. Di Bab 5, kita akan mempelajari cara memilih algoritme pembelajaran ML, termasuk yang tidak diawasi. Pembaca didorong untuk belajar lebih lanjut dengan memecahkan masalah pekerjaan rumah.

Tugas dan Latihan

1. Ada sekelompok wanita, tanpa informasi langsung tentang berat badan mereka yang diukur. Cari tahu metode untuk memprediksi urutan berat badan mereka tanpa menanyakannya secara langsung. Apa atribut atau fitur yang akan digunakan dalam proses prediksi Anda? Misalnya, atribut dapat berupa tinggi badan, usia, ras, kekayaan, atau faktor terkait lainnya. Membenarkan kelayakan model pembelajaran Anda untuk memprediksi bobot mereka dengan beberapa tingkat akurasi.
2. Pertimbangkan kepadatan Nitric Oxide (NO), suatu polutan udara, di lingkungan perkotaan di mana kendaraan mengeluarkan NO selama pergerakannya. Polusi di udara

terbukti berbahaya bagi kesehatan manusia. Kepadatan NO dikaitkan dengan lalu lintas kendaraan, suhu, kelembaban udara dan kecepatan angin. Tabel 4.16 menunjukkan data lingkungan yang dikumpulkan di berbagai daerah yang diamati. Gunakan metode regresi linier untuk mengestimasi kerapatan NO dengan vektor data {1436, 28.0, 68, 2.00}.

3. Informasi pengguna telepon pintar diberikan pada Tabel 4.17. Untuk setiap hari, data rata-rata yang dikumpulkan mencakup total durasi panggilan (menit), volume lalu lintas seluler (MB), jumlah panggilan masuk, dan apakah pengguna di rumah atau di luar (1: rumah, 0: di luar). Mengingat data pengujian (A: 90, B: 60, C: 8), tentukan apakah orang ini ada di rumah? Demikian juga, bagaimana dengan pengguna lain dengan statistik panggilan (A: 80, B: 50, C: 10)?
4. Ketika sebuah bisnis mengajukan permohonan pinjaman ke bank atau organisasi keuangan, pemberi pinjaman perlu menilai kemampuan kredit peminjam. Biarkan y = 0 menunjukkan peminjam dengan catatan buruk, sedangkan y = 1 mewakili peminjam yang dikreditkan. Tiga ciri peminjam diwakili oleh X1, X2 dan X3 pada Tabel 4.18. Bangun model prediksi untuk menguji pelanggan tertentu dengan catatan kredit (X1, X2, X3) = (–25, 2.5, 0.5). Nilai keakuratan model prediksi Anda.

Tabel 4.16 Densitas oksida nitrat diukur di berbagai daerah yang diamati.

Lalu Lintas Kendaraan (X1)	Suhu (X2)	Kelembaban udara (X3)	Kecepatan angin (X4)	Kepadatan NO (Y)
1300	20	80	0.45	0.066
948	22.5	69	2.00	0.005
1444	23.0	57	0.50	0.076
1440	21.5	79	2.40	0.011
786	26.5	64	1.5	0.001
1084	28.5	59	3.00	0.003
1652	23.0	84	0.40	0.170
1844	26.0	73	1.00	0.140
1756	29.5	72	0.9	0.156
1116	35.0	92	2.80	0.039
1754	30.0	76	0.80	0.120
1656	20.0	83	1.45	0.059
1200	22.5	69	1.80	0.040
1536	23.0	57	1.50	0.087
1500	21.8	77	0.60	0.120
960	24.8	67	1.50	0.039

Tabel 4.17 Bagian dari informasi aplikasi pengguna ponsel pintar.

ID	Total Durasi Panggilan (A)	Volume (B)	Frekuensi (C)	Di rumah (D)
1	20	45	2	1
2	120	46	4	1
3	90	55	10	0
4	81	56	19	0
5	200	55	8	0

Tabel 4.18 Sampling data laporan kredit bank debitur.

X1	X2	X3	Y	X1	X2	X3	Y
–48.2	6.8	1.6	0	43.0	16.4	1.3	1
–49.2	–17.2	0.3	0	47.0	16.0	1.9	1
–19.2	–36.7	0.8	0	-3.3	4.0	2.7	1
–18.1	–6.5	0.9	0	35.0	20.8	1.9	1
–98.0	–20.8	1.7	0	46.7	12.6	0.9	1
–129.0	–14.2	1.3	0	20.8	12.5	2.4	1
–4.0	–15.8	2.1	0	33.0	23.6	1.5	1
–8.7	–36.3	2.8	0	26.1	10.4	2.1	1
–59.2	–12.8	2.1	0	68.6	13.8	1.6	1
–13.1	–17.6	0.9	0	37.3	33.4	3.5	1
–38.0	1.6	1.2	0	59.0	23.1	5.5	1
–57.9	0.7	0.8	0	49.6	23.8	1.9	1
–8.8	–9.1	0.9	0	12.5	7.0	1.8	1
–64.7	–4.0	0.1	0	37.3	34.1	1.5	1
–11.4	4.8	0.9	0	35.3	4.2	0.9	1

5. Saat otot tubuh berkontraksi, sinyal elektromiografi (EMG) dihasilkan di permukaan kulit (Tabel 4.19). Sinyal EMG dapat digunakan untuk mengontrol komputer, sebagai semacam antarmuka pengguna. Tujuannya adalah untuk merancang perangkat yang dapat mendeteksi sinyal EMG yang ditandai dengan tiga parameter fitur: frekuensi, kekuatan dan waktu. Anda dapat mengembangkan program komputer dan menggunakan pemodelan pohon keputusan untuk mengekstrak seperangkat aturan untuk melakukan klasifikasi berbasis aturan dari tindakan otot atau gerakan yang diperlukan untuk mengontrol bagian dari operasi komputer, seperti menghidupkan dan mematikan, operasi keyboard atau mouse , dll.

Tabel 4.19 Data eksperimental EMG dan klasifikasi tindakan yang sesuai.

Frekuensi (F)	Kekuatan (S)	Waktu (T)	Tindakan (A)
1	810	1	A1
1	864	0.5	A2
1	485	1	A3
1	950	0.5	A2
1	1003	0.5	A2
1	524	1	A3
1	736	0.5	A4
1	661	0.5	A4
2			A5
*		*	

6. Kartu kredit memungkinkan pemegang kartu untuk meminjam uang dari bank penerbit kartu, memfasilitasi penarikan tunai di muka atau pembayaran barang. Bank mengharapkan pelanggan untuk membayar kembali dengan batas waktu tertentu setiap bulan. Bank mempelajari statistik, apakah kebiasaan membayar kembali uang pinjaman pada kartu kredit tepat waktu atau tidak. Ada tiga ciri, yaitu Gender, Umur dan Pendapatan, yang menyebabkan penilaian seperti itu, seperti yang ditunjukkan pada Tabel 4.20. Gunakan keputusan atau hutan acak untuk memprediksi peringkat kredit adat. Mengingat informasi pelanggan sebagai berikut: Gender: perempuan, Usia: 26–40, dan Pendapatan: Tingkat menengah, perkirakan apakah dia akan membayar kembali uang pinjaman dengan kartu kredit tepat waktu.

Tabel 4.20 Data pemegang kartu kredit dari larangan penerbitan kartu.

ID Pemegang Kartu	Jenis Kelamin	Usia (A)	Pendapatan (I)	Bayar Kembali Tepat Waktu? (P)
1	Pria	>40	Tinggi	Ya
2	Perempuan	26∼40	Tinggi	Ya
3	Pria	<15	Rendah	Tidak
4	Perempuan	15∼25	Rendah	Tidak
5	Pria	15∼25	Tengah	Ya
6	Perempuan	15∼25	Tengah	Ya
7	Pria	26∼40	Tinggi	Ya
8	Perempuan	26∼40	Rendah	Tidak
9	Pria	26∼40	Rendah	Ya
10	Perempuan	<15	Tengah	Tidak

7. Untuk kondisi cuaca khusus, orang akan memutuskan apakah pergi bermain tenis sesuai Tabel 4.21. Di sini, kami memberikan dataset 14 hari. Tentukan semua jalur yang sesuai pada pohon keputusan yang mengarah ke keputusan permainan.

Tabel 4.21 Kondisi cuaca pada dua minggu pengamatan.

Hari	Pandangan	Kelembaban	Berangin	Bermain
1	Cerah	Tinggi	Lemah	Tidak
2	Cerah	Tinggi	Kuat	Tidak
3	Mendung	Tinggi	Lemah	Ya
4	Hujan	Tinggi	Lemah	Ya
5	Hujan	Normal	Lemah	Ya
6	Hujan	Normal	Kuat	Tidak
7	Mendung	Normal	Kuat	Ya
8	Cerah	Tinggi	Lemah	Tidak
9	Cerah	Normal	Lemah	Ya
10	Hujan	Normal	Lemah	Ya
11	Cerah	Normal	Kuat	Ya
12	Mendung	Tinggi	Kuat	Ya
13	Mendung	Normal	Lemah	Ya
14	Hujan	Tinggi	Kuat	Tidak

8. Kita semua tahu keuntungan melakukan olahraga teratur. Namun kondisi cuaca mungkin menghalangi kita untuk berolahraga. Tabel 4.22 mencatat catatan latihan Cindy selama dua minggu. Gunakan jaringan Bayesian untuk memperkirakan probabilitas bahwa Cindy harus pergi bermain tenis, dengan asumsi pasangannya ada dan hari itu cerah.

Tabel 4.22 Catatan olahraga luar ruangan Cindy dalam dua minggu.

Hari	Cuaca	Bermain	Hari	Cuaca	Bermain
1	Cerah	Tidak	8	Hujan	Tidak
2	Mendung	Ya	9	Cerah	Ya
3	Hujan	Ya	10	Hujan	Ya
4	Cerah	Ya	11	Cerah	Tidak
5	Cerah	Ya	12	Mendung	Ya
6	Mendung	Ya	13	Mendung	Ya
7	Hujan	Tidak	14	Hujan	Tidak

BAB 5
ALGORITMA *MACHINE LEARNING* TANPA PENGAWASAN

5.1 PENDAHULUAN DAN ANALISIS ASOSIASI

Dalam bab ini, kami memperkenalkan beberapa kelas utama dari algoritma *Machine learning* tanpa pengawasan. Pertama, kami mempelajari metode analisis asosiasi data untuk *Machine learning* tanpa pengawasan. Metode *Machine learning* tanpa pengawasan dan semi-diawasi lainnya disajikan di bagian selanjutnya. Metrik kinerja *Machine learning* dan metode pemilihan untuk berbagai algoritme ML diberikan di Bagian 5.4.

Pengantar *Machine learning* Tanpa Pengawasan

Dalam pembelajaran tanpa pengawasan, pelajar harus mengerjakan fungsi untuk menggambarkan struktur tersembunyi dari data yang tidak berlabel. Karena contoh yang diberikan tidak berlabel, tidak ada kesalahan atau sinyal penghargaan untuk mengevaluasi solusi potensial. Ini membedakan pembelajaran tanpa pengawasan dari pembelajaran terawasi dan pembelajaran penguatan. Pendekatan tanpa pengawasan terkait dengan dua kemampuan mendasar: i) estimasi kepadatan dalam statistik data masukan; dan ii) kemampuan untuk meringkas dan menjelaskan fitur data utama.

Machine learning tanpa pengawasan menuntut lebih banyak keterampilan dalam penambangan data, penataan data, pra-pemrosesan data, ekstraksi fitur, dan pengenalan pola. Tanpa bantuan sampel berlabel, pengguna harus memilah-milah data yang tidak terstruktur untuk mendapatkan asosiasi di antara item data, mengelompokkan data dengan kesamaan, mengurangi dimensi ruang fitur, atau mengubah format representasi untuk memungkinkan visualisasi, dll. Banyak metode *Machine learning* yang diterapkan dalam pembelajaran tanpa pengawasan didasarkan pada metode penambangan data untuk memproses data yang tidak berlabel terlebih dahulu. Pendekatan utama untuk pembelajaran tanpa pengawasan meliputi kelas-kelas berikut:

- **Analisis Asosiasi:** Prinsip apriori dan aturan asosiasi dipelajari dalam Bagian 5.1.2 hingga 5.1.4;
- **Pengelompokan Data:** Pengelompokan linier, pengelompokan logistik, k-means, KNN (K tetangga terdekat), hierarki dan metode pengelompokan berbasis kepadatan dipelajari di Bagian 5.2;
- **Pengurangan Dimensi:** Pengurangan dimensi, analisis komponen utama (PCA) dan dekomposisi nilai tunggal (SVD) dipelajari di Bagian 5.2.

Ada juga model jaringan saraf tiruan (JST), peta pengorganisasian mandiri (SOM), dan teori resonansi adaptif (ART) untuk *Machine learning* tanpa pengawasan. Kita akan mempelajari ANN dan ekstensinya untuk *Deep learning* di Bab 8. SOM adalah organisasi topografi di mana lokasi terdekat di peta mewakili input dengan properti serupa. ART memungkinkan pengguna

untuk mengontrol tingkat kesamaan antara anggota cluster yang sama. SOM dan ART tidak akan dibahas dalam buku ini. Beberapa metode semi-diawasi akan dibahas dalam Bagian 5.3.3, termasuk pembelajaran penguatan dan representasi.

Analisis Asosiasi dan Prinsip Apriori

Analisis asosiasi, juga dikenal sebagai penambangan asosiasi, mengacu pada penemuan pola, asosiasi, korelasi, atau struktur kausal yang sering muncul dalam kumpulan proyek dan kumpulan objek dalam data transaksi, data relasional, atau pembawa informasi lainnya. Dalam istilah awam, analisis asosiasi adalah cara untuk mengetahui asosiasi menarik yang tersembunyi dalam kumpulan data yang besar. Tautan yang ditemukan dinyatakan dalam aturan asosiasi: X→ Y , di mana X dan Y adalah objek atau pola data. Aturan tersebut menunjukkan hubungan yang kuat antara X dan Y. Aplikasi paling umum dari analisis asosiasi adalah menghubungkan data keranjang belanja dengan pelanggan atau ID transaksi. Tabel 5.1 menunjukkan hubungan seperti itu.

Tabel 5.1 Asosiasi yang menghubungkan data keranjang belanja dengan pembeli.

ID pembeli	kumpulan barang
1	{susu, bir, popok}
2	{cola, bir, roti, popok}
3	{roti, susu, popok, cola}
4	{makanan bayi, bir, popok, susu}
5	{apel, air, telur, popok}

Melalui observasi, kita dapat melihat bahwa beberapa pembeli yang memesan popok juga membeli susu. Itu menunjukkan ada hubungan yang kuat antara popok dan penjualan susu. Anda menggunakan aturan: {popok → susu}. Untuk data keranjang belanja pada Tabel 5.1, kita simpulkan bahwa setiap baris sesuai dengan transaksi, sesuai dengan pesanan I = {i1, i2, , id} adalah kumpulan semua item dalam data keranjang belanja, dan T = { t1, t2, , tN } adalah kumpulan dari semua transaksi. Kami akan menggunakan notasi ini untuk memperkenalkan konsep aturan asosiasi.

Himpunan item yang termasuk dalam setiap transaksi ti adalah subset dari I. Dalam analisis asosiasi, satu set item didefinisikan sebagai kumpulan dari 0 item atau lebih. Jika suatu himpunan item berisi k item, maka disebut himpunan k-item. Misalnya, {cola, bir, roti, popok} terdiri dari 4 item. Atribut penting dari itemset adalah jumlah dukungannya yang didefinisikan sebagai jumlah transaksi yang berisi kumpulan item tertentu. Secara matematis, jumlah dukungan untuk itemset X dinyatakan sebagai

$$\sigma(X) = \left|\{t_i | X \subseteq t_i, t_i \in T\}\right| \qquad (5.1)$$

Dalam Persamaan (5.1), notasi $|\cdot|$ mewakili kardinalitas suatu himpunan. Misalnya, dalam hal itemset {susu, bir, popok}, jumlah dukungan {susu, bir, popok} adalah 2. Karena hanya

dua transaksi yang berisi tiga item, transaksi tersebut terkait dengan BuyerID 1 dan BuyerID 4, seperti yang ditunjukkan pada Tabel 5.1. Untuk memvisualisasikan aturan asosiasi, notasi dan asumsi berikut digunakan:

$$X \rightarrow Y,\ X \cap Y = \emptyset \tag{5.2}$$

Kami mendefinisikan tingkat dukungan s(·) dan tingkat kepercayaan c(·) untuk mewakili kekuatan aturan asosiasi, di mana tingkat dukungan s(·) diwakili oleh frekuensi fakta bahwa aturan tersebut tercermin dalam dataset , dan tingkat kepercayaan c(·) berarti frekuensi fakta bahwa Y muncul dalam transaksi yang mengandung X. Ekspresi matematika adalah:

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$
$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{5.3}$$

di mana N adalah jumlah total transaksi. Jelas, semakin besar tingkat dukungan dan tingkat kepercayaan, semakin besar intensitas aturan asosiasi.

Tujuan dari analisis asosiasi adalah untuk menemukan aturan asosiasi yang derajat dukungan dan tingkat kepercayaannya relatif besar dalam suatu himpunan transaksi tertentu. Proses ini didefinisikan sebagai penemuan aturan asosiasi. Rumus matematikanya didefinisikan sebagai:

$$\left\{ X \rightarrow Y \mid \begin{cases} s(X \rightarrow Y) \geq minsup \\ c(X \rightarrow Y) \geq minconf \end{cases} \right\} \tag{5.4}$$

di antaranya, X → Y adalah aturan, minsup adalah nilai ambang tingkat dukungan, dan minconf adalah nilai ambang tingkat kepercayaan. Yaitu, tujuannya adalah untuk menemukan semua aturan asosiasi di mana tingkat dukungan lebih besar dari atau sama dengan minsup dan tingkat kepercayaan lebih besar dari atau sama dengan minconf.

Bagaimana menemukan aturan asosiasi adalah masalah utama. Biasanya merupakan metode sederhana untuk menghitung semua aturan yang mungkin, tetapi tidak praktis untuk himpunan dengan sejumlah besar transaksi. Untuk kumpulan data yang berisi d item, jumlah total aturan asosiasi yang mungkin ada sama dengan $R = 3^d - 2^{d+1} + 1$. Misalnya, kumpulan data yang berisi 7 item memiliki 1932 aturan asosiasi. Oleh karena itu, untuk mengetahui aturan asosiasi dengan lebih baik, kami memperkenalkan konsep frequent itemset dan aturan kuat.

Untuk himpunan item X dan subsetnya Xi, jumlah aturannya adalah (X) (Xi), karena suatu transaksi yang berisi himpunan item tertentu juga harus mengandung himpunan bagian dari himpunan ini. Frequent itemset didefinisikan sebagai sekumpulan item yang memenuhi ambang batas minimum support count, yaitu semua subsetnya adalah frequent set item. Aturan kuat didefinisikan sebagai aturan asosiasi yang tingkat kepercayaannya tinggi pada item yang sering. Jadi, dua subtugas utama yang ditemukan dalam aturan asosiasi adalah: i) untuk mengetahui

semua frequent itemset, yang dikenal sebagai generasi frequent set; dan ii) untuk mengetahui semua aturan yang kuat, yang dikenal sebagai generasi aturan yang kuat.

Hal pertama yang harus dilakukan adalah menemukan aturan asosiasi. Kumpulan item yang sering menuntut suatu algoritma untuk menentukan prosedur pembangkitannya. Dengan definisi frequent itemset, kita melihat bahwa jika sebuah itemset adalah frequent, maka semua subset darinya juga harus frequent. Ini dikenal sebagai prinsip apriori. Sebaliknya, jika sebuah itemset jarang, maka semua supersetnya juga jarang. Menggunakan prinsip ini, strategi didasarkan pada jumlah dukungan untuk memangkas set eksponensial. Teknik ini memanfaatkan sifat anti-monoton dari tingkat dukungan, yang berarti bahwa tingkat dukungan dari sebuah itemset tidak pernah bisa melebihi subsetnya.

Algoritma 5.1 Generasi A priori Frequent Itemset

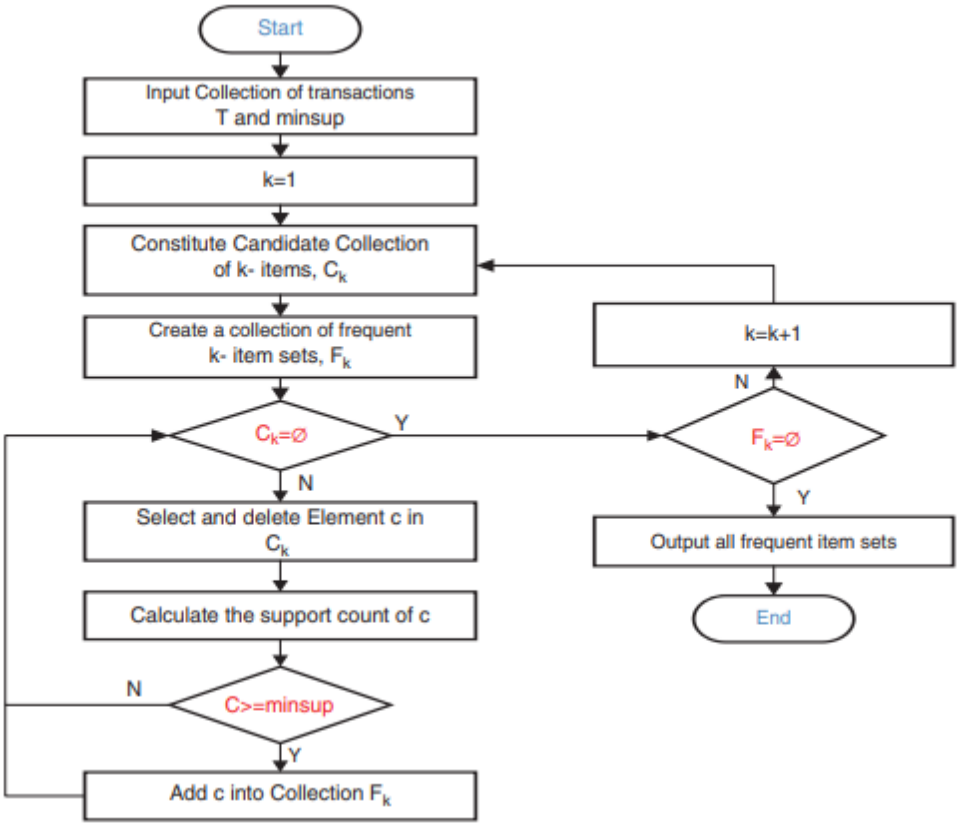
Input: T: Dataset yang berisi transaksi

minsup: Ambang tingkat dukungan

Output: Semua set item yang sering digunakan

Prosedur:

1) Misalkan k = 1
2) Sementara
3) Temukan semua k-itemsets, yang merupakan kumpulan kandidat dari k-itemsets, C_k, buat kumpulan frequent k-itemsets, Fk
4) untuk Setiap kandidat itemset c ϵ C_k
5) Buatlah derajat dukungannya σ(c) = 0
6) untuk setiap transaksi t ϵ T
7) jika Transaksi t mencakup semua item dalam c
8) σ (c) = σ (c) + 1
9) berakhir jika
10) berakhir untuk
11) jika (c) $\geq$ menit
12) Tambahkan c ke Koleksi F_k
13) berakhir jika
14) berakhir untuk
15) k = k + 1
16) sampai $F_k \neq \emptyset$
17) Keluaran F = $\cup F_k$



Gambar 5.1 Algoritma 5.1 diilustrasikan dengan diagram alir dengan lebih detail.

Menurut prinsip apriori, algoritma pembangkitan frequent itemset diberikan di bawah ini. Algoritma ini menghasilkan frequent itemsets dan menambang aturan asosiasi. Ini menggunakan teknik pemangkasan berdasarkan tingkat dukungan untuk memecahkan masalah ledakan eksponensial. Langkah-langkah spesifik dari pembangkitan frequent itemset apriori secara formal ditentukan dalam Algoritma 5.1.

Algoritma 5.1 selanjutnya diilustrasikan oleh diagram alir pada Gambar 5.1. Poin kuncinya adalah bagaimana menghasilkan banyak kandidat itemset Ck. Ada tiga metode untuk memangkas itemset yang tidak perlu. Tabel 5.2 menyajikan kompleksitas (urutan besarnya) dari tiga metode pemangkasan. Hanya deskripsi singkat yang diberikan di kolom sebelah kanan. Metode-metode ini diterapkan dalam Contoh 5.1 secara numerik.

Contoh 5.1 Menggunakan Prinsip Apriori untuk Memprediksi Kenaikan Harga di Toserba

Tabel 5.3 memberikan data harga barang komersial di department store selama delapan bulan pertama tahun tertentu. Kami menggunakan nomor 1 untuk mengacu pada kenaikan harga dan 0 untuk tidak ada kenaikan. Data ini digunakan untuk menganalisis hubungan harga apakah ada keterkaitan antara sepasang barang. Menggunakan metode pada Tabel 5.2 dan langkah-langkah yang ditentukan dalam Algoritma 5.1, frequent itemsets dihasilkan di bagian akhir.

Tabel 5.2 Kompleksitas tiga metode pemangkasan itemset.

metode	Kompleksitas	Keterangan
Metode lengkap	$O\left(\sum_{k=1}^{d} kC_d^k\right) = O(d \cdot 2^{d-1})$	Batas atas untuk menghitung semua kumpulan item yang mungkin, di mana d adalah jumlah total item data yang dipertimbangkan.
$F_{k-1} \times F_1$ Metode	$O\left(\sum_k k\lvert F_{k-1}\rvert\lvert F_1\rvert\right)$	Penggunaan frequent 1-itemset untuk memperluas frequent k-1 itemset, untuk menghasilkan frequent k-itemsets, di mana adalah kardinalitas himpunan.
$F_{k-1} \times F_{k-1}$ Metode	$O\left(\sum_k k\lvert F_{k-1}\rvert\lvert F_{k-1}\rvert\right)$	Batas atas untuk menggabungkan frequent k-1 itemset untuk menghasilkan frequent k-itemsets.

Di sini, kami mengatur minsup dan minconf masing-masing menjadi 0,4 dan 0,6. Metode $F_{k-1} \times F_1$ digunakan untuk membangkitkan k-candidate itemset untuk menentukan frequent itemset. Prosesnya meliputi langkah-langkah berikut:

1) Pertama, tingkat dukungan kandidat 1-itemset dapat dihitung:

$$\sigma(A) = 6,\ \sigma(B) = 4,\ \sigma(C) = 5,\ \sigma(D) = 2,\ \sigma(E) = 3.$$

2) Menurut ambang kepercayaan 0,6, B, D dan E dapat dipangkas, dan dengan demikian mendapatkan 1-itemset {A}, {C} yang sering.
3) Menggunakan metode $F_{k-1} \times F_1$ dapat menghasilkan kandidat 2-itemset berikut: {A, C}.
4) Hitung derajat dukungan dari kandidat 2-itemset: ({A, C}) = 5.
5) Akhirnya, kita mendapatkan semua k- frequent itemsets sebagai berikut: {A}, {C}, {A, C}.

Pembuatan Aturan Asosiasi

Setelah frequent itemset dibangkitkan, masing-masing frequent itemset dan kombinasinya dapat memenuhi ambang batas derajat dukungan. Kami menjelaskan teknik pemangkasan aturan berdasarkan konsep, yang disebut derajat kepercayaan. Jika metode enumerasi digunakan, setiap k- frequent itemset dapat menghasilkan sebanyak 2^k - 2 aturan asosiasi.

Tabel 5.3 Variasi harga yang dilaporkan dari barang-barang komersial di sebuah department store.

Bulan Baik	Barang A	Barang B	Barang E	Barang D	Barang E
1	1	0	1	0	1
2	0	1	0	1	0
3	1	0	1	0	0
4	0	1	0	0	1
5	1	0	0	0	0
6	1	1	1	0	1

7	1	1	1	0	0
8	1	0	1	1	0

Misalnya, jika itemset adalah {1, 2, 3}, frequent 3-itemset, maka dapat menghasilkan enam aturan sebagai berikut:

$$\{1,2\} \rightarrow \{3\},\ \{1,3\} \rightarrow \{2\},\ \{2,3\} \rightarrow \{1\}\ ,$$
$$\{1\} \rightarrow \{2,3\},\ \{2\} \rightarrow \{1,3\},\ \{3\} \rightarrow \{1,2\} \quad (5.5)$$

Jumlah aturan yang dihasilkan oleh pendekatan ini terlalu besar, dan tidak selalu memenuhi persyaratan ambang kepercayaan. Jadi kita membutuhkan ukuran kepercayaan untuk mengurangi jumlah aturan untuk mencapai pemangkasan. Perhatikan skenario berikut, jika dua aturan X' → Y X', X → Y X memenuhi persyaratan X' X, dan tingkat kepercayaannya masing-masing adalah:

$$c(X' \rightarrow Y - X') = \frac{\sigma(Y)}{\sigma(X')}$$
$$c(X \rightarrow Y - X) = \frac{\sigma(Y)}{\sigma(X)} \quad (5.6)$$

Dari soal di atas, untuk himpunan item yang memiliki hubungan inklusif (X') (X) ada. Jadi tingkat kepercayaan dari aturan sebelumnya dalam persamaan di atas tidak boleh melebihi tingkat kepercayaan yang terakhir. Akibatnya, kita memiliki aturan asosiasi: jika aturan X → Y X tidak memenuhi ambang kepercayaan, aturan seperti X' → Y X' (X' X) tidak akan memenuhi ambang kepercayaan.

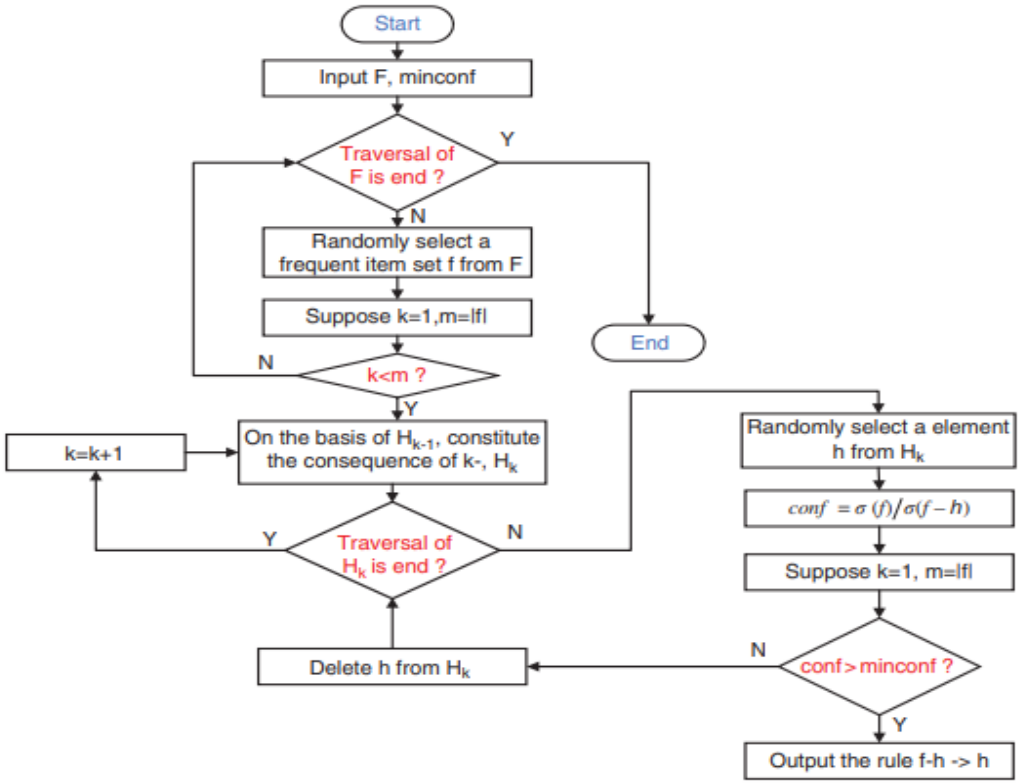
Berdasarkan teorema di atas, algoritma pembangkitan aturan Apriori diajukan. Algoritme menggunakan metode lapis demi lapis untuk menghasilkan aturan asosiasi, di mana setiap lapisan sesuai dengan jumlah item dalam aturan. Awalnya, semua aturan dengan keyakinan tinggi diekstraksi dari aturan yang hanya akan berisi satu item setelah ekstraksi. Kemudian aturan-aturan ini digunakan untuk menghasilkan aturan kandidat baru. Langkah-langkah spesifik ditentukan pada Gambar 5.2 untuk pembuatan aturan Apriori.

Algoritme pembangkitan aturan A priori tidak perlu memindai dataset lagi untuk menghitung tingkat kepercayaan aturan kandidat, karena kita dapat menggunakan jumlah dukungan dari pembangkitan frequent item untuk menentukan tingkat kepercayaan setiap aturan . Alur algoritma ditunjukkan pada Gambar 5.2.

Contoh 5.2 Pemeriksaan Fisik untuk Menghubungkan Gejala dengan Penyakit

Tabel 5.4 adalah pendataan orang yang pemeriksaan fisiknya tidak memenuhi syarat perlemakan hati, obesitas, tekanan darah tinggi, diabetes, dan batu ginjal dari Rumah Sakit Umum Tingkat II di Wuhan, di antaranya 1 menyatakan menderita penyakit tersebut dan 0 berarti tidak. Data ini digunakan untuk menganalisis apakah ada hubungan antara penyakit ini. Seringkali

ada hubungan antara penyakit yang berbeda. Penyakit yang satu dapat diturunkan dari penyakit yang lain. Hubungan yang ditemukan antara penyakit dapat membantu dokter untuk meningkatkan efisiensi diagnosis dan mengurangi tingkat kesalahan diagnosis. Dengan mengamati data pada Tabel 5.4 dan menggunakan analisis korelasi, data tersebut digunakan untuk menganalisis apakah ada hubungan antara penyakit-penyakit tersebut.



Gambar 5.2 Diagram alir untuk pembuatan aturan A priori.

Untuk mengetahui apakah ada hubungan antara sampel transaksi, pertama-tama kita perlu menentukan semua frequent itemset dalam dataset transaksi. Kemudian kita menggunakan item frequent untuk menghasilkan aturan asosiasi. Di sini, kami berasumsi bahwa ambang batas dukungan dan kepercayaan masing-masing adalah 0,4 dan 0,6. A, B, C, D, E masing-masing mengacu pada hati berlemak, obesitas, tekanan darah tinggi, diabetes dan batu ginjal. Proses dan aturan Proses pembangkitan frequent itemset dan aturan terutama sebagai berikut:

1) Gunakan metode $F_{k-1} \times F_1$ untuk membangkitkan k-candidate itemset untuk menentukan frequent itemset.

Tabel 5.4 Catatan pemeriksaan fisik dari beberapa kelompok yang tidak memenuhi syarat.

Tidak	Hati berlemak	Kegemukan	Tekanan darah tinggi	Diabetes	Batu ginjal
1	1	1	1	0	0

2	1	0	0	0	1
3	0	1	0	1	0
4	1	1	0	0	0
5	0	1	1	0	0
6	1	1	1	0	0
7	1	1	0	0	1
8	1	1	0	1	0
9	1	0	1	0	0
10	0	0	0	0	1

a) Hitung hasil dari kandidat 1-itemset sebagai berikut:

$$\sigma(a) = 7,\ \sigma(b) = 7,\ \sigma(c) = 4,\ \sigma(d) = 2,\ \sigma(e) = 3.$$

b) Menurut ambang batas dukungan 0,4, d dan e dapat dipangkas. Dapatkan 1-itemset yang sering: {a}, {b}, {c}.

c) Menggunakan metode $F_{k-1} \times F_1$, hasilkan kandidat 2-itemset berikut: {a, b}, {a, c}, {b, c}

d) Hitung derajat tumpuan 2-itemset: ({a, b}) = 5, σ ({a, c}) = 3, σ({b, c}) = 3

e) Perhatikan bahwa {a, c}, {b, c} dapat dipotong. 2-itemset yang sering dapat diturunkan {a, b}.

f) Menggunakan kembali metode $F_{k-1} \times F_1$, kandidat 3-itemset berikut dibangkitkan sebagai {a, b, c}.

g) Jumlah dukungannya adalah ({a, b, c}) = 2, jadi pangkas itemnya. Dengan demikian kita mendapatkan k- frequent itemsets sebagai berikut: {a}, {b}, {c}, {a, b}.

2) Gunakan algoritma aturan apriori untuk menghasilkan aturan asosiasi. Kita perlu memastikan bahwa jumlah item dalam frequent itemset lebih dari 2. Kemudian hitung frequent item sehingga hanya {a, b} yang memenuhi persyaratan. Untuk kumpulan item yang sering {a, b}:

a) Konsekuensi dari 1 item yang menghasilkan aturan: H_1 = {a, b}.

b) Rules yang dapat dihasilkan adalah: a → b, b → a.

c) Hitung tingkat kepercayaan aturan-aturan ini: c(a → b) = $\frac{5}{7}$, c(b → a) = $\frac{5}{7}$.

d) Oleh karena itu, aturan a → b, b → a dapat memenuhi persyaratan ambang batas kepercayaan dan dapat digunakan sebagai aturan asosiasi.

Faktanya, a dan b masing-masing mengacu pada perlemakan hati dan obesitas. Aturan asosiasi di atas menunjukkan bahwa orang yang menderita perlemakan hati adalah orang yang kelebihan berat badan. Demikian juga, orang yang kelebihan berat badan umumnya memiliki tingkat perlemakan hati tertentu. Aturan asosiasi dapat mengingatkan Anda untuk memperhatikan diet

Anda, menjalani pemeriksaan fisik secara teratur dan mencegah perlemakan hati ketika Anda kelebihan berat badan.

5.2 METODE PENGELOMPOKAN TANPA LABEL

Kita dapat menggunakan algoritma klasifikasi untuk menganalisis data berlabel, tetapi bagaimana menemukan informasi tersembunyi tentang data yang tidak berlabel dan bagaimana menemukan hubungan antar data? Metode analisis yang umum digunakan adalah metode clustering yang merupakan salah satu metode pembelajaran klasikal tanpa pengawasan. Metode pengelompokan membagi data menjadi kelompok yang berarti atau berguna (disebut klaster). Dalam hal analisis data, cluster adalah kelas potensial, sedangkan analisis clustering adalah teknik untuk menemukan kelas ini secara otomatis. Bagian ini memperkenalkan tiga metode cluster: K-means clustering, agglomerative hierarchical clustering dan density-based clustering.

Analisis Cluster untuk Prediksi dan Peramalan

Analisis klaster memberikan satu set pengamatan untuk mengelompokkan ruang data sampel ke dalam klaster. Elemen data di kelas yang sama serupa menurut beberapa metrik kesamaan yang telah ditentukan sebelumnya. Cluster dipisahkan oleh fitur atau properti yang berbeda. Metode pengelompokan lainnya didasarkan pada perkiraan kepadatan dan konektivitas grafik. Analisis klaster bertujuan untuk memisahkan data untuk tujuan klasifikasi. Ini adalah proses untuk membagi objek data ke dalam cluster. Biarkan X menjadi himpunan n objek data dan Xi label cluster. Cluster adalah himpunan bagian terpisah yang didefinisikan di bawah ini:

$$X = \bigcup_{i=1}^{n} X_i,\ X_i \cap X_j = \emptyset\ (i \neq j) \qquad (5.7)$$

Contoh 5.3 Analisis Klaster Catatan Pemeriksaan Rumah Sakit

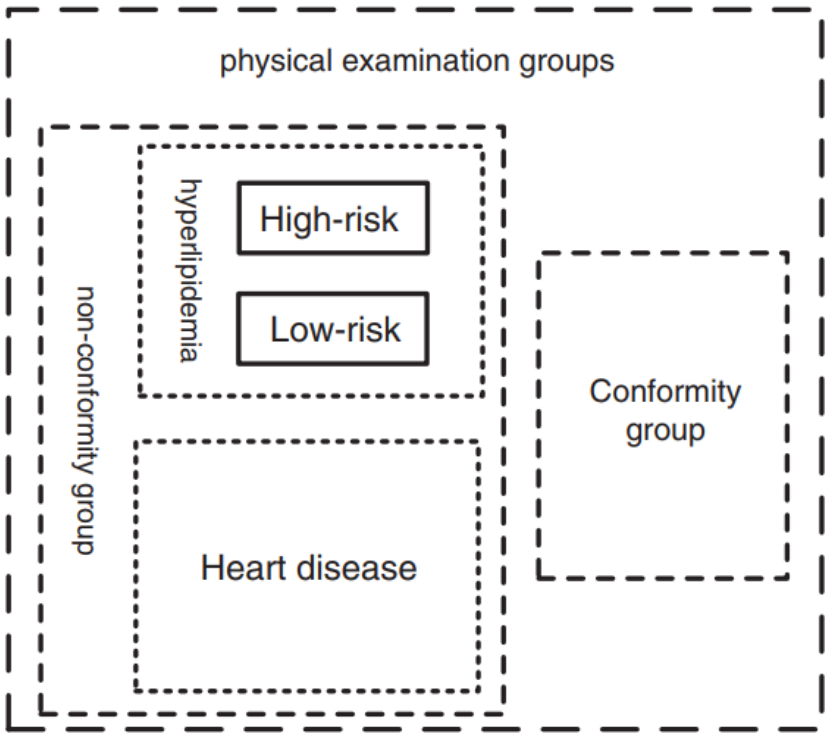
Gambar 5.3 menunjukkan contoh analisis klaster catatan pemeriksaan fisik rumah sakit. Kelompok pemeriksaan fisik dibagi menjadi kelompok sesuai dan kelompok tidak sesuai berdasarkan pengelompokan karakteristik. Ketidaksesuaian dapat dibagi menjadi subkelompok dengan hiperlipemia dan subkelompok penyakit jantung. Dengan cara yang sama, kelompok hiperlipemia dapat dibagi menjadi subkelompok risiko tinggi dan risiko rendah.

Perbedaan antara pengelompokan dan klasifikasi adalah bahwa pembagian berbasis pengelompokan tidak pasti. Dari perspektif *Machine learning*, pengelompokan adalah proses pembelajaran tanpa pengawasan dengan pencarian cluster yang konstan. Klasifikasi adalah proses pembelajaran terawasi untuk membagi objek yang ada ke dalam kelompok yang berbeda dengan berbagai label. Clustering sering membutuhkan algoritma untuk menentukan label sendiri. Lalu bagaimana cara melakukan clustering jika diberikan objek atau dataset tertentu? Ini membutuhkan desain algoritma khusus untuk pengelompokan. K-means clustering adalah metode dasar clustering.

K-means Clustering untuk Klasifikasi

Asumsikan bahwa dataset D berisi n objek dalam ruang Euclidean. Kita perlu membagi objek di D menjadi k cluster C_1, C_2, C_k, membuat

$$1 \le i,\ j \le k,\ C_i \subset D,\ C_i \cap C_j \neq \emptyset.$$



Gambar 5.3 Pengelompokan item dalam laporan pemeriksaan fisik rumah sakit.

Risiko tinggi: pasien dengan risiko tinggi hiperlipidemia
Risiko rendah: pasien dengan risiko rendah hiperlipidemia
Kelompok kesesuaian: Orang normal dalam pemeriksaan fisik

Untuk itu perlu dilakukan evaluasi kualitas divisi dengan mendefinisikan fungsi tujuan, yaitu memiliki objek yang memiliki kemiripan yang tinggi dalam suatu cluster dan antar cluster yang memiliki kemiripan yang rendah. Untuk mewujudkan cluster lebih visual, pusat pusat cluster didefinisikan untuk mewakili cluster, yang didefinisikan sebagai:

$$\bar{x}_{C_i} = \frac{\sum_{i=1}^{n_i} \vec{x}_i}{n_i},\ i = 1, 2, \cdots, k \qquad (5.8)$$

di mana n_i menunjukkan jumlah elemen dalam sebuah cluster, dan $\vec{x}_i$ menunjukkan koordinat vektor elemen cluster. Kemudian $\bar{x}_{Ci}$ menunjukkan koordinat centroid dari C_i. Gunakan d(x, y) untuk menyatakan jarak Euclidean antara dua vektor. Fungsi tujuan adalah didefinisikan sebagai:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} [d(x, \bar{x}_{C_i})]^2 \quad (5.9)$$

Nilai kualitas pembagian dengan fungsi tujuan E di atas. Faktanya, fungsi tujuan E adalah jumlah kesalahan kuadrat dari semua objek dalam kumpulan data D ke pusat massa sebuah cluster. Dengan demikian, tujuan K-means dijelaskan sebagai berikut. Untuk dataset yang diberikan dan k yang diberikan, cari grup dari cluster C_1, C_2, … C_k untuk meminimalkan fungsi tujuan E, yaitu:

$$\min E = \min \sum_{i=1}^{k} \sum_{x \in C_i} [d(x, \bar{x}_{C_i})]^2 \quad (5.10)$$

Metode K-mean Clustering ditentukan dalam Algoritma 5.3.

Algoritma 5.3 K-means Clustering

Input: k: Jumlah cluster hasil

D: Kumpulan data yang berisi n objek

Output: Set k cluster

Prosedur:

1) Pilih k objek secara acak dari D sebagai k cluster awal
2) Menurut rata-rata objek di setiap cluster, bagi objek yang tersisa ke dalam cluster terdekat
3) Hitung ulang rata-rata objek di setiap cluster
4) Sampai tidak ada perubahan lagi.

Contoh 5.4 Menggunakan K-mean Clustering untuk Mengklasifikasikan Pasien Menjadi Tiga Cluster

Hiperlipemia adalah penyakit yang umum, yang disebabkan oleh tingginya kadar lipid darah. Dengan demikian, pada pemeriksaan fisik, kandungan trigliserida dan kolesterol total dalam darah selalu digunakan untuk menentukan apakah subjek menderita hiperlipemia. Oleh karena itu, menurut dua indeks yang dibahas di atas, orang dapat dibagi menjadi dua kategori, yaitu orang normal dan penderita penyakit.

Tabel 5.5 Data pemeriksaan fisik Rumah Sakit untuk Contoh 5.4.

Nomor Seri	Trigliserida (mmol/L)	Total kolesterol (mmol/L)	Nomor Seri	Trigliserida (mmol/L)	Total kolesterol (mmol/L)
1	1.33	4.19	10	2.63	5.62
2	1.94	5.47	11	1.95	5.02
3	1.31	4.32	12	1.13	4.34
4	2.48	5.64	13	2.64	5.64

5	1.84	5.17	14	1.86	5.33
6	2.75	6.35	15	1.25	3.18
7	1.45	4.68	16	1.30	4.36
8	1.33	3.96	17	1.94	5.39
9	2.43	5.62	18	1.90	5.19

Tabel 5.5 adalah dataset kadar trigliserida dan kolesterol total pemeriksaan fisik di rumah sakit biasa. Untuk membagi orang-orang ini ke dalam kelompok yang berbeda, perlu dilakukan analisis klaster.

Menurut tabel di atas, semua dataset dapat diperoleh sebagai:

$$D = \{(1.33, 4.19),\ (1.94, 5.47),\ \cdots,\ (2.43, 5.62),\ (1.90, 5.19)\}$$

Pertama, untuk menentukan jumlah cluster, dengan asumsi kita perlu membagi menjadi tiga kelompok yang diidentifikasi dengan k = 3. Kedua, perlu untuk memilih tiga objek secara acak sebagai cluster awal dan memperoleh tiga objek dengan menggunakan nomor acak untuk membentuk cluster awal, masing-masing:

$$C_1 = \{(1.94, 5.47)\}, C_2 = \{(1.30, 4.36)\}, C_3 = \{(1.86, 5.33)\}$$

Pilih objek di D sesuai dengan jarak Euclidean terdekat dan masukkan ke dalam tiga cluster di atas. Objek e = (1.33, 4.19), misalnya jarak Euclidean objek ke ketiga cluster adalah:

$$d_{(e,C_1)} = \sum_{i=1}^{n} (x_{ei} - y_{C_1 i})^2 = (1.33 - 1.94)^2 + (4.19 - 5.47)^2 = 2.0105$$

$$d_{(e,C_2)} = \sum_{i=1}^{n} (x_{ei} - y_{C_2 i})^2 = (1.33 - 1.30)^2 + (4.19 - 4.36)^2 = 0.0298$$

$$d_{(e,C_3)} = \sum_{i=1}^{n} (x_{ei} - y_{C_3 i})^2 = (1.33 - 1.86)^2 + (4.19 - 5.33)^2 = 1.5805$$

Jika objek tersebut paling dekat dengan cluster C_2, maka objek tersebut harus dibagi menjadi cluster C_2. Dengan cara ini, kita dapat memperoleh hasil sebagai berikut:

$$C_1 = \{(2.48, 5.64),\ (2.64, 5.64),\ (2.63, 5.62),\ (2.75, 6.35),\ (2.43, 5.62),\ (1.94, 5.47),\ (1.94, 5.39)\}$$

$$C_2 = \{(1.33, 4.19),\ (1.31, 4.32),\ (1.13, 4.34),\ (1.30, 4.36),\ (1.25, 4.18),\ (1.45, 4.68),\ (1.33, 3.96)\}$$

$$C_3 = \{(1.95, 5.02),\ (1.84, 5.17),\ (1.86, 5.33),\ (1.90, 5.19)\}$$

Kemudian hitung kembali mean dari objek cluster dan dapatkan hasil sebagai berikut:

$$\bar{v}_{C_1} = \left(\frac{2.48 + 2.64 + \cdots + 1.94}{7}, \frac{5.64 + 5.64 + \cdots + 5.39}{7}\right) = (2.4014, 5.6757)$$

$$\bar{v}_{C_2} = \left(\frac{1.33 + 1.31 + \cdots + 1.33}{7}, \frac{4.19 + 4.32 + \cdots + 3.96}{7}\right) = (1.3000, 4.1471)$$

$$\bar{v}_{C_3} = \left(\frac{1.95 + 1.84 + \cdots + 1.90}{4}, \frac{5.02 + 5.17 + \cdots + 5.19}{4}\right) = (1.8875, 5.1775)$$

Karena perbedaan mean dan cluster awal, realokasi setiap objek dari dataset D dengan cara yang sama menempatkan objek divisi pertama ke dalam setiap cluster. Kita dapat memperoleh hasil sebagai berikut:

$$C_1 = \{(2.48, 5.64),\ (2.64, 5.64),\ (2.63, 5.62),\ (2.75, 6.35),\ (2.43, 5.62)\}$$

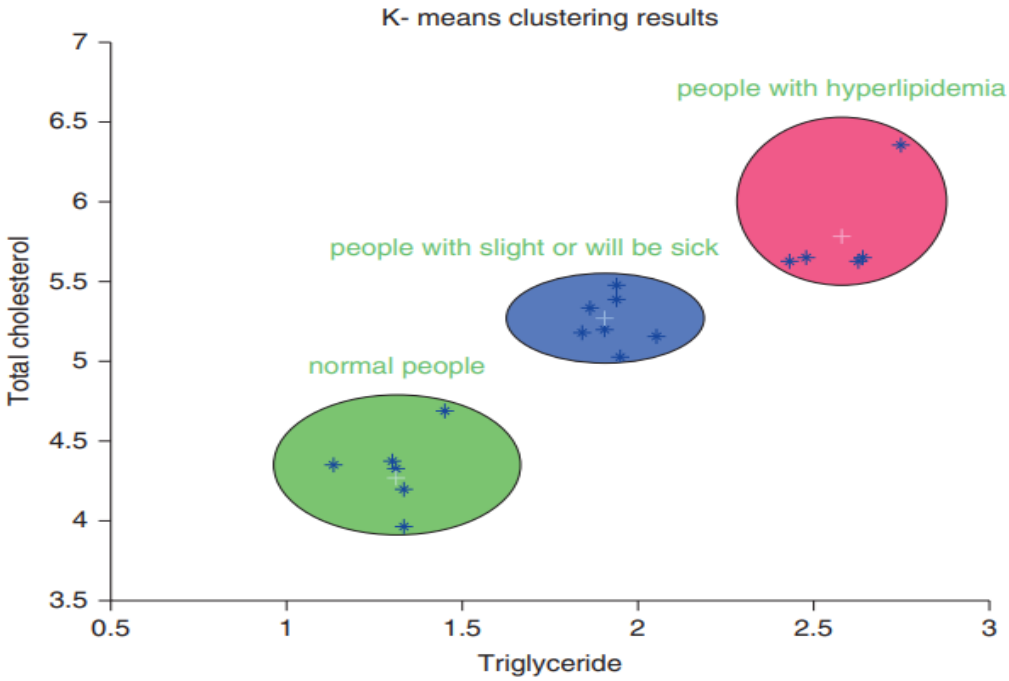
$$C_2 = \{(1.33, 4.19),\ (1.31, 4.32),\ (1.13, 4.34),\ (1.30, 4.36),\ (1.25, 4.18),\ (1.45, 4.68),\ (1.33, 3.96)\}$$

$$C_3 = \{(1.95, 5.02),\ (1.84, 5.17),\ (1.86, 5.33),\ (1.90, 5.19),\ (1.94, 5.47),\ (1.94, 5.39)\}$$

Nilai mean dan mean yang direlokasi bisa sangat dekat untuk menghentikan proses. Ini berakhir dengan klasifikasi akhir, yang hasil klasifikasinya ditunjukkan pada Gambar 5.4. Dengan demikian, orang-orang yang diperiksa secara fisik di rumah sakit ini dapat dikelompokkan menjadi tiga kategori, yaitu orang normal, orang dengan gejala ringan atau berpotensi sakit, dan orang dengan hiperlipemia. Rumah sakit dapat memberikan rekomendasi yang berbeda kepada kelompok yang berbeda untuk perawatan khusus.

Pengelompokan Hirarki Agglomerative

Pengelompokan hierarki dan pengelompokan K-means adalah dua cara pengelompokan tradisional tetapi dengan titik awal yang berbeda. K-means clustering didasarkan pada jumlah cluster yang diberikan dan mengumpulkan objek data mentah ke setiap cluster untuk hasil akhir clustering. Pengelompokan hierarkis tidak memerlukan sejumlah kategori tertentu. Itu dimulai dengan setiap objek dan secara bertahap mengumpulkan setiap objek berdasarkan matriks yang berdekatan dari objek tersebut sampai semua objek dimiliki oleh suatu kelas (atau dari keseluruhan, secara bertahap memisahkan setiap objek hingga setiap objek berada dalam kategori yang sama).



Gambar 5.4 K-means pengelompokan pasien menjadi tiga kelompok perlakuan.

Oleh karena itu, pengelompokan hierarki mencakup dua kategori:

1) Agglomerative hierarchical clustering: Mulailah dengan objek individual terhadap sebuah cluster, gabungkan dua objek atau cluster terdekat, hingga semua objek berada dalam satu cluster (yaitu, semua kumpulan data).
2) Pembagian hierarkis pengelompokan: Mulai dari cluster yang berisi semua titik (yaitu kumpulan semua data), membagi cluster dari setiap divisi, dan mendapatkan dua cluster yang terjauh satu sama lain, hingga tidak dapat dibagi (yaitu, hanya tersisa satu titik cluster).

Pengelompokan hierarki aglomeratif membutuhkan penggabungan konstan dari dua cluster yang paling berdekatan. Perlu ditentukan kedekatan setiap cluster, sehingga kriteria tertentu harus diberikan untuk pengukuran ini. Ini adalah kunci untuk pengelompokan hierarki aglomeratif, karena kriteria pengukuran yang berbeda dapat menghasilkan hasil pengelompokan yang berbeda. Ada lima definisi umum dari cara kedekatan, yaitu rantai tunggal, rantai utuh, rata-rata kelompok, metode Ward dan metode centroid. Tabel 5.6 menjelaskan definisi dan penjelasan sederhana dari kelima kedekatan tersebut.

Perhatikan, dimana Ci, Cj adalah cluster yang berbeda, xi, xj adalah objek dari cluster, ni, nj adalah jumlah objek dalam cluster, epre adalah kesalahan sebelum penggabungan, dan eafter adalah kesalahan setelah penggabungan. Objek dapat dianggap sebagai cluster titik tunggal, karena kedekatan antara objek ini dan cluster dapat dilihat sebagai kedekatan antara cluster dan cluster. Tiga metode pertama divisualisasikan pada Gambar 5.5.

Tabel 5.6 Lima rantai umum untuk menentukan kedekatan antar cluster.

definisi	Rumus	Keterangan
rantai tunggal	$\min d(x_i, x_j), x_i \in C_i x_j \in C_j$	Kedekatan cluster didefinisikan oleh MIN sebagai
Seluruh rantai	$\max d(x_i, x_j), x_i \in C_i x_j \in C_j$	jarak antara dua titik terdekat dalam cluster
Rata-rata grup	$\frac{\sum_{i=1}^{n_i}\sum_{j=1}^{n_j} d(x_i, x_j)}{n_i n_j}, x_i \in C_i x_j \in C_j$	MAX adalah jarak antara dua titik balik matahari dari
Metode lingkungan	$\min \Delta e = \min(e_{after} - e_{pre})$	dua cluster (biasanya menggunakan jarak Euclidean)
Metode pusat	$\min d(v_{C_i}, v_{C_j})$	Jarak rata-rata antara dua titik tengah cluster (biasanya menggunakan jarak Euclidean)

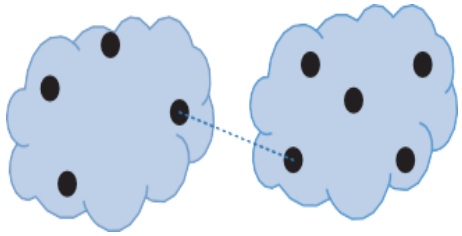
Algoritma 5.4 Generasi Peta Pohon Cluster

Input: D: Kumpulan data berisi n objek

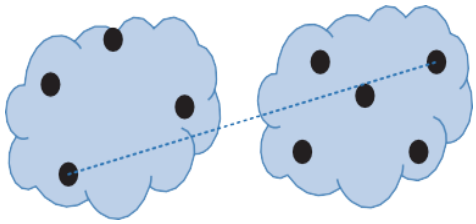
Output: Peta pohon hasil pengelompokan

Prosedur:

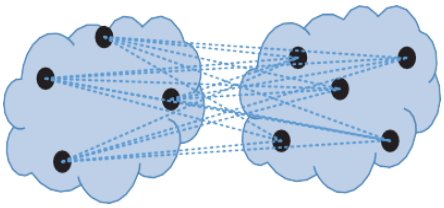
1) Mengambil setiap objek sebagai cluster, menghitung kedekatan antar cluster, mendapatkan matriks kedekatan; ketika
2) Menurut kedekatan cluster, gabungkan dua cluster terdekat
3) Hitung ulang matriks kedekatan cluster
4) Hingga hanya tersisa satu cluster.



(a) Rantai tunggal (MIN)



(b) Seluruh rantai (MAX)



(c) Rata-rata kelompok

Gambar 5.5 Representasi kedekatan cluster.

Tabel 5.7 Bagian dari data pemeriksaan fisik rumah sakit.

Nomor Seri	tinggi (cm)	Berat (kg)	Detak jantung (kali/menit)
1	154	45.5	59
2	165	65.4	108
3	166.5	76.2	58
4	166.5	74.7	54
5	161	55.6	45
6	165.5	62.3	58

Contoh 5.5 Analisis Pengelompokan Data Check-up Rumah Sakit

Tabel 5.7 memberikan dataset berat, tinggi dan detak jantung dari beberapa orang yang tidak memenuhi syarat dalam pemeriksaan fisik. Kita perlu melakukan analisis clustering untuk mereka.

Seperti yang ditunjukkan pada Tabel 5.7, semua dataset dapat diperoleh sebagai:

$$D = \{(154, 45.5, 59),\ (165, 65.4, 108),\ \cdots,\ (165.5, 62.3, 58)\}$$

Karena satuan data barang pemeriksaan fisik tidak menyatu, maka dilakukan standarisasi untuk menghilangkan pengaruh terhadap hasil satuan tersebut. Hasil standar adalah:

$$data\_st = \begin{pmatrix} 0.92 & 0.60 & 0.55 \\ 0.99 & 0.86 & 1 \\ 1 & 1 & 0.54 \\ 1 & 0.98 & 0.50 \\ 0.97 & 0.73 & 0.42 \\ 0.99 & 0.82 & 0.54 \end{pmatrix}$$

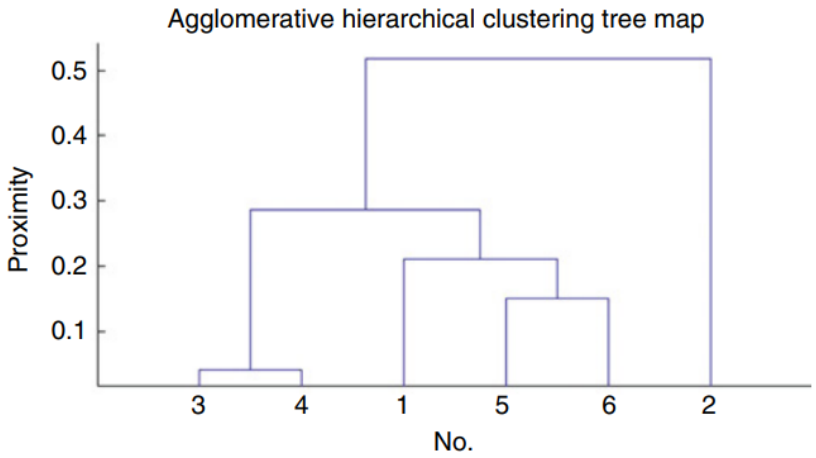
Menggunakan jarak Euclidean sebagai kedekatan cluster, matriks kedekatan dapat dihitung sebagai berikut:

$$dist = \begin{pmatrix} 0 & 0.53 & 0.41 & 0.39 & 0.19 & 0.23 \\ 0.53 & 0 & 0.48 & 0.51 & 0.60 & 0.46 \\ 0.41 & 0.48 & 0 & 0.04 & 0.30 & 0.18 \\ 0.39 & 0.51 & 0.04 & 0 & 0.27 & 0.17 \\ 0.19 & 0.60 & 0.30 & 0.27 & 0 & 0.15 \\ 0.23 & 0.46 & 0.18 & 0.17 & 0.15 & 0 \end{pmatrix}$$

Terlihat bahwa kedekatan cluster 3 dan cluster 4 minimal 0,04. Kemudian gabungkan kedua cluster ini, hitung kembali kedekatan setiap cluster dan adopsi metode rantai tunggal (metode MIN) sebagai definisi kedekatan. Ini menghasilkan:

$$cluster: \quad 1 \quad 2 \quad 3,4 \quad 5 \quad 6$$

$$dist = \begin{pmatrix} 0 & 0.53 & 0.39 & 0.19 & 0.23 \\ 0.53 & 0 & 0.48 & 0.60 & 0.46 \\ 0.39 & 0.48 & 0 & 0.27 & 0.17 \\ 0.19 & 0.60 & 0.27 & 0 & 0.15 \\ 0.23 & 0.46 & 0.17 & 0.15 & 0 \end{pmatrix}$$



Gambar 5.6 Peta pohon untuk pengelompokan hierarki aglomeratif.

Terlihat bahwa kedekatan cluster 5 dan cluster 6 minimal 0,15. Kemudian gabungkan kedua cluster tersebut, hitung kembali kedekatan setiap cluster dan ulangi hingga hanya ada satu cluster. Urutan penggabungan masing-masing adalah 3, 4 → 5, 6 → 1, {5, 6} →{3, 4}, {1, {5, 6}} → {{3, 4}, {1, {5, 6}}}, 2. Pengelompokan hierarki aglomeratif hasil peta pohon ditunjukkan pada Gambar 5.6.

Dari hasil pemeriksaan fisik orang No 2 memiliki perbedaan dengan yang lain. Dari data tabel, kami menemukan bahwa detak jantung No. 2 cepat, sementara yang lain terlalu lambat. Dari Gambar 5.6, terlihat bahwa orang No. 3 dan No. 4 yang diperiksa secara fisik sangat mirip (berat badan hampir sama dan tinggi badan keduanya relatif sama), mengelompok menjadi satu kelompok.

Pengelompokan Berbasis Kepadatan

K-means clustering dan agglomerative hierarchical clustering biasanya hanya dapat menemukan cluster globular, tetapi tidak cluster yang berbentuk sembarang, seperti cluster seperti cincin, dll. Namun dalam kehidupan nyata, ada berbagai macam bentuk yang tidak spherical, tetapi berbentuk S atau seperti cincin, dll. Akibatnya, sulit untuk k-means atau pengelompokan hierarkis untuk memenuhi persyaratan praktis, terutama ketika melibatkan klasifikasi outlier kebisingan yang terakhir, yang biasanya di interior ring atau tetap berada di luar kelompok.

Untuk mencari outlier semacam itu, perlu untuk membuat cluster dengan bentuk yang sewenang-wenang. Salah satu sudut pandangnya adalah membagi data spasial ke dalam berbagai jenis zona sesuai dengan kepadatan data. Setiap zona sesuai dengan cluster tertentu, mengisolasi outlier. Pengelompokan berbasis kepadatan semacam itu adalah salah satu cara untuk menyelesaikan masalah outlier. Bagian ini terutama membahas satu jenis pengelompokan spasial berbasis kepadatan aplikasi dengan kebisingan (DBSCAN).

Tabel 5.8 Tiga jenis titik kepadatan data.

Jenis	Rumus	Keterangan
Titik inti	$\begin{cases} card(\{x : d(x,a) \leq Eps\}) \geq MinPts \\ x \in D \end{cases}$	card() adalah jumlah elemen himpunan yang diperoleh, d(x,a) adalah fungsi jarak, a adalah titik inti, Eps adalah parameter jarak dan MinPts adalah nilai ambang yang diterapkan.
Titik perbatasan	$\begin{cases} card(\{x : d(x,b) \leq Eps\}) < MinPts \\ x \in D,\ b \in A \end{cases}$	b adalah titik perbatasan, Eps adalah jarak, MinPts adalah nilai ambang dari jumlah titik internal dan A mengacu pada kumpulan tetangga dari titik inti.
Titik kebisingan	$\begin{cases} card(\{x : d(x,c) \leq Eps\}) < MinPts \\ x \in D,\ c \notin A \end{cases}$	d() adalah jumlah elemen himpunan yang diperoleh, d(x,c) mengacu pada fungsi jarak, c adalah titik noise, Eps adalah jarak, MinPts adalah ambang batas dan A adalah kumpulan titik inti

Titik-titik dalam ruang data dapat diklasifikasikan ke dalam tiga jenis berikut sesuai dengan derajat intensif: Pengelompokan berbasis kepadatan tercantum pada Tabel 5.8 dan ditentukan oleh Algoritma 5.5:

1) **Titik inti:** adalah titik di dalam daerah padat. Tetangganya ditentukan oleh fungsi jarak (biasanya digunakan jarak Euclidean), parameter jarak yang ditentukan oleh pengguna dan nilai ambang dari jumlah titik internal. Jika titik ini adalah titik inti, maka jumlah titik di bidang yang ditentukan akan melampaui nilai ambang batas yang diberikan.

2) **Titik perbatasan:** adalah titik di tepi daerah padat. Jumlah titik dalam lingkungan titik ini kurang dari nilai ambang batas jumlah titik internal yang ditentukan oleh pengguna, tetapi titik ini terletak di interior lingkungan satu titik inti tertentu.
3) **Titik kebisingan:** adalah titik di daerah yang jarang. Jumlah titik dalam lingkungan titik ini kurang dari nilai ambang batas jumlah titik internal yang ditentukan oleh pengguna. Tetapi titik ini tidak terletak di interior lingkungan titik inti mana pun.

Algoritma 5.5 Menggunakan Pengelompokan Objek Data Berbasis Kepadatan

Input: D: kumpulan data, yang berisi n subjek

Eps: parameter jarak. *MinPts:* nilai ambang batas kepadatan lingkungan

Output: tipe: hasil analisis clustering berbasis densitas, yaitu tipe objek data yang relevan

Prosedur:

1) Secara acak pilih objek p yang belum diberi label, tandai objek ini
2) Hitung kerapatan lingkungan Pts dari objek ini
3) jika Poin MinPts
4) Siapkan cluster C baru, dan tambahkan p ke C
5) Temukan semua benda bertetangga dari benda ini, membentuk himpunan N
6) untuk setiap objek P' dalam N
7) Jika p' belum diberi label, tandai objek ini
8) Jika kerapatan lingkungan p' lebih besar dari MinPts
9) Temukan semua objek tetangga dari objek ini dan tambahkan ke N
10) tandai p' sebagai titik perbatasan
11) Jika p' bukan milik salah satu anggota cluster, tambahkan p' ke C
12) akhir
13) Tandai objek dalam cluster C sebagai titik inti, output C
14) jika tidak, tandai p sebagai titik kebisingan
15) akhir
16) Tentukan jenis, dan gunakan untuk semua objek, mewakili titik kebisingan, titik perbatasan, dan titik inti dengan -1, 0, 1
17) Sampai semua objek telah ditandai.

Tiga jenis titik ditunjukkan pada Gambar 5.7. Lingkaran adalah titik inti, persegi panjang adalah titik perbatasan, dan segitiga adalah titik kebisingan. Jumlah titik dalam beberapa titik tertentu didefinisikan sebagai kepadatan lingkungan. Jika kedua benda p dan q adalah titik inti, dan satu berada di dalam lingkungan, kedua benda berhubungan kerapatan:

$$\begin{cases} d(p,q) \leq Eps \\ p,q \in A \end{cases} \tag{5.11}$$

Ekspresi d(p,q) mengacu pada fungsi jarak, dan A mengacu pada kumpulan tetangga dari titik inti. Yang disebut kepadatan-terjangkau mengacu pada dua objek yang dihubungkan oleh serangkaian objek kepadatan-terjangkau langsung. Kedua objek tersebut dapat dilihat sebagai titik inti atau titik perbatasan. Tujuan dari DBSCAN adalah untuk mengetahui core point, frontier point dan noise point. Langkah-langkah dari suatu algoritma ditunjukkan pada Algoritma 5.5.

Contoh 5.6 Menggunakan Densitas Data untuk Melakukan Pengelompokan dalam Analisis Sel Darah

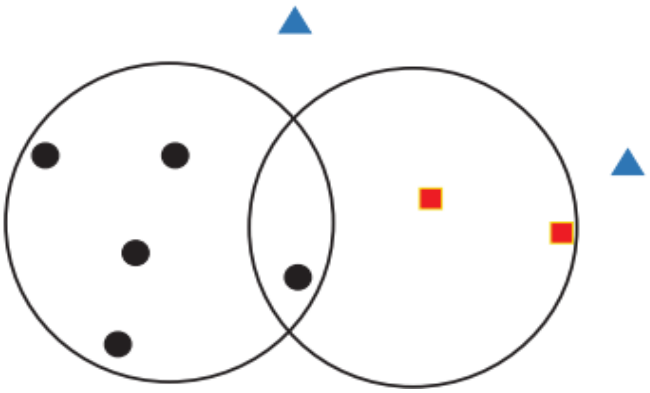
Tabel 5.9 menyajikan pengumpulan data kandungan sel darah putih dan sel darah merah dari beberapa orang yang diperiksa secara fisik di sebuah rumah sakit di Wuhan, Cina. Perlu ditegaskan bahwa orang yang abnormal di antara kelompok yang diperiksa bisa jadi adalah orang yang data pemeriksaannya salah.

Berdasarkan tabel di atas, seluruh dataset dapat diperoleh sebagai:

$$D = \{(9.4, 5.33),\ (6.0, 4.26),\ \cdots,\ (5.0, 4.44),\ (5.6, 6.78)\}$$

Nilai ambang kerapatan lingkungan dan parameter jarak (jarak Euclidean) yang ditunjukkan di sini adalah:

$$\begin{cases} MinPts = 4 \\ Eps = 0.9016 \end{cases}$$



Gambar 5.7 Titik inti, titik perbatasan dan titik kebisingan dalam dua cluster.

Tabel 5.9 Data sel darah putih dan sel darah merah untuk Contoh 5.6

sel darah putih	Sel darah merah	sel darah putih	Sel darah merah	sel darah putih	Sel darah merah
9.4	5.33	6.6	4.41	4	4.43
6	4.26	5.4	4.62	8.6	5.15
6	4.62	6	4.92	8.7	5.43
6	4.12	8.6	5.44	5.5	5
5.5	5.45	6.5	5.34	5	5.95

6	5.90	4.2	4.54	4.7	5.04
5.5	5.73	3.5	4.80	5.6	5.42
4.8	4.51	4.9	4.46	2.1	3.79
5.9	4.24	3	4.79	7.8	5.73
5	4.46	5.8	5.20	4.5	4.35
4.8	3.97	3	4.17	12.6	5.27
3.6	4.46	8.1	4.82	4.4	5.32
7.5	5.51	5.1	4.16	5.6	4.39
8.3	4.92	6.9	5.18	4.6	3.74
8.6	5.97	6.3	3.99	5	4.44
5.3	5.33	6.8	3.91	5.6	6.78
4.2	4.87	5.9	4.29		

Pilih secara acak sebuah objek p(4.8, 4.51), hitung jarak antara setiap objek ke objek ini secara iteratif. Misalnya jarak benda p ke benda pertama (9.4, 5.33) pada Tabel 5.9 adalah:

$$d(p,q) = \sqrt{(9.4 - 4.8)^2 + (5.33 - 4.51)^2} = 4.67$$

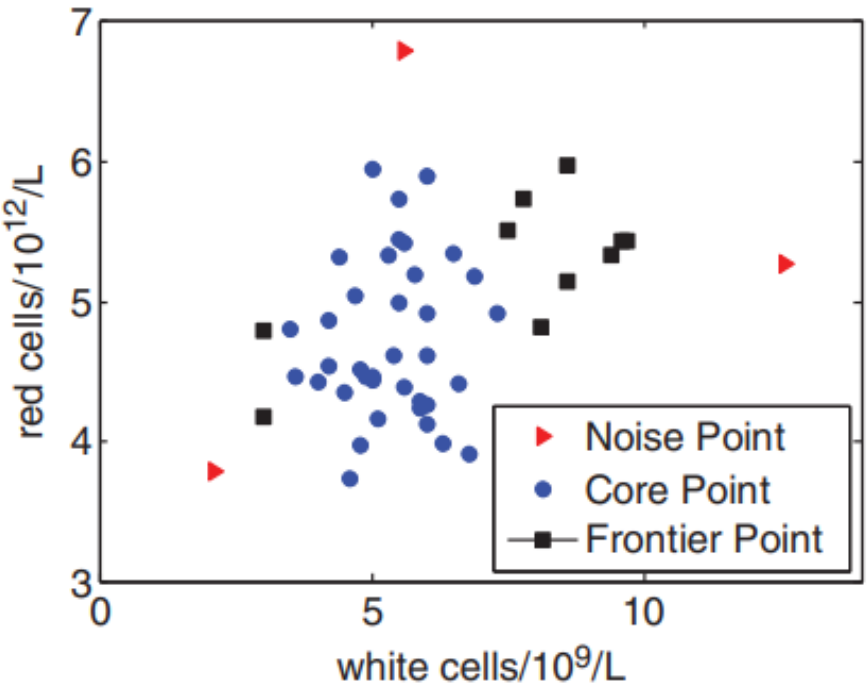
Jelas, hasilnya lebih besar dari Eps = 0.9016. Jadi objek ini tidak berada dalam lingkungan p. Sebagai contoh lain, jarak antara benda (5, 4.46) dan benda p adalah:

$$d(p,q) = \sqrt{(5 - 4.8)^2 + (4.46 - 4.51)^2} = 0.21$$

Jelas, hasilnya kurang dari Eps = 0.9016. Jadi objek ini berada dalam lingkungan p. Setelah dihitung, jenis benda dalam contoh ini berjumlah 15, jadi kerapatan lingkungan benda ini adalah $Pts_v = 15 > MinPts = 4$.

Jadi, objek p(4.8, 4.51) adalah objek inti. Ulangi prosedur untuk objek lain dalam lingkungan objek p. Misalnya, (5, 4.46), kerapatan lingkungan benda ini adalah 14, dan ia berada di lingkungan benda p. Dengan demikian objek dan objek p ini dapat dijangkau dengan kepadatan, dan harus ditandai sebagai titik inti. Jika satu objek dan titik inti tertentu dapat dijangkau oleh kepadatan dan kepadatan lingkungan tidak kurang dari MinPts = 4, maka tandai objek tersebut sebagai titik inti.

Jika satu objek dan titik inti tertentu dapat dijangkau oleh kepadatan tetapi kepadatan lingkungan kurang dari MinPts = 4, tandai objek ini sebagai titik perbatasan. Jika satu objek dan titik inti tertentu tidak dapat dijangkau oleh kepadatan, maka tandai objek ini sebagai titik kebisingan. Dengan cara ini, himpunan titik kebisingan yang diperoleh adalah {(2.1, 3.79), (5.6, 6.78), (9.7, 5.43)}. Hasil clustering berbasis kepadatan ditunjukkan pada Gambar 5.8.



Gambar 5.8 Gambar hasil analisis clustering berbasis kepadatan.

Seperti yang digambarkan pada Gambar 5.8, bentuk setelah pengelompokan tidak teratur, dan tidak sulit untuk memilih titik inti, titik perbatasan, dan titik kebisingan. Kita dapat dengan mudah mengetahui bahwa titik-titik di kiri bawah, tetapi posisi atas dan kanan tampaknya salah dalam hal konten sel darah putih dan sel darah merah. Orang yang diperiksa terkait perlu diperiksa ulang untuk memastikan hasilnya lebih lanjut.

5.3 PENGURANGAN DIMENSI DAN ALGORITMA LAINNYA

Di bawah reduksi dimensi tinggi, jarak (seperti jarak Euclidean) antara titik sangat mirip atau dua vektor ortogonal dan dengan demikian menyebabkan kesulitan dalam klasifikasi, regresi dan terutama pengelompokan. Fenomena ini disebut "kutukan dimensi". Banyak algoritma pengurangan dimensi yang diusulkan untuk memecahkan masalah.

Pengurangan dimensi mengacu pada transfer titik dalam ruang dimensi tinggi ke ruang dimensi rendah melalui fungsi pemetaan untuk menghilangkan "kutukan dimensi". Pengurangan dimensi mungkin tidak hanya mengurangi korelasi data, tetapi juga mempercepat kecepatan operasi algoritma (penurunan volume data). Bab ini akan menjelaskan ide inti dari algoritma pengurangan dimensi.

Metode Pengurangan Dimensi

Pengurangan dimensi dalam bidang *machine learning* mengacu pada pemetaan titik-titik data dalam ruang berdimensi tinggi ke ruang berdimensi rendah dengan metode pemetaan tertentu. Inti dari pengurangan dimensi adalah mempelajari fungsi pemetaan: f : x → y, di mana

x adalah ekspresi titik data asli dalam bentuk ekspresi vektor. y adalah ekspresi dari vektor berdimensi rendah setelah pemetaan titik data. Umumnya, dimensi y lebih kecil dari x. f mungkin eksplisit, implisit, linier atau nonlinier (Tabel 5.10).

Saat ini, sebagian besar algoritme reduksi dimensi memproses data ekspresi vektor, sementara beberapa algoritme reduksi dimensi memproses data ekspresi tensor orde tinggi. Alasan untuk menggunakan data reduksi dimensi adalah bahwa ada informasi yang berlebihan dan informasi kebisingan di ruang dimensi tinggi asli dan dapat menyebabkan kesalahan dan mengurangi akurasi dalam aplikasi praktis (yaitu identifikasi gambar). Melalui pengurangan dimensi, kami ingin mengurangi kesalahan yang disebabkan oleh informasi yang berlebihan dan meningkatkan akurasi pengenalan (atau aplikasi lain). Atau, kami ingin mencari karakteristik struktur penting dari data melalui algoritma pengurangan dimensi. Pengurangan dimensi linier meliputi PCA dan metode analisis diskriminan linier, dan pengurangan dimensi nonlinier diwakili oleh LLE dan metode pemetaan fitur isometrik.

Tabel 5.10 Metode pengurangan dimensi untuk *Machine learning*.

Metode	**Ide Dasar**
Analisis Komponen Utama (PCA)	Gunakan beberapa indikator agregat (komponen utama) untuk mengganti semua indikator dalam data asli
Dekomposisi Nilai Singular (SVD)	Ambil nilai tunggal dalam matriks untuk diselesaikan. Pilih nilai singular yang lebih besar dan abaikan nilai singular yang lebih kecil untuk mengurangi dimensi matriks
Analisis Faktor (FA)	Temukan tautan intrinsik setiap properti melalui analisis struktur data untuk mengetahui sifat umum (faktor)
Metode Kuadrat Terkecil Sebagian	Mengintegrasikan keunggulan PCA, metode analisis korelasi kanonik dan metode analisis regresi linier multivariat. Pengurangan dimensi dan prediksi dapat dicapai
Pemetaan Sammon	Sambil menjaga struktur jarak titik-titik, petakan data dalam ruang dimensi tinggi ke ruang dimensi rendah
Analisis Diskriminasi (DA)	Memproyeksikan data (titik) dalam ruang berdimensi tinggi dengan label kelas ke ruang berdimensi rendah, sehingga diklasifikasikan dalam ruang berdimensi rendah.
Penyematan Linier Lokal (LLE)	Sebagai semacam algoritma pengurangan dimensi nonlinier, mungkin membuat data pengurangan dimensi mempertahankan struktur manifold asli.
Peta Eigen Laplace	Kami membutuhkan titik-titik terkait (titik terhubung di peta) untuk menjadi dekat satu sama lain di ruang setelah pengurangan dimensi

Analisis Komponen Utama (PCA)

Dalam situasi sebenarnya, objek memiliki banyak komposisi properti. Misalnya, laporan pemeriksaan kesehatan tubuh manusia terdiri dari banyak item pemeriksaan fisik. Setiap properti adalah cerminan dari objek. Ada sedikit banyak korelasi antara objek-objek ini. Korelasi tersebut menyebabkan terjadinya tumpang tindih informasi. Tumpang tindih dan korelasi yang tinggi dari informasi properti (variabel atau karakteristik) dapat menimbulkan banyak kendala dalam penerapan dan analisis data metode statistik. Pengurangan dimensi properti diperlukan untuk memecahkan informasi yang tumpang tindih, yang dapat sangat mengurangi jumlah variabel yang berpartisipasi dalam pemodelan data, dan tidak akan menyebabkan kehilangan informasi. PCA adalah jenis metode analisis yang banyak digunakan untuk mereduksi dimensi variabel secara efektif.

PCA dirancang untuk mentransfer beberapa indikator (variabel dalam regresi) ke beberapa indikator agregat (komponen utama) dengan ide pengurangan dimensi. Setiap komponen utama dapat mencerminkan sebagian besar informasi dari variabel asli, dan informasi yang disertakan tidak diulang. Dalam kasus umum, setiap komponen utama adalah kombinasi linier dari variabel asli, dan setiap komponen utama tidak berhubungan. Metode ini dapat meringkas faktor-faktor kompleks menjadi beberapa komponen utama sambil memasukkan banyak variabel untuk menyederhanakan masalah dan memperoleh informasi data yang ilmiah dan efektif.

Penting untuk dicatat bahwa ada kehilangan informasi di PCA, terlepas dari penurunan dimensi properti. Hilangnya informasi ini dapat diperkuat dalam iterasi algoritma *Machine learning*, menyebabkan kesimpulan yang tidak akurat. Oleh karena itu, pertimbangan yang cermat harus diambil saat menggunakan PCA. Komponen utama adalah beberapa variabel baru yang dibentuk secara menyeluruh oleh variabel asli. Ini disebut komponen utama pertama, komponen utama kedua, dll. sesuai dengan volume informasi dalam komponen utama. Ada beberapa hubungan antara komponen utama dan variabel asli:

1) Komponen utama menyimpan sebagian besar informasi dari variabel asli.
2) Jumlah komponen utama jauh lebih sedikit daripada variabel asli.
3) Setiap komponen utama tidak berhubungan.
4) Setiap komponen utama adalah kombinasi linier dari variabel asli.

Tujuan PCA adalah untuk menggabungkan kembali variabel terkait ke sekelompok variabel komprehensif baru yang tidak terkait untuk menggantikan variabel asli. Secara umum, metode pemrosesan matematis adalah kombinasi linier dari variabel asli sebagai variabel komprehensif baru. Bagaimana memilih dari begitu banyak kombinasi? Jika kombinasi linier pertama (variabel komprehensif pertama) dilambangkan sebagai F1, kita membutuhkannya untuk mencerminkan lebih banyak informasi dari variabel asli. Dalam PCA, “informasi” diukur dengan varians, yaitu jika Var(F_1) lebih besar, ini menunjukkan bahwa F_1 mengandung lebih banyak informasi. Oleh karena itu, varians yang dipilih pada semua kombinasi linier adalah yang terbesar, sehingga F_1 disebut

komponen utama pertama. Jika komponen utama pertama tidak cukup untuk mewakili informasi variabel p, kami mempertimbangkan untuk memilih kombinasi linier kedua. Untuk mencerminkan informasi asli secara efektif, informasi di F_1 tidak akan muncul di F_2. Ekspr F_2 ession adalah Cov(F_1, F_2) = 0, dan disebut komponen utama kedua. Kita dapat membangun komponen utama ke-3, ke-4 dan ke-p dengan melakukan seperti ini.

Misalkan ada n objek evaluasi (contoh, seperti orang yang melakukan pemeriksaan fisik) dan m indikator evaluasi (seperti tinggi, berat, dll.), maka dapat menyusun matriks berukuran n × m. Dinotasikan sebagai $x = (x_{ij})_{n\times m}$, di mana x_i, i = 1, 2, , m adalah vektor kolom. Matriks tersebut disebut matriks evaluasi.

Setelah mendapatkan matriks evaluasi, langkah-langkah umum PCA ditunjukkan sebagai berikut:

1) Hitung nilai rata-rata $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ dan varians $S_j = \sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)/(n-1)}$ dari data sampel asli. Nilai rata-rata dan simpangan baku dihitung di setiap kolom.
2) Hitung data standar $X_{ij} = {(x_{ij} - \bar{x}_j)}/{S_i}$ dan matriks evaluasi berubah menjadi matriks setelah standarisasi:

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nm} \end{bmatrix} = (X_1,\ X_2,\ \cdots\ , X_m) \qquad (5.12)$$

3) Menerapkan matriks setelah standarisasi untuk menghitung matriks korelasi $C = (c_{ij})_{m\times m}$ dari setiap indikator evaluasi (atau matriks kovarians), sehingga C adalah matriks simetris dan positif. Oleh karena itu $c_{ij} = X_i^T X_j/(n-1).$
4) Hitung nilai karakteristik dan vektor karakteristik dari matriks korelasi (atau matriks kovarians). Susun nilai karakteristik dalam urutan menurun: $\lambda_1 > \lambda_2 > \cdots > \lambda_m$, , lalu susun vektor-vektor karakteristik yang sesuai dengan nilai karakteristiknya. Misalkan vektor karakteristik j adalah $\xi_j = (\xi_{1j}, \xi_{2j}, \cdots, \xi_{mj})^I$, , maka komponen utama j adalah

$$F_j = \xi_j^T X = \xi_{1j}X_1 + \xi_{2j}X_2 + \cdots + \xi_{mj}X_m \qquad (5.13)$$

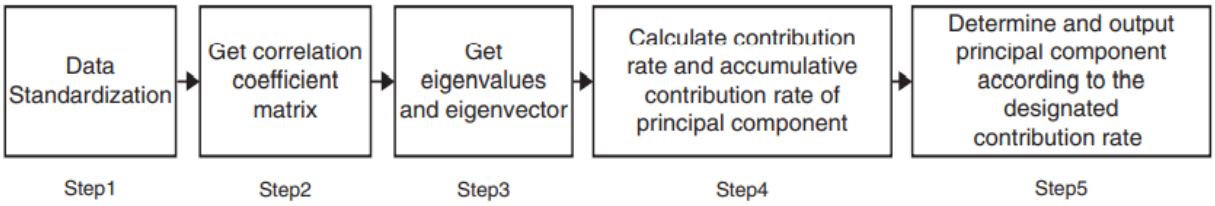
Ketika j = 1, F_1 adalah komponen utama pertama.

5) Menurut nilai karakteristik dari matriks korelasi, hitung tingkat kontribusi η dan tingkat kontribusi akumulatif Q dari komponen utama adalah

$$\eta_i = \frac{\lambda_i}{\lambda_1 + \lambda_1 + \cdots + \lambda_m},\ Q_i = \eta_1 + \eta_2 + \cdots + \eta_i,\ i = 1, 2, \ldots, m \qquad (5.14)$$

6) Terakhir, menurut tingkat kontribusi yang ditentukan oleh pengguna, tentukan jumlah komponen utama dan dapatkan komponen utama dari matriks evaluasi (Gambar 5.9). Secara umum, tingkat iuran adalah 0,85, 0,9 dan 0,95. Tiga tingkat tingkat kontribusi yang berbeda

ditentukan sesuai dengan skenario tertentu. Langkah-langkah umum PCA ditunjukkan di bawah ini:



Gambar 5.9 Langkah-langkah Analisis Komponen Utama (PCA).

Tabel 5.11 Data pasien untuk klasifikasi PCA pada Contoh 5.7.

Id	Trigliserida	Total kolesterol	Hdl-c	Ldl-c	Usia	Bobot	Protein total	Gula darah
1	1.05	3.28	1.35	1.8	60	56.8	66.8	5.6
2	1.43	5.5	1.66	3.69	68	57.4	79.4	5.3
3	1.16	3.97	1.27	2.55	68	70.7	74.7	5.4
4	6.8	5.95	0.97	2.87	50	80.1	74	5.6
5	3.06	5.25	0.9	3.81	48	82.7	72.4	5.8
6	1.18	5.88	1.77	3.87	53	63.5	78	5.2
7	2.53	6.45	1.43	4.18	57	61.3	75	7.3
8	1.6	5.3	1.27	3.74	47	64.9	73.6	5.4
9	3.02	4.95	0.95	3.53	39	88.2	79	4.6
10	2.57	6.61	1.56	4.27	60	63	80	5.6

Contoh 5.7 Analisis Komponen Utama Data Pasien

Tabel 5.11 merupakan himpunan trigliserida, kolesterol total, kolesterol high-density lipoprotein (hdl-c), kolesterol lipoprotein densitas rendah (ldl-c), umur, berat badan, total protein dan kadar gula darah pada data pemeriksaan fisik di rumah sakit kelas A kelas dua di Kota Wuhan, China. Gunakan PCA untuk menentukan komponen utama orang untuk mencapai pengurangan dimensi data.

Karena data di setiap kolom mencerminkan aspek yang berbeda dari orang yang melakukan pemeriksaan fisik dan unit indikatornya berbeda, kami membakukan data yang asli. Misalnya, indikator orang No. 1 diberikan sebagai $x'_{11} = \frac{x_{11}}{\max(x_1)} = \frac{1.05}{6.8} = 0.15,$ diwakili oleh matriks evaluasi dan matriks korelasi berikut:

$$x = \begin{bmatrix} 0.15 & 0.50 & \cdots & 0.77 \\ 0.21 & 0.83 & \cdots & 0.73 \\ \vdots & \vdots & \ddots & \vdots \\ 0.38 & 1 & \cdots & 0.77 \end{bmatrix} \qquad corr(x) = \begin{bmatrix} 1 & 0.40 & \cdots & 0.09 \\ 0.40 & 1 & \cdots & 0.35 \\ \vdots & \vdots & \ddots & \vdots \\ 0.09 & 0.35 & \cdots & 1 \end{bmatrix}$$

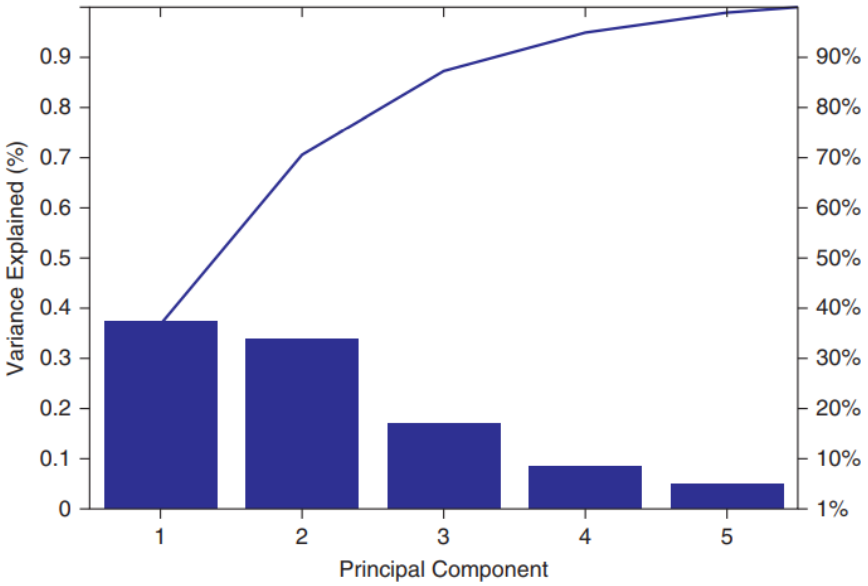
Hitung nilai karakteristik dan vektor karakteristik menurut matriks korelasi:

$$\lambda = [2.96, 2.65, 1.33, 0.62, 0.33, 0.0024, 0.07, 0.036]$$

di mana vektor karakteristik yang sesuai dengan nilai karakteristik pertama adalah:

$$\xi_1 = [00.42, 0.02, 0.53, 0.06, 0.46, -0.54, 0.07, 0.16]^T$$

Hitung tingkat kontribusi masing-masing komponen utama dan plot, seperti pada Gambar 5.10. Tingkat iuran yang ditetapkan adalah 85%. Kita dapat menghitung komponen utama sebagai:

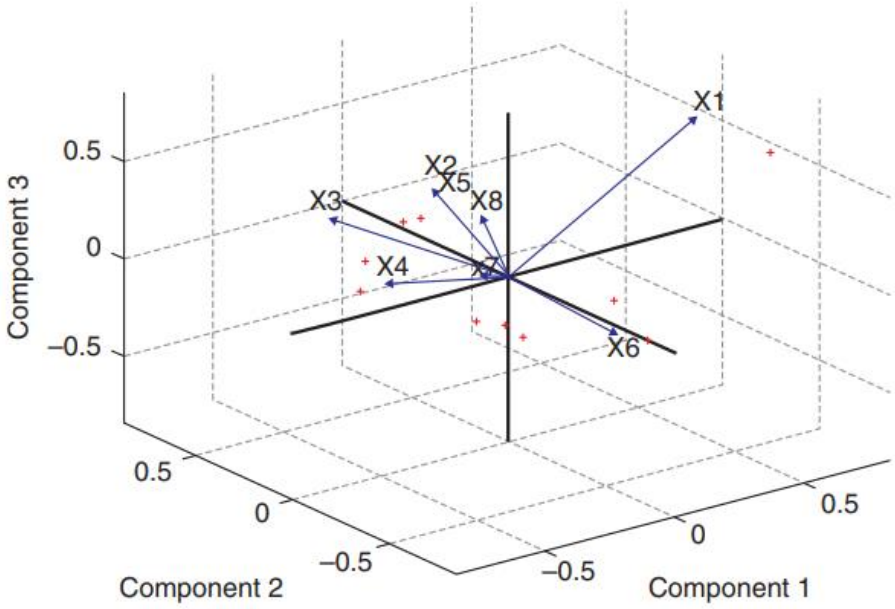


Gambar 5.10 Tingkat kontribusi komponen utama.

Penjelasan ketiga komponen utama di atas untuk masing-masing indikator pemeriksaan fisik (X_1, X_2, … , X_8) ditunjukkan pada Gambar 5.11. Oleh karena itu, gunakan PCA untuk menentukan tiga komponen utama:

$$F_1 = \sum_i \xi_{1i} x_{1i} \ , F_2 = \sum_i \xi_{2i} x_{2i} \ , F_3 = \sum_i \xi_{3i} x_{3i} \quad i = 1, 2, \cdots, m \tag{5.15}$$

Di sini, m = 8, yang merupakan indikator pemeriksaan fisik. Kami dapat menggunakan tiga komponen utama untuk mencerminkan delapan indikator dalam data pemeriksaan fisik. Tingkat retensi informasi adalah 86,86%, yang sangat mengurangi dimensi data dan memfasilitasi analisis data pada tahap selanjutnya:



Gambar 5.11 Penjelasan komponen utama indikator pemeriksaan fisik.

Metode *Machine learning* Semi-Diawasi

Di subbagian ini, kami memperkenalkan tiga algoritme ML alternatif yang berbeda dari *machine learning* yang diawasi. Kebanyakan algoritma pembelajaran tanpa pengawasan bertujuan untuk menemukan representasi yang lebih baik dari input yang diberikan. Hanya konsep dasar dari model pembelajaran ini yang diberikan di bawah ini. Ketiga metode ini tidak memenuhi syarat sebagai algoritma ML yang sama sekali tidak diawasi yang disajikan di bagian sebelumnya.

Machine learning Penguatan

Ini dianggap sebagai algoritma ML yang tidak diawasi. Pasangan input/output yang benar tidak pernah ditampilkan, atau tindakan suboptimal dikoreksi secara eksplisit. Metode ini mengharapkan pengguna untuk mengambil tindakan proaktif untuk memperkuat kualitas data input untuk membantu akurasi prediksi. Ini dianggap sebagai penghargaan kinerja jangka panjang. Sebuah algoritma pembelajaran yang diperkuat menuntut kebijakan yang menghubungkan keadaan model prediksi ke tingkat tindakan penguatan yang akan diambil. Ide pembelajaran penguatan terinspirasi oleh psikologi perilaku. Misalnya, tindakan penguatan dapat dikaitkan dengan teori permainan, teori kontrol, riset operasi, teori informasi, kecerdasan kerumunan, statistik, dan algoritme genetika.

Tujuan akhirnya adalah untuk mencapai beberapa bentuk keseimbangan di bawah rasionalitas terbatas. Beberapa peneliti juga menghubungkan pembelajaran penguatan dengan proses keputusan Markov (MDP) menggunakan teknik pemrograman dinamis. Tindakan penguatan tidak dimulai dengan pembelajaran yang diawasi. Penekanannya adalah untuk

mencapai kinerja on-line. Kita perlu menemukan tradeoff yang tepat antara eksplorasi eksploitasi data yang tidak diketahui dari pengetahuan yang diurutkan dari data yang tersedia. Untuk memperkuat proses pembelajaran, pengguna harus terlebih dahulu mendefinisikan optimalitas. Alih-alih menggunakan kekerasan, pengguna dapat menggunakan pendekatan nilai-fungsi berdasarkan Monte Carlo atau metode perbedaan temporal. Juga, kita dapat mempertimbangkan pendekatan pencarian kebijakan langsung.

Kami juga dapat mempertimbangkan metode pembelajaran penguatan terbalik (IRL). Di sini, tidak ada fungsi hadiah yang diberikan. Sebaliknya, pengguna mengetahui kebijakan di bawah beberapa perilaku yang diamati. Tujuannya adalah untuk meniru perilaku yang diamati menuju optimalitas. Jika proses IRL menyimpang dari jalur perilaku yang diamati, pelatih memerlukan rencana darurat untuk mengembalikan trek ke stabilitas. Ini pada dasarnya adalah pendekatan coba-coba, di mana pengguna mengulangi perilaku yang diamati beberapa kali dengan perubahan kecil setiap kali.

Machine learning Representasi

Pendekatan ML ini mencoba untuk melestarikan informasi penting dalam data input, sementara transformasi dilakukan dengan data untuk menghasilkan model tanpa pengawasan yang lebih baik. Hal ini dapat dilakukan pada tahap pra-pemrosesan sebelum melakukan klasifikasi atau prediksi. Ini mungkin menuntut rekonstruksi input dengan beberapa distribusi data yang tidak diketahui. Contoh klasik pembelajaran representasi meliputi PCA dan analisis klaster yang disajikan dalam Bagian 6.1 dan 6.2. Metode ini juga dikenal sebagai pembelajaran fitur. Pembelajar mencoba mempelajari fitur melalui transformasi input data mentah menjadi representasi yang dapat meningkatkan model prediksi dengan lebih baik.

Metode ini memungkinkan komputer untuk mempelajari tugas tertentu menggunakan fitur dan mempelajari fitur itu sendiri untuk meningkatkan proses pembelajaran. Pembelajaran fitur dimotivasi oleh fakta bahwa klasifikasi menuntut data input yang lebih sederhana untuk mengurangi kompleksitas computing. Data dunia nyata seperti gambar, video dan pengukuran sensor seringkali sangat kompleks. Pelajar harus menemukan fitur atau representasi yang berguna dari data mentah. Fitur kerajinan tangan tradisional seringkali membutuhkan tenaga manusia yang mahal dan seringkali bergantung pada pengetahuan ahli.

Menuju desain teknik pembelajaran fitur yang efisien untuk mengotomatisasi proses pembelajaran, kami memiliki dua pendekatan berdasarkan pembagian antara pembelajaran terawasi dan tidak terawasi:

1) Pembelajaran fitur terbimbing dilakukan dengan data masukan berlabel. Contohnya termasuk jaringan saraf tiruan, perceptron multilayer dan pembelajaran kamus yang diawasi.
2) Pembelajaran fitur tanpa pengawasan menghasilkan fitur dengan data input yang tidak berlabel. Contohnya termasuk pembelajaran kamus, analisis komponen independen, autoencoder, faktorisasi matriks, dan berbagai bentuk pengelompokan yang diperkenalkan sebelumnya: https://en.wikipedia.org/wiki/Feature_learning-citenote-coates2011-3

Machine learning semi-diawasi

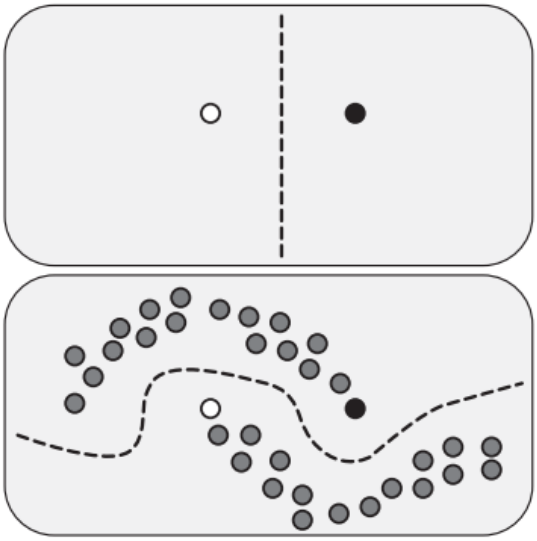
Pendekatan ini menawarkan campuran antara pembelajaran terawasi dan tidak terawasi. Dalam hal ini, pelatih diberikan set data pelatihan yang tidak lengkap dengan beberapa keluaran target (label) hilang. Transduksi adalah kasus khusus dari prinsip ini dimana seluruh rangkaian masalah diketahui pada saat pembelajaran, kecuali bagian dari target yang hilang. Di satu sisi, baik algoritma penguatan dan representasi adalah subkelas dari metode ML semi-terawasi.

Banyak peneliti *Machine learning* menemukan bahwa penggunaan bersama data tidak berlabel dengan sejumlah kecil data berlabel dapat meningkatkan akurasi pembelajaran. Penemuan beberapa data berlabel yang berguna seringkali menuntut keahlian domain atau serangkaian eksperimen fisik untuk dilakukan. Biaya yang terkait dengan proses pelabelan mencegah penggunaan set pelatihan berlabel lengkap. Dengan kata lain, penggunaan sebagian data berlabel lebih masuk akal.

Faktanya, pembelajaran semi-diawasi lebih dekat dengan pembelajaran manusia karena kemampuan kita untuk menangani ketidakjelasan. Tiga asumsi dasar tentang pembelajaran semi-supervised diberikan di bawah ini. Untuk menggunakan data yang tidak berlabel, kita harus mengasumsikan beberapa distribusi data. Algoritme pembelajaran semi-supervised yang berbeda dapat mengasumsikan setidaknya satu dari asumsi berikut (sumber: Wikipedia https://en.wikipedia.org/wiki/Semi-supervised_learning2016):

- **Asumsi kelancaran:** Sampel titik data yang dekat satu sama lain lebih cenderung berbagi label. Hal ini juga umumnya diasumsikan dalam pembelajaran terawasi. Asumsi ini dapat menyebabkan preferensi untuk batas-batas keputusan yang sederhana secara geometris. Dalam kasus pembelajaran semi-diawasi, asumsi kelancaran menghasilkan preferensi untuk batas-batas keputusan di daerah dengan kepadatan rendah.
- **Asumsi klaster:** Data cenderung membentuk klaster diskrit, dan titik-titik dalam klaster yang sama lebih cenderung berbagi label. Berbagi label dapat tersebar di beberapa cluster, asumsi yang terkait dengan pembelajaran fitur dengan algoritma clustering.
- **Asumsi manifold:** Data terletak kira-kira pada manifold dengan dimensi yang jauh lebih rendah daripada ruang input. Dalam hal ini, kita mempelajari manifold menggunakan data berlabel dan tidak berlabel untuk menghindari masalah dimensi. Dengan demikian, pembelajaran semi-terawasi dapat dilanjutkan dengan menggunakan jarak dan kepadatan yang ditentukan pada manifold.

Asumsi manifold praktis ketika dataset berdimensi tinggi ditemui. Misalnya, suara manusia dikendalikan oleh beberapa pita suara, dan gambar berbagai ekspresi wajah dikendalikan oleh beberapa otot. Kami ingin dalam kasus ini untuk menggunakan jarak dan kehalusan di ruang data alami, daripada di ruang semua kemungkinan gelombang akustik atau gambar masing-masing. Contoh berikut menunjukkan keuntungan *Machine learning* semi-diawasi.



Gambar 5.12 Contoh untuk menggambarkan *Machine learning* semi-diawasi.

Contoh 5.8 *Machine learning* Semi-Diawasi

Contoh ini dari Wikipedia (https://en.wikipedia.org/wiki/Semi-supervised_ learning). Tujuannya adalah untuk menunjukkan pengaruh data yang tidak berlabel pada pembelajaran semi-diawasi. Panel atas pada Gambar 5.12 menunjukkan batas keputusan yang mungkin kita adopsi setelah melihat hanya satu contoh positif (lingkaran putih) dan satu negatif (lingkaran hitam).

Panel bawah menunjukkan batas keputusan yang mungkin kita adopsi jika, selain dua contoh berlabel, kita diberi kumpulan data tak berlabel (lingkaran abu-abu). Ini dapat dilihat sebagai melakukan pengelompokan dan kemudian memberi label pada klaster dengan data berlabel, mendorong batas keputusan menjauh dari daerah dengan kepadatan tinggi, atau mempelajari manifold 1-D tempat data berada.

5.4 BAGAIMANA MEMILIH ALGORITMA *MACHINE LEARNING*?

Metode untuk memilih model yang tepat untuk *Machine learning* dipelajari di bawah ini. Beberapa strategi dan solusi yang masuk akal diperkenalkan. Kami mempertimbangkan pemahaman data melalui visualisasi, pemilihan algoritme ML, dan solusi yang terlalu pas atau tidak pas. Terakhir, kami menyajikan prosedur untuk pemilihan algoritma ML. Kami juga membahas keuntungan dan kerugian menggunakan fungsi kerugian yang berbeda.

Metrik Kinerja dan Pemasangan Model

Setiap algoritma ML memiliki potensi aplikasinya sendiri. Mengingat kumpulan data, kinerja algoritma yang diberikan mungkin sangat baik, tetapi algoritma lain mungkin sebaliknya. Selain itu, mengubah ke kumpulan data yang berbeda juga dapat mengubah kesimpulan secara drastis. Dengan demikian, agak sulit untuk menilai algoritma mana yang lebih baik daripada yang

lain dalam kasus-kasus umum. Sangat penting untuk memperkenalkan beberapa metrik umum untuk mengevaluasi algoritme ML. Beberapa metrik dapat diadopsi untuk mengungkapkan manfaat relatif. Lainnya dapat digunakan untuk menemukan algoritme serupa yang dapat lebih mudah diterapkan.

Metrik Kinerja Algoritma ML

Kami mempertimbangkan tiga metrik dasar untuk memenuhi berbagai persyaratan kinerja seperti yang dinyatakan:

1) Akurasi: Ini adalah kriteria paling penting untuk mengevaluasi kinerja ML, berdasarkan pengujian kumpulan data. Ada dua kasus: algoritma over-fitting atau under-fitting. Jelas, semakin tinggi kinerja yang ditunjukkan oleh set pelatihan, semakin baik kecocokan algoritma.
2) Waktu pelatihan: Ini mengacu pada kecepatan konvergensi suatu algoritma, atau waktu yang dibutuhkan untuk menetapkan optimalitas model kerja. Jelas, semakin pendek waktu pelatihan, semakin baik model yang dibangun untuk tampil dengan biaya implementasi yang lebih rendah.
3) Linearitas: Ini adalah properti model yang mencerminkan kompleksitas algoritma ML yang diterapkan. Kinerja linier menyiratkan beberapa bentuk kinerja yang dapat diskalakan. Dalam praktiknya, algoritma linier dengan kompleksitas yang lebih rendah lebih sering diinginkan, karena dapat menyebabkan waktu pelatihan yang lebih pendek atau akurasi yang lebih tinggi dengan biaya yang lebih rendah.

Prapemrosesan Data

Sebelum menganalisis data, pola pertumbuhan data dan korelasi antar data harus diungkapkan. Ini menuntut visualisasi data, terutama ketika kumpulan *Big data* atau multi-dimensi ditemukan. Karena hanya grafik 2-D atau 3-D yang dapat ditampilkan, praproses diperlukan untuk mengurangi dimensi ruang data ke dimensi yang lebih sedikit.

Kualitas data memengaruhi efektivitas proses pelatihan ML dan juga kinerja. Untuk mencapai tujuan kinerja di atas, kami mempertimbangkan beberapa metode untuk meningkatkan kualitas data selama tahap penemuan data, pengumpulan, persiapan, dan pra-pemrosesan. Tujuannya agar data tersebut lebih lengkap, relevan, dan teratur untuk digunakan sebagai data latih atau menghasilkan hasil validasi silang yang lebih tinggi:

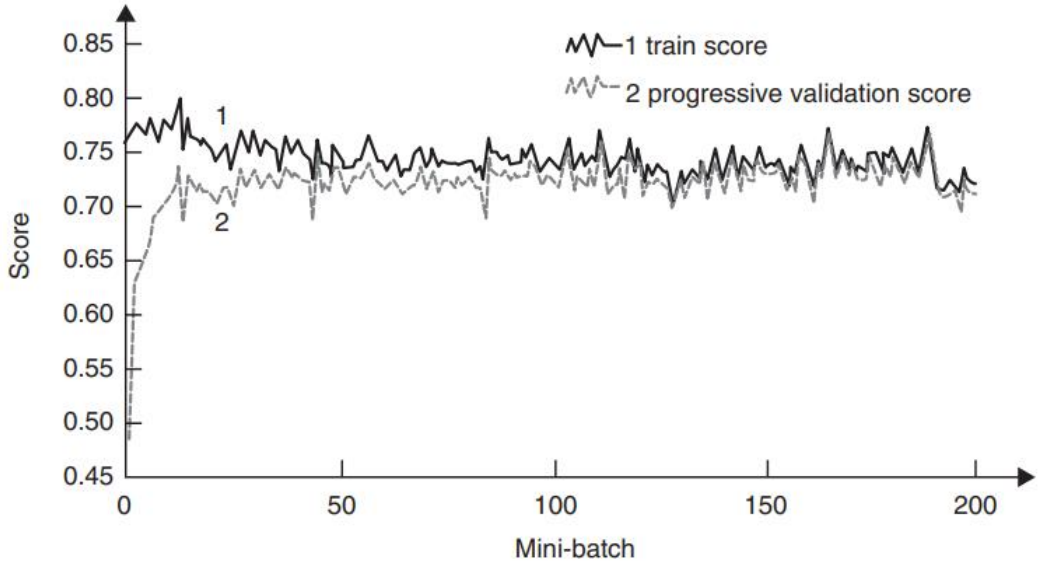
- **Mengisi data yang hilang:** Karena statistik yang tidak lengkap atau catatan tulisan tangan yang tidak dapat dikenali, kami tidak dapat menjamin data lengkap di setiap baris dan setiap kolom dari kumpulan data mentah. Dalam hal ini, untuk merekonstruksi data yang hilang diinginkan. Umumnya, penyelesaian Mean/mode atau imputasi dek panas diadopsi untuk mengisi data yang hilang agar lebih lengkap untuk memenuhi kebutuhan kita.
- **Pra-pemrosesan data yang salah:** Seringkali kita menemukan beberapa data yang jelas-jelas salah karena kesalahan ketik atau kesalahan pencatatan. Terkadang, data dapat menyimpang dari keteraturan karena nilainya terlalu besar atau terlalu kecil. Ini

menuntut pemasangan data atau beberapa metode interpolasi atau ekstrapolasi untuk merampingkan data itu.

- **Regularisasi data:** Banyak fitur data dan unit fisiknya sangat berbeda. Jika dua unit data berbeda beberapa kali lipat, maka diperlukan proses normalisasi atau standarisasi. Untuk tujuan ini, Persamaan (5.19) digunakan, di mana x adalah data yang tidak diatur atau tidak dinormalisasi dan x' adalah data yang diatur atau dinormalisasi. Proses regularisasi diatur oleh salah satu dari perhitungan ini:

$$\begin{cases} x' = \dfrac{x}{\max(x)} \\ x' = \dfrac{x - \min(x)}{\max(x) - \min(x)} \end{cases} \tag{5.16}$$

- Dukungan Visualisasi: Untuk data multidimensi, visualisasi dilakukan dengan ekstraksi fitur dan pengurangan dimensi. Pengguna mungkin ingin memvisualisasikan fitur utama terlebih dahulu. Metode pengurangan dimensi diadopsi untuk menghilangkan beberapa fitur yang tidak signifikan. Misalnya, angka numerik dapat dikonversi dari data yang direkam ke bentuk digital atau grafik.



Gambar 5.13 Skor pelatihan dan skor validasi silang cocok dengan baik dalam model *Machine learning* yang dibuat dengan baik.

Skor Kinerja Machine learning

Untuk mengukur kinerja algoritma ML, kita dapat mendefinisikan beberapa skor kinerja. Skor ini dinormalisasi sebagai persentase dengan 100% untuk skor sempurna dan pecahan kecil untuk skor yang lebih rendah. Skor ini sering merupakan fungsi tertimbang dari ketiga metrik kinerja yang diperkenalkan di Bagian 5.4.1. Grup pengguna yang berbeda dapat menerapkan

fungsi pembobotan yang berbeda untuk menekankan pilihan pilihan mereka. Seringkali akurasi bobot tertinggi dan waktu pelatihan mungkin sekunder. Linieritas bisa menjadi yang paling penting atau diabaikan begitu saja jika pelajar dibatasi oleh biaya implementasi.

Kinerja algoritma ML sering diplot sebagai kurva kinerja pembelajaran, seperti yang ditunjukkan pada Gambar 5.13. Dalam kurva pembelajaran ini, skor ditampilkan pada sumbu y terhadap sampel pelatihan atau ukuran data pengujian pada sumbu x. Ada dua skor bersaing yang diilustrasikan dalam kurva kinerja seperti itu. Skor pelatihan didorong oleh set data pelatihan yang diterapkan. Skor validasi silang didasarkan pada pengujian progresif dari semua data yang masuk. Secara umum, skor pelatihan lebih tinggi dari skor validasi karena model dibangun dari dataset pelatihan. Gambar 5.13 menunjukkan kasus yang ideal, di mana kedua skor bertemu dengan cepat setelah pengujian yang cukup oleh kumpulan data mini.

Kasus Pemasangan Model dalam Proses Machine learning

Kami mempertimbangkan di bawah dua kasus model yang kurang pas dalam proses memilih algoritme *Machine learning* yang dapat diterima untuk diterapkan dalam berbagai kondisi kinerja set data pelatihan dan pengujian:

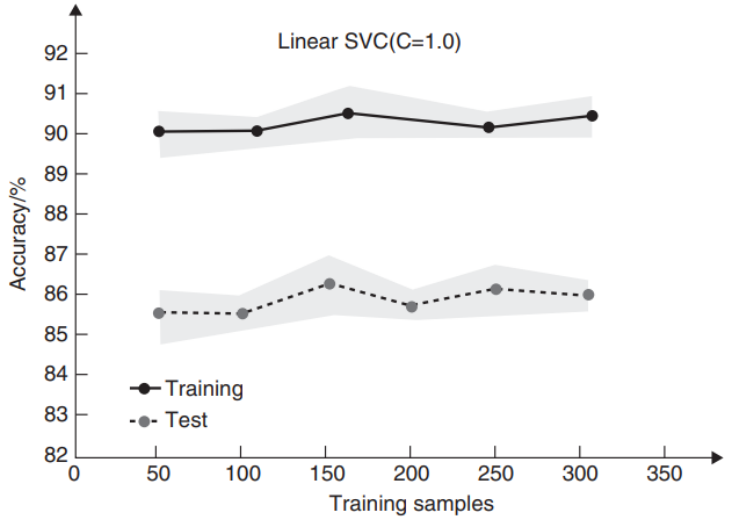
- **Pemodelan Over-Fitting:** Ini adalah kasus di mana skor pelatihan sangat tinggi, tetapi skor validasi silang sangat rendah untuk menguji kumpulan data yang diterapkan. Seperti yang ditunjukkan pada Gambar 5.14, kedua skor dipisahkan berjauhan satu sama lain. Status ini menyiratkan bahwa model sangat cocok dengan set pelatihan. Namun, model telah mengabaikan margin noise dalam dataset validasi. Dengan kata lain, set pelatihan sangat bias pada set data pelatihan tertentu. Kumpulan data sampel ini jauh dari distribusi atau karakteristik data umum dalam aplikasi umum. Dalam hal ini, model overfitting tidak dapat memodelkan data pengujian secara akurat.
- **Under-Fitting Modeling:** Ini adalah kasus di mana model yang dihasilkan oleh set pelatihan tertentu berakhir dengan kinerja skor yang sangat rendah, yang jauh di bawah harapan pengguna. Fenomena under-fitting menyiratkan bahwa set pelatihan yang buruk dipilih dan model yang dilatih sehingga diperoleh tidak dapat bekerja dengan baik pada dataset pengujian nyata. Oleh karena itu, model ini sama sekali tidak dapat diterima oleh pengguna.

Baik over-fitting maupun under-fitting tidak dapat diterima. Ini menyiratkan bahwa kita harus memilih dataset pelatihan yang tepat yang dapat mewakili dataset umum. Dalam subbagian berikutnya, kami akan menyarankan sejumlah metode untuk mengatasi dua kekurangan ini dalam pemodelan *Machine learning*. Singkatnya, model yang ideal harus bekerja dengan baik baik di dataset pelatihan dan dataset lainnya dalam aplikasi umum.

Dengan bertambahnya ukuran data sampel, skor pelatihan sedikit menurun sementara peningkatan kecil dari skor validasi silang diamati. Ini dikirim melalui algoritma Linear-SVC dengan faktor regulasi C = 1. Secara umum, skor pelatihan seringkali lebih tinggi daripada skor validasi silang. Tetapi kesenjangan di antara mereka harus diminimalkan dengan segala cara. Ini

menyiratkan bahwa model lebih sering jatuh ke dalam kasus yang terlalu pas. Ketika kita mengikuti perilaku set pelatihan, kita harus memperhatikan penurunan skor validasi silang. Kami akan menyerang masalah over-fitting di Bagian 5.4.2 dan masalah under-fitting di Bagian 5.4.3.

Secara umum, skor pelatihan seringkali lebih tinggi daripada skor validasi silang. Ini menyiratkan bahwa model lebih sering jatuh ke dalam kasus yang terlalu pas. Dengan demikian, model melakukan segala upaya untuk mengikuti perilaku set pelatihan. Skor validasi silang yang lebih rendah dipengaruhi oleh noise yang jauh lebih tinggi dalam kumpulan data pengujian, jadi kita harus mengatasi kesulitan yang berasal dari masalah over-fitting dan under-fitting. Beberapa pendekatan diberikan dalam dua bagian berikutnya.



Gambar 5.14 Kasus over-fitting saat membuat model pembelajaran menggunakan algoritma linear-SVC dengan dataset kecil hingga 160 sampel.

Metode untuk Mengurangi Model Over-Fitting

Alasan utama untuk over-fitting adalah bahwa model sengaja mengingat sifat distribusi dari sampel pelatihan. Dengan kata lain, model yang dibuat terlalu bias oleh perilaku data sampel. Model over-fitting mendapat skor sangat tinggi pada set pelatihan tertentu, tetapi skor buruk pada set data lainnya. Dengan kata lain, kesenjangan skor besar harus ditutup di berbagai kumpulan data yang diterapkan. Tercantum di bawah ini adalah beberapa metode untuk mengurangi efek buruk ini.

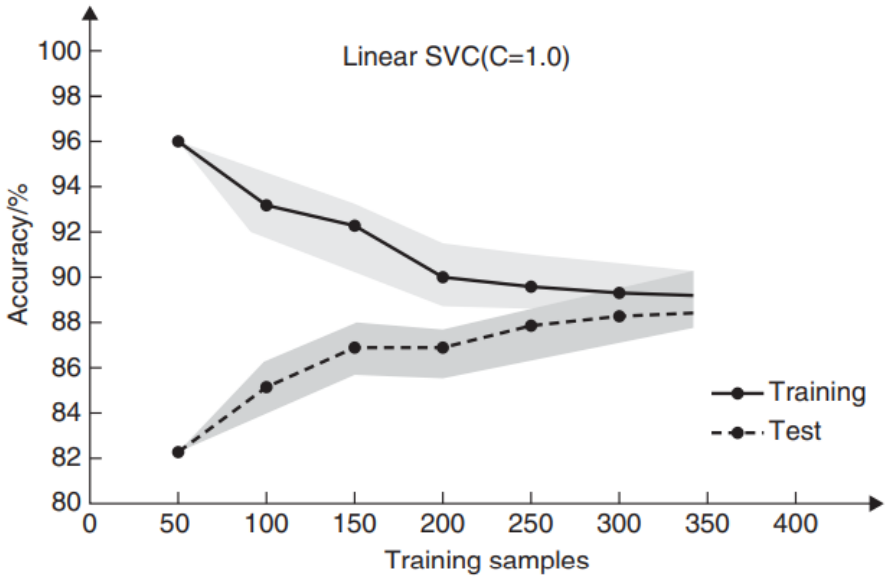
Meningkatkan Ukuran Data Pelatihan

Peningkatan jumlah sampel dapat membuat set pelatihan lebih representatif untuk mencakup lebih banyak variasi dan kelangkaan data. Peningkatan sampel yang diterapkan mencerminkan efek kebisingan yang lebih baik, dengan nilai rata-rata kebisingan dikurangi menjadi nol. Artinya, pengaruh noise pada data pengujian bisa sangat dikurangi. Metode umum untuk meningkatkan ukuran sampel adalah dengan mengumpulkan lebih banyak data di bawah skenario yang sama. Terkadang, pelabelan manual ditambahkan untuk menghasilkan beberapa

sampel pelatihan buatan. Misalnya, kita dapat menerapkan pengenalan gambar, transformasi cermin, dan rotasi untuk memperbesar jumlah sampel. Meskipun operasi ini mungkin padat karya, metode ini meningkatkan ketergantungan sampel. Ini akan meningkatkan model dengan menghindari bias pelatihan.

Contoh 5.9 Memperbesar Sample Dataset untuk Algoritma Linear SVC

Seperti yang ditunjukkan pada Gambar 5.15, kumpulan data sampel sekarang diperbesar dari 150 menjadi 400. Kumpulan data yang diperbesar ini menghasilkan konvergensi yang baik dari kurva skor saat kumpulan data meningkat melampaui 200. Kedua skor menjadi sangat dekat satu sama lain karena ukuran data sampel meningkat melebihi 300.



Gambar 5.15 Mengurangi efek model over-fitting dengan memperbesar set pelatihan menjadi 800 sampel.

Dalam keadaan dengan ukuran data sampel kecil yang tidak dapat ditingkatkan lebih lanjut, kita dapat mengurangi efek noise dengan mengubah kumpulan sampel yang ada, seperti dengan menggunakan analisis wavelet. Tujuannya adalah untuk mengurangi kebisingan rata-rata menjadi nol. Sementara itu, varians noise juga berkurang, sehingga mengurangi pengaruh noise pada semua data yang akan diuji. Saat sampel pelatihan meningkat, perbedaan antara skor pelatihan dan skor validasi silang dapat dikurangi, seperti yang ditunjukkan oleh contoh berikut.

Metode Penyaringan Fitur dan Pengurangan Dimensi

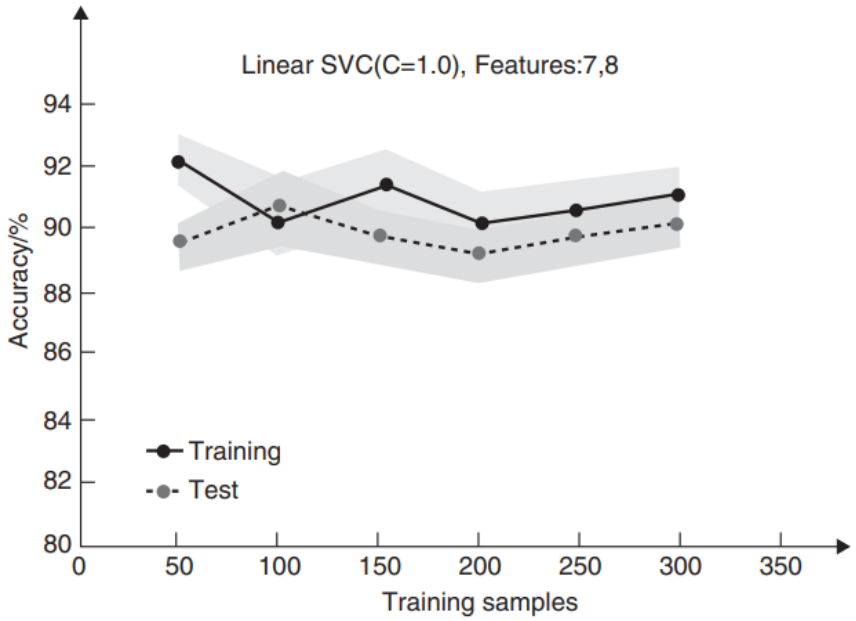
Terkadang kita mungkin memiliki kumpulan data pelatihan besar yang dicirikan oleh banyak fitur sampel. Dengan mengungkapkan korelasi antar fitur, kita dapat memotong beberapa fitur untuk mengurangi efek over-fitting. Fitur-fitur dengan kekuatan perwakilan terbatas dihapus. Ini disebut penyaringan fitur atau pengurangan dimensi. Bahkan, kita dapat melintasi semua gaya kombinasi fitur dan memilih fitur yang lebih penting. Dalam kasus sampel

dengan dimensi tinggi, analisis asosiasi atau analisis korelasi dapat diadopsi untuk menghilangkan beberapa fitur lemah dengan mengurangi dimensi.

Kadang-kadang sulit untuk menentukan hubungan antara fitur ortogonal. Dalam hal ini, algoritma PCA menerapkan pengurangan dimensi. Dalam kasus di mana dimensi ruang fitur tidak tinggi, kami melakukan penyaringan fitur untuk mengurangi kompleksitas model. Ada tiga metode pendekatan untuk melakukannya: i) penurunan derajat polinomial dalam model ML; ii) mengurangi lapisan jaringan saraf tiruan dan jumlah node di setiap lapisan; dan iii) meningkatkan bandwidth RBF-kernel dalam algoritma SVM.

Contoh 5.10 Menggunakan Lebih Sedikit Fitur (Dimensi) untuk Mengurangi Efek Over-Fitting

Kami dapat menerapkan analisis asosiasi untuk menilai berbagai dampak fitur dalam algoritma PCA. Pada Gambar 5.16, kami mendemonstrasikan efek penggunaan lebih sedikit fitur dalam algoritma Linear-SVC. Dalam hal ini, fitur 7 dan fitur 8 dipilih secara manual setelah diamati perannya yang lebih berat dalam proses pembelajaran. Terkadang sulit untuk menentukan hubungan antara fitur ortogonal. Dalam hal ini, algoritma PCA harus diterapkan untuk pengurangan dimensi.



Gambar 5.16 Efek penggunaan lebih sedikit fitur dalam algoritma Linear-SVC.

Pengaruh Regularisasi Data

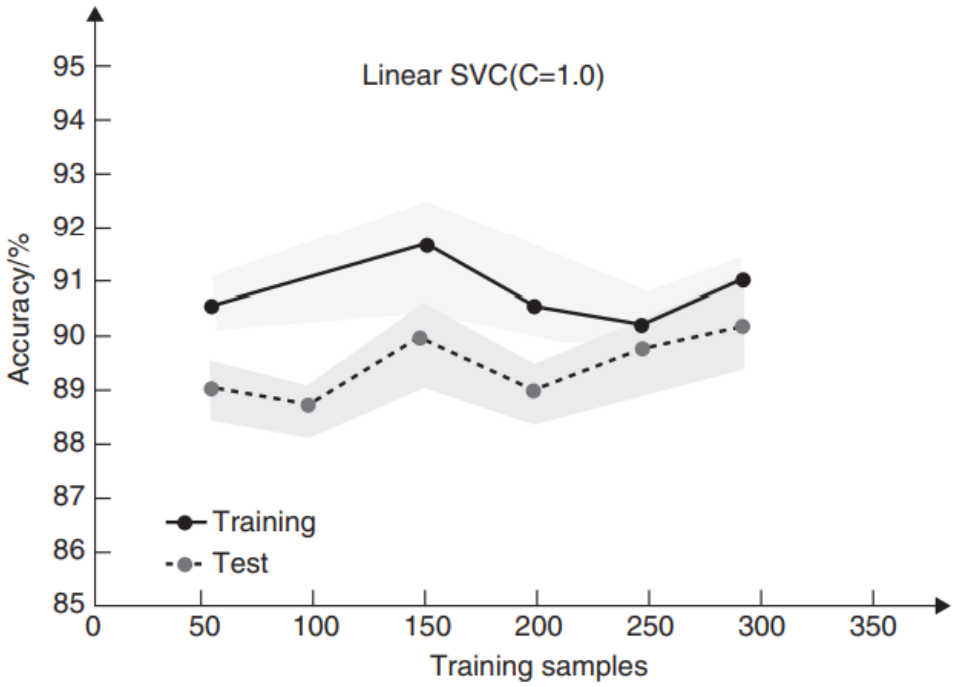
Dengan menyesuaikan parameter regularisasi data, kita dapat mengurangi efek over-fitting sampai batas tertentu. Pencarian grid dapat dilakukan pada set validasi silang untuk menemukan parameter regularisasi yang optimal. Untuk tujuan ini, kita baru saja melihat jalannya pemilihan fitur dengan SelectKBest dalam pemilihan sklearn.feature. Seperti disebutkan di atas, kursus ini mungkin sangat lambat dalam kasus fitur dengan dimensi tinggi. Kami harus

membedakan beberapa fitur melalui beberapa metode regularisasi data, seperti dalam dua kasus:

1) **Regulerisasi L1:** Metode regularisasi ini membuat bobot karakteristik terdistribusi secara jarang. Dengan kata lain, karakteristik, yang tidak penting untuk hasil akhir, dapat diberikan dengan bobot nol.
2) **Regulerisasi L2:** Metode ini menyebarkan bobot fitur pada berbagai dimensi fitur sejauh mungkin. Hamburan mencegah berat dari berkonsentrasi pada dimensi tertentu.

Contoh 5.11 Menerapkan Regularisasi Data L1 untuk Mengurangi Efek Over-Fitting

Kami menerapkan regulasi-L1 pada Gambar 5.17 untuk membuat bobot fitur terdistribusi secara jarang dalam algoritma Linear-SVC dalam aplikasi klasifikasi. Metode ini secara otomatis mendiskriminasi fitur dengan bobot yang sangat rendah atau nol. Mempertimbangkan semua bobot fitur 5-D, 9-D, 11-D, 12-D, 17-D, dan 18-D, kami mempertahankan fitur 11-D agar memiliki bobot tertinggi. Faktor bobot ditandai sebagai C = 1,0 pada Gambar 5.17. Dengan ukuran kecil 300 sampel, skor pelatihan dan pengujian bertemu satu sama lain dengan baik.



Gambar 5.17 Pengaruh penerapan regularisasi data L1 terhadap kinerja linear-SVC.

Metode untuk Menghindari Model yang Kurang Pas

Under-fitting terjadi dalam dua situasi: i) dataset tidak dipersiapkan dengan baik, dan tidak dapat bekerja dengan baik dalam proses pelatihan dan validasi; dan ii) algoritme *Machine learning* salah dipilih, dengan mempertimbangkan sifat lingkungan masalah. Dengan kata lain, kumpulan data yang berbeda mungkin berlaku untuk algoritme yang dipilih secara berbeda.

Untuk alasan ini, masalah under-fitting sulit untuk dipecahkan sepenuhnya. Pendekatan yang lebih layak adalah menemukan cara untuk menghindari masalah yang kurang pas. Di bagian ini, kami menunjukkan dua metode untuk mengurangi efek yang kurang pas pada skor kinerja. Di Bagian 5.4.4, kami akan menyarankan metode intuitif untuk memilih algoritme *Machine learning* yang paling sesuai dari lima kategori pada tingkat fungsional tertinggi.

Perubahan Parameter Campuran

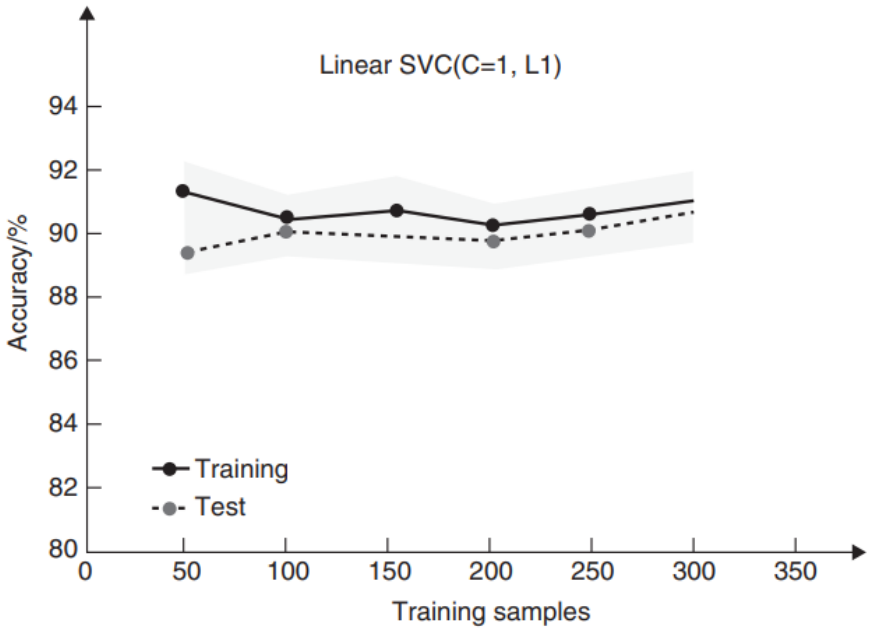
Pertimbangkan masalah yang kurang pas saat menggunakan model SVM (Support Vector Machine) untuk memecahkan masalah klasifikasi. Salah satu caranya adalah dengan memanfaatkan jaringan syaraf tiruan (JST) untuk melatih sistem agar menghasilkan model fit yang lebih baik. Dalam contoh lain dari model under-fitting, kita dapat memodifikasi fungsi kernel untuk menutupi kasus klasifikasi non-linear. Sebagai contoh, kita dapat mengganti classifier SGD (Stochastic Gradient Descent) dengan multi-layer JST. Di sini pendekatan kernel diadopsi untuk menyelesaikan tugas. Contoh 5.12 menunjukkan modifikasi dua parameter dalam algoritma SVC Linear, yang ditunjukkan pada contoh sebelumnya, untuk menghasilkan kecocokan yang lebih baik untuk pengembangan model pembelajaran SVC.

Contoh 5.12 Mengubah Model Linear-SVC dengan Reduced C = 0.1 dan L1 Penalty

Perubahan skor pada data kecil setelah iterasi 50 mini-batch data sampling. Skornya rendah untuk mencerminkan under-fitting. Performa menurun dengan cepat jika menggunakan model yang kurang pas. Untuk model LinearSVC yang telah kami uji dalam contoh sebelumnya, kami dapat mengurangi faktor regularisasi C menjadi 0,1 dari 1 dan menerapkan penalti regularisasi L1 pada waktu yang sama. Gambar 5.18 memplot skor tinggi yang ditingkatkan (sekitar 0,91) di kedua skor di berbagai ukuran kumpulan data sampel dari 50 hingga 300. Jadi, melalui perubahan campuran parameter model ini, kami berakhir dengan kecocokan yang cukup dekat antara skor pelatihan dan skor validasi silang.

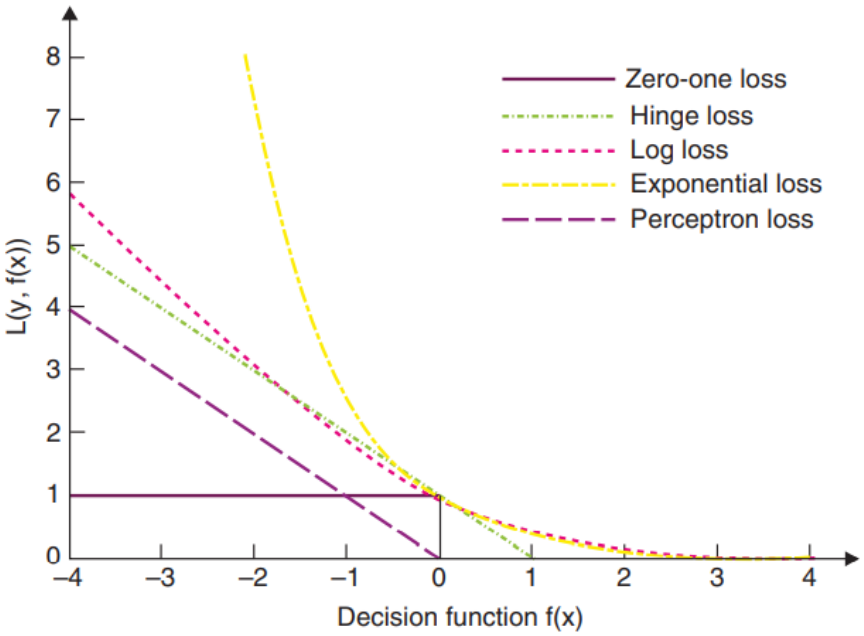
Mengubah Fungsi Rugi

Masalah *Machine learning* juga dapat dilihat sebagai minimalisasi dari beberapa fungsi kerugian dari contoh pelatihan. Fungsi kerugian mengungkapkan perbedaan antara prediksi model terlatih dan contoh masalah yang sebenarnya. Misalnya, masalah klasifikasi mengharuskan pengguna untuk menetapkan label ke contoh tersebut, dan dengan menggunakan model terlatih untuk memprediksi label sampel pelatihan, fungsi kerugian kemudian mencerminkan perbedaan dari dua jenis set label. Dengan demikian, fungsi kerugian mengungkapkan efek kehilangan kinerja yang diharapkan dari algoritma ML. Algoritme yang dioptimalkan harus meminimalkan kerugian dalam set pelatihan, sementara *Machine learning* berkaitan dengan meminimalkan kerugian dalam sampel yang tidak terlihat. Pemilihan fungsi kerugian sangat penting untuk mendapatkan model prediksi yang lebih baik atau optimal. Kami mempertimbangkan di bawah lima pilihan desain fungsi kerugian. Efeknya diplot pada Gambar 5.19.



Gambar 5.18 Hasil Under-Fitting dalam algoritma Linear-SVC.

1) **Fungsi zero-one loss:** Kebijakan ini menawarkan pemisahan yang sangat tajam antara keberhasilan dan kegagalan. Fungsi kerugian 0-1 menghitung jumlah prediksi yang salah dalam masalah klasifikasi. Namun, ini adalah fungsi non-cembung, tidak praktis dalam aplikasi kehidupan nyata.
2) **Fungsi kehilangan engsel:** Ini sering digunakan dalam aplikasi SVM (Support Vector Machines) karena kekuatan relatifnya untuk mencerminkan sensitivitas yang tidak biasa terhadap efek kebisingan. Fungsi ini tidak didukung oleh distribusi probabilistik.
3) **Fungsi kerugian log:** Fungsi kerugian ini dapat mencerminkan distribusi probabilistik. Dalam klasifikasi multi-kelas, kita perlu mengetahui kepercayaan dari klasifikasi tersebut. Fungsi log-loss lebih cocok. Namun, kekurangannya terletak pada kepekaan yang lebih rendah terhadap kebisingan dan kurangnya kekuatan penilaian.
4) **Fungsi kerugian eksponensial:** Ini telah diterapkan di AdaBoost. Hal ini sangat sensitif terhadap pemisahan dari keramaian dan kebisingan. Gaya prediksinya sederhana dan efektif dalam menangani algoritma boosting.
5) **Fungsi kehilangan perceptron:** Ini dapat dianggap sebagai variasi dari kehilangan engsel. Kehilangan engsel menimbulkan hukuman berat dalam salah menilai titik batas, sementara kehilangan perceptron dipenuhi dengan klasifikasi yang akurat dari data sampel. Skor ini mengabaikan jarak dari batas penilaian. Keuntungannya adalah lebih sederhana daripada menggunakan fungsi kerugian engsel. Kekurangannya terletak pada kenyataan bahwa ia menawarkan model yang lebih lemah untuk diterapkan pada masalah umum, karena kurangnya batas margin maksimum.



Gambar 5.19 Pengaruh penggunaan fungsi kerugian yang berbeda dalam pemilihan model *Machine learning*.

Modifikasi Model Lain atau Pendekatan Ensemble

Algoritma PCA (Analisis Komponen Utama) yang dipelajari di Bagian 5.3.2 menawarkan pendekatan pengurangan dimensi untuk mengurangi kompleksitas model. Kami juga dapat mempertimbangkan untuk menggunakan beberapa komponen utama untuk menghubungkan elemen data. Selain itu, independensi masing-masing komponen utama kuat, sehingga sangat mengurangi koneksi internal antar data.

Munculnya algoritma ensemble memberikan solusi lain untuk masalah underfitting, di mana setiap model individu tidak berkinerja baik dalam kumpulan data yang diberikan. Kami dapat mempertimbangkan untuk menggunakan beberapa algoritme secara bersamaan dalam kumpulan data yang sama, lalu memilih salah satu yang paling sesuai dengan skor kinerja. Sebagai contoh penerapan model Adaboost dan Decision Tree secara bersama-sama untuk meningkatkan akurasi hasil prediksi.

Efek Penggunaan Fungsi Rugi yang Berbeda

Mengingat set data dari domain aplikasi yang diketahui, prosedur berikut dalam Algoritma 5.6 menunjukkan cara memilih algoritme *Machine learning* yang tepat, berdasarkan karakteristik set data dan persyaratan kinerja. Lima kategori algoritma *Machine learning* dipertimbangkan. Secara umum, opsi berikut dapat digunakan untuk menyelesaikan masalah yang kurang pas. Metode-metode ini muncul, khususnya, untuk memperbaiki masalah klasifikasi, karena kinerja model sangat sensitif terhadap kumpulan data yang diterapkan. Kami

mempertimbangkan tiga opsi dalam memilih kumpulan data. Pemilihan dataset didorong oleh permintaan kinerja.

Tiga pilihan pilihan diberikan di bawah ini:

1) **Common Datasets:** Membagi dataset asli menjadi dua bagian: dengan karakteristik dan distribusi yang sama dalam training set dan testing set. Subdivisi ini harus menghasilkan kinerja model untuk menghindari masalah over-fitting atau under-fitting.
2) **Validasi Silang:** Membagi dataset asli menjadi k bagian dan memilih satu bagian secara bergantian sebagai set pengujian, dengan sisanya sebagai set pelatihan. Ini menuntut uji validasi k berjalan. Performa model ini menunjukkan akurasi rata-rata model pada banyak set pengujian yang dibagi lagi.
3) **Siklus Bootstrap:** Pengambilan sampel secara acak dengan penggantian beberapa elemen data berulang kali dalam sampel pelatihan yang berbeda. Biarkan data sampel menjadi set pelatihan, dengan sisanya sebagai set pengujian. Ulangi siklus pengambilan sampel ini sebanyak k kali. Kami mungkin berakhir dengan kinerja rata-rata tertimbang dari semua set tes yang diterapkan.

Algoritma 5.6 Memilih Algoritma *Machine learning* dari lima Kategori

Input: Dataset input

Output: Kategori algoritma

Prosedur:

1) Preprocessing untuk meningkatkan kualitas data (regulasi dapat ditunda ke proses model)
2) Persyaratan visualisasi (memilih algoritme berdasarkan hasil visualisasi)
3) Tentukan Fungsi Tujuan (berfokus pada properti fitur atau hasil yang diharapkan)
4) Jika properti fitur dipilih, pilih algoritma korelasi fitur (Kategori 1)
5) Atau pilih hasil yang diharapkan
6) Bagi seluruh kumpulan data (set data pelatihan versus set data pengujian)
7) Apakah label ada dalam kumpulan data?
8) Jika ya, maka keluar dari label
9) Pilih algoritma klasifikasi (Kategori 2)
10) Atau hanya sebagian label yang ada
11) Pilih algoritma pembelajaran semi-unsupervised (Kategori 3)
12) Atau tidak ada label
13) Pilih algoritma pengelompokan (Kategori 4)
14) Jalankan model yang diperoleh
15) Mengukur hasil (membuat prediksi)
16) Dapatkan nilai ujian
17) Jika skor memuaskan

18) Keluarkan hasilnya
19) Atau skor akurasinya terlalu rendah
20) Pilih algoritma ensemble (Kategori 5), lanjutkan ke Langkah 4
21) Ulangi jumlah iterasi yang telah ditentukan sebelumnya
22) Outputnya diragukan, jadi pilih dataset yang berbeda atau kurangi standarnya.

5.5 KESIMPULAN

Machine learning sangat diminati dengan munculnya ilmu data dan industri *Big data*. Semakin banyak sarjana mulai fokus pada bidang ini dan semakin banyak algoritma telah diusulkan. Bab ini berfokus pada pembelajaran tanpa pengawasan dan pembelajaran semi-diawasi dalam *Machine learning*. Bagian terakhir dari bab ini memperkenalkan cara memilih algoritme yang sesuai di antara banyak algoritme *Machine learning*. Detail lebih lanjut dari berbagai algoritma ML dapat ditemukan dalam referensi di bawah ini.

Tugas dan Latihan

1. Tabel 5.12 menyajikan absis dan ordinat titik data dalam eksperimen *Machine learning*. Terapkan algoritma pengelompokan K-means untuk membagi titik-titik ini menjadi tiga cluster menggunakan fungsi jarak Euclidean. Anda harus mengidentifikasi pusat (centroids) dan menunjukkan proses pengelompokan rinci langkah demi langkah. Gambarlah diagram ruang Euclidean 2-D untuk menunjukkan hasil partisi akhir pada 10 titik data dan batas cluster.

Tabel 5.12 Absis dan ordinat beberapa titik data.

ID Poin	Absis	Ordinat	ID Poin	Absis	Ordinat
1	0	0	6	4	11
2	2	3	7	6	9
3	4	2	8	8	10
4	0	6	9	12	6
5	3	10	10	7	9

2. Anda telah mempelajari empat metode pengelompokan yang berbeda di Bagian 5.2. Bedakan perbedaan mendasar mereka dalam metodologi, kekuatan dan kelemahan. Analisis kompleksitas computing dan persyaratan implementasi pada cloud yang dipilih jika kumpulan data yang sangat besar diproses. Diskusikan skenario di mana setiap metode paling cocok. Berdasarkan fungsi kesamaan atau jarak yang diterapkan, Anda harus mengidentifikasi satu aplikasi kehidupan nyata untuk setiap metode pengelompokan. Membenarkan pengamatan dan klaim Anda dengan penalaran analitis dan penilaian kinerja numerik, dengan data yang tersedia dari domain publik atau literatur.

3. Selalu ada hubungan yang pasti antara komoditas komersial dan harga berlabelnya di department store. Kebijakan penetapan harga adalah apa yang Anda diminta untuk bekerja dengan kumpulan data kecil yang diberikan pada Tabel 5.13. Ada lima produk yang diamati selama 8 bulan variasi harganya. Nilai biner “1” mewakili kenaikan harga selama 8 bulan terakhir. "0" tidak menandai kenaikan harga barang produk tertentu. Buat seperangkat aturan asosiasi untuk menentukan kebijakan harga. Terapkan aturan untuk menentukan harga kolom produk baru. Justifikasi aturan asosiasi yang telah Anda peroleh dan terapkan.
4. Tingkat perkembangan suatu kota dinilai dengan banyak indikator, seperti jumlah penduduk, jumlah volume angkutan penumpang dan angkutan barang. Tabel 5.14 menunjukkan enam indikator sosial dan ekonomi untuk delapan kota di Cina Judul kolom: IL, TGOV, TIOV, TVPT, TVFT dan FB menunjukkan tingkat pendapatan total (dalam 10.000 yuan), total hasil pertanian (10 miliar yuan), total industri output (100 miliar yuan), total nilai transportasi (100 juta yuan), total nilai transportasi barang (100 juta yuan) dan anggaran keuangan (100 miliar yuan), masing-masing. Gunakan metode analisis komponen utama (PCA) untuk menentukan peringkat tingkat perkembangan kota-kota ini. Berikan alasan lengkap tentang pengurangan dimensi yang dilakukan dalam pemeringkatan kota.

Tabel 5.13 Kondisi kenaikan harga barang.

Barang, Bulan	Barang A	Barang B	Barang C	Barang D	Barang E
1	1	0	1	0	1
2	0	1	0	1	0
3	1	0	1	0	0
4	0	1	0	0	1
5	1	0	0	0	0
6	1	1	1	0	1
7	1	1	1	0	0
8	1	0	1	1	0

Tabel 5.14 Indikator sosial dan ekonomi dari delapan kota di Cina.

Kota	ILP	TGOV	TIOV	TVPT	TVFT	FB
Beijing	1249.90	1.84	2.00	2.03	4.56	2.79
Tianjin	910.17	1.59	2.26	0.33	2.63	1.13
Shijiazhuang	875.40	2.92	0.69	0.29	0.19	0.71
Taiyuan	299.92	0.24	0.27	0.19	1.19	0.39

Hohhot	207.78	0.37	0.08	0.24	0.26	0.14
Shenyang	677.08	1.30	0.58	0.78	1.54	0.90
Dalian	545.31	1.88	0.84	1.08	1.92	0.76
Changchun	691.23	1.85	0.60	0.48	0.95	0.48

5. Untuk mengetahui apakah mahasiswa menderita penyakit hiperlipemia saat semester dimulai, diperlukan pemeriksaan fisik untuk mendeteksi trigliserida, Metrik yang diuji antara lain kolesterol, high-density lipoprotein dan low-density lipoprotein, dll. Karena human error, ada data yang hilang dari daftar pemeriksaan fisik. Berdasarkan data yang diberikan pada Tabel 5.15, pilih algoritma *Machine learning* yang sesuai untuk membangun model klasifikasi untuk memprediksi apakah seorang siswa menderita hiperlipemia atau tidak.

Tabel 5.15 Data pemeriksaan fisik dan status hiperlipemia.

Id	Trigliserida	Total kolesterol	Lipoprotein Kepadatan Tinggi	Lipoprotein Kepadatan Rendah	Hiperlipemia atau tidak
1	1.05	3.28	1.35	1.8	No
2	1.43	5.5	1.66	3.69	No
3	1.16	3.97	1.27	2.55	Yes
4	6.8	5.95	0.97	2.87	Yes
5	3.06	5.25	0.9	3.81	Yes
6	1.18	5.88	1.77	3.87	No
7	2.53	6.45	1.43	4.18	Yes
8	1.6	5.3	1.27	3.74	No
9	3.02	4.95	0.95	3.53	Yes
10	2.57	6.61	1.56	4.27	Yes

Tabel 5.16 Status pertumbuhan tanaman di lingkungan yang berbeda.

Nomor Tanaman	Kelembaban (% RH)	Suhu (OC)	Cahaya (intensitas cahaya)	CO2 (ppm)	Status Tumbuh
1	64.14	23.04	1.87	817	0
2	65.97	23.11	36.99	702	0
3	65.3	21.01	6.36	803	1
4	71.75	20.58	125.82	822	1
5	63.6	21.53	94.23	772	0
6	64.51	21.47	58.47	888	1

7	65.01	22.03	3.19	719	0
8	66.98	23.66	14.23	754	0
9	67.73	21.61	16.49	760	1
10	67.04	20	6.68	890	1
11	67.79	19.86	121.1	842	1
12	65.6	20.82	15.92	694	0
13	64.92	23.06	86.38	849	1
14	65.8	23.67	86.56	752	0
15	64.91	22.48	73.74	806	1

6. Sejumlah faktor telah disarankan untuk menilai kondisi pertumbuhan tanaman rumah kaca, seperti kelembaban, suhu, cahaya, kandungan CO2, dll. Berdasarkan data pada Tabel 5.16 untuk taman dengan 15 tanaman, evaluasi kondisi pertumbuhan dari 15 tanaman tersebut. Kami menggunakan "1" untuk menunjukkan kondisi tumbuh dengan baik dan "0" sebaliknya. Dengan menganalisis data tersebut, diterapkan model SVM untuk mengestimasi kondisi pertumbuhan tanaman di lingkungan yang bercirikan vektor {68.08, 20.27, 59.25, 775}.
7. Minum secukupnya memang baik untuk melancarkan peredaran darah, tetapi minum berlebihan dapat membahayakan kesehatan. Alkohol telah terbukti memiliki efek kelumpuhan pada otak manusia. Tabel 5.17 menunjukkan properti yang relevan dari 10 merek bir. Anda diminta untuk menggunakan dataset ini untuk menganalisis efek kelumpuhannya. Tentukan fungsi tujuan menggunakan salah satu model *Machine learning* tanpa pengawasan yang telah Anda pelajari dalam bab ini. Tujuannya adalah untuk mengklasifikasikan tingkat kerusakan dari merek-merek bir tersebut. Anda mungkin memerlukan beberapa pengetahuan medis untuk mendefinisikan model kelumpuhan ini dengan benar. Hubungi pakar kesehatan atau konsultasikan dengan beberapa otoritas, atau cari bantuan dari Wikipedia atau pencarian Google.

Tabel 5.17 Sifat deskriptif 10 merek bir.

Bir merek	**Panas (Joule)**	**Natrium (mmol/L)**	**Alkohol (derajat)**	**Harga (Rp)**
Budweiser	144	19	4.7	0.43
ionenbra	157	15	4.9	0.48
Kronenso	170	7	5.2	0.73
Oldmin	145	23	4.6	0.26
Budweiser	113	6	3.7	0.44
warna	140	16	4.6	0.44

Coorslic	102	15	4.1	0.46
Kirin	149	6	5	0.79
Olympia	72	6	2.9	0.46
Schite	97	7	4.2	0.47

8. Di Bagian 5.5, Anda telah mempelajari beberapa kriteria pemilihan algoritme *Machine learning* (ML) agar sesuai dengan permintaan aplikasi. Tujuannya adalah untuk membuat prediksi, peramalan atau klasifikasi kumpulan *Big data* lebih akurat atau efisien. Jawablah secara singkat lima pertanyaan berikut tentang pemilihan algoritma ML. Buktikan jawaban Anda dengan alasan atau dengan contoh kasus:
   a) Kompilasi tabel untuk menyarankan metrik kinerja utama yang harus diterapkan untuk setiap algoritme ML yang telah Anda pelajari sejauh ini di Bab 4 dan 5. Jelaskan entri tabel Anda dengan alasan dengan menerapkan contoh yang Anda ketahui.
   b) Tentukan skor kinerja metode *Machine learning*, seperti yang dikirim pada Gambar 5.13.
   c) Bedakan antara kasus over-fitting dan under-fitting dalam memilih model prediksi TPPU.
   d) Tentukan fungsi kerugian yang berbeda yang digunakan dalam memplot kurva tersebut pada Gambar 5.19.
   e) Menjelaskan konsep kumpulan data umum, validasi silang, dan siklus bootstrap dalam pemilihan algoritma ML.

BAB 6
DEEP LEARNING DENGAN JARINGAN SYARAF TIRUAN

6.1 PENDAHULUAN

Deep learning mensimulasikan operasi dalam lapisan jaringan saraf tiruan. Ini banyak digunakan untuk mengekstrak dan mempelajari fitur dari data. Jaringan saraf multilayer mencakup satu lapisan input, satu lapisan output dan beberapa lapisan tersembunyi. Kekuatan koneksi antar neuron disesuaikan dalam proses pembelajaran. Arsitektur *Deep learning* yang umum diperkenalkan di bagian selanjutnya, termasuk ANN dasar, jaringan saraf convolutional, jaringan kepercayaan mendalam dan jaringan saraf berulang, dll.

***Deep learning* Meniru Indra Manusia**

Pada bulan Maret 2016, AlphaGo menghasilkan publisitas tinggi dan perdebatan tentang persaingan manusia-mesin dalam intelijen. Setelah lima putaran kompetisi, akhirnya komputer mengalahkan Sedol Lee, pemain Go kelas dunia. Go adalah permainan yang rumit, karena diketahui bahwa hanya kecerdasan manusia yang dapat mengatasi kerumitan yang terlibat. Tidak ada atlet profesional yang pernah dikalahkan oleh perangkat lunak Go sebelumnya. Dua puluh tahun yang lalu, komputer Deep Blue menggunakan metode algoritma pencarian untuk mengalahkan master catur internasional Garry Kasparov dalam pertandingan catur internasional.

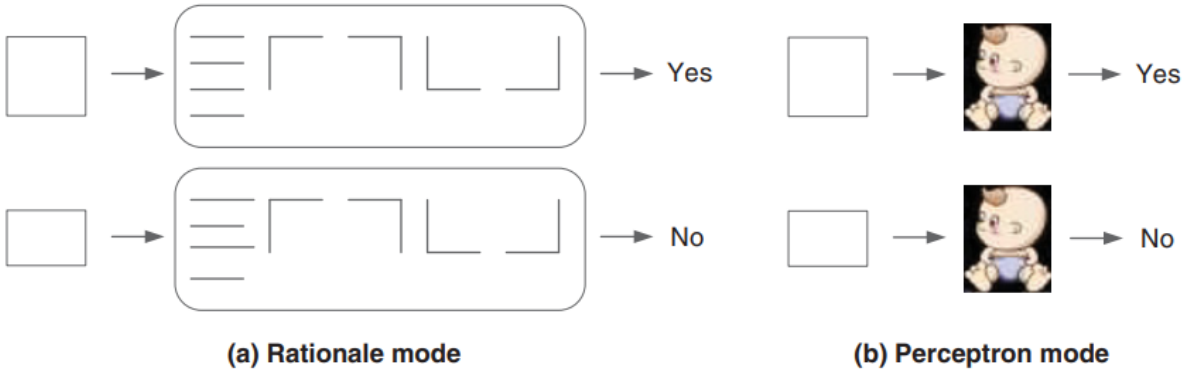
Bermain Go lebih sulit daripada bermain game catur lainnya. AlphaGo yang diimplementasikan komputer menggabungkan *Deep learning* dan algoritma pencarian pohon untuk membuat keputusan cerdas dalam menempatkan pion. Untuk pertama kalinya, Google AlphaGo telah mencapai level yang hampir manusiawi dalam memainkan Go, secara cerdas tanpa gangguan emosi. Ini merupakan langkah besar dalam memajukan kecerdasan buatan untuk mencapai tingkat kinerja manusia.

Kemajuan lain yang dilaporkan dibuat di Google Brain Project. Pada Juni 2012, puluhan juta gambar acak dari YouTube dikenali oleh platform komputer yang dibuat dengan lebih dari 16.000 inti CPU di Google. Mereka menggunakan model pelatihan melalui jaringan saraf dalam yang dibangun dengan 1 miliar neuron buatan. Sistem model ini mengidentifikasi fitur dasar gambar, mempelajari cara menyusun fitur ini, dan secara otomatis mengidentifikasi gambar kucing. Selama pelatihan, sistem tidak memperoleh informasi "Ini adalah kucing", tetapi memahami konsep "kucing" itu sendiri. Pemimpin proyek Andrew Ng mengatakan: "Kami secara langsung memasukkan *Big data* ke dalam sistem buatan dan sistem belajar dari data, secara otomatis."

Dari kemenangan AlphaGo hingga kesuksesan Google Brain Project, sepertinya *deep learning* memiliki kemampuan belajar mandiri. Pertanyaan yang bermakna untuk diajukan adalah: Bagaimana *Deep learning* dapat bersaing dengan manusia melalui pembelajaran mandiri? Jika kita ingin menilai apakah suatu segi empat itu bujur sangkar atau tidak, pendekatan

analitis rasional adalah dengan mencari ciri-ciri persegi, seperti panjang keempat sisinya yang sama dan empat sudut sudut 90 derajat. Hal ini membutuhkan pemahaman tentang konsep sudut siku-siku dan panjang tampak samping, seperti yang ditunjukkan pada Gambar 6.1.

Jika kita menunjukkan gambar persegi kepada seorang anak laki-laki dan mengatakan kepadanya bahwa ini adalah persegi, dia akan mengidentifikasi persegi secara akurat setelah beberapa kali mencoba. Metode mengidentifikasi persegi secara rasional mirip dengan metode di mana identifikasi fitur dirancang secara artifisial. Tetapi cara seorang anak mengidentifikasi persegi mengikuti metode persepsi. Secara rasional, mudah untuk menggambarkan dan menyadari masalah ini dengan komputer. Namun, mudah bagi manusia untuk memahami banyak masalah dalam kenyataan, tetapi sulit bagi komputer untuk memahami dan memecahkan masalah ini secara rasional.



Gambar 6.1 Dasar pemikiran versus mode perceptron untuk mengenali objek persegi.

Misalnya, jika kita ingin mengidentifikasi orang melalui foto dengan metode rasional melalui komputer, maka perlu menentukan fitur wajah manusia mana yang dapat digunakan untuk identifikasi, seperti hidung, mata, alis dan mulut, dll. Tentu saja, sulit untuk memilih fitur yang tepat yang dapat membedakan orang secara akurat. Pengaruh besar akan dibawa oleh faktor-faktor seperti perubahan cahaya dalam foto, perbedaan sudut pengambilan gambar, dan apakah kacamata sudah dipakai atau tidak.

Mari kita pertimbangkan pengakuan seorang anak terhadap seseorang sebagai contoh. Anak tidak perlu mencari ciri-ciri orang untuk dikenali. Tetapi dia dapat secara akurat mengidentifikasi orang tersebut setelah melihat orang itu atau fotonya beberapa kali. Perubahan cahaya dalam foto, perbedaan sudut pengambilan gambar, dan apakah kacamata dipakai tidak akan mempengaruhi identifikasi. Kita mungkin menafsirkan bahwa seorang anak mengenali seseorang melalui kesan. Ada semacam pemetaan antara foto masukan dan nama keluaran dalam memori manusia.

Dengan berkembangnya aplikasi komputer, orang semakin menyadari bahwa metode rasional atau analitis akan menjadi tidak efisien atau tidak mungkin ketika memecahkan banyak masalah di dunia nyata. Mekanisme bahwa orang melakukan sesuatu secara intuitif tampaknya

cukup efisien di depan ilmu pengetahuan dan teknologi modern. Metode intuitif atau metode mengikuti kata hati ini secara sederhana dapat diartikan sebagai menetapkan pemetaan tertentu antara input dan output. Tetapi masih belum diketahui bagi kita bagaimana otak manusia mewujudkan penyandian, pemrosesan, dan penyimpanan informasi dengan 100 miliar neuron.

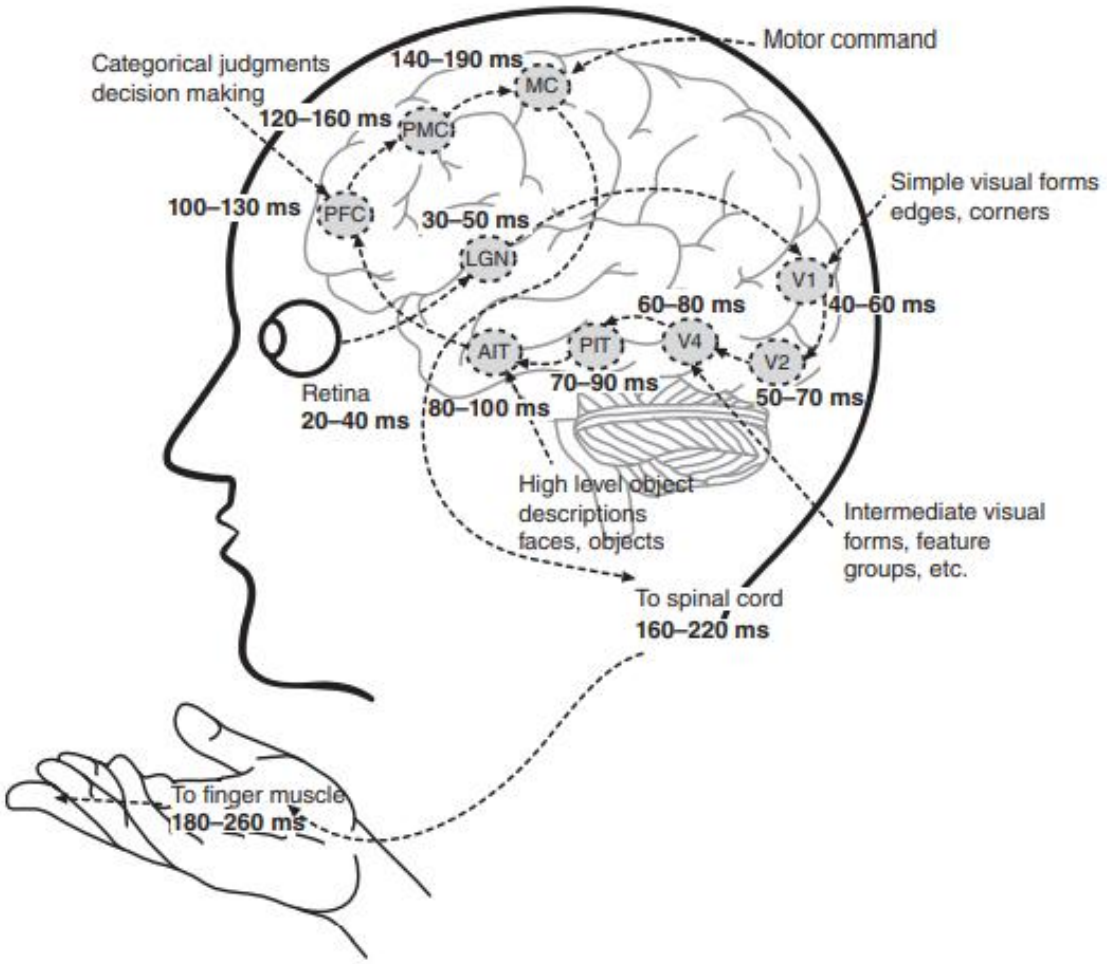
Kontribusi utama dari David Hubel dan Torsten Wiesel (pemenang Hadiah Nobel Kedokteran 1981) adalah bahwa mereka menemukan bahwa pemrosesan informasi dari sistem visual (yaitu korteks visual) adalah hierarkis, seperti yang ditunjukkan pada Gambar 6.2. Di Universitas John Hopkins, pada tahun 1958, mereka mempelajari korespondensi antara area pupil dan neuron di korteks serebral. Melalui banyak percobaan, telah dibuktikan bahwa ada semacam korespondensi antara rangsangan yang diterima oleh pupil dan neuron visual yang berbeda yang terletak di korteks serebral posterior. Mereka menemukan sejenis neuron yang mereka namakan sebagai "sel selektif orientasi". Ketika pupil menangkap tepi suatu objek, dan jika tepi ini menunjuk ke arah tertentu, neuron yang sesuai akan aktif.

Pemrosesan informasi dari sistem visual manusia ditafsirkan sebagai mengekstraksi fitur tepi dari area V1, dan mengekstraksi fitur bentuk atau beberapa bagian penyusun suatu tujuan dari area V2. Kemudian tingkat pemahaman yang lebih tinggi tercapai. Dari fitur tingkat rendah hingga tinggi, tingkat abstraksi meningkat. Kombinasi fitur tingkat rendah berfungsi sebagai masukan ke tingkat yang lebih tinggi. Dengan demikian, fitur tingkat yang lebih tinggi mengungkapkan lebih banyak semantik. Ketika abstraksi meningkat, kebingungan konteks menjadi lebih sedikit, yang baik untuk tujuan klasifikasi atau identifikasi.

Neuron Biologis versus Neuron Buatan

Otak manusia memiliki struktur yang sangat rumit. Tetapi unit konstitusionalnya adalah neuron, yang menghasilkan output (kegembiraan) per input. Pemrosesan informasi hierarkis otak manusia diwujudkan melalui banyak neuron yang saling terhubung. Pada neuron biologis yang dimodelkan pada Gambar 6.3(a), ujung kiri dendrit dihubungkan ke sitomembran sebagai input, sedangkan ujung kanan akson adalah output. Apa yang terutama dikeluarkan oleh neuron adalah impuls listrik. Ada banyak cabang dendrit dan akson, dengan ujung akson sering terhubung ke dendrit neuron lain.

Neuron memperoleh input dari neuron lapisan atas, menghasilkan output dan mengirimkannya ke neuron di lapisan berikutnya. Jika otak manusia dapat disimulasikan, neuron harus disimulasikan terlebih dahulu. Gambar 6.3(b) menunjukkan struktur neuron buatan. Masukan ke neuron buatan semuanya dari sinyal perangsang eksternal, dilambangkan dengan xi untuk i = 1, 2, ….n. Neuron buatan menghitung jumlah tertimbang dari sinyal input, di mana bobot dilambangkan sebagai wi untuk i = 1, 2, …, n. Sebuah fungsi nonlinier diterapkan untuk menghasilkan sinyal keluaran y.



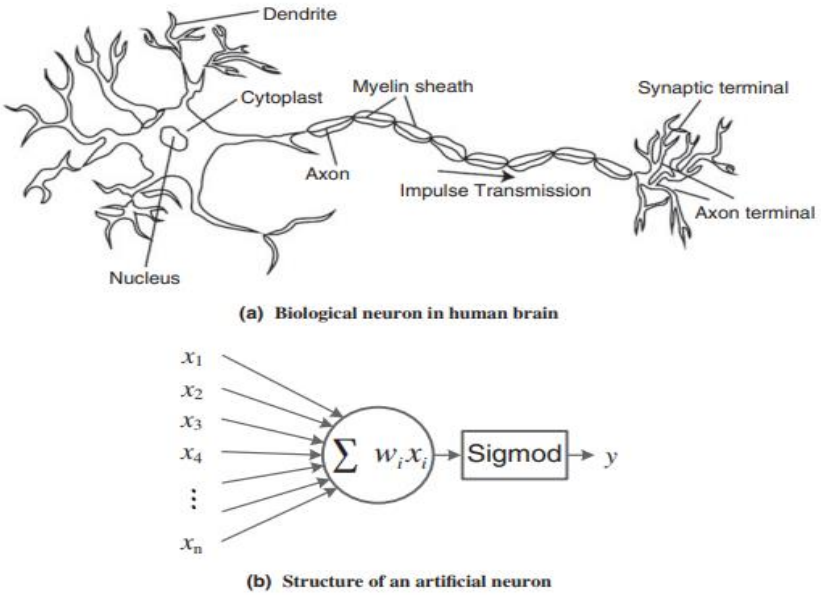
Gambar 6.2 Hirarki aliran sinyal korteks visual manusia di otak, retina dan jari.

Fungsi sigmoid nonlinier berikut disarankan untuk memodelkan operasi neuron buatan, seperti yang ditunjukkan pada Gambar 6.3(b):

$$y = Sig\ \mathrm{mod}(x) = \frac{1}{1 + e^{-x}} \tag{6.1}$$

di mana x mewakili jumlah input berbobot.

Pada tahun 2006, sebuah artikel oleh Hinton in Science mengilhami era baru *Deep learning*. Jaringan saraf tiruan dengan beberapa lapisan tersembunyi belajar mandiri disarankan dalam artikel itu. Setiap lapisan tersembunyi mencakup beberapa neuron. Output dari layer sebelumnya dijadikan sebagai input dari layer berikutnya. Struktur jaringan saraf yang dalam seperti itu pertama-tama mengadopsi neuron buatan untuk mensimulasikan neuron biologis di otak manusia. Kemudian, struktur pembelajaran berlapis dari jaringan dalam mensimulasikan struktur hierarkis pemrosesan informasi oleh otak manusia.



Gambar 6.3 Diagram skema neuron biologis versus neuron buatan.

Kesimpulan utama dalam artikel tersebut adalah:

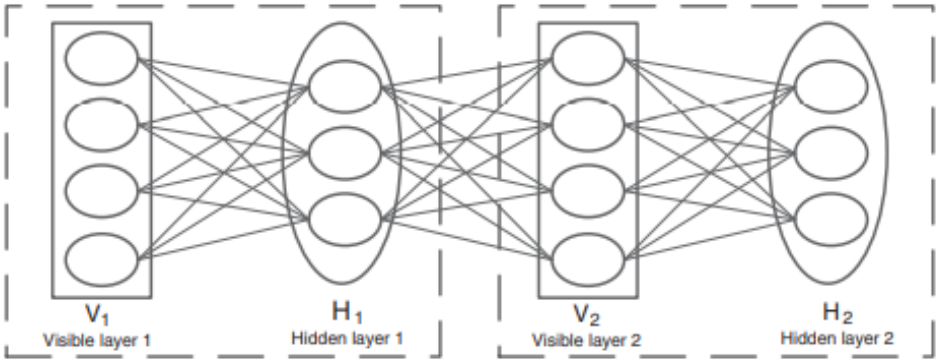
1) Kemampuan belajar untuk mendapatkan fitur oleh jaringan saraf tiruan yang dalam dengan banyak lapisan tersembunyi kuat. Terlebih lagi, fitur yang diperoleh dengan pembelajaran progresif di beberapa lapisan dapat mewakili data secara akurat;
2) Metode pra-pelatihan layer-wise memecahkan kesulitan dalam melatih JST. Sementara itu, pembelajaran tanpa pengawasan diadopsi selama pra-pelatihan berlapis.

Seperti yang ditunjukkan pada Gambar 6.4, daerah V1 dan H1 menyusun lapisan pertama dengan V1 sebagai input dan H1 sebagai output, di mana data di V1 berasal dari data asli. V2 dan H2 menyusun lapisan kedua. Masukan V2 berasal dari keluaran H1, sedangkan H2 merupakan keluaran pada lapisan kedua.

Singkatnya, keberhasilan *Deep learning* tergantung pada faktor-faktor berikut: peningkatan algoritma, mewujudkan ekstraksi fitur berlapis, simulasi kemampuan otak manusia dalam belajar, dan simulasi struktur hierarki otak manusia selama pemrosesan informasi. Selain itu, ada dua alasan eksternal untuk membuat *deep learning* populer. Salah satunya adalah adopsi GPU dan peningkatan kemampuan computing komputer untuk mendukung pelatihan jaringan saraf dalam skala besar. Lainnya adalah kemudahan memperoleh data pelatihan dalam jumlah besar di era *Big data* dengan penginderaan IoT.

Meskipun manusia dapat memperoleh nilai abstrak dari *Big data* dengan kekuatan *Deep learning*, masih ada kesenjangan yang signifikan antara AI saat ini dan simulasi otak manusia dengan fidelitas tinggi. Misalnya, hanya diperlukan beberapa kali untuk mengajari anak mengenali seseorang, dan anak dapat beradaptasi dengan efek cahaya dan perubahan

penampilan apa pun. Namun, jika kita ingin komputer mengenali seseorang, sejumlah besar gambar yang disimpan diperlukan untuk pembelajaran, dan sulit bagi komputer untuk beradaptasi dengan perubahan efek cahaya, pakaian, kacamata, atau faktor lainnya.



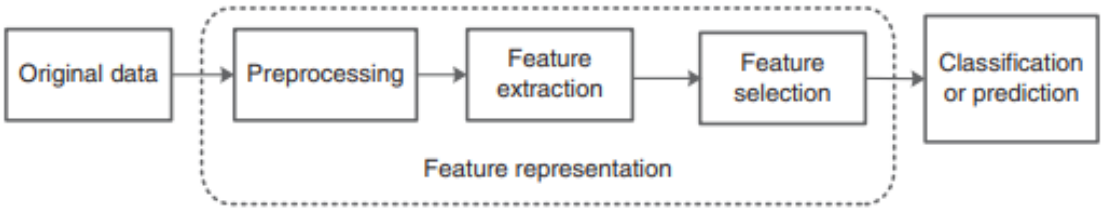
Gambar 6.4 Konsep *deep learning* dengan ANN yang memiliki dua lapisan tersembunyi.

***Deep learning* versus Pembelajaran Dangkal**

Jaringan saraf dalam (DNN) adalah jaringan saraf tiruan (JST) dengan beberapa lapisan unit tersembunyi antara lapisan input dan output. DNN dapat memodelkan hubungan non-linier yang lebih kompleks daripada jaringan saraf dangkal. Untuk mengenali pola sederhana, alat klasifikasi dasar berdasarkan pembelajaran dangkal, misalnya pohon keputusan, SVM, dll, sudah cukup baik. Dengan bertambahnya jumlah fitur input, ANN mulai menunjukkan performa superiornya.

Selanjutnya, ketika pola menjadi lebih kompleks, jaringan saraf dangkal menjadi tidak dapat digunakan, karena jumlah node yang dibutuhkan di setiap lapisan tumbuh secara eksponensial dengan meningkatnya jumlah kemungkinan pola. Kemudian pelatihan menjadi mahal dan akurasi mulai menurun. Jadi, ketika suatu pola menjadi sangat kompleks, pembelajaran yang mendalam sangat dibutuhkan. Memberi pengenalan wajah sebagai contoh, pembelajaran dangkal atau jaring saraf dangkal tidak layak, jadi satu-satunya pilihan praktis adalah *Deep learning* melalui jaring dalam. Alasan penting untuk *Deep learning* untuk mengungguli semua pesaing mereka adalah bahwa kapasitas computing ditingkatkan secara ekstensif untuk membuat proses pelatihan jauh lebih cepat daripada sebelumnya.

Pada setiap lapisan algoritme *Deep learning*, sinyal diubah oleh unit pemrosesan, seperti neuron buatan, yang parameternya "dipelajari" melalui pelatihan. Tidak ada batas atas kedalaman yang disepakati secara universal yang membagi pembelajaran dangkal dengan jaring saraf dangkal dari pembelajaran dalam, tetapi sebagian besar peneliti di lapangan setuju bahwa pembelajaran dalam memiliki beberapa lapisan nonlinier (CAP > 2) dan Schmidhuber menganggap CAP > 10 sebagai pembelajaran yang sangat dalam. . [https://en.wikipedia.org/wiki/Deep_learning]



Gambar 6.5 Proses klasifikasi atau prediksi berdasarkan representasi fitur.

Pembelajaran representasi menawarkan cara untuk merepresentasikan data secara lebih akurat melalui representasi fitur, seperti yang diilustrasikan pada Gambar 6.5. Ini mengadopsi fitur data yang rumit untuk mewakili data input asli. Ini diterapkan di banyak bidang *Machine learning*, seperti pengenalan gambar, pengenalan suara, pemahaman bahasa alami, prakiraan cuaca dan rekomendasi konten, dll. Proses untuk memecahkan masalah klasifikasi atau prediksi ditunjukkan pada Gambar 6.5.

Setelah mendapatkan data asli, kami melakukan preprocessing terlebih dahulu, dilanjutkan dengan ekstraksi fitur dan seleksi fitur. Dengan fitur-fitur ini, klasifikasi atau prediksi dapat dicapai. Kombinasi antara preprocessing data, ekstraksi fitur dan seleksi fitur disebut dengan representasi fitur. Menemukan representasi fitur yang baik memainkan peran penting dalam memperoleh akurasi yang tinggi dari klasifikasi akhir atau prediksi. Pemilihan fitur yang dibuat dengan tangan membutuhkan pengetahuan profesional, dan kemanjuran fitur yang dipilih dipertanyakan. Dengan pembelajaran fitur otonom, *Deep learning* memecahkan masalah ini secara efisien.

Konsep *deep learning* berasal dari pengembangan JST. Struktur *Deep learning* sering kali mencakup beberapa lapisan tersembunyi, dan representasi pembelajaran dari data asli dilakukan lapis demi lapis. Melalui kombinasi fitur tingkat rendah, *Deep learning* membentuk representasi atribut yang lebih abstrak atau fitur tingkat tinggi. Perbedaan utama antara *Deep learning* dan metode *Machine learning* lainnya adalah kemampuan "pembelajaran fitur". Ini dapat diartikan sebagai situasi di mana “deep model” adalah sebuah metode dan “feature learning” adalah tujuannya.

Deep learning menyesuaikan model pembelajaran yang memiliki banyak lapisan tersembunyi melalui pelatihan dan pembelajaran dengan banyak data, kemudian memperoleh representasi berbasis lapisan dari data asli. Dengan demikian ia mempelajari representasi fitur data yang efektif, dan akhirnya meningkatkan akurasi dalam klasifikasi atau prediksi.

Deep learning berbeda dari pembelajaran dangkal dalam empat aspek:

a) Ini menekankan kedalaman struktur JST. Dibandingkan dengan pembelajaran dangkal konvensional, lebih banyak lapisan tersembunyi digunakan dalam *Deep learning*.
b) Pentingnya pembelajaran fitur disorot. Dengan menggunakan transformasi fitur layer-wise, fitur data dari ruang asli diwakili oleh fitur di ruang fitur baru. Dengan metode ini, klasifikasi atau prediksi menjadi lebih mudah dan akurat.

c) *Deep learning* berasal dari JST. Namun, model pelatihan mereka berbeda. Pelatihan lapisan-bijaksana diadopsi dalam *Deep learning*, yang memecahkan masalah gradien menghilang.
d) Banyak data yang digunakan untuk mempelajari fitur-fitur dalam *Deep learning*, yang tidak harus dalam pembelajaran dangkal.

6.2 JARINGAN SYARAF TIRUAN (JST)

Jaringan syaraf tiruan (JST) adalah sejenis model matematika abstrak yang bertujuan untuk mencerminkan struktur dan fungsi otak manusia. Ini banyak digunakan di banyak bidang seperti pengenalan pola, pemrosesan gambar, kontrol cerdas, optimasi kombinatorial, prediksi dan manajemen keuangan, komunikasi, robotika, dan sistem pakar. Ada banyak kesamaan antara JST dan jaringan saraf biologis di otak manusia. JST terdiri dari sekelompok unit input/output yang terhubung. Setiap koneksi dinyatakan sebagai tepi berbobot. Pada tahap pembelajaran, kami menyesuaikan bobot ini berdasarkan kesenjangan antara keluaran yang diprediksi dan data uji berlabel.

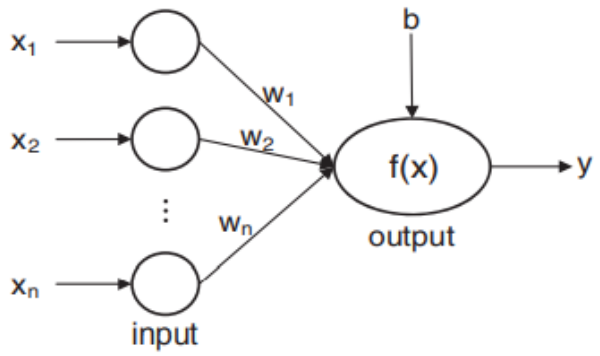
Jaringan Syaraf Tiruan Lapisan Tunggal

Perceptron adalah JST paling sederhana, yang mencakup lapisan input dan lapisan output tanpa lapisan tersembunyi. Node lapisan input digunakan untuk menerima data, sedangkan node lapisan output menghasilkan data output. Bagan struktur perceptron ditunjukkan pada Gambar 6.6. Karena kita mensimulasikan perceptron sebagai sistem saraf manusia, node input sesuai dengan input neuron dan output node sesuai dengan pengambilan keputusan neuron, sedangkan parameter bobot sesuai dengan kekuatan koneksi antar neuron. Dengan terus-menerus merangsang neuron, otak manusia dapat memperoleh pengetahuan yang tidak diketahui. Fungsi aktivasi, f(x), digunakan untuk meniru stimulasi neuron di otak manusia. Ini adalah bagaimana jaringan saraf tiruan mendapatkan namanya.

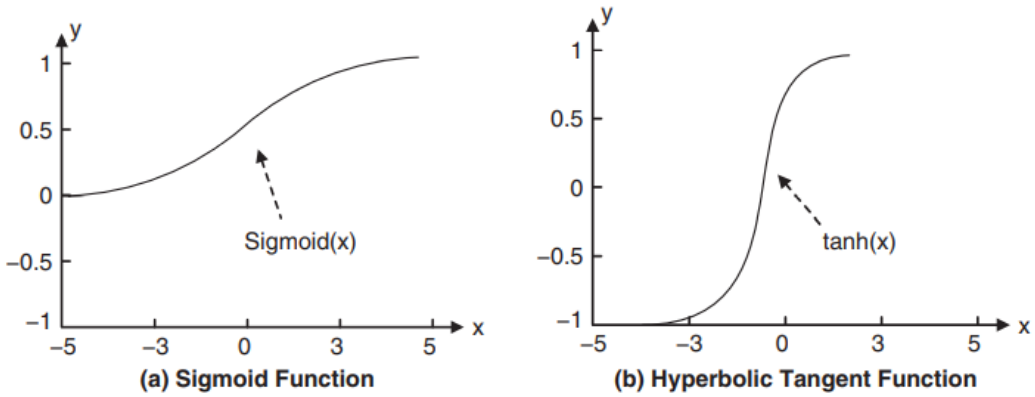
Dari perspektif matematika, setiap item input sesuai dengan atribut objek, sedangkan bobot menunjukkan derajat di mana atribut tersebut mencerminkan objek. Dikalikan dengan derajat deviasi, diperoleh input x. Kemudian, outputnya dihitung

$$x = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \rightarrow y = f(x) \qquad (6.2)$$

Biasanya, kita belum tentu mendapatkan hasil yang ideal. Dalam arti sempit, harus diadopsi untuk mewakili hasil keluaran perceptron. Persamaan modelnya adalah = f (w x), di mana, w dan x adalah vektor berdimensi n. Biasanya, fungsi sigmoid $(\text{sig mod}\,(x) = \frac{1}{1+e^{-x}})$ atau fungsi tangen hiperbolik $(\text{fungsi}(x) = \frac{e_x+e_{-x}}{e_x-e_{-x}})$, digunakan untuk f(x), seperti yang ditunjukkan pada Gambar 6.7.



Gambar 6.6 Diagram konseptual mesin perceptron.



Gambar 6.7 Fungsi aktivasi umum untuk perceptron.

Jika kita mengharapkan hasil yang baik, diperlukan bobot yang sesuai. Namun, kami tidak dapat mengetahui nilai mana yang harus ditetapkan untuk setiap bobot terlebih dahulu. Oleh karena itu, nilai bobot harus disesuaikan secara dinamis selama latihan. Persamaan pembaruan berat ditunjukkan sebagai:

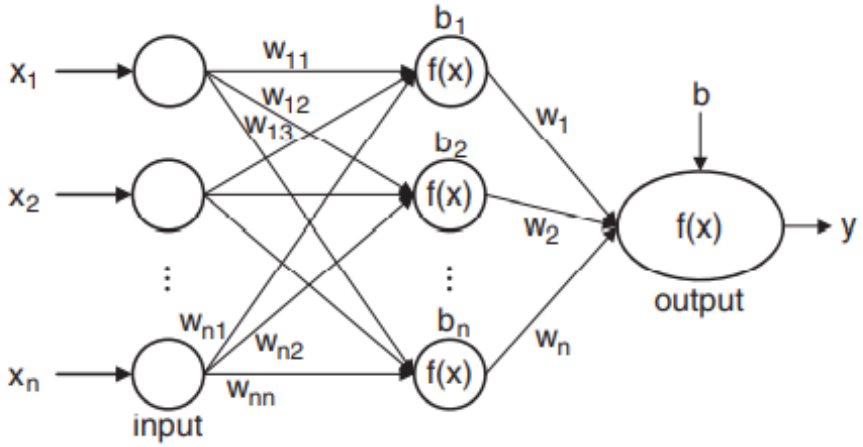
$$w_j^{(k+1)} = w_j^{(k)} + \lambda\left(y_i - \hat{y}_i^{(k)}\right)x_{ij} \quad j = 1, 2, \cdots n \tag{6.3}$$

dimana, $w_j^{(k)}$ adalah nilai bobot dari input node j setelah iterasi sebanyak k kali, λ dikenal sebagai learning rate, dan x_{ij} adalah nilai input dari node j dalam sampel data training ke-i.

Tingkat pembelajaran berada dalam interval [0, 1]. Jika λ mendekati 0, nilai bobot baru terutama dipengaruhi oleh nilai bobot lama. Tingkat pembelajarannya lambat, tetapi nilai bobot yang optimal dapat ditemukan dengan mudah. Jika λ mendekati 1, nilai bobot baru terutama dipengaruhi oleh jumlah penyesuaian saat ini. Tingkat pembelajarannya cepat, tetapi mungkin melewatkan nilai bobot optimal. Oleh karena itu, dalam beberapa keadaan, nilai λ akan lebih besar pada beberapa iterasi sebelumnya, tetapi akan berkurang secara bertahap pada iterasi berikutnya.

Jaringan Syaraf Tiruan Multilayer

Perceptron merupakan neural network dengan dua lapisan, yaitu input layer dan output player. Sebagai perbandingan, jaringan saraf tiruan multilayer terdiri dari satu lapisan input, satu atau lebih lapisan tersembunyi dan satu lapisan keluaran, seperti yang ditunjukkan pada Gambar 6.8.



Gambar 6.8 Struktur jaringan syaraf tiruan dua lapis.

Unit utama JST adalah neuron. Ada tiga elemen dasar neuron:

1) sekelompok koneksi, sesuai dengan sinapsis dari neuron biologis. Kekuatan sambungan dinyatakan dengan nilai bobot setiap sambungan. Nilai bobot mewakili aktivasi jika bernilai positif, sedangkan bobot mewakili penekanan saat bernilai negatif. Persamaan matematikanya adalah

$$\begin{cases} w = (w_1, w_2, \cdots, w_n) \\ w_i = (w_{i1}, w_{i2}, \cdots, w_{in}) \quad i = 1, 2, \cdots n \end{cases} \tag{6.4}$$

2) satu unit penjumlahan. Ini digunakan untuk menghitung jumlah bobot (kombinasi linier) untuk setiap sinyal input, dan umumnya bersama-sama dengan offset atau ambang batas. Persamaan matematisnya adalah

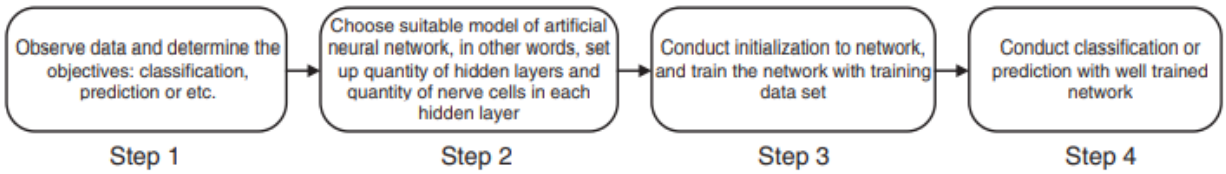
$$\begin{cases} \mu_k = \sum_{j=1}^{n} w_{kj} x_j \\ v_k = \mu_k + b_k \end{cases} \tag{6.5}$$

3) satu fungsi aktivasi nonlinier. Ini memainkan peran pemetaan nonlinier dan membatasi amplitudo keluaran neuron dalam kisaran tertentu [sering (0, 1) atau (-1, 1)]. persamaan matematikanya adalah

$$y_k = f(v_k) \tag{6.6}$$

di mana f (·) adalah fungsi aktivasi.

Tidak ada regulasi yang seragam untuk jumlah lapisan tersembunyi dalam jaringan syaraf tiruan; jumlah neuron pada input, output dan setiap hidden layer, serta cara memilih fungsi aktivasi neuron pada setiap layer. Juga tidak ada standar untuk beberapa kasus tertentu, yang perlu dipilih secara independen atau dipilih sesuai pengalaman pribadi. Oleh karena itu, ada sifat heuristik tertentu pada pemilihan jaringan, dan inilah mengapa jaringan saraf tiruan dianggap sebagai algoritma heuristik. Empat langkah untuk menghasilkan model JST diberikan di bawah ini pada Gambar 6.9:



Gambar 6.9 Langkah-langkah umum untuk memodelkan jaringan saraf tiruan

Propagasi Maju dan Propagasi Mundur di JST

Seperti model perceptron, untuk jaringan multilayer, cara mendapatkan sekumpulan bobot yang sesuai agar jaringan memiliki fungsi tertentu dan nilai aplikasi praktis adalah penting. Jaringan saraf tiruan memecahkan masalah ini dengan algoritma propagasi balik. Sebelum memperkenalkan algoritma propagasi mundur, kita perlu mengetahui bagaimana jaringan menyebar ke depan.

Propagasi Maju

Perambatan maju dimulai dari lapisan input menuju lapisan tersembunyi. Untuk unit tersembunyi i, inputnya adalah h_i^k, yang merupakan input dari unit tersembunyi i pada lapisan k, sedangkan b_i^k adalah offset dari unit tersembunyi i pada lapisan k. Keadaan keluaran yang sesuai adalah:

$$h_i^k = \sum_{j=1}^{n} w_{ij} x_j + b_i^k \rightarrow H_i^k = f\left(h_i^k\right) = f\left(\sum_{j=1}^{n} w_{ij} x_j + b_i\right) \tag{6.7}$$

Untuk memudahkan, kita sering membiarkan $x_0 = b$, $w_{i0} = 1$. Maka persamaan perambatan maju dari unit tersembunyi lapisan k ke lapisan k + 1 adalah:

$$\begin{cases} h_i^{k+1} = \sum_{j=1}^{m_k} w_{ij}^k h_j^k \\ H_i^{k+1} = f\left(h_i^{k+1}\right) = f\left(\sum_{j=1}^{n} w_{ij}^k h_j^k\right) \end{cases} \quad i = 1, 2, \cdots, m_{k+1} \tag{6.8}$$

di mana m_k adalah jumlah neuron dalam unit tersembunyi dari lapisan k, dan w_{ij}^k adalah matriks vektor bobot dari lapisan k ke lapisan k+1. Output akhir diperoleh sebagai

$$O_i = f\left(\sum_{j=1}^{m_{M-1}} w_{ij}^{M-1} H_j^{M-1}\right) \quad i = 1, 2, \cdots, m_o \tag{6.9}$$

di mana mo adalah jumlah unit keluaran (bisa ada beberapa keluaran dalam jaringan saraf tiruan, tetapi umumnya satu keluaran akan diatur secara default), M adalah jumlah total lapisan jaringan saraf tiruan, dan O_i adalah singkatan dari nilai hasil unit keluaran i.

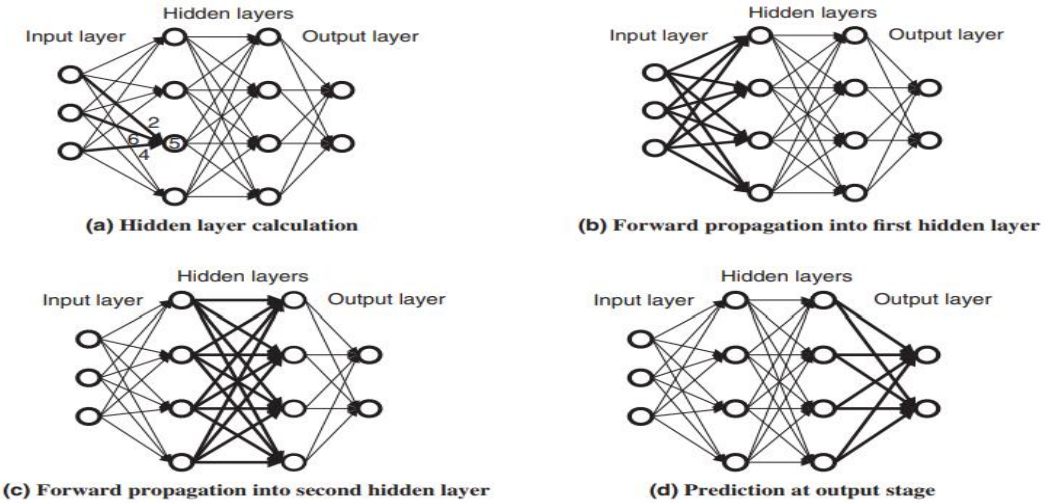
Contoh 6.1 Prediksi Output Berbasis Propagasi Maju di JST

Dalam JST, setiap set input dimodifikasi oleh bobot dan bias yang unik. Seperti yang ditunjukkan pada Gambar 6.10(a), ketika menghitung aktivasi neuron ketiga pada lapisan tersembunyi pertama, input pertama dimodifikasi dengan bobot 2, yang kedua dengan 6, yang ketiga dengan 4, dan kemudian bias 5 adalah ditambahkan di atas. Setiap aktivasi bersifat unik, karena setiap edge memiliki bobot yang unik dan setiap node memiliki bias yang unik.

Aktivasi sederhana ini akan membanjiri seluruh jaringan. Set input pertama dilewatkan ke lapisan tersembunyi pertama, seperti yang ditunjukkan pada Gambar 6.10(b). Aktivasi lapisan tersembunyi pertama diteruskan ke lapisan tersembunyi berikutnya, seperti yang ditunjukkan pada Gambar 6.10(c). Hingga mencapai lapisan keluaran, skor setiap simpul keluaran berdampak pada hasil klasifikasi, seperti yang ditunjukkan pada Gambar 6.10(d). Prosedur klasifikasi seperti itu dalam JST disebut propagasi maju, yang akan diulang untuk set input lainnya.

Propagasi Mundur

Persamaan 6.8 dan 6.9 menjelaskan bagaimana data input jaringan saraf disebarkan ke depan. Selanjutnya, kami akan memperkenalkan algoritma backpropagation dan cara memperbarui bobot wij melalui proses pembelajaran atau pelatihan. Kami berharap keluaran dari jaringan syaraf tiruan identik dengan nilai standar dari sampel pelatihan. Keluaran seperti ini dinamakan keluaran ideal. Sebenarnya, tidak mungkin untuk mencapai tujuan ini secara akurat. Kami hanya bisa berharap output aktual sedekat mungkin dengan output ideal.



Gambar 6.10 Prediksi keluaran berbasis Propagasi Maju dalam JST sederhana dengan dua lapisan tersembunyi dan empat neuron pada setiap lapisan.

Simbol i dan s mewakili nilai hasil unit keluaran i jika sampel pelatihan dilambangkan sebagai s. O_i^s adalah nilai hasil dari unit keluaran i jika sampel pelatihan adalah s. Jadi, masalah menemukan sekelompok bobot yang sesuai secara alami bermuara pada masalah bahwa E(W) mencapai minimum dengan mencari nilai W yang sesuai, seperti yang ditunjukkan di bawah ini:

$$E(W) = \frac{1}{2}\sum_{i,s}\left(T_i^s - O_i^s\right)^2 = \frac{1}{2}\sum_{i,s}\left(T_i^s - f\left(\sum_{j=1}^{m_{M-1}} w_{ij}^{M-1} H_j^{M-1}\right)\right)^2$$
$$\rightarrow \min E(W) \quad i = 1, 2, \cdots, m_o \tag{6.10}$$

Adapun setiap variabel w_{ij}^k , ini adalah fungsi nonlinier yang dapat diturunkan secara kontinu. Untuk menghitung minimum, kami biasanya mengadopsi metode penurunan paling curam. Sesuai metode ini, kami terus memperbarui bobot ke arah gradien negatif hingga kondisi yang ditetapkan oleh pelanggan terpenuhi. Arah gradien yang disebut adalah untuk mengerjakan turunan parsial fungsi:

Asumsikan bobotnya adalah $w_{ij}^{(k)}$ setelah pembaruan pada waktu ke-k. Jika $\nabla E(W) \neq 0$, maka berat baru pada waktu (k+1) dinyatakan dengan

$$\nabla E(W) = \frac{\partial E}{\partial w_{ij}^k} \rightarrow w_{ij}^{(k+1)} = w_{ij}^{(k)} - \eta \nabla E\left(w_{ij}^{(k)}\right) \tag{6.11}$$

di mana, adalah kecepatan belajar jaringan itu. Ini memainkan peran yang sama dengan kecepatan belajar di perceptron. Ketika $\nabla E(W) = 0$ atau $\nabla E(W) < \varepsilon$ (ε adalah kesalahan yang diizinkan), ia berhenti memperbarui.w_{ij}^k, saat ini, akan menjadi bobot akhir dari jaringan saraf tiruan. Proses dimana jaringan secara konstan menyesuaikan bobot disebut proses pembelajaran jaringan syaraf tiruan. Algoritma yang digunakan dalam proses pembelajaran ini disebut algoritma perambatan mundur jaringan.

Contoh 6.2 Koreksi Bobot/Bias Berbasis Propagasi Mundur pada JST

Keakuratan prediksi bergantung pada bobot dan bias. Tujuannya adalah untuk membuat output yang diprediksi sedekat mungkin dengan output aktual. Sama seperti metode *Machine learning* lainnya, kunci untuk meningkatkan akurasi adalah pelatihan. Biarkan y menunjukkan output dari propagasi maju, dan y* menunjukkan output yang benar. Biaya yang dilambangkan dengan (y y), adalah selisih antara y dan y* . Setelah proses pelatihan yang sangat lama, biayanya harus semakin berkurang.

Selama pelatihan, JST menyesuaikan bobot dan bias selangkah demi selangkah hingga keluaran yang diprediksi sesuai dengan keluaran sebenarnya. Untuk mencapainya, dilakukan tiga langkah:

1) Saat memperbarui bobot antara neuron pertama di lapisan output dan neuron di lapisan tersembunyi kedua dan bias dari neuron output pertama, kesalahan antara propagasi maju dari output dan hasil aktualnya perlu dihitung terlebih dahulu. . Kesalahannya adalah 3 melalui computing.

2) Kemudian, gradien dari masing-masing bobot dan bias dihitung. Misalnya, bobot dan bias masing-masing adalah 5, 3, 7, 2, 6, seperti yang ditunjukkan pada Gambar 6.11(a). Kemudian, gradien yang sesuai adalah -3, 5, 2, -4, -7.
3) Akhirnya, bobot dan bias yang diperbarui dapat dihitung, seperti 5 – 0,1 × (–3) = 5,3, di mana 0,1 adalah kecepatan pembelajaran yang ditetapkan oleh pengguna. Pada Gambar 6.11(b), bias node direvisi sebagai berikut: 6 – 0,1 × (–7) = 6,7.

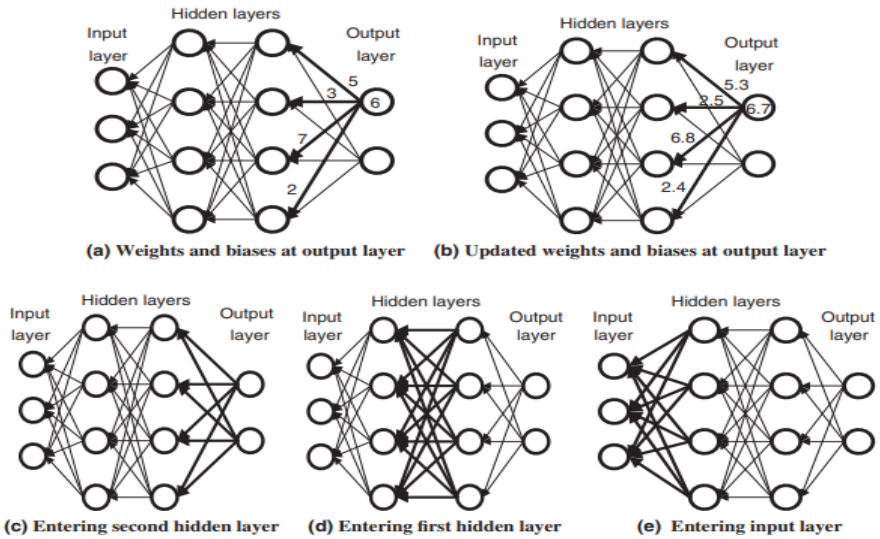
Kesalahan sederhana ini akan membanjiri seluruh jaringan. Seperti yang ditunjukkan pada Gambar 6.11(c), kesalahan keluaran akan merambat mundur ke lapisan tersembunyi kedua. Bobot dan bias terkait akan diperbarui. Ada operasi yang sama pada Gambar 6.11(d).

Operasi tersebut akan diulang sampai kesalahan disebarkan ke lapisan input. Pada saat ini, bobot dan bias dari seluruh jaringan diperbarui, seperti yang ditunjukkan pada Gambar 6.11(e). Prosedur pemutakhiran bobot dalam JST seperti itu disebut perambatan mundur, yang akan diulangi untuk rangkaian kesalahan lainnya.

Proses propagasi mundur jaringan saraf menunjukkan bahwa pengaruh propagasi kesalahan semakin kecil, yang akan membatasi jumlah lapisan tersembunyi dalam jaringan. Jika jumlah lapisan tersembunyi terlalu besar, kesalahan tidak akan diteruskan ke beberapa lapisan sebelumnya dalam proses propagasi mundur, yang mengakibatkan pemutakhiran bobot dan bias yang sesuai tidak dapat dilakukan.

Contoh 6.3 Diagnosis Hiperlipemia Menggunakan Jaringan Syaraf Tiruan

Sebagai contoh, Tabel 6.1 adalah kumpulan data trigliserida, high-density lipoprotein, low-density lipoprotein dan apakah hiperlipemia atau tidak (“1” untuk ya dan “0” untuk tidak) dalam data pemeriksaan kesehatan orang di rumah sakit. Mari kita coba melakukan penilaian pendahuluan apakah seseorang yang menerima pemeriksaan kesehatan mengalami hiperlipemia jika data pemeriksaan kesehatannya berada pada urutan {3.16, 5.20, 0.97, 3.49}.



Gambar 6.11 Prediksi output berbasis propagasi mundur dalam JST sederhana dengan dua lapisan tersembunyi dan empat neuron per lapisan.

Tabel 6.1 Data pemeriksaan pasien yang diduga hiperlipemia.

identitas pasien	Trigliserida (mmol/L)	Total kolesterol (mmol/L)	Lipoprotein Kepadatan Tinggi (mmol/L)	Lipoprotein Kepadatan Rendah (mmol/L)	Hiperlipemia or not
1	3.62	7	2.75	3.13	1
2	1.65	6.06	1.1	5.15	1
3	1.81	6.62	1.62	4.8	1
4	2.26	5.58	1.67	3.49	1
5	2.65	5.89	1.29	3.83	1
6	1.88	5.4	1.27	3.83	1
7	5.57	6.12	0.98	3.4	1
8	6.13	1	4.14	1.65	0
9	5.97	1.06	4.67	2.82	0
10	6.27	1.17	4.43	1.22	0
11	4.87	1.47	3.04	2.22	0
12	6.2	1.53	4.16	2.84	0
13	5.54	1.36	3.63	1.01	0
14	3.24	1.35	1.82	0.97	0

Tabel 6.2 Tabel parameter untuk jaringan syaraf tiruan.

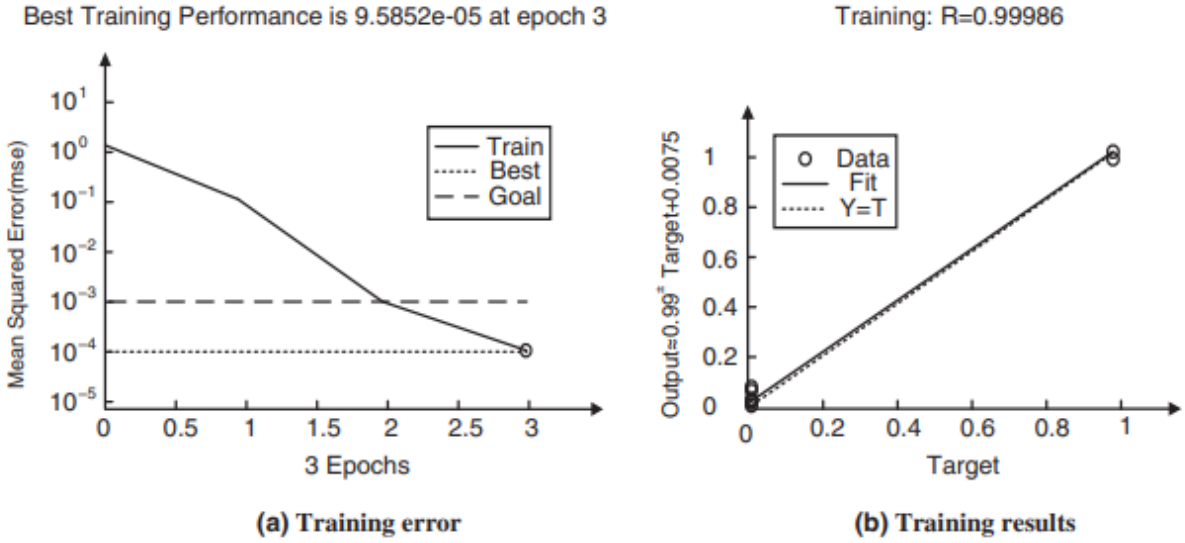
Neuron pada lapisan input	Lapisan tersembunyi	Neuron di lapisan tersembunyi	Neuron di lapisan keluaran
4	1	5	1
Kesalahan yang diizinkan	**Waktu untuk pelatihan**	**Kecepatan belajar**	**Fungsi aktivasi**
10–3	10.000	0.9	Tansig dan purelin

Dari data pada tabel diketahui bahwa masalah ini merupakan masalah dikotomi (“1” untuk hiperlipemia atau “0” untuk sehat) dengan empat atribut. Oleh karena itu, kita dapat melakukan prediksi dan klasifikasi menggunakan jaringan syaraf tiruan.

Karena tidak ada cukup data sampel untuk pelatihan dalam kasus ini, tidak perlu menyiapkan terlalu banyak lapisan tersembunyi dan neuron. Di sini satu lapisan tersembunyi diatur, dan jumlah neuron di setiap lapisan adalah lima. Fungsi Tansig dipilih sebagai fungsi aktivasi antara lapisan input dan lapisan tersembunyi, sedangkan fungsi purelin dipilih sebagai fungsi antara lapisan tersembunyi dan lapisan keluaran (memilih fungsi lain tidak banyak berpengaruh pada hasil kasus ini). Parameter jaringannya tercantum pada Tabel 6.2.

Kemudian kita latih jaringan dengan data pada Tabel 6.1 diatas. MATLAB digunakan untuk pemrograman. Proses pelatihan jaringan ditunjukkan pada Gambar 6.12(a). Kesalahan antara keluaran aktual jaringan dalam proses pelatihan dan keluaran ideal berkurang secara bertahap. Keadaan yang memuaskan dicapai setelah kedua kali perambatan mundur. Hasil akhir pelatihan ditunjukkan pada Gambar 6.12(b). Pengklasifikasi jaringan saraf membagi data pelatihan menjadi dua kategori.

Data pelatihan dibagi menjadi kedua ujungnya dan membentuk dua kelas. Kelas 0 berarti sehat sedangkan kelas 1 berarti hiperlipemia. Hasil klasifikasinya adalah {1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0}. Akurasi untuk klasifikasi mencapai 100%, sehingga jaringan ini dapat digunakan untuk prediksi. Akhirnya, mari kita prediksi apakah seseorang yang datanya {3.16, 5.20, 0.97, 3.49} masing-masing memiliki hiperlipemia atau tidak, dengan jaringan saraf tiruan yang disebutkan di atas. Hasilnya adalah kelas = 1. Oleh karena itu, kita dapat memprediksi orang tersebut mengalami hiperlipemia.



Gambar 6.12 Kesalahan dan hasil pelatihan pada JST pada Contoh 6.3.

6.3 AUTOENCODER BERTUMPUK DAN JARINGAN KEPERCAYAAN YANG MENDALAM

Saat melatih JST, nilai biaya, yaitu kesenjangan antara keluaran prediksi JST dan keluaran aktual, digunakan untuk menyesuaikan bobot dan bias berulang kali selama proses pelatihan. Kemajuan pelatihan mengikuti kecenderungan gradien, yang analog dengan kemiringan. Proses pelatihan seperti menggulingkan batu menuruni lereng. Sebuah batu bergerak cepat di sepanjang permukaan jika gradiennya tinggi. Ketika gradiennya kecil, proses pelatihan ANN berjalan lambat. Namun, gradien berpotensi menghilang selama propagasi mundur.

Biasanya, gradien jauh lebih kecil di lapisan sebelumnya. Akibatnya, lapisan awal sulit untuk dilatih. Namun, lapisan awal sesuai dengan pola dasar dan blok bangunan, terutama dalam pengenalan wajah. Kesalahan akan menyebar di lapisan berikut di JST. Pada tahun-tahun

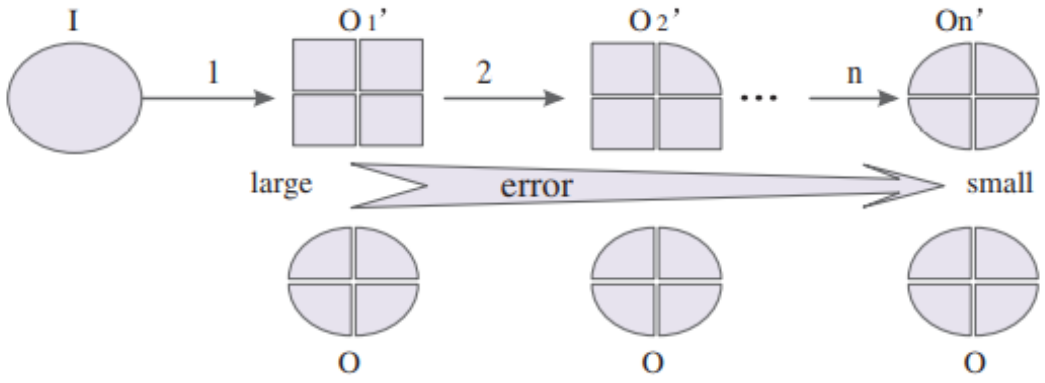
sebelum 2006, tidak ada cara untuk melatih DNN, karena masalah mendasar dari gradien hilang selama proses pelatihan. Masalah ini dapat diselesaikan dengan mengganti lapisan konvensional dengan AutoEncoder, yang secara otomatis menemukan pola dengan merekonstruksi input. Masalah gradien yang hilang akan dihilangkan di AutoEncoder yang ditumpuk.

RBM berisi dua lapisan: lapisan terlihat dan lapisan tersembunyi. Kedua lapisan terhubung tidak terarah. Tidak ada hubungan antar neuron pada lapisan yang sama. RBM melatih parameter jaringan menggunakan metode tanpa pengawasan. DBN berisi beberapa RBM, di mana lapisan tersembunyi dari RBM sebelumnya berfungsi sebagai lapisan yang terlihat dari RBM berikutnya.

AutoEncoder

AutoEncoder dapat diperlakukan sebagai JST khusus. Ada data sampel tetapi tidak ada data label yang sesuai dalam tahap pelatihan jaringan ini. Sebagai gantinya, AutoEncoder mengekstrak data keluaran untuk merekonstruksi data masukan dan membandingkannya dengan data masukan asli. Setelah banyak iterasi, nilai fungsi tujuan mencapai optimalitasnya, yang berarti bahwa data input yang direkonstruksi mampu mendekati data input asli secara maksimal. AutoEncoder adalah alat pembelajaran yang diawasi sendiri dan termasuk dalam pelatihan yang diawasi. Gambar 6.13 menunjukkan contoh pembelajaran dengan empat elemen menyusun lingkaran I.

Biarkan O menunjukkan jawaban yang benar, yaitu lingkaran terdiri dari empat sektor yang identik. Hasil pembelajaran pertama O1’ terdiri dari empat persegi panjang. Kesalahan yang dihitung antara O1’ dan O relatif besar. Kesalahan inilah yang dimaksud untuk memperbaiki model pada pembelajaran kedua. Seperti yang ditunjukkan pada Gambar 6.13, hasil O2’ terdiri dari tiga persegi panjang dan satu sektor. Kesalahan antara O2’ dan O berkurang dan digunakan sekali lagi untuk memodifikasi model. Proses pembelajaran seperti ini akan berulang sebanyak n kali. Akhirnya, hasil On’ mengungkapkan bahwa lingkaran terdiri dari empat sektor.

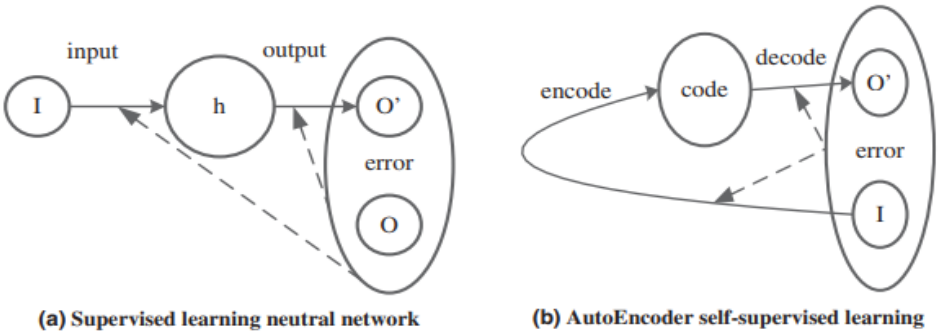


Gambar 6.13 Komposisi siklus pembelajaran terawasi.

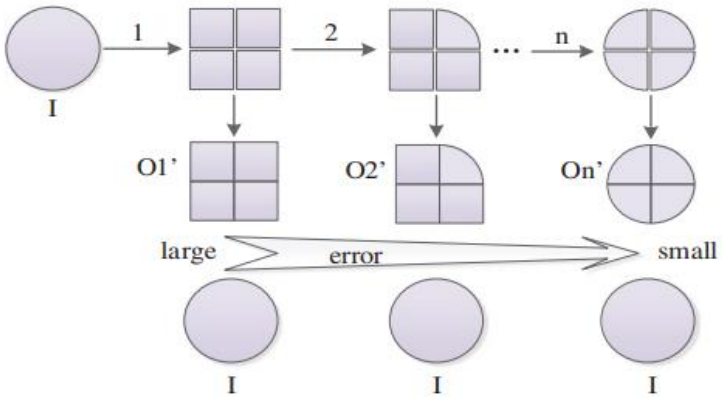
Ketika kesalahan antara On’ dan O mencapai minimum, proses pembelajaran dihentikan. Selama seluruh proses pembelajaran, kami menggunakan keluaran saat ini dan jawaban yang benar untuk menghitung kesalahan. Langkah demi langkah, kami menyesuaikan pengetahuan

dan belajar secara bertahap untuk mendapatkan jawaban yang diantisipasi: lingkaran terdiri dari empat sektor yang identik. Ini disebut pembelajaran terawasi, yaitu ketika masalah (data masukan) dan jawaban yang benar (label) diketahui, jawaban saat ini disesuaikan terus menerus untuk mencapai atau sama dengan jawaban akhir yang benar.

Dengan metode pembelajaran terawasi, semua data input (sampel) pelatihan jaringan netral sesuai dengan nilai yang diharapkan (label). Selisih antara nilai keluaran saat ini dan nilai yang diharapkan digunakan untuk mengatur parameter pada semua lapisan masukan dan keluaran. Kesalahan akan mencapai minimum setelah banyak iterasi. Gambar 6.14(a) menunjukkan struktur alur kerja dari jaringan netral yang diawasi, termasuk lapisan masukan, satu lapisan tersembunyi, dan lapisan keluaran. Nilai keluaran O' diturunkan dari data masukan I melalui lapisan tersembunyi h. O berarti label I. Kesalahan antara O' dan O dihitung. Kemudian, propagasi mundur digunakan untuk mengatur parameter jaringan. Dengan bertambahnya jumlah iterasi, maka error akan berkurang hingga diperoleh nilai yang diminimalkan.



Gambar 6.14 Pembelajaran terawasi dalam JST versus pembelajaran terawasi sendiri dalam AutoEncoder.



Gambar 6.15 Komposisi siklus belajar mandiri.

Gambar 6.15 menunjukkan versi self-supervised dari contoh pada Gambar 6.13. Data masukan lingkaran I diketahui, tetapi tidak ada jawaban referensi yang diberikan. Bagaimana kita

bisa mengetahui kebenaran hasil keluaran? Satu-satunya cara adalah menyusun hasil keluaran dan memverifikasi apakah itu setara dengan I (yaitu lingkaran) atau tidak. Kami menurunkan empat persegi panjang dari pembelajaran pertama dan menggabungkannya untuk mendapatkan hasil O_1'. Kesalahan perhitungan antara O_1' dan I relatif besar. Jadi kami memodifikasi pengetahuan kombinasi berdasarkan kesalahan.

Kami menurunkan tiga persegi panjang dan satu sektor dari pembelajaran kedua dan menggabungkannya untuk mendapatkan hasil O_2'. Kesalahan perhitungan antara O_2' dan I berkurang dan pengetahuan dimodifikasi sekali lagi berdasarkan kesalahan. Setelah mengulangi proses pembelajaran tersebut selama n kali, On mengungkapkan pengetahuan menggabungkan empat sektor yang sama. Akhirnya, kesalahan perhitungan antara On' dan I dikurangi menjadi nol, dan pengetahuan yang diinginkan diperoleh tanpa mengetahui jawaban yang benar sebelumnya. AutoEncoder adalah alat pembelajaran yang diawasi sendiri. Karena jawaban yang benar tidak diberikan, data inputnya juga berfungsi sebagai data berlabel. Gambar 6.14(b) menunjukkan alur kerja pembelajaran mandiri di AutoEncoder.

Data masukan I merepresentasikan vektor m-dimensi $I \in R^m$. Pembelajaran mandiri AutoEncoder terdiri dari dua bagian: encoder dan decoder. Dengan proses encoding, kode, yang merupakan kode vektor n-dimensi $\in R^n$, dihitung. Dengan proses decoding, O' direkonstruksi, dimana $O' \in R^m$ merepresentasikan vektor berdimensi-m. Kemudian, kesalahan antara O' dan I diperoleh. Dengan menggunakan penurunan gradien stokastik, parameter encoder dan decoder disesuaikan untuk pengurangan kesalahan. Ketika kesalahan antara I dan O' diminimalkan, kode dapat dianggap sebagai inkarnasi dari I. Dengan kata lain, kode adalah semacam representasi fitur yang diekstraksi dari I.

Algoritma 6.1 Konstruksi AutoEncoder

Input: T: Kumpulan sampel

Output: Representasi fitur dari data input

Prosedur:

1) Inisialisasi set parameter θ={w_1, w_2, b_1, b_2,}
2) Encode: Hitung representasi dari lapisan tersembunyi
3) Decode: Gunakan representasi dari lapisan tersembunyi untuk merekonstruksi input
4) Hitung L_{recon}(I, O') dan nilai fungsi tujuan J(θ)
5) Nilai apakah J(θ) memenuhi kondisi akhir atau tidak?
6) Jika ya, kembalikan 7), jika tidak kembalikan 6).
7) Memodifikasi set parameter dengan propagasi mundur dan kembali 2;
8) Akhir

Algoritma singkat AutoEncoder disajikan dalam Algoritma 6.1, yang mencakup tiga langkah utama:

1) **Langkah 1 – Encode:** Ubah data input I menjadi kode hidden layer dengan kode = f (I) = $s_1(w_1 \cdot I + b_1)$, di mana $w_1 \in R^{m \times n}$ dan $b_1 \in R^n$. S_1 adalah fungsi aktivasi. Fungsi sigmoid atau tangen hiperbolik dapat digunakan.
2) **Langkah 2 – Decode:** Berdasarkan kode di atas, rekonstruksi nilai input I dengan persamaan g(kode) = $s_2(w_2 \cdot \text{kode} + b_2)$, di mana $w_2 \in R^{n \times m}$ dan $b_2 \in R^m$. Fungsi aktivasi S_2 sama dengan S_1.
3) **Langkah 3 – Hitung galat kuadrat:** $L_{recon}(I, O') = ||I - O'||^2$, yang merupakan fungsi biaya galat. Minimalisasi kesalahan dicapai dengan mengoptimalkan fungsi tujuan: $J(\theta) = \sum_{I \in D} L(I, g(f(I))\theta = \{w_1, w_2, b_1, b_2\}$

Melalui AutoEncoder, representasi fitur diperoleh untuk mewakili input asli dengan kode. Untuk tujuan klasifikasi, kita masih membutuhkan classifier. Pengklasifikasi terhubung dengan lapisan tersembunyi AutoEncoder. Baik SVM atau softmax dapat digunakan sebagai pengklasifikasi. Metode pelatihan yang diawasi seperti penurunan gradien stokastik dapat digunakan untuk melatih pengklasifikasi. Kode yang diperoleh dari AutoEncoder berfungsi sebagai input ke pengklasifikasi, sedangkan outputnya (Y' pada Gambar 6.16) dibandingkan dengan data berlabel (Y) untuk fine tuning yang diawasi.

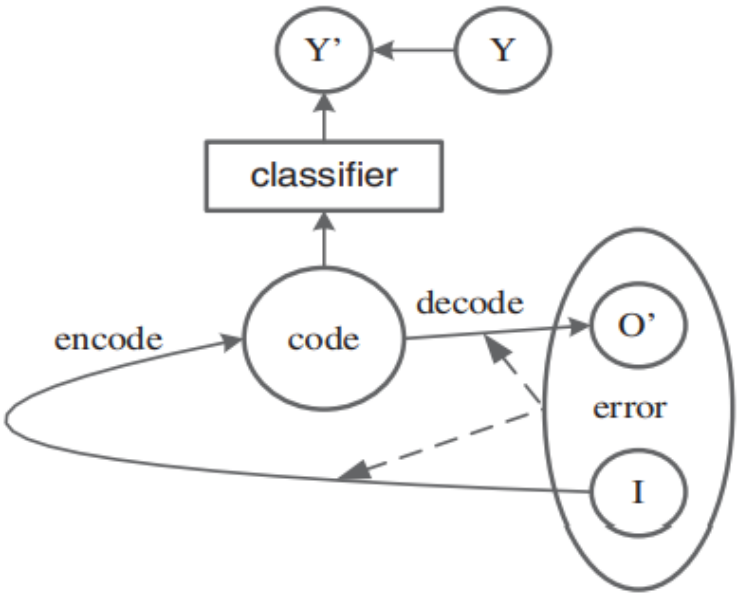
AutoEncoder Bertumpuk

Stacked AutoEncoder (SAE) adalah jaringan netral dengan beberapa lapisan tersembunyi antara lapisan input dan lapisan output, sementara setiap lapisan tersembunyi berhubungan dengan AutoEncoder. Gambar 6.17 menunjukkan struktur AutoEncoder bertumpuk, yang mencakup AutoEncoder multi-lapisan dan pengklasifikasi.

AutoEncoder multi-lapisan

Data masukan asli digunakan pada lapisan AutoEncoder pertama. Setelah melalui beberapa putaran encoding dan decoding, kesalahan rekonstruksi secara bertahap diminimalkan dan parameter encoder dan decoder diperbarui secara iteratif di lapisan pertama. Akhirnya, kami mendapatkan kode lapisan pertama, yang mewakili fitur data input asli. Setelah membangun struktur jaringan lapisan AutoEncoder pertama, lapisan AutoEncoder kedua dilatih dengan cara yang sama. Namun perbedaannya adalah kode keluaran lapisan pertama diperlakukan sebagai data masukan lapisan kedua.

Algoritma pelatihan encoder multi-layer adalah versi iteratif dari Algoritma 6.1. Biarkan n menunjukkan jumlah lapisan di SAE. Biarkan T_n menunjukkan jumlah dataset sampel. Untuk setiap sampel data x_m, m = 1,2,3,…, Tn, Algoritma 6.1 akan dilakukan sebanyak n kali secara iteratif. Kode keluaran dari lapisan sebelumnya selalu diperlakukan sebagai masukan dari lapisan yang terakhir. Pelatihan multi-layer seperti itu akan berkontribusi untuk memperoleh representasi fitur multi-layer dari data input asli. Akhirnya, jaringan AutoEncoder multi-layer dibangun.

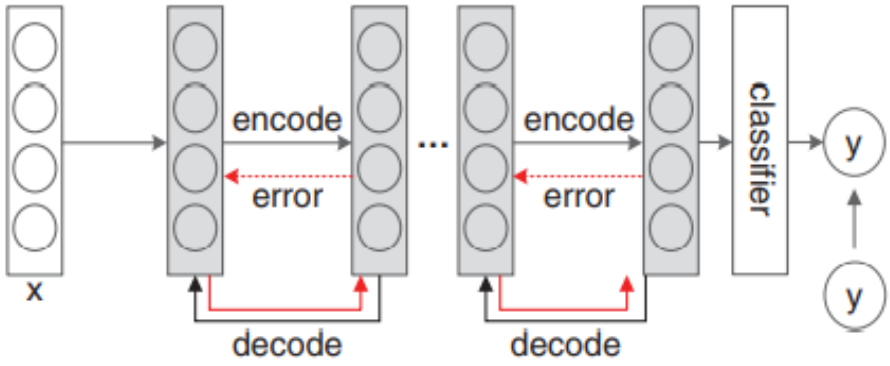


Gambar 6.16 Proses klasifikasi oleh AutoEncoder.

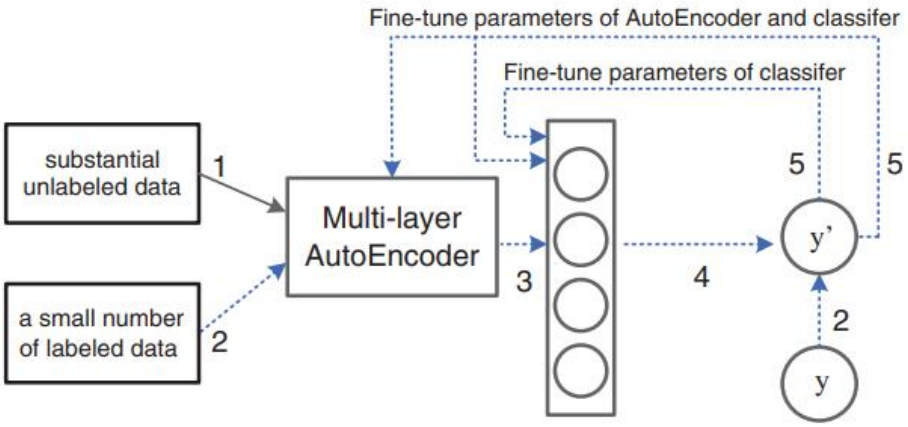
Penyetelan Halus yang Diawasi

Gambar 6.18 menunjukkan peta pelatihan SAE. Pengklasifikasi SAE terhubung dengan lapisan terakhir AutoEncoder. Ada dua metode untuk penyetelan halus yang diawasi. Salah satunya adalah dengan menyesuaikan parameter classifier saja. Yang lainnya adalah menyesuaikan parameter pengklasifikasi dan semua lapisan AutoEncoder. Tanda No 1 sampai No 5 pada Gambar 6.18 mewakili urutan operasi dalam pelatihan. Ada dua operasi No. 5, yang mewakili dua jenis operasi fine-tuning. Kami merangkum lima langkah untuk menggunakan algoritma fine tuning parameter di SAE:

- **Langkah 1:** Data substansial yang tidak berlabel digunakan untuk melatih AutoEncoder multi-layer. Dan ekstraksi fitur data multi-layer dibangun.
- **Langkah 2:** Dapatkan kode di AutoEncoder lapisan tertinggi.



Gambar 6.17 Struktur AutoEncoder bertumpuk.



Gambar 6.18 Sketsa peta pelatihan AutoEncoder bertumpuk.

- **Langkah 3:** Hitung hasil klasifikasi Y' (yaitu output dari classifier). Hitung kesalahan antara Y dan Y' berdasarkan fungsi biaya. Perhatikan bahwa sejumlah kecil data berlabel akan digunakan.
- **Langkah 4:** Menilai apakah kondisi akhir terpenuhi berdasarkan fungsi biaya? Jika ya, struktur SAE sudah terlatih dengan baik, dan penyetelan parameter yang diawasi dengan baik dihentikan. Jika tidak, lanjutkan dengan Langkah 5.
- **Langkah 5:** Sesuaikan parameter pengklasifikasi (penyesuaian untuk parameter AutoEncoder multi-layer adalah opsional) melalui penurunan gradien stokastik. Kemudian kembali ke Langkah 3.

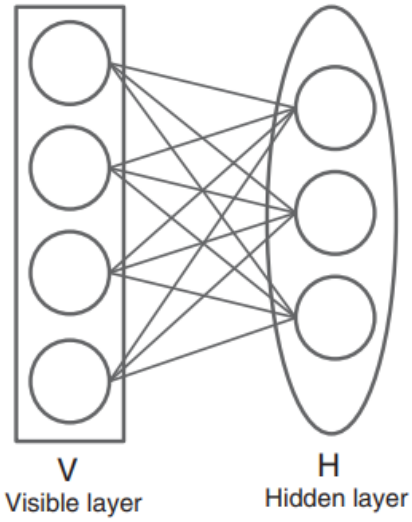
Mesin Boltzmann Terbatas

Mesin Boltzmann terbatas (RBM) adalah model jaringan saraf yang dapat mewujudkan pembelajaran tanpa pengawasan. Ini mencakup dua lapisan, lapisan terlihat V dan lapisan tersembunyi H, yang dihubungkan oleh graf tak berarah. Tidak ada hubungan antar neuron pada lapisan yang sama. Pada bagian ini, kami hanya memperkenalkan V dan H sebagai unit biner. Masukan dari RBM adalah data vektor m-dimensi V, dimana V = (v_1, v_2,…,v_m) dan $v_i \in \{0, 1\}$. v_i adalah singkatan dari keadaan biner neuron i pada lapisan yang terlihat. Keluaran dari RBM adalah vektor n-dimensi H, dimana H = (h_1, h_2, … , h_n) dan hj $\in \{0, 1\}$. h_j adalah singkatan dari keadaan biner neuron j di lapisan tersembunyi (Gambar 6.19).

Sekarang, kami memberikan contoh sederhana untuk memahami proses pembelajaran RBM, seperti yang ditunjukkan pada Gambar 6.20. Tujuannya adalah untuk mendapatkan jawaban dari pertanyaan berikut: Bentuk apa yang menyusun grafik input? Hanya ada dua jenis komponen dalam grafik, yaitu persegi dan segitiga. Mari kita gunakan kode 1 dan kode 0 untuk masing-masing mewakili bentuk persegi dan segitiga. Mari kita asumsikan urutan pengkodean adalah sudut kiri atas, sudut kanan atas, sudut kiri bawah dan sudut kanan bawah. Kemudian, grafik masukan tersebut sesuai dengan kode empat digit, yaitu 1011. Pada Gambar 6.20, kami memperoleh lapisan H dengan pemetaan dari lapisan V ke lapisan H. Kemudian kami

menggunakan pemetaan simetris untuk merekonstruksi lapisan V berdasarkan lapisan H. Sebagai ditunjukkan pada Gambar 6.20, kami melakukan operasi berikut:

1) Sebuah kode 1011 diperoleh sebagai V. Melalui pemetaan, ekspresi 01 pada lapisan H dihitung, yang berarti grafik input terdiri dari segitiga. Kemudian, pemetaan simetris diadopsi untuk mendapatkan 0011 sebagai nilai V1.

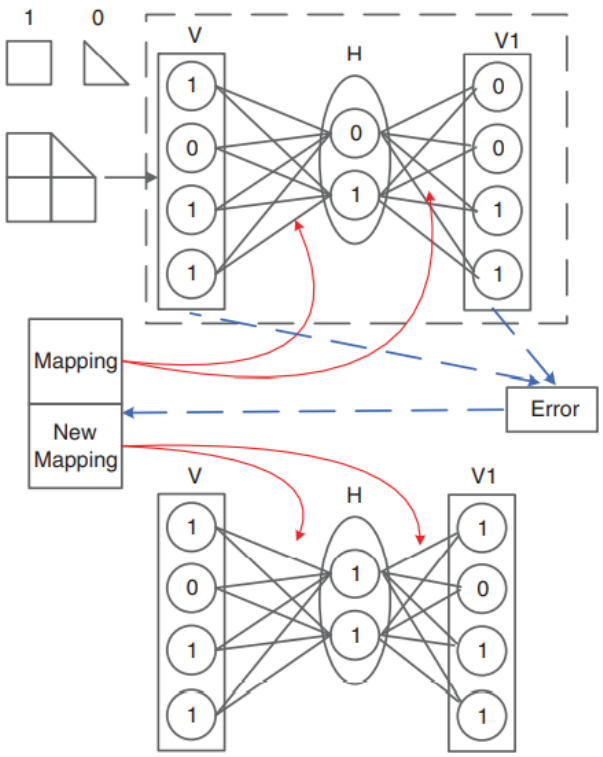


Gambar 6.19 Struktur satu tahap mesin Boltzmann terbatas (RBM).

2) Hitung kesalahan antara distribusi V dan V1, dan perbaiki parameter pemetaan sesuai dengan kesalahan.
3) Ulangi langkah 1) dan langkah 2) dengan pemetaan baru. Melakukan pelatihan dan menetapkan RBM setelah selesai pelatihan. Ini termasuk lapisan V, lapisan H dan pemetaan antara dua lapisan.

Seperti yang terlihat dari Gambar 6.20, yang pada akhirnya kita inginkan adalah menemukan Pemetaan yang baik. Sekarang kami menjelaskan algoritma divergensi kontrastif (CD), yang dapat dengan cepat mendapatkan Pemetaan.

Kami mendefinisikan θ = (w, bv, bh) sebagai parameter Pemetaan yang dipelajari dari jaringan RBM. w menunjukkan satu set bobot antara lapisan terlihat V dan lapisan tersembunyi H, $w \in R^{m \times n}$. w_{ij} adalah bobot antara neuron tampak i dan neuron tersembunyi j. bv adalah vektor yang terdiri dari simpangan setiap neuron pada lapisan tampak, dan bvi merupakan nilai simpangan dari neuron tampak i, $bv \in R^m$. bh adalah vektor yang terdiri dari simpangan setiap neuron pada lapisan tersembunyi, $bh \in R^n$. bh_j adalah singkatan dari nilai deviasi dari neuron tersembunyi j. Tugas mempelajari RBM adalah membentuk struktur jaringan RBM atau memperoleh parameter $\theta *$ yang optimal dengan pembelajaran CD.



Gambar 6.20 Diagram skema untuk mempelajari komposisi gambar dengan RBM

Algoritma pembelajaran RBM memanfaatkan distribusi Boltzmann dan menyelesaikan parameter θ melalui estimasi kemungkinan maksimum. Dengan kata lain, P (v | θ) tunduk pada distribusi Boltzmann:

$$P(\mathrm{v} \mid \theta) = \frac{\sum_{\mathrm{h}} e^{-E(\mathrm{v},\mathrm{h}|\theta)}}{Z(\theta)} \tag{6.12}$$

Dimana

$$E(\mathrm{v},\mathrm{h}|\theta) = -\sum_{i=1}^{\mathrm{m}} bv_i v_i - \sum_{j=1}^{\mathrm{n}} bh_j h_j - \sum_{i=1}^{\mathrm{m}} \sum_{j=1}^{\mathrm{n}} v_i w_{ij} h_j,\ Z(\theta) = \sum_{\mathrm{v},\mathrm{h}} e^{-E(\mathrm{v},\mathrm{h}|\theta)}$$

Fungsi kemungkinan maksimum dijelaskan dalam Persamaan 6.13, yang dapat diselesaikan dengan metode penurunan gradien:

$$L(\theta) = \sum_{t=1}^{T} \log P(\mathrm{v}^{(t)} \mid \theta) \tag{6.13}$$

Gunakan Persamaan (6.14) untuk menghitung probabilitas aktivasi neuron tersembunyi j pada lapisan tersembunyi, yaitu menghitung probabilitas ketika keadaan neuron tersembunyi j sama dengan 1. Dengan demikian kita dapat memperoleh keadaan h dari lapisan tersembunyi:

$$P(h_{1j} = 1 \mid \mathrm{v}_1, \theta) = \sigma\left(bh_j + \sum_{i=1}^{\mathrm{m}} v_{1i} w_{ij}\right) \quad (6.14)$$

dimana fungsi adalah fungsi sigmoid. Kami merangkum lima langkah untuk menghitung P(h_1 = 1lv_1, θ):

- **Langkah 1:** Input status lapisan terlihat v_1, parameter jaringan θ, jumlah neuron pada lapisan tampak m, jumlah neuron pada lapisan tersembunyi n.
- **Langkah 2:** j = 1.
- **Langkah 3:** Hitung probabilitas aktivasi untuk neuron di lapisan tersembunyi sebagai Persamaan (6.14).
- **Langkah 4:** Sesuai dengan distribusi seragam, hasilkan bilangan floating point acak antara 0 dan 1. Jika lebih kecil dari P(h_{1j} = 1 | v_1, θ) nilai h1j adalah 1; jika tidak, nilainya adalah 0.
- **Langkah 5:** j++.
- **Langkah 6:** Jika j < m, maka kembali ke Langkah 3.

Algoritma 6.2 Algoritma pembelajaran cepat berdasarkan Divergensi Kontrastif

Input: X: sampel pelatihan – datum asli yang dimasukkan x
m: jumlah neuron di lapisan tersembunyi
λ: kecepatan belajar
P: jumlah iteratif pelatihan

Output:
Menetapkan RBM; parameter jaringan θ = {w, bh, bv}

Prosedur:

1) Inisialisasi: keadaan awal V_1 untuk neuron di lapisan yang terlihat; V_1 = x; untuk inisialisasi, θ = {w, bh, bv}
2) untuk t = 1,2,3,…P
3) Hitung probabilitas aktivasi untuk semua neuron di lapisan tersembunyi, P(h_1 = 1 | v_1, θ)
4) Hitung probabilitas aktivasi untuk semua neuron di lapisan tersembunyi, V2
5) Hitung probabilitas aktivasi untuk semua neuron di lapisan tersembunyi, P(h_2 = 1 | v_2, θ)
6) Perbarui bobot: w = w + λ (P(h_1 = 1|v_1,θ)v_1^T - P(h_2 = 1|v_2,θ)v_2^T)
7) Perbarui penyimpangan pada lapisan yang terlihat : bv = bv + λ(v_1 – v_2)
8) Perbarui penyimpangan pada lapisan tersembunyi: bh = bh + λ (P(h_1 = 1 |v_1,θ) - P(h_2 = 1 | v_2, θ))
9) akhir

Setelah mendapatkan semua status pada lapisan tersembunyi, maka peluang aktivasi neuron i pada lapisan tampak dapat dihitung dengan menggunakan Persamaan (6.15) secara simetris. Kemudian, keadaan aktivasi untuk neuron i dapat diperoleh dengan memanfaatkan hasil probabilitas aktivasi $v_i \in \{0, 1\}$.

$$P(v_{2i} = 1 \mid \mathrm{h}_1, \theta) = \sigma\left(bv_i + \sum_{j=1}^{\mathrm{n}} w_{ij} h_{1j}\right) \quad (6.15)$$

Algoritma untuk menghitung peluang aktivasi neuron pada lapisan tampak mirip dengan lapisan tersembunyi, sebagai berikut:

- **Langkah 1:** Masukkan status lapisan tersembunyi h_1, parameter jaringan , jumlah neuron pada lapisan tampak m, jumlah neuron pada lapisan tersembunyi n.
- **Langkah 2:** I = 1.
- **Langkah 3:** Hitung probabilitas aktivasi untuk neuron di lapisan yang terlihat, seperti Persamaan (6.15)
- **Langkah 4:** Sesuai dengan distribusi seragam, hasilkan bilangan floating point acak antara 0 dan 1. Jika lebih kecil dari $P(v_{2i} = 1 \mid h_1, \theta)$, nilai v_{2j} adalah 1; jika tidak, nilainya adalah 0.
- **Langkah 5:** saya++
- **Langkah 6:** Jika I < n, maka kembali ke Langkah 3.

Contoh 6.4 Merekomendasikan Film dengan Mesin Boltzman Terbatas

Kami dapat menggunakan model RBM untuk merekomendasikan film kepada pengguna. Memanfaatkan data peringkat pengguna yang diketahui untuk membangun model RBM, model ini akan diadopsi untuk memprediksi peringkat pengguna untuk semua film. Pertama, kami memilah semua data peringkat dalam urutan menurun. Kemudian, kita bisa mendapatkan film yang mendapat rating tertinggi dan belum pernah ditonton oleh pengguna sebelumnya. Dengan demikian kami dapat merekomendasikannya kepada pengguna.

Tabel 6.3 Peringkat film menurut pemirsa.

Id	Film 1	Film 2	Film 3	Film 4
Pengguna1	3	4	4	1
Pengguna15	3	5	0	0

Dengan asumsi ada empat film, peringkat film adalah bilangan bulat dan bervariasi dari 1 hingga 5, dan 0 menunjukkan tidak ada peringkat. Tabel 6.3 menunjukkan peringkat pengguna 1 dan pengguna 15.

Langkah-langkah merekomendasikan film dengan mesin Boltzman terbatas adalah sebagai berikut:

Langkah 1: Dataset dan struktur RBM

Peringkat semua film oleh satu pengguna dapat diwakili oleh matriks 5 × 4 v, di mana v(i, j) = 1 berarti nilai pengguna i untuk film j. Dengan demikian, matriks peringkat pengguna 1 dan pengguna 15 adalah sebagai

$$v_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad v_{15} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Biarkan jumlah neuron di lapisan terlihat dan lapisan tersembunyi masing-masing menjadi 20 dan 5 dalam model RBM. Kita dapat mengubah matriks rating menjadi vektor dengan 20 elemen. Dengan demikian v1 dan v15 dapat ditransformasikan menjadi {0,0,1,0,0,0,0,0,1,0, 0,0,0,1,0, 1,0, 0,0,0} dan masing-masing {0,0,1,0,0,0,0,0,0,1, 0,0,0,0, 0,0, 0,0,0}.

Langkah 2: Pelatihan

Setelah memasukkan data rating pengguna, kita dapat menggunakan Algoritma 6.2 untuk melatih RBM. Penyimpangan yang sesuai dari lapisan yang terlihat dapat diperoleh sebagai

$$a = \begin{pmatrix} -0.6 & -0.6 & -0.3 & 0 \\ -0.3 & -0.3 & 0.6 & 0.3 \\ 0.6 & 0.3 & 0 & -0.6 \\ -0.3 & 0.3 & 0.6 & 0.0 \\ -0.3 & 0.3 & -0.9 & 0 \end{pmatrix}$$

Penyimpangan lapisan tersembunyi adalah b = (−1.2 0.6 0.6 0.3 0.3).

Matriks bobot adalah matriks 20 × 5, di mana ia dapat diwakili oleh lima matriks sesuai dengan neuron lapisan tersembunyi. Setiap matriks menunjukkan bobot koneksi di antara 20 neuron pada lapisan tampak dan 1 neuron pada lapisan tersembunyi.

Langkah 3: Rekomendasi film

1) Masukkan data rating pengguna 15, dan gunakan Persamaan (6.16) untuk menghitung probabilitas aktivasi semua unit tersembunyi, yaitu ph= (0,0214, 0,0151, 0,0453, 0,1404, 0,6583). Bobot dan deviasi lapisan tersembunyi dapat diperoleh melalui pelatihan:

$$p(h_j = 1|V) = \frac{1}{1 + \exp\left(-b_j - \sum_{i=1}^{M} \sum_{k=1}^{K} v_i^k W_{ij}^k\right)} \quad (6.16)$$

$$w(:,:,1) = \begin{pmatrix} -0.8388 & -0.1559 & -1.1827 & -0.2224 \\ -0.7873 & 0.0986 & -0.0791 & -0.8191 \\ -0.6809 & -1.0458 & -0.2480 & -0.3867 \\ -1.2027 & 0.0872 & -0.1243 & -0.5868 \\ -1.1445 & -0.6589 & -0.8836 & -0.1172 \end{pmatrix}$$

$$w(:,:,2) = \begin{pmatrix} -0.1580 & -0.6287 & 0.3352 & -0.2182 \\ 0.1031 & 0.0961 & 0.3511 & -0.2471 \\ 0.1854 & 0.6668 & 0.4663 & -0.1070 \\ -0.1052 & -0.2801 & -0.2151 & -0.7534 \\ -0.2325 & -0.0457 & -0.0993 & 0.4049 \end{pmatrix}$$

$$w(:,:,3) = \begin{pmatrix} 0.0467 & -0.2369 & -0.0054 & 0.5772 \\ -0.0050 & -1.1134 & 0.1793 & 0.0143 \\ 0.5374 & -0.2505 & -0.1696 & 0.2379 \\ 0.5298 & -0.1375 & 0.4546 & -0.8972 \\ 0.8843 & -0.2688 & -0.6814 & -0.3714 \end{pmatrix}$$

$$w(:,:,4) = \begin{pmatrix} -0.8423 & 0.0999 & -0.7604 & 0.0074 \\ -0.3207 & -0.6556 & 0.4505 & 0.2922 \\ 1.1088 & -0.1351 & -0.1854 & -0.2619 \\ 0.4764 & 0.2176 & 0.2992 & 0.0226 \\ 0.4378 & 0.1666 & -0.6949 & -0.2281 \end{pmatrix}$$

$$w(:,:,5) = \begin{pmatrix} -0.7458 & -0.5607 & -0.0101 & 0.1658 \\ -0.4698 & 0.0969 & 0.4129 & 0.7707 \\ 0.7059 & 0.7355 & 0.0868 & -0.3635 \\ 0.5583 & 0.2992 & 0.4809 & -0.1271 \\ 0.7589 & 1.2223 & -0.2460 & 0.3042 \end{pmatrix}$$

Distribusi seragam menghasilkan angka floating point acak antara 0 dan 1. Jika kurang dari ph_j, maka $h_j = 1$, jika tidak $h_j = 0$. Maka kita bisa mendapatkan h = (0, 0, 0, 0, 1).

2) Menurut ph probabilitas aktivasi lapisan tersembunyi yang dihitung pada langkah sebelumnya, gunakan Persamaan (6.17) untuk menghitung pv probabilitas aktivasi lapisan yang terlihat. Matriks bobot dan deviasi lapisan yang terlihat dapat diperoleh dengan pelatihan. Kita dapat menghitung

probabilitas pengaktifan yang sesuai dari setiap peringkat k untuk setiap film i, di mana k = 1,…,5. pv(i.j) berarti probabilitas rating film j i:

$$p(v_i^k = 1|h) = \frac{1}{1 + \exp\left(-a_i^k - \sum_{j=1}^{F} W_{ij}^k h_j\right)} \tag{6.17}$$

$$pv = \begin{pmatrix} 0.2066 & 0.2385 & 0.4231 & 0.5414 \\ 0.3165 & 0.4494 & 0.7336 & 0.7447 \\ 0.7868 & 0.7380 & 0.5217 & 0.2762 \\ 0.5642 & 0.6455 & 0.7467 & 0.4683 \\ 0.6128 & 0.8209 & 0.2412 & 0.5755 \end{pmatrix}$$

3) Kami memilih probabilitas tertinggi dari setiap kolom sebagai peringkat oleh pengguna 15 untuk film i. Misalnya, untuk film 1, probabilitas peringkat 1,2,3,4,5 berturut-turut adalah 0,2066, 0,3165, 0,7868, 0,5642, dan 0,6128. Probabilitas tertinggi adalah 0,7868, peringkat yang sesuai adalah 3, jadi pengguna 15 menilai film 1 sebagai 3. Setelah menyimpulkan sisanya dari ini, peringkat pengguna untuk empat film adalah (3,5,4,2).
4) Karena pengguna 15 belum menilai film 3 dan film 4, kemungkinan pengguna 15 belum melihat kedua film tersebut. Menurut peringkat, kami merekomendasikan film 3 terlebih dahulu, dan film 4 detik.

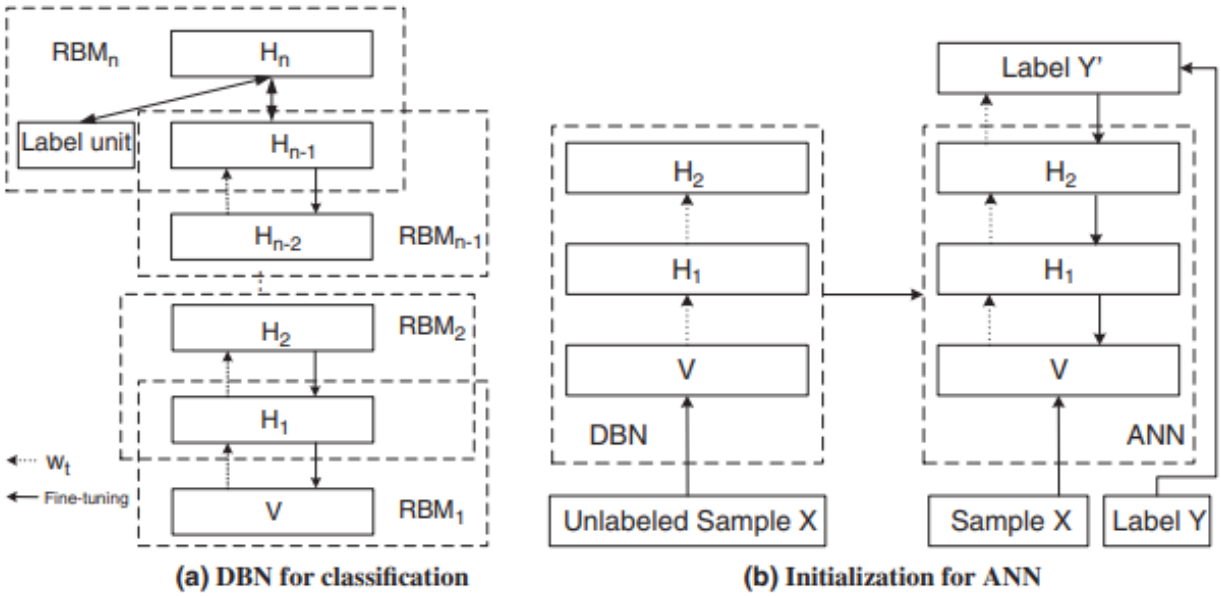
Jaringan Kepercayaan yang Dalam

Jaringan kepercayaan dalam (DBN) adalah model *Deep learning* hibrida, yang diusulkan oleh Hinton et al. pada tahun 2006. Gambar 6.21(a) menunjukkan bahwa DBN mencakup satu lapisan terlihat V dan n lapisan tersembunyi. DBN terdiri dari n stacked RBM, yang berarti lapisan tersembunyi dari RBM sebelumnya adalah lapisan yang terlihat dari RBM berikutnya. Input asli adalah lapisan terlihat V. Lapisan V ini, bersama dengan lapisan tersembunyi H_1, terdiri dari satu RBM. H_1 adalah lapisan V_2 yang terlihat dari RBM kedua. Lapisan tersembunyi H_1, bersama dengan lapisan tersembunyi H_2, terdiri dari RBM lain dan seterusnya. Semua dua lapisan yang berdekatan membentuk RBM.

Pada Gambar 6.21(a), lapisan yang terlihat dan lapisan tersembunyi dari lapisan atas adalah bagian dari koneksi tidak terarah, yang dikenal sebagai memori asosiatif. Lapisan terlihat RBM di lapisan atas terdiri dari lapisan tersembunyi H_{n-1} di RBM sebelumnya dan label kategori. Pelatihan DBN dibagi menjadi dua bagian, yaitu unsupervised training dan supervised fine tuning. Pelatihan tanpa pengawasan menggunakan sejumlah besar data tanpa label untuk melatih RBM satu per satu. Penyesuaian halus yang diawasi menggunakan sejumlah kecil data dengan label untuk menyesuaikan parameter setiap lapisan di seluruh jaringan dengan halus.

Pelatihan tanpa pengawasan mengadopsi input asli V sebagai lapisan yang terlihat dari RBM pertama, menggunakan metode yang diperkenalkan di Bagian 6.3.1 untuk melatih RBM_1 dan memperbaiki parameter koneksi hingga akhir pelatihan. Kemudian kita mendapatkan lapisan

tersembunyi H_1 dan biarkan H_1 ini menjadi lapisan terlihat V_2 dari RBM kedua. Metode serupa diterapkan untuk melatih RBM_2 dan mendapatkan lapisan tersembunyi H_2. Ulangi proses ini sampai semua pelatihan RBM selesai. Pelatihan RBM multi-lapisan tanpa pengawasan diberikan dalam Algoritma 6.3.



Gambar 6.21 Struktur jaringan kepercayaan yang mendalam (DBN).

Algoritma 6.3 Pelatihan RBM multi-layer tanpa pengawasan

Input: V: Data pelatihan

rn: Jumlah lapisan RBM

Output: Status aktivasi setiap neuron di lapisan terlihat V

prosedur:

1) V_1= V
2) untuk t = 1 sampai rn
3) Parameter awal θ_t = {w_t, bh_t, bv_t} dari jaringan RBM ke-t
4) Gunakan algoritma divergensi kontras untuk melatih RBM ke-t dan mendapatkan h_t
5) t++;
6) Vt = Ht-1
7) akhir

6.4 JARINGAN SARAF KONVOLUSI (CNN) DAN EKSTENSI

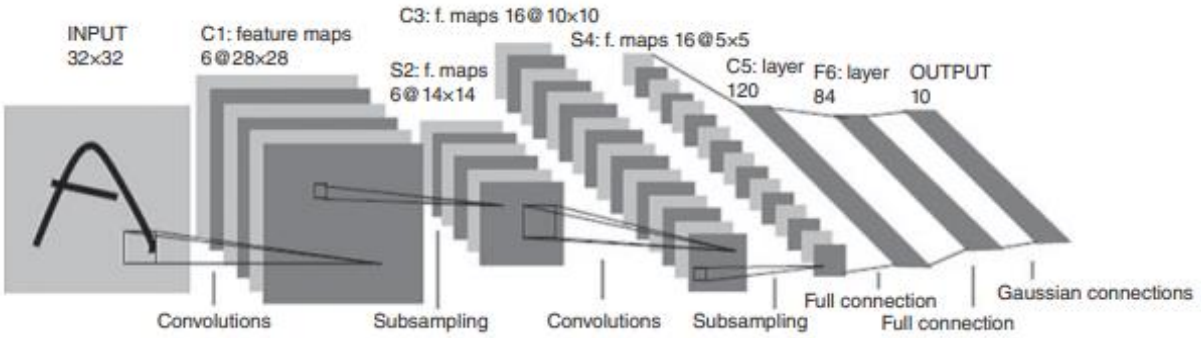
Jaringan saraf convolutional (CNN) adalah jenis jaringan saraf feed-forward, yang menggunakan konvolusi dan mengurangi jumlah bobot dalam jaringan dan mengurangi kompleksitas perhitungan dibandingkan dengan jaringan saraf feed-forward tradisional. Struktur jaringan semacam ini mirip dengan jaringan saraf biologis. CNN termasuk dalam metode

pembelajaran terawasi, dan diterapkan secara luas pada pengenalan suara dan bidang gambar. Konvolusi dan penyatuan telah diperkenalkan sebelumnya. Proses pelatihan CNN juga dihadirkan.

CNN terdiri dari banyak lapisan. Umumnya, ini mencakup komponen dasar seperti input, convolution, pooling, fully connected dan output layer. Kita perlu memutuskan berapa banyak convolution layer dan pooling layer yang harus digunakan dan jenis pengklasifikasi apa yang harus dipilih sesuai dengan setiap masalah aplikasi tertentu, seperti menumpuk blok bangunan. Gambar 6.22 menunjukkan struktur jaringan saraf convolutional yang digunakan oleh LeNet-5. Ini mencakup 7 lapisan, yaitu 3 konvolusi, 2 penyatuan, 1 terhubung penuh dan 1 lapisan keluaran. Konsep tersebut diilustrasikan pada Gambar 6.22, berdasarkan karya LeCun (1998) [12].

Konvolusi di CNN

Berikut ini akan dijelaskan konsep konvolusi pada CNN dengan aplikasi pemahaman citra. Saat mempertimbangkan kebutuhan untuk memproses gambar berukuran 500 × 500 piksel, jumlah neuron di lapisan input harus diatur ke 500 × 500. Kami mengasumsikan satu set 108 neuron di lapisan tersembunyi. Setiap koneksi antara satu neuron di lapisan input dan satu neuron di lapisan tersembunyi harus ditetapkan sebagai parameter bobot.



Gambar 6.22 Convolutional Neural Network yang digunakan di LeNet-5.

Di sini mari kita bandingkan jumlah parameter bobot yang diperlukan dalam dua kasus saat menggunakan jaringan saraf yang terhubung penuh dan menggunakan jaringan saraf yang terhubung secara lokal.

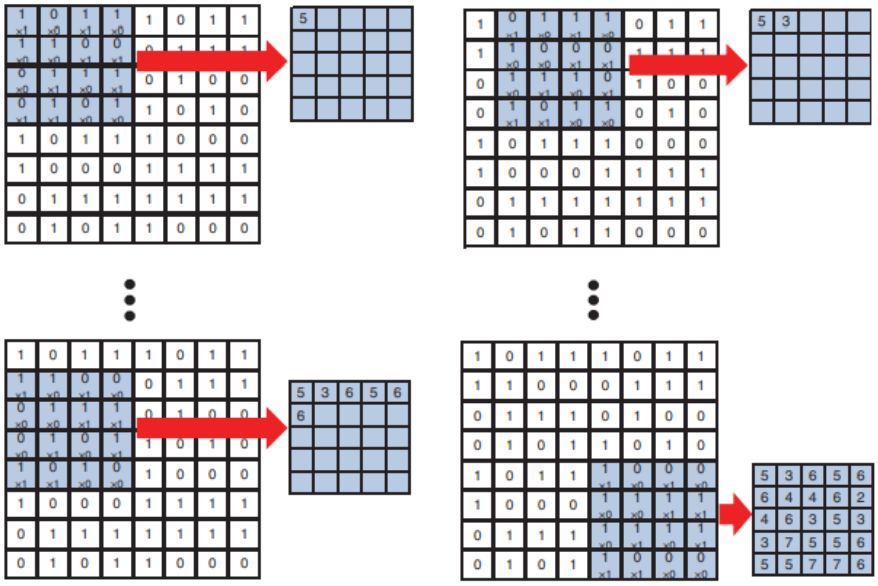
Saat membangun jaringan saraf yang sepenuhnya terhubung untuk menangani gambar, jumlah parameter bobot antara lapisan input dan lapisan tersembunyi akan menjadi 500 × 500 × 108 = 25 × 1012. Memahami gambar besar dengan metode konfigurasi penuh - koneksi menghadapi masalah bahwa parameter terlalu banyak dan beban computing terlalu besar. Orang hanya dapat melihat sebagian dari informasi gambar pada satu waktu. Dengan menggunakan metode orang yang memahami gambar sebagai referensi, kami dapat merancang filter untuk mengekstrak fitur lokal dari suatu gambar dan menerapkannya untuk memahami keseluruhan gambar.

Dengan kata lain, kami menggunakan filter untuk mewujudkan koneksi lokal antara lapisan input dan lapisan tersembunyi. Asumsikan filter 10 × 10 dirancang untuk meniru mata manusia untuk memvisualisasikan wilayah gambar lokal. Kemudian neuron lapisan tersembunyi terhubung ke area 10 × 10 dari lapisan input melalui filter. Lapisan tersembunyi memiliki 108 neuron tersembunyi, sehingga jumlah filter antara lapisan tersembunyi dan lapisan input adalah 108. Setiap koneksi sesuai dengan area 10 × 10 dari lapisan input melalui filter. Dengan demikian besaran parameter bobot koneksi antara input layer dan hidden layer menjadi 10 × 10 × 108 = 1010.

Sambil mengurangi jumlah bobot dari 25 × 1012 menjadi 1010 dengan mengubah konfigurasi penuh koneksi ke koneksi lokal, masalah bahwa parameter bobot terlalu banyak dan beban computing terlalu besar masih belum terpecahkan, karena ada fitur intrinsik dalam gambar alami. Dengan kata lain, fitur statistik untuk satu bagian gambar sama dengan fitur untuk bagian lain. Ini berarti fitur yang dipelajari dari satu bagian gambar juga dapat dimanfaatkan untuk bagian lain. Oleh karena itu, untuk semua lokasi gambar yang sama, kita dapat menggunakan fitur pembelajaran yang sama. Filter adalah matriks bobot dan mewakili fitur lokal dari suatu gambar.

Kemudian filter yang sama dapat digunakan untuk semua lokasi gambar secara alami. 108 filter 10 × 10 sesuai dengan 106 area berbeda dari 10 × 10 gambar. Jika filter benar-benar identik, yang berarti fitur lokal digunakan untuk seluruh gambar, ada 1 filter dengan 100 parameter bobot antara lapisan input dan lapisan tersembunyi. Dengan berbagi bobot, kuantitas parameter bobot berkurang dari 25 × 1012 menjadi 100, yang sangat mengurangi kuantitas parameter bobot dan beban computing. Gambar-gambar ini disediakan oleh Yann LeCun dan Marc'Aurelio Ranzato pada tahun 2013 [12].

Konsep koneksi lokal dan pembagian bobot yang kami gunakan mencapai operasi konvolusi. Filter 10 × 10 persis kernel konvolusi. Kernel konvolusi hanya mewakili satu jenis fitur lokal dari sebuah gambar. Ketika kita perlu merepresentasikan lebih banyak fitur lokal, kita dapat menggunakan beberapa kernel konvolusi. Jika lapisan konvolusi pertama pada Gambar 6.22 memiliki enam filter, maka kita dapat memperoleh enam peta fitur lapisan tersembunyi dengan operasi konvolusi.



Gambar 6.23 Diagram skema untuk JST convolutional (CNN) (sumber: http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial).

Contoh 6.5 Cara Kerja Konvolusi untuk Membangun Jaringan Neural Konvolusi

Kita dapat memahami bagaimana mewujudkan konvolusi melalui operasi konvolusi untuk gambar 8 × 8, seperti yang ditunjukkan pada Gambar 6.23. Dimensi kernel konvolusi adalah 4 × 4; dan matriks fiturnya adalah:

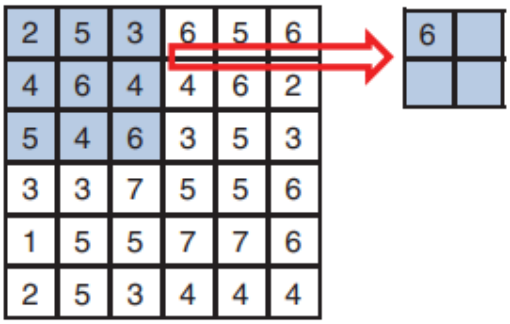
$$w = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Pertama, kami mengekstrak gambar x_1 dari 4 × 4 dari gambar 8 × 8 untuk operasi konvolusi dengan matriks fitur. Di sini, kita memperoleh nilai y_1 untuk neuron pertama di lapisan tersembunyi dengan menggunakan persamaan yi = w xi. Ukuran langkah konvolusi diatur ke 1. Kami melanjutkan mengekstraksi citra x_2 dari 4 × 4, dan memperoleh nilai y_2 untuk neuron kedua melalui operasi konvolusi. Kami mengulangi langkah-langkah di atas sampai melintasi keseluruhan gambar selesai. Setelah perhitungan untuk semua nilai neuron di lapisan tersembunyi selesai, peta fitur yang sesuai dengan kernel konvolusi diperoleh.

Biasanya, kami menghitung nilai peta fitur keluaran di lapisan tersembunyi dengan menggunakan fungsi aktivasi. Fungsi aktivasi yang sering digunakan adalah: fungsi sigmoid $[\sigma(x) = \frac{1}{1+e^{-x}}]$,

fungsi tangen hiperbolik $[\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}]$ dan fungsi [RELU(x) = max(0,x)].

Dengan asumsi jumlah peta fitur input lapisan konvolusi l adalah n, kita mengadopsi persamaan n $y_j = f(\sum_{i=1}^{n}(w_{ij} * x_i + b_i))$ untuk menghitung peta fitur output di lapisan konvolusi l, di mana b adalah singkatan dari deviasi , w untuk matriks bobot dan f untuk fungsi aktivasi.



Gambar 6.24 Konsep pooling dari grid 6 x 6 ke grid 2 x 2 (sumber: http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial)

Jika lapisan konvolusi l memiliki m filter, jumlah matriks bobot w adalah n × m sesuai dengan m filter, dan lapisan konvolusi l akan memiliki m outpnut−mn aps.

Jumlah neuron pada lapisan tersembunyi adalah $n_y = (\left|\frac{n_{l-1} - n_k}{s}\right| + 1)$ x $(\left|\frac{n_{l-1} - n_k}{s}\right| + 1)$ x m , di mana dimensi data input adalah n_{l-1}x m_{l-1} , ukuran saringannya adalah n_k x m_k , ukuran langkah konvolusi adalah s (jarak konvolusi bergerak pada setiap waktu) dan jumlah filter adalah m. Seperti yang ditunjukkan pada Gambar 6.23, item diagram skematik konvolusi adalah ukuran data input 8 × 8, jendela konvolusi 4 × 4, ukuran konvolusi 1 langkah, 1 peta fitur dan jumlah neuron di lapisan tersembunyi adalah

$$n_l = \left(\left|\frac{8-4}{1}\right| + 1\right) \times \left(\left|\frac{8-4}{1}\right| + 1\right) \times 1 = 5 \times 5$$

Penyatuan di CNN

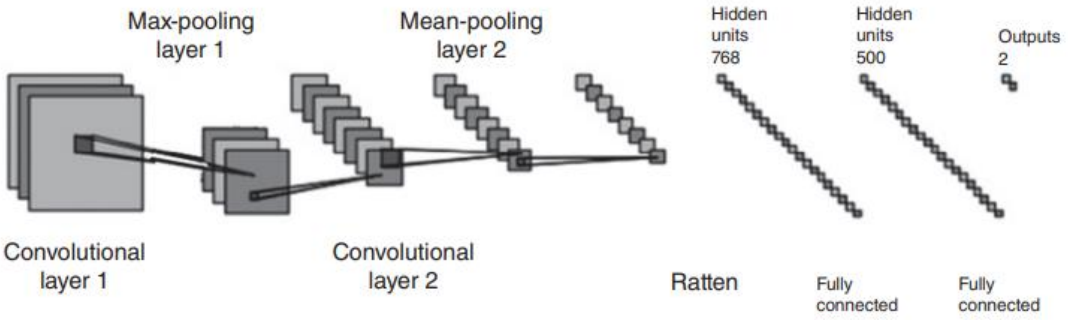
Sebuah peta fitur atau fitur dari suatu gambar diperoleh dengan cara konvolusi. Namun saat menggunakan fitur tersebut untuk melatih pengklasifikasi secara langsung, kami menghadapi tantangan beban computing yang sangat besar. Misalnya, untuk gambar 96 × 96 piksel, dengan asumsi 400 filter konvolusi digunakan, dimensi konvolusi adalah 8 × 8, dan setiap grafik fitur mencakup (96 8 + 1) × (96 8 + 1)= 892 = 7921-dimensi fitur konvolusi. Karena terdapat 400 filter, setiap sampel citra masukan akan memperoleh 892 × 400 = 3.168.400 neuron tersembunyi. Ini mungkin melibatkan overhead computing yang berat.

Gambar umum memiliki atribut yang bersifat statis. Fitur-fitur yang berguna di satu area gambar sangat mungkin untuk diterapkan di area lain. Oleh karena itu, untuk menggambarkan gambar besar, kami dapat melakukan statistik agregasi untuk fitur di lokasi yang berbeda. Misalnya, orang dapat menghitung nilai rata-rata (atau nilai maksimum) di area gambar. Fitur

statistik yang diperoleh dengan agregasi tersebut tidak hanya mengurangi dimensi tetapi juga meningkatkan hasil (dengan mencegah over-fitting). Operasi dari agregasi tersebut disebut pooling. Sesuai metode computing yang berbeda, ini dibagi menjadi penyatuan rata-rata dan penyatuan maksimum. Gambar 6.24 menunjukkan operasi penyatuan 3 × 3 untuk gambar 6 × 6; gambar dibagi menjadi empat area yang tidak saling tumpang tindih. Gambar 6.24 menunjukkan hasil setelah pooling maksimum di satu area. Grafik fitur setelah pooling adalah 2 × 2.

Contoh 6.6 Konvolusi dan pooling untuk CNN

Dalam beberapa tahun terakhir, CNN telah banyak digunakan dalam pengolahan citra digital dengan pesatnya perkembangan CNN. Misalnya, menggunakan jaringan saraf convolutional DeepID ini, tingkat pengenalan wajah manusia yang benar dapat mencapai maksimum 99,15%. Teknik ini dapat memainkan peran penting dalam pencarian orang hilang dan pencegahan kejahatan teroris. Gambar 6.25 menunjukkan CNN yang digunakan.



Gambar 6.25 Konvolusi dan pooling untuk CNN

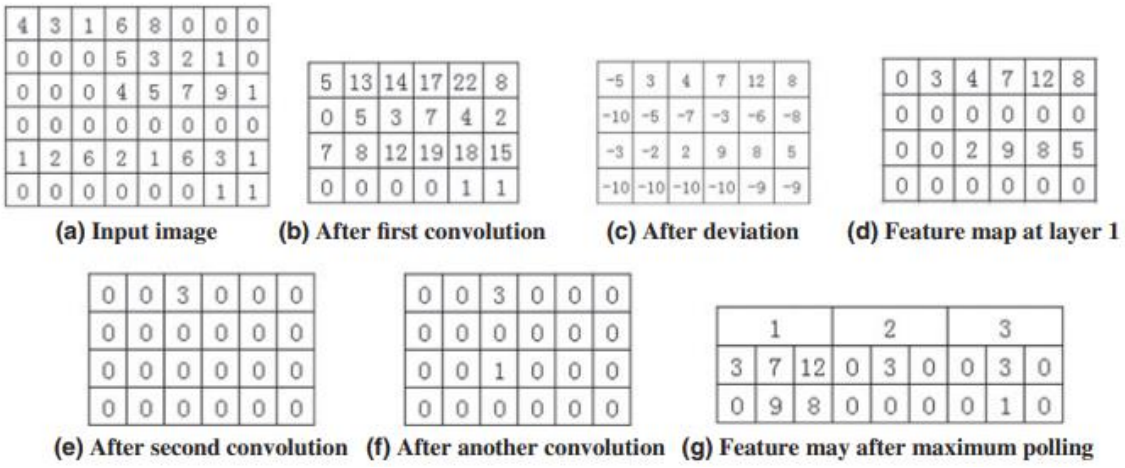
Jika citra masukan yang diberikan adalah seperti yang ditunjukkan pada Gambar 6.26(a), ukuran citra adalah 6 × 8. Kami mengadopsi ukuran kernel konvolusi sebagai 3 × 3 dan ukuran satu grafik fitur pada lapisan konvolusi 1 sebagai ((6 3) + 1) × ((8 3) + 1) = 4 × 6. Dengan asumsi kita menggunakan tiga filter, matriks bobot yang sesuai adalah:

$$w_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad w_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad w_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Asumsikan deviasi b = –10, fungsi aktivasi RELU(x) = max(0, x). Untuk mendapatkan grafik fitur dari lapisan konvolusi 1, kita memerlukan operasi berikut:

Langkah 1: Gunakan w_1 untuk melakukan konvolusi data input. Hasilnya ditunjukkan pada Gambar 6.26(b).

Langkah 2: Gambar 6.26(c) menunjukkan hasil dengan tambahan deviasi.



Gamnar 6.26 Hasil dengan tambahan deviasi

Langkah 3: Setelah aktivasi, kami memperoleh peta fitur 1 dari lapisan konvolusi 1, seperti yang ditunjukkan pada Gambar 6.27(d).

Langkah 4: Mengulangi langkah-langkah di atas pada matriks bobot w_2 dan w_3, kami memperoleh peta fitur 2 dan 3 dari lapisan konvolusi 1, seperti yang ditunjukkan pada Gambar 6.27(e) dan (f).

Untuk mendapatkan grafik fitur dari max-pooling layer 1, kami memilih nilai maksimum dari setiap area 2 × 2 yang tidak tumpang tindih di setiap peta fitur keluaran dan melakukan operasi max-pooling 2 × 2. Gambar 6.27(g) menunjukkan peta fitur yang dihasilkan setelah max-pooling pada layer 1.

Jaringan Saraf Konvolusi Dalam

Faktanya, pembelajaran jaringan saraf convolutional adalah untuk mendapatkan hubungan pemetaan antara input dan output melalui sejumlah besar pembelajaran data dengan tag. Tetapi hubungan pemetaan antara input dan output ini sulit diungkapkan dengan presisi matematis. Ketika pelatihan hubungan pemetaan tersebut selesai, data keluaran yang sesuai dapat diperoleh melalui pemetaan dengan memasukkan data ke dalam jaringan.

Pelatihan jaringan saraf convolutional mirip dengan jaringan BP. Ini dibagi menjadi dua tahap, satu adalah propagasi maju untuk menghitung output dan yang lainnya adalah propagasi mundur untuk penyesuaian parameter dengan kesalahan. Algoritma untuk propagasi maju ditunjukkan pada Algoritma 6.4. Selama propagasi maju, kami memasukkan data sampel x ke dalam jaringan saraf convolutional, melakukan operasi pada data input lapis demi lapis, menjadikan data keluaran dari lapisan sebelumnya sebagai data input lapisan berikutnya, dan akhirnya mendapatkan hasil keluaran dari y’.

Selama propagasi mundur, kami menghitung kesalahan antara tag kanan y dari data input x dan hasil keluaran y’, mengirimkan kesalahan dan menyesuaikan parameter lapis demi lapis

dari lapisan keluaran hingga mencapai lapisan input. Langkah-langkah detailnya adalah sebagai berikut:

Langkah 1: y adalah singkatan dari tag data sampel, y' adalah keluaran melalui perhitungan, hitung kesalahan : O = y - y' . Hitung fungsi biaya : $E = \frac{1}{2}\sum_{N}\sum_{K}(y-y')^2$, N singkatan dari jumlah sampel, K singkatan dari kelas.

Langkah 2: Hitung turunan kesalahan E ke bobot w: $\frac{\partial E}{\partial w}$ E lapis demi lapis dari keluaran ke lapis masukan, perbarui bobot dengan menggunakan $w = w + \Delta w = w + \lambda\frac{\partial E}{\partial w}$.

Langkah 3: Dengan menggunakan Algoritma 6.4 dari baris 2 ke baris 15, hitung y' .

Langkah 4: Tentukan apakah kondisi terminal dari propagasi mundur tercapai; jika tidak, lakukan Langkah 1, atau akhiri pelatihan.

Pengaturan untuk Setiap Lapisan Eksperimen Gambar LeNet-5

Data masukan adalah data gambar 32 × 32. Lapisan C1 adalah lapisan konvolusi dan 6 filter 5 × 5 diadopsi untuk operasi konvolusi. Ukuran langkah konvolusi adalah 1, dan 6 grafik fitur 28 × 28 diperoleh. Setiap neuron dalam grafik fitur terhubung ke area input 5 × 5. Ada 156 parameter pada layer C1 untuk pelatihan: ada 25, yaitu parameter bobot 5 × 5 dan satu parameter deviasi untuk setiap filter, dan ada enam filter. Jadi ada 156, yaitu (25 + 1) × 6 parameter secara total. Algoritma diberikan di bawah ini:

Algoritma 6.4 Propagasi maju di CNN

Input: x: data sampel

Output: y': keluaran melalui perhitungan

Prosedur:

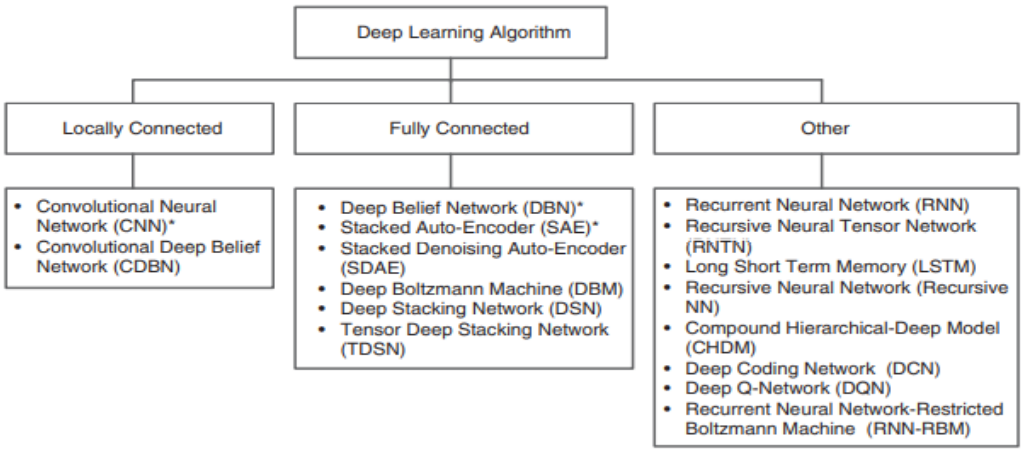
1) Lakukan inisialisasi ke semua parameter dalam n lapisan
2) p_1= x
3) untuk I = 1,2,…n
4) jika saya adalah lapisan konvolusi
5) Melakukan operasi konvolusi untuk p_i, dan mengeluarkan q_i 6)
6) i++;
7) $p_i = q_{i-1}$
8) berakhir jika
9) jika saya mengumpulkan lapisan
10) Melakukan operasi pooling untuk pi, dan output qi 11)
11) i++;
12) $p_i = q_{i-1}$
13) berakhir jika
14) akhir
15) Keluarkan y' melalui perhitungan

Lapisan S2 adalah lapisan penyatuan. Kami melakukan operasi maxpooling 2 × 2 untuk 6 grafik fitur untuk output lapisan C1. Kemudian diperoleh 6 grafik fitur 14 × 14. Sedangkan untuk lapisan C3, kami melakukan operasi konvolusi ke lapisan keluaran S2 dengan 16 kernel konvolusi 5 × 5. Dengan demikian, 16 grafik fitur dengan 10 × 10 neuron diperoleh. Setiap grafik fitur di lapisan C3 terhubung ke semua 6 grafik fitur, yang merupakan output dari lapisan S2.

Lapisan S4 adalah lapisan penyatuan. Kami melakukan operasi maxpooling 2 × 2 untuk 16 grafik fitur yang merupakan output dari lapisan C3. Kemudian diperoleh 16 graf fitur 5 × 5. Lapisan C5 juga merupakan lapisan konvolusi dan mencakup 120 filter 5 × 5, dan 120 grafik fitur dapat diperoleh. Setiap saraf dalam grafik fitur terhubung ke area 5 × 5 di semua 16 unit di lapisan S4. Ada 84 neuron (dalam desain lapisan keluaran, jumlah neuron diatur ke 84) di lapisan F6, dengan koneksi penuh ke lapisan C5. Sama seperti dalam jaringan saraf klasik, produk titik antara vektor input dan vektor bobot dihitung di lapisan F6, dan offset juga ditambahkan. Kemudian mengirimkan hasilnya ke fungsi sigmoid untuk menghasilkan keadaan Unit i.

Jaringan *Deep learning* Lainnya

Arsitektur *Deep learning* mencakup lapisan input, lapisan output, dan beberapa lapisan tersembunyi. Arsitektur *Deep learning* berarti jaringan netral yang dalam, yang memiliki propagasi maju dan pembelajaran terbalik. Ada banyak jenis arsitektur *Deep learning*. Sebagian besar arsitektur ini digunakan untuk mengubah arsitektur umum. Di sini kita membagi arsitektur *Deep learning* menjadi tiga jenis menurut model koneksionis dari neutron: sepenuhnya terhubung, terhubung secara lokal, dan banyak jaringan *Deep learning* lainnya, seperti yang ditunjukkan pada Gambar 6.27.



Gambar 6.27 Klasifikasi arsitektur berbagai model *Deep learning*. (Yang ditandai dengan * tercakup dalam buku ini.)

Konektivitas Jaringan Deep learning

- **Jaringan yang terhubung penuh:** Dalam jaringan netral tradisional, koneksi antara lapisan dari lapisan input, lapisan tersembunyi ke lapisan output terhubung sepenuhnya. Satu

neutron di lapisan sebelumnya terhubung dengan setiap neuron di lapisan berikutnya. Arsitektur *deep learning* yang terhubung sepenuhnya mencakup Deep Belief Network (DBN), Deep Boltzmann Machine (DBM), Stacked Auto-Encoder (SAE), Stacked Denoising Auto-Encoder (SDAE), Deep Stacking Network (DSN) dan Tensor Deep Stacking Network (TDSN)).

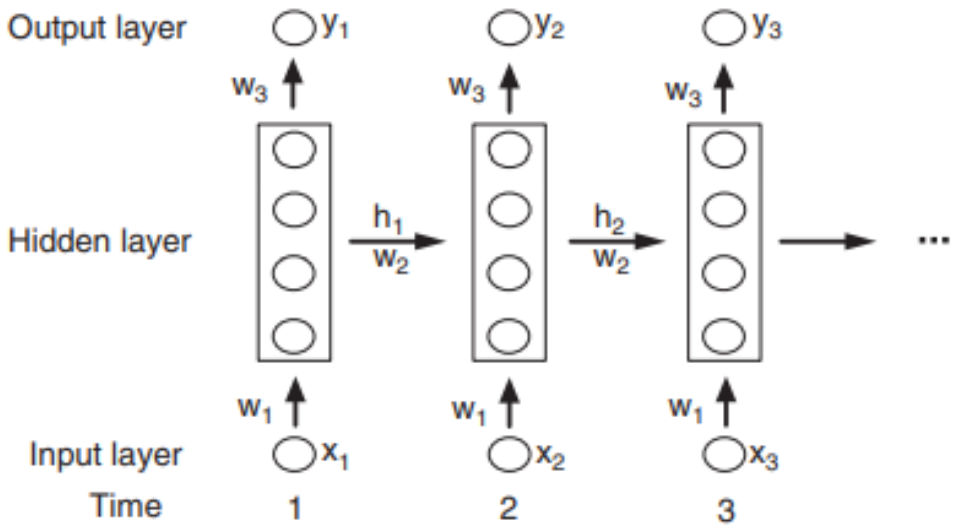
- **Jaringan yang terhubung secara lokal:** Arsitektur *Deep learning* yang terhubung secara lokal berarti mode koneksi antara lapisan input dan lapisan output terhubung secara lokal. Arsitektur *deep learning* semacam ini mengambil Convolutional Neural Network (CNN) sebagai kelas representatif. Ini menggunakan konsep koneksi parsial dan pembagian bobot operasi konvolusi untuk menggambarkan fitur lokal secara keseluruhan. Dengan demikian, ini sangat mengurangi jumlah berat. Convolutional Deep Belief Networks (CDBN) juga terhubung secara lokal.
- **Lainnya:** Kelas ini mencakup Recurrent Neural Network (RNN), Recursive Neural Tensor Network (RNTN) dan Long Short Term Memory (LSTM), dll. Jaringan terkait lainnya termasuk Recurrent Neural Network-Restricted Boltzmann Machine (RNN-RBM).), Deep Q-Network (DQN), Compound Hierarchical-Deep Models (CHDM) dan Deep Coding Network (DPCN), dll. ANN konvensional, yang tersambung secara lokal atau tersambung sepenuhnya, mungkin memiliki penerapan yang terbatas, karena kinerjanya buruk atau menjadi tidak berdaya saat berurusan dengan aliran data. Kami memperkenalkan jaringan saraf berulang (RNN) sebagai berikut:

Jaringan Saraf Berulang (RNNs)

Sebuah RNN sesuai dengan JST dengan tiga lapisan pada setiap titik waktu, sehingga pelatihan ini mirip dengan JST tradisional. Namun, RNN menganggap keluaran arus dari aliran data juga terkait dengan keluaran sebelumnya. Itu berarti pemrosesan informasi pada saat ini perlu mempertimbangkan output dari waktu terakhir. Melatih RNN satu lapis untuk 100 langkah waktu setara dengan melatih jaringan umpan maju dengan ratusan lapisan, seperti yang ditunjukkan pada Gambar 6.28. Ketika RNN memproses urutan data, keluaran sebelumnya akan memberikan umpan balik sebagai bagian dari data masukan. RNN harus mengingat output sebelumnya untuk menghitung output saat ini secara iteratif. Ada koneksi antara node lapisan tersembunyi dalam struktur jaringan. Input dari lapisan tersembunyi perlu menggunakan output dari lapisan input dan output itu sendiri secara iteratif.

Sebagian besar jaringan *deep learning* adalah jaringan feedforward, seperti SAE, dan DBN, yang berarti aliran proses sinyal dalam satu lapisan searah dari input ke output. Tidak seperti jaringan saraf feedforward, RNN menerima urutan input dan juga menghasilkan nilai urutan sebagai output. RNN adalah jaringan saraf termasuk perilaku temporal. Ini menyiratkan bahwa output dari urutan diumpankan sebagai input ke input berikutnya. Daya tarik RNN adalah untuk memodelkan bahasa atau proses pengenalan suara. Node di antara lapisan tersembunyi tidak

lagi tidak terhubung tetapi terhubung, dan input dari lapisan tersembunyi tidak hanya mencakup input saat ini tetapi juga output dari lapisan tersembunyi terakhir kali.

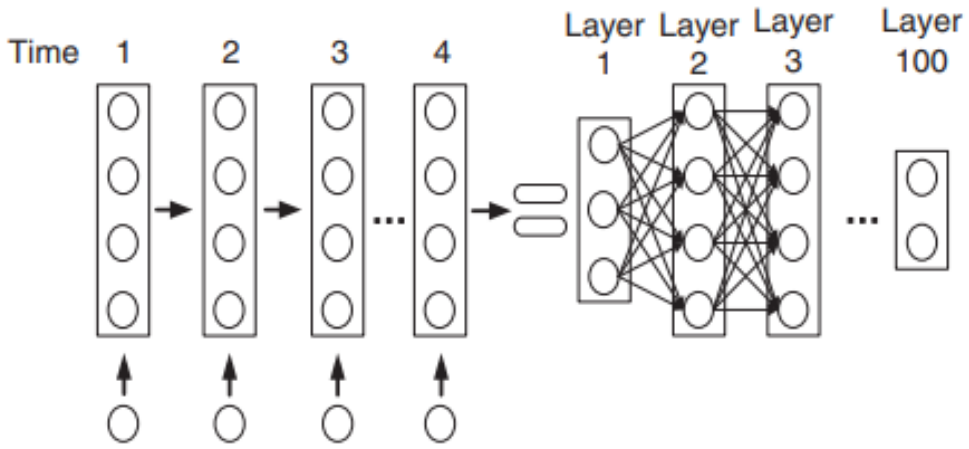


Gambar 2.28 Struktur jaringan saraf berulang (RNN).

Pada setiap titik waktu t, RNN sesuai dengan JST dengan tiga lapisan. Input dan output RNN pada waktu t direpresentasikan sebagai xt dan yt', masing-masing, dan lapisan tersembunyi direpresentasikan sebagai ht. Kami menggunakan parameter jaringan yang sama (w1, w2, w3) setiap saat, di mana bobot koneksi antara input dan lapisan tersembunyi adalah w1, bobot antara lapisan tersembunyi waktu t – 1 dan lapisan tersembunyi waktu t adalah w2, dan bobot antara lapisan tersembunyi dan lapisan keluaran adalah w3, seperti yang diilustrasikan pada Gambar 6.29. Perhitungan maju dilakukan sebagai berikut: input pada waktu t adalah xt, nilai ht dari lapisan tersembunyi dihitung dengan nilai input saat ini xt dan nilai ht−1 dari lapisan tersembunyi pada waktu t – 1, maka output nilai y' diperoleh dengan memperhatikan ht sebagai input dari lapisan output. Gambar 6.29 mencoba mengungkapkan perbedaan antara RNN dan ANN.

Jaringan Tensor Saraf Rekursif (RNTN)

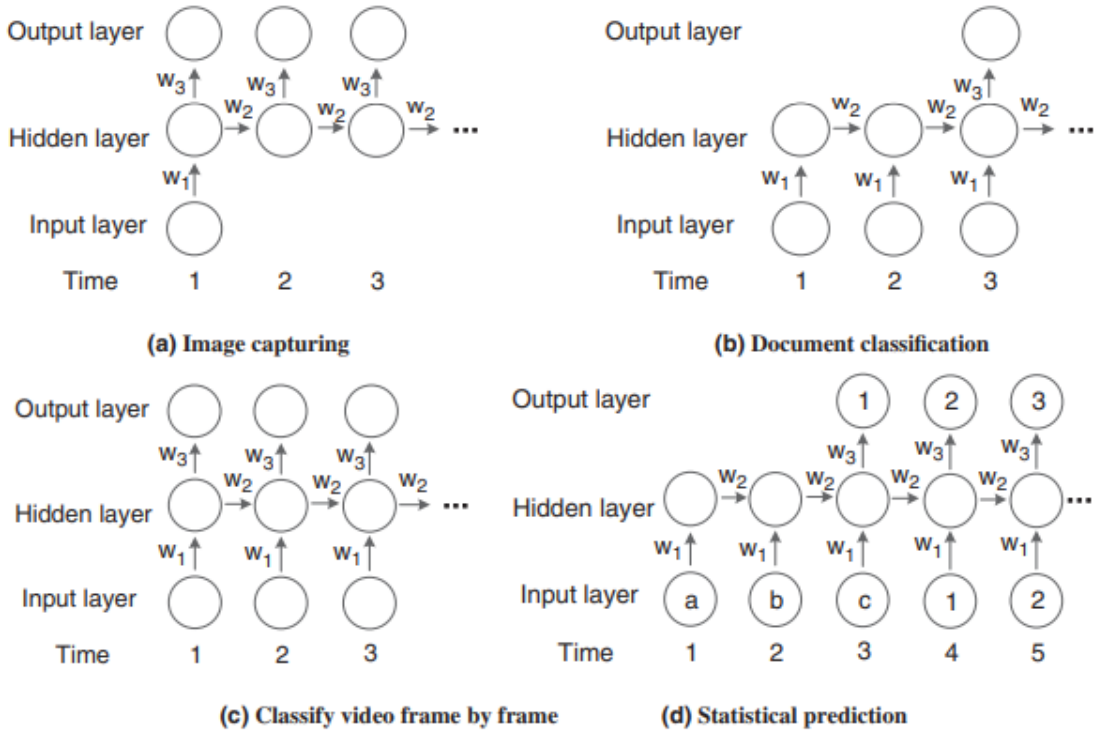
Jaringan tensor saraf rekursif (RNTN) memiliki struktur jaringan saraf dalam yang rekursif. Kelas ini disarankan untuk memproses data input dengan panjang variabel atau membuat prediksi multistage. RNTN dan RNN keduanya memiliki perilaku rekursif [22]. Mereka berbeda sebagai berikut: RNN adalah rekursi dari urutan waktu dan RNTN adalah rekursi dari struktur data, yang disebut tensor (akan dipelajari di Bab 9).



Gambar 6.29 Kontras antara arsitektur RNN dan ANN.

Hubungan Input dan Output di Jaringan Saraf Berbeda

RNN menerima urutan input dan juga menghasilkan urutan output. Menurut aplikasi yang berbeda, ada berbagai bentuk pasangan input atau output, yang ditunjukkan pada Gambar 6.30 dalam empat kasus. Pada Gambar 6.30(a), struktur I/O sangat menarik untuk aplikasi pengambilan gambar. Gambar 6.30(b) menunjukkan struktur I/O memiliki banyak input dengan satu output, yang cocok dengan klasifikasi dokumen.



Gambar 6.30 Berbagai bentuk input atau output yang diterapkan dalam aplikasi *Deep learning* yang berbeda.

Pada Gambar 6.30(c), baik input maupun output ditampilkan secara berurutan. Ini dipraktikkan di RNN untuk aplikasi streaming video, bingkai demi bingkai. Arsitektur ini juga cocok untuk prediksi statistik situasi masa depan. Pada Gambar 6.30(d), kami memasukkan data yang diketahui pada waktu 1 dan waktu 2, dan prediksi dimulai pada waktu 3. Setelah memasukkan data pada waktu 3, kami mendapatkan hasil output 1. Ini menyiratkan bahwa kami telah memprediksi data pada waktu berikutnya sebagai 1. Dengan cara yang sama, kita mendapatkan hasil keluaran 2 setelah memasukkan data 1 pada waktu 4 dan memprediksi data pada waktu 5 sebagai 2.

Jaringan Neural Deep learning Lainnya

- **Convolutional Deep Belief Networks:** (CDBN) adalah struktur jaringan yang menggabungkan CNN dengan DBN. Ini dapat digunakan untuk memecahkan masalah perluasan DBN untuk memproses gambar dengan ukuran penuh dan dimensi tinggi.
- **Deep Q-Networks (DQN):** DQN adalah struktur jaringan saraf dalam, yang diajukan oleh Google Deep-Mind. Ini menggabungkan metode pembelajaran penguatan Q-learning dan jaringan saraf tiruan.
- **Deep Boltzmann Machines (DBM):** (DBM) mencakup satu lapisan sel yang terlihat dan serangkaian lapisan sel yang tersembunyi. Tidak ada koneksi di antara lapisan yang sama. DBM adalah struktur dalam yang menumpuk beberapa RBM, dan ada koneksi tidak langsung antara dua lapisan.
- **Stacked Denoising Auto-Encoder (SDAE):** Struktur Stacked Denoising Auto-Encoder (SDAE) mirip dengan auto-encoder bertumpuk. Satu-satunya perbedaan adalah mengubah AE menjadi Denoising Auto-Encoder (DAE). DAE menggunakan metode pelatihan tanpa pengawasan, yang mencakup tiga langkah, yaitu korup, encoder dan decoder.
- **Deep Stacking Network (DSN):** DSN menggunakan modul jaringan saraf sederhana untuk menumpuk jaringan dalam, dan jumlah modul tidak pasti. Output dari setiap modul adalah sebuah kategori. Masukan modul pada lapisan pertama adalah data awal. Dari lapisan kedua, input modul adalah koneksi seri data awal x dan output y dari lapisan sebelumnya.
- **Tensor Deep Stacking Network (TDSN):** TDSN ini merupakan perpanjangan dari Deep Stacking Networks (DSN). Ini mencakup banyak blok bertumpuk. Setiap blok yang ditumpuk mencakup tiga lapisan, yaitu lapisan input x, dua lapisan tersembunyi paralel h1 dan h2, dan lapisan output y. Long Short Term Memory (LSTM): (LSTM) adalah peningkatan dari RNN, yang menambahkan modul memori di lapisan tersembunyi dari RNN dasar. LSTM dapat memecahkan masalah pengaruh lemah lapisan tersembunyi di titik waktu sebelumnya ke lapisan tersembunyi di titik waktu berikutnya saat menggunakan RNN dasar untuk pelatihan.

- **Deep Coding Networks (DPCN):** (DPCN) adalah model generatif hierarkis, yang merupakan jaringan *Deep learning* yang dapat menggunakan data konteks untuk mewujudkan pembaruan mandiri.
- **Compound Hierarchical-Deep Model (CHDM):** Model ini terdiri dari jaringan dalam dengan model Bayesian non-parametrik. Fitur dapat dipelajari menggunakan arsitektur yang dalam seperti DBN, DBM, auto encoder bertumpuk, dll.
- **Recurrent Neural Network-Restricted Boltzmann Machine (RNN-RBM):** (RNN-RBM) adalah sejenis RBM temporal berulang.

6.5 KESIMPULAN

Dalam bab ini, kami memperkenalkan *Deep learning* secara rinci, termasuk konsepnya, algoritme *Deep learning* yang representatif, dan proses pelatihan. Pada Bagian 6.1, melalui perbandingan *Deep learning* dan pembelajaran dangkal, konsep *Deep learning* mudah dipahami. Di Bagian 6.2, kami memperkenalkan dasar *Deep learning*, yaitu jaringan saraf tiruan. Kemudian, algoritme *Deep learning* yang populer, seperti SAE, DBN, dan CNN, dirinci di Bagian 6.3 dan 6.4. Metode *Deep learning* tambahan akan dibahas di Bab 9, di mana aplikasi *Deep learning* disajikan dengan platform perangkat lunak TensorFlow dalam aplikasi DeepMind.

Tugas dan Latihan

1. Tabel 6.4 menunjukkan kumpulan data bunga Iris dengan hanya dua spesies: setosa (dilambangkan dengan “1”), dan versicolor (dilambangkan dengan “0”). Kita dapat membedakan antara item data ini dengan panjang petal, lebar petal, panjang sepal dan lebar sepal. Rancang model JST untuk mengklasifikasikan bunga Iris menggunakan pendekatan clustering:
    1) Hitung jumlah neuron di lapisan input jaringan saraf, dan gambarkan fitur bunga mana yang masing-masing diwakili oleh neuron ini.
    2) Jelaskan jumlah neuron pada lapisan keluaran, dan bagaimana neuron ini mewakili kategori bunga.
    3) Cukup menggambarkan proses pelatihan dan klasifikasi dan menginterpretasikan hasil akhir pengelompokan.

Tabel 6.4 Data sampel bunga Iris yang dikarakterisasi dalam empat atribut.

nomor identitas	Panjang kelopak	Lebar kelopak	Panjang sepal	Lebar Sepal	Jenis
1	5.1	3.5	1.4	0.2	1
2	7.0	3.2	4.7	1.4	0
3	5.2	3.4	1.6	0.3	1

2. Ada enam mata pelajaran pilihan untuk siswa di sekolah yang jumlahnya adalah (1, 2, 3, 4, 5, 6). Nilai untuk setiap kursus direpresentasikan sebagai (A, B, C). Bangun model DBN untuk memprediksi nilai beberapa siswa, dan nilai mata kuliah sebagai input jaringan, kemudian memprediksi nilai mata kuliah keenam. Mengingat data input setiap neuron pada input layer DBN adalah 0 atau 1, maka:
   1) Rancang metode input untuk nilai mata kuliah siswa, dan jelaskan bagaimana metode ini mewujudkan input nilai siswa. Dan jika demikian, berapa banyak neuron di lapisan visual?
   2) Jika nilai Jerry adalah (A, B, A, C, B), tuliskan data input yang sesuai.
3. Banyak sensor gas menghadapi masalah sensitivitas silang, yaitu menggunakan sensor gas seringkali tidak dapat secara akurat mendeteksi keberadaan gas beracun. Kita dapat memecahkan masalah dengan menggunakan serangkaian sensor untuk mendeteksi karakteristik sensitivitas silang melalui jaringan saraf tiruan. Tabel 6.5 menunjukkan tiga macam pengukuran sensor gas. Kondisi gas1 mengacu pada keberadaan gas dan 0 berarti tidak ada gas. Rancang model JST dengan data yang diberikan untuk membedakan jenis gas yang dicirikan oleh X1:0.4, X2:0.5, X3:0.4.
4. Sensor gas semikonduktor adalah sensor yang resistansinya akan berubah dengan adanya gas. Kita dapat menentukan konsentrasi gas sesuai dengan perubahan rasio resistensi (sensitivitas). Namun, resistansi sensor semikonduktor juga akan bervariasi dengan suhu dan kelembaban. Dataset dari 16 kelompok sesuai dengan sensitivitas sensor diberikan pada Tabel 6.6. Gunakan dataset ini untuk merancang model jaringan syaraf tiruan untuk berspekulasi konsentrasi gas dengan X1:28, X2:50 dan X3:0.4.

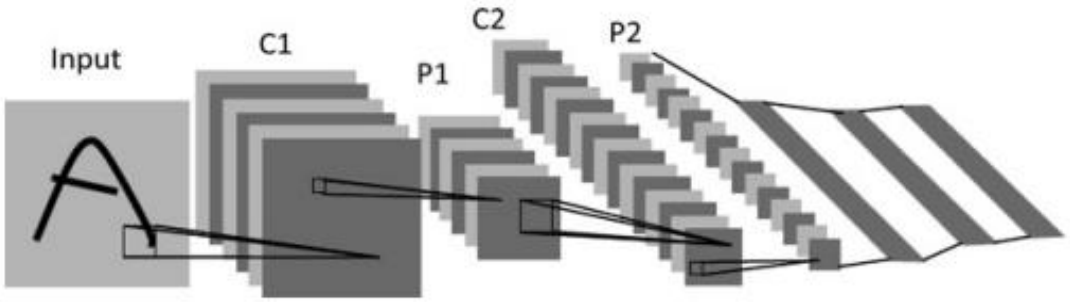
Tabel 6.5 Data sensor gas dan kondisi gas yang sesuai.

Sensitivitas (X_1)	Sensitivitas (X_2)	Sensitivitas (X_3)	Gas A (Y_1)	Gas A (Y_2)
0.63	0.56	0.68	1	0
0.55	0.44	0.65	0	1
0.46	0.78	0.64	0	1
0.37	0.55	0.44	1	1
0.58	0.43	0.33	1	0
0.65	0.79	0.35	0	0
0.89	0.35	0.40	0	1
0.58	0.99	0.36	0	1
0.54	0.89	0.32	1	1
0.40	0.55	0.31	1	0
0.69	0.38	0.39	1	0

Tabel 6.6 Suhu dan kelembaban dan konsentrasi gas untuk mengukur sensitivitas sensor.

Suhu (X₁)	Kelembaban udara (X₂)	Sensitivitas (X₃)	Konsentrasi gas (Y)
20	45	0.50	20
22.5	60	0.46	23
23.0	57	0.43	33
21.5	57	0.44	34
26.5	64	0.33	45
28.5	59	0.35	44
23.0	37	0.40	41
26.0	66	0.36	47
29.5	72	0.32	45
35.0	83	0.31	48
30.0	76	0.29	56
20.0	45	0.45	39
22.5	77	0.39	40
23.0	57	0.35	52
21.8	46	0.39	48
24.8	67	0.32	51

5. Gunakan jaringan saraf convolutional (Gambar 6.31) untuk melakukan klasifikasi citra. Gambar memiliki lapisan input dengan resolusi 32 × 32, dan jaringan mencakup lapisan konvolusi (C1) dan lapisan maxpooling (P1). Ikuti langkah-langkah di bawah ini untuk melakukan tugas klasifikasi.

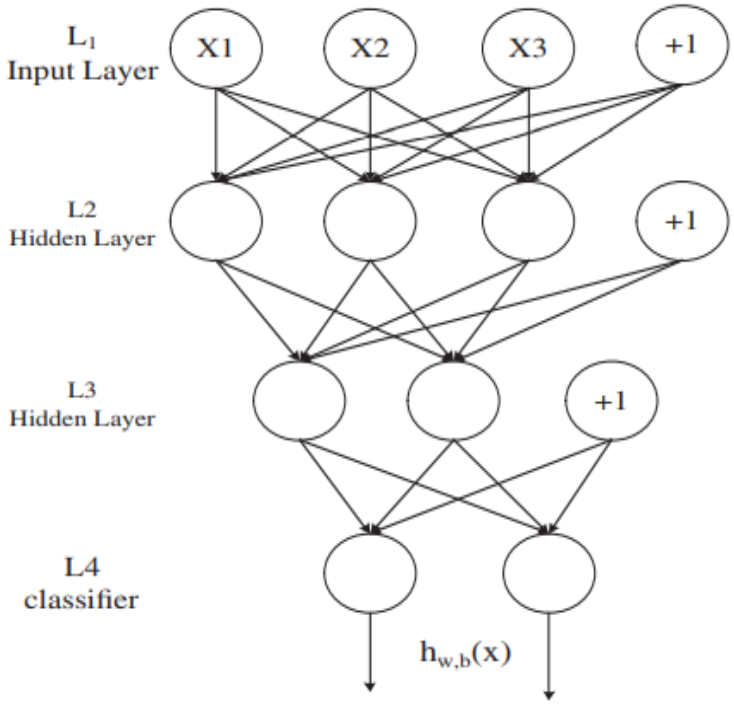


Gambar 6.31 Diagram struktur jaringan saraf konvolusi untuk klasifikasi citra.

1) Dengan asumsi ukuran kernel konvolusi C1 di CNN adalah 5 × 5, langkahnya adalah 3 dan jumlah peta fitur adalah 6, hitung ukuran setiap peta fitur di C1.
2) Luas pooling area di P1 adalah 2 × 2; menghitung ukuran setiap peta fitur di lapisan penyatuan.

3) Dengan asumsi ukuran kernel konvolusi C1 di CNN adalah 3 x 3, langkahnya adalah 1 dan jumlah peta fitur adalah 6, hitung ukuran setiap peta fitur di C1.
4) Luas pooling area di P1 adalah 3 × 3; menghitung ukuran setiap peta fitur di lapisan penyatuan.

6. Dalam pengenalan angka tulisan tangan, penggunaan Stacked AutoEncoder menghasilkan hasil dengan akurasi tinggi. Untuk menyederhanakan proses pelatihannya, kami memberikan gambar struktur sederhana dari Stacked AutoEncoder, seperti yang ditunjukkan pada Gambar 6.32. Struktur yang diberikan mencakup satu lapisan input (L1), dua lapisan tersembunyi (L2 dan L3) dan satu pengklasifikasi (L4). Dataset masukan mencakup sejumlah besar data tidak berlabel dan sejumlah kecil data berlabel. Dengan menggunakan penyetelan halus parameter yang diawasi, hitung jumlah parameter yang sesuai dengan jaringan dan tulis langkah-langkah pelatihan jaringan Stacked AutoEncoder ini.



Gambar 6.32 Diagram Stacked AutoEndcoder.

7. Pertimbangkan jaringan pengenalan angka tulisan tangan dengan tiga lapisan: lapisan konvolusi, lapisan max-pooling dan lapisan output. Buatlah program untuk menghitung grafik fitur setelah lapisan konvolusi dan lapisan max-pooling. Matriks karakteristik dari lapisan konvolusi harus dirancang. Ukuran area kolam adalah 2 × 2, dan matriks input 8 × 8, seperti yang ditunjukkan pada Tabel 6.7.

Tabel 6.7 Input data citra matriks 8×8.

5	3	17	8	34	137	45	0
0	20	0	0	204	13	0	6
4	0	0	253	0	0	0	2
0	0	198	0	5	0	3	0
6	186	0	146	0	7	0	2
0	139	0	0	176	0	0	0
0	157	0	0	154	0	2	0
4	0	173	182	0	0	0	0

8. Ada 6 neuron di lapisan visual dan 4 neuron di lapisan tersembunyi di RBM. Mengingat vektor input neuron di lapisan visual sebagai (0, 1, 1, 0, 1, 0), bobot koneksinya adalah

$$w = \begin{pmatrix} -0.2 & -0.1 & -0.4 & 0 \\ -0.1 & -0.3 & 0.2 & 0.3 \\ 0.6 & 0.3 & 0 & -0.6 \\ -0.3 & 0.5 & 0.4 & 0.0 \\ -0.1 & 0.3 & -0.9 & 0 \\ -0.1 & 0 & -0.4 & 0 \end{pmatrix}$$

Bias lapisan visual adalah bv = (0,1, 0,3, 0,2, 0, 0,1, 0) dan bias lapisan tersembunyi adalah bh = (0,1, 0,2, 0,1, 0). Bilangan acak adalah 0,6. Hitung nilai neuron menurut bilangan acak dan probabilitas nilai neuronnya adalah 1. Gunakan Persamaan (6.14) untuk menghitung nilai neuron pada lapisan tersembunyi berdasarkan nilai input pada lapisan visual. Berdasarkan perhitungan nilai neuron pada lapisan tersembunyi, gunakan Persamaan (6.15) untuk menghitung neuron pada lapisan keluaran.

BAGIAN 3
ANALISIS *BIG DATA* DI BIDANG KESEHATAN DAN PEMBELAJARAN KOGNOTIF
BAB 7
MACHINE LEARNING UNTUK *BIG DATA* DI BIDANG KESEHATAN

7.1 MASALAH KESEHATAN DAN ALAT *MACHINE LEARNING*

Bab ini dikhususkan untuk aplikasi analitik prediktif dalam perawatan kesehatan dan deteksi penyakit. Pertama, kami meninjau beberapa sistem perawatan kesehatan yang didukung IoT. Kami fokus pada pemantauan kesehatan dan sistem promosi latihan fisik. Persyaratan domain medis ini dinilai terlebih dahulu. Kemudian, kami mempresentasikan sistem solusi analitik menggunakan teknik *Machine Learning* yang didukung oleh cloud, perangkat seluler, dan sumber daya IoT.

Masalah Deteksi Kesehatan dan Penyakit Kronis

Pada tahun 2015, Organisasi Kesehatan Dunia (WHO) merilis laporan di seluruh dunia tentang penuaan dan kesehatan. Laporan ini mengangkat keprihatinan serius tentang penuaan populasi global. Penduduk berusia di atas 60 tahun diproyeksikan meningkat dari 12% pada tahun 2015 menjadi 22% pada tahun 2050. Dengan kecepatan tumbuh dua kali lipat, jumlah lansia berusia 60 tahun ke atas akan mencapai 2 miliar selama 35 tahun ke depan. Situasi di negara-negara Asia bahkan lebih buruk, misalnya, Jepang akan memiliki 30% lansia dalam populasinya dalam 10 tahun ke depan.

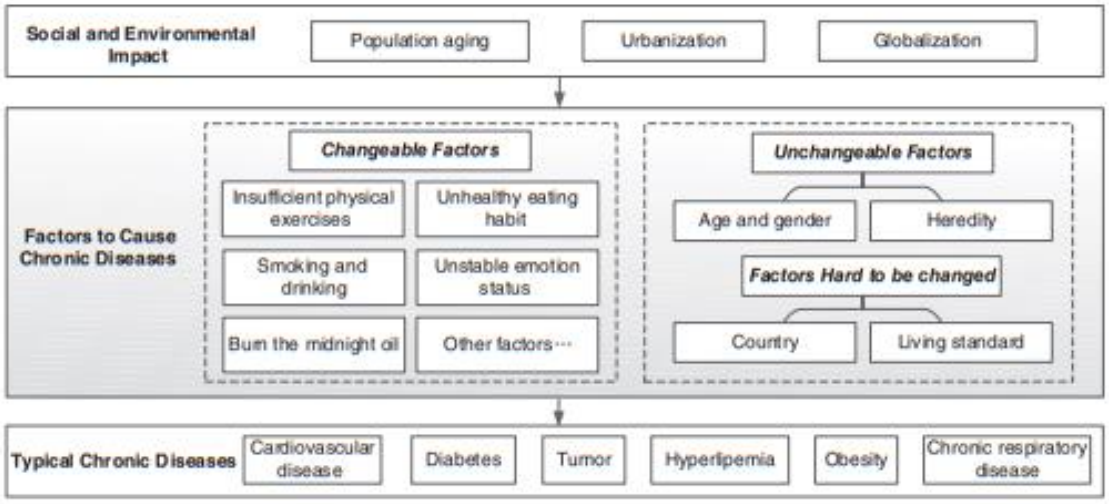
Pada tahun 2050, populasi yang menua di banyak negara akan naik ke tingkat yang sama, yang menyebabkan serangkaian masalah di seluruh dunia. Sistem medis di banyak negara menanggung beban berat, sementara jumlah fasilitas dan personel medis sangat tidak memadai. Salah satu solusi yang mungkin adalah penggabungan teknologi wearable computing dan IoT ke dalam layanan pemantauan kesehatan. Dibandingkan dengan masalah perawatan kesehatan yang khas seperti penuaan populasi, perawatan untuk penyakit kronis menjadi semakin penting saat ini.

Dengan perubahan ekonomi dan lingkungan masyarakat manusia, percepatan penyakit kronis telah menjadi ancaman utama bagi kesehatan manusia. Dan morbiditas terus meningkat. Namun, tidak mudah diakses masyarakat untuk menggunakan layanan kesehatan masyarakat karena kekurangan sumber daya dan fasilitas medis. Sumber daya dan sistem medis tersebut difokuskan di kota-kota perkotaan berdasarkan populasi dan modal. Selama ini sistem kedokteran dan kesehatan di beberapa negara (terutama negara berkembang) sebagian besar ditujukan untuk mengatasi penyakit akut dan penyakit menular, dimana pencegahan dan pengobatan penyakit kronis belum ditekankan.

Anehnya, peningkatan standar hidup mendorong peningkatan penyakit kronis. Di AS, 50% orang menderita satu atau beberapa jenis penyakit kronis pada tingkat yang berbeda. Delapan puluh persen dana medis digunakan untuk mengobati penyakit kronis. Pada 2015, AS menghabiskan sekitar Rp 40.500 triliun untuk pengobatan penyakit kronis. Ini menyumbang 18% dari PDB AS. Pengeluaran medis yang mahal menempatkan beban keuangan yang sangat besar pada masyarakat dan pemerintah.

Penyebab utama penyakit kronis meliputi tiga faktor. Mereka adalah faktor-faktor yang tidak dapat diubah, faktor-faktor yang dapat diubah, dan faktor-faktor yang sulit untuk diubah. Usia dan keturunan merupakan faktor yang tidak dapat diubah dan merupakan 20% penyebab penyakit kronis, seperti yang ditunjukkan pada Gambar 7.1. Keadaan hidup sangat penting bagi kondisi fisik seseorang, yang sulit untuk diubah secara sewenang-wenang.

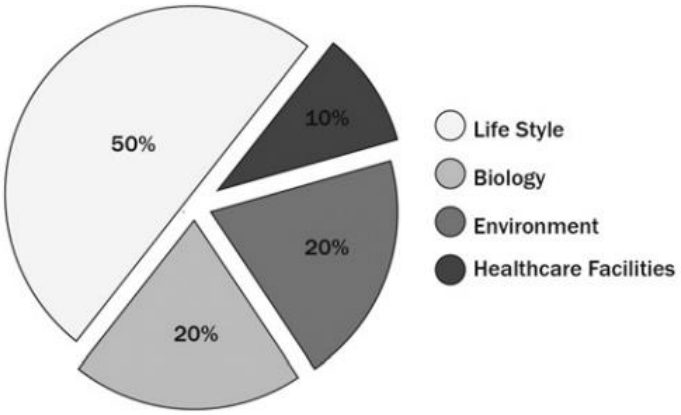
Pada tahun 2015, WHO mengeluarkan laporan penyakit kronis. Ini daftar empat jenis penyakit kronis utama, yaitu penyakit kardiovaskular, kanker, penyakit pernapasan kronis dan diabetes mellitus. Laporan tersebut menunjukkan bahwa pada tahun 2012, sebagian besar kematian penyakit tidak menular disebabkan oleh keempat penyakit tersebut di antara orang-orang yang berusia di bawah 70 tahun. Penyakit kardiovaskular mengambil proporsi terbesar kematian kronis di bawah usia 70 (37%), diikuti oleh kanker (27%) dan penyakit pernapasan kronis (8%). Diabetes mengambil 4% dari kematian dan faktor lain menyumbang sekitar 24% dari kematian.



Gambar 7.1 Faktor-faktor yang mempengaruhi akurasi deteksi dalam mendeteksi penyakit kronis.

Selain faktor lingkungan, tren sosial dan ekonomi dunia, seperti penuaan populasi, urbanisasi dan globalisasi, juga berdampak pada penyebab penyakit kronis. Penuaan populasi adalah alasan langsung meningkatnya jumlah pasien penyakit kronis dan urbanisasi memperburuk pencemaran lingkungan. Misalnya, PM2.5 dan kabut asap yang tinggi telah

memicu peningkatan penyakit paru-paru. Aspek lainnya adalah globalisasi. Orang-orang mulai terbiasa berkomunikasi dengan teman melalui perangkat seluler. Teknologi baru di bidang sosial, seluler, dan jaringan membuat kehidupan perkotaan lebih nyaman. Namun, kemajuan tersebut juga menciptakan berbagai gaya hidup tidak sehat. Misalnya, duduk di depan komputer terlalu lama dan kurang berolahraga menyebabkan masalah obesitas, dll.



Gambar 7.2 Determinan kesehatan (statistik dari pusat pengendalian penyakit tahun 2003).

Menurut laporan WHO baru-baru ini, faktor penentu kesehatan turun ke lima faktor yang berbeda. Faktanya, fasilitas kesehatan, seperti ahli bedah dan pusat kesehatan, hanya dapat menyelesaikan 10% masalah medis. Lima puluh persen tergantung pada gaya hidup seperti kebiasaan hidup, gaya makan dan olahraga. Dua puluh persen terletak pada lingkungan dan sisanya 20% disebabkan oleh faktor biologis seperti keturunan. Hal ini menunjukkan bahwa sebagian besar penyebabnya terkait dengan gaya hidup. Itulah mengapa kita harus lebih berkonsentrasi pada pengawasan kesehatan daripada setelah pengobatan, seperti yang ditunjukkan pada Gambar 7.2. Karena fitur intrinsik jangka panjangnya, penyakit kronis tidak cukup akut untuk dirawat di rumah sakit. Itulah sebabnya pemerintah nasional menghabiskan banyak uang untuk masalah ini. Pemantauan kesehatan berkelanjutan sangat penting untuk memecahkan masalah yang menantang ini.

Pustaka Perangkat Lunak untuk Aplikasi *Machine learning*

Dalam menjalankan tugas ML, kita perlu membuat program aplikasi, atau menggunakan kode yang ada, toolkit, benchmark dari open source atau pembelian dari penyedia layanan. Agar sesuai dengan persyaratan tugas, pendekatan terbaik adalah menulis kode aplikasi Anda sendiri. Pendekatan ini melibatkan pemilihan algoritme, toolkit dan pengumpulan dataset, pengkodean program dan uji coba yang diulang-ulang hingga sempurna. Dengan tenaga ahli atau pemrogram yang terbatas, akan lebih mudah untuk menerapkan kode atau tolok ukur yang ada.

Pada Tabel 7.1, kami mengidentifikasi beberapa perangkat lunak yang dapat membantu pengguna memilih program yang sesuai untuk analisis data. Anehnya, banyak dari paket ML ini berasal dari open source. Pembaca dapat memeriksa situs web pengembang untuk detail lebih

lanjut tentang fungsionalitas dan kemampuan program tersebut atau sistem dukungan runtime yang disediakan.

Hanya informasi pengantar singkat yang diberikan di sini. Kita dapat menggali lebih dalam dengan framework Google TensorFlow. Library Spark dan TensorFlow telah memperkaya kemampuan kami untuk mengembangkan aplikasi ML atau DL baru. Untuk banyak aktivitas kognitif, manusia (bahkan bayi yang baru lahir) dapat melakukan dengan mudah tetapi tidak selalu pasti, tetapi sekarang kita dapat melatih komputer untuk menangani tugas penyaringan dan penyaringan tersebut secara rutin untuk menghemat waktu dan meningkatkan proses keputusan kita dengan bukti atau dasar pendukung yang lebih baik.

Tabel 7.1 Toolkit *Machine learning* yang tersedia secara umum.

Toolkit atau Kerangka Kerja, Bahasa, Situs Web Pengembang	**Deskripsi Singkat Fungsi dan Kemampuan**
Scikit-learn, Python, http://scikit-learn.org/stable/	Dibangun dengan NumPy dan Matplotlib, menyediakan alat matematika sederhana dan efisien untuk penambangan data dan analisis *Big data*
Shogun, C++, http://www.shogun-toolbox.org/	Antarmuka SWIG memungkinkan komunikasi antara C++ dan bahasa target Python, Oktaf, R, Java, C#, dll., dengan fokus pada fungsi kernel SVM
Accord, Aforge.net,.NET, http://accord.codeplex.com/ http://www.aforgenet.com/framework	Diterapkan untuk pemrosesan audio/gambar dalam deteksi wajah dan jahitan gambar pada SIFT, mendukung computing seluler waktu nyata dengan ANN atau pohon keputusan
Mohout, Hadoop, https://mahout.apache.org/	Menggunakan MapReduce untuk berjalan pada satu atau beberapa node dari cluster Hadoop, sangat meningkatkan volume data
MLlib, Spark http://spark.apache.org/mllib/	MLlib dirancang untuk memungkinkan banyak algoritme ML berjalan cepat di kluster besar. Ini mendukung desain kode ML yang dipersonalisasi
Cloudera, Hadoop, http://www.cloudera.com/	Disediakan oleh distribusi Cloudera Hadoop, memungkinkan model *Machine learning* berjalan pada aliran data waktu nyata, seperti pemfilteran email spam
GoLearn, Go, https://github.com/sjwhitworth/golearn	Dikembangkan oleh Go with Google untuk mendukung desain kode yang disesuaikan dengan alat sederhana untuk memperluas struktur data dan kode sumber

Weka, Java, http://weka.wikispaces.com/	Weka dirancang untuk aplikasi penambangan data, pra-pemrosesan, klasifikasi, regresi, dan pengelompokan dengan dukungan visualisasi
CUDA-Convnet, C++, https://code.google.com/p/cuda-convnet	CUDA adalah toolkit percepatan GPU, sedangkan CUDA-Convnet adalah library *Machine learning* untuk ANN berdasarkan penggunaan cluster GPU cepat
ConvNetJS, JavaScript, http://www-cs-faculty.stanford.edu/people/karpathy/convnetjs/	Layanan pelatihan online untuk *Deep learning*, yang membantu pengguna memahami algoritme secara intuitif dengan menunjukkan beberapa demo sederhana
FBLearner Flow, Python, https://code.facebook.com/posts/1072626246134461/	Platform ini menggunakan kembali banyak algoritme dalam produk yang berbeda, dengan merentangkan ke dalam ribuan eksperimen simulasi yang disesuaikan. Ini juga menawarkan generasi otomatis pengalaman antarmuka pengguna dari kode Python

7.2 SISTEM DAN APLIKASI KESEHATAN BERBASIS IOT

Health Internet of Things (Health-IoT) dikhususkan untuk memecahkan masalah kesehatan medis, dan juga memiliki arti realistis yang penting untuk mempromosikan pengembangan industri kesehatan medis dan meningkatkan kualitas hidup masyarakat. Dibandingkan dengan IoT yang berorientasi pada hal-hal tradisional, Health-IoT adalah "berpusat pada manusia", dan semua akses jaringan, analisis data, dan layanan dilakukan di sekitar manusia. Misalnya, sensor pada lapisan pengumpulan data bukanlah sensor umum, tetapi sensor tubuh untuk mengumpulkan parameter kesehatan fisiologis.

Health-IoT sebelumnya menekankan desain sensor tubuh manusia dan pengumpulan data fisiologis tubuh manusia, tetapi tidak sepenuhnya mempertimbangkan mobilitas pengguna. Oleh karena itu, tidak nyaman untuk digunakan dalam kehidupan sehari-hari dan bahkan dapat mempengaruhi kehidupan sehari-hari. Perkembangan mobile internet membawa integrasi dunia fisik, dunia cyber dan jaringan sosial, sehingga menghasilkan Cyber-Physical Society System (CPSS). Mengintegrasikan Health-IoT ke dalam CPSS, memungkinkan pengguna untuk mendapatkan layanan dan kenyamanan yang dibawa oleh kesehatan seluler dan perawatan medis seluler, sementara mobilitas pengguna tidak terbatas dan ruang jejaring sosial merupakan tren yang tak terhindarkan untuk pengembangan Health-IoT.

IoT tradisional telah banyak diterapkan di industri lalu lintas, logistik, dan ritel. Dengan kedewasaannya, IoT menarik perhatian masyarakat di bidang kesehatan. Namun, banyak aplikasi yang mempromosikan layanan kesehatan kepada keluarga atau individu dengan memanfaatkan teknologi IoT, belakangan terbukti tidak berhasil. Karena pentingnya meningkatkan kualitas perawatan medis dan efisiensi layanan, Health-IoT menjadi tonggak sejarah dalam

pengembangan informasi kesehatan. Ini akan memainkan peran penting dalam meningkatkan tingkat kesehatan masyarakat dan meningkatkan kualitas hidup mereka.

Penginderaan IoT untuk Sinyal Tubuh

Pengumpulan informasi fisiologis adalah dasar dari Health-IoT, sedangkan sensor adalah penghubung terpenting dari pengumpulan informasi fisiologis, dan merupakan jembatan antara dunia fisiologis dan sistem elektronik. Perangkat penginderaan bertanggung jawab untuk mengumpulkan data fisiologis dari tubuh manusia, dan data ini akan membantu pengguna memeriksa situasi fisik mereka sendiri dan membantu dokter dalam mendiagnosis mereka.

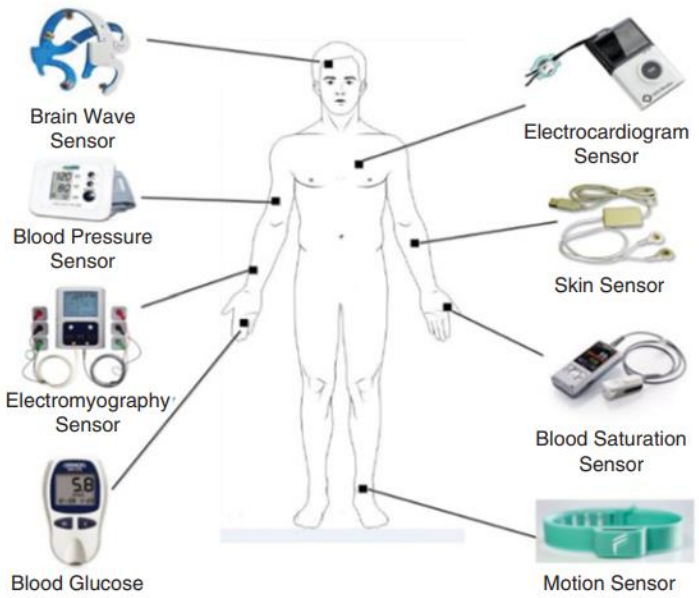
Menurut permintaan pengguna untuk perawatan medis seluler dan sistem kesehatan, perangkat pengumpul informasi fisiologis dalam aplikasi Health-IoT dibagi menjadi dua kategori besar, satu kategori mengumpulkan informasi fisiologis melalui komponen penginderaan yang terintegrasi pada perangkat seluler universal (GMD: Perangkat Seluler Umum) (yaitu telepon seluler); dan yang lainnya mengacu pada perangkat pengumpul kesehatan medis khusus (MHS: Sensor Kesehatan Medis), yang mengumpulkan informasi kesehatan dengan merancang dan mengintegrasikan satu atau beberapa sensor kesehatan khusus. Berikut ini mengacu pada fitur masing-masing dari dua jenis perangkat pengumpul ini.

Perangkat pengumpul bergerak universal memiliki keuntungan dari biaya rendah serta kenyamanan dalam membawa dan menggunakan. Namun, mereka juga memiliki kekurangan. Misalnya, ketepatan pengumpulan data rendah dan jenis informasi fisiologis yang dikumpulkan terbatas. Sistem Perawatan Kesehatan Medis (Medical Healthcare System/MHS) yang lengkap seringkali dilengkapi dengan perangkat atau sensor berikut, seperti yang ditunjukkan pada Gambar 7.3.

Berdasarkan MHS, ada kekhawatiran berikut untuk penginderaan sinyal tubuh IoT:

- **Sensor Tertanam:** Perangkat MHS mengadopsi sensor khusus, dan memiliki keunggulan presisi pengumpulan yang tinggi, tetapi juga memiliki kelemahan biaya tinggi dan portabilitas serta kegunaan yang tidak memadai. Perangkat semacam ini memiliki beberapa fitur berikut.
- **Daya tahan pakai:** Sebagian besar MHS harus ditempatkan pada tubuh manusia untuk mengumpulkan data yang akurat, karena pengumpulan datanya menargetkan tanda-tanda vital manusia. Oleh karena itu, hampir semua alat pengumpul kesehatan medis yang ada menjadikan wearability sebagai persyaratan dasar. Dalam hal ini, kenyamanan pengguna harus diperhatikan dan keakuratan data yang dikumpulkan harus dijamin selama prosedur pengumpulan. Tata letak sensor tubuh manusia yang umum ditunjukkan pada Gambar 7.3.
- **Waktu kerja yang lama:** Metode perangkat pengumpul kesehatan medis khusus bervariasi dengan perangkat pengumpul seluler universal. Tujuan dari yang pertama adalah untuk mengumpulkan data dari tubuh manusia dalam jangka waktu yang relatif lama, yang membutuhkan kemampuan catu daya MHS yang tinggi.

- **Stabilitas:** MHS masih dapat mengumpulkan data secara normal saat pengguna melakukan olahraga berat atau di lingkungan yang ekstrem.
- **Tingkat partisipasi pengguna yang rendah:** Berbeda dari metode GMD, fungsi MHS relatif independen, dan sebagian besar perangkat MHS tidak memerlukan intervensi pengguna selama prosedur pengumpulan data, dan pengguna hanya perlu menyalakan sumber daya , dan MHS kemudian akan mulai mengumpulkan.
- **Memiliki mekanisme penyimpanan data sementara:** Berat dan dimensi MHS mungkin dibatasi secara ketat untuk memenuhi persyaratan fitur yang dapat dikenakan. Oleh karena itu, sebagian besar perangkat MHS tidak akan mengintegrasikan modul transmisi data, tetapi akan memilih modul penyimpanan data dengan dimensi yang relatif kecil, dan mengadopsi mekanisme penyimpanan sementara data untuk menyimpan data yang dikumpulkan terlebih dahulu, dan kemudian mengirimkan data melalui perangkat akses jaringan lainnya. .



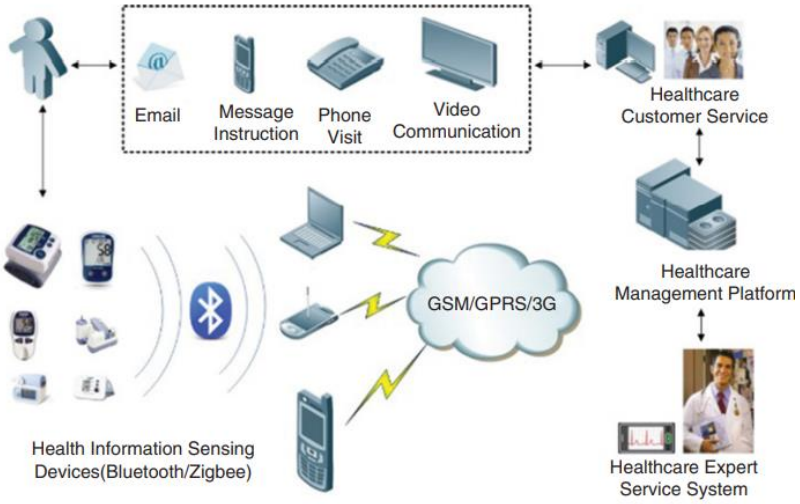
Gambar 7.3 Tata letak sensor tubuh manusia yang umum.

Sistem Pemantauan Layanan Kesehatan

Dalam beberapa tahun terakhir, pemantauan kesehatan jarak jauh untuk lingkungan rumah telah berkembang pesat, dan telah mengintegrasikan sensor kesehatan, komunikasi nirkabel, dan computing cloud. Ini telah menjadi penggunaan khas Health-IoT dan sistem cloud. Data yang dikumpulkan oleh sensor dapat ditransmisikan ke ponsel, sedangkan ponsel dan sensor dapat dihubungkan melalui Bluetooth, dan data tersebut kemudian ditransmisikan ke platform layanan manajemen kesehatan di cloud.

Aplikasi semacam ini terutama berlaku untuk orang tua, pasien dengan penyakit kronis dan dari kelompok sub-kesehatan. Ini dapat memantau parameter fisiologis orang, termasuk

oksigen/denyut darah, tekanan darah, glukosa darah, EKG, suhu tubuh, dan pernapasan. Gambar 7.4 mengacu pada keseluruhan arsitektur sistem pemantauan kesehatan umum berdasarkan layanan masyarakat. Layanan biasanya disediakan oleh sistem pemantauan kesehatan, seperti yang ditunjukkan pada Tabel 7.2. Sebagai versi yang disederhanakan, perangkat pendeteksi kesehatan khusus dapat dihubungkan dengan ponsel cerdas melalui Bluetooth. Data fisiologis yang terdeteksi ditransmisikan ke ponsel untuk visualisasi.



Gambar 7.4 Sistem pemantauan kesehatan umum berdasarkan layanan masyarakat.

Sistem pemantauan kesehatan yang berbeda memiliki ciri khasnya masing-masing, dan sesuai dengan berbagai kelompok populasi, seperti orang tua, orang yang tidak bersarang, dan pasien dengan penyakit kronis, dll. Berikut ini, kami mengklasifikasikan sistem pemantauan kesehatan ke dalam kategori berikut:

- **Sistem Cyber-Physical Kesehatan:** Sistem Cyber-Physical (CPS) seluler berorientasi kesehatan memainkan peran penting dalam aplikasi pemantauan medis yang ada, seperti diagnosis, pengobatan penyakit dan penyelamatan darurat, dll. Beberapa sistem jaringan cerdas medis elektronik cocok untuk sejumlah besar pasien telah dirancang. Keterlambatan end-to-End penyampaian informasi medis menjadi perhatian utama, terutama pada saat terjadi kecelakaan, atau pada saat terjadi wabah epidemi.
- **Pemantauan Kesehatan Seluler:** Beberapa tahun yang lalu, sistem pemantauan kesehatan seluler berdasarkan peralatan medis portabel dan telepon pintar diusulkan. Ponsel pintar digunakan untuk mengumpulkan sinyal fisiologis dari tubuh manusia dari berbagai perangkat pemantauan kesehatan berdasarkan perangkat lunak aplikasi ponsel pintar khusus. Kemudian sinyal fisiologis tersebut ditransmisikan ke pusat kesehatan. Jika perlu, itu juga dapat memberi tahu pengasuh dan institusi darurat medis menggunakan layanan pesan singkat ponsel.

- **Computing yang Dapat Dipakai untuk Pemantauan Kesehatan:** Selama jangka waktu yang lama, perangkat yang dapat dikenakan dan computing yang dapat dikenakan adalah topik penelitian utama untuk memungkinkan pemantauan kesehatan. Sebagai node sensor tubuh jenis baru, ponsel pintar dan jam tangan pintar diadopsi untuk mengukur SpO2 dan detak jantung; namun, data pengukuran tersebut memiliki akurasi yang rendah, jenis sinyal yang sedikit dan penggunaan medis yang terbatas (Tabel 7.3).

Tabel 7.2 Layanan pemantauan kesehatan umum.

No.	Konten Layanan	Metode layanan
1	Menyediakan EKG jarak jauh 24 jam/tekanan darah/glukosa darah/oksigen darah/denyut nadi/pernapasan/tidur	Layanan pemantauan waktu nyata
2	Memberikan peringatan real-time untuk memantau kelainan	Pesan singkat
3	Menyediakan layanan memberi tahu kerabat tentang informasi pemantauan	Pesan singkat
4	Menyediakan layanan konsultasi ahli pemesanan	Video atau pesan singkat
5	Menyediakan panggilan darurat dan layanan bantuan	Panggilan telepon otomatis
6	Menyediakan layanan penentuan posisi keluarga	Pemosisian
7	Layanan laporan penilaian kesehatan reguler	Pesan singkat atau email
8	Layanan perawatan promosi kesehatan reguler	Pesan singkat atau email
9	Layanan tindak lanjut reguler	Telepon
10	Layanan manajemen catatan kesehatan seumur hidup	Pertanyaan situs web
11	Layanan pertanyaan swadaya data pengguna	Pertanyaan situs web
12	Menyediakan layanan hotline konsultasi 24 jam	Telepon

Tabel 7.3 Beberapa alat pemantau kesehatan umum.

Nama Perangkat	Memantau konten	Fungsi tambahan
Pemantau tekanan darah	Tekanan darah	Merekam data tekanan darah historis
Monitor tekanan darah Cloud e-health	Tekanan darah	Mengintegrasikan platform cloud, data historis
Sunstudy GPS LBS	Melacak orang tua	kurva dan mentransmisikan informasi marabahaya

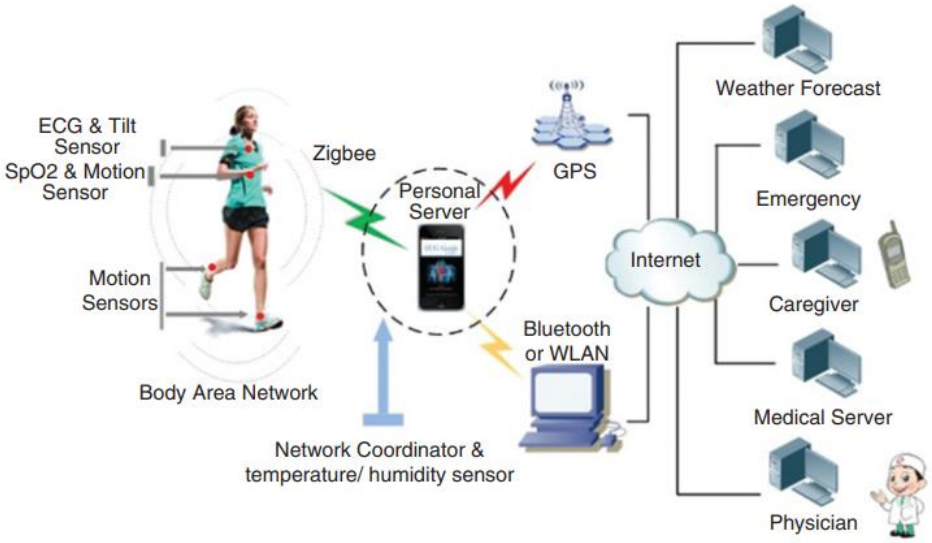
Perangkat tekanan darah pintar	Tekanan darah dan detak jantung	Komunikasi ponsel, gangguan SOS
jam tangan jWatch	Tekanan darah dan detak jantung	alarm mencari bantuan dan memutar tiga nomor berturut-turut; mengunggah posisi pelacakan secara teratur dan alarm berdaya rendah
Pemantauan bayi jarak jauh	Memantau bayi	Pemantauan tekanan darah dan detak jantung dapat menghindari fibrilasi atrium dengan menghubungi dokter untuk mendapatkan perawatan yang tepat dan mengetahui situasi pasien lain

- **Health Internet of Things:** Health IoT adalah cara lain untuk menyediakan layanan pemantauan kesehatan. Penginderaan seluler, lokalisasi, dan analisis jaringan berdasarkan teknologi IoT dapat digunakan untuk perawatan kesehatan.
- **Ambient Assisted Living:** Ambient Assisted Living (AAL) bertujuan untuk meningkatkan kualitas hidup pasien, dan dapat memberi tahu kerabat, pengasuh, dan pakar perawatan kesehatan yang relevan. Teknologi terkait AAL mencakup teknologi penginderaan, pemantauan sinyal fisiologis, pemantauan lingkungan rumah, penginderaan berbasis video, teknologi rumah pintar, analisis pola, dan *Machine learning*.
- **Pemantauan Kesehatan berbasis Jaringan Area Tubuh:** Pekerjaan yang ada pada jaringan area tubuh (BAN) berfokus pada penghematan energi node sensor, desain jaringan intra-BAN, sensor mikro implan, akuisisi sinyal fisiologis, dll. Sistem pemantauan kesehatan portabel pintar yang dapat dipakai berdasarkan BAN telah dikembangkan. Namun, stabilitas, keberlanjutan, dan keandalan sistem perlu ditingkatkan.

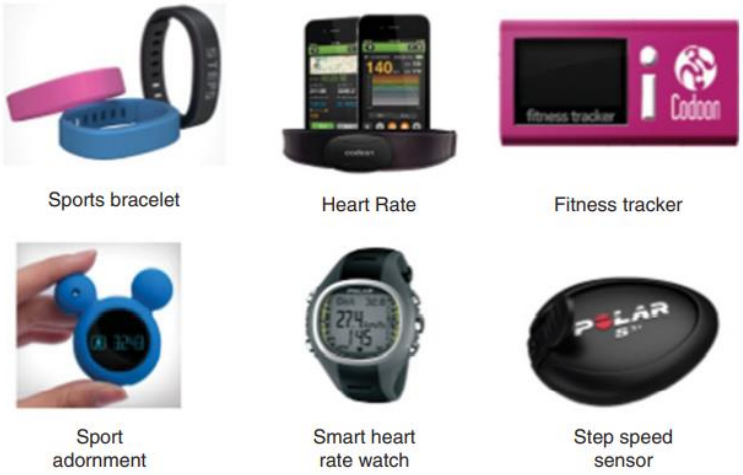
Promosi Latihan Fisik dan Pakaian Cerdas

Dengan meningkatnya perangkat wearable dan meningkatnya perhatian masyarakat terhadap kesehatan, industri promosi olahraga berbasis perangkat wearable berkembang pesat. Perangkat wearable dapat merekam jumlah olahraga, konsumsi makanan, dan status tidur pengguna setiap hari, sehingga secara efektif mengawasi dan mendesak mereka untuk meningkatkan jumlah olahraga agar tubuh tetap sehat. Sebuah arsitektur komunikasi perangkat promosi latihan telah diusulkan, seperti yang ditunjukkan pada Gambar 7.5.

Perangkat promosi olahraga profesional dapat mengukur berbagai indeks fisik seperti detak jantung dan pernapasan dengan lebih tepat, memantau datanya termasuk kecepatan, jarak lari, dan daya tahan di lapangan olahraga, serta memberikan dukungan untuk meningkatkan pencapaian olahraga. Dengan demikian, pelatih dapat mengetahui status anggota timnya secara lebih visual, dan memilih atlet yang paling cocok untuk berpartisipasi dalam acara olahraga.



Gambar 7.5 Arsitektur komunikasi perangkat promosi olahraga.



Gambar 7.6 Produk promosi olahraga yang tersedia di tahun 2016

Produk promosi olahraga saat ini hampir dapat dikenakan, seperti gelang pintar, strip detak jantung, dan jam tangan pintar, seperti yang ditunjukkan pada Gambar 7.6. Perangkat ini dapat mewujudkan penghitungan langkah olahraga, pelacakan olahraga, pemantauan detak jantung, serta pemantauan olahraga dan tidur secara real-time, pelacakan dan kualitas tidur, pelacakan diet, konsumsi kalori dan pembakaran kalori, pelacakan emosional, kursus jarak jauh dan penghitungan. ing, pengingat gerak, alarm khusus, jam alarm pintar, dan jam alarm pintar tanpa suara.

Contoh 7.1 Perangkat Lunak Aplikasi Smart Clothing dan Testbed Setting Saat ini, smart clothing menjadi perangkat wearable yang inovatif untuk promosi latihan fisik. Definisi pakaian pintar diberikan dalam contoh: pakaian pintar adalah sejenis sistem baru yang mengintegrasikan

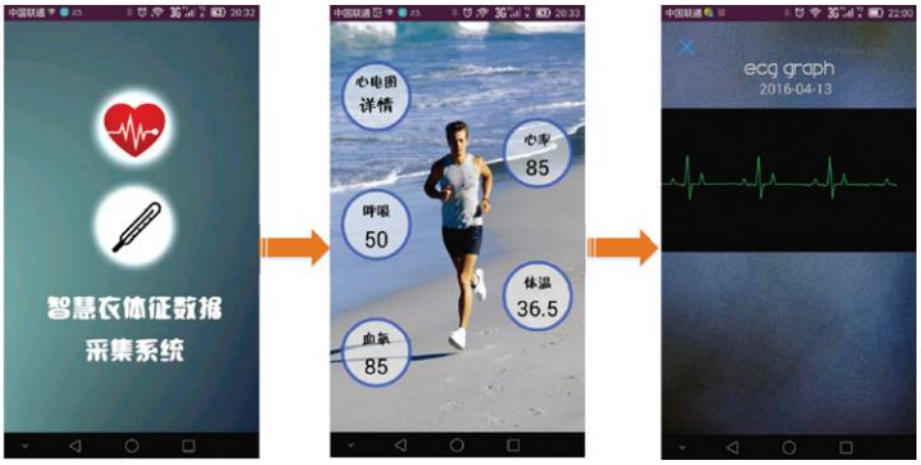
berbagai sensor mikro untuk pengumpulan sinyal fisik. Dibandingkan dengan perangkat wearable tradisional, smart clothing memiliki fitur berikut: nyaman, nyaman, dapat dicuci, sangat andal, dan tahan lama.

Pada pakaian pintar, sensor tubuh diintegrasikan ke dalam tekstil, yang mempertimbangkan berbagai faktor, seperti jenis sensor, lokasi strategis untuk penempatan sensor, dan tata letak kabel listrik fleksibel. Kain pakaian pintar mengadopsi kain tekstil elastis, yang cocok untuk dipakai di samping kulit. Perangkat lunak APP pakaian pintar yang dipasang di ponsel ditunjukkan pada Gambar 7.7(a), dan sistem pengujian pakaian pintar ditunjukkan pada Gambar 7.7(b).

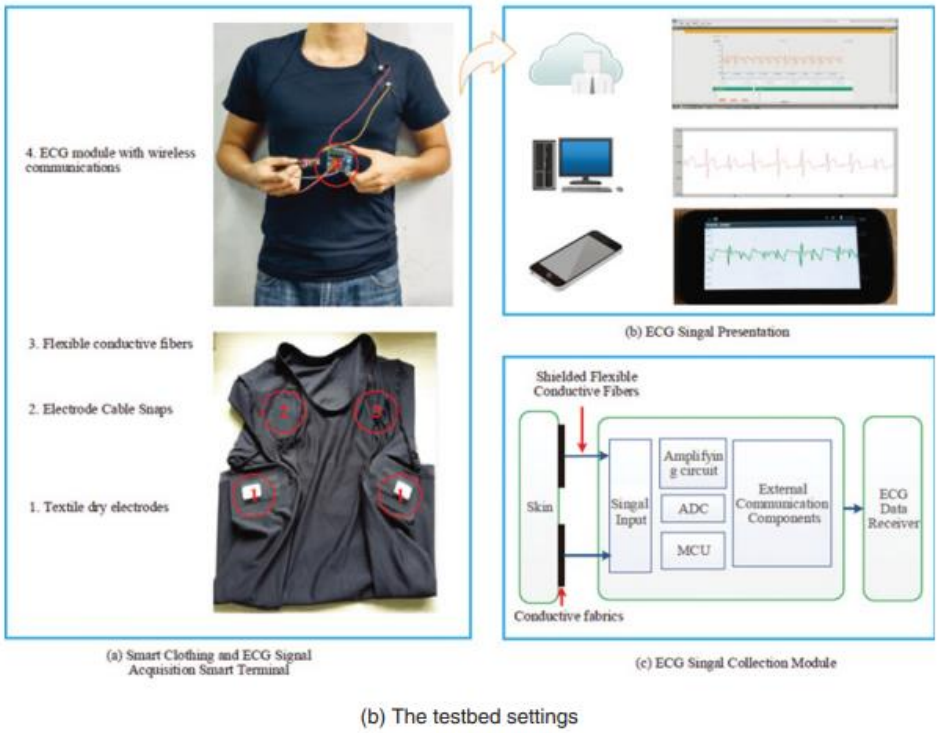
Robotika Perawatan Kesehatan dan Cloud Kesehatan Seluler

Computing cloud adalah jenis baru computing dan mode layanan berbasis Internet. Melalui metode ini, pengumpulan sumber daya dan informasi perangkat keras dan perangkat lunak dapat diberikan kepada peminta layanan berdasarkan kebutuhan. Robot tradisional selalu dibatasi dalam fungsi perangkat keras dan perangkat lunak di mana ada masalah serius. Tetapi computing cloud, sebagai pendukung yang baik untuk teknologi robot, dapat dengan mudah menggabungkan computing cloud dengan teknologi robot untuk membangun robot cloud.

Sebagai peralatan frontend, robot mengambil alih pengumpulan sinyal, kinerja aksi spesifik dan beberapa tugas sederhana dalam hal analisis dan proses, sementara tugas yang lebih rumit dipindahkan ke cloud. Dengan menggunakan kapasitas penyimpanan dan computingnya yang kuat, cloud membangun model yang efektif dengan menerapkan algoritme ML dan mengirimkan hasil analisis kembali ke robot. Robot juga akan menangani beberapa computing secara lokal berdasarkan sumber daya yang tersedia.



(a) Mobile application software



Gambar 7.7 Perangkat lunak aplikasi pakaian pintar dan pengaturan ranjang percobaan.

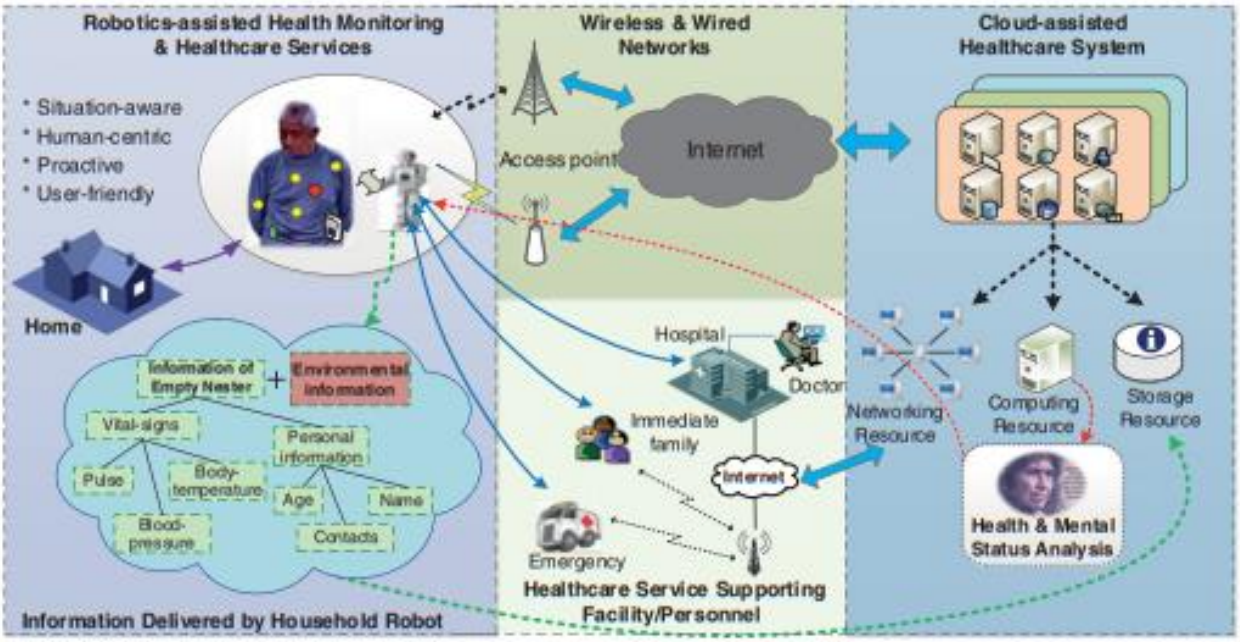
Contoh 7.2 Perangkat Seluler, Robot, dan Lingkungan Cloud untuk Perawatan Kesehatan

Gambar 7.8 menunjukkan kumpulan perangkat bergerak, robot, dan cloud untuk aplikasi perawatan kesehatan. Integrasi robot dan teknologi computing cloud dalam sistem perawatan kesehatan dapat sangat meningkatkan kualitas dan tingkat layanan. Pengguna menerapkan perangkat yang dapat dikenakan untuk mengumpulkan data fisiologis, dan kemudian data yang dikumpulkan diteruskan ke platform cloud jarak jauh oleh robot. Robot dapat menyimpan data sensorik, berinteraksi dengan manusia dan mengintegrasikan berbagai modul komunikasi nirkabel termasuk ZigBee, WiFi dan LTE. Cloud digunakan untuk menyimpan data kesehatan skala besar, analisis dan prediksi kesehatan, dan menyediakan layanan yang dipersonalisasi.

Teknologi robotika memiliki pengaruh besar pada masyarakat, ekonomi, dan kehidupan masyarakat. Perkembangan teknologi jaringan nirkabel dan computing cloud membuka jalan bagi robot untuk bergerak dari bidang kontrol industri ke bidang layanan. Saat ini, robot yang beredar di pasaran terutama berfokus pada pendidikan anak usia dini, hiburan, dan layanan rumah tangga (yaitu robot pembersih). Sebagian besar dikendalikan oleh perangkat lunak dengan fungsi yang disesuaikan. Robot berjejaring memungkinkan operasi dan manajemen jarak jauh. Kurangnya tautan komunikasi yang efisien dan kemampuan belajar adalah kelemahan utama dalam sistem interaksi cloud-robot saat ini.

Arsitektur robot cloud dibagi menjadi dua tingkatan: tingkat mesin ke mesin (M2M) dan tingkat mesin ke cloud (M2C). Di level M2M, sekelompok robot terhubung satu sama lain melalui

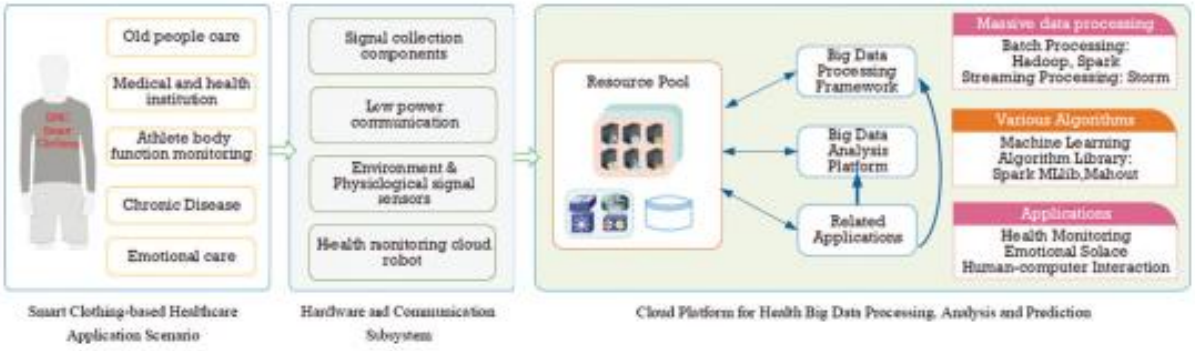
jaringan nirkabel untuk membentuk infrastruktur cloud kolaboratif robot ad hoc. Di lapisan M2C, ia menyediakan kumpulan sumber daya computing dan penyimpanan bersama, yang memungkinkan robot untuk memindahkan tugas computing ke cloud. Grup riset Google telah mengembangkan sistem robot berbasis telepon pintar, belajar melalui cloud. Gambar 7.8 menunjukkan arsitektur robotika dan sistem perawatan kesehatan yang dibantu cloud.



Gambar 7.8 Robotika dan sistem perawatan kesehatan berbantuan cloud.

Gambar 7.9 menunjukkan contoh sistem cloud kesehatan seluler. Dari ujung ke cloud, sistem cloud kesehatan seluler melibatkan pakaian pintar, ponsel, gerbang komunikasi, sistem cloud kesehatan dan pusat data, dll. Perangkat lunak terkait mencakup perangkat lunak terkait pakaian pintar, perangkat lunak aplikasi ponsel pintar, analisis *Big data* yang dibantu cloud alat, dll. Perangkat lunak dari setiap bagian perlu dikembangkan secara terpisah, dan akhirnya diintegrasikan ke dalam paket layanan untuk akses yang luas.

Seluruh sistem perangkat lunak melibatkan pengembangan sistem tertanam, aplikasi seluler, dan penyebaran cloud yang digabungkan dengan teknologi *Big data*. Pengembangan perangkat lunak pakaian pintar perlu menangkap sinyal fisiologis. Komunikasi, penyimpanan data, sistem alarm, dan fungsi lainnya dibatasi oleh konsumsi daya yang rendah dan kapasitas computing yang tertanam. APLIKASI pakaian pintar yang dipasang di telepon seluler ditunjukkan pada Gambar 7.7(a).



Gambar 7.9 Sistem pemantauan kesehatan khas yang dibuat dengan pakaian pintar dan cloud backend.

Perangkat lunak pakaian pintar harus mendukung dua fungsi utama: i) interkoneksi dengan sensor tubuh untuk akuisisi sinyal tubuh dengan parameter yang dapat disesuaikan dan mengunggah data yang dikumpulkan oleh pakaian pintar ke cloud; ii) menyediakan layanan kesehatan yang dipersonalisasi untuk pengguna dan menunjukkan kepada pengguna semua jenis indikator fisiologis, memberikan kemudahan bagi pengguna untuk menanyakan data historis, menerima pesan alarm dan panduan kesehatan yang dikirim oleh cloud, dan mengingatkan pengguna tentang tindakan pencegahan kesehatan mereka.

Perangkat lunak cloud seluler menghasilkan kecerdasan berdasarkan sistem terpusat. Pertama, kita perlu membangun alat manajemen sumber daya untuk mengoordinasikan computing, penyimpanan, dan sumber daya jaringan. Dengan bantuan manajer sumber daya cloud, mekanisme alokasi dinamis dapat diterapkan untuk mengakses sumber daya cloud untuk aplikasi perawatan kesehatan berbasis pakaian pintar. Untuk memberikan saran dan diagnosis kesehatan yang lebih akurat, statistik data, dan perpustakaan *Machine learning*, API untuk analitik *Big data* di cloud perlu diaktifkan untuk memprediksi tren perkembangan tersembunyi dari status pengguna. Cloud juga dapat menyediakan layanan antara pengguna dan institusi medis dan kesehatan pihak ketiga. Perangkat lunak cloud adalah dasar dari sistem pakaian pintar, sementara aplikasi ponsel menyediakan jembatan untuk layanan ini.

7.3 ANALISIS *BIG DATA* UNTUK APLIKASI PERAWATAN KESEHATAN

Dengan pertumbuhan *Big data* dalam komunitas biomedis dan perawatan kesehatan, analisis data medis bermanfaat untuk deteksi dini penyakit, perawatan pasien, dan layanan masyarakat. Namun, akurasi analisis berkurang ketika kualitas data medis tidak lengkap atau hilang dalam berbagai dimensi atribut. Selain itu, daerah yang berbeda menunjukkan karakteristik yang unik mengenai penyakit daerah, yang memperumit penilaian wabah penyakit. Di bagian ini, kami menyederhanakan model dan algoritme *Machine learning*, terutama untuk mendeteksi wabah penyakit kronis di komunitas yang teridentifikasi. Kami bereksperimen

dengan model deteksi yang dimodifikasi atas data rumah sakit kehidupan nyata yang dikumpulkan dari Cina tengah dari 2012 hingga 2014.

Untuk mengatasi kesulitan data yang tidak lengkap, kami mengembangkan metode dekomposisi matriks baru untuk merekonstruksi data yang hilang. Kami mengidentifikasi penyakit kronis regional seperti hiperlipemia dan diabetes, dan berkonsultasi dengan ahli rumah sakit untuk mengekstraksi karakteristik penyakit tersebut. Sebuah model penilaian risiko baru dikembangkan menggunakan lima model *Machine learning*: naive Bayesian (NB), k-nearest neighbor (KNN), support vector machine (SVM), jaringan saraf tiruan (JST) dan pohon keputusan (DT). Membandingkan kinerja relatif dari model prediksi ini, kami menemukan bahwa metode DT dan SVM menampilkan kinerja yang lebih tinggi daripada model lain untuk mendeteksi penyakit kronis. Tingkat akurasi deteksi dapat dilatih hingga 90%.

Prapemrosesan *Big data* Layanan Kesehatan

Penerapan *big data* kesehatan erat kaitannya dengan keberlanjutan pemantauan kesehatan. Dari sudut pandang pengumpulan data, tanpa pengumpulan data fisiologis jangka panjang yang didukung oleh pemantauan kesehatan yang berkelanjutan, volume data tidak dapat mencapai tingkat *big data*. Dari sudut pandang cloud, analitik *Big data* kesehatan di cloud memberikan kecerdasan untuk pemantauan kesehatan yang lebih efisien dan membuatnya lebih berkelanjutan.

Secara umum, *Big data* sangat penting dalam mengoptimalkan biaya sistem kesehatan publik dan swasta. *Big data* kesehatan mempromosikan gaya hidup dan aktivitas yang sehat, menghindari terjadinya penyakit kronis (yaitu hipertensi), memperlambat penyakit kronis dan memindahkan pasien ketergantungan ke pusat pemantauan. Di era *Big data* saat ini, menjadi mungkin untuk mengumpulkan data medis dan kesehatan besar-besaran berdasarkan perangkat yang dapat dikenakan bersama dengan penerapan sejumlah besar platform bisnis jaringan area tubuh. Penelitian tentang pengenalan aktivitas manusia dengan menggunakan penelitian teknologi *big data* telah menjadi arah penelitian penting dari *big data* kesehatan.

Pada bagian ini, kita akan membahas pra-pemrosesan perawatan kesehatan berdasarkan data nyata dari sebuah rumah sakit di Wuhan, Cina. Data yang disediakan oleh pihak rumah sakit antara lain EHR, data citra medis dan data gen. Data mencakup total lebih dari 30.000 pasien. Empat tabel penting diambil dari dataset medis:

- Tabel penyakit: jumlah penyakit dan nama penyakit yang sesuai;
- Tabel hasil: hasil pemeriksaan pasien dan saran dokter;
- Tabel pasien: informasi dasar pasien, seperti jenis kelamin, usia, kebiasaan hidup, dan item inspeksi;
- Tabel pasien-penyakit: catatan pasien.

Sebagai contoh, Tabel 7.4 memberikan contoh data pemeriksaan fisik, termasuk statistik pasien, kebiasaan hidup, item dan hasil pemeriksaan, penyakit pasien, biaya perawatan kesehatan yang dikeluarkan oleh pasien dan saran dokter. Pada tahun 2012, hiperlipemia tidak muncul pada

sejumlah besar pasien. Namun, ada populasi risiko tinggi hiperlipemia (yaitu trigliserida tinggi); dengan demikian, kami mengembangkan model risiko hiperlipemia untuk mengidentifikasi penyakit lebih awal. Kami prihatin dengan jumlah kegagalan pemeriksaan dan proporsi pria dan wanita dan pengeluaran tahunan rata-rata mereka untuk perawatan kesehatan. Jumlah orang yang gagal dalam pemeriksaan fisik meningkat setiap tahun. Ada lebih banyak pasien laki-laki daripada pasien perempuan di wilayah ini. Artinya, penyakit sensitif terhadap perbedaan gender. Meningkatnya angka penyakit kronis tercermin dari biaya pengobatan pasien.

Sebelum imputasi data, pertama-tama kami menggunakan integrasi data untuk pra-pemrosesan data. Keakuratan prediksi risiko tergantung pada fitur keragaman data rumah sakit. Kami dapat mengintegrasikan data medis untuk menjamin atomitas data: yaitu kami mengintegrasikan tinggi dan berat badan untuk mendapatkan indeks massa tubuh (BMI). Variabel laten adalah variabel yang tidak dapat diamati secara langsung dalam model tertentu. Model faktor laten disajikan untuk menjelaskan variabel yang dapat diamati dalam kaitannya dengan variabel laten. Pendekatan faktorisasi matriks adalah salah satu realisasi model faktor laten yang paling berhasil.

Tabel 7.4 Istilah terkait medis yang sering ditemukan di database rumah sakit.

Barang	**Keterangan**
Demografi	Jenis kelamin pasien, usia, dll.
Kebiasaan hidup	Apakah pasien merokok, apakah ia memiliki riwayat genetik, dll.
Hasil ujian	Termasuk 132 item, seperti darah, dll.
penyakit	Penyakit yang diderita pasien, seperti hipertensi, diabetes, dll.
Biaya perawatan kesehatan	Pengeluaran pasien secara rinci.
Saran dokter	Nasihat dokter mengenai penyakit pasien, keadaan risikonya.

Analisis Prediktif untuk Deteksi Penyakit

Kami telah mempelajari algoritme *Machine learning* di Bab 5 dan 6 untuk pendekatan terawasi dan tidak terawasi. Di bagian ini, kami menyajikan tiga contoh aplikasi perawatan kesehatan konkret dalam menggunakan analitik prediktif berdasarkan lima algoritme *Machine learning* yang berbeda, yaitu regresi logistik, pengklasifikasi Baysian, pohon keputusan, metode KNN dan SVM. Di bagian berikutnya, kami akan membandingkan lebih banyak pilihan metrik kinerja untuk membuat pilihan model *Machine learning* yang sesuai untuk deteksi penyakit kronis.

Contoh 7.3 Diagnosis Penyakit Prediktif menggunakan Regresi Logistik

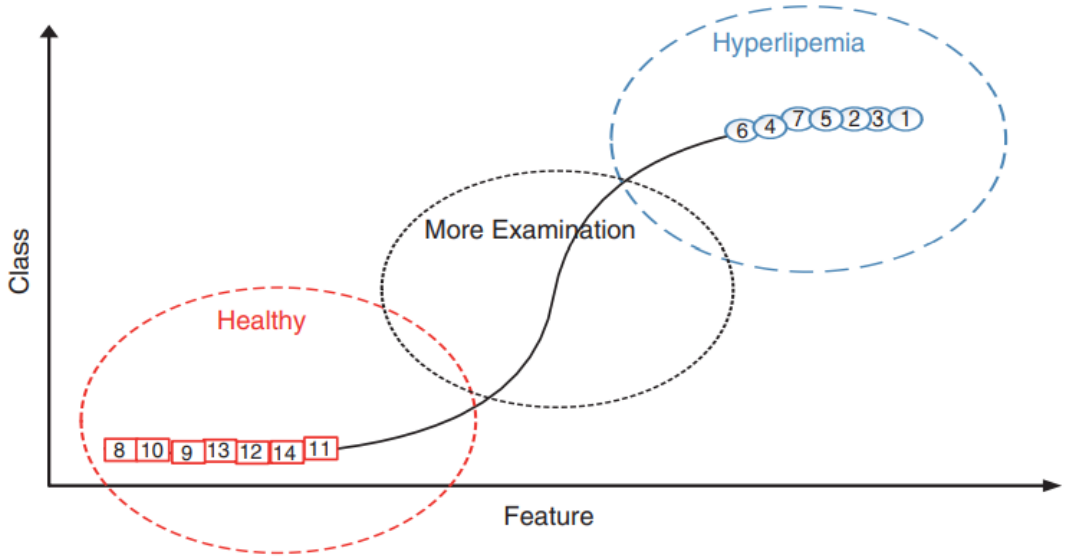
Tabel 7.5 mencantumkan kumpulan data trigliserida, kandungan kolesterol total, lipoprotein densitas tinggi, lipoprotein densitas rendah dan hiperlipemia atau tidak (“1” untuk ya dan “0” untuk tidak). Ini dikumpulkan dari data pemeriksaan kesehatan di sebuah rumah sakit di Wuhan, China. Mari kita coba melakukan penilaian pendahuluan apakah orang tersebut

mengalami hiperlipemia, jika data pemeriksaan kesehatannya adalah {3.16, 5.20, 0.97, 3.49} secara berurutan.

Untuk mendeteksi hiperlipemia, kami memilih pendekatan regresi logistik dengan "1" untuk hiperlipemia atau "0" untuk sehat, dengan mempertimbangkan empat atribut (fitur). Pertama, kami mengekstrak keempat atribut dan menggabungkannya menjadi satu atribut, sebagai $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$, di mana x_1, x_2, x_3, x_4 adalah trigliserida, kandungan kolesterol total, lipoprotein densitas tinggi dan lipoprotein densitas rendah masing-masing, dan z singkatan dari fitur setelah kombinasi. Kedua, memperkirakan bobot β dengan menggunakan metode kemungkinan maksimum, mengadopsi perangkat lunak MATLAB di sini, dan melakukan iterasi solusi untuk persamaan kemungkinan yang ditetapkan dengan Metode Newton-Raphson.

Tabel 7.5 Data pemeriksaan kesehatan pasien hiperlipemia.

identitas pasien	Trigliserida	Total kolesterol	Lipoprotein Kepadatan Tinggi	Lipoprotein Kepadatan Rendah	Apakah hiperlipemia atau tidak
1	3.62	7	2.75	3.13	1
2	1.65	6.06	1.1	5.15	1
3	1.81	6.62	1.62	4.8	1
4	2.26	5.58	1.67	3.49	1
5	2.65	5.89	1.29	3.83	1
6	1.88	5.4	1.27	3.83	1
7	5.57	6.12	0.98	3.4	1
8	6.13	1	4.14	1.65	0
9	5.97	1.06	4.67	2.82	0
10	6.27	1.17	4.43	1.22	0
11	4.87	1.47	3.04	2.22	0
12	6.2	1.53	4.16	2.84	0
13	5.54	1.36	3.63	1.01	0
14	3.24	1.35	1.82	0.97	0



Gambar 7.10 Hasil klasifikasi menggunakan regresi logistik pada Contoh 7.3.

Sesuai dengan hasil di atas, β_2 relatif besar; dengan demikian dapat diketahui bahwa seseorang mengalami hiperlipemia atau tidak sangat dipengaruhi oleh total kandungan kolesterol yang diukur dalam pemeriksaan kesehatan. Kemudian kerjakan kelas untuk setiap sampel dalam dataset pelatihan dengan menggunakan fungsi sigmoid. Hasilnya adalah kelas = [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]. Hasilnya ditunjukkan seperti Gambar 7.10.

$$\beta_0 = -132.3, \beta_1 = -3.1, \beta_2 = 39.6, \beta_3 = -2.9, \beta_4 = 3.2$$

Angka pada gambar menunjukkan ID orang yang diuji, dan lingkaran di garis putus-putus menunjukkan kelas. Terlihat dari gambar bahwa akurasi klasifikasi dengan regresi logistik dalam hal ini adalah 100%, sehingga model ini dapat diadopsi untuk prediksi. Terakhir, mari kita prediksi apakah seseorang yang datanya {3.16, 5.20, 0.97, 3.49} masing-masing memiliki hiperlipemia. Mengadopsi model di atas dan melakukan penyelesaian-persamaan-per-substitusi, maka kelas = 1. Oleh karena itu, orang tersebut diprediksi mengalami hiperlipemia.

Contoh 7.4 Penggunaan Baysian Classifier dalam Analisis dan Prediksi Diabetes

Contoh ini menganalisis pasien diabetes dan memprediksi apakah mereka telah tertular penyakit tersebut. Prediksi tersebut didasarkan pada pelatihan dari data sampel pada pasien berlabel tentang obesitas dan kadar gula darah mereka. Data sampel diberikan pada Tabel 7.6. Di sini, Ya berarti obesitas atau pasien diabetes dan Tidak untuk berat badan normal atau orang sehat.

Untuk mempermudah, kami menyatakan atribut A untuk obesitas dan atribut B untuk kadar gula darah. Berdasarkan statistik dari Tabel 7.7, kami memperoleh distribusi probabilitas berikut pada obesitas pasien dan kadar gula darah. Untuk memprediksi label kelas seseorang yang mendapat pemeriksaan kesehatan, jika X = (A = Ya, B = 7,9), diperlukan perhitungan P(Ya |X) dan P(Tidak|X). Dengan menggunakan data statistik, kami memiliki:

$$\begin{cases} P(A = Yes \mid Yes) = \frac{3}{4} & P(A = No \mid Yes) = \frac{1}{4} \\ P(A = Yes \mid No) = \frac{2}{5} & P(A = No \mid No) = \frac{3}{5} \end{cases} \begin{cases} P(Yes) = \frac{4}{9} \\ P(No) = \frac{5}{9} \end{cases}$$

Tabel 7.6 Data pemeriksaan kesehatan pasien diabetes.

pengenal	Obesitas (A)	Kadar Gula Darah (B) (mmol/L)	Penderita Diabetes atau Bukan
1	No	14.3	Yes
2	No	4.7	No
3	Yes	17.5	Yes
4	Yes	7.9	Yes
5	Yes	5.0	No
6	No	4.6	No
7	No	5.1	No
8	Yes	7.6	Yes
9	Yes	5.3	No

Adapun indeks kadar gula darah, jika kelas Ya, maka:

$$\begin{cases} \bar{x}_{yes} = \frac{14.3 + 17.5 + 7.9 + 7.6}{4} = 11.83 \\ s^2_{yes} = \frac{(14.3 - 11.83)^2 + (17.5 - 11.83)^2 + \cdots + (7.6 - 11.83)^2}{4} = 18.15 \end{cases}$$

Jika kelasnya Tidak, maka

$$\begin{cases} \bar{x}_{yes} = \frac{4.7 + 5.0 + 4.6 + 5.1 + 5.3}{5} = 4.94 \\ s^2_{yes} = \frac{(4.7 - 4.94)^2 + (5.0 - 4.94)^2 + \cdots + (5.3 - 4.94)^2}{5} = 0.07 \end{cases}$$

Dengan distribusi Gaussian dalam kandungan gula darah, kami memiliki

$$\begin{cases} P(B = 7.9 \mid Yes) = \frac{1}{\sqrt{2\pi} \times \sqrt{18.15}} e^{-\frac{(7.9-11.83)^2}{2 \times 18.15}} = 0.062 \\ P(B = 7.9 \mid No) = \frac{1}{\sqrt{2\pi} \times \sqrt{0.07}} e^{-\frac{(7.9-4.94)^2}{2 \times 0.07}} = 9.98 \times 10^{-28} \end{cases}$$

Saat ini, lakukan klasifikasi untuk X dengan metode klasifikasi naive Bayesian:

$$P(X|Yes) = P(A = Yes|Yes)P(B = 7.9|Yes) = \frac{3}{4} \times 0.062 = 0.0465$$

Tabel 7.7 Hasil probabilistik pada pasien obesitas dan kadar gula darah.

	Kegemukan		**Kadar Gula Darah (mmol/L)**	
penderita diabetes	**Ya**	**Tidak**	**Nilai Rata-rata**	**Perbedaan**
Ya	3/4	1/4	11.83	18.15
Tidak	2/5	3/5	4.94	0.07

Tabel 7.8 Sampel berlabel dari laporan pemeriksaan 20 pasien hiperlipemia.

identitas pasien	Trigliserida (mmol/L)	Jumlah Kolesterol (mmol/L)	Lipoprotein Kepadatan Tinggi (mmol/L)	Lipoprotein Densitas Rendah (mmol/L)	Hiperlipemia atau tidak
1	3.07	5.45	0.9	4.02	1
2	0.57	3.59	1.43	2.14	0
3	2.24	6	1.27	4.43	1
4	1.95	6.18	1.57	4.16	1
5	0.87	4.96	1.36	3.61	0
6	8.11	5.08	0.73	2.05	1
7	1.33	5.73	1.88	3.71	1
8	7.77	3.84	0.53	1.63	1
9	8.84	6.09	0.95	2.28	0
10	4.17	5.87	1.33	3.61	1
11	1.52	6.11	1.29	4.58	1
12	1.11	4.62	1.63	2.85	0
13	1.67	5.11	1.64	3.06	0
14	0.87	3.45	1.25	1.92	0
15	0.61	4.05	1.87	2.05	0
16	9.96	4.57	0.53	1.73	1
17	1.38	5.61	1.77	3.62	0
18	1.65	5.1	1.77	3.16	0
19	1.22	5.71	1.53	3.93	1
20	1.65	5.24	1.47	3.41	1

Dengan cara yang sama, probabilitas P(X|No) diperoleh sebagai berikut dengan estimasi kesalahan:

$$P(X|No) = P(A = Yes|No)P(B = 7.9|No) = \frac{2}{5} \times 9.98 \times 10^{-28} = 3.99 \times 10^{-28}$$

$$\begin{cases} P(Yes\,|X) = \dfrac{P(X|Yes)P(Yes)}{P(X)} = \varepsilon \times \dfrac{4}{9} \times 0.062 = \varepsilon \times 0.0276 \\ p(No\,|X) = \dfrac{P(X|No)P(No)}{P(X)} = \varepsilon \times \dfrac{5}{9} \times 3.99 \times 10^{-28} = \varepsilon \times 2.218 \times 10^{-28} \end{cases} \quad \varepsilon = \frac{1}{P(X)}$$

Kita peroleh P(Ya | X)P(X) = 0,0276 > 2,218 × 10^{-28} = P(X)P(Tidak | X). Oleh karena itu, kelas orang tersebut adalah Ya jika X = (A = Ya, B = 7,9). Dengan demikian, orang tersebut telah memperoleh diabetes.

Contoh 7.5 Pemilihan Metode Deteksi Hiperlipemia atas Data Medis Untuk menentukan apakah siswa menderita hiperlipemia, dilakukan pemeriksaan fisik untuk mengukur trigliserida, kolesterol total, high-density lipoprotein dan low-density lipoprotein, dan proyek lainnya. Tabel 7.8 mencantumkan data mentah dari 20 siswa yang diuji. Di sini, para siswa yang terdeteksi menderita hiperlipemia ditandai dengan "1" dan mereka yang tidak dengan "0" di kolom sebelah kanan.

Tabel 7.9 Kinerja yang diukur dari tiga pilihan classifer yang bersaing.

Algoritma ML	Permintaan Memori (dalam KB)	Waktu Pelatihan (dalam detik)	Ketepatan
Pohon Keputusan	1,768	1.226	90%
KNN	556	0,741	100%
SVM	256	0.196	100%

Berdasarkan sampel kecil yang diperoleh, pilih pengklasifikasi mesin yang sesuai untuk membangun sistem *Machine learning* guna mendeteksi potensi masalah pada siswa. Tiga metode pengklasifikasi kandidat sedang dipertimbangkan. Karena data sampel agak kecil, pilihan terakhir mungkin tidak dapat mencakup kasus data yang benar-benar besar. Kami menggunakan contoh terutama untuk mengilustrasikan pilihan seleksi.

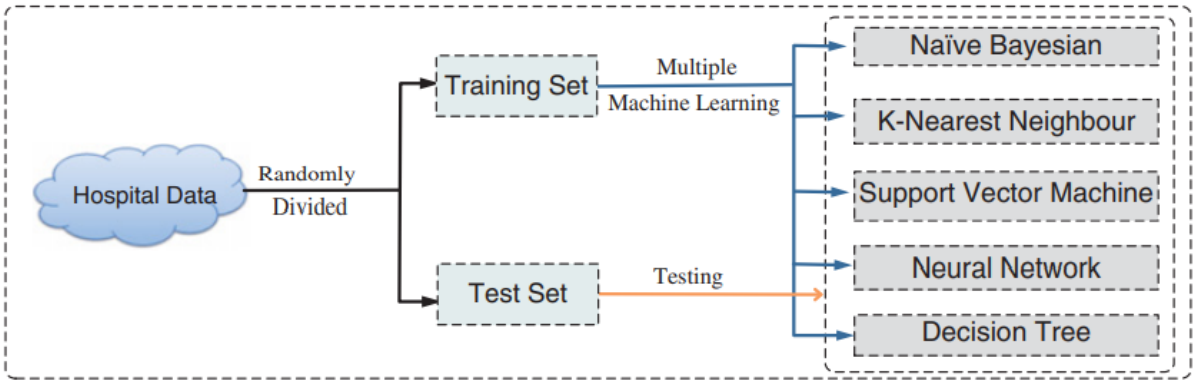
Dengan mengamati kumpulan data sampel, kita mengetahui bahwa semua data memiliki label kelas, sehingga hal ini dapat diselesaikan dengan metode klasifikasi terawasi. Tabel 7.9 merangkum permintaan memori, waktu pelatihan, dan akurasi yang diukur dengan penggunaan tiga kandidat metode *Machine learning*. Dalam hal permintaan akurasi, jelas metode KNN dan SVM sempurna untuk memenuhi tujuan kami. Jika permintaan memori dan waktu pelatihan penting, metode SVM adalah pilihan yang lebih baik.

Analisis Kinerja Lima Metode Deteksi Penyakit

Big data dapat diterapkan untuk memprediksi apakah seseorang termasuk dalam populasi berisiko tinggi untuk penyakit kronis tertentu, berdasarkan informasi pribadi mereka

seperti usia, jenis kelamin, prevalensi gejala, riwayat medis dan kebiasaan hidup (misalnya merokok atau tidak, dll.). Misalnya, kita dapat menentukan model prediksi risiko menggunakan metode *Machine learning* yang diawasi yang dipelajari di Bab 5. Gambar 7.11 mencantumkan lima metode *Machine learning* yang berbeda, yaitu naive Bayesian (NB), k-nearest tetangga (KNN), SVM, jaringan saraf (NN) dan pohon keputusan (DT) yang kami gunakan untuk deteksi penyakit.

Kerangka dasar model ditunjukkan pada Gambar 7.11. Kami secara acak membagi data menjadi data pelatihan dan data pengujian, dan rasio set pelatihan dan pengujian adalah 3:1. Metode yang disebutkan di atas digunakan untuk melatih model.



Gambar 7.11 Lima model *Machine learning* untuk prediksi penyakit berdasarkan *Big data* medis.

Prediksi menggunakan Algoritma Nearest Neighbor

Klasifikasi NB adalah pengklasifikasi probabilistik sederhana yang disajikan dalam Bagian 6.3.3. Berdasarkan vektor fitur input pasien x = (x_1, x_2, … , x_n), kita dapat menghitung p(x|c_i) dan distribusi probabilitas sebelumnya p(c_i). Teorema Bayesian, $p(c_i|x) = \frac{p(c_i)p(x|c_i)}{p(x)}$, diterapkan untuk mendapatkan distribusi probabilitas posteriori, p(c_i | x). Melalui pemecahan masalah argmaxci p(c_i |x), pengklasifikasi NB dapat memprediksi penyakit pasien.

Prediksi Risiko Menggunakan Algoritma Nearest Neighbor

KNN dibahas dalam Bagian 4.3.2. Dalam contoh ini, kami menggunakan jarak Euclidean. Berdasarkan *big data* medis, x = (x_1, x_2, … , x_n) dan y = (y_1, y_2, … , y_n) adalah vektor karakteristik dari dua pasien yang diberikan, dengan masing-masing vektor mengandung n karakteristik tik. Jarak Euclidean antara dua pasien dihitung sebagai berikut: d(x, y) = $\sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$. Parameter K sensitif terhadap kinerja model. Kami memilih dari 5 hingga 25 dalam aplikasi perawatan kesehatan yang khas. Untuk dataset yang kami gunakan, ketika K = 10, model menunjukkan kinerja tertinggi. Jadi, kita atur K ke 10.

Prediksi menggunakan Support Vector Machine

SVM dipelajari di Bagian 4.3.4. Ini digunakan untuk menemukan hyperplane maks untuk membagi ruang n-dimensi menjadi subruang. Dalam aplikasi medis yang khas, vektor

karakteristik pasien x = (x_1, x_2, … , x_n) tidak dapat dipisahkan secara linier. Untuk memetakan data ke ruang fitur yang diubah, menggunakan pembelajaran berbasis kernel, lebih mudah untuk mengklasifikasikan permukaan keputusan linier dan, oleh karena itu, merumuskan kembali masalah sehingga data dipetakan secara eksplisit ke ruang ini. Fungsi kernel dapat memiliki banyak bentuk. Di sini, kami menggunakan kernel fungsi dasar radial (RBF). Pengklasifikasi SVM dapat diimplementasikan menggunakan perpustakaan LibSVM.

Prediksi menggunakan Neural Network

Pengklasifikasi NN diciptakan dengan meniru jaringan saraf biologis. Dalam contoh ini, kita perlu mengatur parameter terlebih dahulu: i) jumlah lapisan. Model NN berisi empat lapisan umumnya, termasuk lapisan input, dua lapisan tersembunyi dan lapisan output dan; ii) jumlah neuron di setiap lapisan. Di sini, dimensi lapisan input sama dengan jumlah karakteristik pasien. Input dilambangkan dengan x = (x_1, x_2, … , x_n). Dalam contoh ini, kami menetapkan 10 neuron di lapisan tersembunyi pertama, sementara menetapkan 5 sebagai jumlah neuron di lapisan tersembunyi kedua. Outputnya hanya memiliki dua hasil, yaitu risiko tinggi atau risiko rendah. Dengan demikian, lapisan keluaran hanya berisi dua neuron.

Setelah membangun struktur NN, kita perlu melatih modelnya. Untuk setiap bobot koneksi w dan bias b di setiap lapisan, kami menggunakan algoritma propagasi balik. Untuk fungsi aktivasi, kami menerapkan fungsi sigmoid.

Prediksi menggunakan Pohon Keputusan

Klasifikasi berdasarkan Pohon Keputusan (DT) disajikan pada Bagian 4.3.1. Ide dasarnya adalah bahwa suatu objek diklasifikasikan dengan meminimalkan pengotor data, yang ditentukan oleh penggunaan perolehan informasi. Perolehan informasi didasarkan pada konsep entropi, yang definisinya adalah sebagai berikut: $\mathrm{H(S)} = -\sum_i p_i \log p_i$, di mana $p_i = |C_{i,s}|/|S|$ adalah probabilitas bukan nol dari Ci. Informasi yang diharapkan diperlukan untuk klasifikasi S menurut atribut A dilambangkan dengan $H_A(S)$. Kemudian, kita dapat memperoleh $H_A(S) = \sum_{v \in V} |S_v|/|S| H(S_v)$, di mana v mewakili himpunan bagian v dibagi dari S menurut atribut A. Kita kemudian dapat memperoleh gain informasi sebagai berikut: Gain(S, A) = H(S) $H_A(S)$.

Untuk menyempurnakan model, digunakan metode 10-fold cross-validation pada training set, dimana data dari peserta testing tidak digunakan pada fase training. Misal TP, FP, TN, dan FN adalah true positive (jumlah instance sah yang diprediksi dengan benar), false positive (jumlah instance sah yang diprediksi salah), true negatif (jumlah instance negatif diprediksi dengan benar) dan false negative (jumlah contoh negatif salah diprediksi), masing-masing. Kami mendefinisikan empat pengukuran: akurasi, presisi, recall dan F1-Measure sebagai berikut:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + \text{FN}} \tag{7.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7.3}$$

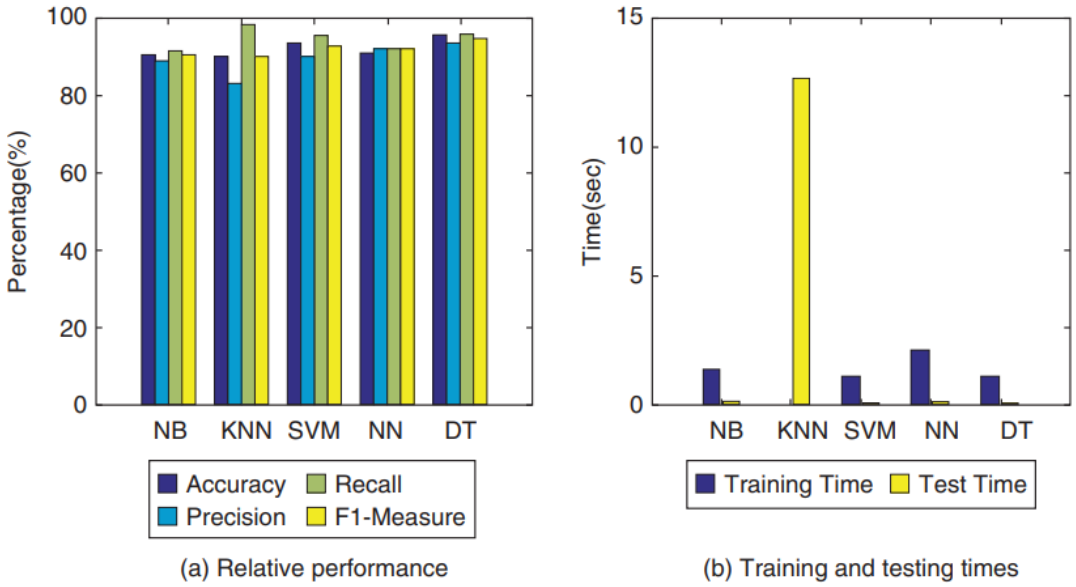
$$\text{F1} - \text{Measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7.4}$$

F1-Measure adalah rata-rata harmonik tertimbang dari presisi dan recall mewakili kinerja keseluruhan. Selain kriteria evaluasi di atas, kami paling sering menggunakan kurva karakteristik operasi penerima (ROC) dan area di bawah kurva (AUC) untuk mengevaluasi pro dan kontra dari pengklasifikasi. Kurva ROC menunjukkan trade-off antara true positive rate (TPR) dan false positive rate (FPR), di mana TPR = TP/(TP + FN) dan TFR = FP/(FP + TN). Ketika area lebih dekat ke 1, semakin baik modelnya.

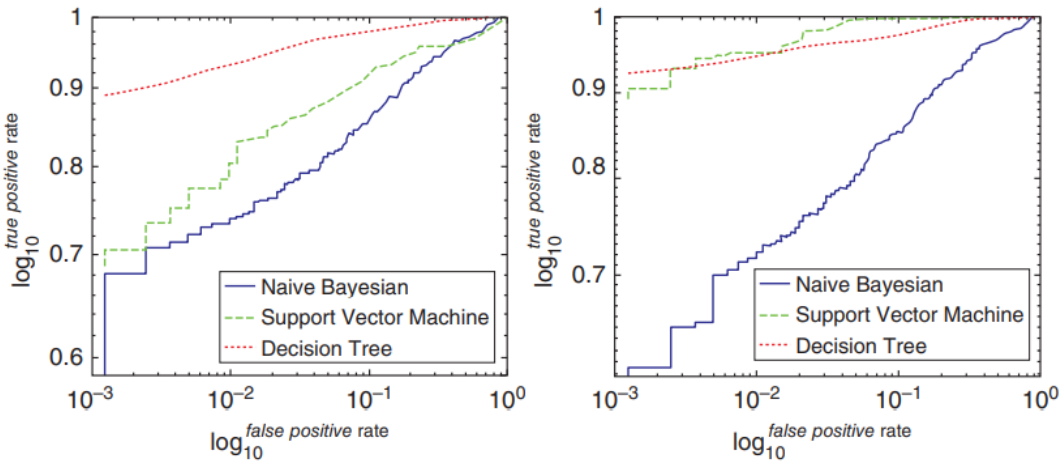
Contoh 7.6 Prediksi Penyakit Berisiko Tinggi dengan Lima Algoritma *Machine learning*

Input ke model adalah nilai atribut pasien, dilambangkan dengan x = (x_1, x_2, … , x_n). Nilai keluarannya adalah C = {c_0, c_1}, dimana c0 menunjukkan apakah pasien termasuk dalam kelas populasi risiko tinggi hiperlipemia, dan c1 menunjukkan apakah pasien termasuk dalam kelas populasi risiko rendah hiperlipemia. Kami memperhatikan akurasi, presisi, recall, dan F1-Measure dari dataset rumah sakit. DT memiliki akurasi tertinggi di set pelatihan dan set tes. Kinerja relatif dan waktu pelatihan dari lima model *Machine learning* diberikan pada Gambar 7.12.

Gambar 7.12(a) memplot akurasi, presisi, tingkat recall dan kinerja F1 dari kelima metode prediksi. Berdasarkan kumpulan data yang telah kami proses, semuanya melakukan rentang yang sama antara 82% dan 95%. Mempertimbangkan akurasi saja, metode SVM dan DT lebih tinggi sekitar 92%, sedangkan tiga metode lainnya tetap sekitar 90%. Dengan ukuran presisi, kami menemukan bahwa NN dan DT lebih baik daripada KNN, yang terendah sekitar 80%. Dalam hal recall rate, metode KNN adalah yang tertinggi, sedangkan empat algoritma lainnya tetap pada level yang sama di atas 90%. Akhirnya, DT memiliki ukuran F1 tertinggi 95%, sementara yang lain tetap sekitar 90%.



Gambar 7.12 Performa relatif dari 5 metode *Machine learning* untuk prediksi penyakit.



Gambar 7.13 Kurva ROC hasil prediksi penyakit menggunakan data rumah sakit.

Singkatnya, dalam hal waktu pelatihan, seperti yang diplot pada Gambar 7.12(b), kami menemukan KNN membutuhkan waktu lebih lama untuk dilatih, sedangkan sisanya memiliki waktu pelatihan yang jauh lebih rendah. Berdasarkan hasil ini, kami memberi peringkat metode DT sebagai yang tertinggi dalam kinerja dan metode KNN sebagai yang terendah dalam skor keseluruhan. Namun, kami harus menunjukkan bahwa hasil peringkat ini sama sekali tidak sama dalam situasi umum. Kinerja relatif sangat sensitif terhadap ukuran dan karakteristik kumpulan data. Dengan hasil ROC, kami menemukan bahwa SVM menunjukkan kinerja yang lebih baik untuk kasus berdimensi tinggi, sedangkan DT bekerja lebih baik untuk kasus berdimensi rendah

(Gambar 7.13). Terakhir, kami merangkum pro dan kontra dari penggunaan kelima model *Machine learning* ini pada Tabel 7.10.

Tabel 7.10 Kekuatan dan kelemahan metode deteksi penyakit

algoritma	Kekuatan	Kelemahan
Bayesian Naif	Mudah diimplementasikan; ketahanan yang kuat terhadap variasi atribut independen di bawah dampak kebisingan, waktu pelatihan singkat dan waktu deteksi cepat	Asumsi atribut terjadi secara independen. Secara umum, akurasi klasifikasi tidak setinggi metode lain
K-Tetangga Terdekat	Mudah dimengerti; tidak ada asumsi tentang distribusi dataset; datanya bisa multidimensi	Kecepatan klasifikasi lambat; semua set pelatihan harus disimpan dalam memori untuk mendapatkan kecepatan pemrosesan dan sensitif terhadap gangguan kebisingan
Mendukung Mesin Vektor	Dapat menangani data berdimensi tinggi; umumnya, akurasinya tinggi; nilai abnormal memiliki kemampuan pemrosesan yang baik	Dengan dimensi yang tinggi, membutuhkan fungsi kernel yang baik; waktu pelatihan lebih lama dengan tuntutan tinggi untuk penyimpanan dan daya CPU
Jaringan syaraf	Menangani beberapa data fitur; kecepatan klasifikasi cepat; itu dapat mengatasi karakteristik yang berlebihan	Waktu pelatihan relatif lama; sensitif terhadap kebisingan konsentrasi
Pohon Keputusan	Kurang ketergantungan pada distribusi data. Klasifikasi data cepat dan mudah untuk menginterpretasikan hasil deteksi	Rentan terhadap masalah fragmen data; alat DT yang bagus sulit ditemukan

Big data Seluler untuk Pengendalian Penyakit

Pendekatan tradisional untuk menganalisis pola mobilitas penyakit sering kali didasarkan pada survei rumah tangga dan informasi yang diperoleh dari data sensus. Kumpulan data yang dikumpulkan secara tradisional ini mengalami bias ingatan dan keterbatasan dalam ukuran sampel populasi yang terlibat dalam analisis, terutama karena biaya yang berlebihan dalam perolehan data dan waktunya. Model mobilitas manusia yang dibantu cloud yang berasal dari data jaringan seluler memiliki potensi untuk mengatasi kekurangan metode tradisional.

Yang menarik dan kuat adalah fakta bahwa ponsel terhubung, meninggalkan jejak digital, yang dapat digunakan untuk menganalisis dan memodelkan perilaku manusia pada tingkat individu dan agregat. Analisis jejak digital ini telah berhasil diterapkan di berbagai bidang, termasuk perencanaan kota, pemodelan mobilitas manusia, dan pemahaman struktur jaringan sosial atau pengukuran pembangunan ekonomi. Dan itu akan diterapkan dalam pengendalian penyakit berbantuan cloud. Dengan popularitas platform seluler dan cloud, ada potensi luar biasa untuk menggunakan berbagai jenis data jaringan seluler untuk kesehatan masyarakat dan pengendalian penyakit. Semakin banyak penelitian yang berfokus pada peluang untuk memodelkan mobilitas penduduk dan untuk mengkarakterisasi penyebaran penyakit.

Beberapa tahun terakhir telah melihat pertumbuhan dramatis dalam lalu lintas seluler. Lalu lintas jaringan seluler global dari perangkat seluler diperkirakan akan melampaui 24 exabyte (satu exabyte kira-kira sama dengan 1018 byte) per bulan pada tahun 2019, yang sembilan kali lebih besar daripada lalu lintas yang dilayani oleh jaringan seluler yang ada pada tahun 2014. volume besar lalu lintas seluler membentuk *Big data* seluler skala besar. Ini menyediakan sumber data yang paling nyaman untuk menganalisis mobilitas populasi untuk pengendalian penyakit yang dibantu cloud.

Mobilitas individu dan agregat manusia tentu saja merupakan variabel kunci untuk mengukur, memodelkan, dan memprediksi kesehatan masyarakat. Model mobilitas manusia dapat dibangun dari data jaringan seluler yang dikumpulkan secara pasif, dengan janji besar untuk membantu pengambilan keputusan dalam pengendalian penyakit, terutama saat memerangi penyakit menular, menghadapi risiko pandemi, atau saat menghadapi konsekuensi bencana alam.

Untuk mempercepat adopsi data seluler untuk pengendalian penyakit, orang perlu mengumpulkan data ponsel pasien, seperti CDR (Call Detail Records) ponsel yang digerakkan oleh peristiwa, data lokasi ponsel, triangulasi sinyal lokasi seluler seperti GPS, dan sebagainya. Masing-masing jenis data seluler memiliki kekuatan dan keterbatasan teknisnya sendiri, jadi kami perlu menggabungkannya untuk memodelkan mobilitas kerumunan pasien.

Dalam skenario pengendalian penyakit, perlu dan penting untuk menggabungkan data ponsel dengan variabel dari sumber data eksternal, seperti informasi kesehatan masyarakat atau rekam medis, data cuaca, data kualitas udara dan informasi sosial pasien. Volume massal data ponsel dan keterkaitan set data yang berbeda menimbulkan tantangan teknis dan privasi yang perlu ditangani. Dalam sistem pengendalian penyakit yang dibantu cloud, orang menggunakan cloud sebagai platform untuk pengumpulan data, penyimpanan data, transmisi data, dan sumber daya computing untuk teknik analisis *Big data*.

Misalnya, variabel mobilitas khas dalam CDR mencakup jumlah stasiun pangkalan yang digunakan, radius girasi (yaitu jarak kuadrat rata-rata akar antara stasiun pangkalan), total jarak yang ditempuh dan diameter area pengaruh. Ini mengacu pada area grafis tempat aktivitas pengguna berlangsung. Ini dapat diukur sebagai jarak maksimum antara BTS. Dalam penularan

malaria, kita mungkin perlu menganalisis CDR dari hampir 15 juta pelanggan seluler Kenya dengan menggunakan model penyebaran epidemi. Pendekatan ini didasarkan pada kenyataan bahwa mobilitas manusia berkontribusi lebih besar terhadap penyebaran malaria daripada penyebaran nyamuk.

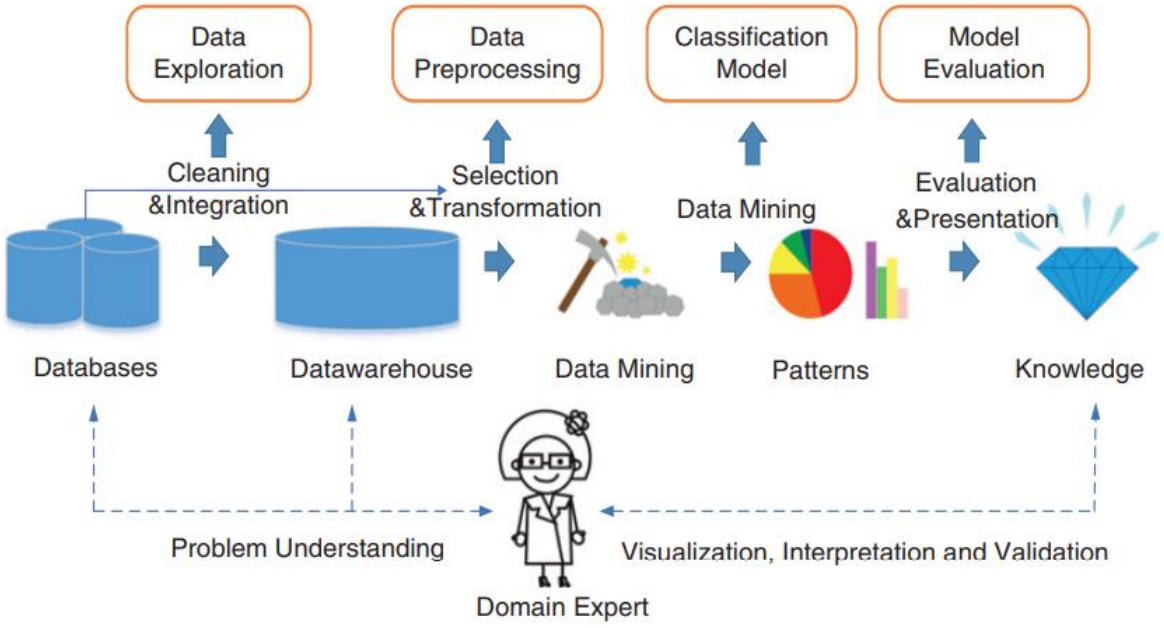
Selain memahami mobilitas penduduk dalam kasus epidemi atau bencana alam, menambang data jaringan seluler dapat memberikan informasi berharga untuk pengawasan kesehatan masyarakat rutin yang berkelanjutan. Setelah gempa Haiti pada Januari 2010, diikuti dengan wabah kolera pada Oktober 2010, para peneliti di Institut Karolinska di Swedia menganalisis data pergerakan harian dari 2 juta ponsel. Mereka mampu mengidentifikasi daerah kritis wabah kolera dan menghitung populasi yang terkena dampak bencana dan pergerakan mereka pada periode berikutnya. Studi ini mengilustrasikan nilai luar biasa bagi petugas pengendalian penyakit dan layanan darurat dari data jaringan seluler ketika tersedia langsung setelah bencana terjadi.

Dari perspektif kesehatan masyarakat dan pengendalian penyakit, menambang data jaringan seluler dengan bantuan cloud berpotensi memungkinkan kami mengidentifikasi populasi pasien dan situasi penyakit di mana intervensi (yaitu pesan, panggilan telepon, atau kunjungan) dapat memicu pos perubahan perilaku yang positif atau mendorong kepatuhan terhadap terapi, yang akan berkontribusi untuk meningkatkan efisiensi pengendalian penyakit dan menurunkan biaya perawatan kesehatan. Namun, ada banyak tantangan dalam menggunakan platform cloud untuk pengendalian penyakit dengan data seluler. Privasi data dan keamanan basis data juga menetapkan beberapa batasan. Misalnya, karena keterbatasan peraturan dan hukum serta kurangnya keahlian teknis, banyak inisiatif penyelamatan dibatalkan untuk menghentikan wabah Ebola di Afrika.

Metode keseluruhan analisis data untuk pengendalian penyakit ditunjukkan pada Gambar 7.14. Penelitian ini menggunakan data mining umum, meliputi data pre-processing, model data-mining dan data post-processing. *Big data* medis harus didiskusikan dengan dokter untuk mendapatkan pemahaman tentang masalah dan datanya. Data rumah sakit disimpan di cloud. Untuk melindungi privasi dan keamanan pengguna, kami membuat mekanisme akses keamanan. Kami pertama-tama memproses data, termasuk pemrosesan dan pengurangan dimensi nilai yang hilang, nilai ulangi, dan nilai pengecualian. Menurut pendapat dokter untuk mengekstrak nilai fitur, kami menggunakan algoritme *Machine learning* untuk mengevaluasi model risiko pasien; akhirnya, model terbaik dipilih melalui evaluasi menggunakan metode matematika.

Agregasi dilakukan pada data pelatihan dengan mengimplementasikan demografi, faktor risiko, kerentanan di mana pra-pemrosesan dilakukan dan transformasi data input. Pembersihan data termasuk pembersihan dan pra-pemrosesan data dengan memutuskan strategi mana yang akan digunakan dalam menangani bidang yang hilang dan untuk mengubah data sesuai dengan persyaratan. Kami pertama-tama mengidentifikasi data medis yang tidak pasti, tidak akurat, tidak

lengkap, atau tidak masuk akal dan kemudian memodifikasi atau menghapusnya untuk meningkatkan kualitas data.



Gambar 7.14 Metode untuk proses prediksi pasien berisiko tinggi yang meliputi tahap eksplorasi, prapemrosesan, dan evaluasi.

Dalam proses pembersihan, kami memeriksa format, integritas, kewajaran, dan batasan data. Pembersihan data sangat penting untuk menjaga konsistensi dan akurasi analisis data. Keakuratan prediksi risiko tergantung pada keragaman fitur data rumah sakit. Kami dapat mengintegrasikan data medis untuk menjamin atomitas data, yaitu kami mengintegrasikan tinggi dan berat badan untuk mendapatkan BMI. Menurut diskusi dengan pakar domain dan analisis korelasi Pearson, kami mengekstrak karakteristik statistik pengguna dan beberapa karakteristik yang terkait dengan hiperlipemia dan kebiasaan hidup (yaitu merokok).

7.4 APLIKASI PERAWATAN KESEHATAN KONTROL EMOSI

Penelitian yang muncul tentang interaksi emosional manusia-mesin didasarkan pada computing yang dapat dikenakan untuk konteks penginderaan IoT dan analitik *Big data* di cloud. Untuk mengaktifkan perawatan emosi otomatis dengan bantuan mesin, sistem harus dirancang untuk mengumpulkan data tentang emosi, ekspresi, atau gerak tubuh manusia. Kontrol emosi menimbulkan persyaratan yang lebih tinggi pada jumlah dan kualitas sinyal tubuh daripada perawatan kesehatan tradisional. Beberapa perawatan kesehatan kontrol emosi membutuhkan interaksi afektif dengan pengguna. Di bagian ini, kami membahas masalah yang relevan dan meninjau beberapa skema solusi yang diusulkan dalam beberapa tahun terakhir.

Sistem Perawatan Kesehatan Mental

Perawatan emosi membantu mereka yang menderita masalah mental atau terjebak dalam depresi berat atau frustrasi dalam kehidupan sehari-hari mereka. Ini termasuk orang tua atau orang yang hidup sendiri, keluarga di bawah garis kemiskinan, warga yang terbelakang atau kurang mampu, pengemudi truk jarak jauh, dan pasien yang sakit secara psikologis, dll. Untuk memecahkan masalah, banyak masyarakat beradab memiliki pekerja sosial atau kekuatan kebijakan untuk membantu keluar. Di sini, kami mempelajari bagaimana Smart Machine dapat digunakan untuk membantu atau memberikan bantuan tambahan di luar agen perawatan manusia. Seseorang dapat menerapkan beberapa indeks fisiologis atau sitologis untuk mendeteksi emosi manusia.

Saat menemukan seseorang dalam suasana hati yang sangat rendah, sistem harus meningkatkan kewaspadaan kepada korban atau mereka yang merawatnya. Misalnya, pengingat suara atau pemutaran musik, dll. Dapat diterapkan untuk menghentikan situasi darurat atau upaya bunuh diri. Robot pendeteksi emosi dapat dirancang untuk memandu keadaan emosi pasien. Perintah interaksi emosi dapat dibantu oleh sumber daya cloud atau IoT. Deteksi emosi menggunakan Wearable Standar 2.0 dapat dirancang untuk menggunakan nilai indeks fisiologis komprehensif, yang dipantau oleh perangkat IoT (termasuk ponsel pintar) atau mesin pintar seperti sistem robot cerdas. Deteksi emosi didasarkan pada ekspresi atau gerak tubuh manusia. Merancang aplikasi seluler khusus mungkin berguna untuk mendeteksi keadaan marah untuk menghindari respons yang tidak pantas. Pendekatan lain termasuk mengintegrasikan data pekerja sosial, informasi lokasi, catatan panggilan telepon seluler, dll, dalam sistem deteksi emosi otomatis.

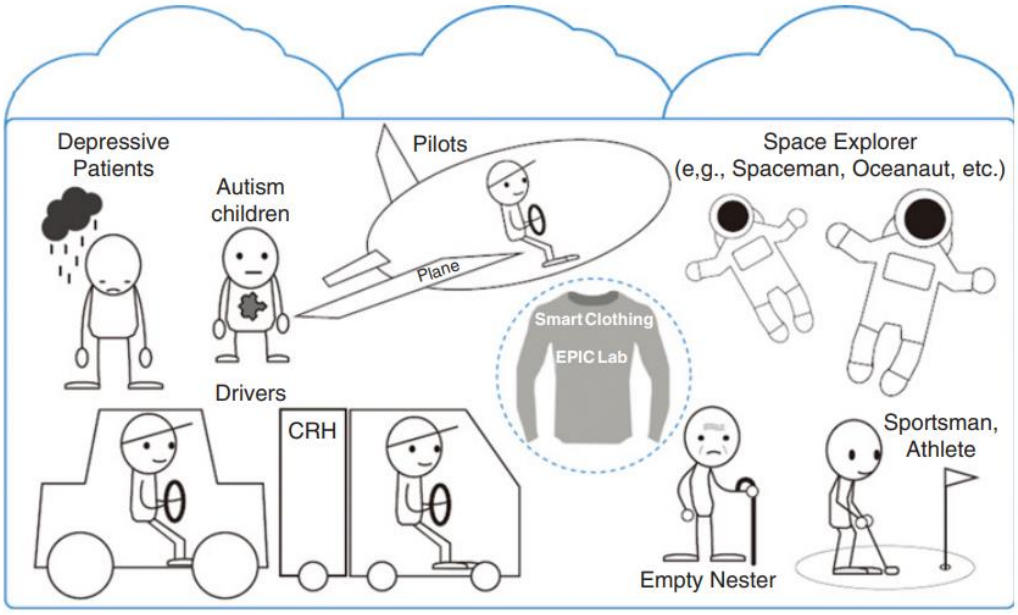
Untuk memberikan perawatan emosi yang tepat dan efektif, kita perlu mengembangkan model emosi berdasarkan pelatihan data fisiologis di cloud. Sistem harus menetapkan respons unik untuk pola emosi pengguna yang berbeda. Misalnya, untuk menggunakan EKG (Elektrokardiografi), sinyal ditransmisikan ke cloud melalui pakaian kebijaksanaan dengan fungsi akuisisi dan transmisi EKG. Saat cloud menerima data EKG, cloud akan melakukan analisis dan pemrosesan secara real time. Selanjutnya, menurut identifikasi unik pengguna, keadaan emosi pengguna diprediksi oleh model terlatih, sedangkan data lain yang dikumpulkan dari terminal seluler dapat membantu prediksi emosi.

Saat mendeteksi bahwa pengguna memiliki emosi negatif, panggilan langsung dilakukan ke peralatan dan sumber daya yang relevan untuk berinteraksi secara emosional dengan pengguna. Misalnya, dengan emosi sedih, untuk memutar musik yang dapat meringankan kesedihan pengguna, sistem bahkan dapat mengirim perintah ke robot di rumah dan membiarkan robot berinteraksi secara emosional dengan pengguna melalui serangkaian metode tindakan. , suara, dll. Dan akhirnya, sistem menyadari efek perawatan emosional. Seperti yang ditunjukkan pada Gambar 7.15, populasi yang membutuhkan perawatan emosional termasuk

orang-orang yang tidak punya apa-apa, pasien depresi, anak autisme, pengemudi jarak jauh, pilot dan angkascloud, tahanan atau budak, dll.

Computing dan Layanan Kontrol Emosi

Meskipun pengembangan sensor dan perangkat yang dapat dikenakan telah melahirkan sejumlah besar aplikasi seluler, pervasif dan cerdas serta model layanan baru, status psikologis pengguna menghadapi tantangan untuk menghadapi meningkatnya tuntutan perawatan emosional. Sebagian besar aplikasi sadar emosi yang ada merasakan emosi melalui hubungan antara emosi pengguna dan pola perilaku penggunaan ponsel. Namun, akurasi inferensi pengenalan emosi dibatasi oleh data skala kecil yang dikumpulkan oleh ponsel pintar, atau bergantung pada proses pelabelan manual yang padat karya, yang membatasi penyediaan perawatan berorientasi emosi yang memadai. Jadi, tujuan utama dari aplikasi tersebut adalah untuk hiburan berbasis ponsel pintar.



Gambar 7.15 Perawatan kesehatan mental untuk kelompok populasi khusus.

Pengumpulan Data dan Ekstraksi Fitur

Perangkat yang dapat dikenakan dan ponsel digunakan untuk mengumpulkan data setiap 30 menit. Data yang terkumpul kemudian dikategorikan menjadi data fisik, data siber dan data jejaring sosial. Data fisik terdiri dari data fisiologis, tingkat aktivitas, informasi lokasi, lingkungan, hidup/mati layar ponsel, dan video tubuh. Data siber meliputi log panggilan telepon, log SMS, log email, dan log penggunaan aplikasi. Data jaringan sosial termasuk SNS. Di sisi lain, status emosional pengguna diperoleh terutama melalui dua metode berikut: i) label diri oleh pengguna; dan ii) label melalui transfer learning. Tabel 7.11 menunjukkan pengumpulan data secara rinci. Pra-pemrosesan data terutama berisi empat aspek berikut: pembersihan data, menghilangkan redundansi, integrasi data, dan normalisasi deret waktu.

Kami membagi data menjadi tiga kelas: data statistik, data deret waktu, dan data konten. Kami memperoleh fitur data fisik, data dunia maya, dan data jejaring sosial. Untuk data fisik, misalnya panggilan, SMS, email, detak jantung, laju pernapasan, suhu kulit dan waktu tidur, kami menghitung jumlah masing-masing atribut. Untuk data deret waktu, misalnya tingkat aktivitas, dengan rentang antara rendah, sedang, dan tinggi, mewakili seseorang yang statis, berjalan, dan berlari. Dengan informasi lokasi, kami mengelompokkan rangkaian data lokasi kami melalui DBSCAN, yang dapat memperoleh lokasi yang dikunjungi pengguna. Untuk data konten, kami menggunakan SentiWordNet untuk menyaring kata-kata yang peka emosi dengan skor milik $[-1, -0.4] \cup [0.4, 1]$.

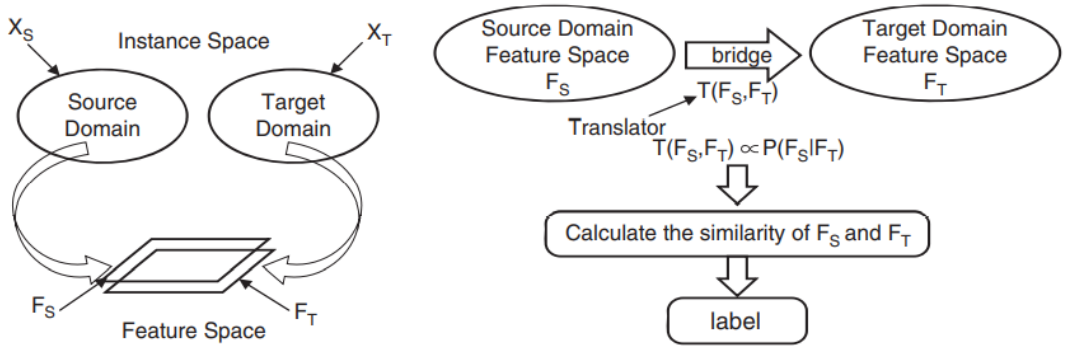
Tabel 7.11 Macam-macam tipe data dalam memberikan layanan pengendalian emosi.

Gaya Data	Tipe data	Petunjuk Penggunaan
Data fisik	Data fisiologis	Detak jantung, Frekuensi pernapasan, Suhu kulit, Durasi waktu tidur
	Tingkat aktifitas	Statis, Berjalan, Berlari
	Lokasi	Koordinat lintang dan bujur, Waktu retensi pengguna
	Lingkungan	Suhu, Kelembaban
	Layar ponsel hidup/mati	Layar waktu hidup/mati
	Video tubuh	Ekspresi wajah, Gerakan kepala, Kedipan mata, dan Video perilaku
Data Cyber	Panggilan	Jumlah panggilan masuk/keluar, Durasi panggilan rata-rata, Jumlah panggilan tak terjawab
	SMS	Jumlah pesan, Panjang pesan, Isi setiap SMS
	Email	Jumlah email yang dikirim/diterima
	Aplikasi	Aplikasi Numbers Office, Peta, Game, Obrolan, Kamera, dan Aplikasi Video/Musik
Data Jejaring Sosial	SNS	ID pengguna dan nama layar, Jumlah teman, Posting konten, posting ulang dan komentar, Posting gambar, posting ulang dan komentar, Konten atau waktu pembuatan Gambar

Pelabelan Berbasis Pembelajaran Transfer untuk Deteksi Emosi

Biasanya, setiap orang memiliki pola perilakunya sendiri dalam hal keadaan perilaku dan kebiasaan hidup, yaitu orang yang berbeda mungkin memiliki sinyal fisiologis dan kebiasaan hidup yang berbeda di bawah emosi yang sama. Seperti yang ditunjukkan pada Gambar 7.16, berbagai orang mengekspresikan emosi kebahagiaan mereka melalui perilaku yang berbeda,

yang dapat dirasakan oleh data multimodal person-centric. Salah satu titik penetrasi utama adalah mencocokkan satu jenis emosi dengan berbagai perilaku pengguna melalui pembelajaran transfer. Konsep transfer learning diilustrasikan di bawah ini. Berbagai tipe data disajikan pada Tabel 7.11 dalam memberikan layanan pengendalian emosi.



(a) The instance space (b) The concept of transfer learning

Gambar 7.16 Instance space dan feature space untuk transfer *machine learning*.

Biarkan X_S menjadi ruang instance sumber, yaitu data yang dikumpulkan yang memiliki label mood, dan biarkan X_T menjadi ruang instance target, yaitu data yang dikumpulkan yang tidak memiliki label mood. F_S dan F_T masing-masing adalah ruang fitur yang terkait dengan X_S dan X_T. Seperti yang ditunjukkan pada Gambar 7.16(a), C menunjukkan ruang label dari sejumlah mode emosional:{kebahagiaan, kesedihan, ketakutan, kemarahan, jijik, kejutan}. Model pembelajaran transfer menerapkan rantai Markov ($c \rightarrow f_s \rightarrow f_t \rightarrow x_t$), dimana $x_t \in X_T$, $f_t \in F_T$, $f_s \in F_S$ dan $c \in C$.

Tujuan kami adalah untuk memperkirakan probabilitas bersyarat $p(c \mid x_t)$. Pertama, kita perlu mencari penerjemah $T(f_t,f_s) \propto p(f_t|f_s)$ untuk menghubungkan dua ruang fitur. Kesamaan fitur digunakan untuk menilai kesamaan domain fitur. Seperti yang ditunjukkan pada Gambar 7.16(b), kami menghubungkan fitur fs dan ft melalui persamaan berikut:

$$D_{JS}(P_T||P_S) = \frac{1}{2}(D_{KL}(P_T||M) + D_{KL}(P_S||M)) \quad (7.5)$$

dimana $M = 1/2(P_S + P_T)$ dan D_{KL} divergensi KL didefinisikan sebagai

$$D_{KL}(P_T||P_S) = \sum_{x \in X} P_T(x) \log \frac{P_T(x)}{P_S(x)} \quad (7.6)$$

Untuk data deret waktu, pertama-tama kita normalkan menjadi [0, 1]. Deret waktu yang dikumpulkan dari domain sumber dan domain target masing-masing dilambangkan dengan MS dan M_T. Dynamic time warping (DTW) digunakan untuk mengukur kesamaan M_S dan M_T sebagai

$$D(i,j) = d(m_i, n_j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\} \quad (7.7)$$

dimana,

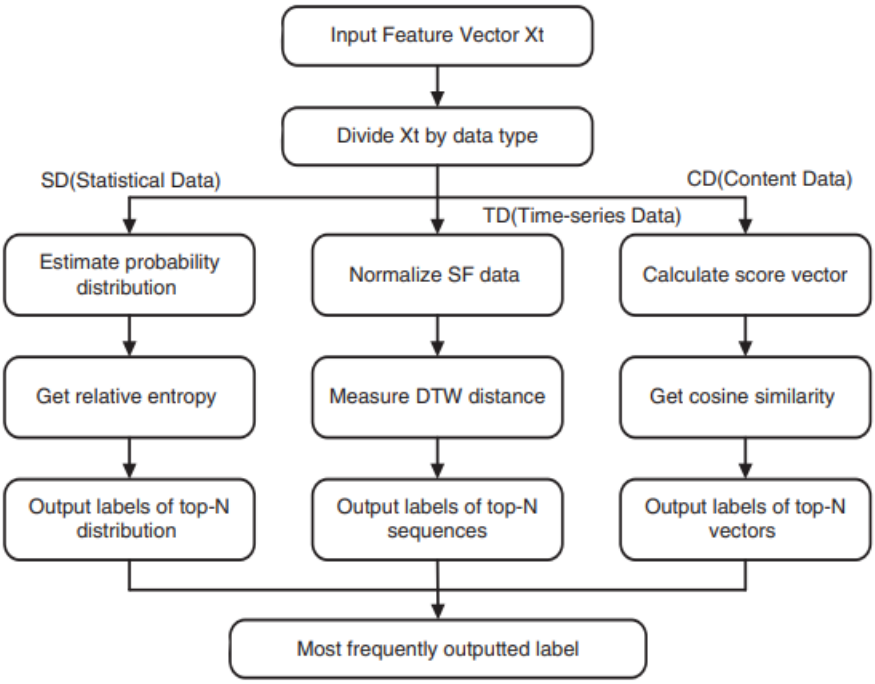
$$d(m_i, n_j) = \sqrt{(m_i - n_j)^2}, m_i \in M_S, n_j \in M_T.$$

Dengan penurunan D(i, j), MS dan MT menjadi lebih mirip. Jadi kita keluarkan deret yang mirip dengan n rendah. Sekarang kita menghubungkan ft dan fs, sehingga kita dapat menghitung label urutan probabilitas paling atas-N.

Untuk teks, kami mengekstrak kata yang skornya antara [−1, −0.4] ∪ [0.4, 1]. Menurut SentiWordNet, vektor skor dari domain sumber dan target dilambangkan dengan V_S dan V_T. Sekarang kami menggunakan kesamaan kosinus untuk mengukur kesamaan antara V_S dan V_T sebagai:

$$\cos(\theta) = \frac{V_S \cdot V_T}{||V_S|| \cdot ||V_T||} \tag{7.8}$$

Sekarang kita hubungkan f_s dan f_t, sehingga kita dapat menghitung label vektor probabilitas paling atas-N. Seperti yang ditunjukkan pada Gambar 7.17, P_S adalah distribusi probabilitas f_s dari domain sumber, misalnya plot frekuensi nilai suhu tubuh. Untuk data fisiologis, panggilan, SMS, email dan aplikasi, kami mengadopsi metode yang sama untuk memperkirakan distribusi. P_T adalah distribusi probabilitas f_t dari domain target, karena divergensi Jensen-Shannon banyak digunakan untuk mengukur kesamaan antara dua distribusi probabilitas. $D_{JS}(P_T \,||\, P_S)$ sama dengan nol jika dan hanya jika kedua distribusi P_T dan P_S identik. Jadi kami mengeluarkan distribusi serupa dengan n rendah. Sekarang kita hubungkan f_t dan f_s, sehingga kita dapat menghitung label distribusi probabilitas N teratas.



Gambar 7.17 Konsep pembelajaran transfer untuk pelabelan emosi.

Interaksi Emosi melalui IoT dan Clouds

Prediksi afektif tradisional telah dihasilkan dari menganalisis satu jenis data emosional. Hal ini dapat menyebabkan ketidakakuratan dalam memvalidasi hasil deteksi. Untuk mengatasi kesulitan ini, kami menyajikan arsitektur deteksi emosi, bernama AIWAC pada Gambar 7.18. AIWAC adalah singkatan dari Affective Interaction through Wearable Computing and Cloud. Sistem ini mengumpulkan data emosional dari berbagai sumber: yaitu ruang siber, fisik, dan sosial. Di ruang fisik, data fisiologis pengguna dikumpulkan, termasuk berbagai sinyal tubuh, seperti EEG, EKG, elektromiografi (EMG), tekanan darah, oksigen darah, dan pernapasan.

Di ruang maya, kami menggunakan komputer untuk mengumpulkan, menyimpan, dan mentransfer konten video wajah dan/atau perilaku pengguna. Di ruang sosial, profil pengguna, data perilaku, dan konten sosial interaktif diekstraksi. Dengan tersedianya layanan jejaring sosial, kerangka kerja IoT (ditunjukkan dalam Bab 2) dan jaringan seluler 4G/5G, data afektif yang dikumpulkan benar-benar merupakan sumber *Big data* selama periode pengamatan yang panjang. AIWAC memberi pengguna dukungan perawatan kesehatan fisiologis dan psikologis. Seperti yang ditunjukkan pada Gambar 7.18, AIWAC dikembangkan menjadi tiga lapisan: i) lapisan terminal pengguna dengan perangkat yang dapat dipakai untuk pengumpulan data fisiologi dan umpan balik emosional; ii) lapisan komunikasi; dan iii) lapisan cloud untuk interaksi afektif. Lapisan ini ditentukan di bawah ini.

Lapisan Terminal Pengguna

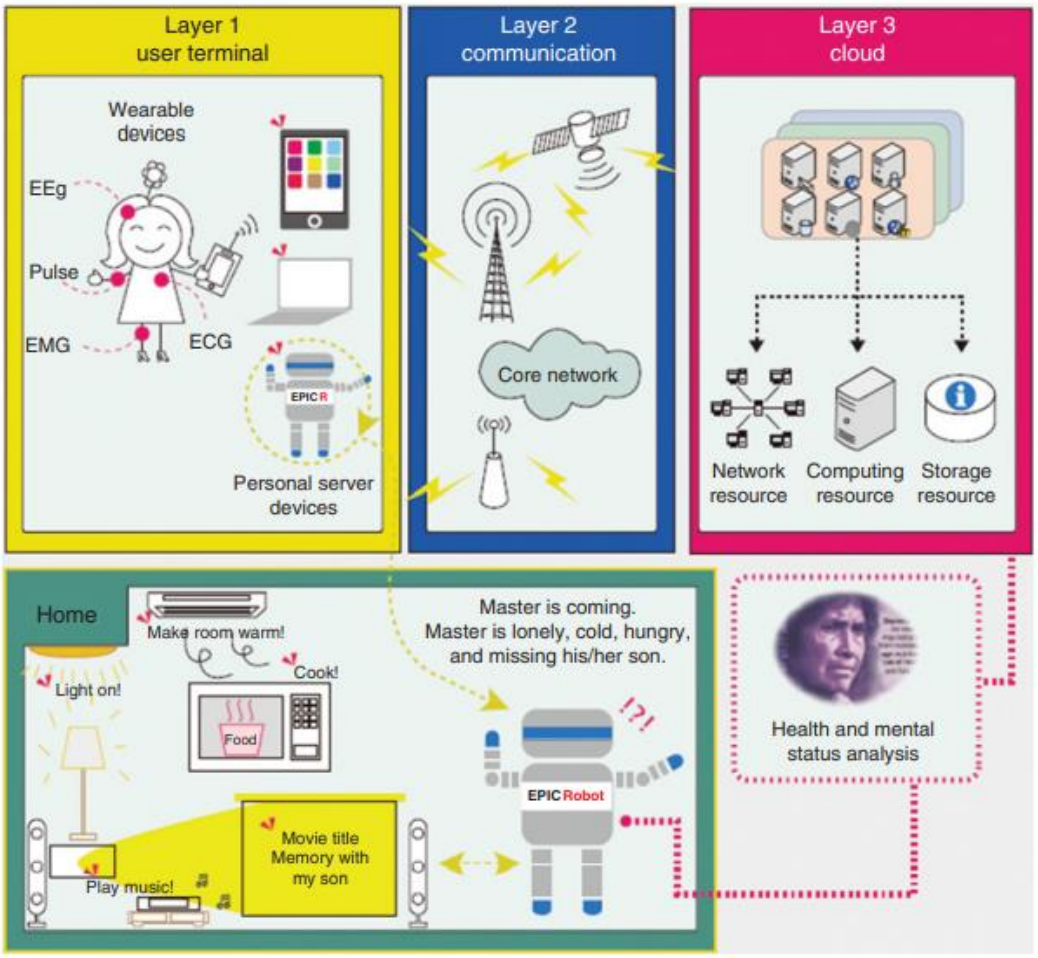
Lapisan ini terdiri dari perangkat wearable dan smart, dimana perangkat wearable terutama digunakan untuk mengumpulkan berbagai data fisiologis, seperti EEG, EKG, elektromiografi (EMG), tekanan darah, oksigen darah, respirasi, dll., sedangkan perangkat pintar memberikan dukungan untuk interaksi emosional. Dalam banyak kasus, terminal adalah perangkat yang dapat dipakai dan juga perangkat pintar. Kami menggunakan terminal robotik untuk memberikan interaksi dan presentasi afektif dengan ketelitian tinggi, dan terutama robot dirancang dengan penampilan antropomorfik dan pola perilaku manusia (suara, senyum, anggukan, dan tindakan lainnya).

Lapisan Komunikasi

Lapisan ini terdiri dari modul store-and-forward dan modul akses komunikasi, dan ponsel pintar, komputer, tablet, dan perangkat pintar lainnya dengan fungsi akses 2G, 3G, 4G atau WIFI, terintegrasi dengan kedua modul ini. Perangkat ini memerlukan dukungan perangkat lunak untuk menerima berbagai data fisiologis dan psikologis real-time yang ditransmisikan dari lapisan terminal pengguna, dan melakukan praproses, memformat, dan mengklasifikasikan data ini (termasuk pengkodean, penguraian kode, pemfilteran, dan operasi lainnya). Melalui modul akses komunikasi, praproses data dikirim ke cloud melalui seluler atau jaringan lain.

Lapisan ini adalah inti dari AIWAC, yang menyediakan analisis data fisiologis dan psikologis melalui pusat data di platform cloud. Pusat data terutama bertanggung jawab untuk penyimpanan data, ekstraksi fitur dan klasifikasi, dan pemodelan emosi individu. Dengan

kekuatan computing yang besar dari layanan berbasis cloud, AIWAC mampu secara efisien menanggapi permintaan afektif dari terminal pengguna. Selain itu, AIWAC dapat menyediakan pemantauan kesehatan masyarakat berdasarkan data fisiologis dan psikologis yang besar.



Gambar 7.18 Arsitektur berlapis dari sistem pemantauan emosi AIWAC (dicetak ulang dengan izin dari Zhang et al., 2015 [17]).

Lapisan Cloud

Kinerja cloud afektif tergantung pada kualitas data yang dikumpulkan. Semakin besar kumpulan data yang dikumpulkan, semakin tinggi akurasi analisis emosionalnya. Namun, sulit untuk memenuhi persyaratan pengumpulan data emosional dengan sumber daya perangkat yang dapat dikenakan yang terbatas. Oleh karena itu, kami menghasilkan bantuan cloud backend untuk melakukan sebagian besar operasi analitik sinyal.

Contoh 7.7 Sistem pemantau emosi AIWAC Dikembangkan di Universitas Sains dan Teknologi Huazhong

Sistem AIWAC ini merupakan prototipe penelitian yang dibangun di Universitas Sains dan Teknologi Huazhong, Cina pada tahun 2015. Idenya diilustrasikan pada Gambar 7.18, di mana

sumber daya cloud berlimpah untuk melakukan operasi yang diperlukan dalam sistem. Platform cloud media ini digunakan untuk menyimpan, mengelola, dan menganalisis data emosional dari ruang multidimensi. Tujuannya adalah untuk mencapai layanan interaktif emosi yang cepat bagi pengguna ponsel. Teknologi computing cloud seluler digunakan untuk mengatasi kekurangan kendala waktu dan ruang yang buruk pada perangkat seluler genggam atau sensor yang dapat dikenakan. Hasil analisis sentimental cloud diumpankan kembali ke klien.

Ketika interaksi emosional sementara terganggu dan lingkungan pengguna berubah, cloud perlu dengan cepat memahami dan mengoptimalkan alokasi sumber daya di lingkungan baru untuk melanjutkan interaksi. Karena emosi pengguna dipengaruhi oleh banyak faktor, berbagai macam data harus dikumpulkan oleh perangkat yang dapat dikenakan untuk membuat penilaian emosi pengguna yang tepat waktu dan akurat. Selain itu, merupakan tantangan besar untuk mengevaluasi jenis data mana yang berguna untuk analisis sentimen. Pertama, pemodelan harus dibuat berdasarkan faktor-faktor yang mempengaruhi emosi pengguna. Kemudian berdasarkan model persepsi emosional yang telah ditetapkan, kami memerlukan bantuan untuk memutuskan perangkat wearable mana yang akan digunakan untuk mengumpulkan data penting.

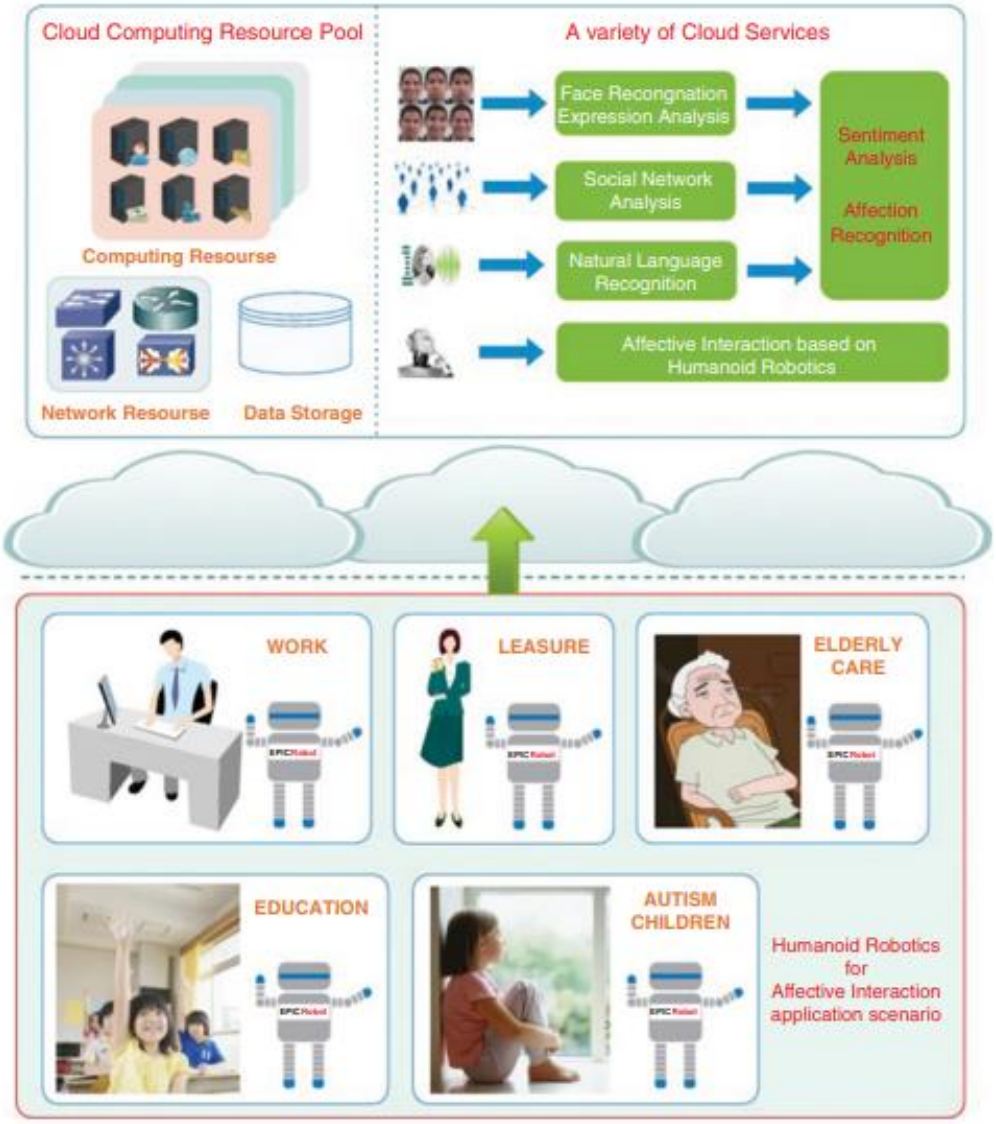
Untuk umpan balik yang manusiawi dan cerdas, interaksi emosional harus afinitas. Dengan analisis dan prediksi emosi pengguna yang akurat, berbagai cara manusiawi untuk interaksi emosional akan secara langsung memengaruhi pengalaman pengguna. Kami bermaksud untuk membangun robot berjalan tegak yang cerdas dengan mengintegrasikan hasil studi interdisipliner di banyak bidang. Robot biofidelitas tinggi yang terintegrasi dengan banyak sensor dapat bekerja sama dengan perangkat pintar lainnya untuk merasakan informasi lingkungan. Robot akan menjadi salah satu pembawa front-end yang paling intim dan dapat diandalkan secara emosional untuk interaksi emosional dengan orang-orang. Sementara itu, mengandalkan komunikasi nirkabel dan teknologi computing cloud, robot tersebut dapat memberikan kecerdasan bergerak.

Kontrol Emosi melalui Teknologi Robotika

Dalam beberapa tahun terakhir, penelitian robot telah menjadi salah satu bidang penelitian paling populer di industri dan akademisi. Terutama robot cerdas humanoid telah menarik banyak minat. Sementara mempromosikan robot industri, negara-negara di seluruh dunia semakin memperhatikan robot layanan cerdas. Domain aplikasi robot berkembang secara bertahap dari industri ke keluarga dan layanan pribadi. Dengan berkembangnya penelitian produk robot dengan kemampuan dan kecerdasan otonom, manusia akan terbebas dari tugas-tugas sederhana, monoton dan berbahaya.

Robot humanoid telah membuat kemajuan besar, tetapi juga menghadapi banyak tantangan teknis untuk membuatnya sepenuhnya terintegrasi ke dalam kehidupan manusia, di antaranya untuk melengkapi robot manusia dengan kemampuan interaksi emosional adalah salah satu masalah yang paling menantang. Di masa lalu, sebagian besar robot dirancang untuk

melakukan pekerjaan berulang sesuai dengan program yang ditulis sebelumnya. Mereka biasanya tidak memiliki kemampuan otonom. Mereka tidak cerdas atau tingkat kecerdasan mereka sangat rendah. Mereka tidak menyadari perasaan manusia. Pada Gambar 7.19, robotika humanoid diperlihatkan dengan interaksi afektif antara AIWAC dan klien.



Gambar 7.19 Robotika humanoid untuk interaksi afektif antara AIWAC dan klien.

Dengan teknologi computing cloud, pengguna tidak perlu memahami setiap detail infrastruktur computing cloud, pengetahuan profesional yang sesuai, atau kontrol langsung. Robot tradisional selalu dibatasi dalam fungsi perangkat keras dan perangkat lunak di mana ada masalah serius. Tetapi computing cloud, sebagai dukungan yang baik untuk teknologi robot, dapat dengan mudah menggabungkan computing cloud dengan teknologi robot untuk membangun robot cloud. Sebagai peralatan front-end, robot bertanggung jawab dalam

pengumpulan sinyal, kinerja tindakan spesifik dan beberapa tugas analisis dan pemrosesan sederhana, sedangkan tugas yang lebih rumit yang membutuhkan cluster computing skala besar akan bergantung pada cloud. Cloud itu sendiri memiliki kemampuan penyimpanan dan penghitungan yang kuat, pelatihan, pembelajaran, dan pembuatan model yang efektif dengan algoritme *Machine learning* tingkat lanjut dan mengirimkan hasil penghitungan atau analisis kembali ke robot. Dengan cara itu, robot akan diberikan otak bijaksana dengan bantuan analisis dan kemampuan pemrosesan cloud yang kuat.

Kami melatih robot dengan kemampuan interaksi emosi dengan menggabungkan teknologi *cloud computing* dengan robot humanoid. Persepsi dan kognisi cerdas diintegrasikan ke dalam robot untuk meningkatkan tingkat kecerdasannya. Berbagai teknologi komunikasi nirkabel diadopsi untuk memastikan stabilitas dan keandalan komunikasi robot dan platform cloud. Karena robot adalah ujung depan untuk berkomunikasi dengan manusia, untuk memastikan interaksi manusia-komputer berkualitas tinggi, robot juga harus memiliki penampilan humanoid yang mudah didekati, dengan tindakan dan ekspresi dasar manusia.

Untuk membuat robot humanoid mampu berinteraksi secara emosional, kita perlu menganalisis sejumlah besar data yang terkait dengan emosi (yaitu ekspresi manusia, bahasa, tindakan, kata-kata dan gambar yang digunakan dalam jejaring sosial, dll.), tetapi robot tidak dapat menyelesaikan tugas analisis sumber daya karena kapasitas pemrosesan dan penyimpanannya yang terbatas. Computing cloud telah mengubah model pengiriman perangkat lunak tradisional, menyediakan computing, penyimpanan, dan sumber daya jaringan kepada pengguna dalam hal layanan.

Arsitektur sistem dibagi menjadi tiga lapisan, yaitu pengguna, robot dan layanan cloud (Gambar 7.19). Pengguna berarti pengguna robot, dan juga objek interaksi emosional. Untuk meningkatkan akurasi pengenalan emosi, pengguna akan diminta untuk memakai perangkat yang dapat dikenakan (yaitu jam tangan pintar atau gelang pintar) untuk mengumpulkan indikator fisiologis pengguna (misalnya suhu tubuh, detak jantung dan gaya berjalan, dll.). Data yang diperoleh oleh perangkat yang dapat dikenakan ini dikirim ke platform cloud back-end melalui robot. Karena sejumlah besar tugas computing dan penyimpanan diselesaikan oleh platform cloud, keandalan koneksi jaringan antara robot dan platform cloud menjadi sangat penting.

Untuk memastikan kelancaran komunikasi robot dan platform cloud, robot perlu dimasukkan dengan berbagai modul komunikasi nirkabel (4G-LTE, 3G, WiFi), dan dapat secara otomatis beralih koneksi jaringan di modul komunikasi yang berbeda. Untuk mewujudkan interaksi emosional, robot perlu dikonfigurasi dengan sensor audio dan video, lampu LED dan komponen lainnya. Tubuh robot harus cukup fleksibel untuk menyelesaikan berbagai tindakan. Untuk memahami informasi lingkungan dari posisi pengguna, robot juga perlu diintegrasikan dengan berbagai sensor lingkungan.

Platform cloud mengadopsi penyimpanan *Big data* dan mesin computing berdasarkan Hadoop dan Spark, dengan bantuan algoritma analisis sentimen berdasarkan *Deep learning* dan mengintegrasikan data yang dikumpulkan oleh robot dan data jaringan sosial pengguna untuk menyelesaikan analisis dan prediksi kondisi emosi pengguna. Cloud mengeluarkan hasil analisis emosi dan instruksi interaksi emosional ke robot, dan membuat robot menyelesaikan tugas-tugas interaktif tertentu. Selama interaksi emosional, robot akan melaporkan efek interaksi emosional real-time ke platform. Platform cloud menyesuaikan interaksi emosional tepat waktu sesuai dengan umpan balik robot. Seperti yang ditunjukkan pada Gambar 7.20, komponen robot meliputi kepala, ekstremitas atas, kaki, perangkat penginderaan, komunikasi, dll.

Sistem cloud adalah otak dari kecerdasan robot humanoid dan pusat penyimpanan data. Cloud menerima aliran data yang stabil dari robot dan jejaring sosial. Data dari robot mencakup persepsi robot terhadap lingkungan sekitar, sinyal fisiologis manusia, dan data terkait lainnya (yaitu ekspresi manusia, suara, gerakan, dll.). Sumber data jejaring sosial termasuk blog Mikro pengguna, foto bersama, video, dll. Karena ada berbagai macam data, data perlu didistribusikan ke mesin penyimpanan yang sesuai sesuai dengan jenis data dan persyaratan sentimen. analisis.

Robot menangkap video dengan kamera secara real time, dan memposisikan ekspresi wajah manusia dengan video yang diambil, dan kemudian mengirimkan gambar ekspresi ke platform cloud jarak jauh untuk identifikasi ekspresi. Platform cloud dapat melatih modul ekspresi wajah manusia dengan sejumlah besar data wajah manusia, dan mengenali ekspresi wajah yang ditransmisikan oleh robot berdasarkan modul pelatihan untuk mengidentifikasi keadaan emosional manusia (yaitu bahagia atau tertekan). Pada saat yang sama, robot juga menerima sinyal suara dan fisiologis pengguna (misalnya detak jantung, tekanan darah), dan sinyal lingkungan sekitar, lalu mengirimkan data ke cloud. Cloud mengumpulkan data pengguna di jejaring sosial. Terakhir, cloud memprediksi keadaan emosional pengguna dengan analisis fusi data. Platform cloud mengirimkan instruksi interaksi afektif ke robot front-end berdasarkan hasil prediksi emosional. Tujuannya adalah untuk menginstruksikan robot untuk mengambil tindakan yang tepat, seperti yang diilustrasikan pada Gambar 7.20.

Sistem Perawatan Kesehatan 5G Cloud-Centric

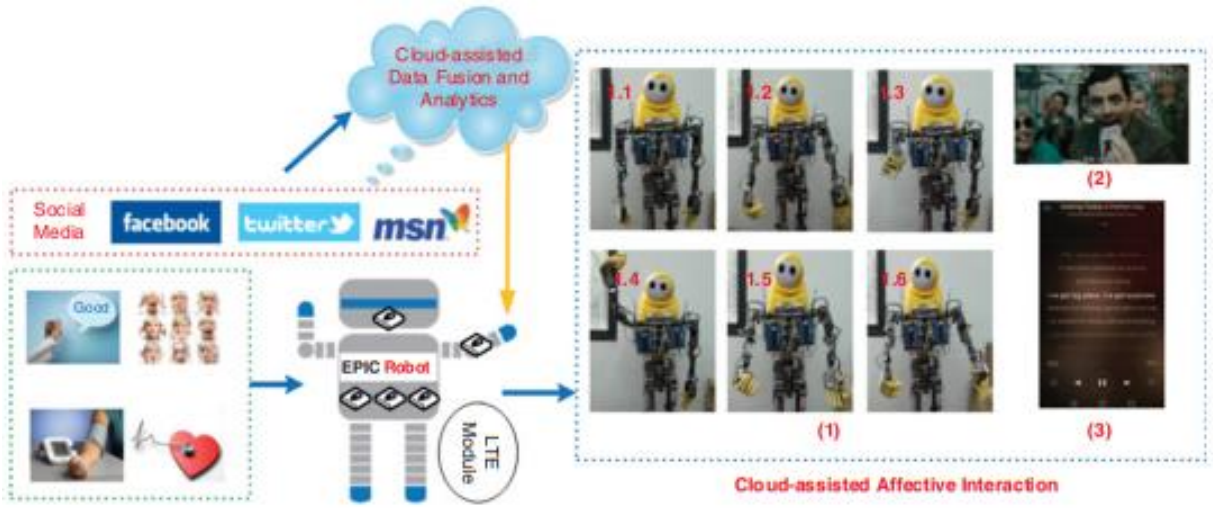
Biasanya, aplikasi kognitif memiliki persyaratan tinggi dalam hal latensi dan keandalan. Jaringan seluler 5G ditujukan untuk memecahkan batasan waktu dan ruang untuk sistem kognitif. Bandwidth seluler yang ditingkatkan menjamin akses yang lebih cepat ke konten, layanan, dan data multimedia untuk aplikasi yang berpusat pada manusia. Pada Gambar 7.21, konsep sistem kognitif cerdas diilustrasikan dengan fitur-fitur berikut:

- Melalui teknologi telekomunikasi masa depan 5G, sensor, perangkat kognitif, dan robot berinteraksi dengan lancar dengan komunikasi latensi rendah yang sangat andal.
- Desain jaringan ditingkatkan sehingga dapat memindahkan data dengan cepat. Untuk pengambilan atau akses *Big data* yang tersimpan, jaringan 5G menghubungkan perangkat

terminal dan pusat data dengan kecepatan yang sangat cepat, memfasilitasi respons pembelajaran yang cepat.

- Belajar dari data adalah inti dari computing kognitif, dan pusat data cloud adalah fasilitas perangkat keras utama untuk pembelajaran lanjutan.
- Computing kognitif membutuhkan banyak data yang tersedia, karena cloud diimplementasikan dan dikonfigurasi untuk menyimpan dan memproses data tersebut.

Untuk membangun sistem kognitif cerdas di era 5G, sistem tersebut perlu menyertakan tiga komponen fungsional:

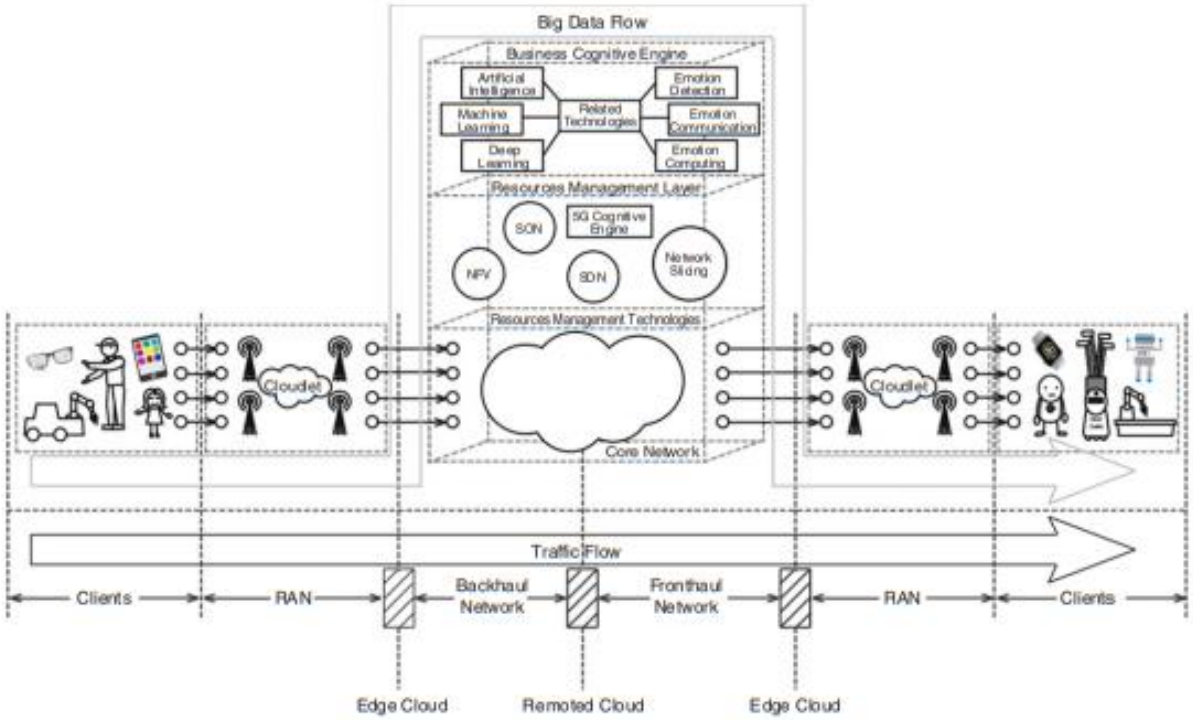


Gambar 7.20 Interaksi afeksi robot berbasis *cloud computing*.

1) **Terminal interaksi perilaku:** perilaku kognitif dalam sistem kognitif harus ditampilkan di terminal; untuk mencapai ini, robot dari berbagai jenis dan fungsi yang semakin kuat adalah alternatif yang menguntungkan;
2) **Komponen persepsi lingkungan:** realisasi kognisi harus didasarkan pada *Big data*, dan komponen kognitif harus mewujudkan persepsi komprehensif pendengaran, penglihatan, sentuhan dan emosi manusia;
3) **Komponen penalaran kognitif:** model penalaran kognitif cerdas dapat secara efektif mensimulasikan proses kognitif manusia, dan teknologi terkait termasuk AI, *Machine learning*, *Deep learning*, computing cloud, dan alat efektif lainnya yang digunakan untuk membangun model penalaran kognitif.

Sistem ini dibagi menjadi tiga lapisan: Lapisan pertama dibangun dengan terminal pintar, RAN berbasis cloud dan Jaringan Inti berbasis cloud. Jaringan akses heterogen menghubungkan terminal pintar, seperti ponsel pintar, jam tangan pintar, robot, mobil pintar, dan perangkat lainnya. Edge cloud dan remote cloud adalah infrastruktur untuk mendukung realisasi fungsi kognitif dalam hal penyimpanan dan sumber daya computing. Lapisan kedua adalah untuk manajemen sumber daya untuk mendukung mesin kognitif sumber daya untuk mencapai

optimalisasi sumber daya dan efisiensi energi yang tinggi. Lapisan ketiga menyediakan kemampuan kognitif data. Dalam mesin kognitif data, AI dan teknik pembelajaran *Big data* digunakan untuk analitik *Big data* kognitif, seperti dalam domain perawatan kesehatan. Aliran *Big data* mewakili proses pengumpulan, penyimpanan, dan analisis *Big data*-besaran dengan dukungan cloud atau IoT. Arus lalu lintas terdiri dari paket dan pesan kontrol selama komunikasi ujung ke ujung pengguna.

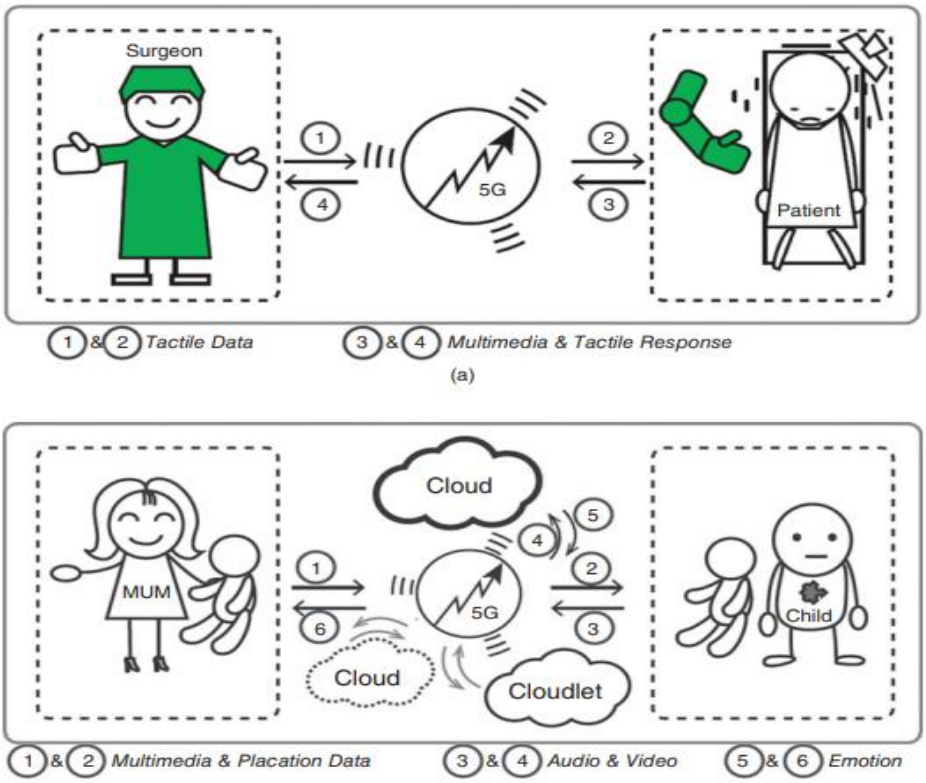


Gambar 7.21 Arsitektur sistem kognitif berbasis cloud/IoT/5G cerdas.

Dua aplikasi pola dasar dari sistem kognisi cerdas ditunjukkan pada Gambar 7.22. Dalam telemedicine, operasi jarak jauh dapat dirancang untuk menyelamatkan nyawa dalam domain perawatan kesehatan. Menggunakan jaringan 5G, tindakan operasi kritis dan persepsi haptic dari ahli bedah akan dipetakan ke lengan robot di meja operasi jarak jauh dengan penundaan yang sangat singkat dan keandalan yang tinggi. Selain itu, semua data vital pasien dapat diproses dengan alat analisis di cloud jarak jauh secara real-time untuk memandu tim penyelamat melakukan beberapa operasi penyelamatan jiwa awal sebelum membawa pasien ke rumah sakit. Aplikasi pola dasar kedua adalah untuk mendeteksi emosi manusia dengan bantuan robot pintar, yang berinteraksi dengan cloud untuk melakukan beberapa tindakan responsif untuk menenangkan pasien. Banyak eksperimen penelitian telah disarankan di masa lalu. Sistem berbasis cloud/IoT dapat membantu memecahkan masalah pengendalian emosi di masa mendatang.

7.5 KESIMPULAN

Dalam bab ini, kami telah berfokus pada aplikasi *Big data* di bidang bio-medis dan perawatan kesehatan. Namun, kumpulan data yang telah kami uji dalam contoh kasus yang diilustrasikan tidak cukup besar dalam skala untuk menarik kesimpulan umum tentang kumpulan data TB atau PB. Bab ini membutuhkan latar belakang dari bab-bab sebelumnya. *Big data* dan cloud menuntut perombakan besar-besaran terhadap program pendidikan kami di bidang sains dan teknologi. Tidak ada solusi unik atau umum untuk masalah *Big data*, karena ketergantungan yang besar pada domain aplikasi tertentu.



Gambar 7.22 Dua aplikasi sistem kognitif cerdas.

Kita harus memanfaatkan penggunaan cloud dan analitik *Big data* dalam menyimpan, memproses, dan menambang *Big data*, yang berubah dengan cepat dalam ruang dan waktu. Cloud, seluler, IoT, dan jejaring sosial mengubah dunia kita, membentuk kembali hubungan manusia, mempromosikan ekonomi global, dan memicu reformasi sosial dan politik dalam skala dunia. Metode *Machine learning* tersebut dapat bekerja secara berbeda, jika set data non-medis atau non-kesehatan diproses atau diuji. Namun, mempelajari metodologi *Machine learning* lebih penting dalam ilmu *Big data* umum dan aplikasi computing cloud.

Tugas dan Latihan

1. Membangun sistem demo pemantauan perawatan kesehatan berdasarkan teknologi *Big data*. Persyaratan dan saran terperinci adalah:

a) Anda dapat menggunakan TI CC2530 Development Toolkit-CC2530DK atau CrossBow TelosB. Setidaknya satu sinyal fisiologis dapat dikumpulkan, seperti suhu tubuh, detak jantung, atau saturasi oksigen darah.
b) Sinyal fisiologis harus ditransmisikan ke platform proses *Big data* Hadoop atau Spark dalam periode tertentu. Mencapai analisis sederhana dan presentasi visual untuk data fisiologis yang dikumpulkan.

2. Dalam masyarakat saat ini, jejaring sosial telah menjadi alat penting bagi orang untuk berkomunikasi. Selain itu, semakin banyak orang mengekspresikan ide dan emosi mereka di jejaring sosial. Dengan demikian, analisis emosi berdasarkan data jejaring sosial memiliki signifikansi praktis yang kuat. Sekarang berikan sekelompok kumpulan data teks yang diekstrak dari Twitter[1], di mana jenis bahasa teks adalah bahasa Arab dan jumlah sampel teks dari emosi positif dan emosi negatif adalah 1000, yang secara manual ditandai oleh tiga ahli linguistik. Algoritma RNN dalam *deep learning* memiliki pengaruh yang besar dalam pengolahan klasifikasi teks. Sekarang, lengkapi aplikasi klasifikasi untuk dataset ini berdasarkan algoritma RNN dan bandingkan hasil klasifikasi dengan algoritma yang digunakan dalam literatur [2]. Jika akurasi lebih rendah dari algoritme dalam literatur, coba debug aplikasi untuk meningkatkan akurasi klasifikasi.
3. Saat ini, kanker telah menjadi pembunuh utama, di mana kanker payudara telah mempengaruhi sebagian besar pasien wanita. Dengan demikian, ada signifikansi praktis yang besar untuk diagnosis kanker payudara yang efektif. Ada sekelompok dataset standar untuk deteksi kanker payudara. Merancang dan mengimplementasikan aplikasi diagnosis kanker payudara berdasarkan algoritma *Machine learning* dengan menggunakan alat *Machine learning* seperti Spark MLlib dan memilih algoritma *Machine learning* yang sesuai. Berdasarkan dataset yang ada dan aplikasi yang Anda buat sendiri, mohon dipikirkan bagaimana meningkatkan akurasi diagnosis kanker payudara sebanyak mungkin. Informasi dasar dari dataset ditunjukkan pada Tabel 7.12.

Tabel 7.12 Atribut dan karakteristik data untuk deteksi kanker.

Properti	Nilai
Jumlah Instance	699
Jumlah Atribut	10
Properti Atribut	01. Contoh nomor kode: nomor id
	02. Ketebalan Rumpun: 1–10
	Keseragaman Ukuran Sel: 1–10
	Keseragaman Bentuk Sel: 1–10
	Adhesi Marjinal: 1–10
	Ukuran Sel Epitel Tunggal: 1–10

	Inti Telanjang: 1–10
	Kromatin hambar: 1–10
	Nukleolus Normal: 1–10
	Mitosis: 1–10
Kelas	2 untuk jinak, 4 untuk ganas

Unduh URL DataSet:
http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29
Unduh URL Kumpulan Data:
http://archive.ics.uci.edu/ml/datasets/Twitter+Data+set+for+Arab+Sentiment+Analysis
Unduh URL Sastra:
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6716448&newsearch=true&queryText=Arabic%20Sentiment%20Analisis:%20Corpus-based%20and%20Lexicon-based

4. Analisis data memiliki aplikasi di banyak bidang. Kita dapat menggunakan sejumlah besar data cuaca untuk menghitung probabilitas hari cerah atau hari hujan. Bisakah Anda membangun aplikasi yang dapat menganalisis data iklim dan memprediksi cuaca? Anda mungkin memerlukan pengetahuan tentang Naive Bayesian dan Anda dapat menggunakan kumpulan data melalui tautan berikut: http://cdiac.ornl.gov/ftp/ndp026b/
5. Facebook adalah salah satu platform media sosial paling populer saat ini. Satu miliar orang berbagi pengalaman kehidupan sehari-hari mereka di Facebook. Beberapa topik sangat menarik dan menarik banyak orang untuk membahasnya bersama. Bisakah Anda membangun aplikasi untuk mengetahui item yang terkait dengan topik hangat berdasarkan teknologi analisis data? Anda mungkin juga perlu menggunakan teknologi perayap web untuk mengumpulkan data. Berikut adalah beberapa informasi yang berguna:
   - Anda harus memiliki akun Facebook dan mendaftarkan hash kunci yang dihasilkan di Facebook untuk aplikasi Anda.
   - Dokumentasi resmi dapat ditemukan di tautan ini: http://developers.facebook.com/docs/reference/api/
   - Anda dapat menggunakan URL ini untuk mengumpulkan beberapa informasi pribadi: https://graph.facebook.com/fql?q=SELECT status_id, time, source, message FROM status where uid=me()&access_token=
6. Biasanya, sistem pemantauan kesehatan mengadopsi perangkat yang dapat dipakai atau perangkat pintar untuk mengumpulkan berbagai data fisiologis. Jika perawatan emosional diperlukan, sistem membutuhkan Smart Machine yang lebih kuat untuk memungkinkan interaksi emosi. Di antara berbagai metode deteksi fisiologis, EEG dapat

merekam aktivitas otak secara akurat. Melalui analisis data pola EEG, kita dapat secara efektif mendiagnosis epilepsi, penyakit mental, dll. Terdapat penelitian besar untuk menguji korelasi EEG dari kecenderungan genetik terhadap alkoholisme. Ini berisi pengukuran dari 64 elektroda yang ditempatkan pada sampel kulit kepala pada 256 Hz. Menggabungkan dengan pengetahuan terkait *Machine learning* dan menggunakan toolkit Weka, ConvNetJS, dll., coba rancang aplikasi apakah pasien memiliki kecenderungan genetik untuk alkoholisme atau tidak, berdasarkan dataset EEG ini. Unduh URL kumpulan data EEG:
http://archive.ics.uci.edu/ml/datasets/EEG+Database

7. Dalam beberapa tahun terakhir, penelitian tentang robotika menjadi semakin populer. Di bidang kontrol emosi, robotika humanoid telah menunjukkan keunggulannya dalam interaksi afektif. Di [1], ada dataset navigasi robot yang mengikuti dinding. Data dikumpulkan saat robot SCITOS G5 menavigasi melalui ruangan mengikuti dinding searah jarum jam, selama empat putaran, menggunakan 24 sensor ultrasound yang diatur dalam lingkaran di sekitar "pinggangnya".
   Unduh dan coba gunakan pengklasifikasi saraf nonlinier, seperti jaringan MLP, untuk melakukan analisis dan pemrosesan kumpulan data ini yang dikombinasikan dengan pengetahuan yang telah Anda pelajari. Kemudian bandingkan hasil dengan klasifikasi menggunakan jaringan saraf berulang (misalnya jaringan Elman). Agar robot yang mengikuti dinding menyelesaikan tugas tersebut dengan sukses, Anda harus meningkatkan akurasi pengklasifikasi saraf sebanyak mungkin berdasarkan keunggulan kedua pendekatan tersebut. Unduh URL kumpulan data navigasi robot yang mengikuti dinding: http://archive.ics.uci.edu/ml/datasets/Wall-Following+Robot+Navigation+Data
8. Penyakit Parkinson (PD) adalah penyakit kronis yang disebabkan oleh gangguan pergerakan sistem saraf pusat dan disfungsi ganglia basal. Biasanya, gaya berjalan merupakan indikator penting untuk mengidentifikasi dan mengevaluasi PD. Untuk mengevaluasi perubahan gaya berjalan lansia dengan penyakit Parkinson secara terus menerus tanpa campur tangan manusia, tekanan langkah kaki dapat diukur saat pasien PD berjalan, dan mode center of pressure (CoP) dapat diperoleh. Cari tahu perbedaan CoP antara orang normal dan pasien PD. Pernyataan mana yang benar?
   a) Sensor tekanan ditempatkan di bawah kaki pasien PD.
   b) Sensor tekanan dipasang di tanah.
   c) Untuk mendapatkan CoP, tekanan bagian depan, tengah atau belakang kaki harus dikumpulkan.
   d) Ukur data tekanan saat pasien PD berdiri atau berjalan.

BAB 8
PEMBELAJARAN PENGUATAN MENDALAM DAN ANALISIS MEDIA SOSIAL

8.1 SISTEM *DEEP LEARNING* DAN INDUSTRI MEDIA SOSIAL

Dalam bab ini, kami memperkenalkan perpustakaan perangkat lunak atau platform yang telah dikembangkan oleh industri dan akademisi untuk aplikasi *machine learning* (ML) dan *deep learning* (DL). Seperti yang telah dipelajari di bab-bab sebelumnya, *Deep learning* adalah bagian dari keluarga metode *Machine learning* yang lebih luas. Perbedaannya terletak pada representasi pembelajaran data. Misalnya, pemeriksaan citra sinar-X direpresentasikan dalam banyak cara seperti vektor, matriks, atau tensor. Beberapa representasi diilhami oleh kemajuan ilmu saraf.

Sistem *Deep learning* dan Dukungan Perangkat Lunak

Sejauh ini, kami telah mempelajari *Machine learning* dan algoritma *Deep learning*. Dengan data dalam jumlah besar, cloud memiliki sumber daya yang cukup untuk melatih model hingga sempurna.

Tabel 8.1 Perbandingan lima pustaka perangkat lunak sumber terbuka untuk aplikasi *Deep learning*.

Perangkat Lunak, Pembuat, Bahasa/Antarmuka, Lisensi, dan Situs Web	**Platform, Alat Perangkat Lunak**	**Model DL Didukung**	**Deskripsi Singkat**
Caffe, Berkeley Vision and Learning Center, C++, Python, MATLAB, BSD2, http://caffe.berkeleyvision.org/	AWS, OSX, Windows, OpenCL, CUDA	RNN, CNN	Kerangka kerja DL mengadopsi arsitektur C++/CUDA murni untuk kemudahan peralihan antara CPU dan GPU
CNTK, Microsoft, C++, Python, .NET, Gratis, http://github.com/Microsoft/CNTK	Windows, Linux, OpenMP, CUDA	RNN, CNN	Perangkat lunak DL gratis untuk aplikasi lintas platform
TensorFlow, Tim Google Brain, C/C++, Python, Apache 2.0, https://www.tensorflow.org/	OpenCL pada peta jalan, CUDA	RNN, CNN, RBM, DBN	Berdasarkan DistBelief, memungkinkan tensor mengalir melalui grafik ANN dari satu ujung ke ujung lainnya

Theano, U. of Montreal, SD, Python http://deeplearning.net/software/theano	Lintas platform, Open MP, CUDA	RNN, CNN, RBM, DBN	Kerangka kerja yang menggunakan perpustakaan ANN modular Torch fora yang mendukung operasi GPU dan CPU bersama
Obor, Ronan Collobert, C, Lua, BSD, http://torch.ch/			Dibangun dengan iTorch dan fbcunn, untuk meningkatkan kinerja ANN dalam visi komputer dan pemrosesan bahasa alami

Deep learning tentu saja merupakan bidang *Machine learning* yang diperluas. Ini bertujuan untuk mendapatkan fitur tingkat tinggi abstrak melalui kombinasi fitur tingkat rendah. Kami membangun jaringan saraf tiruan untuk mensimulasikan fungsi pembelajaran dan analisis otak manusia, untuk memahami berbagai jenis *Big data* seperti gambar, audio, dan teks.

Lima perpustakaan perangkat lunak *Deep learning* dibandingkan pada Tabel 8.1, dalam hal kemampuan pemodelan, antarmuka, kinerja, lintas platform, dan kinerja yang dilaporkan. Kemampuan pemodelan adalah metrik kunci untuk menilai kegunaannya dalam aplikasi *Deep learning*. Kami mengevaluasi kemampuan setiap toolkit untuk melatih jaringan saraf umum dan canggih. Perangkat jaringan saraf mencakup ANN yang dikembangkan dalam beberapa tahun terakhir, seperti ConvNets CNN: AlexNet, OxfordNet, GoogLeNet; RNN: RNN biasa, LSTM/GRU, RNN dua arah, dll.

Caffe adalah toolkit paling populer dalam komunitas visi komputer. Secara umum, dukungannya untuk jaringan berulang dan pemodelan bahasa terbatas karena arsitektur warisannya. Caffe memiliki antarmuka pycaffe tetapi itu hanyalah alternatif sekunder untuk antarmuka baris perintah. Model harus didefinisikan di Google protobuf. CNTK, alat *Deep learning* terpadu oleh Microsoft Research, lebih dikenal di komunitas pidato daripada di komunitas DL umum. Seperti TensorFlow dan Theano, CNTK ditetapkan sebagai grafik simbolis dari operasi vektor, seperti penambahan/perkalian matriks atau konvolusi. Lapisan dalam jaringan saraf hanyalah komposisi dari operasi tersebut. Granularitas halus dari blok bangunan (operasi) memungkinkan pengguna untuk menemukan jenis lapisan kompleks baru.

TensorFlow mendukung model tercanggih: RNN, CNN, RBM, dan DBN; begitu juga Theano dan Torch. TensorFlow menggunakan grafik simbolik dari pendekatan operasi vektor, dan setiap aliran computingnya dibangun sebagai grafik statis. Itu membuat beberapa perhitungan menjadi sulit, seperti pencarian sinar. TensorFlow mendukung dua antarmuka: Python dan C++.

Theano memiliki implementasi untuk sebagian besar jaringan saraf tercanggih. Perangkat lunak ini memelopori tren penggunaan grafik simbolik untuk memprogram jaringan. API simbolisnya membuat penerapan RNN menjadi mudah dan efisien. Minimnya antarmuka tingkat

rendah membuat Theano kurang menarik bagi pengguna industri. Torch mendukung jaringan saraf convolutional dengan baik. Antarmuka asli untuk konvolusi temporal di Torch membuatnya lebih intuitif untuk digunakan. Torch berjalan di LuaJIT, yang sangat cepat. Namun, Lua bukanlah bahasa arus utama, yang membatasi penerapannya secara prospektif. Semua paket perangkat lunak *Deep learning* berjalan di sistem komputer atau kluster cloud yang dibangun dengan akselerator CPU, GPU, atau TPU.

Banyak produk layanan *Deep learning* telah dikembangkan oleh industri sosial dan layanan web, termasuk Facebook, Google, Microsoft, Twitter, Baidu dan WeChat, dll. Berikut kami ulas beberapa contoh sistem ML atau DL dari industri. Misalnya, banyak produk baru Google terkait dengan peningkatan mesin pencari Google. Dengan mempelajari teknologi dan alat pengembangan yang mendasarinya, kami dapat mengembangkan aplikasi kami sendiri untuk memenuhi permintaan khusus. Tidak berarti, apakah kami bertindak untuk mendukung produk tertentu. Pembaca didorong untuk belajar dari produk layanan berkualitas tinggi dari semua perusahaan IT dan jaringan.

Tentu saja, *Deep learning* dengan JST telah terbukti memiliki beberapa keberhasilan yang mengesankan dalam beberapa tahun terakhir. Seperti yang telah kami ulas di Tabel 8.1, banyak produk dan sistem ML dan DL yang menarik dikembangkan oleh industri dan akademisi. Misalnya, kesuksesan mesin pencari Google dalam beberapa tahun terakhir dicapai dengan menambahkan banyak fitur cerdas baru. Ketika memulai dengan sistem PageRank oleh salah satu pendirinya, pencariannya agak sederhana. Selama bertahun-tahun, produk layanan cerdas seperti Gmail, Google Map, YouTube, dan beberapa fungsi pencarian yang dipersonalisasi, ditambahkan ke layanan Google. Dalam konteks ini, banyak alat produktivitas juga dikembangkan oleh perusahaan IT besar; yang terkenal adalah Apple Apps Store, perpustakaan *Machine learning* AWS, paket Spark MLlib, platform TensorFlow Google, dll.

Microsoft Office 365 menawarkan Outlook Web Access (OWA) berbasis cloud untuk mengelola email dari puluhan ribu organisasi. Demikian pula, iCloud menawarkan layanan email gratis untuk mengunci pengguna iPhone atau iPad mereka. Produk menarik lainnya termasuk Siri dan ID sidik jari Apple, WatsApp dan Cendekia Google, layanan SalesForce CRM, WeChat Tencent dan Microsoft OneDrive, dll. Semua aplikasi yang berpusat pada manusia ini telah membangun beberapa fitur Smart Machine. Fitur-fitur ini harus memuaskan sejumlah besar klien individu, serta banyak perusahaan di sektor bisnis, pendidikan dan pemerintahan.

Pada 15 Maret 2016, Google memperkenalkan Google Analytics 360 Suite. Ini adalah rangkaian produk data dan analisis pemasaran terintegrasi, yang dirancang khusus untuk memenuhi kebutuhan pemasar kelas perusahaan. Ini mungkin bersaing dengan penawaran cloud pemasaran yang ada oleh Adobe, Oracle, Salesforce, dan IBM. Sistem terjemahan mesin saat ini berupaya mencakup hampir 100 bahasa yang berbeda, termasuk pengenalan tulisan tangan. Ucapan dapat digunakan sebagai input, dan teks yang diterjemahkan dapat diucapkan melalui

sintesis ucapan. Perangkat lunak ini menggunakan teknik linguistik korpus, di mana program "belajar" dari dokumen yang diterjemahkan secara profesional.

Selain pemrosesan ucapan/bahasa, persepsi mesin, pemahaman gambar otomatis, analisis penglihatan, dan produk AR/VR adalah produk yang sangat populer di industri TI dan hiburan. Untuk memberi pembaca kami perasaan tentang teknologi terdepan dalam menghadirkan AI ke cloud saat ini, kami sekarang memeriksa Proyek Google Brain untuk mengembangkan sistem *Machine learning* dan produk layanan kognitif.

Prinsip Pembelajaran Penguatan

Kami telah memperkenalkan definisi dasar pembelajaran penguatan (RL) di Bab 5. Prinsip-prinsip operasional RL akan diilustrasikan lebih lanjut di bagian ini. Penggunaan algoritme RL akan diberikan di Bagian 8.3, bersama dengan studi kasus program Google DeepMind untuk mencapai penghargaan maksimal dalam proses *Deep learning*. RL memang merupakan subkelas dari ML yang tidak diawasi, karena pasangan input/output yang benar tidak pernah ditampilkan. David Silver dari Google DeepMind telah memberikan tutorial tentang pembelajaran penguatan, yang diterapkan dalam program AlphaGo.

Pembelajaran penguatan dianggap sebagai kerangka tujuan umum kecerdasan buatan. Secara matematis, proses keputusan Markov (MDP) diterapkan dalam lingkungan belajar. RL menekankan kinerja online dengan menemukan keseimbangan antara eksplorasi wilayah yang tidak diketahui dan eksploitasi pengetahuan saat ini dalam proses pengambilan keputusan. Secara formal, model RL dibangun dengan lima bagian berikut:

1) Lingkungan belajar yang dicirikan oleh seperangkat keadaan;
2) Serangkaian tindakan yang dapat dilakukan oleh agen RL. Setiap tindakan mempengaruhi keadaan agen di masa depan. Agen memiliki kapasitas untuk menilai konsekuensi jangka panjang dari tindakannya;
3) Aturan transisi antara status RL;
4) Aturan yang menentukan imbalan langsung dari transisi keadaan;
5) Aturan yang menentukan apa yang dapat dipatuhi oleh agen.

Aturan di atas sering bersifat stokastik atau probabilistik. Pengamatan melibatkan imbalan skalar yang terkait dengan transisi terakhir. Agen RL berinteraksi dengan lingkungannya dalam langkah waktu yang berbeda. Ada tradeoff antara imbalan jangka panjang dan jangka pendek. Algoritme RL telah berhasil diterapkan dalam kontrol robot, penjadwalan elevator, radio kognitif, pemecahan masalah logistik, dan permainan seperti catur, catur, permainan Go dan Atari, dll. Singkatnya, idenya adalah memilih tindakan untuk memaksimalkan imbalan masa depan. Hal ini sangat mirip dengan situasi di mana siswa mencoba berbagai metode belajar untuk mendapatkan nilai terbaik sehingga mencapai karir yang memuaskan sebagai imbalan masa depan.

Algoritma RL mendorong penggunaan sampel untuk mengoptimalkan kinerja dan penggunaan pendekatan fungsi untuk menangani lingkungan yang besar. Kedua fitur ini membuat RL sangat efektif dalam menangani tiga lingkungan *Machine learning*: i) lingkungan

model yang dikenal dengan solusi analitik; ii) lingkungan optimasi berbasis simulasi; dan iii) lingkungan yang memungkinkan pengumpulan informasi dengan berinteraksi dengan itu. Asumsi dasar pada lingkungan pembelajaran penguatan meliputi:

- Semua peristiwa bersifat episodik sebagai rangkaian episode. Sebuah episode berakhir ketika beberapa keadaan terminal tercapai.
- Apa pun tindakan yang mungkin diambil agen, pemutusan hubungan kerja tidak dapat dihindari.
- Harapan dari total imbalan didefinisikan dengan baik untuk setiap kebijakan dan setiap distribusi awal atas negara bagian.

Orang yang cerdas harus mampu menyusun algoritma RL untuk menemukan kebijakan dengan keuntungan maksimum yang diharapkan. Algoritma perlu mencari kebijakan yang optimal untuk mencapai reward yang maksimal. Seringkali, kami menerapkan kebijakan stasioner deterministik, yang memilih tindakan secara deterministik, hanya berdasarkan keadaan saat ini atau terakhir yang dikunjungi. Ada sejumlah pendekatan dalam merancang algoritma pembelajaran penguatan.

Metode brute force adalah memilih kebijakan dengan pengembalian yang diharapkan terbesar. Kesulitan utama dengan pendekatan ini adalah bahwa pilihan kebijakan bisa sangat besar atau bahkan tidak terbatas. Pendekatan fungsi nilai berusaha menemukan kebijakan yang memaksimalkan pengembalian dengan mempertahankan serangkaian perkiraan pengembalian yang diharapkan untuk beberapa kebijakan. Metode ini mengandalkan teori MDP, di mana optimalitas didefinisikan lebih kuat dari yang di atas. Suatu kebijakan disebut optimal jika mencapai pengembalian yang diharapkan terbaik dari setiap keadaan awal. Dari teori MDP diketahui bahwa kebijakan yang optimal selalu menghasilkan pemilihan tindakan yang optimal dengan nilai tertinggi dari setiap keadaan.

Skema RL lainnya termasuk metode perbedaan waktu, yang memungkinkan kebijakan berubah sebelum nilai penghargaan diselesaikan. Metode pencarian kebijakan langsung menemukan kebijakan yang baik dengan mencari langsung dari ruang kebijakan. Baik metode berbasis gradien dan bebas gradien termasuk dalam kelas ini. Metode berbasis gradien dimulai dengan pemetaan dari ruang dimensi (parameter) hingga ke ruang kebijakan. Metode pencarian kebijakan seringkali terlalu lambat untuk mencapai pilihan yang optimal. Pembelajaran penguatan sering dikaitkan dengan model pembelajaran atau perolehan keterampilan manusia. Dalam Bagian 8.3, kami menerapkan ide pembelajaran penguatan untuk mengimplementasikan program AlphaGo.

Industri Media Sosial dan Dampak Global

Di Bab 7, kita telah mempelajari analitik *Big data* untuk membangun sistem perawatan kesehatan otomatis dengan pakaian pintar, robotika, dan cloud. Ini telah menyediakan obat-obatan yang dipersonalisasi dan analitik preskriptif, intervensi risiko klinis dan analitik prediktif untuk mengurangi pemborosan, mengotomatiskan pelaporan eksternal dan internal data pasien,

menggunakan istilah medis standar, dan menyediakan pendaftaran pasien dan solusi perawatan waktu nyata. Di bidang pendidikan, kami juga melihat kemajuan dalam melatih siswa dalam aplikasi *big data*.

Industri media sosial bergerak menjauh dari media datar seperti surat kabar, majalah, dan acara televisi. Sebaliknya, e-book, pembayaran seluler, mobil Uber, belanja online, dan jejaring sosial secara bertahap menjadi arus utama. Caranya adalah dengan menangkap atau menargetkan pengguna pada waktu yang optimal di lokasi yang ideal. Tujuan utamanya adalah untuk menyampaikan pesan atau konten yang sesuai dengan pola pikir konsumen. Misalnya, surat kabar elektronik dan e-book menggantikan buku cetak dan surat kabar. Penargetan konsumen terkait erat dengan metode pengambilan data, lebih dari sebelumnya. Ini paling baik dilihat dengan menemukan korelasi IoT dan *Big data*.

Berbagai teknologi penginderaan IoT telah mengubah cara industri media, perusahaan bisnis, dan bahkan pemerintah beroperasi. Hal ini berdampak pada pertumbuhan ekonomi dan daya saing. Industri media sosial menyediakan alat yang dimediasi komputer yang memungkinkan orang atau perusahaan untuk membuat, berbagi, atau bertukar informasi. (https://en.wikipedia.org/wiki/Social_media#cite_note-Buettner2016b-1).

Layanan media sosial disajikan dalam empat area berikut dalam aktivitas kita sehari-hari:

- Layanan media sosial adalah bagian dari aplikasi layanan web Web 2.0.
- Konten yang dibuat pengguna adalah sumber kehidupan organisme media sosial.
- Pengguna membuat profil khusus layanan untuk organisasi media sosial dan situs web.
- Media sosial memfasilitasi perkembangan jejaring sosial online dalam aktivitas sosial dan bisnis.

Media sosial memungkinkan perubahan mendasar pada komunikasi antara bisnis, organisasi, komunitas, dan individu. Perubahan ini menuntut industri media sosial untuk beroperasi dari banyak sumber ke banyak penerima. Ini berbeda dari media tradisional yang beroperasi dari satu sumber ke banyak penerima. Teknologi media sosial mengambil banyak bentuk yang berbeda termasuk blog, jaringan bisnis, jaringan sosial perusahaan, forum, mikroblog, berbagi foto, ulasan produk/layanan, bookmark sosial, permainan sosial, jejaring sosial, berbagi video dan dunia virtual, dll. .

Contoh 8.1 Antarmuka Pemrograman Aplikasi Media Sosial (API)

Antarmuka pemrograman aplikasi (API) adalah perangkat lunak pertama yang mengakses komputer, situs web, atau platform cloud. API ini memungkinkan pengguna atau pemrogram untuk mulai menggunakan sistem yang diprogram. API media sosial digunakan di jejaring sosial, pesan instan, layanan kencan, Kehidupan kota, pribadi, layanan lokasi, hobi, perjalanan, sumber kerumunan, blogging, Obrolan, perpesanan, dan Avator, dll. Tabel 8.2 mencantumkan sepuluh API representatif untuk media sosial besar aplikasi data. Kami mengkarakterisasi setiap API dengan fungsionalitas, protokol, format data, dan keamanan yang diterapkan.

Semua penyedia komputer, cloud, dan media sosial memiliki alat API mereka sendiri. Pembaca harus mengunjungi situs web mereka untuk mempelajari alat API khusus yang akan digunakan dalam penambangan *Big data*, prapemrosesan, *Machine learning*, dan aplikasi analitik. Di antara mereka, REST dikenal dengan protokol paling populer, JASON JSON adalah format yang paling banyak digunakan, dan kunci API untuk sebagian besar kontrol keamanan. Yang tercantum di atas hanyalah beberapa yang representatif, dan masih banyak lagi untuk berbagai perusahaan IT dan situs web sosial.

8.2 PENGENALAN TEKS DAN GAMBAR MENGGUNAKAN ANN DAN CNN

Bagian ini memperkenalkan bagaimana memanfaatkan jaringan saraf tiruan (JST) dan jaringan saraf konvolusi (CNN) untuk mencapai pengenalan angka tulisan tangan. Kami memberikan pseudo-code untuk membantu pembaca memahami struktur dari algoritma *deep learning* yang diterapkan. Rincian ANN dan CNN dibahas di Bab 6. Kami menerapkan CNN dalam pengenalan wajah manusia dan analisis teks medis. Pengenalan angka tulisan tangan adalah masalah klasifikasi. Seperti yang ditunjukkan pada Gambar 8.1, inputnya adalah gambar angka tulisan tangan dan outputnya adalah angka yang diubah dari gambar. Agar pembaca dapat belajar dan berlatih dengan mudah, kami menggunakan kumpulan angka tulisan tangan klasik, MNIST, sebagai kumpulan data aplikasi. MNIST mencakup 60.000 gambar angka tulisan tangan, dan setiap gambar berukuran 28 × 28 piksel.

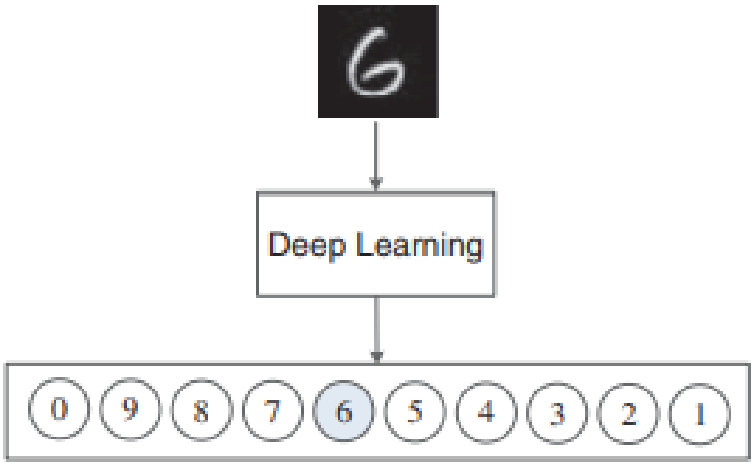
Tabel 8.2 Antarmuka pemrograman aplikasi media sosial (API).

Nama API	Kegunaan	Protokol diterapkan	Format data	Keamanan
API Grafik Facebook	Pemrosesan grafik sosial Facebook, deteksi komunitas dan pencarian teman, dll.	ISTIRAHAT	JSON	OAuth
Google+ API	Untuk menyediakan akses ke Google+, situs web media sosial dengan opsi tautan, status, dan foto	ISTIRAHAT	JSON	Kunci API, OAuth
API Sebutan Sosial	Akses terprogram untuk berinteraksi dengan situs web Sebutan Sosial, API RESTful	HTTP	PHP	Kunci API
API yang lezat	Izinkan pengguna mengakses, mengedit, dan mencari bookmark	ISTIRAHAT	JSON, RSS	OAuth, HTTP/Dasar
MySpace API	Untuk mengakses berbagai fungsi MySpace dan mengintegrasikan aplikasi ke dalam MySpace	Javascript	Tidak dikenal	OAuth

API Pertemuan	Untuk menggunakan Topik, Grup, dan Acara yang dibuat oleh Meetup ke dalam aplikasi mereka sendiri	ISTIRAHAT	JSON, XML KML, RSS	PAith, kunci API
FindMeOn API v.1,0	Akses terprogram ke pencarian media sosial dan fungsi manajemen FindMeOn	HTTP	JSON	Kunci API
API Fliptop	API Orang untuk mendapatkan data sosial berdasarkan alamat email, atau menggunakan pegangan Twitter/Facebook untuk mendapatkan pengembalian data	ISTIRAHAT	JSON, XML	Kunci API
Cisco JTAPI	Cisco Java Telephony API memungkinkan aplikasi Java berinteraksi dengan sumber daya telepon	SABUN, HTTP	XML	Dukungan SSL
API Data YouTube v3.0	Lakukan tindakan yang tersedia di situs web YouTube	ISTIRAHAT, HTTP	JSON	Kunci API

Pengenalan Angka menggunakan TensorFlow untuk ANN

Contoh berikut menunjukkan cara menggunakan TensorFlow dalam memprogram jaringan saraf tiruan (JST), yang disebut pengklasifikasi MINST. Dengan contoh ini, pembaca dapat memperoleh wawasan konkret tentang operasi TensorFlow. Sistem pengklasifikasi ini dapat diterapkan pada pengenalan angka tulisan tangan. JST diperkenalkan di bagian 7.2.



Gambar 8.1 Sistem *Deep learning* untuk mengenali angka tulisan tangan.

Contoh 8.2 Memprogram JST dengan TensorFlow

Kami mempertimbangkan konstruksi ANN 4-lapisan, yang disebut pengklasifikasi MNIST. Ada empat langkah untuk membangun JST mereka. Kami menentukan prosedur setiap langkah, secara terpisah, menggunakan kode semu dengan komentar, yang agak mirip dengan kode Python.

Langkah 1: **Kumpulkan data:** Kami menggunakan data MNIST yang diambil dari situs web Yann LeCun: http://yann.lecun.com/index.html. TensorFlow telah menyertakan beberapa kode Python (bernama input_data.py). Data akan diinstal saat menjalankan file yang diunduh. Kami mengimpornya dengan kode Python import input_data. Kode Python dan penjelasannya ditentukan di bawah ini:

```
# import tensorflow, numpy and input_data to this program
import tensorflow as tf
import numpy as np
import input_data
# load the data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
trX, trY, teX, teY = mnist.train.images, mnist.train.labels,
 mnist.test.images,
mnist.test.labels
```

Langkah 2: **Membangun model JST:** Kami memilih jaringan saraf 4-lapisan untuk membangun classifier, yang berisi satu lapisan input, dua lapisan tersembunyi dan satu lapisan output. Kode Python untuk langkah ini diberikan di bawah ini. Kode berikut mendefinisikan model secara eksplisit. Ada dua lapisan tersembunyi dan tiga putus sekolah. Putus sekolah berarti bahwa beberapa bobot node tidak berfungsi di jaringan, pekerjaan node tersebut sementara dianggap sebagai bagian dari struktur jaringan, tetapi bobotnya dipertahankan (hanya sementara tidak diperbarui). tf.matmul adalah fungsi perkalian dan tf.nn.relu adalah sejenis fungsi aktivasi:

```
// The following is for weight initialization in the ANN
  construction
def init_weights(shape):
return tf.Variable(tf.random_normal(shape, stddev=0.01))
def model(X, w_h, w_h2, w_o, p_drop_input, p_drop_hidden):
X = tf.nn.dropout(X, p_drop_input) #dropout
h = tf.nn.relu(tf.matmul(X, w_h))
h = tf.nn.dropout(h, p_drop_hidden) # dropout
h2 = tf.nn.relu(tf.matmul(h, w_h2))
h2 = tf.nn.dropout(h2, p_drop_hidden) # dropout
return tf.matmul(h2, w_o)
```

Kode berikut mendefinisikan placeholder. X bukanlah nilai spesifik, melainkan placeholder, nilai yang akan kita masukkan saat kita meminta TensorFlow untuk menjalankan computing. Kami ingin memasukkan sejumlah gambar MNIST,

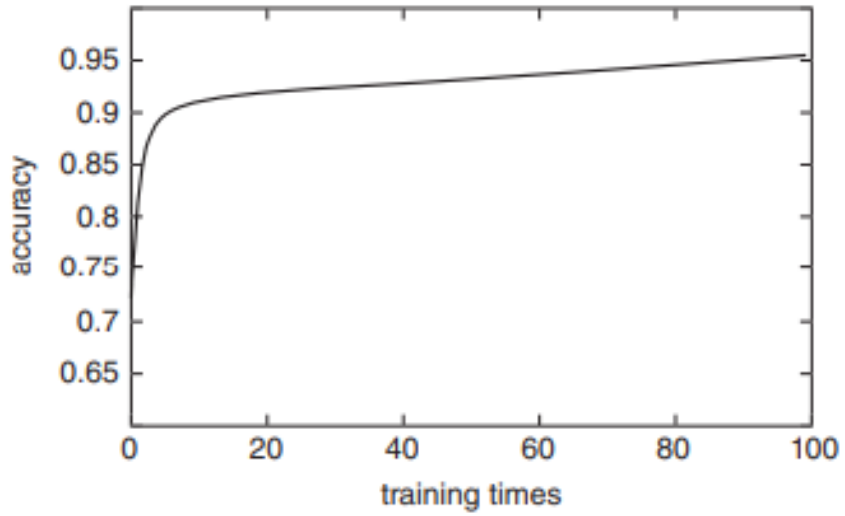
masing-masing diratakan menjadi vektor 784 dimensi. Kami merepresentasikan ini sebagai tensor 2-D dari bilangan floating-point, dengan bentuk [None, 784]. Di sini None berarti bahwa suatu dimensi dapat memiliki panjang berapa pun. Demikian pula, Y adalah vektor 10 dimensi yang mewakili 10 angka, melalui inisialisasi bobot. Entitas w_h adalah matriks 784 × 625, w_h2 matriks 625 × 625, dan w_o matriks 625 × 10:

```
X = tf.placeholder("float", [None, 784])
Y = tf.placeholder("float", [None, 10])
w_h = init_weights([784, 625])
w_h2 = init_weights([625, 625])
w_o = init_weights([625, 10]) // Define p_keep as the
                                 probability of dropout
p_keep_input = tf.placeholder("float")
p_keep_hidden = tf.placeholder("float") // The model is
                                           set as follows
py_x = model(X, w_h, w_h2, w_o, p_keep_input, p_keep_hidden)
```

Langkah 3: **Latih model:** Dengan membandingkan output data pelatihan dan labelnya, algoritme akan menyesuaikan parameter jaringan. Kode Python diberikan di bawah ini: bagian ini mendefinisikan cross-entropy sebagai fungsi kerugian. Kemudian kami meminta TensorFlow untuk meminimalkan cross-entropy menggunakan algoritme RMSPropOptimizer. argmax adalah fungsi yang sangat berguna, yang memberi kita indeks entri tertinggi dalam tensor di sepanjang beberapa sumbu. Misalnya, tf.argmax(y,1) adalah label yang dimiliki model kita untuk setiap input, sedangkan tf.argmax(y_,1) adalah label yang benar. Kita dapat menggunakan tf.equal untuk memeriksa apakah prediksinya benar:

```
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits
 (py_x, Y))
train_op = tf.train.RMSPropOptimizer(0.001, 0.9).minimize(cost)
predict_op = tf.argmax(py_x, 1) //Create a session object to
                                   launch the graph
sess = tf.Session()
init = tf.initialize_all_variables()
sess.run(init)
for i in range(100):
 or start, end in zip(range(0, len(trX), 128), range(128,
len(trX), 128)):
sess.run(train_op, feed_dict = {X: trX[start:end],
    Y: trY[start:end], p_keep_input: 0.8, p_keep_hidden: 0.5})
  print i, np.mean(np.argmax(teY, axis=1) ==
  sess.run(predict_op, feed_dict= {X: teX, Y: teY,
  p_keep_input: 1.0, p_keep_hidden: 1.0}
  endfor
  endfor
```

Langkah 4: **Uji jaringan:** Algoritme akan membandingkan keluaran data uji dan label yang sesuai dan menghitung akurasinya. Data latih digunakan untuk melatih parameter model, tetapi data uji tidak digunakan untuk melatih parameter. Jadi kita bisa menggunakan data uji untuk mendapatkan akurasi model kereta. Seperti yang ditunjukkan pada Gambar 8.2, akurasi menjadi lebih tinggi setelah setiap pelatihan. Setelah melatih sistem selama 100 kali, akurasi 0,9851 tercapai.



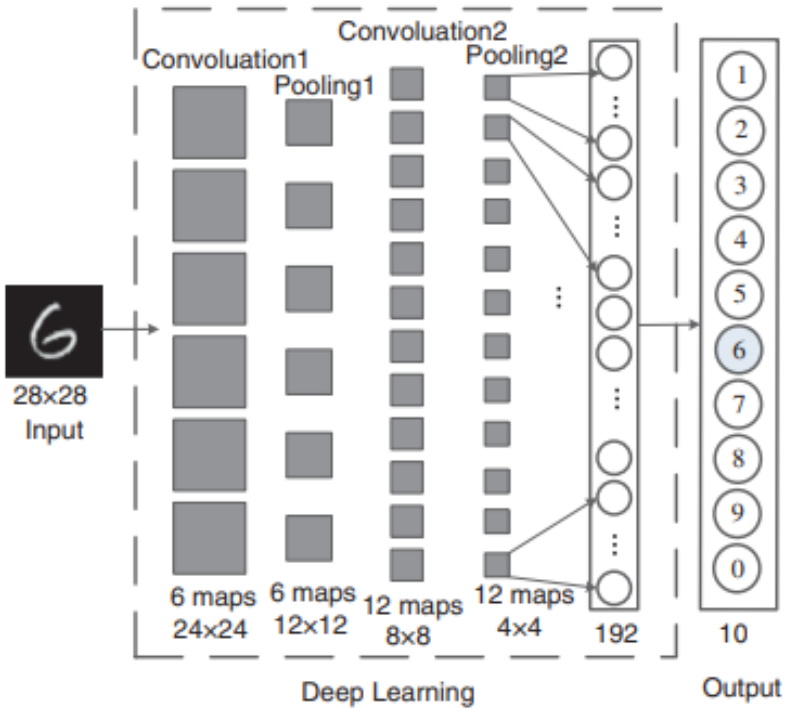
Gambar 8.2 Hasil TensorFlow berdasarkan pemrograman jaringan saraf tiruan.

Pengenalan Angka menggunakan Jaringan Saraf Konvolusi

Lapisan antara mengekstrak fitur dan mengklasifikasikan dengan CNN, seperti yang ditunjukkan pada Gambar 8.3, yaitu memanfaatkan CNN untuk mewujudkan pengenalan angka tulisan tangan. Dataset MNIST serba guna dari angka tulisan tangan diadopsi di lapisan input. Dengan beberapa lapisan konvolusi, lapisan penyatuan dan lapisan yang terhubung penuh, hasil pengenalan dapat diperoleh di lapisan keluaran.

Struktur Convolutional Neural Network (CNN)

Struktur lapisan pembelajaran 5 dalam dalam jaringan saraf convolutional (CNN) ditunjukkan pada Gambar 8.3. Ada dua lapisan konvolusi dalam struktur ini. Setiap lapisan konvolusi diikuti oleh lapisan penyatuan. Lapisan terakhir adalah lapisan yang terhubung penuh dan klasifikasi keluaran akhirnya dilakukan. Grafik CNN ini dimodifikasi dari karya Rasmus Palm: http://github.com/rasmusbergpalm/DeepLearnToolbox



Gambar 8.3 Struktur pengenalan angka tulisan tangan dengan jaringan CNN.

Contoh 8.3 Penggunaan Convolutional Neural Network untuk Pengenalan Angka

Untuk tujuan yang sama seperti dalam dua contoh di atas, kami ingin mengenali angka tulisan tangan menggunakan dataset MNIST dan konstruksi CNN:

1) **Input:** Input x adalah angka tulisan tangan 28 × 28 piksel.
2) **Konvolusi 1:** Ukuran kernel konvolusi dari lapisan konvolusi pertama adalah 5 × 5 dengan 6 peta fitur, yaitu matriks bobot 1 × 6 wij diadopsi, di mana $w_{ij} \in R^{5x5}$. Ia melakukan computing untuk graf ciri dengan $y_j = f(\sum_i (w_{ij} * x_i + b_i))$, dimana f adalah fungsi sigmoid, yaitu $f(x) = \frac{1}{1+e^{-x}}$. $J = \{1,2,\ldots,6\}$ adalah singkatan dari serial number (SN) dari peta fitur keluaran dan i adalah singkatan dari SN dari peta fitur masukan. Ukurannya output feature map yang didapat adalah 24×24. Output feature graph pada layer ini akan menuju ke layer berikutnya, Pooling 1 sebagai input.
3) **Pooling 1:** Lapisan Pooling (subsampel) pertama mengadopsi operasi pooling maksimum dan memilih nilai maksimum di setiap area yang tidak tumpang tindih 2 × 2 sebagai output. 6 peta fitur keluaran pooling layer 1 akan masuk ke layer berikutnya Convolution 2 sebagai input.
4) **Konvolusi 2:** Ukuran kernel konvolusi di lapisan ini adalah 5 × 5 dan 12 kernel konvolusi diadopsi, yaitu 6 × 12 matriks bobot wij diadopsi, di mana $w_{ij} \in R^{5x5}$. Kami melakukan perhitungan untuk peta fitur dengan $y_j = f(\sum_i (w_{ij} * x_i + b_i))$, di mana f adalah fungsi sigmoid, j = {1,2,…,12} adalah SN dari peta fitur keluaran, dan i = {1,2,…,6} adalah SN peta

fitur masukan. Dua belas grafik fitur keluaran dari 8 8 diperoleh pada lapisan ini sebagai keluaran akan berfungsi sebagai masukan peta fitur Pooling 2.

5) **Pooling 2:** Gunakan operasi pooling maksimum untuk memilih nilai maksimum di setiap area non-overlapping 2 × 2 sebagai output. Ada 12 peta fitur 4 × 4 secara total.
6) **Koneksi penuh:** Setiap peta fitur keluaran dari lapisan penyatuan akan dibuka menjadi vektor 1 × 16, dan 12 grafik fitur terhubung ke dalam vektor 1 × 192.
7) **Output:** Ini mengadopsi pengklasifikasi softmax untuk klasifikasi dan mengeluarkan hasil klasifikasi. Hasil dari Gambar 8.3 adalah {6}.

Implementasi sistem pengenalan angka tulisan tangan dengan jaringan CNN mencakup empat langkah rinci, seperti yang dijelaskan di bawah ini:

Langkah 1: **Muat kumpulan data**

Pertama, kita perlu memuat dataset dan melakukan preprocess masing-masing train set train_x dan label set train_y. Demikian pula, kita juga perlu melakukan preprocess test dataset test_x dan label set test_y. Kode semu dari tes ini diberikan di bawah ini:

```
load mnist_uint8;
train_x = double(reshape(train_x',28,28,50000))/255;
test_x = double(reshape(test_x',28,28,10000))/255;
train_y = double(train_y');
```

Langkah 2: **Inisialisasi di CNN**

Pada langkah ini, kita perlu menginisialisasi struktur convolution layer dan pooling layer; untuk lapisan konvolusi, jumlah kernel konvolusi (peta keluaran) dan ukuran kernel konvolusi (layerCkernel). Untuk pooling layer, ukuran pooling area (layerScale) harus ditentukan. Pseudo-code diberikan di bawah ini:

```
layerNumber=5
layer(1, 'i') // Input layer
layer(2, 'c', 6, 5) // Convolution layer, 6 5x5 kernel,
                      layerCkernel=5, layerName='c', outputmaps=6
layer (3, 'p', 2) // Pooling layer, 2x2 Pooling,
                    layerScale=2, layerName='p'
layer (4, 'c', 12, 5) // Convolution layer, 12 5x5 kernel,
                    layerCkernel=5, layerName='c', outputpmaps=12
layer (5, 'p',2) //Pooling layer, 2 x 2 Pooling,
                  layerScale=2, layerName='p'
```

Untuk setiap lapisan penyatuan, ukuran peta fitur (ukuran peta) harus diinisialisasi. Untuk setiap lapisan konvolusi, kita perlu mengkonfigurasi beberapa parameter, seperti bobot koneksi (w), bias (b), dan ukuran peta fitur (ukuran peta). Setelah inisialisasi selesai pada Layer l, peta fitur keluarannya akan digunakan sebagai peta fitur input Layer ±1. Kode semu ditunjukkan di bawah ini:

```
for l = 1 to layerNumber
          if layer(l).layerName= 'p' // Pooling layer
                    layer(l).mapsize = layer(l).mapsize /
                    layer(l).layerScale
          endif
          if layer(l).layerName='c' // Convolution layer
                    layer(l).mapsize = layer(l).mapsize -
                    layer(l).layerCkernel+ 1;
                    for j = 1 to layer(l).outputmaps % output map
                              for i = 1 to layer(l).mapsize %
                              input map
                                        layer(l).w(j) =rand(i,i)
                                        layer(l).b(j) = 0;
                              endfor
                    endfor
          endif
          layer(l+1).inputmaps= layer(l).outputmaps;
          layer(l+1).mapsize= layer(l).mapsize;
endfor
```

Langkah 3: Pelatihan CNN

Selama tahap pelatihan, pertama-tama kita memasukkan gambar angka tulisan tangan. Kelas keluaran dari citra masukan akan diperoleh setelah algoritma perambatan maju melewati semua lapisan di CNN. Kemudian, algoritma perambatan mundur digunakan untuk menghitung kesalahan antara kelas output dan kelas label, dengan metode layer-wise mulai dari lapisan output ke lapisan input. Terakhir, kami menyesuaikan parameter setiap lapisan, untuk mengurangi kesalahan. Setelah pelatihan selesai, parameter jaringan diperbaiki untuk mendapatkan CNN yang terlatih dengan baik. Langkah ini mencakup fungsi cnnff(), fungsi cnnbp() dan updatepara(), dan pseudo-code ditampilkan seperti di bawah ini:

1) **CNN forward propagating:** Fungsi cnnff() digunakan untuk mengembalikan hasil pengenalan angka. Data citra masukan terlebih dahulu masuk ke Layer 1, dan melewati Layer 2 untuk mencapai Layer terakhir (dilambangkan dengan nomor layer). Akhirnya, hasil klasifikasi, y', akan diperoleh pada lapisan keluaran. Jika lapisan saat ini adalah lapisan konvolusi, semua peta fitur input akan digunakan untuk operasi konvolusi untuk mendapatkan peta fitur output. Jika layer saat ini adalah layer pooling, operasi max-Pooling akan dilakukan untuk semua peta fitur input:

```
layer (1) =x; // x is sample data set
inputmaps = 1;
for l = 2 to layerNumber
          if layerName= 'c'
                    for i=1 to outputmaps
                              initialize every outputmaps
                              Calculate convolution of inputmap
                              get outputmaps
                    endfor
          endif
          if layersname= 's'
                    for j = 1 to inputmaps
                              Figure out maximum values in layer
                              Scale area
                    endfor
          endif
                    layer(l+1). inputmaps= layer(l+1).outputmaps
endfor
Calculate output for full connection layer
Multiple classifier outputs of classification y'
```

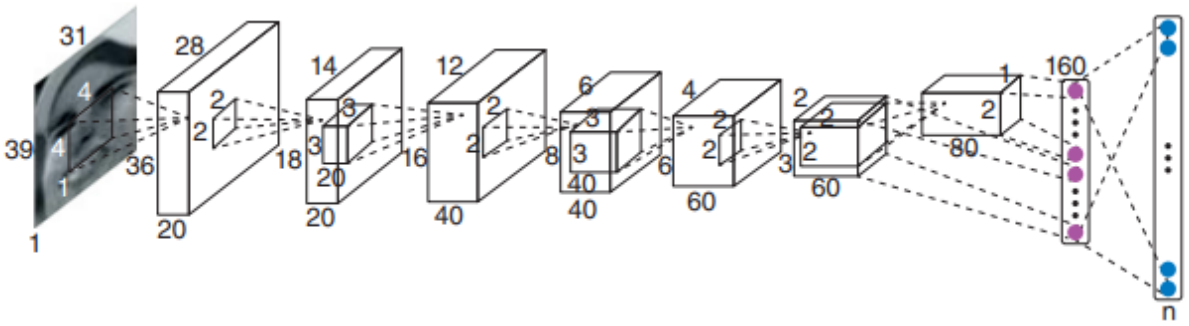
2) **Propagasi mundur CNN:** Berdasarkan output perambatan maju (y') dan data label (y), kesalahan dan fungsi biaya akan dihitung. Dari lapisan terakhir (layerNumber+1) sampai ke lapisan pertama, kesalahan disebarkan ke belakang. Kode semu dari fungsi cnnbp() diberikan di bawah ini:

```
error = y'- y; // Recognition error
L = 1/2*sum(error^ 2) / size(error); // Loss function
// back propagating error
for i= layerNumber+1 to 1 step -1 // layerNumber+1 is the
                                     number of layers
 Calculate error of layer of No i
endfor
```

3) **Perbarui bobot:** Operasi ini diterapkan dari lapisan input ke lapisan yang terhubung penuh. Jika lapisan saat ini bukan lapisan Pooling, bobot dan bias antara Lapisan l dan Lapisan l-1 akan diperbarui. Kode semu dari fungsi updatepara() untuk pembaruan parameter CNN diberikan di bawah ini:

```
for i= 2 to layerNumber+1 // layerNumber+1 is total
 number of layers
          // update weights of each layer except the
             Pooling layer.
          if layerName != 'p'
                    Update the connection weights between
                     the i-th layer and (i-1)-th layer
                    Update bias of i-th layer
         endif
endfor
```

4) **Uji CNN** Pengujian akhir membutuhkan input test_x dan menggunakan cnnff() untuk mengeluarkan hasil pengenalan. Tag pengujian set test_y digunakan untuk menguji keakuratan proses pengenalan.



Gambar 8.4 Struktur sistem pengenalan wajah manusia Deep ID dengan CNN dengan 10 lapisan termasuk lapisan I/O.

Jaringan Saraf Konvolusi untuk Pengenalan Wajah

Pengenalan wajah manusia adalah arah penelitian yang sangat penting di bidang visi komputer. Saat ini, di bidang ini, *deep learning* telah mencapai atau melampaui level manusia. Misalnya, tingkat pengenalan untuk kumpulan data LFW adalah 99,47%, yang lebih tinggi dari nilai 99,25% pada kumpulan data ini oleh mata manusia. Dalam dataset LFW, terdapat 13.233 gambar dari 5749 orang, dan gambar tersebut diperoleh dari Yahoo News. Hanya ada satu gambar untuk 4069 orang, dan ada beberapa gambar untuk 1680 orang.

Gambar 8.4 menunjukkan struktur CNN yang diadopsi dalam algoritma Deep ID, diusulkan oleh The Chinese University of Hong Kong pada tahun 2014, yang tujuannya adalah pengenalan wajah manusia. Algoritma Deep ID mencakup empat lapisan konvolusi, tiga lapisan penyatuan dan satu lapisan koneksi penuh. Tingkat pengenalan wajah manusia dalam dataset LFW dengan CNN semacam itu mencapai 97,45%.

Sekarang, kita tentukan algoritma Deep ID sebagai berikut:

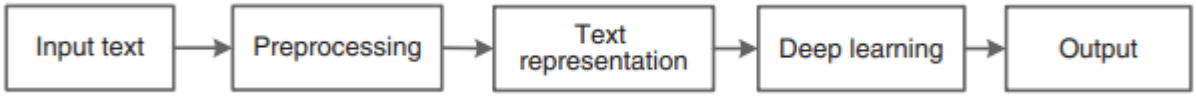
- **Input:** lapisan input adalah gambar wajah manusia berukuran 39 × 31 piksel, masing-masing dalam kumpulan data LFW.
- **Convolution 1:** Pada lapisan konvolusi 1, algoritme menetapkan ukuran kernel konvolusi sebagai 4 × 4, dan menggunakan 20 kernel konvolusi. Jadi, pada lapisan ini, 20 matriks bobot 4 × 4 perlu diatur. Selanjutnya, langkah konvolusi diatur ke 1. Setelah konvolusi, kita dapat memperoleh 20 grafik fitur, dan ukuran masing-masing grafik fitur adalah ((39 – 4) + 1) × ((28 – 4) + 1) = 36 × 28. Untuk setiap peta fitur x, fungsi RELU(x) = max(0,x) digunakan sebagai fungsi aktivasi.
- **Pooling 1:** Pada pooling layer 1, algoritma melakukan pooling maksimum untuk setiap area non-overlapping pooling 2 × 2 untuk 20 output grafik fitur oleh konvolusi layer

sebelumnya 1. Setelah pooling, kita juga dapat memperoleh 20 grafik fitur, dan sekarang ukurannya masing-masing graf ciri adalah (36/2) × (28/2) = 18 × 14.

- **Convolution 2:** Pada convolution layer 2, algoritma menggunakan output dari layer pooling 1 sebelumnya sebagai input. Ukuran kernel konvolusi adalah 3 × 3, dan 40 kernel konvolusi (40 matriks bobot 3 × 3) diadopsi. Mirip dengan konvolusi 1, langkah konvolusi juga diatur ke 1. Dengan demikian, 40 grafik fitur akan diperoleh setelah konvolusi, dan ukuran setiap grafik fitur adalah ((18– 3) + 1) × ((14 – 3) + 1) = 16 × 12. Kami juga menggunakan fungsi RELU sebagai fungsi aktivasi.
- **Pooling 2:** Pada pooling layer 2, algoritma melakukan operasi untuk 40 output grafik fitur oleh konvolusi layer 2 sebelumnya, dan melakukan pooling maksimum untuk setiap area yang tidak tumpang tindih 2 × 2. Kemudian akan diperoleh 40 grafik fitur, dan ukuran masing-masing grafik fitur adalah (16/2) × (12/2) = 8 × 6.
- **Convolution 3:** Pada convolution layer 3, algoritma menggunakan output dari layer pooling 2 sebelumnya sebagai input. Ukuran kernel konvolusi adalah 3 × 3, dan 60 kernel konvolusi (60 matriks bobot 3 × 3 akan diatur) diadopsi. Mirip dengan konvolusi lapisan 1 dan 2, langkah konvolusi diatur ke 1. Enam puluh grafik fitur akan diperoleh setelah konvolusi, dan ukuran masing-masing grafik fitur adalah ((8 – 3) + 1) × ((6 – 3) + 1) = 6 × 4. Fungsi aktivasi adalah fungsi RELU.
- **Pooling 3:** Pada pooling layer 3, algoritma melakukan operasi untuk 60 output grafik fitur oleh konvolusi layer 3 sebelumnya, dan melakukan pooling maksimum untuk setiap area yang tidak tumpang tindih sebesar 2 × 2. Kemudian akan diperoleh 60 grafik fitur, dan ukuran setiap grafik fitur adalah (6/2) × (4/2) = 3 × 2.
- **Convolution 4:** Pada konvolusi layer 4, algoritma menggunakan output dari pooling layer 3 sebelumnya sebagai input. Ukuran kernel konvolusi adalah 2 × 2, dan 80 kernel konvolusi (80 matriks bobot 2 × 2) diadopsi. Langkah konvolusi diatur ke 1. Delapan puluh grafik fitur akan diperoleh setelah konvolusi, dan ukuran setiap grafik fitur adalah ((3 – 2) + 1) × ((2 – 2) + 1) = 2 × 1. fungsi aktivasi adalah fungsi RELU.
- **Deep ID:** Dalam algoritma ini, Deep ID adalah layer koneksi penuh dan berisi 160 neuron tersembunyi yang melakukan koneksi penuh ke convolution 4 dan pooling 3. Softmax (lapisan keluaran): menggunakan pengklasifikasi softmax dan mengeluarkan hasil pengenalan.

Analisis Teks Medis oleh Convolutional Neural Networks

Untuk melakukan pemahaman teks dengan metode *deep learning*, pertama-tama kita perlu melakukan ekspresi digital untuk teks, kemudian menggunakan algoritma *deep learning* untuk ekstraksi fitur dan pemahaman teks. Penerapan pendekatan ini diilustrasikan di bawah ini untuk analisis teks medis.



Gambar 8.5 Proses klasifikasi teks dengan *deep learning*.

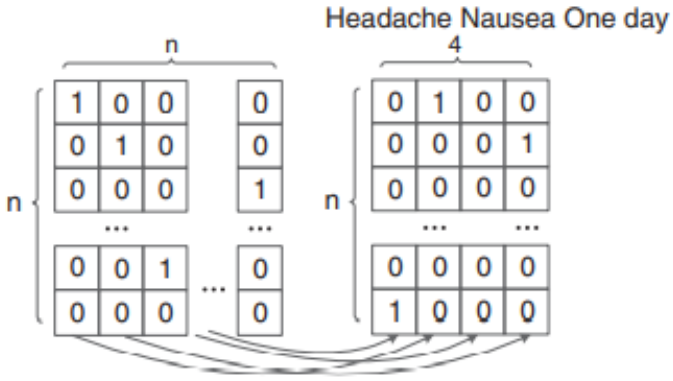
Penyematan Kata melalui Deep learning

Gambar 8.5 menunjukkan langkah-langkah pemahaman teks dengan *deep learning*. Metode serupa dapat diadopsi dalam pemahaman teks medis juga. Di bagian ini, kami akan memperkenalkan kursus terperinci di mana kami mengekstrak fitur representasi teks medis dan kemudian mengadopsinya untuk memahami teks medis dengan metode *Deep learning*. Kami menggunakan penilaian risiko penyakit dengan teks medis sebagai contoh.

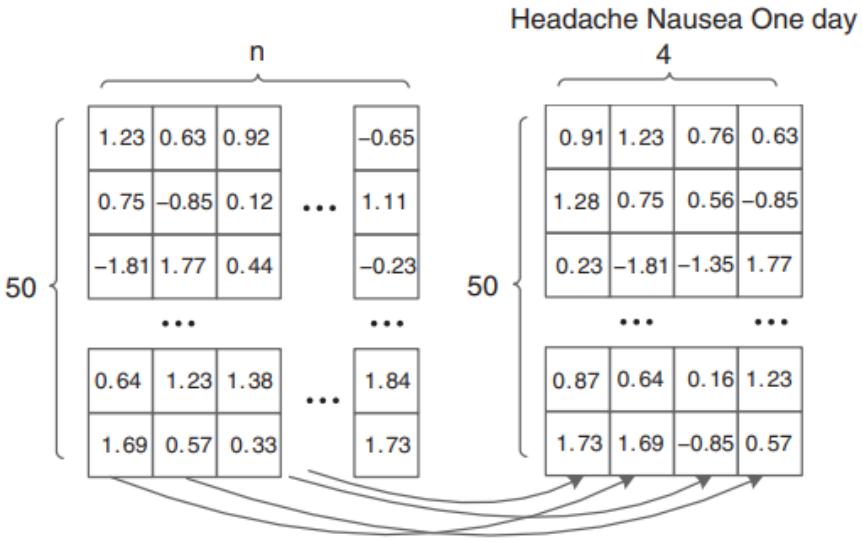
Sebelum penggunaan *Deep learning* untuk pemahaman bahasa alami, setiap kata harus ditransformasikan dari teks ke representasi digital terlebih dahulu. Secara umum, kami mengadopsi metode penyisipan kata untuk representasi. Melakukan representasi kata dengan penyisipan kata adalah membentuk kosakata di mana satu kata berkorespondensi dengan satu vektor. Ada dua metode representasi word embedding: representasi one-hot dan representasi terdistribusi. Representasi one-hot lebih sederhana dan lebih langsung. Kuantitas total kata dalam kosakata juga merupakan dimensi vektor.

Tetapi hanya ada satu nilai yang sama dengan 1 dalam komposisi vektor setiap kata, sedangkan nilai lainnya sama dengan 0. Dalam metode ini, nilai unik diadopsi untuk mengidentifikasi setiap kata, jadi ini adalah metode sparse. Ketika penyisipan kata ditetapkan, hubungan semantik antara dua kata yang berbeda tidak dipertimbangkan. Tidak ada hubungan antara vektor, bahkan untuk kata-kata dengan semanteme yang sama, yang disebut "celah kata". Karena dimensi penyisipan kata sama dengan jumlah total kata, beban computing untuk aplikasi dalam beberapa tugas terlalu tinggi dan dapat menyebabkan bencana dimensi.

Gambar 8.6 menunjukkan representasi teks dengan metode representasi one-hot. Dengan metode Representasi Terdistribusi, setiap kata diwakili oleh vektor bilangan real, seperti [0.792, 0.177, 0.107, 0.109, 0.542, …]. Dengan demikian, dimensi vektor jauh lebih kecil daripada jumlah total kata. Jika Word Embedding dibuat dengan metode Representasi Terdistribusi, sejumlah besar korpus teks nyata akan dibutuhkan untuk pelatihan dan pembelajaran. Alat Word2vec sering digunakan untuk melatih penyisipan kata. Dimensi vektor akan ditentukan dalam proses pembelajaran word embedding. Misalnya, dimensi dapat diatur menjadi 50. Dengan metode ini, penyisipan kata memperkenalkan semanteme kata, yaitu kedekatan antar kata membuat vektor kata mereka lebih dekat dalam ruang vektor. Gambar 8.7 menunjukkan representasi teks dengan metode representasi terdistribusi. Dibandingkan dengan representasi one-hot, dimensi penyisipan kata sebagian besar berkurang dan jarak vektor antara semanteme yang relevan atau semanteme serupa dekat.



Gambar 8.6 Representasi one-hot untuk penyisipan kata



Gambar 8.7 Representasi Terdistribusi untuk penyisipan kata

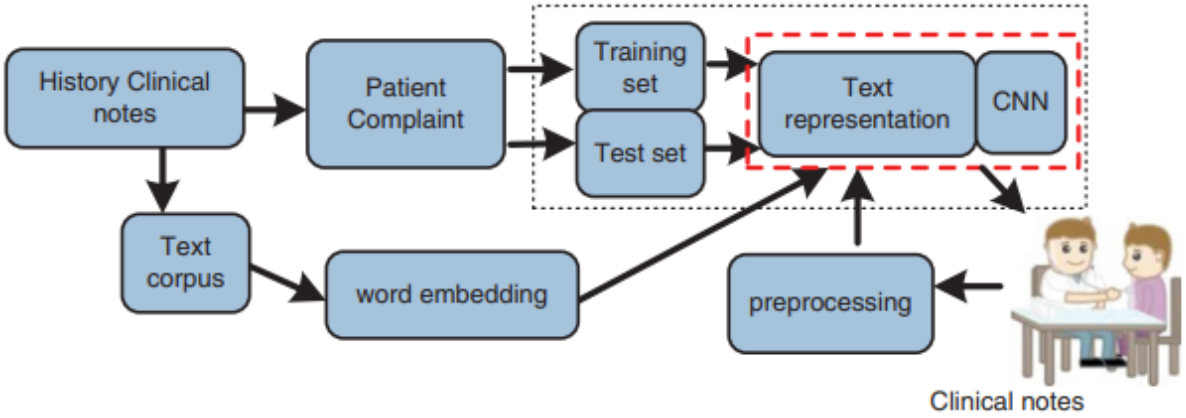
Word Embedding adalah matriks D = Rd× C , di mana d adalah dimensi untuk word embedding dan |C| adalah jumlah kata dalam kosakata. Lokasi kata dalam kosakata disimpan dalam C. Hanya ada satu lokasi yang sama dengan 1 untuk vektor yang sesuai dengan setiap kata. Representasi vektor untuk kata i dalam kosakata disimpan di kolom ke-i, yaitu matriks vektor Word Embedding. Saat mengubah teks menjadi kata-kata digital, ia mengekstrak representasi vektor t untuk setiap kata c dari matriks vektor Word Embedding.

Setiap sampel teks input x(x_1, x_2, …, x_N) menyertakan N kata. Kami mencari representasi vektor xwn yang sesuai dengan setiap kata x_n dalam teks dari penyisipan kata dengan t = D · C. Representasi penyisipan kata xw(xw_1, xw_2, …, xw_N) akan diperoleh untuk sampel input.

Analisis Teks Medis menggunakan CNN

Gambar 8.8 menunjukkan model pemahaman teks medis berdasarkan jaringan saraf convolutional. Model ini terutama mencakup tiga bagian:

1) **Setup Word Embedding:** mengekstrak data historis pasien dalam catatan klinis, melakukan pembersihan data dan pra-pemrosesan data, dan melatih penyisipan kata dengan data yang telah diproses sebagai korpus. Algoritma N-skip gram atau algoritma lain dalam word2vec dapat diadopsi untuk melatih penyisipan kata. Dimensi diatur untuk penyisipan kata.
2) **Melatih CNN untuk mempelajari fitur teks medis:** memilih data penyakit dari data dalam catatan klinis; setelah data cleaning dan data preprocessing pilih "Patient Complaint", "Diagnosis Record through Interrogation", dll. Dan data tersebut menjadi sample data. Kemudian melakukan representasi digital untuk data sampel dengan penyisipan kata, dan memasukkan hasilnya ke CNN untuk pembelajaran terawasi fitur dalam penilaian risiko penyakit.
3) **Uji dan aplikasi:** proses pengujian dan aplikasi adalah sama. Ini memasukkan "Keluhan Pasien" dan data teks yang relevan dengan penyakit, melakukan pra-pemrosesan dan representasi teks untuk data, memasukkan hasil ke CNN dan mengeluarkan hasil penilaian risiko penyakit.

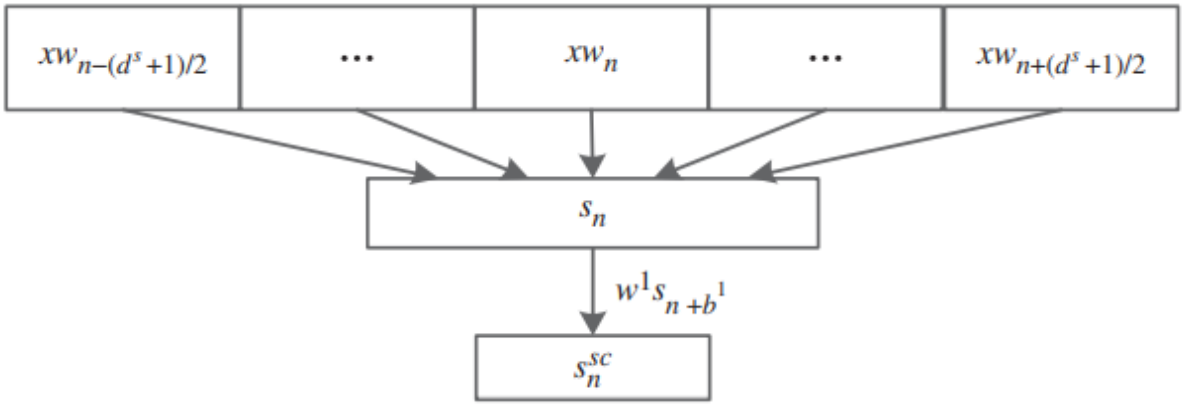


Gambar 8.8 Model penilaian risiko penyakit melalui pembelajaran teks kedokteran menggunakan jaringan CNN.

Konstruksi Jaringan Saraf Konvolusi

Untuk representasi penyisipan kata xw(xw_1, xw_2, …, xw_N) teks input, menghitung vektor konvolusi s_n^{sc} untuk setiap kata dalam xw secara berurutan. Perhitungan vektor konvolusi untuk kata n diberikan pada Gambar 8.9. Ukuran jendela konvolusi adalah d_s. Dengan kata saat ini n di tengah, memotong kata ds dari teks input dan menghubungkan vektor dari kata-kata ini, kita memperoleh $S_n \in R^{dds}$ dengan $S_n = (xw_{n-(ds-1)/2}, \dots, xw_n, \dots xw_{n+(ds\ 1)/2})^T$. Sistem menghitung vektor konvolusi ssc untuk kata n menggunakan persamaan berikut, di mana $w^1 \in R^{lsc \times dds}$ adalah matriks bobot dan b^1 adalah simpangannya:

$$s_n^{sc} = w^1 s_n + b^1 \qquad (8.1)$$



Gambar 8.9 Konvolusi untuk kata xwn (kata No. n).

Variabel s_n^{sc} adalah singkatan dari h_n^1 , ekspresi dari kata xw_n di lapisan tersembunyi. Setelah 4 memperoleh h_n^1, kami menghitung output h_n^2 dari lapisan tersembunyi dengan fungsi tanh $(\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}})$ dan menjadikannya input dari lapisan tersembunyi berikutnya:

$$h_n^2 = \tanh(w^1 s_n + b^1) \tag{8.2}$$

1) **Pooling layer:** Output dari convolution layer digunakan sebagai input dari pooling layer. Nilai maksimum antara N elemen dalam h_n^2 dihitung dengan

$$h^3 = max_{1 \le n \le N} h_n^2 \tag{8.3}$$

Operasi pooling dibagi menjadi operasi pooling maksimum dan operasi pooling rata-rata. Di sini kita melakukan operasi penyatuan maksimum karena fungsi setiap kata dalam teks tidak sepenuhnya sama, yaitu elemen-elemen yang memainkan peran penting dalam teks akan dipilih oleh operasi penyatuan maksimum. Sampel memiliki panjang yang berbeda, begitu juga input $x(x_1, x_2, \ldots, x_N)$. Namun, teks akan diubah menjadi vektor dengan panjang tetap setelah melewati convolution layer dan pooling layer:

2) **Lapisan keluaran:** Ada lapisan koneksi penuh jaringan saraf setelah lapisan penyatuan. Pengklasifikasi Softmax diadopsi untuk menghasilkan hasil klasifikasi:

$$h^4 = w^4 h^3 + b^4 \rightarrow y_i = \frac{e^{h_i^4}}{\sum_{m=1}^{n} e^{h_m^4}} \tag{8.4}$$

di mana n menunjukkan ada n kelas. Kami mendefinisikan semua parameter yang memerlukan pelatihan sebagai set parameter $\theta = \{w^1, w^4, b^1, b^4\}$

Tujuan pelatihan adalah untuk mempelajari nilai kemungkinan log maksimum dari . Kami menggunakan bilangan acak sebagai parameter inisialisasi θ dan penurunan gradien stokastik untuk pelatihan parameter θ. Untuk revisi parameter, kami menggunakan ekspresi berikut, D adalah kumpulan sampel pelatihan, $class_y$ adalah klasifikasi sampel yang akurat dan α adalah kecepatan pembelajaran:

$$\max_{\theta} \sum_{y \in D} \log p\left(class_y | D, \theta\right) \rightarrow \theta = \theta + \alpha \frac{\partial \log p\left(class_y | D, \theta\right)}{\partial \theta} \quad (8.5)$$

Pemahaman Teks Medis dengan Jaringan CNN

Kode semu untuk model penilaian risiko penyakit dalam pemahaman teks medis dengan jaringan CNN ditentukan dalam Algoritma 8.1. Data teks medis termasuk set pelatihanX dan set tes X', dengan data hasil diagnosis yang sesuai dari dokter dibagi menjadi set label pelatihan Y dan set label uji Y'.

Pertama, kita membaca satu data teks medis x dari X, kemudian merepresentasikannya sebagai vektor, akhirnya mendapatkan hasil prediksi y setelah lapisan konvolusi, lapisan penyatuan dan lapisan koneksi penuh. Pembaruan parameter θ di CNN ini diselesaikan melalui algoritma propagasi mundur. CNN setelah pelatihan dapat digunakan untuk penilaian risiko penyakit menurut teks medis. Dengan cara yang sama, kami menggunakan set uji X' dan set label yang sesuai Y' untuk menguji model. Kami mendapatkan hasil prediksi y dan membandingkannya dengan label sebenarnya di Y' untuk memperkirakan kinerja CNN.

Algoritma 8.1 Pembuatan CNN untuk Pengenalan Teks

Input: X: Sampel pelatihan, data input asli dalam catatan klinis

Y: Tag untuk sampel pelatihan, hasil diagnosis penyakit pasien yang sesuai dalam catatan klinis

Output: Buat CNN, parameter jaringan θ = {w^1, w^4, b^1, b^4}, hasil tes

Algoritma:

1) Inisialisasi: c = 5 //ukuran kernel konvolusi;
   T = 50 // ukuran batch sampel
2) untuk I = 1,2, …, I // (I adalah jumlah iterasi)
3) untuk j = 1,2, …, m // (m adalah jumlah batch sampel)
4) Baca dalam sampel batch x dan tag yang sesuai y
5) Representasi vektor untuk data x (untuk setiap kata, cari representasi vektornya dalam penyisipan kata).
6) untuk n = 1,2, …,T
7) Hitung jumlah kata (n) dalam sampel
8) Hitung konvolusi
9) Max-Pooling untuk n kata
10) Hubungkan lapisan koneksi penuh dengan pengklasifikasi softmax untuk klasifikasi, dapatkan hasil y_n^* .
11) Selesai
12) Gunakan algoritma perambatan mundur untuk memperbarui dengan Formula 10.5
13) Selesai

8.3 DEEPMIND DENGAN PEMBELAJARAN PENGUATAN MENDALAM

Di bagian ini, kami mempelajari teknologi DeepMind yang saat ini digunakan oleh program kecerdasan buatan Google. Skema Deep Reinforcement Learning disajikan bersama dengan aplikasinya di AlphaGo dan program game lainnya di Google cloud.

Program AI Google DeepMind

Pada tahun 2010, sebuah perusahaan kecerdasan buatan Inggris memulai DeepMind Technologies, yang telah menerima Penghargaan "Perusahaan Tahun Ini" oleh laboratorium Komputer Cambridge di Inggris. Selanjutnya, pada tahun 2014, DeepMind bergabung dengan Google dengan harga £500 juta. Proyek ini menerapkan jaringan dalam konvolusi yang belajar bermain video game dengan cara yang meniru memori jangka pendek otak manusia. Go adalah permainan yang sangat kompleks untuk pemain manusia dan komputer, karena ruang pencarian yang sangat besar yang terlibat.

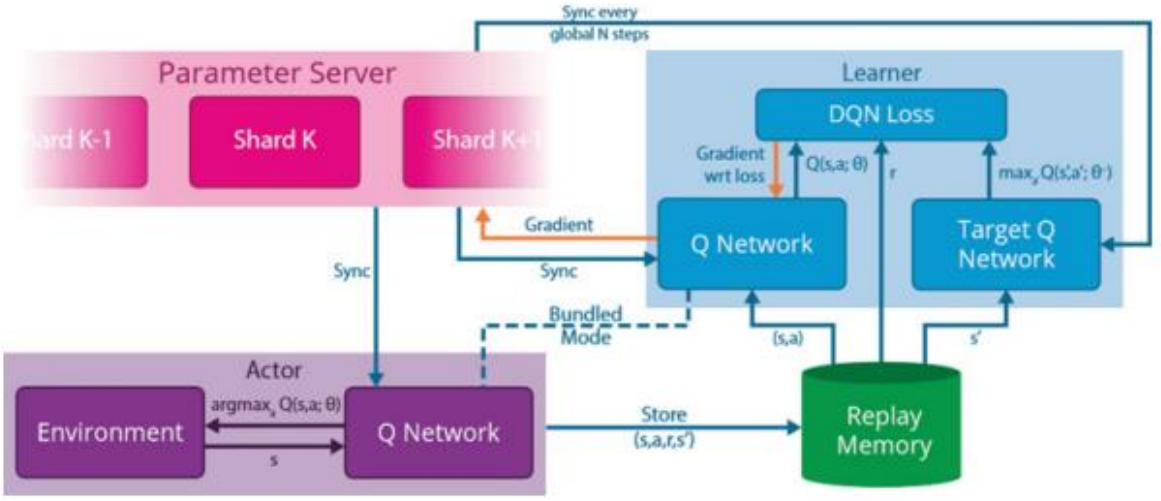
Pada tahun 1997, komputer IBM Deep Blue mengalahkan juara catur dunia Garry Kasparov dalam pertandingan terbuka. Sejak saat itu, program AI terkuat untuk memainkan Go hanya mencapai level amatir 5-dan, yang masih belum bisa mengalahkan pemain Go profesional tanpa cacat. Misalnya, program perangkat lunak Zen, yang dijalankan pada empat klaster PC, mengalahkan Masaki Takemiya (9p) dua kali dengan handicap 5 dan 4 batu. Program Crazy Stone mengalahkan Yoshio Ishida (9p) dengan handicap 4 batu.

Permainan Go dimainkan dengan batu hitam dan putih di papan mesh 19 × 19. Gim ini memiliki kompleksitas pohon pencarian yang sama dengan bd, di mana b adalah luasnya gim (jumlah gerakan ilegal di setiap negara bagian) dan d adalah kedalaman (jumlah gerakan sebelum permainan berakhir). Ini berarti bahwa pencarian brute force tidak mungkin dilakukan oleh komputer untuk mengevaluasi siapa yang menang. Di masa lalu, tidak ada komputer yang pernah mengalahkan pemain Go manusia hingga Maret 2016. Faktanya, Go jauh lebih kompleks daripada game lain seperti catur. Ini dikaitkan dengan jumlah kemungkinan yang jauh lebih besar di papan permainan Go. Kompleksitasnya melibatkan langkah-langkah dalam yang bahkan pemain profesional tidak dapat melacak di luar langkah-langkah tertentu dengan kemungkinan imbalan yang dievaluasi secara akurat.

Proyek penelitian AlphaGo dibentuk sekitar tahun 2014 untuk menguji seberapa baik jaringan saraf menggunakan *Deep learning* dapat memenangkan pemain profesional Go. Ini menunjukkan peningkatan yang signifikan dibandingkan program Go sebelumnya. AlphaGo yang berjalan di banyak komputer memenangkan 500 game yang dimainkan melcloud program Go lainnya. Sistem terdistribusi yang digunakan pada pertandingan Oktober 2015 ini menggunakan 1202 CPU dan 176 GPU. Pada Januari 2016, tim menerbitkan makalah di jurnal Nature yang menjelaskan algoritma yang digunakan di AlphaGo. Pada bulan Maret 2016, program komputer ini mengalahkan Lee Sedol, pemain 9-dan Go di dunia dengan skor 4 banding 1 dalam 5 pertandingan.

AlphaGo tidak secara khusus dilatih untuk menghadapi Lee dan memenangkan permainan sepenuhnya dari Smart Machine tanpa cacat. Meskipun kalah dari Lee di game keempat, dan Lee mengundurkan diri di game terakhir, memberikan skor akhir 4 game untuk 1 mendukung AlphaGo. Sebagai pengakuan atas mengalahkan Lee, AlphaGo dianugerahi kehormatan 9-dan oleh Asosiasi Baduk Korea. Program Google DeepMind ditujukan untuk memecahkan masalah kecerdasan yang sangat sulit yang memanfaatkan *Machine learning* dan sistem ilmu saraf.

Pertandingan AlphaGo dan Lee membuktikan bahwa komputer dapat dilatih untuk memformalkan proses kecerdasan manusia. Selain pertandingan Go, tujuh video game Atari: Pong, Breakout, Space Invaders, Seaquest, Beamrider, Enduro, dan Q bert juga diuji menggunakan program komputer serupa. Semua permainan ini melibatkan pemikiran strategis dari konten informasi yang tidak sempurna atau tidak pasti. DeepMind mengklaim bahwa program AI mereka tidak diprogram sebelumnya. Setiap gerakan dibatasi hingga 2 detik. Program belajar dari pengalaman hanya menggunakan piksel mentah sebagai input data. Secara teknis, program ini menggunakan *Deep learning* pada jaringan saraf convolutional.



Gambar 8.10 Arsitektur Gorila untuk menerapkan sistem pembelajaran penguatan Google (dicetak ulang dengan izin dari David Silver, Google DeepMind, http://www0.cs.ucl.ac.uk/staff/d.silver/web/Resources_files/).

Tim DeepMind telah mengusulkan skema baru Q-learning, berdasarkan pembelajaran penguatan. Gambar 8.10 menunjukkan diagram sistem skema dari Arsitektur Pembelajaran Penguatan Google, yang dikenal sebagai Gorila. Sistem ini diimplementasikan pada sekelompok besar server di Google. Dengan 64 utas pencarian, cluster terdistribusi dari 1930 CPU dan 280 GPU digunakan dalam kompetisi AlphaGo dan Lee. Akting paralel menghasilkan interaksi baru dengan memori replay terdistribusi untuk menghemat iterasi. Pembelajaran paralel menghitung gradien dari iterasi yang diulang. CNN terdistribusi memperbarui jaringan dengan gradien.

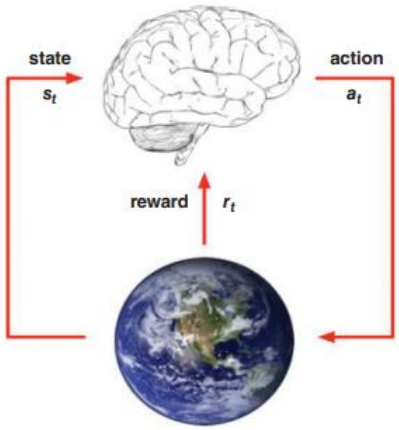
Google DeepMind telah menggabungkan *Deep learning* dan algoritma penguatan untuk mencapai kinerja tingkat manusia dalam beberapa aplikasi AI yang inovatif. Algoritma baru ini disebut deep reinforcement learning (DRL). DRL mengadopsi sekelompok agen untuk memilih tindakan terbaik. Pendekatan DRL pertama dikenal sebagai Deep Q-network (DQN), diusulkan oleh David Silver dari DeepMind. Dia juga salah satu penulis AlphaGo. DQN menggabungkan algoritma CNN dan Q-network. Jaringan Q digunakan untuk menilai hadiah setelah agen melakukan tindakan tertentu. Fungsi kotak lain pada Gambar 8.10 akan menjadi lebih transparan di Bagian 8.5.2, di mana algoritma DRL disajikan.

Program kecerdasan belajar memainkan permainan secara langsung setelah jumlah permainan pembelajaran yang cukup. Untuk sebagian besar game, DeepMind bermain jauh di bawah rekor dunia saat ini. Misalnya, aplikasi program DeepMind untuk video game 3-D, seperti Doom, masih dalam pengembangan pada tahun 2016. Menurut salah satu pendiri DeepMind, Mustafa Suleyman, teknologi DeepMind juga memperluas aplikasinya ke program Deep-Mind Health. Program ini terutama dirancang untuk memberikan layanan klinis kepada komunitas perawatan kesehatan. Ini akan membuka layanan kesehatan cerdas, yang menguntungkan semua pasien. Berikut ini, kami memperkenalkan pendekatan DeepMind dengan menggabungkan *Deep learning* dengan ide-ide pembelajaran penguatan. Kemudian kami memeriksa algoritma yang digunakan di AlphaGo dan di game Floppybird, termasuk implementasi dan proses pembelajaran yang diterapkan.

Algoritma Pembelajaran Penguatan Dalam

Seperti yang diperkenalkan di Bagian 8.1.2, proses DRL terutama ditampilkan oleh interaksi antara agen pembelajaran dan lingkungan kerjanya. Pembelajaran penguatan menawarkan algoritma untuk memecahkan masalah pengambilan keputusan berurutan. Imbalan kumulatif dimaksimalkan oleh agen perangkat lunak setelah melakukan serangkaian tindakan di lingkungan kerja. Tanpa mengetahui aturan sebelumnya, agen mengamati keadaan lingkungan saat ini dan mencoba beberapa tindakan untuk meningkatkan proses pembelajaran yang mendalam. Hadiah adalah umpan balik kepada agen dengan menyesuaikan strategi tindakannya. Setelah banyak penyesuaian, algoritma penguatan memperoleh pengetahuan tentang tindakan optimal untuk mencapai hasil terbaik untuk situasi tertentu dalam lingkungan keputusan.

Gambar 8.11 menunjukkan interaksi agen dan lingkungannya selama proses pembelajaran. Pada setiap waktu t, agen menerima status st. dan menjalankan suatu tindakan di . Kemudian, menerima observasi ot. dan hadiah rt terkait dengan tindakan. Lingkungan biasanya diformulasikan sebagai proses keputusan Markov (MDP) untuk memungkinkan agen berinteraksi dengannya. Setelah menerima tindakan, lingkungan memancarkan status dan hadiah skalar. Tujuan pembelajaran penguatan adalah untuk mengumpulkan hadiah, sebanyak mungkin pada langkah-langkah berturut-turut.



Gambar 8.11 Interaksi agen dan lingkungan dalam *Deep learning* (dicetak ulang dengan izin dari presentasi David Silver di Konferensi Internasional tentang *Machine learning*, ICML 2016) [20].

- Pada setiap langkah t agen:
  - Menerima status s_t
  - Menerima hadiah skalar r_t
  - Menjalankan tindakan a_t
- Lingkungan:
  - Menerima tindakan a_t
  - Memancarkan status s_t
  - Memancarkan hadiah skalar r_t

Urutan pengamatan, tindakan dan penghargaan, $\{o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t\}$, membentuk pengalaman, sedangkan keadaan adalah fungsi dari pengalaman, yaitu:

$$s_t = f(o_1, r_1, a_1, \cdots, a_{t-1}, o_t, r_t) \tag{8.6}$$

Algoritme AlphaGo menggunakan pencarian pohon Monte Carlo (MCTS) untuk menemukan pergerakannya berdasarkan pengetahuan yang sebelumnya "dipelajari" oleh *Machine learning*. ANN pembelajaran yang mendalam digunakan oleh pelatihan ekstensif, baik dari permainan manusia maupun komputer. MCTS dipandu oleh "jaringan nilai" dan "jaringan kebijakan", keduanya diimplementasikan menggunakan teknologi jaringan saraf dalam. Deteksi fitur khusus game dalam jumlah terbatas diterapkan ke input pada tahap pra-pemrosesan, sebelum dikirim ke jaringan saraf.

Jaringan saraf sistem pada awalnya di-bootstrap dari keahlian gameplay manusia. AlphaGo awalnya dilatih untuk meniru permainan manusia dengan mencoba mencocokkan gerakan pemain ahli dari permainan sejarah yang direkam, menggunakan database sekitar 30 juta gerakan. Setelah mencapai tingkat kemahiran tertentu, ia dilatih lebih lanjut untuk siap memainkan sejumlah besar game melcloud instance lain dari dirinya sendiri. Hal ini dilakukan dengan pembelajaran penguatan untuk meningkatkan permainannya. Tujuannya adalah untuk

menghindari membuang-buang waktu lcloud. Program diprogram untuk mengundurkan diri jika penilaian probabilitas menangnya berada di bawah ambang batas yang diberikan. Untuk pertandingan AlphaGo-Lee 2016, ambang pengunduran diri ditetapkan sebesar 20%.

Dalam Konferensi Internasional ke-33 tentang *Machine learning* (ICML 2016), Silver et al. mempresentasikan detail aplikasi DeepMind dengan menggunakan pendekatan DRL. Secara khusus, fungsi nilai, kebijakan dan model diwakili oleh jaringan saraf yang dalam. Dalam pendekatan DRL, AI dicapai dengan pembelajaran penguatan dan *Deep learning* secara bersama-sama. Tugas tingkat manusia dapat diselesaikan oleh agen tunggal dengan pembelajaran penguatan untuk mencapai tujuan yang ditetapkan oleh mekanisme *Deep learning*. Setelah tindakan dipilih oleh agen, kebijakan dan fungsi nilai memainkan peran penting dalam kinerjanya:

1) Kebijakan: adalah fungsi perilaku yang memilih tindakan yang diberikan status. Ada dua kebijakan tipikal. Salah satunya adalah kebijakan deterministik yang pasti mengeksekusi beberapa tindakan a di bawah keadaan tertentu s, yaitu = (s). Yang lainnya adalah kebijakan stokastik, yang berarti ada kemungkinan untuk melakukan beberapa tindakan di bawah keadaan s, yaitu (a|s) = P[a|s].
2) Fungsi nilai: memprediksi imbalan di masa depan, dan mengevaluasi kemanjuran suatu tindakan atau keadaan. Misalnya, Qπ(s, a) adalah total imbalan yang diharapkan dari keadaan s dan tindakan a berdasarkan kebijakan . Ini menghitung nilai yang diharapkan dari akumulasi hadiah yang diperoleh di masa depan, yaitu t + 1, t + 2, t + 3…, dll. Namun, hadiah masa depan didiskontokan seiring berjalannya waktu. Faktor-diskon ∈ [0, 1] diterapkan untuk mengurangi penghargaan dalam keadaan masa depan. Tidak ada model yang sempurna untuk memprediksi apa yang sebenarnya akan terjadi di masa depan:

$$Q^{\pi}(a|s) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 t_{t+3} + \cdots |s, a] \quad (8.7)$$

Tujuannya adalah untuk mendapatkan nilai maksimum Qπ(s, a). Kebijakan optimal diperoleh dengan memaksimalkan nilai fungsi sebagai:

$$Q^*(s, a) = E[r_{t+1} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})|s, a] \quad (8.8)$$

Persamaan 8.8 dikaitkan dengan Bellman, yang menggunakan pemrograman dinamis untuk mencapai nilai optimal melalui beberapa iterasi. Berikan tindakan a di bawah keadaan s, hadiah r_{t+1} diperoleh pada keadaan s_{t+1}. Untuk mencapai nilai Q maksimum, keadaan s_{t+1} perlu optimal. Demikian pula, nilai Q dari keadaan s_{t+2} harus dioptimalkan untuk menjamin nilai Q yang optimal dari keadaan s_{t+2}, dll. Proses iteratif berlangsung sampai keadaan akhir. Ketika jumlah status dan tindakan kecil, tabel tindakan status dapat dibuat untuk merekam nilai Q yang optimal. Untuk keadaan tak hingga, diperlukan fungsi aproksimasi untuk merepresentasikan hubungan antara keadaan, tindakan, dan nilai. Jaringan saraf adalah pilihan terbaik untuk tujuan ini.

DQN menyediakan tiga solusi stabil untuk mengatasi masalah di atas: i) menggunakan replay pengalaman untuk memutuskan korelasi dalam data dan membawa kita kembali ke pengaturan independen dan terdistribusi identik (iid). Kami menyimpan semua kebijakan

sebelumnya dalam memori replay dan belajar darinya; ii) membekukan jaringan-Q target untuk menghindari osilasi dan memutus korelasi antara jaringan-Q dan target; dan iii) klip hadiah atau normalisasi jaringan secara adaptif ke kisaran yang masuk akal. Ini menuntut penggunaan metode gradien yang kuat.

Dibandingkan dengan RL konvensional, DQN mengadopsi jaringan saraf untuk menghitung nilai Q. Untuk setiap status sebagai input, nilai Q individual akan dihitung untuk setiap tindakan. Setelah propagasi maju tunggal dari jaringan saraf, nilai Q untuk semua tindakan diperbarui. Diberikan transit < s, a, r, s' >, algoritma pemutakhiran tabel nilai Q diberikan di bawah ini:

1) Lakukan propagasi maju mulai dari keadaan saat ini s dan dapatkan nilai Q yang diprediksi untuk semua tindakan;
2) Lakukan propagasi maju mulai dari keadaan berikutnya s', hitung nilai Q maksimum, yaitu max Q(s', a');
3) Tetapkan nilai Q target r + max Q(s', a') berdasarkan hasil yang dihitung pada Langkah2 dan nilai prediksi Q(s, a) yang dihitung pada langkah 1.
4) Perbarui bobot s dari jaringan saraf dengan algoritma propagasi mundur. Fungsi kerugian ditunjukkan sebagai

$$L = \frac{1}{2}[\underbrace{r + \max Q(s', a')}_{target} - \underbrace{Q(s, a)}_{prediction}]^2 \qquad (8.9)$$

5) Model mengacu pada ekspresi deskriptif untuk menspesifikasikan perilaku agen dalam lingkungan belajar. Perancang dapat berinteraksi dengan model dan belajar dari eksperimen. Setiap tindakan agen didasarkan pada lingkungan tertentu. Untuk lingkungan seperti itu, keadaan agen berikutnya memiliki banyak kemungkinan situasi, dan sulit untuk menentukan dengan tepat mana keadaan berikutnya, yaitu, probabilitas transisi agen masuk ke keadaan berikutnya yang spesifik, dan kemungkinan hadiah. tidak dapat dikonfirmasi. Selama lingkungan berubah, agen harus melintasi semua kemungkinan keadaan berikutnya setiap saat, yang menyebabkan penurunan efisiensi pembelajaran. Akibatnya, agen harus mampu mengambil strategi tindakan yang tepat berdasarkan lingkungan saat ini dan pengalaman masa lalu, mengingat situasi bahwa lingkungan tidak diketahui atau di bawah perubahan konstan.

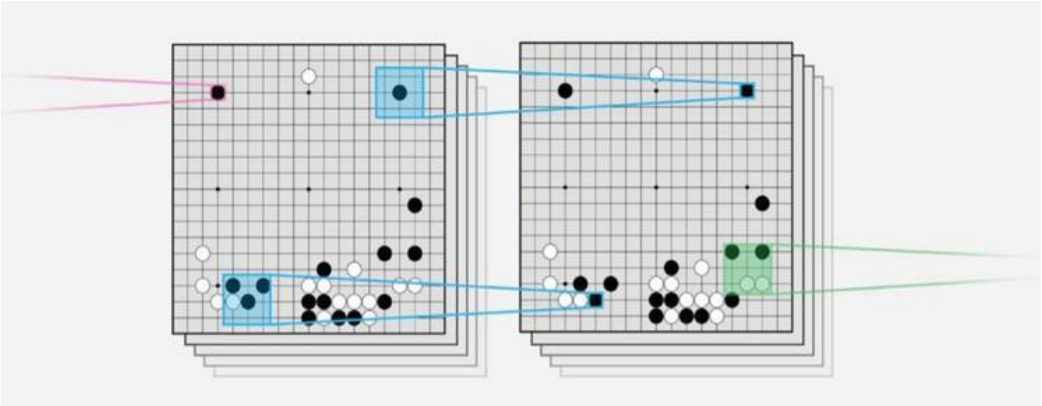
Jika agen mempelajari kebijakan optimal hanya dengan iterasi nilai tanpa belajar dari model, pengalaman setiap pengetahuan pembelajaran tidak akan sepenuhnya digunakan, yang mengarah pada tingkat konvergensi yang relatif lambat dan solusi suboptimal daripada solusi optimal. Meskipun penerapan RL berhasil, status fiturnya perlu disetel secara manual. Dengan demikian, sulit untuk menangani adegan yang kompleks, dan terkadang menemui masalah, dimensi bencana. Di DQN, DL diperkenalkan untuk mempelajari fitur secara otomatis. Dengan demikian, menggabungkan RL dan DL memungkinkan pembelajaran otomatis fitur dalam adegan dinamis, sementara keputusan pembelajaran, yaitu tindakan, dipilih secara optimal oleh RL

Kompetisi Game Google AlphaGo

Program AlphaGo dibangun dengan empat bagian fungsional: Jaringan Kebijakan, Peluncuran Cepat, Jaringan Nilai, dan Pencarian Pohon Monte Carlo (MCTS). Jaringan fast rollout dilatih dengan model linier yang memanfaatkan fitur lokal, yang memiliki kecepatan tinggi tetapi akurasi rendah, sedangkan Jaringan Kebijakan memiliki kecepatan rendah tetapi akurasi tinggi, yang diimplementasikan menggunakan jaringan saraf konvolusi dalam berdasarkan fitur global. Nilai Jaringan memperkirakan siapa yang akan menang mengingat keadaan saat ini, catur hitam atau putih. Gabungan MCTS dipandu oleh tiga bagian di atas.

Konstruksi CNN di AlphaGo dan Proses Pelatihannya

Permainan Go dimainkan pada papan kotak 19 × 19, seperti yang ditunjukkan pada Gambar 8.12. Batu hitam dan putih ditempatkan di papan, satu per satu oleh dua pemain secara bergantian. Setelah batu dengan warna yang sama sepenuhnya dikelilingi oleh batu lcloud, mereka akan dikeluarkan dari papan. Pemenangnya berakhir dengan mengendalikan area yang lebih besar. Ini pada dasarnya adalah permainan merebut-dan-kontrol. Permainan ini melibatkan ruang pencarian yang besar di setiap gerakan batu. Sebuah jaringan saraf konvolusi (CNN) dapat dibangun (Gambar 8.12), berdasarkan penyisipan batu berturut-turut di lokasi grid strategis dari sisi kiri ke sisi kanan.



Gambar 8.12 Konstruksi jaringan saraf convolutional di atas papan permainan Go.

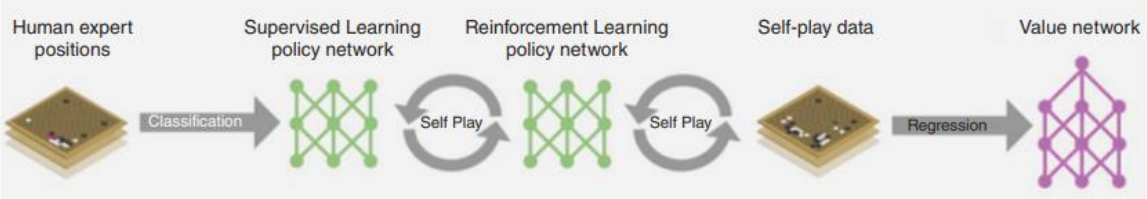
Gambar 8.13 mengilustrasikan proses pelatihan jaringan saraf menggunakan posisi ahli manusia. Pipa belajar mengalir dari kiri ke kanan. Posisi ahli berturut-turut berfungsi sebagai input. Jaringan kebijakan dimulai dengan algoritma pembelajaran yang diawasi untuk memaksimalkan kemungkinan dengan gradien stokastik yang layak. Setelah bermain sendiri, jaringan kebijakan diperkuat menggunakan algoritma RL. Sistem kemudian bergerak untuk menghasilkan data yang dapat diputar sendiri untuk memberi makan jaringan nilai guna menilai nilainya yang bermanfaat. Proses ini diulangi dengan banyak iterasi sampai kondisi pemenang terpenuhi. Pembelajaran terawasi dan pembelajaran penguatan dijelaskan dalam Contoh 8.4.

Contoh 8.4 Pembelajaran Terawasi dan Penguatan Jaringan Kebijakan sebelum Memasukkan Data Self-Play ke Jaringan Nilai menggunakan Regresi

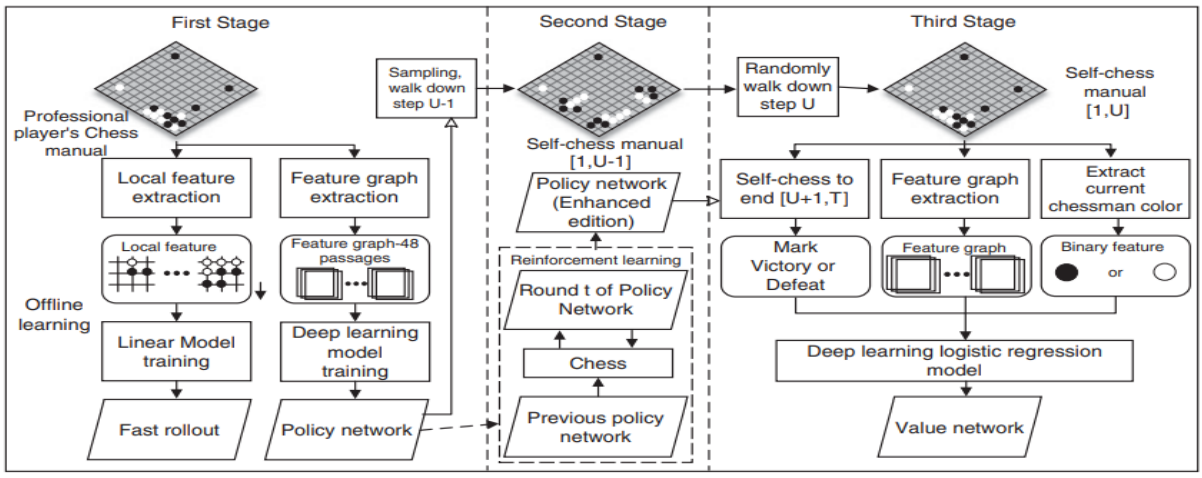
Contoh ini menunjukkan rincian pelatihan khusus dari pembelajaran terawasi yang diikuti dengan pembelajaran penguatan pada jaringan kebijakan. Kemudian pelatih menerapkan regresi untuk menilai penghargaan dalam jaringan nilai. Jaringan kebijakan menggunakan jaringan saraf konvolusi 12 lapis dengan permainan permainan mandiri sebagai data pelatihannya. Algoritma pelatihan untuk proses RL mencoba untuk memaksimalkan kemenangan z dengan pembelajaran penguatan gradien, yaitu:

$$\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial\sigma} z \tag{8.10}$$

di mana s adalah keadaan, a tindakan, dan σ imbalan. Jaringan kebijakan harus dilatih selama satu minggu pada 50 GPU di server di Google cloud. Kemudian, hasil yang sangat baik adalah bahwa jaringan kebijakan setara dengan 3 amatir dan dengan akurasi 80% berbanding 57% pada data uji yang diadakan dari pembelajaran terawasi.



Gambar 8.13 Jalur pelatihan permainan mandiri antara jaringan kebijakan dan jaringan nilai (dicetak ulang dengan izin dari David Silver, Google DeepMind, http://icml.cc/2016/tutorials/AlphaGo-tutorial-slides.pdf) [20].



Gambar 8.14 Proses pembelajaran off-line dari program AlphaGo (milik karya seni oleh Lu Wang dan Yiming Miao, Universitas Sains dan Teknologi Huazhong, Cina).

Jaringan nilai menerapkan 12 lapisan CNN untuk melakukan pembelajaran penguatan. CNN ini mirip dengan yang digunakan dalam jaringan kebijakan. Data pelatihan dari 30 juta posisi digunakan dari permainan ahli manusia (KGM 5+ dan). Pelatihannya, algoritma meminimalkan kesalahan kuadrat rata-rata dengan metode penurunan gradien stokastik yang dicirikan oleh:

$$\Delta\theta \propto \frac{\partial v_\theta(s)}{\partial\theta}(z - v_\theta(s)) \qquad (8.11)$$

Pelatihan jaringan nilai dilakukan serupa dengan pelatihan jaringan kebijakan. Seluruh jalur kebijakan dan jaringan nilai dapat mencapai evaluasi posisi kuat pertama yang belum pernah dicapai sebelumnya oleh program Go lainnya.

Arsitektur Sistem untuk Pembelajaran Penguatan Mendalam di Program AlphaGo

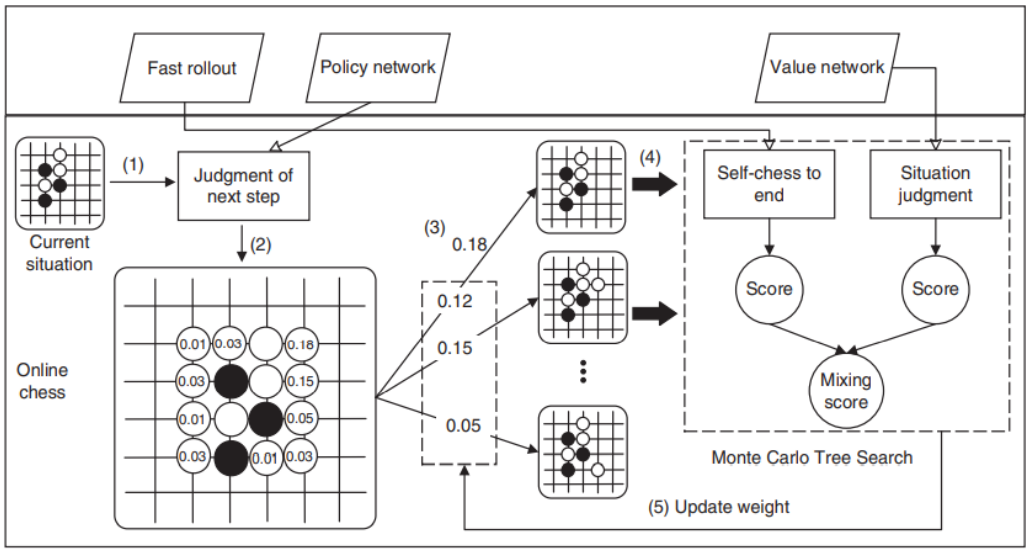
Pada Gambar 8.14, kami menyediakan diagram blok skema untuk menggambarkan aliran data dalam program AlphaGo dalam tiga tahap pembelajaran:

Ketiga tahap secara fungsional dijelaskan di bawah ini:

Tahap 1: Tahap ini melakukan *Deep learning* off-line, seperti yang ditunjukkan di sisi kiri Gambar 8.14. Metode *Deep learning* yang diawasi diterapkan terhadap manual catur pemain profesional. Tujuannya adalah untuk melakukan dua tugas secara paralel: i) ekstraksi fitur lokal dengan pelatihan model linier untuk menghasilkan peluncuran cepat untuk digunakan dalam pencarian Pohon Monte Carlo (Gambar 8.15); dan ii) menjalankan grafik fitur dalam 48 lintasan dengan model pembelajaran kesepakatan untuk memperbarui jaringan kebijakan untuk digunakan dalam dua tahap berikutnya pada Gambar 8.14 dan dalam proses eksekusi online pada Gambar 8.15.

Tahap 2: Tahap ini memperbarui jaringan kebijakan sebelumnya melalui pembelajaran penguatan ke jaringan kebijakan edisi yang disempurnakan, siap digunakan di Tahap 3. Manual catur mandiri digunakan untuk mengambil sampel jalan acak dari langkah u-1 ke langkah u akan dilakukan pada Tahap 3.

Tahap 3: Tahap ini menerapkan manual catur mandiri pada langkah u dalam tiga tugas paralel untuk menandai kemenangan atau kekalahan dan mengekstrak fitur berguna dan warna catur saat ini. Keluaran dari tiga tugas digabungkan untuk dimasukkan ke dalam model regresi logistik *Deep learning* untuk bekerja dengan jaringan nilai. Peluncuran yang diperbarui, jaringan kebijakan, dan jaringan nilai akan digunakan dalam proses eksekusi online dalam lima langkah yang ditentukan dalam Gambar 8.15.



Gambar 8.15 Proses bermain game AlphaGo online (milik Artwork oleh Xiaobo Shi dan Ping Zhou, Universitas Sains dan Teknologi Huazhong, Cina).

Langkah-Langkah Eksekusi dalam Pertandingan AlphaGo Online dengan Pemain Manusia

Pada Gambar 8.15, kami menunjukkan lima langkah eksekusi dalam program AlphaGo online. Pada dasarnya, langkah-langkah ini menggunakan jaringan kebijakan yang diperbarui untuk membuat keputusan dalam penempatan batu berikutnya menggunakan pencarian pohon Monte Carlo.

Kelima langkah tersebut secara singkat dijelaskan di bawah ini:

Langkah 1: Ekstrak fitur berdasarkan status batu yang ditempatkan saat ini.

Langkah 2: Perkirakan probabilitas bahwa setiap lokasi kosong akan ditempatkan dengan jaringan kebijakan.

Langkah 3: Hitung bobot untuk langkah selanjutnya pada setiap lokasi kosong berdasarkan probabilitas nilai awal.

Langkah 4: Periksa jaringan nilai dan jaringan peluncuran cepat untuk memperbarui skor. Di sini peluncuran cepat menuntut kecepatan tinggi daripada akurasi tinggi. Ulangi proses skor ini secara iteratif setelah setiap gerakan dalam pertandingan. Tingkat kemenangan diperkirakan di setiap lokasi penyisipan. Jaringan nilai mendapatkan hasil estimasi pada setiap state.

Langkah 5: Pilih keputusan dengan bobot maksimum untuk melakukan langkah selanjutnya. Pembaruan bobot ini dapat dilakukan secara paralel. Jika waktu mengunjungi suatu lokasi melebihi nilai tertentu, maka langkah selanjutnya akan dicari pada layer berikutnya di Monte Carlo Tree.

MCTS melakukan tugas-tugas berikut menggunakan jaringan nilai dan jaringan kebijakan, secara kolaboratif:

1) Pilih beberapa kemungkinan strategi yang akan dipilih lcloud untuk langkah selanjutnya berdasarkan situasi saat ini.
2) Menilai strategi lcloud, pilih langkah yang paling menguntungkan untuk melintasi subpohon kanan. Pohon pencarian AlphaGo tidak akan memperluas semua node, kecuali sepanjang jalur optimal dari subtree yang dilalui.
3) Bagaimana memutuskan tindakan terbaik di langkah selanjutnya membutuhkan perkiraan probabilitas menang dengan jaringan nilai. Pencarian pohon Monte Carlo perlu memprediksi hasil yang lebih dalam di sepanjang lapisan pohon. Hasil yang saling mendukung dari kedua jaringan ini menjadi kunci bagi AlphaGo untuk memenangkan permainan.
4) Setelah memutuskan tindakan terbaik yang akan diambil, kami memperkirakan kemungkinan langkah lcloud selanjutnya dan strategi yang sesuai melalui jaringan kebijakan berdasarkan lokasi tindakan terbaik.

Singkatnya, algoritme AlphaGo menggabungkan *Deep learning* dan pembelajaran penguatan, dilatih dengan pemain manusia dan manual mesin Go. Metode pembelajaran penguatan didasarkan pada pohon pencarian Monte Carlo dari jaringan nilai dan jaringan kebijakan, yang keduanya diimplementasikan dengan jaringan saraf yang dalam.

Contoh 8.5 Hasil Kinerja yang Dilaporkan pada Pencarian Pohon Monte-Carlo

Proses MCTS pada dasarnya menghabiskan semua kemungkinan gerakan dan penghargaan. Membangun pohon pencarian lookahead yang besar untuk mencakup jutaan kemungkinan, program AlphaGo 19 × 19 menggunakan MCTS untuk menghasilkan akurasi tinggi. Jaringan nilai dilatih untuk memprediksi gerakan manusia yang ahli, menggunakan database besar game Go profesional. Data kinerja terperinci dapat ditemukan di David Silver, Google DeepMind, http://www0.cs.ucl.ac.uk/staff/d.silver/web/Resources_files/deep_rl.pdf

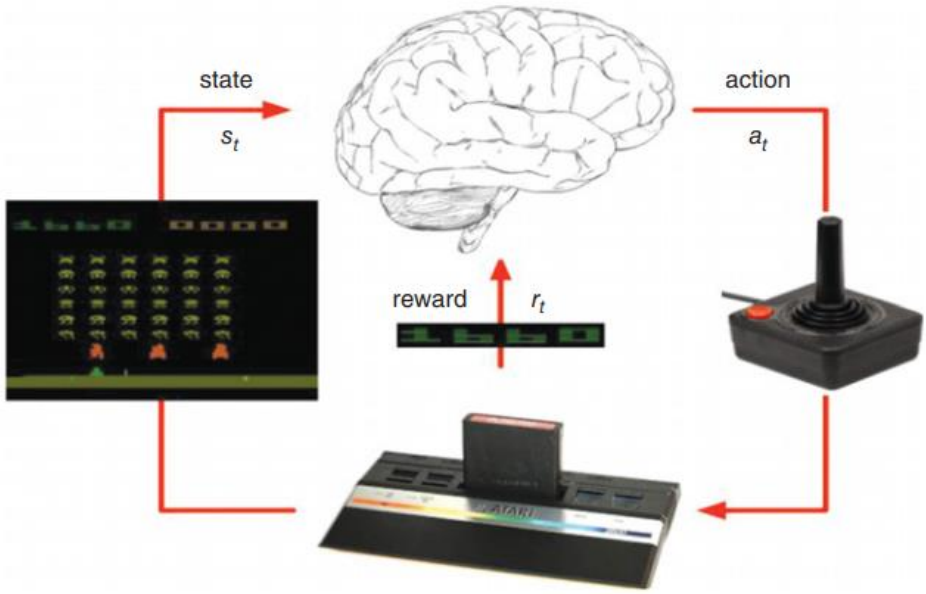
Akurasi prediktif CNN 12-layer mencapai 55%, yang merupakan peningkatan signifikan dibandingkan akurasi prediksi 31% dan 39% yang dilaporkan untuk program Go sebelumnya. Jaringan saraf jauh lebih kuat daripada program berbasis pencarian tradisional GnuGo, dan kinerjanya setara dengan MoGo untuk 100.000 peluncuran per gerakan. Pachi menjalankan pencarian yang dikurangi sebanyak 10.000 peluncuran per gerakan. Ini memenangkan sekitar 11% pertandingan melcloud Pachi, dengan 100.000 peluncuran per gerakan.

Game Flappybird menggunakan Reinforcement Learning

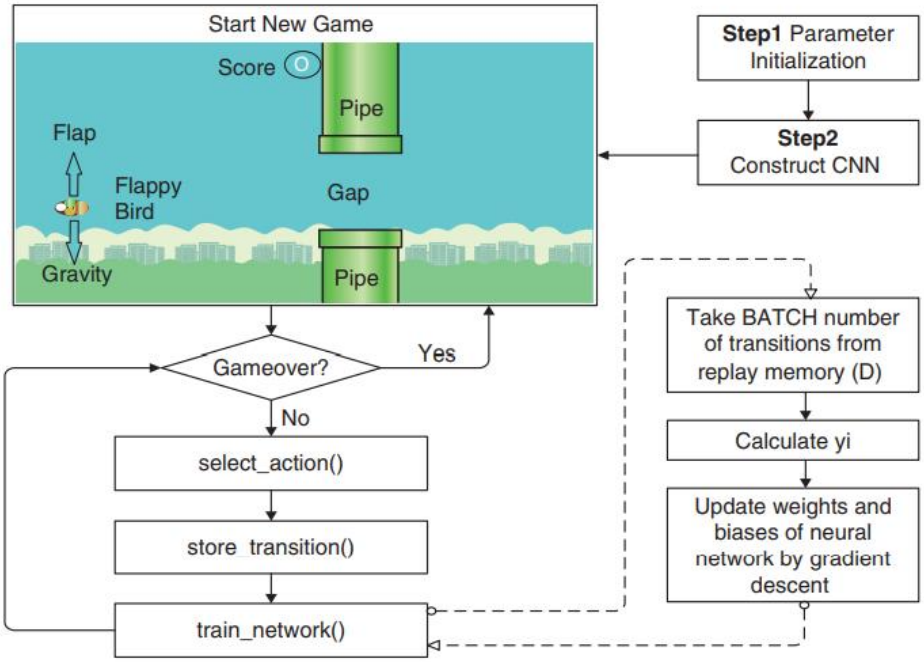
Salah satu aplikasi perwakilan DQN adalah memainkan Atari 2006, kumpulan game komputer hiburan populer. Ini mencakup 49 game independen, seperti Breakout dan game klasik lainnya. Masukan dari algoritma terdiri dari gambar layar permainan dan skor permainan. Tanpa mengetahui aturan permainan, DQN belajar bagaimana memainkan permainan itu sendiri dan menemukan strategi terbaik untuk bermain. Gambar 8.16 menunjukkan pengaturan game Artari yang melibatkan joystick dan kontrol geser untuk memainkan game. Interaksi antara keadaan belajar, tindakan dan penghargaan ditunjukkan oleh panah.

Contoh 8.6 Pengerjaan Deep Q Network Neural Network di Artari Games

Flappybird adalah permainan sederhana yang berbasis DQN, seperti yang diilustrasikan oleh flowchart pada Gambar 8.17. Pemain harus mengendalikan seekor burung agar tidak terbang terlalu tinggi atau terlalu rendah untuk menabrak pipa air.



Gambar 8.16 Reinforcement learning yang diterapkan dalam game play Artari.



Gambar 8.17 Diagram alir algoritma DQN untuk memainkan game Flappybird.

CNN ini mencakup tiga lapisan konvolusi dan satu lapisan koneksi penuh. Dalam permainan, ada dua tindakan yang dapat dilakukan pemain: menekan tombol "atas", yang membuat burung melompat ke atas atau tidak menekan tombol apa pun, yang membuatnya turun dengan kecepatan konstan. Kode semu untuk membuat CNN diberikan di bawah ini:

```
def createNetwork():
 # Weight of neural network
 # Input layer
 # Hidden layer
   # Output layer
      Qvalue= tf.matmul(h_fc1, W_fc2) + b_fc2 // Predict the value of Q
     return s, Qvalue, h_fc1
     a = tf.placeholder("float", [None, ACTIONS]) // Allocating space
                                                     to action
     y = tf.placeholder("float", [None]) // y is the Q value of
                                            optimal goal
 Qvalue_action = tf.reduce_sum(tf.mul(Qvalue, a), reduction_indices=1)
       //Neural network prediction value
 cost = tf.reduce_mean(tf.square(y - Qvalue_action)) // Loss function
 train_step = tf.train.AdamOptimizer(1e-6).minimize(cost)
     //Neural network optimization by minimizing the loss function
```

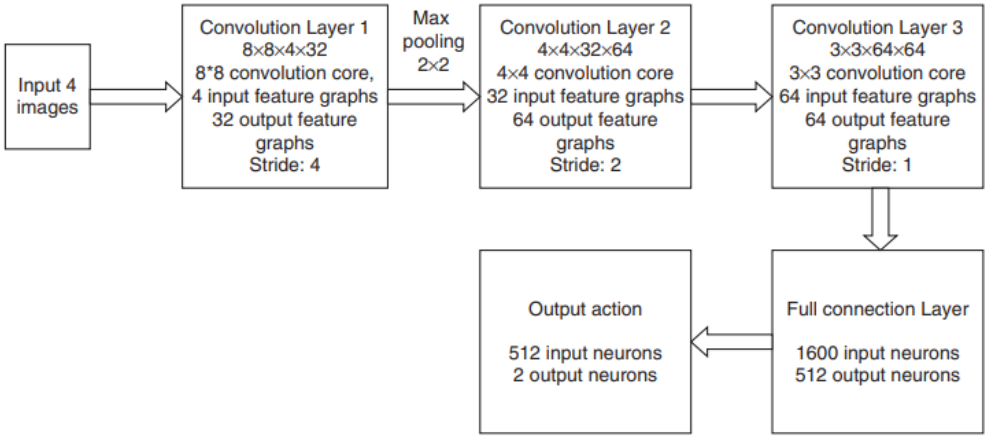
Konstruksi CNN mengikuti kaskade masukan empat gambar, tiga lapisan konvolusi dan satu lapisan koneksi penuh, dan tindakan keluaran seperti yang diilustrasikan pada Gambar 8.18. Menyetel ACTIONS=2 berarti Flappybird hanya memiliki dua tindakan, baik "naik" atau "turun". Untuk memecahkan masalah asinkron seperti itu, algoritme menetapkan FRAME_PER_ACTION sebagai jumlah sampel sebelum tindakan. Parameter GAMMA adalah faktor diskon dari imbalan di masa mendatang. REPLAY_MEMORY mewakili kapasitas memori replay. Sampel yang dikumpulkan adalah data deret waktu dan ada jalan keluar yang tumpang tindih antar sampel. Efisiensi rendah jika proses pemutakhiran nilai Q dilakukan setiap kali sampel diperoleh.

Sampel disimpan dalam memori replay dengan bentuk urutan (ϕ_t, a_t, r_t, ϕ_{t+1}) kemudian mengekstrak BATCH (jumlah minibatch yang dikeluarkan dari memori replay setiap kali) secara acak untuk melakukan pelatihan. Dalam kebanyakan kasus, adegan permainan dapat ditangkap dalam waktu yang sangat singkat, sementara pengkodean fitur oleh DL dan pembuatan kebijakan oleh RL membutuhkan computing intensif dengan penundaan yang lebih lama. Jadi, ketika setiap frame baru muncul selama bermain game, perlu untuk memeriksa apakah agen menyelesaikan perhitungan.

Setiap empat bingkai layar game bersama-sama sebagai sampel pelatihan. Setelah mendapatkan gambar baru, keadaan selanjutnya akan bergerak maju satu frame untuk memastikan bahwa gambar tersebut masih empat frame. Kemudian, sampel pelatihan akan mendapatkan set operasi. Untuk setiap sampel, kita harus menginisialisasi status s1 terlebih

dahulu. Sekarang mulai permainannya. Setelah menerima sampel, kita dapat memilih tindakan dengan dua cara:

1) Pilih tindakan secara acak berdasarkan epsilon, yang tidak akan berkurang seiring waktu.
2) Masukkan status saat ini ke dalam jaringan untuk menghitung nilai Q untuk setiap tindakan yang dipilih untuk langkah berikutnya.



Gambar 8.18 Konstruksi jaringan saraf convolutional yang digunakan dalam Game FlappyBird.

Ini menyiratkan bahwa memilih tindakan di akan menghasilkan nilai Q optimal yang dihitung. Kode semu untuk memilih tindakan diberikan di bawah ini:

```
def select_action():
Qvalue_t = Qvalue.eval(feed_dict={s : [s_t]})[0] // get Qvalue_t through
                                                    neural network
  a_t = np.zeros([ACTIONS]) // allocate space for action
  action_index = 0 // index for action
if t % FRAME_PER_ACTION == 0: // each action skips on frame of samples
  if random.random() <= epsilon: //randomly choose one action
   action_index = random.randrange(ACTIONS)
   a_t [random.randrange(ACTIONS)] = 1
  else: // take action based on the best policy
   action_index = np.argmax(Qvalue_t) // predict all the Q values for
     each action, and choose the action with maximum Q value
   a_t [action_index] = 1
else: a_t [0] = 1 //do nothing
```

Pelatihan CNN dalam dilakukan oleh kode semu berikut. Hadiah r_t dan input gambar x_{t+1} dari jaringan berikutnya akan diperoleh saat aksi at dijalankan di emulator, lalu masukkan urutan (ϕ_t, a_t, r_t, ϕ_{t+1}) ke dalam kumpulan pemutaran. Jika kapasitas kolam pemutaran penuh, buang urutan yang dimasukkan paling awal. Masukannya hanyalah gambar dan skor permainan, keluaran tindakan setelah pelatihan dengan jaringan saraf konvolusi. Karena tidak ada label, pembelajaran

penguatan diterapkan dan keputusan dibuat, setiap kali memilih tindakan dengan nilai maksimum sebagai output. Kode pseudo berikut cukup jelas dengan komentar:

```
def train_network():
if t > OBSERVE:
 minibatch = random.sample(D, BATCH)  //choose BATCH sequences from D
 s_j_batch = [d[0] for d in minibatch] //current state of corresponding
                                        sequence
 a_batch = [d[1] for d in minibatch] //action of corresponding sequence
 r_batch = [d[2] for d in minibatch] //reward of corresponding sequence
 s_j1_batch = [d[3] for d in minibatch] //next state of corresponding
                                         sequence
 y_batch = [] //allocate space for y
 Qvalue_j1_batch = Qvalue.eval(feed_dict = {s : s_j1_batch})
// compute the max Q value for the next state
 for i in range(0, len(minibatch)):
 terminal = minibatch[i][4] //next state of sequence
 if terminal:
 y_batch.append(r_batch[i])
 else:
 y_batch.append(r_batch[i] + GAMMA × np.max(Qvalue_j1_batch[i]))
 train_step.run(feed_dict = {
 y : y_batch,
 a : a_batch,
 s : s_j_batch})
```

8.4 ANALISIS DATA UNTUK APLIKASI MEDIA SOSIAL

Media sosial adalah sumber utama agregasi *Big data* dalam aktivitas kita sehari-hari. Di bagian ini, kami menilai teknologi analitik data yang diterapkan di industri media sosial dan dampaknya pada semua pekerjaan kehidupan. Kemudian kami mempelajari jaringan sosial dan analisis grafik komunitas sosial. Terakhir, kami menghadirkan sumber daya cloud pintar yang diperlukan untuk mendukung aplikasi analitik *Big data*. Jaringan sosial online dibentuk dengan individu atau organisasi melalui Internet. Entitas individu atau organisasi ini terkait, terhubung, atau terkait dengan minat khusus atau ketergantungan tertentu.

Pembangunan jaringan sosial didasarkan pada persahabatan pribadi, kekerabatan, orientasi profesional, kepentingan bersama, pertukaran keuangan, komunitas atau kelompok ras, keyakinan agama atau politik, pengetahuan atau prestise, penggemar selebriti, dll. Dalam jaringan sosial, node mewakili individu, dan ikatan antara simpul mewakili hubungan seperti persahabatan, kekerabatan dan rekan kerja. Layanan jejaring sosial online dibangun untuk mencerminkan hubungan sosial di antara orang-orang. Layanan ini diperkenalkan sebagai alat

komunikasi di antara orang-orang. Komunitas online tradisional lebih berorientasi pada kelompok, sementara situs web sosial modern sebagian besar dibangun secara individual.

Persyaratan *Big data* dalam Aplikasi Media Sosial

Kami meninjau persyaratan umum di bawah ini dalam aplikasi *Big data* di domain media sosial, keuntungan pemasaran dari mikroblog, dan streaming video. Layanan konsumen lebih suka menggunakan forum dan sistem seluler. Penjualan menikmati ulasan produk/layanan sebagian besar. Sumber daya manusia lebih memilih untuk memanfaatkan jaringan bisnis. Sebagian besar organisasi menerapkan jaringan sosial perusahaan. Pengguna media sosial seluler memanfaatkan fitur sensitif lokasi dan/atau waktu dari kumpulan *Big data* yang dikumpulkan. Mereka bertujuan untuk mengelola hubungan pelanggan, promosi penjualan, dan program insentif seperti yang dinilai dalam empat bidang yang dijelaskan di bawah ini:

1) **Riset pemasaran:** Dalam aplikasi media sosial seluler, pengguna sering mengumpulkan data dari pergerakan konsumen offline terlebih dahulu, sebelum mereka pindah ke perusahaan online. Pengumpulan data online dapat meningkat dengan cepat ke jumlah yang besar. Mereka harus ditangani tepat waktu dalam mode streaming terus menerus. Persyaratannya adalah agar semua pihak atau perusahaan terkait mendapat informasi yang baik tentang waktu transaksi yang tepat dan komentar yang dibuat selama transaksi atau kunjungan jejaring sosial.
2) **Komunikasi dalam pertukaran media sosial:** Komunikasi media sosial seluler mengambil dua bentuk: bisnis-ke-konsumen (B2C), di mana perusahaan dapat membangun koneksi ke konsumen berdasarkan lokasinya dan memberikan ulasan tentang konten yang dibuat pengguna. Misalnya, McDonald's menawarkan kartu hadiah $5 dan $10 kepada 100 pengguna yang dipilih secara acak di antara mereka yang check-in di salah satu restorannya. Promosi ini meningkatkan penjualan sebesar 33% dan menghasilkan banyak posting blog dan umpan berita melalui pesan Twitter.
3) **Promosi penjualan dan diskon:** Meskipun pelanggan harus menggunakan kupon cetak di masa lalu, media sosial seluler memungkinkan perusahaan menyesuaikan promosi untuk pengguna tertentu pada waktu tertentu. Misalnya, ketika meluncurkan layanan California-Cancun, Virgin America menawarkan penumpang dua penerbangan ke Meksiko dengan harga satu. Pengembangan hubungan dan program loyalitas dapat dibentuk untuk meningkatkan hubungan jangka panjang dengan pelanggan. Misalnya, perusahaan dapat membuat program loyalitas untuk memungkinkan pelanggan yang check-in secara teratur di suatu lokasi untuk mendapatkan diskon atau fasilitas.
4) **e-Commerce:** Aplikasi media sosial seluler seperti Amazon.com dan Pinterest telah mulai mempengaruhi tren peningkatan popularitas dan aksesibilitas e-commerce, atau pembelian online. Peristiwa e-commerce tersebut dapat dilakukan sebagai B2B (business-to-business), B2C (business-to-customer), C2B (customer-to-business atau C2C (customer-to-customer in

a peer-to-peer (P2P) fashion). Belakangan ini, transaksi O2O juga terjadi secara online ke offline atau offline ke penjualan online atau pertukaran bisnis.

Tabel 8.3 menyajikan daftar 14 jejaring sosial terkemuka berdasarkan jumlah akun pengguna aktif, per April 2016. Jelas, Facebook dan WhatsApp sama-sama sukses menarik pengguna. Di Cina, pengguna QQ dan WeChat berkembang pesat, yang melibatkan hampir dua pertiga populasi Cina. Gladwell telah mengindikasikan bahwa media sosial dibangun di sekitar ikatan yang lemah. Misalnya, peran media sosial dalam mendemokratisasi partisipasi media mungkin kurang ideal, memungkinkan siapa saja yang memiliki koneksi Internet untuk menjadi pembuat konten. Ini mungkin dapat memberdayakan pengguna "aktif" untuk menginspirasi pengguna "pasif". Tetapi data survei internasional menunjukkan bahwa anggota audiens media online sebagian besar merupakan konsumen pasif, sementara pembuatan konten didominasi oleh beberapa. Contoh berikut menilai plus dan minus industri jejaring sosial dalam beberapa tahun terakhir.

Tabel 8.13 14 Jejaring Sosial Teratas Berdasarkan Populasi Pengguna Global Tahun 2016.

Jaringan sosial	Pengguna Aktif	Jaringan sosial	Pengguna Aktif
Facebook	1,59 Miliar	Indonesia	320 Juta
Ada apa	1,00 Miliar	Baidue Tieba	300 juta
QQ	853 Juta	Skype	300 juta
Wechat	697 Juta	Viber	249 Juta
zona Q	640 Juta	Sina Weibo	222 Juta
Tumblr	555 Juta	Garis	215 Juta
Instagram	400 juta	Snapchat	200 juta

Contoh 8.7 Dampak Positif dan Negatif Beberapa Jejaring Sosial

Media sosial mendorong industri komunikasi. Jejaring sosial telah memengaruhi kehidupan kita sehari-hari dan mendorong beberapa perubahan sosial. Jelas, dampak pengaruh jejaring sosial meningkat seiring dengan populasi penggunanya. Facebook dan Twitter menyediakan forum terbuka bagi penggunanya untuk menjalin hubungan, berbagi informasi, dan membuka diskusi tentang masalah publik atau pribadi. Mereka bahkan telah mempengaruhi pemilihan presiden AS dan memicu beberapa reformasi atau revolusi politik di belahan dunia tertentu.

Orang-orang di seluruh dunia memanfaatkan media sosial untuk kenyamanannya dengan hampir tanpa biaya bagi masyarakat umum. Kami meninjau beberapa kasus peristiwa jejaring sosial untuk menilai dampak positif dan negatifnya. Lingkaran pertemanan, kelompok bisnis, kelompok politik dan minat khusus dapat secara terbuka mendiskusikan keprihatinan mereka di platform ini. Misalnya, pandangan positif dalam menggunakan jejaring sosial dapat mengatasi

masalah kesehatan, keselamatan dan sosial, seperti pengendalian alkohol, pencegahan narkoba, menghindari pelecehan seksual atau kekerasan dalam rumah tangga, dan menghentikan kejahatan terorganisir. Semua ini harus didorong, mengungkapkan sisi terang dan positif dari masyarakat kita.

Di sisi negatif, ada beberapa efek yang sangat buruk seperti menyebarkan desas-desus yang tidak sehat, menghancurkan reputasi orang secara sembarangan, menyebabkan kerusuhan di lingkungan, dan ketegangan rasial di suatu negara. Sebagai contoh ekstrim, jika seseorang menampilkan adegan bunuh diri secara online, hal itu dapat memicu anak-anak atau orang yang putus asa untuk mengikutinya. Pornografi, jika tidak diatur, dapat menyebar lebih luas untuk meracuni generasi muda. Penjualan obat-obatan ilegal atau penyalahgunaan obat dapat melukai banyak orang yang tidak bersalah.

Ringkasnya, kita melihat beberapa efek positif dari kebebasan berbicara dan mempromosikan harmoni dalam nilai-nilai sosial. Namun, jika kita tidak belajar dari konsekuensi negatif dengan membiarkan senjata menjadi benar-benar di luar kendali dan membiarkan kebebasan yang dilepaskan untuk melukai orang yang tidak bersalah secara massal, kita semua pada akhirnya akan menderita. Kesimpulannya, penggunaan negatif jejaring sosial harus diatur oleh penegakan hukum dan ketertiban. Jejaring sosial tidak boleh dibiarkan mengarah ke arah yang tidak bermoral atau tidak sehat.

Jaringan Sosial dan Analisis Grafik

Jejaring sosial berguna dalam ilmu sosial untuk mempelajari hubungan antara individu, kelompok, organisasi atau bahkan seluruh masyarakat. Istilah ini digunakan untuk menggambarkan struktur sosial yang ditentukan oleh interaksi semacam itu. Ikatan antara anggota merupakan konvergensi dari berbagai kontak sosial. Aksioma untuk memahami interaksi sosial didasarkan pada sifat-sifat hubungan antara dan di dalam kelompok sosial. Karena adanya banyak hubungan yang berbeda, analisis jaringan berguna untuk berbagai konstruksi jaringan sosial. Dalam ilmu sosial, kajian ini berkaitan dengan antropologi, biologi, ilmu komunikasi, ekonomi, geografi, ilmu informasi, ilmu organisasi, psikologi sosial, sosiologi dan sosiolinguistik.

Secara umum, jaringan sosial mengatur dirinya sendiri, muncul dan kompleks, sehingga pola yang koheren secara global muncul dari interaksi lokal elemen-elemen yang membentuk sistem. Pola-pola ini menjadi lebih jelas ketika ukuran jaringan meningkat. Namun, analisis jaringan global dari semua hubungan interpersonal di dunia tidak mungkin dilakukan. Keterbatasan praktis adalah karena etika, rekrutmen peserta dan pertimbangan ekonomi.

Tingkat Jaringan Media Sosial

Nuansa sistem lokal mungkin hilang dalam analisis jaringan besar, oleh karena itu kualitas informasi mungkin lebih penting daripada skalanya untuk memahami properti jaringan. Dengan demikian, jaringan sosial dianalisis pada skala yang relevan dengan pertanyaan teoretis peneliti. Meskipun tingkat analisis tidak selalu eksklusif satu sama lain, ada tiga tingkat umum di mana

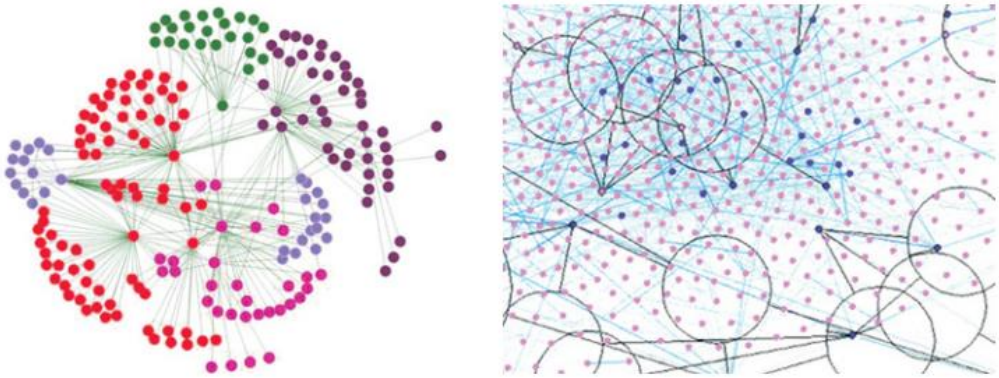
jaringan dapat jatuh: tingkat mikro, tingkat meso, dan tingkat makro. Contoh berikut menunjukkan perbedaan mereka dalam pembentukan jejaring sosial tersebut.

Contoh 8.8 Pembangunan Jejaring Sosial dalam Tiga Tingkat

Pada tingkat mikro, penelitian jaringan sosial biasanya dimulai dengan seorang individu, bola salju sebagai hubungan sosial ditelusuri, atau mungkin dimulai dengan sekelompok kecil individu dalam konteks sosial tertentu. Ini ditunjukkan pada Gambar 8.19(a), di mana kelompok-kelompok kecil terbentuk, rata-rata, dengan seratus atau kurang node peer. Node anggota dalam grup yang sama mungkin memiliki lebih banyak ikatan koneksi tepi. Grup yang berbeda (atau disebut komunitas) terhubung secara longgar dengan koneksi tepi yang jauh lebih sedikit.

Daripada menelusuri interaksi interpersonal, jaringan sosial tingkat makro umumnya berkembang dari hasil interaksi yang lebih besar, seperti sumber daya ekonomi atau lainnya. Jaringan skala besar adalah istilah yang agak sinonim dengan jaringan sosial "tingkat makro", seperti yang ditunjukkan pada Gambar 8.19(b). Ini sering digunakan dalam ilmu sosial, komputer atau perilaku sehubungan dengan kelas ekonomi, masyarakat profesional atau afiliasi politik.

Teori tingkat meso dimulai dengan ukuran populasi yang berada di antara tingkat mikro dan makro. Namun, tingkat meso juga dapat merujuk ke jaringan yang dirancang khusus untuk mengungkapkan koneksi antara tingkat mikro dan makro. Jaringan tingkat meso memiliki kepadatan rendah dan mungkin menunjukkan proses kausal yang berbeda dari jaringan tingkat mikro antarpribadi. Pada Gambar 8.19(b), grafik jaringan tingkat makro mungkin jauh melebihi batas cutoff yang ditunjukkan di semua sisi jaringan. Grup jaringan yang dilingkari berada di tingkat mikro, sedangkan hubungan tebal di antara kelompok mikro sesuai dengan koneksi tingkat meso. Beberapa jaringan mikro yang terikat pada beberapa simpul pusat membentuk apa yang disebut jaringan meso.



(a) Micro level networks (b) Macro-level and meso-level networks

Gambar 8.19 Konstruksi jaringan sosial tingkat mikro, tingkat meso, dan tingkat makro.

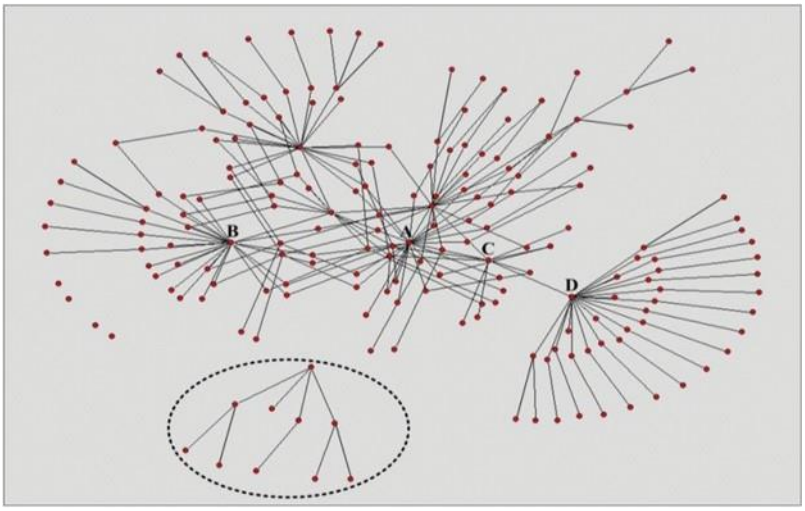
Karakteristik Grafik Sosial

Analisis jaringan sosial telah muncul sebagai teknik kunci dalam sosiologi modern. Mengkarakterisasi hubungan yang ada di antara kelompok sosial seseorang adalah tugas utama

dalam analisis jaringan sosial. Juga, pengguna menghadapi apa yang disebut masyarakat “dunia kecil”, di mana semua orang terkait dalam rantai pendek kenalan sosial, dengan satu atau lain cara. Semua jaringan sosial tidak begitu kacau atau acak seperti yang diperkirakan sebelumnya, tetapi mereka memiliki struktur yang mendasarinya. Hubungan sosial sering dipetakan ke dalam grafik berarah atau tidak berarah, kadang-kadang disebut grafik kenalan atau hanya grafik hubungan sosial.

Node dalam grafik sosial sesuai dengan pengguna atau aktor dan tepi grafik atau tautan mengacu pada ikatan atau hubungan di antara node. Grafik dapat menjadi kompleks dan terstruktur secara hierarkis untuk mencerminkan hubungan di semua tingkatan. Ada banyak jenis ikatan antara node. Jejaring sosial beroperasi dari tingkat keluarga hingga tingkat nasional dan global. Ada pro dan kontra dari jejaring sosial. Sebagian besar masyarakat bebas menyambut jaringan sosial. Untuk alasan politik atau agama, beberapa negara memblokir penggunaan jejaring sosial untuk mencegah kemungkinan penyalahgunaan:

- **Properti Grafik Jaringan Sosial:** Jaringan sosial memainkan peran penting dalam pemecahan masalah, menjalankan organisasi, dan menghitung sejauh mana individu berhasil mencapai tujuan mereka. Jejaring sosial hanyalah peta dari semua ikatan yang relevan antara semua node aktor. Jaringan juga dapat digunakan untuk mengukur modal sosial – nilai yang diperoleh individu dari jaringan sosial. Konsep-konsep ini sering ditampilkan dalam grafik jaringan sosial. Contoh grafik jejaring sosial ditunjukkan pada Gambar 8.20. Titik-titik hitam adalah node (pengguna) dan ujung-ujungnya menghubungkan node di bawah hubungan dasi yang ditentukan. Di bawah ini adalah beberapa sifat menarik dari grafik sosial.



Gambar 8.20 Grafik representasi jaringan sosial.

- **Derajat Node, Jangkauan, Panjang Jalur dan Keterantaraan:** Derajat node adalah jumlah tetangga node terdekat dari sebuah node. Jangkauan didefinisikan sebagai sejauh mana setiap anggota jaringan dapat mencapai anggota lain dari jaringan. Panjang jalur

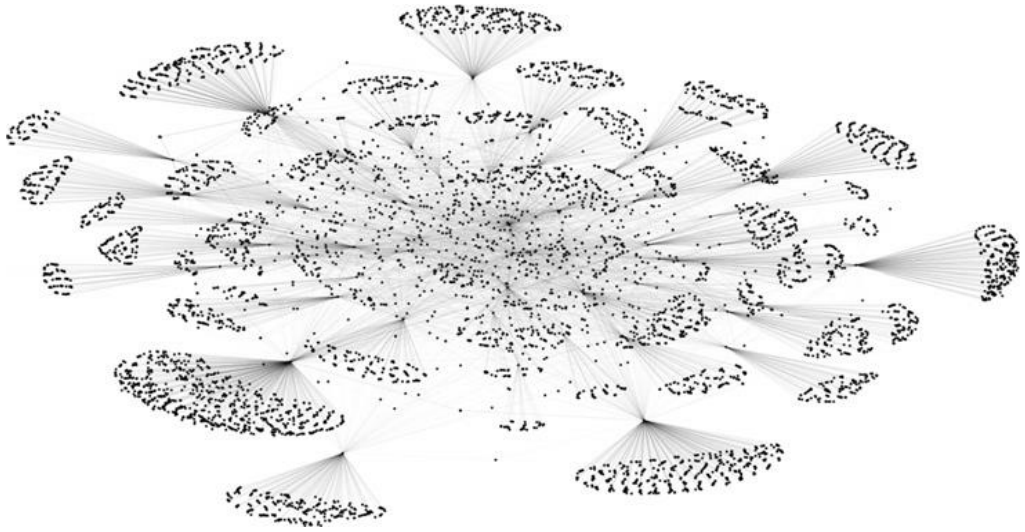
mengukur jarak antara pasangan node dalam jaringan. Panjang jalur rata-rata adalah rata-rata jarak ini antara semua pasangan simpul. Keantaraan mengungkapkan sejauh mana sebuah simpul terletak di antara simpul-simpul lain dalam jaringan. Ini mengukur jumlah orang yang terhubung secara tidak langsung dengan seseorang melalui tautan langsung mereka.

- **Kedekatan dan Kohesi:** Tingkat kedekatan seorang individu dengan semua individu lain dalam suatu jaringan (langsung atau tidak langsung). Ini mencerminkan kemampuan untuk mengakses informasi melalui anggota jaringan. Dengan demikian, kedekatan adalah kebalikan dari jumlah jarak terpendek antara setiap individu dan setiap orang lain dalam jaringan. Kohesi adalah sejauh mana aktor terhubung langsung satu sama lain melalui ikatan kohesif. Kelompok diidentifikasi sebagai “klik” jika setiap individu secara langsung terikat dengan setiap individu lainnya.
- **Sentralitas dan Sentralisasi:** Sentralitas menunjukkan kekuatan sosial dari sebuah node berdasarkan seberapa baik mereka "menghubungkan" jaringan. Node A, B dan D pada Gambar 8.20 adalah semua node sentralitas dengan derajat node yang berbeda.
- **Lingkaran Sosial atau kelompok:** Ini mengacu pada beberapa kelompok terstruktur. Jika kontak langsung kurang ketat atau sebagai blok kohesif struktural, maka lingkaran sosial dapat dibuat baik secara longgar atau erat, tergantung pada aturan ketat yang diterapkan. Node-node di dalam lingkaran Gambar 8.20 membentuk sebuah cluster. Koefisien pengelompokan adalah kemungkinan bahwa dua asosiasi dari sebuah node adalah asosiasi itu sendiri.
- **Jaringan Terpusat versus Terdesentralisasi:** Sentralitas memberikan indikasi kasar tentang kekuatan sosial sebuah simpul berdasarkan seberapa baik mereka "menghubungkan" jaringan. Antara-ness, kedekatan dan derajat semua ukuran sentralitas. Jaringan terpusat memiliki tautan yang tersebar di sekitar satu atau beberapa node, sedangkan jaringan terdesentralisasi adalah jaringan di mana ada sedikit variasi antara jumlah tautan yang dimiliki setiap node.
- **Jembatan dan Jembatan Lokal:** Tepi adalah jembatan jika menghapusnya akan menyebabkan titik akhirnya terletak pada kelompok atau komponen grafik yang berbeda. Sebagai contoh, tepi antara node C dan D pada Gambar 8.20 adalah sebuah jembatan. Titik akhir jembatan lokal tidak memiliki tetangga yang sama. Sebuah jembatan lokal terkandung dalam sebuah siklus.
- **Prestise dan Radialitas:** Dalam grafik sosial, prestise menggambarkan sentralitas simpul. “Gelar Prestise”, “Kedekatan Prestise” dan “Status Prestise” adalah semua ukuran Prestise. Radialitas adalah sejauh mana jaringan menjangkau dan memberikan informasi dan pengaruh baru.
- **Kohesi Struktural, Kesetaraan dan Lubang:** Kohesi struktural adalah jumlah minimum anggota yang, jika dikeluarkan dari grup, akan memutuskan grup. Kesetaraan struktural

mengacu pada sejauh mana node memiliki seperangkat hubungan yang sama ke node lain. Node ini tidak memiliki ikatan satu sama lain. Sebuah lubang struktural dapat diisi dengan menghubungkan satu atau lebih link untuk mencapai node lain. Ini terkait dengan modal sosial: dengan menghubungkan dua orang yang terputus, kita dapat mengontrol komunikasi mereka.

Contoh Analisis Grafik Sosial

Layanan jejaring sosial online disatukan melalui identitas, percakapan, berbagi, telepresence, hubungan, afiliasi, dll. Ini ditawarkan melalui akses Internet dan layanan web. Layanan jejaring sosial online awal mencakup pencarian pekerjaan, kencan atau layanan papan buletin, dll. Komunitas online tradisional diatur dalam kelompok yang berbeda berdasarkan minat dan wilayah yang berbeda, sementara situs jejaring sosial modern selalu berorientasi pada individu atau mengikuti peer-to-peer interaksi.



Gambar 8.21 Grafik representasi jaringan sosial.

Di bawah ini adalah beberapa ide dalam menyediakan layanan jejaring sosial online:

1) Halaman atau profil pribadi untuk setiap pengguna yang ditautkan oleh koneksi sosial;
2) Grafik sosial melintasi sepanjang tautan atau jaringan sosial tertentu;
3) Alat komunikasi antar peserta atau pengguna terdaftar;
4) Bagikan informasi khusus seperti musik, foto, video, dll. dengan teman atau grup profesional;
5) Mengoperasikan komunitas di bidang topik khusus seperti perawatan kesehatan, olahraga, hobi, dll.;
6) Perangkat lunak atau basis data yang disesuaikan mungkin diperlukan untuk menyiapkan layanan jejaring sosial;
7) Loyalitas pelanggan yang kuat dan pertumbuhan keanggotaan yang viral merupakan ciri khas komunitas jejaring sosial;

8) Jejaring sosial memperoleh pendapatan dengan menjual keanggotaan premium dan akses ke konten premium.

Contoh 8.9 Representasi Grafik Jaringan Pertukaran Email

Pada Gambar 8.21, kami menunjukkan jaringan pertukaran email 430-node dalam laboratorium penelitian kecil. Hanya pertukaran email internal yang ditampilkan; email ke dan dari sumber eksternal tidak ditampilkan di sini. Tepi sesuai dengan mereka yang telah mengirim email satu sama lain. Jumlah tepi dapat dengan mudah mencapai 200.000 jika terhubung sepenuhnya. Di sini kita hanya melihat grafik yang terhubung sebagian dengan kira-kira 5000 tepi.

Di University of Southern California, jaringan pertukaran email seperti ini mungkin harus mencakup 40.000 node. Ini mungkin berakhir sebagai grafik koneksi dengan 1,5 juta tepi pertukaran email. Jika email eksternal disertakan, jaringan email akan diperbesar secara signifikan untuk mencakup 10 juta node dalam skala global.

Twitter tidak menyalin semua hubungan offline ke situs web. Oleh karena itu, daya tarik untuk menggunakan Twitter jauh lebih sedikit daripada Facebook saat ini. Di sisi lain, Facebook tidak seterbuka Twitter. Fitur-fitur ini membuat orang lebih percaya Facebook daripada Twitter, karena masalah privasi akan membuat orang memilih sistem yang lebih tertutup.

Last but not least, Facebook lebih kompleks dalam fungsi daripada Twitter. Meskipun Twitter memiliki banyak aplikasi pihak ketiga, tetap saja tidak nyaman bagi pengguna pemula. Facebook telah menyematkan banyak fitur umum di situs web. Jika seseorang tidak suka menjelajahi aplikasi pihak ketiga untuk beberapa fungsi umum, mereka akan pergi ke Facebook daripada Twitter, berdasarkan tren saat ini.

Teknik Penyaringan dan Sistem Rekomendasi

Kita perlu membangun sistem rekomendasi untuk film, pariwisata, dan restoran agar aktivitas kehidupan sehari-hari kita lebih terorganisir, nyaman, dan menyenangkan; penyaringan sosial atau kolaboratif data yang tidak diinginkan dengan polling pendapat massa untuk membuat keputusan berdasarkan peringkat. Pemfilteran berbasis konten diperlukan untuk merekomendasikan item berdasarkan fitur produk dan peringkat oleh pengguna. Penyaringan demografis membantu membuat keputusan berdasarkan informasi demografis pengguna. Terakhir, pemfilteran berbasis pengetahuan membuat keputusan yang bijaksana berdasarkan pengetahuan keahlian dan reputasi rekan, dll. Pemfilteran hibrid menggabungkan keunggulan teknik pemfilteran di atas untuk membuat keputusan yang lebih cerdas.

Mendorong Analisis Data untuk Penegakan Keamanan Cloud/Jaringan

Ini adalah area penelitian panas untuk menerapkan *Big data* untuk penegakan keamanan siber. Analitik *Big data* sangat dibutuhkan dalam keamanan jaringan, analitik peristiwa perusahaan, pemantauan aliran bersih untuk mengidentifikasi botnet, deteksi ancaman terus-menerus, berbagi data, asal, dan teknik tata kelola untuk manajemen kepercayaan dengan sistem reputasi.

Dukungan Cloud untuk IoT dan Aplikasi Jejaring Sosial

Dalam sistem cyber-fisik (CPS), algoritma analitik dapat bekerja lebih akurat dalam konfigurasi sistem, pengetahuan fisik, dan prinsip kerja. Untuk mengintegrasikan, mengelola, dan menganalisis mesin, kami ingin menangani data secara lebih efisien selama berbagai tahap siklus hidup mesin. Model kopling antara manusia dan mesin sangat difasilitasi oleh penggunaan penyimpanan cloud dan sistem analitik. Ini melibatkan operasi penginderaan, penyimpanan, sinkronisasi, sintesis, dan layanan.

Aplikasi cloud yang cerdas dan meresap sangat diminati oleh individu, rumah, komunitas, perusahaan dan pemerintah, dll. Ini termasuk kalender terkoordinasi, rencana perjalanan, manajemen pekerjaan, acara, dan layanan manajemen catatan konsumen (CRM). Bidang minat lainnya termasuk pengolah kata kooperatif, presentasi on-line, desktop berbasis web, berbagi dokumen on-line, kumpulan data, foto, video dan database, distribusi konten, dll. Penyebaran cluster konvensional, grid, P2P dan jejaring sosial aplikasi sangat banyak diminati di lingkungan cloud. Aplikasi Earthbound mungkin menuntut elastisitas dan paralelisme untuk menghindari pergerakan data yang besar dan untuk mengurangi biaya penyimpanan.

Arsitektur Jaringan Sosial Online

Sangat diinginkan untuk menyesuaikan OSN untuk mempertahankan persaingan di bidang ini. Penyedia jaringan sosial harus memilih nama merek dengan antarmuka API dan variabel profilnya sendiri. Kategori forum yang dipilih harus relevan dengan komunitas pengguna yang cukup besar. Platform OSN harus menyertakan fungsionalitas khusus yang memudahkan pengguna untuk bergabung dan menikmati layanan. Selanjutnya, penyedia harus membuktikan konsep pemasaran online untuk menarik anggota untuk bergabung dan keluar dengan bebas. Perangkat lunak yang sangat canggih, pusat data virtual, atau platform cloud pemrosesan dan penyimpanan diperlukan.

Komunitas jejaring sosial harus beroperasi dengan andal dengan ketersediaan dan kinerja yang tinggi. Logikanya, OSN menyediakan platform P2P. Namun, layanan jejaring sosial populer modern semuanya dibangun dengan arsitektur client-server untuk pengelolaan dan pemeliharaan yang mudah. Ini berarti bahwa semua entri blog, foto, video, dan hubungan jejaring sosial disimpan dan dikelola oleh cloud pribadi yang dimiliki oleh penyedia layanan. Dengan ratusan juta pengguna, situs jejaring sosial besar harus memelihara pusat data yang besar. Untuk melayani klien dengan lebih baik, banyak pusat data divirtualisasikan untuk menyediakan layanan cloud standar dan personal di semua tingkatan.

Contoh 8.10 Cloud *Mashup* untuk MapReduce Deteksi Spam dalam Aliran Pesan

Ide menggunakan MapReduce untuk menyaring spam dalam *mashup* dari dua platform cloud ditunjukkan pada Gambar 8.22. Di sini, kami menganggap spam yang tidak dikenal tertanam dalam aliran blog yang sah dari aplikasi Twitter. Aliran data ini bisa sangat besar, dalam rentang TB atau PB, jadi kami harus menerapkan cloud Amazon EC2 untuk menerapkan pengklasifikasi Na¨ıve Bayesian guna mendeteksi keberadaan spam dalam aliran data.

Pengklasifikasi Baysian dilatih dengan membandingkan hasil yang terdeteksi dengan beberapa sampel berlabel untuk meningkatkan akurasi klasifikasi.

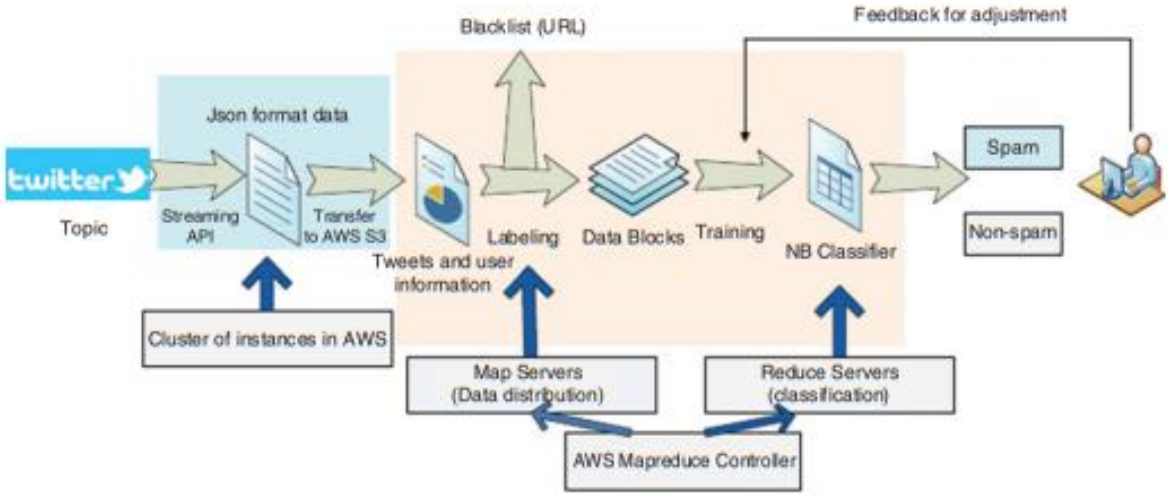
Di sisi input di ujung kiri, pemeriksaan awal tautan URL dapat menghilangkan beberapa spam yang diketahui. Aliran blog mengalir melalui mesin MapReduce dari kiri ke kanan. Kami menerapkan fungsi Map untuk memetakan file input ke instance mesin yang berbeda, dan fungsi Reduce untuk konstruksi pengklasifikasi Naive Bayesian. Pelatihan classifier Baysian dilakukan dengan menggunakan umpan balik dari hasil yang terdeteksi terhadap beberapa spam dengan label yang dikenal. Dengan aliran data input sebesar 1 TB, deteksi spam EC2 dapat dilakukan dalam waktu kurang dari 10 detik, membandingkan 1000 detik yang diperlukan untuk mendeteksi spam di komputer desktop. Akurasi deteksi yang dicapai dilaporkan lebih tinggi dari 90%.

Alat Perangkat Lunak Analisis Prediktif

Beberapa alat analisis prediktif komersial diperkenalkan di bawah ini. Alat-alat ini tidak dapat dipisahkan dalam media sosial dan aplikasi bisnis dari sumber daya *Big data*. Mereka dapat diterapkan di banyak aplikasi nyata penting yang menggunakan penambangan data, *Machine learning*, dan teknik statistik untuk mengekstrak informasi dari kumpulan data bisnis atau pemerintah. Tujuannya adalah untuk mengungkapkan pola dan tren tersembunyi dan memprediksi hasil di masa depan. Alat analitik sumber terbuka dan komersial tersedia dari perusahaan perangkat lunak besar atau kecil atau organisasi penelitian, seperti IBM, SAP, Oracle, MATLAB, SAS, Predixion, dll.

Aplikasi Analisis Prediktif

Aplikasi penting dari perangkat lunak analitik prediktif tercantum di bawah ini. Kebanyakan dari mereka terkait masalah keuangan, analisis pemasaran, perawatan kesehatan dan manajemen sosial, dll.



Gambar 8.22 Deteksi spam dari 1 TB blog Twitter di cluster EC2 menggunakan classifier Baysian (dicetak ulang dengan izin dari Y. Shi, S. Ablilash dan K. Hwang, IEEE *Mobile cloud*, 2015) [23].

Baik teknik regresi maupun *machine learning* sering diterapkan dalam mengimplementasikan aplikasi ini:

- Manajemen hubungan pelanggan analitis (CRM);
- Dukungan keputusan klinis dan prediksi penyakit;
- Deteksi penipuan, persetujuan pinjaman, dan analisis penagihan;
- Perlindungan anak, perawatan kesehatan dan perawatan lansia;
- Retensi pelanggan dan pemasaran langsung;
- Portofolio, produk atau prediksi ekonomis;
- Penjaminan emisi dan manajemen risiko.

Perangkat Lunak Komersial untuk Analisis Prediktif

Pada Tabel 8.4, kami merangkum fungsionalitas dan domain aplikasi dari lima paket perangkat lunak analitik prediktif representatif. Ini dipilih dari daftar panjang 31 paket analisis prediktif yang dilaporkan di https://www.predictiveanalyticstoday.com/what-is-predictive-analytics/

IBM menawarkan portofolio analitik prediktif yang memenuhi kebutuhan spesifik pengguna yang berbeda. Paket ini mencakup: server analitik IBM SPSS, pengumpulan, statistik, dan pemodel; manajemen keputusan analitis, analisis media sosial; dan jawaban analitik IBM. IBM SPSS Modeler menawarkan platform analitik prediktif ekstensif yang dirancang untuk menghadirkan kecerdasan prediktif pada keputusan yang dibuat oleh individu, kelompok, sistem, dan perusahaan bisnis. Solusinya menyediakan berbagai algoritme dan teknik canggih yang mencakup analitik teks, analitik entitas, manajemen keputusan, dan pengoptimalan. Statistik IBM SPSS adalah rangkaian produk terintegrasi yang menangani seluruh proses analitis, mulai dari perencanaan hingga pengumpulan data hingga analisis, pelaporan, dan penerapan.

Tabel 8.4 Lima teratas sistem perangkat lunak analitik prediktif komersial.

Nama Perangkat Lunak	**Fungsionalitas dan Domain Aplikasi**
IBM Predictive Analytics	Portofolio analitik prediktif dari IBM termasuk SPSS Modeler, Analytical Decision Management, Social Media Analytics, SPSS Data Collection, Statistics, Analytic Server, dan Analytic Answers
Analisis Prediktif SAS	SAS mendukung prediktif, pemodelan deskriptif, penambangan data, analisis teks, perkiraan, pengoptimalan, simulasi, dan desain eksperimental
Analisis Prediktif SAP	Perangkat lunak analitik prediktif SAP bekerja dengan lingkungan data yang ada serta dengan SAP Business Objects BI Platform untuk menambang dan menganalisis data bisnis, mengantisipasi perubahan

	bisnis, mendorong pengambilan keputusan yang lebih cerdas dan lebih strategis
Pembuatan GraphLab	Platform *Machine learning* dari Dato yang memungkinkan ilmuwan data dan pengembang aplikasi untuk dengan mudah membuat aplikasi cerdas dalam skala besar
Prediksi	Platform pemodelan prediktif berbasis cloud pada tahun 2010. Platform ini mendukung kemampuan analitik prediktif end-to-end mulai dari pembentukan data hingga penerapan. Model berevolusi dari perpustakaan *Machine learning* oleh Microsoft SQL Analysis Services, R dan Apache Mahout

SAP Predictive Analytics membantu untuk memahami pelanggan, menyediakan produk dan layanan yang ditargetkan, dan mengurangi risiko. Perangkat lunak ini bekerja dengan lingkungan data yang ada serta dengan SAP BusinessObjects BI Platform untuk menambang dan menganalisis data bisnis, mengantisipasi perubahan bisnis, dan mendorong pengambilan keputusan yang lebih cerdas dan lebih strategis. Mereka melakukan pemodelan prediksi intuitif, berulang atau real-time, visualisasi dan integrasi data tingkat lanjut. GraphLab Create adalah platform *Machine learning* dari Dato yang memungkinkan ilmuwan data dan pengembang aplikasi untuk dengan mudah membuat aplikasi cerdas dalam skala besar. Paket mereka menawarkan untuk membersihkan data, mengembangkan fitur, melatih model, dan membuat layanan prediktif.

Oracle Data Mining (ODM) berisi beberapa data mining dan algoritma analisis data untuk klasifikasi, prediksi, regresi, asosiasi, pemilihan fitur, deteksi anomali, ekstraksi fitur dan analitik khusus. Ini juga menyediakan sarana untuk pembuatan, pengelolaan, dan penyebaran operasional model penambangan data di dalam lingkungan basis data. Oracle Spreadsheet Add-In menyediakan operasi analitik prediktif dalam spreadsheet Microsoft Excel.

Predixion merilis platform pemodelan prediktif berbasis Cloud pertama pada tahun 2010. Predixion Insight tersedia di lingkungan Cloud publik, pribadi, atau hybrid, serta di tempat dan mendukung kemampuan analitik prediktif end-to-end yang lengkap mulai dari pembentukan data hingga penerapan. Model di Predixion dibuat dengan memanfaatkan berbagai perpustakaan *Machine learning* terintegrasi seperti Microsoft SQL Server Analysis Services, R atau Apache Mahout. Perangkat lunak SAS untuk aplikasi analitik prediktif dijelaskan dalam Contoh 8.11.

Contoh 8.11 Analisis SAS untuk Pemodelan Prediktif dan Deskriptif

SAS Predictive Analytics menyediakan paket perangkat lunak komersial untuk prediktif terintegrasi, pemodelan deskriptif, penggalian data, analisis teks, peramalan, pengoptimalan, simulasi, dan desain eksperimental. Domain aplikasi SAS Analytics mencakup analitik prediktif, penambangan data, analitik visual, peramalan, ekonometrika, dan analisis deret waktu. Paket ini

juga dapat diterapkan dalam manajemen dan pemantauan model, riset operasi, peningkatan kualitas, statistik, analitik teks, dan analitik untuk Microsoft Office.

Komponen analisis prediktif dan penambangan data membangun model deskriptif dan prediktif dan menyebarkan hasil ke seluruh perusahaan. Fungsionalitasnya meliputi analisis data eksplorasi, pengembangan dan penerapan model, penambangan data kinerja tinggi, analisis kredit, akselerasi analitik, akselerasi penilaian, dan manajemen dan pemantauan model. SAS Enterprise Miner menyederhanakan proses penambangan data untuk membuat model yang akurat. Dasbor keluaran SAS menampilkan tabel, histogram, bagan ROC, dan diagram rentang dalam melaporkan hasil prediksi.

Deteksi Komunitas di Jejaring Sosial

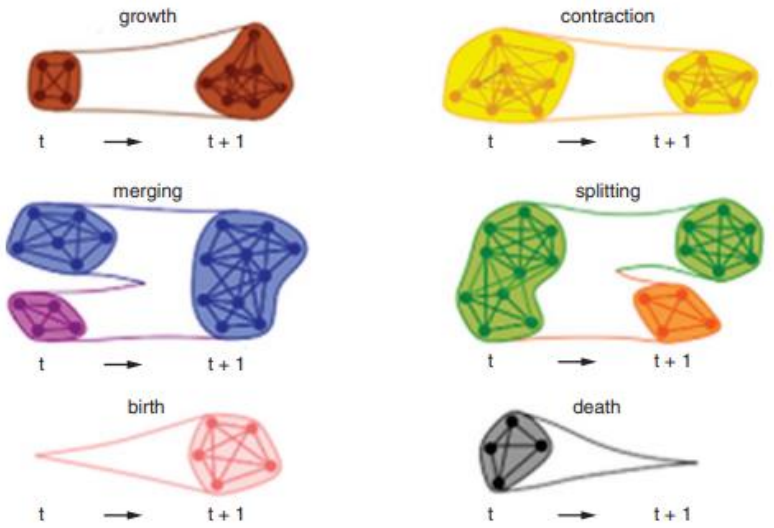
Dalam ilmu sosial, komunitas (atau cluster) dibentuk oleh sekelompok orang di bawah beberapa hubungan yang mengikat. Mendeteksi komunitas sangat penting dalam sosiologi, biologi, dan ilmu komputer. Struktur komunitas sering diwakili oleh grafik sosial. Setiap grafik sosial untuk komunitas yang terbentuk dengan baik diatur sebagai satu set node (simpul), dengan banyak tepi bergabung dengan simpul internal komunitas dan beberapa tepi yang menghubungkan ke simpul eksternal dalam grafik global asli. Komunitas-komunitas tersebut dapat bersifat terpisah-pisah atau tumpang-tindih. Komunitas yang terputus tidak berbagi node, sementara komunitas yang tumpang tindih berbagi beberapa node.

Untuk kesederhanaan, untuk menyajikan masalah deteksi komunitas dan solusinya, kami menganggap di sini hanya komunitas yang terputus-putus, di mana ada lebih banyak tepi di dalam komunitas daripada tepi yang menghubungkan ke simpul eksternal dalam grafik sosial yang sedang dipelajari. Dalam arti otonomi, komunitas adalah subgraf dengan kohesi yang lebih tinggi dengan simpul internal dan koneksi yang sangat ringan dengan sistem graf lainnya. Kami fokus pada subgraf yang mewakili komunitas dengan beberapa properti umum. Pembentukan subgraf komunitas mengikuti beberapa fungsi kesamaan di antara simpul-simpulnya. Seperti diilustrasikan pada Gambar 8.23, enam operasi graf dapat mengubah topologi graf. Seperti komunitas manusia, grafik sosial juga dapat bervariasi selama siklus hidupnya.

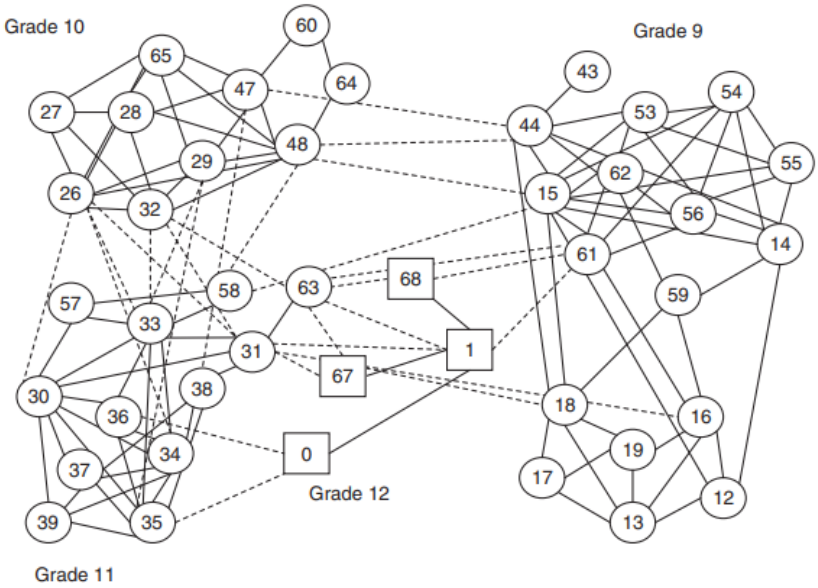
Komunitas didefinisikan sebagai subgraf mandiri dalam grafik global. Untuk analisis jaringan sosial, kami mengikuti empat properti subgraf: mutualitas lengkap, keterjangkauan, derajat simpul dan kohesi internal versus eksternal dalam mendefinisikan grafik komunitas. Kriteria global untuk mengidentifikasi komunitas dengan aturan pembentukan komunitas yang berbeda. Grafik global mungkin memiliki beberapa properti global yang dimiliki oleh komunitas tetangga. Namun, setiap subgraf komunitas mungkin memiliki aturan uniknya sendiri untuk membentuk struktur komunitas. Subgraf acak diharapkan tidak memiliki struktur seperti itu.

Deteksi komunitas mengacu pada proses mendeteksi keberadaan struktur komunitas dalam grafik sosial yang besar. Model nol digunakan untuk memverifikasi apakah grafik yang dipelajari menampilkan struktur komunitas tertentu atau tidak. Model nol paling populer sesuai dengan subgraf acak dari grafik global. Subgraf acak memiliki tepi yang dipasang ulang secara

acak. Namun, derajat simpul yang diharapkan cocok dengan grafik global. Model nol ini adalah konsep dasar di balik konsep modularitas graf asli. Sebuah grafik sosial dengan modularitas yang baik menyiratkan bahwa ia dapat dengan mudah dibagi menjadi sebanyak yang dapat dipartisi menjadi suatu fungsi, yang mengevaluasi kebaikan partisi dari grafik menjadi cluster.



Gambar 8.23 Membentuk grafik komunitas dengan bergabung, keluar, tumbuh, bergabung, membelah, dan mengecil.



Gambar 8.24 Pembentukan kelas SMA berdasarkan keanggotaan kelas siswa [22].

Modularitas memungkinkan deteksi struktur komunitas. Pengelompokan grafik sering dilakukan berdasarkan sifat modularitas. Berbagai teknik pengelompokan: pengelompokan

dasar, k-mean dan pengelompokan hierarkis (Bab 5) dapat diterapkan di sini untuk mendeteksi komunitas. Subgraf sosial adalah komunitas jika jumlah sisi di dalam subgraf melebihi jumlah subgraf acak dalam model nol. Angka yang diharapkan ini adalah rata-rata dari semua kemungkinan realisasi dari model nol. Kesamaan node adalah alami untuk mengelompokkan node untuk membentuk komunitas. Misalnya, kita dapat menghitung kesamaan antara setiap pasangan simpul dari simpul dengan beberapa kriteria yang telah ditentukan. Ukuran penting lain dari kesamaan simpul didasarkan pada sifat-sifat jalan acak pada grafik.

Contoh 8.12 Deteksi Komunitas Sekolah Menengah Atas Berdasarkan Kelas Kelas

Contoh ini menunjukkan grafik sosial sederhana untuk mengelompokkan anak-anak sekolah menengah berdasarkan kelas kelas mereka. Setiap kelas kelas disebut komunitas di sini. Masalah deteksi komunitas adalah membedakan kelas kelas berdasarkan mata kuliah yang diambil mahasiswa pada tahun yang sama. Tentu saja, ini adalah masalah deteksi komunitas yang tumpang tindih, karena beberapa siswa dapat dicap sebagai kelas 2 kelas. Grafik yang ditunjukkan pada Gambar 8.24 membagi 69 siswa ke dalam 6 kelas kelas berlabel Kelas 7 hingga Kelas 12. Ujung-ujungnya menunjukkan hubungan kelas mereka dengan mengambil mata kuliah umum yang sama. Jelas, siswa di kelas yang sama sering mengambil kursus yang sama. Oleh karena itu, ada lebih banyak koneksi tepi internal di antara mereka.

Karena perbedaan usia atau konflik penjadwalan, beberapa kursus dibagi oleh siswa di seluruh tingkat kelas yang berdekatan atau bahkan dua atau lebih tingkat terpisah karena keterlambatan kemajuan studi mereka. Ini ditunjukkan oleh koneksi tepi silang atau jarak. Tentu saja, ada lebih sedikit tepi lintas derajat daripada tepi internal dalam komunitas kelas yang sama. Ternyata siswa kelas 7 dan 12 lebih mudah dipisahkan dari yang lain. Siswa kelas 9 dan 10 memiliki lebih banyak sisi silang daripada siswa kelas lainnya. Grafik sosial ini dengan jelas menunjukkan perbedaan antara tepi internal dan eksternal yang berafiliasi dengan komunitas kelas yang berbeda. Batas antar komunitas dapat dideteksi berdasarkan koneksi yang terdistribusi di antara para siswa. Data yang diplot pada Gambar 8.24, berasal dari karya asli Xie et al., 2013.

Untuk mendeteksi afiliasi komunitas dalam grafik sosial, kami menyadari bahwa komunitas yang tidak tumpang tindih lebih mudah dideteksi daripada kasus yang tumpang tindih, dan membuat daftar tiga metode untuk mendeteksi komunitas dalam grafik sosial. Metode dibedakan berdasarkan aturan afiliasi keanggotaan yang diterapkan, yang menghasilkan tiga metode berdasarkan interaksi spin-spin, jalan acak, dan sinkronisasi:

1) **Model spin-spin:** Sebuah sistem pemintalan digunakan untuk berputar di antara q keadaan yang mungkin. Interaksinya bersifat feromagnetik, yaitu mendukung keselarasan putaran, sehingga pada suhu nol semua putaran berada dalam keadaan yang sama. Jika interaksi anti-ferromagnetik juga ada, keadaan dasar sistem mungkin bukan keadaan di mana semua putaran disejajarkan, tetapi keadaan di mana nilai putaran yang berbeda hidup berdampingan, dalam kelompok yang homogen. Dengan struktur komunitas, dan

interaksi antara spin tetangga, kemungkinan cluster struktural dapat dipulihkan dari cluster spin nilai-sama dari sistem, karena ada lebih banyak interaksi di dalam komunitas daripada di luar.

2) **Jalan-jalan acak:** Jalan-jalan acak berguna untuk menemukan komunitas. Jika graf memiliki struktur komunitas yang kuat, random walker menghabiskan waktu lama di dalam komunitas karena kepadatan tepi internal yang tinggi dan jumlah jalur yang dapat diikuti. Di sini kami menjelaskan algoritma pengelompokan paling populer berdasarkan jalan acak. Semuanya dapat secara sepele diperluas ke kasus grafik berbobot.
3) **Sinkronisasi:** Dalam keadaan tersinkronisasi, unit-unit sistem berada dalam keadaan yang sama atau serupa setiap saat. Sinkronisasi juga telah diterapkan untuk menemukan komunitas dalam grafik. Jika osilator ditempatkan pada simpul, dengan fase acak awal, dan memiliki interaksi tetangga terdekat, osilator dalam komunitas yang sama melakukan sinkronisasi terlebih dahulu, sedangkan sinkronisasi penuh membutuhkan waktu lebih lama. Jadi, jika kita mengikuti evolusi waktu dari proses, negara-negara dengan kelompok simpul yang disinkronkan dapat stabil dan berumur panjang, sehingga mudah dikenali.

Tujuan akhir dari algoritma pengelompokan adalah mencoba untuk menyimpulkan sifat dan hubungan antara simpul, yang tidak tersedia dari pengamatan/pengukuran langsung. Namun, ada juga aplikasi yang ditujukan untuk memahami sistem nyata. Beberapa hasil telah disebutkan di bagian sebelumnya. Bagian ini seharusnya memberikan gambaran tentang apa yang dapat dilakukan dengan menggunakan algoritma pengelompokan. Oleh karena itu, daftar karya yang disajikan di sini sama sekali tidak lengkap. Sebagian besar studi berfokus pada jaringan biologis dan sosial. Kami juga menyebutkan beberapa aplikasi untuk jenis jaringan lain.

Jaringan media sosial lainnya juga ada di dunia TI saat ini dan diperkenalkan secara singkat di bawah ini. Jaringan ini juga menghasilkan kumpulan *Big data* yang dapat dimasukkan ke dalam cloud dalam membuat keputusan analitik:

- **Jaringan Kolaborasi:** Dalam jaringan sosial seperti itu, individu dihubungkan bersama untuk kepentingan bersama atau kolaborasi bisnis. Kolaborasi dilakukan melalui konsep tujuan implisit kenalan. Misalnya, kita mungkin menganggap orang lain sebagai teman, sementara yang terakhir mungkin tidak setuju. Sebuah tim kolaborasi formal disatukan melalui kesepakatan atau lampiran khusus. Contoh terbaik adalah organisasi virtual yang melibatkan IBM, Apple dan Motorola dalam mengembangkan seri komputer PowerPC di masa lalu.

Analisis struktur jaringan kolaborasi ilmiah telah memberikan pengaruh besar pada pengembangan ilmu jaringan modern. Kolaborasi ilmiah dikaitkan dengan penulisan bersama. Dua ilmuwan terhubung jika mereka telah menulis bersama setidaknya satu makalah. Informasi tentang co-authorship dapat diambil dari database besar karya yang diterbitkan di berbagai bidang. Beberapa jaringan kolaborasi dilampirkan dengan cloud pribadi untuk tujuan perlindungan hak cipta intelektual.

- **Jaringan kutipan:** telah digunakan untuk memahami pola kutipan penulis dan untuk mengungkapkan hubungan antar disiplin. Rosvall dan Bergstrom telah menggunakan jaringan kutipan lebih dari 6000 jurnal ilmiah untuk mendapatkan peta sains. Mereka menggunakan teknik pengelompokan berdasarkan mengompresi informasi tentang jalan-jalan acak yang terjadi pada grafik kutipan. Jalan acak mengikuti aliran kutipan dari satu bidang ke bidang lain, dan bidang muncul secara alami dari analisis pengelompokan.
- **Jaringan legislatif:** memungkinkan kita untuk menyimpulkan hubungan antara politisi melalui aktivitas parlementer mereka, yang mungkin terkait atau tidak dengan afiliasi partai. Banyak penelitian tentang hal ini dilakukan dengan menggunakan data Perpustakaan dari Kongres AS. Mereka meneliti struktur komunitas jaringan komite di DPR AS. Komite yang berbagi anggota umum dihubungkan oleh tepi berbobot. Pengelompokan hierarki mengungkapkan hubungan erat antara beberapa komite.

Palla dkk. [19] telah memelopori studi tentang komunitas sosial yang tumpang tindih. Deteksi komunitas sosial yang berubah secara dinamis jauh lebih terlibat daripada komunitas statis atau terputus-putus. Tidak ada konsensus tentang definisi kuantitatif dari konsep komunitas yang tumpang tindih, karena itu tergantung pada metode yang diadopsi. Secara intuitif, kami berharap bahwa cluster komunitas berbagi node di perbatasan mereka. Ide ini telah menginspirasi banyak algoritma pendeteksian yang menarik. Grafik sosial dinamis yang bervariasi dengan waktu juga lebih sulit untuk dievaluasi. Ini dapat dipelajari dengan menggunakan kumpulan data yang diberi cap waktu. Melacak evolusi struktur komunitas dengan waktu sangat penting untuk mengungkap bagaimana komunitas dihasilkan dan bagaimana mereka berinteraksi satu sama lain secara dinamis.

8.5 KESIMPULAN

Banyak fungsi kognitif yang menarik dapat dibangun dengan alat *Deep learning* melalui berbagai jenis jaringan saraf tiruan. Secara khusus, kami menunjukkan penggunaan TensorFlow untuk menerapkan sistem kecerdasan kognitif di cloud saat ini. Kami juga mempelajari metode pembelajaran penguatan. Secara khusus, kami mempelajari penggunaan gabungan *Deep learning* dan pembelajaran penguatan, yang telah berhasil diterapkan di DeepMind Pertandingan AlphaGo. Analisis prediktif terbukti kuat dalam mendukung aplikasi *Big data* di jejaring sosial, seperti deteksi komunitas dan penyaringan lingkaran teman.

Tugas dan Latihan

1. Melalui penggalian data dan analisis catatan kesehatan elektronik (electronic health record (EHR), berbagai keuntungan dapat dicapai untuk penelitian medis dan kedokteran klinik di masa depan. Tolong sebutkan beberapa keuntungan seperti itu, seperti tantangan yang sesuai?

2. Merancang sistem perawatan kesehatan yang terdiri dari sensor tubuh dan perangkat yang dapat dikenakan untuk mengumpulkan sinyal fisiologis manusia. Sistem ini harus memiliki fungsi sebagai berikut: pemantauan waktu nyata, prediksi penyakit, dan deteksi dini penyakit kronis.
3. Merancang sistem pemantauan dan manajemen yang dapat mengoptimalkan distribusi sumber daya medis dan memfasilitasi pembagian data untuk sumber daya tersebut. Sebutkan beberapa fitur sistem ini dalam hal kecerdasan dan jaringan, dll.
4. Dalam beberapa tahun terakhir, analisis video telah menjadi topik hangat, terutama untuk pemeriksaan keamanan melalui pelacakan video, yang berguna untuk melindungi keselamatan pribadi dan properti. Teknologi keamanan tradisional menekankan respons waktu nyata dan efektivitas verifikasi. Jadi presentasi video dengan resolusi tinggi, tanpa kehilangan dan penundaan rendah telah menjadi arah pengembangan utama industri keamanan selama beberapa tahun terakhir. Saat ini, kita melihat kamera untuk pengawasan kota di mana-mana.

   Dengan meningkatnya penggunaan kamera definisi tinggi, cara mengirimkan data video dalam jumlah besar secara efektif telah menjadi isu utama. Selain itu, melacak penjahat untuk mendapatkan informasi lokasi mereka memakan waktu dan tenaga. Tolong jelaskan bagaimana menggunakan kecerdasan buatan dan teknologi *Machine learning* untuk menganalisis sampel video besar-besaran, secara otomatis melacak target dan menentukan jalur bergerak sesuai dengan fitur target.
5. Merancang sistem yang dapat memantau kondisi mental dan fisik pekerja yang mengoperasikan mesin berbahaya (yaitu kendaraan khusus, pesawat terbang, dan pembangkit listrik tenaga nuklir). Desain sistem seperti itu harus memanfaatkan IoT, penginderaan gambar, dan teknologi sinyal fisiologis.
6. Insiden leukemia di kalangan anak muda telah meningkat, yang membutuhkan transplantasi sel induk sebagai pengobatan wajib. Setelah transplantasi, pasien harus tinggal di rumah selama 12 hingga 24 bulan. Pendekatan tradisional mengharuskan pasien untuk mengirimkan laporan perawatan kesehatan mereka ke tim medis yang bertanggung jawab atas perawatan dan perawatan medis.

   Untuk menghindari perasaan sulit dan tidak menyenangkan pasien selama rehabilitasi, sistem video dapat dirancang untuk membantu komunikasi antara pasien dan tim medis melalui ponsel pintar, tablet, atau komputer pribadi. Sedangkan data pribadi pasien dapat dengan mudah diakses melalui sistem berbasis web. Terutama, jika elemen permainan ditambahkan ke sistem pengambilan data jarak jauh, suasana hati pasien dapat ditingkatkan selama laporan harian. Dengan data kesehatan yang lebih praktis dan sering, tim medis dapat memantau status kesehatan pasien secara lebih akurat dan tepat waktu serta memberikan pengobatan yang lebih efektif:

   1) Tentang sistem video, pernyataan mana yang benar?

a. Kami membutuhkan kerangka data yang sangat fleksibel, untuk memenuhi persyaratan parameter kesehatan khusus.
b. Sumber data eksternal ditransfer melalui bus layanan data e-health ke database.
c. Data hanya bisa sulit untuk didefinisikan, bukan definisi lunak.
d. Game diprioritaskan dengan ponsel pintar dan tablet, tetapi juga dapat dilakukan di browser web.

2) Alur kerja sistem video game mencakup tiga langkah: definisi data; membuat berkas konfigurasi; dan merencanakan tugas permainan. Saat membagikan permainan kecil kepada pasien, tugas dapat menentukan sekelompok pasien dengan praktik terapi fisik sesuai dengan status kesehatan pasien yang dievaluasi oleh tim medis. Tuliskan pendapat Anda tentang tiga langkah di atas.

7. Kekurangan sumber daya medis membuat tidak nyaman untuk menemui dokter atau mengakses fasilitas medis. Dengan berkembangnya teknologi IoT, coba pikirkan beberapa cara untuk mengatasi hal tersebut? Apakah Anda ingin membuat daftar beberapa aplikasi yang menantang di bidang ini? Tentang solusi membangun sistem pertanian cerdas, diskusikan cara menerapkan setiap solusi dengan teknologi nirkabel, sensor, dan GPS terkini:
   a) Teknologi jaringan sensor nirkabel diterapkan dalam sistem pertanian cerdas untuk mencapai pengumpulan dan pengendalian data.
   b) Rumah kaca pertanian cerdas yang dilengkapi dengan sensor nirkabel untuk memantau lingkungan seperti suhu udara/tanah, kelembapan, kelembapan, cahaya, dan konsentrasi CO2.
8. Bagian Studi 8.2.2 dan 8.2.3 untuk memahami bagaimana menggunakan DBN dan CNN untuk pengenalan angka tulisan tangan. Berdasarkan dataset Mnist dan informasi yang tercakup dalam Bagian 7.3.2, gunakan stack auto-encoder (SAE) untuk menerapkan pengenalan angka tulisan tangan. Anda perlu menentukan langkah dan kode semu SAE untuk pengenalan angka tulisan tangan, dan memberikan kode pemrograman terperinci.
9. Pengenalan angka tulisan tangan dan pengenalan wajah manusia termasuk dalam masalah klasifikasi citra. Keduanya bisa diselesaikan oleh CNN. Pertama, pelajari Bagian 7.4 dan 8.2.3 untuk memahami cara menggunakan CNN. Kemudian, pelajari Bagian 8.2.1 dan 8.2.2 untuk memahami bagaimana menggunakan CNN untuk pemahaman gambar dan analisis teks medis. Terakhir, tuliskan persamaan dan perbedaan antara klasifikasi gambar berbasis CNN dan analitik teks medis. Anda perlu mengilustrasikan detail dari aspek operasi konvolusi dan operasi pooling, masing-masing.
10. AlexNet, arsitektur jaringan yang diusulkan oleh Alex, memenangkan kejuaraan dalam ImageNet Large Visual Recognition Challenge pada tahun 2012. AlexNet adalah penyempurnaan dari model jaringan CNN yang diterapkan pada pengenalan gambar, dan struktur jaringan ini memiliki beberapa fungsi baru dan baru. Gunakan platform

TensorFlow dan wujudkan AlexNet dengan TensorFlow. Periksa situs web: https://www.tensorflow.org/ untuk mengetahui perkembangan terbaru TensorFlow. Pekerjaan rumah ini mengharuskan Anda memberikan kode program dan menunjukkan langkah-langkah membangun AlexNet untuk pengenalan digital tulisan tangan dengan TensorFlow.

DAFTAR PUSTAKA

A. Costanzo, A, Faro, D. Giordano, et al., Mobile cyber physical systems for health care: Functions, ambient ontology and e-diagnostics. Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference (CCNC), IEEE, 972-975, 2016.

A. Graves, A. Mohamed and G. Hinton, Speech recognition with deep recurrent neu- ral networks. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6645-6649, 2013.

A. Krizhevsky, I. Sutskever and G.E. Hinton, Imagenet classification with deep convolu- tional neural networks. Advances in Neural Information Processing Systems, 1097-1105, 2012.

A. Milenkovic, C. Otto and E. Jovanov, Wireless sensor networks for personal health monitoring: Issues and an implementation. Computer Communications, 29(13-14), 2521-2533, 2006.

A. Singh, et al., Incorporating temporal EHR data in predictive models for risk stratifica- tion of renal function deterioration. J. Biomed. Inform., 53, 220-228, 2015.

A. Zaslavsky, C. Perera and D. Georgakopoulos, Sensing as a service and big data. Pro- ceedings of the International Conference on Advanced. Cloud Computing (ACC), Banga- lore, India, July 2012, pp. 21-29, 2004.

A.Ranjan, A New Approach for Blind Source Separation of Convolutive Sources, ISBN 978-3-639-07797-1 (this book focuses on unsupervised learning with Blind Source Sep- aration), 2008.

B. Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and its Applications. Wiley, 2015.

B. Hutchinson, D. Li and Y. Dong, Tensor deep stacking networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1-15, 1944-1957, 2012.

B. Qian, X. Wang, N. Cao, H. Li and Y.-G. Jiang, A relative similarity based method for interactive patient risk prediction. Data. Min. Knowl. Disc., 1-24, 2014.

Bluetooth Low Energy Specification, Bluetooth Special Interest Group, http:// www.bluetooth.com

C. Bo¨hm, K Kailing, H.-P. Kriegel and P. Kro¨ger, Density connected clustering with local subspace preferences. 4th IEEE International Conference on Data Mining (ICDM'04), p. 27. 2004.

C. Lavel and Z. Callejas, Sentiment a nalysis: from opinion mining to human-agent inter- action, 2016.

C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, Context aware computing for the Internet of Things: A survey, IEEE Community Surveys Tutorials, 16(1), 414-454, 2013.

C.M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

C.O. Buckee, A. Wesolowski, N.N. Eagle, E. Hansen and R.W. Snow, Mobile phones and malaria: modeling human and parasite travel. Travel Med. Infect. Dis., 11(1):15-22, 2013.

C.W. Mundt, K.N. Montgomery, U.E. Udoh, et al., A multiparameter wearable physio- logic monitoring system for space and terrestrial applications. Proceedings of the IEEE Transactions on Information Technology in Biomedicine, 9(3), 382-391, 2005.

D, Gardner and G.M. Shepherd, A gateway to the future of neuroinformatics. Neuroin- formatics, 2(3), 271-274, 2004.

D. Oliver, F. Daly, F.C. Martin and M.E. McMurdo Risk factors and risk assessment tools for falls in hospital in-patients: a systematic review. Age Ageing, 33, 122-130, 2004.

D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, et al., Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489, 2016.

D. Terdiman, IBM's TrueNorth processor mimics the human brain, http://www.cnet.com/news/ibms-truenorth-processor-mimics-the-human-brain/,2014.

D.E. Goldberg and J.H. Holland, Genetic algorithms and machine learning. Machine Learning, 3(2), 95-99, 1988.

D.J. MacKay, Information Theory, Inference, and Learning Algorithms. Cambridge Uni- versity Press, 2003.

D.W. Bates, S. Saria, L. Ohno-Machado, A. Shah and G. Escobar, Big data in health care: using analytics to identify and manage high-risk and high-cost patients. Health. Affairs, 33, 1123-1131, 2014.

Ding and X. He, K-means clustering via Principal Component Analysis. Proceedings of the International Conference on Machine Learning (ICML 2004), 225-232, July 2004.

E. Alpaydin, Introduction to Machine Learning. The MIT Press, 2010.

E. Cambria, Affective computing and sentiment analysis. IEEE Intelligent Systems, 31(2), 102-107, 2016.

E. Farella, A. Pieracci, L. Benini, L. Rochi and A. Acquaviva, Interfacing human and com- puter with wireless body area sensor networks: The WiMoCA solution. Multimedia Tools and Applications, 38(3), 337-363, 2008.

E. Frias-Martinez, G. Williamson and V. Frias-Martinez. An agent-based model of epi- demic spread using human mobility and social network information. Proceedings of the IEEE

Third International Conference on Privacy, Security, Risk and Trust. Boston: IEEE. 57-64, 2011.

E.A. Lee, Cyber physical systems: design challenges. IEEE International Symposium on Object Oriented Real Time Distributed Computing, 363-369, May 2008.

F. Gers, N. Schraudolph and J. Schmidhuber, Learning precise timing with LSTM recur- rent networks. Journal of Machine Learning Research, 3, 115-143, 2002.

F. Zhang, K. Hwang, S. Khan, and Q. Malluhi, Skyline discovery and composition of multi- cloud mashup services, IEEE Transactions on Service Computing, September 2016.

FisA.cher and C. Igel, Training restricted Boltzmann machines: an introduction. Pattern Recognition, 47(1), 25-39, 2014.

G. Castellano, L Kessous and G. Caridakis, Emotion Recognition through Multiple Modal- ities: Face, Body Gesture, Speech. Affect and Emotion in Human-Computer Interaction. Springer, Berlin Heidelberg, 92-103, 2008.

G. Hinton and R. Salakhutdinov, Efficient learning of deep Boltzmann machines. 3, 448- 455, 2009.

G. Lo, A. Suresh, S. Gonzalez-Valenzuela, L. Stocco and V.C.M. Leung, A wireless sen- sor system for motion analysis of Parkinson's disease patients. Proceedings of the IEEE PerCom, Seattle, WA, March 2011.

G. Palla, I. Derenyi, I. Farkas and T. Vicsek. Uncovering the overlapping community struc- ture of complex networks in nature and society. Nature, 435(7043), 814, 2005.

G.A. Hinton, A practical guide to training restricted Boltzmann machines. Momentum, 9(1), 926, 2010.

G.A. Miller, The cognitive revolution: a historical perspective. Trends in Cognitive Sci- ences 7: 141-144, 2004.

Google Cloud BigQuary: https://cloud.google.com/bigquery/

Google Cloud Datalab: https://cloud.google.com/datalab/

H. Cao, H. Li, L. Stocco and V.C.M. Leung, Wireless three-pad ECG system: Challenges, design and evaluations. Journal of Communications and Networks, 13(2), 113-124, 2011.

H. Chaouchi, The Internet of Things. Wiley, 2010.

H. Karau, et al., Learning Spark: Lightning Fast Data Analysis. O'Reily, 2015.

H. Lee, R. Grosse, R. Ranganath and A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ICML, 609-616, 2009.

H. Li and J. Tan, Heartbeat driven medium access control for body sensor networks. Pro- ceedings of the ACM SIGMOBILE, San Juan, Puerto Rico, 2007.

H. Mannila, Data mining: machine learning, statistics, and databases. IEEE International Conference on Scientific and Statistical Database Management, 1966.

H. Trevor, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data min- ing, inference and prediction. New York: Springer. pp. 485-586, 2009.

H.C. Chao, S. Zeadally and B, Hu, Wearable computing for healthcare. Journal of Medical Systems, 40(4), 1-3, 2016.

H.P. Kriegel, P. Kro¨ger and A. Zimek, Clustering high-dimensional data. ACM Transac- tions on Knowledge Discovery from Data, 3(1), 2009.

Hype Cycle, http://www.gartner.com/newsroom/id/2819918

I. Arel, D.C. Rose, and T.P. Karnowski, Deep machine learning - a new frontier in artificial intelligence research - a survey paper by IEEE Computational Intelligence Magazine, 2013.

I. Jantunen et al., Smart sensor architecture for mobile-terminal-centric ambient intelli- gence. Sensors and Actuators A: Physical, 142(1), 352-360, 2004.

I. Szita, and S. Csaba, Model-based Reinforcement Learning with Nearly Tight Exploration Complexity Bounds (PDF). ICML, Omnipress, pp. 1031-1038, 2010.

I. Tabas and C.K. Glass, Anti-inflammatory therapy in chronic disease: challenges and opportunities. Science, 339(6116), 166-172, 2013.

I.Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 664 pp., 2011.

J. Broekens, T. Bosse and S.C. Marsella, Challenges in computational modeling of affec- tive processes. IEEE Transactions on Affective Computing, 4(3), 242-245, 2013.

J. Cao, K. Hwang, D. Li and A. Zomaya, Optimal multiserver configuration for profit maximization in cloud computing. IEEE Trans. Parallel and Distributed Systems, July 2013.

J. Dean, Large-scale Deep Learning for Intelligent Computer Systems, Slide Presentation, 2016

J. Gobbi, R. Buyya, S. Marusic and M. Palaniswarni, Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29, 1645-1660, 2013.

J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, Third Edition. Morgan Kaufmann, 2012.

J. Kelley, III, Computing, Cognition and The Future of Knowing, IBM Corp, October 2015.http://www.research.ibm.com/software/IBMResearch/multipmedia/Computing_Cognition_WhitePaper.pdf

J. MacQueen, Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium in Mathematical Statistics and Probability, 1(14): 281-297, 1967.

J. Ratsaby and S. Venkatesh, Learning from a mixture of labeled and unlabeled exam- ples with parametric side information. Proceedings of the Eighth Annual Conference on Computational Learning Theory, 412-417, 1995.

J. Schmidhuber, Deep learning in neural networks: An overview. Neural Networks, 61, 85-117, 2015.

J. Schmidhuber, Learning complex, extended sequences using the principle of history compression. Neural Computation, 4(2), 234-242, 1992.

J. Smith and R. Nair, Virtual Machines: Versatile Platforms for Systems and Processes. Morgan Kaufmann, 2005.

J. Wan, C. Zou, S. Ullah, et al., Cloud-enabled wireless body area networks for pervasive healthcare. Network, IEEE, 27(5), 56-61, 2013.

J. Xie, et al., Overlapping community detection in networks. ACM Computing Survey, August 2013.

J. Xie, S. Kelley and B.K. Szymanski, 2013. Overlapping community detection in net- works: The state-of-the-art and comparative study. ACM Comput. Surv., 459(4), Article 43, August 2013.

J. Zhang, et al., Evolutionary computation meets machine learning: A survey. IEEE Com- putational Intelligence Magazine, 6(4), 68-75, 2011

J.C. Ho, C.H. Lee and J. Ghosh, Septic shock prediction for patients with missing data. ACM Transactions on Management Information Systems (TMIS), 5(1), 2014.

J.R. Quinlan, C4. 5: programs for machine learning. Elsevier, 2014.

K. Chen and D. Ran, C-RAN: The Road towards Green RAN. White Paper, China Mobile Research Institute, Beijing, October 2011.

K. Hwang and D. Li, Trusted cloud computing with secure resources and data coloring. IEEE Internet Computing. October 2010.

K. Hwang, G. Fo and J. Dongarra, Distributed and Cloud Computing. Mogan Kaufmann, 2012.

K. Hwang, G. Fox and J. Dongarra, Cloud Computing for Big Data Applications: A Hadoop, Spark and TensorFlow Approach. Morgan Kaufmann Publisher, 2017.

K. Hwang, G. Fox and J. Dongarra, Distributed and Cloud Computing. Morgan Kaufamann, 2011,

K. Hwang, S. Yue and X. Bai, Scale-out and scale-up techniques for cloud performance and productivity. IEEE CloudCom, Workshop on Emerging Issues in Clouds. Singapore, December 18, 2014.

K. Hwang, X. Bai, Y. Shi, M.Y. Li, W.G. Chen and Y.W. Wu, Cloud performance modeling with benchmark evaluation of elastic scaling strategies, IEEE Transactions on Parallel and Distributed Systems, January, 2016.

L. Barroso and U. Holzle, The datacenter as a computer: An introduction to the design of warehouse-scale machines. In Synthesis Lectures on Computer Architecture, edited by M. Hill (ed.). Morgan Claypool, 2009.

L. Bengtsson, X. Lu, A. Thorson, R. Garfield and J. von Schreeb, Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in Haiti. PLoS Med., 8(8), 2011.

L. Breiman, Bagging predictors. Machine Learning, 24(2), 123-140, 1996.

L. Breiman, J. Friedman, C.J. Stone, et al., Classification and Regression Trees. CRC press, 1984.

L. Breiman, Random forests. Machine Learning, 45(1), 5-32, 2001.

L. Deng and D. Yu, Deep Convex Net: A scalable architecture for speech pattern classi- fication (PDF). Proceedings of Interspeech, 2285-2288, 2011.

L. Deng, and D. Yu, Deep learning: methods and applications. Foundations and Trends in Signal Processing, 7: 3-4, 2014.

L.G. Jaimes, J. Calderon, H. Lopez, et al., Trends in mobile cyber-physical systems for health just-in time interventions. SoutheastCon, IEEE, 1-6, 2015.

L.P. Kaelbling, M.L. Littman and A.W. Moore, Reinforcement learning: a survey. Journal of Artificial Intelligence Research, 4, 237-285, 1996.

M. Chen, Big Data Related Technologies. Springer Computer Science Series, 2014.

M. Chen, NDNC-BAN: Supporting Rich Media Healthcare Services via Named Data Networking in Cloud-assisted Wireless Body Area Networks. Information Sciences, 284(10), 142-156, Nov. 2014.

M. Chen, S. Gonzalez, A. Vasilakos, H. Cao and V. Leung, Body area networks: A survey. ACM/Springer Mobile Networks and Applications (MONET), 16(2), 171-193, 2010.

M. Chen, Y. Ma, S. Ullah, et al., ROCHAS: robotics and cloud-assisted healthcare sys- tem for empty nester. Proceedings of the 8th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommu-nications Engineering), 217-220, 2013.

M. Chen, Y. Ma, Y. Hao, Y. Li, D. Wu, Y. Zhang and E. Song, CP-Robot: Cloud-assisted pillow robot for emotion sensing and interaction. ICST IndustrialIoT, 2016.

M. Chen, Y. Zhang, Y. Li, et al., AIWAC: affective interaction through wearable comput- ing and cloud technology. Wireless Communications, IEEE, 22(1), 20-27, 2015.

M. Chen, Y. Zhang, Y. Li, et al., EMC: emotion-aware mobile cloud computing in 5G Network. IEEE, 29(2), 32-38, 2015.

M. Ester, H.P. Kriegel, J. Sander, et al., A density-based algorithm for discovering clusters in large spatial databases with noise. 96(34), 226-231, 1996

M. Hilber and P. Lopez, The world's technological capacity to store. Communicate and compute information. Science, 332(6025), 2011.

M. Mohri, A. Rostamizadeh and A. Talwalkar, Foundations of Machine Learning. MIT Press, 2012.

M. Rosenblum and T. Garfinkel, Virtual machine monitors: Current technology and future trends. IEEE Computer, May 2005, 39-47.

M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, The case for VM-based cloudlets in mobile computing. IEEE Pervasive Computing, 8(4), 14-23, 2009.

M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, 5G-enabled tactile internet. IEEE Journal on Selected Areas in Communications, 34(3), 460-473, 2016.

M. Soleymani, S. Asghari Esfeden, Y. Fu, et al., Analysis of EEG signals and facial expres- sions for continuous emotion detection, 2015.\

M.B. Eisen, P.T. Spellman, P.O. Brown, et al., Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences, 95(25), 14863-14868, 1998.

M.M. Rodgers, P.V. Pai and R.S. Conroy. Recent advances in wearable sensors for health monitoring. Sensors Journal, IEEE, 15(6), 3119-3126. 2015.

Mark Weiser, The computer for the 21st century. Scientific American, 1991.

N. Boulanger-Lewandowski, Y. Bengio and P. Vincent, Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and tran- scription. preprint arXiv:1206.6392, 2012.

N. Jardine and R. Sibson, Mathematical taxonomy. London, John Wiley, 1971.

N. Jouppi, Google Supercharges Machine Learning Tasks with TPU Custom Chip. Google Cloud Platform Blog, Google. May 18, 2016.

N. Torabi and V.C.M. Leung, Robust license-free body area network access for reliable public m-health services. Proceedings of the IEEE HealthCom, Columbia, MO, June 2011.

N. Yigitbasi, A. Iosup, D. Epema and S. Ostermann, C-Meter: A framework for perfor- mance analysis of computing clouds, IEEE/ACM Proceedings of the 9th International Symposium on Cluster Computing and the Grid, (CCGrid), 2009.

O. Chapelle, B. Scho¨lkopf, and A. Zien, Semi-supervised Learning. Cambridge, MA, MIT Press. 2006.

P. Abouzar, K. Shafiee, D. Michelson and V.C.M. Leung, Action-based scheduling tech- nique for 802.15.4/ZigBee wireless body area networks. IEEE PIMRC, Toronto, ON, September 2011.

P. Drineas, P.A. Frieze, R. Kannan, S. Vempala and V. Vinay, Clustering large graphs via the singular value decomposition. Machine Learning, 56, 9-33. Retrieved 2012-08-02, 2004.

P. Groves, B. Kayyali, D. Knott and S. Van Kuiken, The "big data" revolution in healthcare. McKinsey Quarterly, 2013.

P. Merolla, et al., A million spiking neuron integrated circuit with a scalable network and interface. Science, 345(6197), 2014.

P.B. Jensen, I.J. Jensen and S. Brunak, Mining electronic health records: towards better research applications and clinical care. Nat. Rev. Gene, 13, 395-405. 2012.

P.T. Langley, The changing science of machine learning. Machine Learning, 82(3), 275- 279, 2011.

R, Liu, Introduction to Internet of Things. Science Press, Beijing, 2011.

R. Agrawal and J.C. Shafer, Parallel mining of association rules. IEEE Transactions on Knowledge & Data Engineering, 6, 962-969. 1996.

R. Agrawal, T. Imielin´ski and A. Swami, Mining association rules between sets of items in large databases. ACM SIGMOD Record, 22(2), 207-216, 1993.

R. Buyya, J. Broberg and A. Goscinski (eds), Cloud Computing: Principles and Paradigms. Wiley Press, US, February 2011.

R. Chalasani and J. Principe, Deep predictive coding networks. 1-13. 1301-3541, 2013.

R. Collobert, Deep learning for efficient discriminative parsing. Proceedings of the Four- teenth International Conference on Artificial Intelligence and Statistics (AISTATS), 224- 232, 2011.

R. Costa, D. Carneiro, P. Novais, et al., Ambient assisted living. Proceedings of the 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008. Springer, Berlin Heidelberg, 86-94, 2009.

R. Kohavi and F. Provost, Glossary of terms. Machine Learning, 30: 271-274, 1998.

R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection. 4(2), 1137-1145, 1995.

R. Salakhutdinov, A. Mnih and G. Hinton, Restricted Boltzmann machines for collabora- tive filtering. ACM Proceedings of the 24th International Conference on Machine Learn- ing, 791-798, 2007.

R. Salakhutdinov, J.B. Tenenbaum and A. Torralba, Learning with hierarchical-deep models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1958- 1971, 2013.

R. Socher, et al., Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the IEEE Conference on Empirical Methods in Natural Language Processing, 2013.

R. Socher, J. Pennington and E.H. Huang, et al., Semi-supervised recursive AutoEncoders for predicting sentiment distributions. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 151-161, 2011.

R. Sutton and A. Barto, Reinforcement Learning: An Introduction. MIT Press, 1998.

R.S. Basu, et al., Dynamic hierarchical classification for patient risk-of-readmission. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1691-1700, 2015.

S. Gonza´lez-Valenzuela, M. Chen and V.C.M. Leung, Mobility support for health monitoring at home using wearable sensors. IEEE Trans. Information Technology in BioMedicine, 15(4), 539-549, 2011.

S. Bandyopadhyay, et al., Data mining for censored time-to-event data: A Bayesian net- work model for predicting cardiovascular risk from electronic health record data. Data. Min. Knowl. Disc., 1-37, 2014.

S. Farnham, The Facebook Association Ecosystem. O'Reilly Radar Report, 2008.

S. Jerritta, M. Murugappan, K. Wan, et al., Emotion detection from QRS complex of ECG signals using Hurst Exponent for different age groups. Humane Association Conference on Affective Computing and Intelligent Interaction (ACII), IEEE, 849-854, 2013.

S. Liu, T. Chen, et al., Cambricon: An instruction set architecture for neural networks. Proceedings of the 43rd ACM/IEEE International Symposium on Computer Architecture (ISCA'16), 2016.

S. Marcoon, A.M. Chang, B. Lee, R. Salhi and J.E. Hollander, Heart score to further risk stratify patients with low TIMI scores. Critical Pathways in Cardiology, 12, 1-5, 2013.

S. Murali, F.J. Rincon Vallejos and D.A. Atienza Alonso, Wearable device for physical and emotional health monitoring. Computing in Cardiology. 42(EPFL-CONF-213467): 121-124, 2015.

S. Pentland, Healthwear: Medical technology becomes wearable. IEEE Computer, 37(5), 42-49, 2004.

S. Rifai, Y. Bengio and A. Courville, et al., Disentangling Factors of Variation for Facial Expression Recognition. Computer Vision-ECCV. Springer Berlin Heidelberg, 808-822, 2012.

S. Ryza, et al., Advanced Analytics with Spark. O'Reily, 2015.

S. Wold, K. Esbensen and P. Geladi, Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 2(1-3), 37-52, 1987.

S. Zhao, Affective computing of image emotion perceptions. Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 703-703, 2016.

S.J. Bradtke, S.J. Andrew and G. Barto, Learning to predict by the method of temporal differences. Machine Learning, Springer, 22, 33-57, 1996.

T. Chen and Z Du, et al., DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. Proceedings of 19th International Conference on Architec- tural Support for Programming Languages and Operating Systems (ASPLOS'14), 2014.

T. Hey, S. Tansley and K. Tolle (eds), The Fourth Paradigm: Data-Intensive Scientific Dis- covery. Microsoft Research, 2009.

T. Kohonen, The self-organizing map. Proceedings of the IEEE, 78(9), 1464-1480, 1990.

T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representa- tions in vector space. Proceedings of Workshop at International Conference on Learning Representations, 2013.

T. Mitchell, Machine Learning. McGraw Hill, 1997.

The Oxford Handbook of Affective Computing. Oxford University Press, US, 2014.

TinyOS Website, available at http://tinyos.net/,2011.

U. Hansmann, et al., Pervasive Computing: The Mobile World, Second Edition. Springer, 2003.

V. Mnih, et al., Human-level control through deep reinforcement learning. Nature, 518, 529-533, 2015.

V. Pascal, L. Hugo and L. Isabelle, et al., Stacked Denoising AutoEncoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, 2010.

V. Vapnik, Statistical Learning Theory. Wiley-Interscience, 1998.

V. Vapnik, The Nature of Statistical Learning Theory. Springer Science & Business Media,2013.

W.S. Torgerson, Theory and Methods of Scaling, 1958.

X. Chen, KATZLDA: Katz measure for the IncRAN-disease association prediction. Sci. Rep-UK, 5, 2015.

X. Zhu and A. Goldberg, Introduction to Semi-Supervised Learning. Morgan & Claypool, 2009.

Y, Freund and R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139, 1997.

Y. Bengio, A. Courville and P. Vincent, Representation learning: A review and new per-spectives. Pattern Analysis and Machine Intelligence, IEEE Transactions, 35(8), 1798-1828, 2013.

Y. Bengio, et al., Representation learning: A review and new perspectives. Pattern Anal- ysis and Machine Intelligence, 35(8), 1798-1828, 2013.

Y. Bengio, Learning deep architectures for AI. Foundations and Trends in Machine Learn- ing, 2(1): 1-127, 2009.

Y. Bengio, Y. LeCun and G.E Hinton, Deep learning. Nature, 521: 436-444, 2015.

Y. LeCun, L. Botto, Y. Bengio, et al., Gradient-based learning applied to document recog- nition. Proceedings of the IEEE, 86(11), 2278-2324, 1998.

Y. Li, and W. Wang, Can mobile cloudlets support mobile applications? IEEE INFOCOM, April 2014, 1060-1068.

Y. Shi, S. Abhilash and K. Hwang, Cloudlet mesh for securing mobile clouds from intru- sions and network attacks. IEEE Mobile Computing, San Francisco, April 2, 2015.

Y. Sun, X. Wang and X. Tang, Deep learning face representation from predicting 10,000 classes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1891-1898, 2014.

Z. Huang, W. Dong and H. Duan. A probabilistic topic model for clinical risk stratification from electronic health records. J. Biomed. Informa., 58, 28-36, 2015.

Zaslavsky, C. Perera and D. Georgakopoulos, Sensing as a service and big data. Pro- ceedings of the International Conference Advanced Cloud Computing (ACC), Bangalore, India, 21-29, July 2012.

ZigBee Specification, ZigBee Alliance, 2005, http://www.zigbee.org

Dr. Budi Raharjo, S.Kom., M.Kom., MM.

Ilmu Big Data dan Mesin Cerdas

BIODATA PENULIS

Dr. Budi Raharjo, S.Kom, M.Kom, MM lahir di Semarang, tanggal 22 Februari 1985. Beliau adalah Alumni dari Universitas Bina Nusantara (BINUS University) Jakarta dan juga alumni Universitas Kristen Satya wacana (UKSW) Salatiga. Dr. Budi Raharjo telah menjadi Dosen pada Universitas STEKOM pada mata kuliah Kepemimpinan (Leadership), mata kuliah Pengantar Akuntansi, Manajemen Proses, Manajemen Akuntansi dan Manajemen Resiko Bisnis. Selain sebagai dosen Universitas STEKOM, Dr. Budi Raharjo, M.Kom, MM juga mempunyai bisnis sendiri dalam bidang perhotelan dan juga sebagai wirausaha dalam bidang pemasok unggas (ayam) beku, ke berbagai kota besar, khususnya Jakarta dan sekitarnya.

Pengalaman beliau berwirausaha menjadi bekal utama dalam penulisan buku ajar yang diterbitkan oleh Yayasan Prima Agus Teknik (YPAT) Semarang. Oleh sebab itu bukunya berisi langkah langkah praktis yang mudah diikuti oleh para mahasiswa, saat mahasiswa mengikuti proses perkuliahan pada Universitas Sains dan Teknologi Komputer (Universitas STEKOM). Jabatan struktural yang di embannya saat ini adalah Wakil Rektor 1 (Akademik) Universitas STEKOM Semarang.

YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Dr. Budi Raharjo, S.Kom., M.Kom., MM.

Ilmu Big Data dan Mesin Cerdas

YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

YAYASAN PRIMA AGUS TEKNIK