

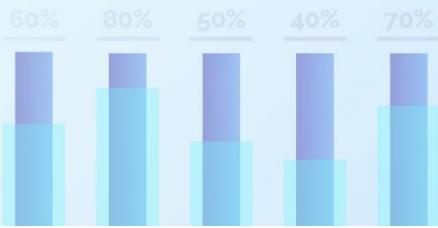
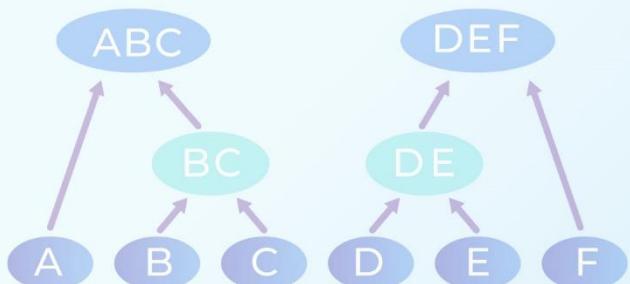
Pengantar **DATA SCIENCE** dan **APLIKASINYA** bagi **Pemula**

Paparan yang mudah dipahami oleh pemula untuk mendapatkan pengetahuan awal tentang teknik-teknik Data Science, Big Data, dan aplikasi-aplikasinya

Oleh:
Program Data Science

Editor:

Veronica S. Moertini
Mariskha T. Adithia



Copy right:

Jurusan Teknik Informatika
Fakultas Teknologi Informasi dan Sains
Universitas Katolik Parahyangan (UNPAR)
Bandung, Indonesia

ISBN: dalam proses pengurusan

Konten buku ini menjadi milik/hak dari Jurusan Teknik Informatika, Universitas Katolik Parahyangan. Keseluruhan maupun bagian-bagian dari konten buku ini tidak diijinkan untuk dipublikasikan pihak lain. Pengutipan konten buku pada artikel di website, artikel pada seminar, atau jurnal dan buku harus mengikuti ketentuan pengutipan (dari daftar pustaka) sesuai ketentuan yang berlaku.

Buku elektronik (dalam format PDF) tersedia untuk diunduh gratis dari:
<https://informatika.unpar.ac.id/buku-pengantar-data-science-dan-aplikasinya/>

Jurusan Teknik Informatika UNPAR tidak mengijinkan file PDF buku ini dipublikasikan di website lain maupun media publikasi dalam bentuk apapun (misalnya aplikasi medsos) lainnya.

Versi Beta, Oktober 2020

Dipersembahkan kepada



Halaman ini sengaja dikosongkan

Daftar Isi

Daftar Isi	i
Kata Pengantar	v
Sambutan Rektor Unviersitas Katolik Parahyangan	vii
Data Science bagi Indonesia	ix
Bagian Pertama	xi
Bab 1 Data Science dan Data Scientist	1
1.1. Data Abad 21	1
1.2. Apa itu Data Science?	3
1.3. Apa saja yang Dikerjakan Data Scientist?	4
1.4. Keahlian dan Skill Data Scientist	9
1.5. Era Industri 4.0 dan Data Science	13
1.6. Kebutuhan Data Science	15
1.7. Informasi Bab-bab Buku	16
Referensi	17
Bab 2 Menjelang Ujian: Ngebut Belajar atau Tidur?	19
2.1. Pendahuluan	19
2.2. Konsep Statistika	21
2.3. Pengumpulan Data dari Peserta Kuliah	27
2.4. Hasil Analisis Data	28
2.5. Kesimpulan	33
Referensi	34
Bab 3 Pengenalan Sistem Rekomendasi pada e-Commerce	35
3.1. Pendahuluan	35
3.2. Sistem Rekomendasi dan Collaborative Filtering	37
3.3. Data e-Commerce	40
3.4. Studi Kasus	43

3.5. Penutup	47
Referensi	47
Bab 4 Pencarian Keterkaitan Bahan Masakan dengan Teknik <i>Clustering</i>	49
4.1. Pendahuluan	49
4.2. Teknik <i>Hierarchical Clustering</i>	51
4.3. Data Resep Masakan	54
4.4. Studi Kasus	56
4.5. Penutup	61
Referensi	61
Bab 5 Analisis Data Penginderaan Jauh Satelit, Kasus: Prediksi Panen Padi	63
5.1. Pendahuluan	63
5.2. Data Penginderaan Jauh Satelit	64
5.3. Analisis Data Satelit SPOT-4 untuk Prediksi Panen Padi	66
5.4. Penutup	73
Referensi	74
Bab 6 Penggalian Insights dari Data COVID-19 dengan Visualisasi, Studi Kasus: Data Korea Selatan	75
6.1. Pendahuluan	75
6.2. Data COVID-19 di Korea Selatan	77
6.3. Bentuk-bentuk Visualisasi	78
6.4. Penggalian Insights	79
6.5. Penutup	95
Referensi	96
Bab 7 Prediksi Kualitas Tidur dari Data Wearable Device	97
7.1. Pendahuluan	97
7.2. Wearable Device	98
7.3. Konsep Dasar	100
7.4. Klasifikasi Data Wearable Device	105
7.5. Penutup	113
Referensi	114
Bab 8 Rekomendasi Film dengan Fuzzy Collaborative Filtering	115
8.1. Pendahuluan	115

8.2. User-based Collaborative Filtering	118
8.3. Algoritma Clustering Fuzzy c-Means	121
8.4. Hasil Penelitian Rekomendasi Film dengan Fuzzy Collaborative Filtering	125
8.5. Penutup	127
Referensi	128
Bab 9 Urun Daya Data Kepadatan Lalu Lintas	129
9.1. Pendahuluan	129
9.2. Pengukuran Kepadatan Lalu Lintas oleh Google Maps	130
9.3. Pemanfaatan Google Traffic untuk Penentuan Waktu Pergi dan Pulang	135
Referensi	139
Bagian Kedua	141
Bab 10 Teknologi Big Data	143
10.1. Pendahuluan	143
10.2. Seputar Big Data	143
10.3. Arsitektur Teknologi Big Data	148
10.4. Ekosistem Hadoop	150
10.5. Teknologi Big Data Komersial	155
10.6. Contoh Penggunaan Teknologi Big Data	160
10.7. Kesimpulan	161
Referensi	161
Bab 11 Pengumpulan Data Twitter dengan Teknologi Big Data	163
11.1. Pendahuluan	163
11.2. Studi Literatur	164
11.3. Pengumpul Data Twitter dengan Spark Streaming	174
11.4. Pengumpul Data Twitter dengan Kafka	179
11.5. Kesimpulan	184
Referensi	184
Bab 12 Algoritma Pengelompokan k-Means Paralel untuk Memproses Big Data	187
12.1. Pengelompokan Data	187
12.2. Manfaat Analisis Klaster	188
12.3. Algoritma Pengelompokan k-Means Non-Paralel	188
12.4. Algoritma k-Means Paralel untuk Big Data	193

12.5. Pengembangan Algoritma k-Means Paralel	198
12.6. Penutup	204
Referensi	205
Bab 13 Estimasi Dimensi Tubuh Manusia dengan Kinect	207
13.1. Pendahuluan	207
13.2. Microsoft Kinect	208
13.3. Principal Component Analysis	210
13.4. Regresi Linier	211
13.5. Metode Estimasi Dimensi Tubuh dan Hasilnya	212
13.6. Pembangunan Perangkat Lunak	217
13.7. Hasil Eksperimen	218
13.8. Kesimpulan	221
Referensi	221
Bab 14 Segmentasi Citra Menggunakan Algoritma <i>Particle Swarm Optimization</i>	223
14.1. Pendahuluan	223
14.2. Studi Literatur	225
14.3. Segmentasi Gambar dengan Algoritma PSO dan <i>K-means</i>	230
14.4. Eksperimen Segmentasi Gambar	233
14.5. Kesimpulan	237
Referensi	237
Biografi Editor dan Para Pengarang	239
Program Data Science UNPAR	241

Kata Pengantar

Pertama-tama kami panjatkan puji syukur kepada Tuhan YME. Berkat karunia kesehatan, kemampuan bekerja dan berkah melimpah dariNya, pada akhirnya kami berhasil menyelesaikan buku ini.

Berdasar hasil survei (secara terbatas) ke lingkungan sekolah menengah atas (SMA) dan masyarakat di Indonesia, kami mendapati bahwa mayoritas dari mereka belum mengenal Data Science. Padahal, para peneliti, penentu kebijakan dan praktisi pada berbagai bidang di dunia sudah mengakui bahwa pada era Industri 4.0 ini, Data Science merupakan salah satu bidang yang penting. Data scientist sedang dan diprediksi akan banyak dibutuhkan di semua bidang (industri, ritel, jasa, pariwisata, pendidikan, dll.). Ulasan lebih rinci tentang hal-hal tersebut kami paparkan pada Bab 1, Subbab 1.6. Di sini, kami ingin menggaris-bawahi hal ini: Pada laporan *Global Skills Index 2020* yang diterbitkan oleh Coursera (penyelenggara kursus daring global), untuk bidang Data Science Indonesia ditempatkan pada posisi *lagging* atau tertinggal. Dari 60 negara (di benua Amerika, Eropa, Asia, Afrika dan Australia) yang ditelaah, Indonesia berada di posisi 56. Untuk mengejar ketertinggalan, kiranya sudah jelas bahwa penyiapan SDM di bidang Data Science perlu digenjot.

Tak kenal maka tak sayang. Kami duga pepatah jadul, warisan nenek moyang kita, itu masih berlaku pada jaman *now*. Para lulusan SMA, mahasiswa/i, praktisi dan masyarakat umum perlu mendapatkan informasi seputar Data Science yang memadai dan mereka pahami. Harapannya tentu saja agar mereka tertarik, lalu bersedia untuk menekuni ilmu dan meningkatkan skill di bidang Data Science. Buku ini dimaksudkan untuk menjawab kebutuhan ini. Kami berpendapat, jika masyarakat mengetahui “indahnya” Data Science, maka mereka akan tertarik untuk menekuni bidang ini. Dengan demikian, di masa depan kebutuhan profesional di bidang Data Science di Indonesia dapat dipenuhi.

Para dosen penulis bab-bab buku ini menyadari bahwa sama sekali tidak mudah untuk menjelaskan kepada masyarakat umum tentang apa itu Data Science, terlebih lagi untuk memberikan impresi bahwa Data Science beserta teknik-tekniknya itu “indah”, menarik untuk dipelajari lebih lanjut. Namun demikian, kami berupaya keras agar bab-bab dalam buku ini, setelah dibaca, dapat membuat para pembaca paham dan terkesan. Sebagai upaya untuk memberikan gambaran yang lebih jelas tentang apa itu Data Science, pada Bagian Pertama, kami memaparkan aplikasi Data Science pada beberapa contoh kasus yang variatif dan dengan bahasan yang sederhana dan mudah dipahami. Selain itu, bagi para pembaca yang tertarik untuk mempelajari konten yang lebih teknis, pada Bagian Kedua, kami menyajikan contoh-contoh hasil penelitian dosen dan mahasiswa yang terkait dengan big data dan Data Science.

Secara khusus, berikut ini kelompok pembaca yang kami sasar:

- Bagian Pertama: Para siswa/i SMA, orang tua murid, mahasiswa/i dan publik yang sedang mencari informasi tentang Data Science.
- Bagian Kedua: Kelompok pembaca di atas yang tertarik ke bahasan yang lebih teknis, para mahasiswa/i S1, peneliti maupun praktisi yang tertarik dengan big data dan contoh hasil penelitian kami (di bidang big data dan analisis data).

Di lingkungan perguruan tinggi, buku ini dapat juga dijadikan salah satu rujukan/referensi pada mata kuliah yang setara dengan pengantar Data Science.

Sebagaimana tertuang pada Bab1, salah satu kompetensi utama dari seorang data scientist adalah mampu berkomunikasi verbal dan tertulis dengan baik atau melakukan *storytelling* yang membuat *audiens* terkesan. Para dosen di bidang Data Science tentu saja harus mampu mengajarkan hal ini. Agar lebih efektif, metoda pengajaran perlu dilaksanakan melalui praktek dan dengan contoh-contoh yang memadai. Para penulis bab-bab buku ini, yang juga dosen di bidang Data Science, telah berupaya menyiapkan bab-bab di buku ini dalam bentuk *storytelling*, dengan harapan dapat menjadi contoh (bagi yang sedang atau akan belajar Data Science).

Buku ini disiapkan di tengah masa pandemi COVID-19 yang membuat seluruh dunia menderita, tidak terkecuali Indonesia. Terasa pedih. Namun kami berupaya untuk tetap optimis, bersemangat dan produktif. Gotong-royong sudah menjadi budaya bangsa Indonesia. Karena itu, melalui karya hasil WFH (*work from home*) ini, kami berharap dapat memberikan kontribusi bagi kemajuan Indonesia, tanah air tercinta.

Bandung, September 2020

Editor

Sambutan Rektor Unviersitas Katolik Parahyangan

“Apa itu *Data Science*?” Apabila pembaca bisa menjawabnya, berarti pembaca hebat. Sebab, saya sebagai orang dengan latar belakang ilmu sosial dan termasuk Generasi-X tidak bisa menjelaskan apa yang disebut *Data Science*.

“Bericaralah dengan data!” Itulah nasihat umum yang ditujukan kepada seseorang jika terlalu banyak bicara dan “ngalor-ngidul”. Nasihat untuk menggunakan “data” dimaksudkan agar pesan yang disampaikan *meyakinkan*. Kedua, tujuannya adalah agar pembicaraan *efisien* dan *efektif*. Lebih jauh lagi, informasinya bisa dicek, diuji, dan *dipertanggungjawabkan*.

Data menjadi sedemikian sentral dalam kehidupan modern. Pengembangan sains dan teknologi yang sedemikian revolusioner didasarkan pada data. Kegiatan bisnis dan ekonomi juga semakin mengandalkan ketersediaan data. Bahkan, dinamika sosial-politik serta budaya dan seni tidak terlepas dari data. Diplomasi dan negosiasi internasional makin bertumpu pada data. Sebagian aspek religiositas dan spiritualitas pun nampaknya memiliki porsi yang cukup besar atas data.

Data yang dalam pemahaman umum adalah kumpulan fakta-fakta dan menjadi sumber informasi dan basis (ilmu) pengetahuan. Penguasaan dan pemilikan atas data selanjutnya menjadi ukuran kemampuan, sumber kekuatan, dan modalitas yang sangat penting untuk melakukan apapun atau untuk menjadi apapun. Sebagai kumpulan fakta, data dengan demikian tersebar, ada dimana-mana, sehingga sejarah peradaban umat manusia bisa disebut sebagai tumpukan atau akumulasi fakta. Fakta-fakta ini hanya menjadi (lebih) bermanfaat ketika berubah menjadi data dan selanjutnya menjadi informasi dan pengetahuan untuk menentukan pilihan-pilihan strategi dan keputusan.

Selamat dan terimakasih kepada para Penulis buku pengantar *Data Science* ini. Melalui penerbitan buku ini, editor, para Penulis dan juga Prodi Informatika, Fakultas Teknologi Informasi dan Sains (FTIS) UNPAR, tidak hanya memperkenalkan tetapi juga memberi informasi yang lebih baik dan lebih lengkap tentang apa itu *Data Science* serta kemanfaatannya dalam berbagai sektor dunia usaha dan dimensi keseharian hidup. Kegiatan belajar, tidur, atau memasak; usaha-usaha ekonomis dari pertanian ke manufaktur dan dunia hiburan (entertainment); managemen bisnis, transportasi, sampai dengan penanggulangan pandemik seperti Covid-19, semuanya memerlukan *Data Science*.

Pengenalan dan pemahaman tentang *Data Science* lewat penerbitan buku ini diharapkan menumbuhkan ketertarikan sekaligus minat untuk mempelajarinya lebih jauh. Jika Anda adalah calon mahasiswa, maka bergabung dengan Jurusan/Program Studi Informatika UNPAR menjadi keputusan yang tepat. Jika Anda adalah pelaku start-up atau ingin mengembangkan usaha bisnis yang sudah ada, maka konsultasi dan kolaborasi dengan para dosen di Informatika UNPAR juga akan sangat menjanjikan. Selain itu, jika Anda adalah awam seperti saya, setelah membaca buku ini, maka Anda dan saya bisa menasihatkan anak atau

cucu untuk belajar dan menjadi ahli di bidang *Data Science*. Sebab, melalui *Data Science*, hidup tidak hanya dimudahkan, tetapi juga hidup yang bisa dipertanggungjawabkan.

Selamat.

Bandung, Oktober 2020

Mangadar Situmorang, Ph.D

Data Science bagi Indonesia

Saya senang dan berbesar hati melihat buku *Pengantar Data Science dan Aplikasinya bagi Pemula* ini bisa terbit. Sebagai pelaku industri, khususnya dalam bidang digitalisasi bisnis dan organisasi, saya sudah lama melihat bahwa Data Science adalah disiplin ilmu dan profesi yang sangat relevan dan diperlukan oleh Indonesia. Namun, pada saat yang sama, saya melihat ilmu ini belum mendapat momentum di kalangan industri, para mahasiswa, serta profesional digital muda.

Oleh karena itu selama ini saya membuat sesi-sesi pengenalan Data Science ke berbagai kalangan di Indonesia, misalnya para mahasiswa, mereka yang baru menyelesaikan pendidikan S1, pihak manajemen, dan publik secara umum. Selain itu, walaupun tidak begitu intensif, saya juga sempat terlibat pada tahap awal dari pembentukan program Data Science di Universitas Parahyangan ini.

Indonesia sangat memerlukan Data Science untuk memecahkan berbagai tantangan besar dan untuk membuat lompatan ke depan, guna menunjukkan kemampuan dan kesungguhannya dalam membangun reputasi. Data Science bisa dikuasi banyak sekali orang Indonesia, sehingga akan ada banyak *data scientist*. *Data scientist* sendiri menjadi profesi yang berorientasi ke depan, bukan profesi lama. Profesi ini sudah sangat diperlukan Indonesia saat ini dan terus diperlukan dalam jumlah makin besar menuju tahun 2050. Ini menjadi tantangan bagi kalangan pemerintah, penyelenggara layanan publik, berbagai organisasi termasuk, tentunya sekolah-sekolah, dan universitas.

Terdapat banyak (sekali) permasalahan di Indonesia yang bisa dijawab melalui Data Science. Penanganan wabah Covid 19 sejak Maret 2020, misalnya, baik dalam pencegahan, penanggulangan dampak ekonomi, maupun kesehatan sangat bisa dibuat lebih efektif, efisien dan sistematis dengan Data Science. Layanan Kesehatan publik melalui BPJS juga sangat memerlukan Data Science agar berbagai pengambilan keputusannya didukung dengan *insights* yang digali dari berbagai data di tingkat layanan primer, rumah sakit, juga industri obat. *Insights* tersebut bahkan dapat digali secara bertingkat dari garda paling depan, yaitu kota, lalu provinsi, sampai ke tingkat nasional.

Perencanaan pembangunan infrastruktur di Indonesia, dengan sasaran pembangunan ekonomi dan kesejahteraan juga membutuhkan Data Science. Hal ini bertujuan agar pembangunan jalan tol, jalan akses, pelabuhan laut, pelabuhan udara, dan alat transportasi darat, dapat dirancang dan diimplementasikan dengan lebih efisien, efektif, runtun dan cepat. Industri distribusi untuk negara seluas dan semajemuk Indonesia dalam tatanan geografisnya tidak bisa efisien dan efektif jika pengambilan keputusan-keputusan operasional maupun strategisnya tidak berdasarkan data dan analisisnya tidak dalam skala *big data*.

Belum lagi, usaha-usaha dan bisnis di industri lama non-digital, tradisional, yang masih tidak tersentuh proses perubahan zaman ini. Sebagian besar masih menjalankan usaha menurut pengalaman tak terstruktur dan perasaan subyektif seperti yang mereka lakukan selama ini dan yang mereka pelajari dari pendahulunya. Sekolah-sekolah dan universitas tidak jauh berbeda, kalaupun ada sekolah atau universitas yang mengaku sudah mulai menggunakan *data analytics*, pada umumnya statusnya hanya sekedar membuat percobaan kecil, bukan bagian dalam arus utama manajemen. Selain itu, *data analytics* belum digunakan dalam pengambilan keputusan yang bertujuan untuk membawa kemajuan dan lompatan organisasi mereka.

Indonesia sangat membutuhkan Data Science untuk bisa melompat maju. Data Science bukan ilmu yang memerlukan aset besar (untuk digalakkan/digenjot) sehingga tidak dibutuhkan modal besar untuk menggunakannya. Modal utamanya adalah kemampuan mahasiswa dan pelaku profesi digital Indonesia yang harusnya berjumlah besar.

Memang sebelum masuk ke *data analytics*, ada hal yang perlu dilakukan dengan baik dan tekun, yaitu mengelola aset paling penting saat ini bagi kita, sebagai individu, lembaga, maupun pemerintah, yaitu data. Sejak web 2.0 diluncurkan di awal abad 21, data dihasilkan dari kegiatan manusia di seluruh bumi dengan kecepatan dan jumlah yang amat masif. Data bertebaran di sekeliling kita setiap saat. Hanya saja data ini tidak dikelola, atau tidak dikelola secara sistematis, dan tidak ada visi pimpinan yang menyentuh itu. *Data warehousing* beserta manajemennya yang terintegrasi perlu segera dibangun oleh pemerintah pusat, provinsi, kabupaten/kota , oleh BPJS, Rumah sakit, PLN, Telkom, Pertamina, BCA, Bank Mandiri, dan perusahaan perusahaan vital dan besar lainnya. *Data warehousing* dan manajemennya juga harus mulai dibangun oleh semua sekolah dan universitas, serta para pelaku usaha tradisional maupun digital. Ini diperlukan dan diperlukan segera. Saya sudah pula menyampaikan ini kepada beberapa membuat keputusan di tingkat pusat, khususnya agar mulai menangani pandemi dengan Data Science. Juga nanti, Data Science perlu digunakan pada administrasi penyebaran vaksin, yang akan sangat masif di Indonesia. Penyebaran vaksin dapat memakan waktu yang lama sekali, mungkin sampai 2024, jika teknologi, termasuk Data Science, tidak digunakan dengan baik. Sebuah *Data Science Operation* nasional perlu segera dibentuk. Ini bukan gedung baru yang besar; ini lebih pada pengorganisasian para talent data secara besar di seluruh Indonesia dan penyediaan *cloud infrastructure and services* yang aman sehingga *data scientist* Indonesia bisa bekerja secara terorganisir dari tempatnya masing-masing.

Rasa senang dan terima kasih saya atas langkah kecil awal, terbitnya buku ini oleh Program Data Science Universitas Parahyangan, Bandung. Langkah-langkah lanjutannya tentu ditunggu. Buku ini memberi gambaran dan memperjelas apa saja yang bisa dilakukan dan apa hubungan Data Science dengan kehidupan nyata kita sehari-hari. Ini bukan fiksi bukan pula utopi. Ini adalah realita kebutuhan hari ini. Saya juga gembira bahwa beberapa organisasi dan usaha di Indonesia mulai mau menoleh, mengerti dan kemudian menggunakan Data Science.

Bandung, Oktober 2020

Suryatin Setiawan
Senior Consultant and Coach,
Business and Organization Digitalization,
Penasihat Yayasan UNPAR Bandung
suryatin.setiawan@gmail.com

Bagian Pertama

Paparan Populer bagi Pemula

Bab 1 Data Science dan Data Scientist

Oleh:

Veronica S. Moertini

1.1. Data Abad 21

Bagi mayoritas orang, terlebih lagi yang belum berkecimpung di dunia kerja, barangkali data dianggap tidak penting. Data bisa jadi dianggap berkonotasi dengan “tumpukan” angka-angka yang membosankan dan “meaningless”. Data dianggap menjadi urusan perusahaan atau pemerintah, sehingga merupakan hal yang “jauh” dari kehidupan sehari-hari. Maka, meskipun “data science” atau ilmu data dan profesi data scientist sudah “terlahir” sejak beberapa tahun yang lalu, dapatlah dipahami bahwa masih banyak orang yang bertanya-tanya tentang apa itu data science, juga apa yang dikerjakan data scientist.

Sejatinya dalam kehidupan sehari-hari kita sudah memanfaatkan atau bahkan “menikmati” hasil data science atau buah karya dari para data scientist. Misalnya:

- Saat kita browsing di toko online, lalu kita klik salah satu item produk, di bawah browser akan diberikan produk-produk lain yang dibeli bersamaan atau yang mungkin kita suka juga. Sama halnya ketika kita browsing di penyedia streaming lagu dan video. Kita juga akan disuguhi dengan rekomendasi item-item lain untuk didengar atau dilihat. Tidak jarang setelah melihat item-item tersebut, kita jadi “tergoda” untuk melihat satu atau lebih item yang direkomendasikan. Bahkan, bisa berujung pada transaksi pembelian, jika item tersebut dijual.
- Buat kita yang tinggal di kota besar dengan trafik padat, adakah yang belum pernah “ngecek” kemacetan di jalan-jalan kota kita? Kita mungkin jadi batal pergi ke tempat tujuan jika jalan di situ dan sekitarnya berwarna “merah”. Ketika kita memilih jalur tercepat dari satu tempat ke tempat lainnya, mesin Google akan memanfaatkan informasi kepadatan lalu-lintas di tiap alternatif jalur untuk memilih yang tercepat. Warna hijau, kuning, oranye dan merah di peta Google telah menjadi informasi penting buat kita!
- Apa saja yang sedang “hot” dibicarakan di dunia maya? Berbagai trending di Twitter menjadi salah satu jawabannya. Di situ juga bisa kita dapatkan informasi sentimen atau persepsi, apakah positif atau negatif, terhadap pesan tertentu.
- Saat kita bepergian, terlebih lagi ke negara 4 musim dimana di suatu wilayah cuacanya dapat berubah dengan cepat (dalam hitungan jam), ponsel kita menjadi sumber informasi yang penting. Kita bisa cek di sekitaran objek wisata yang akan kita kunjungi, pada hari tanggal dan jam kita berada di sana, cuacanya bagaimana. Apakah akan turun hujan/salju? Angin kencang? Suhu *super* dingin atau sangat

panas? Dari situ, kita bisa menentukan fashion bagaimana yang cocok untuk kita kenakan. Bisa juga kita batal pergi ke objek itu.

- Pernah membandingkan hasil search di Google dengan keyword tertentu dari satu orang ke orang lain? Bisa beda. Hasil yang diberikan oleh mesin pencari Google akan dibuat sedemikian rupa, dibuat relevan dengan “kebiasaan” pencarian dan browsing kita di Internet.

Contoh-contoh di atas, baru segelintir dari yang sudah dihasilkan para data scientist. Sampai di sini, mungkin para pembaca sudah dapat merasakan atau menduga bahwa untuk menghasilkan tiap layanan di atas, data scientist bekerja dengan data tertentu. Misalnya, untuk menghasilkan rekomendasi item produk, dia menganalisis data transaksi di toko online (e-commerce). Untuk memberikan trending pembicaraan di dunia maya, data yang diproses adalah pesan-pesan Twitter, sedangkan untuk prediksi cuaca, yang diproses adalah data cuaca yang direkam oleh sensor-sensor di berbagai stasiun cuaca di bumi.

Pada abad ke-21 ini data sudah terbuat dan/atau terkumpul dari berbagai sumber (lihat Gambar 1.1). Pembuat data bisa jadi kita sendiri, yang lalu direkam di berbagai sistem, seperti media sosial, penyedia layanan email, chat, blog, review, foto dan video. Dapat juga berupa data bisnis atau data di organisasi (misalnya transaksi pembelian online, supermarket, perbankan, rumah sakit, instansi pemerintah, sekolah, pabrik, dan masih banyak lagi lainnya). Berbagai sensor (misalnya sensor cuaca dan video perekam di jalan, rumah dan perkantoran) dan satelit di angkasa juga berkontribusi banyak menghasilkan rekaman data. Berbagai alat IoT (Internet of Things), misalnya jam yang kita pakai, alat-alat rumah tangga dan mesin industri, juga senantiasa merekam data. Dari banyak jenis sumber tersebut, dapat dikatakan “data tidak pernah tidur”. Data terbuat terus dari detik ke detik dalam 24 jam dalam sehari. Pada tahun 2020 ini, diprediksi dihasilkan sekitar 35 zettabytes (10^{21} bytes atau 1.000.000.000.000.000.000 bytes) dari seluruh dunia (IBM Cognitive Class-2, 2020).

Dengan sumber-sumber data yang beragam di atas, sudah dapat kita duga bahwa bentuk atau format data yang direkam juga bermacam-macam. Untuk data bisnis atau di organisasi-organisasi, umumnya data terekam dalam format “tabular”, seperti data yang kita buat di sheet-sheet Excel. Data berbentuk teks dapat berasal dari email, chat, blog, review maupun medsos. Data suara dan video, dapat berasal dari medsos maupun sensor. Aliran data “numerik” (berupa angka-angka) dengan format/susunan tertentu diproduksi oleh sensor-sensor. Setiap satelit yang berada di ruang angkasa memiliki tujuan dan kegunaan tertentu, sehingga data yang direkam pun sesuai dengan kegunaannya. Secara umum, data yang direkam adalah “sinyal digital” berupa angka-angka yang contohnya dapat merepresentasikan lokasi, suara dan citra hasil penginderaan satelit itu.



Gambar 1.1. Contoh sumber data.

Barangkali pembaca sudah mendengar atau membaca istilah “big data”. Apa itu big data? Apakah data yang berukuran sangat besar? Dengan banyaknya sumber data, apakah jaman sekarang semua data menjadi big data? Belum tentu. Ulasan mengenai big data dengan lebih jelas dapat dibaca di Bab 10.

Sebagian data yang dibahas di atas tersedia di cloud dan dapat diunduh dengan gratis (misalnya data dari media sosial, cuaca dan sebagian data satelit). Ada juga yang dapat dibeli dengan harga terjangkau.

1.2. Apa itu Data Science?

Setelah mengenal contoh pemanfaatan hasil data science, berbagai sumber dan keragaman data, dapatlah diduga bahwa orang-orang yang “ngoprek” data yaitu data scientist, dibutuhkan di berbagai bidang. Bahkan, pada abad ke-21 ini, dimana semua sistem teknologi informasi telah menghasilkan data, data scientist telah dan akan dibutuhkan di semua bidang (industri, perdagangan, transportasi, layanan, kesehatan, pariwisata, pendidikan, dll). Tapi, apa itu data science?

Sesuai dengan namanya, data science melibatkan data dan sains atau ilmu (yang dibutuhkan untuk memproses data). Data science mulai didengungkan pada tahun 80-an dan 90-an, namun baru benar-benar dipublikasikan pada tahun 2009 atau 2011. Para ahli perintisnya antara lain adalah Andrew Gelman¹ dan DJ Patil².

¹ Profesor di bidang statistik dan ilmu politik dari AS yang telah menulis beberapa buku di bidang data science.

² Ilmuwan di bidang matematika dan ilmu komputer dari AS yang telah menulis beberapa buku di bidang data science.

Ada berbagai pendapat tentang definisi data science tapi Profesor Murtaza Haider dari Ryerson University di Kanada memiliki definisi yang cukup mudah dimengerti:

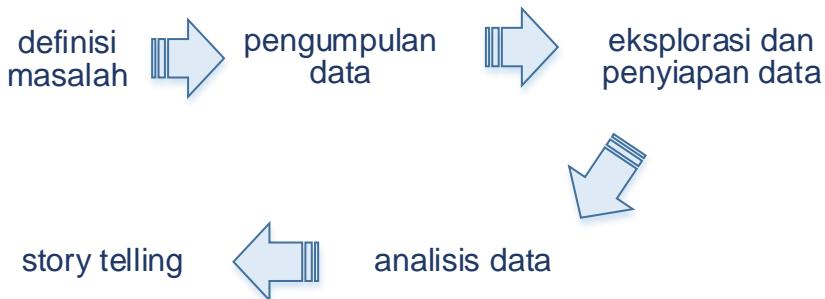
Secara sederhana dapatlah dikatakan bahwa data science “terjadi” ketika kita bekerja dengan data untuk menemukan jawaban atas pertanyaan-pertanyaan (tentunya yang relevan dengan data tersebut). Penekanannya lebih ke data itu sendiri dan bukan tentang sains atau ilmunya (yang dibutuhkan untuk menganalisisnya). Jika kita memiliki data, lalu kita memiliki *curiosity* (rasa ingin tahu) tentang “kandungan” atau “isi” data (yang bermanfaat), lalu untuk menjawab rasa ingin tahu tersebut kita mempelajari data, melakukan eksplorasi terhadap data itu, “memanipulasi”-nya, melakukan berbagai hal untuk menganalisis data tersebut dengan memanfaatkan ilmu dan teknologi tertentu untuk mendapatkan jawaban, itulah data science!

Tujuan akhir dari data science adalah untuk menemukan **insights** dari data. Data science dapat dipandang sebagai proses untuk mendekripsi atau mengekstraksi atau menggali insights dari data. Data yang diolah dapat berukuran sedang hingga sangat besar. Insights tersebut dapat diibaratkan sebagai emas atau berlian, yang meskipun hanya sedikit atau berukuran kecil, namun tetap berharga. Insights dapat berupa informasi penting maupun model-model yang dibuat dari data yang akan bermanfaat dalam mengambil keputusan. Insights yang ingin digali dari data perlu dimulai dengan rasa keingin-tahan yang kuat dari diri sendiri atau dari organisasi tempat dia bekerja (berupa kebutuhan karena ada masalah yang ingin diselesaikan dengan memanfaatkan data). Berbekal ini, seorang data scientist lalu melakukan berbagai aktivitas dengan memanfaatkan ilmu dan teknologi yang sesuai untuk mendapatkan insights yang disasar.

1.3. Apa saja yang Dikerjakan Data Scientist?

Ibaratnya menambang emas dari gunungan tanah yang melalui proses-proses yang berbelit dan membutuhkan berbagai mesin dan peralatan, untuk menemukan insights dari data (yang dapat berukuran sangat besar juga) pun demikian. Seorang data scientist mengerjakan berbagai pekerjaan dengan alat-alat (tools) pada beberapa tahap untuk mendapatkan insights.

Umumnya data scientist dibutuhkan oleh organisasi-organisasi yang telah memiliki sistem-sistem teknologi informasi operasional sebagai sumber data (lihat Gambar 1.1). Karena “data telah menumpuk” lalu ada kesadaran untuk mendapatkan insights yang bermanfaat. Untuk organisasi bisnis (misalnya perusahaan e-commerce, bank, transportasi dan pariwisata), insights bisa ditujukan untuk memperbaiki organisasi. Perbaikan itu misalnya karyawan menjadi lebih produktif, proses bisnis menjadi lebih efisien sehingga menurunkan biaya operasional, penjualan produk/jasa meningkat sehingga menaikkan keuntungan, layanan ke pelanggan menjadi lebih memuaskan sehingga pelanggan lebih loyal. Untuk organisasi pemerintah yang memberikan layanan kepada masyarakat, misalnya untuk meningkatkan produktivitas pegawai dan memperbaiki layanan. Untuk organisasi riset di bidang sains, kebutuhan akan berbeda, misalnya untuk menemukan model dari data yang bermanfaat untuk melakukan prediksi di masa depan. Model itu misalnya model prediksi panen tanaman, bencana, kebutuhan energi, kebutuhan transportasi penduduk, kerusakan lingkungan, dsb.



Gambar 1.2. Tahapan data science.

Disarikan dari (EMC, 2015), ketika seorang data scientist bekerja di organisasi-organisasi di atas, secara umum yang dilakukan adalah (lihat Gambar 1.2):

Pertama, tahap pendefinisian masalah. Data scientist mendapatkan kebutuhan organisasi yang harus dicari jawaban atau solusi dari data, misalnya menurunkan biaya produksi dan membuat pelanggan belanja lebih sering (Gambar 1.3). Dapat juga dia menerima insights spesifik yang akan digali dari data. Jika kebutuhan organisasi bersifat umum (misalnya menurunkan biaya produksi), maka data scientist harus mampu untuk merumuskan insights spesifik yang akan digali. Mulai tahap ini, curiosity menjadi bekal yang penting. Adanya curiosuty akan memberikan motivasi diri yang kuat yang dibutuhkan untuk menghadapi berbagai tantangan dan kesulitan dalam menggali insights.



Gambar 1.3. Hal-hal berharga (insights) apa yang dapat digali dari data?

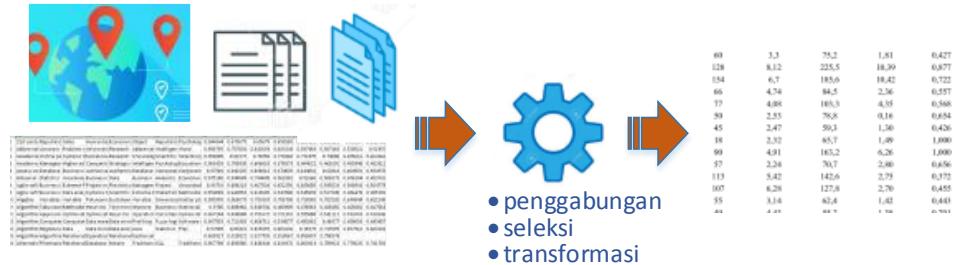
Kedua, tahap pengumpulan data. Berdasarkan insights yang akan digali, data scientist perlu merumuskan data apa saja yang dibutuhkan. Data itu dapat saja sudah tersedia semua atau baru sebagian. Jika baru sebagian, misalnya baru tersedia data transaksi sedangkan untuk menggali insights dibutuhkan data profile pelanggan dan Twitter, maka data scientist perlu mencari dan mengumpulkan data, yang dapat berasal dari satu atau lebih sumber (Gambar 1.4). Dalam hal tugas pengumpulan data ini kompleks atau berat karena harus dilakukan dengan mengakses berbagai sumber data pada sistem yang besar (dan kompleks pula), data scientist akan membutuhkan bantuan praktisi lain, khususnya data engineer yang tugasnya lebih berfokus dalam infrastruktur dan sistem pengelolaan data untuk organisasi. Jika

sebagian data belum terekam di sistem organisasi namun tersedia di luar organisasi (misalnya data harga saham, kependudukan, cuaca, satelit, yang tersedia di cloud), data scientist (bisa dengan bantuan data engineer) perlu “mengambil” data tersebut. Jika data belum tersedia di sistem organisasi maupun di luar, kemungkinan data scientist perlu untuk “mengadakan” data tersebut, misalnya melalui survei. Semua hal yang dilakukan tersebut harus disertai dengan pertimbangan terhadap isu privasi. Tahap ini dapat dikerjakan dengan cepat atau lama, bergantung kepada ketersediaan data.



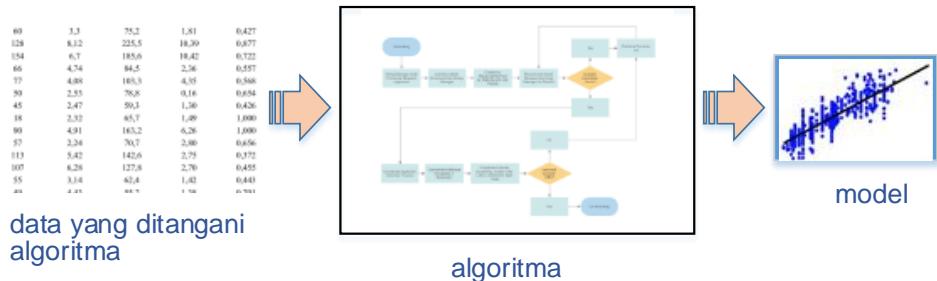
Gambar 1.4. Ilustrasi pengumpulan data kompleks dari berbagai sumber.

Ketiga, tahap eksplorasi dan penyiapan data. Setelah data terkumpul, seluruh komponen data perlu dipelajari dengan seksama. Misalnya, jika data berbentuk sebuah tabel, maka makna dan nilai tiap kolom harus dipahami. Untuk memahami data yang cukup kompleks dan berukuran besar, seringkali perlu dibuat visualisasi, kadang juga perlu komputasi statistik untuk mendapatkan ringkasan data (mencari rata-rata, median, minimum, maksimum juga distribusi data). Data juga harus diperiksa, karena seringkali data hasil pengumpulan tersebut masih “kotor”, berisi nilai yang salah atau ada yang hilang. Maka data perlu dicek, apakah semua nilai konsisten, benar atau tidak menyimpang. Jika data perlu diperbaiki, dalam kasus-kasus tertentu perbaikan data dapat dilakukan dengan memanfaatkan konsep statistika. Untuk data tertentu, mungkin juga perlu dilakukan “transformasi”, yaitu mengubah nilai data ke bentuk yang dibutuhkan dengan tidak menghilangkan maknanya. Untuk menyiapkan data final (berupa fitur-fitur yang siap untuk diumpulkan ke teknik atau algoritma analisis data yang akan digunakan), seringkali dia juga perlu memilih-milah, memilih data (detil ulasan dapat ditemukan di (Han & Kamberlin, 2012)). Ilustrasi pembuatan fitur diberikan pada Gambar 1.5. Jika data kompleks, pekerjaan di tahap ini bisa makan waktu lama dan sumberdaya yang banyak.



Gambar 1.5. Ilustrasi penyiapan data: Berbagai data diintegrasikan, dipilih yang relevan, dan/atau diubah menjadi fitur data yang siap diumpulkan ke sebuah algoritma analisis data.

Keempat, tahap analisis data. Jika data yang disiapkan sudah bagus, tahap ini dapat dilakukan dengan relatif lebih mudah, asalkan data scientist sudah menguasai teknik/algoritma, teknologi atau tools yang akan digunakan. Berdasarkan insights yang akan digali, di sini dipilih teknik atau algoritma yang sesuai (dapat berasal dari algoritma Machine Learning yang merupakan subset dari Artificial Intelligent atau Kecerdasan Buatan). Data scientist perlu memahami data yang ditangani, “behavior”, prinsip kerja, kelebihan dan kekurangan berbagai algoritma agar dapat memilih algoritma yang tepat. Jika tujuannya untuk membuat model, algoritma lalu dijalankan untuk mengolah data yang telah disiapkan agar dihasilkan model, misalnya model klasifikasi atau prediksi (Gambar 1.6). Model lalu diuji apakah sudah memenuhi standar tertentu. Dalam menguji model, misalnya menguji keakuratan dari model prediksi, data scientist perlu menguasai teknik-teknik pengukuran model (yang biasanya berbasis konsep statistika) dan memilih teknik yang tepat. Hasil uji lalu dievaluasi. Jika kualitas belum memenuhi syarat, model berpotensi tidak dapat dimanfaatkan, karena itu pembuatan model perlu diulangi lagi. Salah satu kemungkinan adalah dengan menyiapkan data masukan yang berbeda. Jadi, tahap pertama perlu diulangi lagi dan dilanjutkan ke tahap berikutnya, sampai didapatkan hasil analisis data yang memuaskan.



Gambar 1.6. Ilustrasi analisis data untuk mendapatkan model.

Kelima, *storytelling*. Seorang data scientist harus mampu untuk mengkomunikasikan proses dan hasil temuan analisis data dengan sistematis, menarik, tidak ambigu dan mudah dipahami bagi orang-orang (yang berkepentingan dengan proses maupun hasil itu). Bergantung kebutuhan di organisasi tempat data

scientist bekerja, komunikasi dapat dilakukan secara tertulis (dalam bentuk laporan) maupun tatap-muka pada rapat atau seminar (Gambar 1.7). Ibaratnya “mendongeng” (telling a story), pembaca atau audiens harus dibuat “terpesona” (impressed) dan percaya dengan hasil-hasil temuannya. Agar menarik dan mudah dipahami, paparan perlu dituangkan dalam bentuk-bentuk visual (yang merepresentasikan data, metoda, model, hasil uji model, dll) yang tepat. Karena itu, data scientist harus mampu menyusun laporan yang sistematis, jelas, berkualitas bagus dan menguasai teknik presentasi yang efektif. Insights yang ditemukan akan menjadi dasar pengambilan keputusan yang bisa jadi berdampak luas, karena itu pihak-pihak yang berkepentingan harus dapat diyakinkan tentang kebenaran temuan itu.



Gambar 1.7. Storytelling dengan berbagai visualisasi.

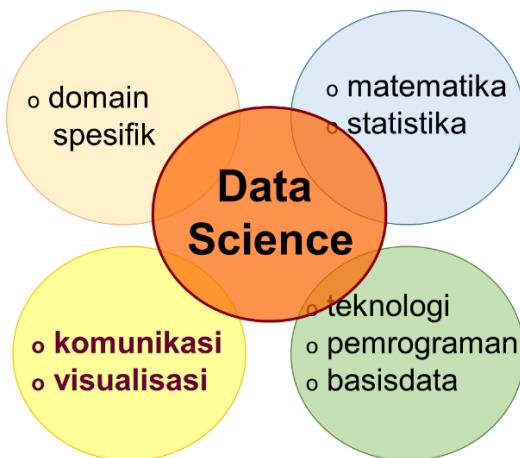
Setelah melakukan storytelling, harapannya tentu saja temuan insights-nya akan dimanfaatkan, menjadi kebijakan, program kerja ataupun *actions* yang tepat terap bagi organisasi. Untuk itu, data scientist perlu memberikan berbagai dukungan yang dibutuhkan. Sesudah hasil temuannya dimanfaatkan, kemungkinan akan muncul masalah-masalah baru yang perlu dicari penyelesaiannya melalui analisis data lagi. Dengan demikian, peran data scientist akan dibutuhkan lagi dan pekerjaan data scientist merupakan pekerjaan yang berkelanjutan.

Jika temuan data scientist berupa model, misalnya yang bermanfaat untuk memprediksi atau memberikan rekomendasi, lalu model tersebut akan “diluncurkan” di aplikasi atau website atau sistem informasi di organisasi, data scientist seringkali perlu bekerja-sama dengan tim pengembang aplikasi/sistem tersebut (karena umumnya pengembangan aplikasi/sistem informasi tidak menjadi ranah kerja para data scientist). Model yang dihasilkan tersebut kemungkinan juga perlu penyesuaian atau pengembangan dari waktu ke waktu seiring dengan perubahan ataupun bertambahnya data yang dianalisis. Jadi, di sini peran data scientist juga berkelanjutan.

1.4. Keahlian dan Skill Data Scientist

Agar dapat melaksanakan kelima tahap data science itu dengan sukses, bekal ilmu, keahlian dan ketrampilan apa saja yang dibutuhkan untuk menjadi seorang data scientist? Untuk menjadi seorang data scientist, orang harus belajar apa saja?

Secara ringkas, data scientist perlu menguasai beberapa ilmu, keahlian dan ketrampilan yang dapat dikelompokkan menjadi empat (IBM Cognitive Class-2, 2020), yaitu (lihat Gambar 1.8): keahlian substansi di bidang khusus tertentu; matematika dan statistik; teknologi, pemrograman dan basisdata; serta komunikasi dan visualisasi. Keterangan dari setiap kelompok tersebut diberikan di bawah ini.



Gambar 1.8. Keahlian dan skill multi-disiplin data scientist.

Keahlian pada Domain Spesifik

Pada abad 21 ini nyaris tidak ada bidang yang tidak membutuhkan data scientist (lihat Subbab 1.2). Masing-masing organisasi yang bergerak di bidang tertentu (misalnya manufaktur, ritel, transportasi, pariwisata, kesehatan dan pendidikan) memiliki data yang spesifik dan kebutuhan unik yang terkait dengan organisasi mereka. Data scientist harus mampu memahami data dan kebutuhan organisasi tempat dia bekerja agar dapat menggali insights yang tepat dari data yang nantinya bermanfaat bagi organisasi tersebut. Itu sebabnya seorang data scientist perlu memiliki keahlian pada bidang atau domain yang spesifik.

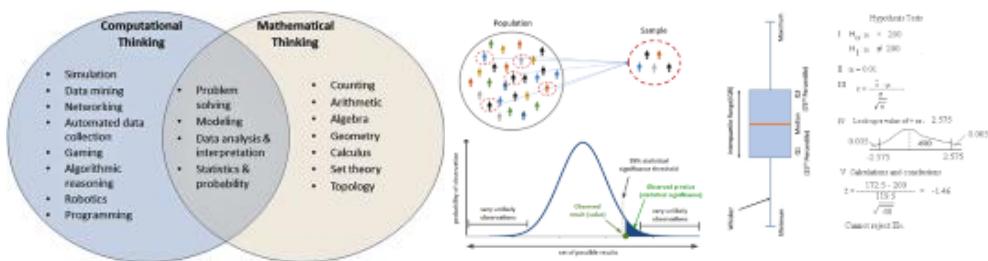
Sebagai contoh, jika seseorang ingin menjadi data scientist bagi perusahaan e-commerce, maka dia membutuhkan ilmu dan skill yang relevan dengan manajemen pelanggan, pemasaran digital, budaya netizen, media sosial dan web analytics. Jika untuk pabrik, misalnya, dia membutuhkan pemahaman

terhadap produk yang dibuat, proses produksi, manajemen rantai pasokan, logistik dan pemasaran. Jika untuk pemasaran bidang pendidikan (di universitas), dia harus paham tentang bidang-bidang pendidikan di universitas, pemasaran digital, hubungan manajemen pelanggan untuk dunia pendidikan dan perilaku siswa sekolah menengah. Keahlian khusus yang dibutuhkan data science di bidang kedokteran, lingkungan (yang terkait dengan bumi dan permasalahannya), lembaga antariksa yang mengelola satelit dan perusahaan penyedia transportasi udara, dapat dibayangkan, akan sangat berbeda dengan masing-masing contoh tersebut.

Walaupun keahlian dan ketrampilan data scientist dapat digolongkan ke dalam 4 kelompok, namun dengan menentukan bidang khusus yang tertentu, nantinya seorang data scientist akan membutuhkan bagian ilmu matematika yang tertentu, juga menguasai teknologi, tools, algoritma dan pemrograman yang tertentu pula. Sebagai contoh, teknologi, teknik-teknik atau algoritma-algoritma yang digunakan untuk menganalisis data satelit, secara umum akan berbeda dengan yang digunakan untuk mengolah data transaksi perusahaan e-commerce dan data klik pengunjung website.

Matematika dan Statistik

Sebelum data science ditemukan, orang sudah memanfaatkan statistik untuk menganalisis data. Misalnya, statistik dimanfaatkan untuk mendapatkan distribusi atau sebaran data, “ringkasan” data (seperti frekuensi kemunculan, rata-rata, median, minimum, maksimum, percentile 25-75%, dsb), pengujian hipotesis, juga membuat sampel data dan melakukan analisis multivariat. Pada saat mempelajari dan mengeksplorasi data, data scientist seringkali menggunakan statistika untuk memahami data. Jika kemudian dia mendapati ada data yang salah atau tidak konsisten, data scientist juga perlu menangani hal ini (istilahnya, “membersihkan data”) antara lain dengan memanfaatkan statistika. Statistika juga dibutuhkan ketika data scientist perlu mengubah satu nilai ke nilai lain (istilahnya, “mentransformasi data”). Bergantung kepada insights yang akan digali dari data, kadang analisis data juga dapat dilakukan dengan statistika (beserta visualisasi hasilnya). Penggunaan statistika juga dibutuhkan ketika data scientist menguji insights yang berupa model, untuk mengukur tingkat kebenaran model atau membandingkan berbagai model yang didapatkan untuk dipilih yang terbaik.



Gambar 1.9. Ilustrasi matematika dan statistik untuk data scientist.

Matematika di sini konteksnya luas, termasuk kemampuan berpikir secara logis, sistematis, dan matematika diskret. Jadi, tidak hanya ilmu matematika seperti aritmatika, aljabar, kalkulus, himpunan, geometri, dsb. Jika insights yang akan digali dari data berupa model, misalnya model yang dapat digunakan untuk melakukan prediksi di masa depan, maka Machine Learning perlu digunakan. Setiap algoritma Machine Learning (seperti pengelompokan, klasifikasi data, regresi, analisis aturan asosiasi, outlier, dll.) dirancang berbasiskan matematika dan statistik. Karena itu, penguasaan matematika menjadi dasar bagi data scientist dalam memahami berbagai algoritma Machine Learning. Berbekal pemahaman yang memadai terhadap algoritma-algoritma itu, data scientist lalu dapat memilih algoritma-algoritma yang cocok untuk digunakan dalam menganalisis data yang sudah disiapkan. Ilustrasi untuk kelompok bidang ini diberikan pada Gambar 1.9.

Teknologi, Pemrograman dan Basisdata

Data yang akan dianalisis pastilah tersimpan di suatu (atau beberapa) tempat penyimpanan data. Sistem yang menyimpan dan mengelola data dinamakan sistem basisdata. Sistem ini dapat mengelola data berformat terstruktur (bertipe tabular), semi terstruktur (misalnya data dengan format HTML, CSV, JSON dan XML, juga data spasial atau data geografis), maupun tidak terstruktur (misalnya dokumen, email, foto dan video). Berdasarkan format yang macam-macam tersebut, sudah dapat dibayangkan bahwa sistem basisdata yang mengelola tiap tipe data juga berbeda. Misalnya, sistem basisdata relasional, menangani data terstruktur, sedangkan basisdata NoSQL utamanya menangani data semi-terstruktur dan tidak terstruktur. Sistem basisdata juga ada yang berjalan di atas sistem big data (misalnya, Hadoop dan Spark) maupun di cloud. Seorang data scientist harus mampu untuk “mengambil” dan memanipulasi data yang tersimpan di basisdata. Maka, dia harus menguasai konsep basisdata dan teknologi basisdata yang menyimpan data yang akan dianalisisnya. Selain itu, dalam mengambil, memilih, memeriksa data dan menyimpan hasil data yang disiapkan ke sistem basisdata, dia juga harus mampu memprogram dengan bahasa pemrograman yang digunakan oleh sistem basisdata itu, misalnya SQL pada basisdata relasional (MySQL, Oracle, SQL Server, dll), HQL pada basisdata berbasis objek, PostgreSQL pada Postgres, HiveQL pada Hive (yang berjalan di atas Hadoop), SparkSQL untuk Spark dan BigQuery untuk datawarehouse Google Cloud (lihat Gambar 1.10).



Gambar 1.10. Berbagai teknologi dan tools analisis data.

Dalam melakukan eksplorasi, menyiapkan maupun menganalisis data yang telah disiapkan, data scientist dapat menggunakan software atau tools yang sesuai dengan data yang diprosesnya. Tools untuk data bisnis yang berformat tabular, akan berbeda dengan tools untuk data teks, citra maupun spasial. Untuk data berukuran kecil sampai sedang, misalnya, Excel dapat digunakan untuk visualisasi, penyiapan data sampai analisis. Namun, jika data scientist perlu menganalisis data teks (misalnya pesan Twitter), dia membutuhkan tools lain. Kemudian, walaupun berformat tabular, tapi jika ukuran data sangat besar (bergiga-giga) dan sudah tersimpan di sistem big data, maka analisis perlu dilakukan dengan software untuk big data (misalnya Hive dan SarkSQL). Berbagai software analisis data maupun layanan cloud sudah menyediakan fitur-fitur Machine Learning. Untuk menganalisis data dengan algoritma tertentu, data science dapat memanfaatkan fitur yang sudah disediakannya. Sekarang sudah tersedia berbagai tools baik untuk data kecil maupun sangat besar, baik yang berjalan di komputer desktop, jaringan maupun cloud, juga untuk berbagai jenis data. Data scientist harus mampu memilih satu atau lebih tools yang tepat dan menggunakan dengan baik untuk melaksanakan tugasnya.

Tools untuk menganalisis data saat ini cukup banyak yang dapat diperoleh dengan gratis.



Gambar 1.11. Contoh bahasa pemrograman bagi data scientist.

Dalam mengumpulkan, mempelajari, menyiapkan data, seorang data scientist seringkali harus memprogram (“ngoding”). Bahkan, di tahap analisis data, jika tidak ada tools yang memiliki fitur yang tepat untuk digunakan, dia juga perlu memprogram (untuk mengimplementasikan algoritma analisis data yang khusus). Untuk dapat memprogram, dia harus mampu berpikir secara sistematis dan terstruktur dan memahami cara bekerja sistem komputer. Dia harus mampu berpikir analitis agar dapat merancang langkah-langkah pada program atau algoritma program. Dia juga harus memiliki pahamanan terhadap matematika dan statistika yang kuat agar dapat menerjemahkan rumus-rumus menjadi program dengan tepat dan benar. Terdapat berbagai pilihan bahasa pemrograman, masing-masing memiliki kegunaan, kelebihan dan kekurangannya sendiri (Gambar 1.11), misalnya Python, R, Java, dan yang digunakan pada sistem basisdata yang sudah dibahas di atas (SQL, HQL, PostgreSQL, dll). Jika dia bekerja menganalisis big data yang juga tersimpan pada sistem big data, dia perlu memprogram dengan salah satu atau lebih dari pilihan ini: MapReduce pada Hadoop, Scala (untuk memanfaatkan library Machine Learning pada Spark) dan SparkSQL (untuk mengakses data terstruktur pada Spark), HiveQL (untuk mengakses data terstruktur pada Hive). Jika data tersimpan di cloud, dia perlu memprogram dengan bahasa yang digunakan di layanan cloud itu, misalnya BigQuery.

Komunikasi, Visualisasi dan Softskill Lainnya

Sebagaimana dipaparkan pada tahap-tahap data science, setelah menemukan insights dari data, data science harus mampu untuk mengkomunikasinya (baik secara tertulis maupun tatap-muka) dengan efektif, menggunakan berbagai bentuk visual yang menarik, bergaya story-telling. Maka, keahlian story-telling dan visualisasi harus dikembangkan terus-menerus oleh data scientist. Karena dia harus mampu merancang bentuk-bentuk visual dengan menerapkan seni, maka dia harus menguasai berbagai tools untuk visualisasi data (misalnya Excel, Tableau atau lainnya seperti ditunjukkan pada Gambar 1.12) atau mampu memprogram untuk menghasilkan bentuk visual khusus yang menarik (misalnya distribusi data pada peta).

Dalam menjalankan tahap-tahap analisis data yang seringkali penuh tantangan dan harus berkoordinasi dengan berbagai pihak, seorang data scientist perlu memiliki passion (kecintaan) terhadap yang dikerjakan, curious terhadap data, hacker-mindset, problem-solver, berpikir strategis, bersikap proaktif, kreatif, inovatif dan kolaboratif.



Gambar 1.12. Contoh tools untuk membuat visualisasi.

1.5. Era Industri 4.0 dan Data Science

Pada awal abad ke 21 ini dunia memasuki era revolusi Industri 4.0. Data Science seringkali dikaitkan dengan era ini. Lalu, apa itu sebenarnya Industri 4.0? Dilansir dari sebuah artikel pada majalah bisnis Forbes, berikut ini ulasan ringkasnya (Marr, 2009):

Sampai saat ini, revolusi industri sudah terjadi 4 kali, yaitu:

- Pertama, terjadinya mekanisasi peralatan industri dengan memanfaatkan tenaga air dan uap.
- Kedua, pabrik-pabrik mampu melakukan perakitan atau produksi barang secara masal dengan menggunakan tenaga listrik.
- Ketiga, industri mulai mengadopsi komputer-komputer dan proses otomatisasi dengan memanfaatkan sistem cerdas, menggunakan data dan algoritma-algoritma Machine Learning.
- Selanjutnya, Industri 4.0 terjadi seiring dengan ketersediaan berbagai sistem teknologi informasi, peralatan Internet of Things (IoT) dan Internet yang makin mudah diakses dan digunakan. Pada era

4.0, berbagai sistem teknologi informasi tersambung secara digital sehingga mampu berkomunikasi untuk saling berbagi data dan informasi. Peralatan dan mesin-mesin makin pintar karena dilengkapi dengan kemampuan untuk menangkap dan memproses atau menganalisis data. Dengan memanfaatkan jaringan dan peralatan yang serba pintar tersebut, sebuah sistem juga dimungkinkan membuat keputusan tanpa campur tangan manusia. Kegiatan industri jadi makin efisien, para produsen makin produktif.

Tiga contoh penerapan Industri 4.0 diberikan di bawah ini:

Identifikasi peluang: Industri 4.0 menawarkan peluang bagi pabrik-pabrik untuk meningkatkan efisiensi a.l. dengan mempercepat proses produksi. Ini dimungkinkan karena masalah (penting) yang terjadi dapat diidentifikasi dengan cepat dan segera dicari solusinya. Pada era ini, mesin-mesin saling terhubung sehingga dapat mengumpulkan berbagai data dalam jumlah yang besar, dimana setelah diproses dapat memberikan informasi yang terkait tentang pemeliharaan, kinerja dan masalah lain yang dapat segera ditindak-lanjuti. Selain itu, data yang terkumpul juga dapat dianalisis untuk mencari pola-pola dan insights (informasi berharga) yang tidak mungkin "digali" secara manual oleh manusia.

Optimasi logistik dan rantai-pasokan (supply chain): Sistem rantai-pasokan yang terintegrasi dapat dibuat lebih responsif atau adaptif. Sistem dapat segera melakukan penyesuaian atau perubahan ketika menerima sebuah informasi baru. Misalnya, ketika sistem menerima informasi tentang terjadinya cuaca buruk yang menghambat pengiriman barang pada bagian delivery (sehingga stok barang menumpuk), sistem "bersikap" proaktif, segera mengubah prioritas produksi di bagian pabrik untuk mengatasinya.

Internet of Things (IoT): Komponen kunci pada Industri 4.0 adalah IoT yang dicirikan dengan saling tersambungnya peralatan-peralatan IoT dan pemanfaatan cloud untuk menyimpan data yang dikirim dari peralatan IoT (secara instant atau real time). Pelaku industri lalu dapat memanfaatkan layanan hasil analisis data di cloud tersebut untuk membuat operasi peralatan-peralatan mereka menjadi lebih efisien (tanpa harus melakukan analisis data sendiri yang dapat membutuhkan sumber daya yang tidak terjangkau).

Dari ulasan ringkas di atas, dipaparkan bahwa di era Industri 4.0, mesin-mesin atau berbagai alat atau sistem dibuat menjadi pintar (seolah-olah mampu berpikir dan memutuskan sendiri) karena dilengkapi dengan kemampuan untuk mengambil data, menganalisis data, atau mengambil informasi penting hasil analisis data dari mesin lain atau dari cloud. Informasi ini lalu digunakan sebagai dasar untuk bertindak (melakukan aksi). Jadi, Industri 4.0 tidak terlepas dari analisis berbagai data yang hasilnya dimanfaatkan berbagai mesin dan alat.

1.6. Kebutuhan Data Science

Beberapa laporan hasil survei dan analisis dari berbagai lembaga menyampaikan bahwa data scientist telah menjadi kebutuhan global maupun di Indonesia.

Untuk lingkup global, berikut ini informasi dari beberapa sumber:

- McKinsey & Company, penyedia layanan konsultasi manajemen dan strategi bisnis, melaporkan bahwa teknologi Artificial Intelligent (AI) yang termasuk Machine Learning makin banyak dibutuhkan karena memberikan keuntungan-keuntungan di bidang bisnis (McKinsey, 2018).
- World Economic Forum (WEF) melaporkan kebutuhan data scientist yang meningkat pada berbagai bidang, misalnya pada industri yang berbasis teknologi informasi, media dan hiburan, layanan finansial dan investasi, layanan profesional, pemerintah, dll. (WEF, 2019).
- Asia-Pacific Economic Cooperation (APEC) pada laporan tahun 2017 menuliskan: *Data Science and Analytics (DSA) skills are in high demand, but supply is critically low with employers facing severe shortages* (APEC, 2017).
- LinkedIn, organisasi yang mengelola jaringan para profesional di Internet yang terbesar, pada laporan tahun 2020 menempatkan data scientist di *top 10 emerging jobs* di berbagai negara, seperti Amerika Serikat (LinkedIn-US, 2020), Kanada (LinkedIn-CA, 2020), Australia (LinkedIn-Aus, 2020). Demikian juga di kawasan ASEAN, seperti Singapore (LinkedIn-Sing, 2020) dan Malaysia (LinkedIn-Malay, 2020).

Bagaimana dengan di Indonesia?

Banyak perusahaan mencari data scientist atau pakar artificial intelligence, maupun data engineer. Semua itu terkait dengan pengolahan data. Hal tersebut dapat kita temui di lowongan-lowongan pekerjaan di banyak perusahaan Indonesia. Pada laporan yang dirilis LinkedIn tahun 2020, kebutuhan data scientist di Indonesia menempati urutan ke empat (LinkedIn-Indon, 2020). Mengapa kebutuhannya begitu besar? Seperti disampaikan oleh Taufik Susanto³, doktor pada bidang data science lulusan Queensland University of Technology Australia dan pendiri konsultan di bidang data science, saat ini pengolahan data menjadi penentu kompetisi bisnis antar perusahaan. Taufik memberi ilustrasi kompetisi Gojek versus Grab. Siapa yang mampu membuat profile pelanggannya, memilih perhitungan harga yang tepat dan promo yang tepat maka dia akan menjadi pemenangnya. Trend ini minimal sampai 5 tahun kedepan akan sama dengan saat ini. Bahkan mungkin bisa lebih intense lagi.

Lalu industri apa saja yang membutuhkan data science? Menurut Taufik, pada jaman sekarang semua perusahaan/industri bahkan institusi pendidikan sangat membutuhkan data science. Kalau perusahaan ritel seperti Tokopedia, Bukalapak, Blibli dan Lazada tidak mampu untuk mengolah data dengan baik maka akan collapse (tidak mampu bersaing). Demikian juga otomotif. Industri pariwisata juga membutuhkan data science.

³ <https://www.techfor.id/pendidikan-data-science-di-indonesia/> [diakses 12 Juni 2020]

Berdasarkan hasil survei, saat ini kebutuhan data scientist di Indonesia diperkirakan baru terpenuhi sekitar 50%⁴. Hasil survei yang dilakukan oleh Sharing Vision terhadap 27 perusahaan (dan dipublikasikan pada Januari 2019) menunjukkan bahwa 66% responden menilai Big Data akan booming di Indonesia pada 1-2 tahun ke depan. Selain itu, hasil survei ini juga menunjukkan bahwa 48% perusahaan sudah memasukkan pengembangan sistem Big Data ke dalam IT Strategic Plan, bahkan 33% di antaranya sudah mengoperasikan sistem tersebut dan 33% lainnya sedang mengembangkan sistem big data.

Belum terpenuhinya lowongan data scientist di Indonesia ini, sejalan dengan ini: Pada laporan *Global Skills Index 2020* yang diterbitkan oleh Coursera (penyelenggara kursus daring global dengan 65 juta peserta anggota), untuk bidang Data Science, Indonesia ditempatkan pada posisi *lagging* atau tertinggal. Dari 60 negara (di benua Amerika, Eropa, Asia, Afrika dan Australia) yang ditelaah, Indonesia berada di posisi 56. Padahal untuk bidang teknologi, Indonesia berada di posisi *emerging*, urutan ke 31(Coursera, 2020).

1.7. Informasi Bab-bab Buku

Konten buku ini dibagi menjadi dua bagian, dengan deskripsi sebagai berikut:

Bagian Pertama: berisi bab ini dan contoh aplikasi data science dan pekerjaan data scientist dalam menggali insights pada berbagai bidang spesifik. Inti konten dari tiap bab diberikan dibawah ini:

- Bab 2: Aplikasi **statistik** dan **visualisasi** terhadap data *smartwatch* yang dikenakan pada para partisipan penelitian untuk mendapatkan pola belajar dan tidur yang mendukung prestasi akademis yang bagus.
- Bab 3: Paparan sederhana tentang bagaimana **rekomendasi item** pada website e-commerce “dihitung” berdasar data *rating* pengunjung, dengan memanfaatkan **algoritma collaborative filtering item-based**.
- Bab 4: Pemanfaatan teknik **clustering** (pengelompokan) untuk menganalisis data menu dan bahan masakan yang hasilnya dapat digunakan untuk membantu kita dalam memilih menu kuliner yang sesuai selera.
- Bab 5: Dari data pengindera jauh satelit beserta data lain yang didapatkan di sawah, dapat dibuat model prediksi dengan **regresi** untuk memperkirakan jumlah panen padi (ketika sawah masih hijau).
- Bab 6: Contoh-contoh pemanfaatan berbagai bentuk **visualisasi** dari data untuk mendapatkan insights dari data dengan studi kasus data COVID-19.
- Bab 7: Informasi yang terkait dengan pola hidup sehat dapat diperoleh dari aplikasi ponsel yang mengambil data *smartwatch*. Bab ini memberikan paparan sederhana tentang **teknik klasifikasi** dengan **Jaringan Syaraf Tiruan**, yang merupakan cikal-bakal dari sistem **deep learning**. Data yang dikumpulkan dan dianalisis adalah data aktivitas pemakai *smartwatch*, sedangkan model klasifikasi yang dibuat dimanfaatkan untuk memprediksi kualitas tidur pemakainya.

⁴ [https://jabar.sindonews.com/berita/4305/1/sdm-data-scientist-di-indonesia-masih-minim#:~:text=%22Kebutuhan%20data%20scientist%20saat%20ini,22%2F1%2F2019\)](https://jabar.sindonews.com/berita/4305/1/sdm-data-scientist-di-indonesia-masih-minim#:~:text=%22Kebutuhan%20data%20scientist%20saat%20ini,22%2F1%2F2019))

- Bab 8: Pemanfaatan algoritma *user-based collaborative filtering* dan algoritma pengelompokan *Fuzzy c-Means* untuk menganalisis data rating film dan hasilnya dapat dimanfaatkan untuk memberikan rekomendasi film yang cocok bagi penonton.
- Bab 9: Pemaparan tentang bagaimana para pengguna ponsel berkontribusi dalam mengumpulkan data yang dimanfaatkan Google untuk memberikan informasi tentang kepadatan trafik di peta Google. Juga tentang bagaimana kita dapat melakukan praktik kecil-kecilan untuk menganalisis data trafik dari peta tersebut bagi kepenting kita.

Bagian Kedua: berisi paparan yang lebih teknis yang terkait tentang big data dan contoh hasil penelitian dosen dan mahasiswa yang terkait dengan big data dan Data Science. Inti konten dari tiap bab adalah:

- Bab 10: Pemaparan tentang **big data**, mengapa sekarang populer, dan **berbagai teknologi** yang sudah tersedia untuk mengumpulkan, memanajemen dan menganalisis big data.
- Bab 11: Contoh **pemanfaatkan** teknologi big data, khususnya **Spark, Hadoop, Kafka** untuk mengumpulkan aliran data Twitter dan menganalisisnya dengan statistika sederhana.
- Bab 12: Pengembangan dan perbandingan **algoritma pengelompokan paralel k-Means** pada lingkungan sistem big data Hadoop dan Spark.
- Bab 13: Pengukuran dimensi tubuh dengan memanfaatkan **data** dari perangkat permainan **konsol Xbox** (yang memiliki sensor untuk menangkap dan mengenali gerakan dan gestur tubuh pemain). Hasilnya berpotensi untuk dimanfaatkan, misalnya, pada penentuan ukuran pada pembelian baju secara daring.
- Bab 14: Data berupa foto (citra) seringkali perlu di-praolah terlebih dahulu agar dapat dianalisis lebih lanjut, misalnya untuk keperluan pengenalan bentuk-bentuk objek pada citra. Bab ini memaparkan segmentasi citra untuk **mempraolah data citra** menggunakan algoritma Particle Swarm Optimization.

Dengan beberapa variasi konten pada bab-bab di atas, diharapkan para pembaca akan mendapatkan pengetahuan awal tentang Data Science dan big data yang memadai.

Referensi

- (APEC, 2017) Asia-Pacific Economic Cooperation (APEC) - Human Resource Development Working Group, *Data Science and Analytics Skills Shortage: Equipping the APEC Workforce with the Competencies Demanded by Employers*, July 2017.
- (Coursera, 2020) Coursera, *Global Skills Index*, 2020
- (EMC, 2015) EMC Education Services, *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*, Wiley Publ., USA, 2015.
- (IBM Cognitive Class-1, 2020) IBM Cognitive Class, *Introduction to Data Science*, <https://cognitiveclass.ai/courses/data-science-101> (diakses 6 Juni 2020)
- (IBM Cognitive Class-2, 2020) IBM Cognitive Class, *Big Data 101*, <https://cognitiveclass.ai/courses/what-is-big-data> (diakses 6 Juni 2020)

- (Han & Kamberlin, 2012) J. Han & Kamberlin, *Data Mining Concept and Techniques 3rd Ed.*, Morgan Kauffman Publ., USA, 2012
- (Linkedin-US, 2020) Linkedin, *2020 US Jobs Trends*, 2020.
- (Linkedin-CA, 2020) Linkedin, *2020 Canada Emerging Jobs*, 2020.
- (Linkedin-Aus, 2020) Linkedin, *2020 Emerging Jobs Report Australia*, 2020.
- (Linkedin-Sing, 2020) Linkedin, *2020 Emerging Jobs Report Singapore*, 2020.
- (Linkedin-Malay, 2020) Linkedin, *2020 Emerging Jobs Report Malaysia*, 2020.
- (Linkedin-Indon, 2020) Linkedin, *2020 Emerging Jobs Report Indonesia*, 2020.
- (Marr, 2009) B. Marr, *What is Industry 4.0? Here's A Super Easy Explanation For Anyone*, Forbes, 2 September 2018, <https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/#318549f99788> [diakses 13 Juni 2020]
- (McKinsey, 2018) McKinsey & Co, *Analytics comes of age*, New York, NY, USA, 2018.
- (WEC, 2019) World Economic Forum, *Data Science in the New Economy: A new race for talent in the Fourth Industrial Revolution*, Swiss, 2019.

Bab 2 Menjelang Ujian: Ngebut Belajar atau Tidur?

Oleh:

Natalia dan Vania Natali

2.1. Pendahuluan

"*Ngebut* belajar atau tidur saja ya?" demikianlah kegalauan yang seringkali dihadapi oleh siswa-siswi atau mahasiswa-mahasiswi saat menjelang ulangan atau ujian. Ada kalanya, demi mengejar penguasaan bahan, seorang pelajar begadang untuk belajar menjelang hari ujian sampai tertidur-tidur (Gambar 2.1). Di sisi lain, tidak jarang para pelajar merasa bingung karena cukup sering mendapat fakta: *si A* belajar sampai tidak tidur, tapi tetap saja mendapat nilai pas-pasan; sedangkan *si B*, hanya baca-baca sekilas materi ujian, lalu tidur nyenyak berjam-jam, ternyata selalu mendapatkan nilai cemerlang. Jadi, apakah mengutamakan tidur adalah solusi terbaik sebelum ujian? Bab ini membahas penelitian yang dimaksudkan untuk mencari jawab dari pertanyaan ini. Bab ini juga sekaligus memberikan contoh pemanfaatan [statistika](#) dan [visualisasi data](#) untuk menganalisis data dalam menggali insights (informasi berharga), yang hasilnya dapat dimanfaatkan oleh para pelajar (siswa/i dan mahasiswa/i) dalam mengatur jadwal tidur dan bangun.



Gambar 2.1. Siswa tertidur ketika belajar.

Banyak orang yang telah berusaha untuk menemukan keterkaitan antara prestasi dalam kegiatan akademik dan kualitas tidur seseorang. Dalam beberapa penelitian, dikatakan bahwa semakin baik kualitas tidur seseorang, akan semakin baik pula prestasi akademiknya. Terdapat pemahaman bahwa jaringan sel syaraf sinaptik yang aktif selama seseorang dalam keadaan sadar, diperkuat keadaannya ketika seseorang sedang dalam keadaan tidur. Syaraf sinaptik tersebut berhubungan dengan konsolidasi daya ingat seseorang. Dengan demikian, tidur dapat meningkatkan daya ingat seseorang, sehingga ia dapat mengingat apa yang telah dipelajarinya. Tentunya hal tersebut akan berpengaruh terhadap prestasi akademik seseorang.

Selain terkait masalah daya ingat, kekurangan tidur juga dihubungkan dengan lemahnya kemampuan konsentrasi dan kognisi seseorang. Kekurangan tidur bukan hanya menyebabkan rasa kantuk dan sakit kepala, melainkan melemahkan kemampuan kognisi seseorang, sehingga ia mengalami kesulitan untuk memahami hal-hal yang seharusnya dapat ia pahami.

Beberapa penelitian yang telah dilakukan, pada umumnya menggunakan ukuran subjektif untuk pencatatan durasi tidur dan kualitas tidur seseorang. Data dikumpulkan melalui laporan masing-masing siswa-siswi yang terlibat dalam penelitian tersebut. Untuk mengatasi faktor subjektivitas tersebut, penelitian pada (Okano, 2019) menggunakan perangkat *smartwatch* Fitbit (Gambar 2.2).



Gambar 1.2. Smartwatch Fitbit⁵

Selain sebagai penunjuk waktu, *smartwatch* Fitbit dapat berperan sebagai perekam data aktivitas (*activity tracker*) penggunanya. Fitbit menggunakan data berisi kombinasi dari gerakan dan pola detak jantung untuk mengestimasi durasi dan kualitas tidur seseorang. Sebagai contoh, untuk menentukan durasi tidur seseorang, Fitbit mengukur waktu selama penggunaan tidak bergerak dan dikombinasikan dengan gerakan yang umum terjadi saat tidur, misalnya badan yang berputar untuk berganti posisi tidur. Untuk menentukan kualitas tidur seseorang, Fitbit menggunakan fluktuasi hasil pengukuran detak jantung yang

⁵ <https://www.fitbit.com/us/products/smartwatches>

dapat menggambarkan tahapan tidur yang berbeda-beda. Contoh hasil pencatatan aktivitas tidur seseorang dapat dilihat pada Gambar 2.3.

Penelitian (Okano, 2019) mengumpulkan data kuantitatif yang didapatkan melalui Fitbit dari hampir 100 orang mahasiswa, yang bertujuan untuk mendapatkan relasi ukuran objektif dari durasi tidur, kualitas tidur, dan konsistensi hasil ujian-ujian di sebuah universitas. Selain itu, penelitian tersebut bertujuan untuk mengetahui efek dari perbedaan *gender* dalam relasi tidur dan prestasi akademik.



Gambar 2.2. Pencatatan aktivitas tidur pada Fitbit⁶

Ada penelitian lain yang mengatakan bahwa prestasi wanita lebih baik daripada pria dalam banyak mata pelajaran dan bahkan dalam hasil *online learning*. Hal ini biasanya dikaitkan dengan konsistensi wanita dalam mengatur waktu dan kedisiplinan wanita, tetapi belum banyak penelitian yang mengaitkan “kontribusi” tidur wanita terhadap prestasi akademik mereka. Oleh karena itu, penelitian (Okano, 2019) juga bertujuan untuk mengetahui “kontribusi” tidur dari *gender* yang berbeda terhadap prestasi akademik mereka.

2.2. Konsep Statistika

Pada bagian ini berbagai konsep statistika yang digunakan pada artikel ini.

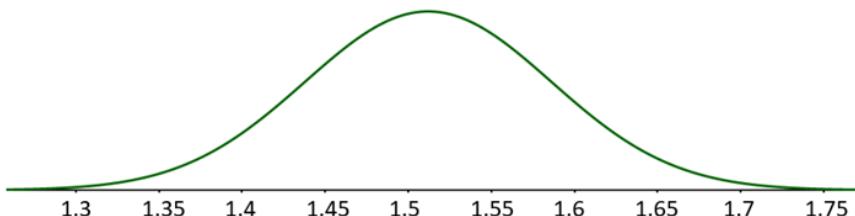
Distribusi Normal

Sebuah peubah acak (*random variable*) dikatakan berdistribusi normal jika sebagian besar nilai dari peubah acak itu berada di dekat nilai rata-rata dari peubah acak tersebut. Semakin jauh sebuah nilai dari nilai rata-rata, nilai tersebut akan semakin jarang muncul dan banyaknya kemunculan nilai peubah acak

⁶ <https://blog.fitbit.com/sleep-study>

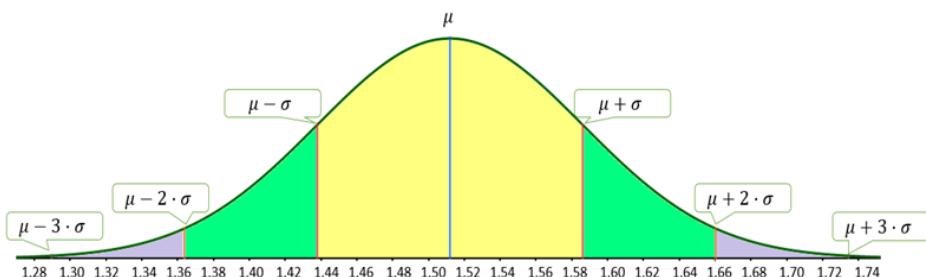
di posisi kiri dan kanan rata-rata simetri. Sangat banyak fenomena yang memiliki distribusi normal, misalnya saja tinggi badan wanita pada umur tertentu. Umumnya wanita berumur 14 tahun memiliki tinggi badan 152 cm dengan simpangan baku 7.41 cm. Tentu ada wanita yang berumur 14 tahun dan memiliki tinggi badan di atas atau di bawah 152 cm. Tetapi semakin jauh dari 152, banyaknya wanita yang memiliki tinggi badan tersebut akan semakin sedikit. Sangat jarang kan kita menemukan wanita berumur 14 tahun yang bertinggi badan hanya 100 cm? Atau wanita berumur 14 tahun yang bertinggi badan 200 cm?

Kurva untuk distribusi normal dapat dilihat pada Gambar 2.4. Sumbu-x pada gambar itu menyatakan tinggi badan wanita yang berumur 14 tahun (dalam meter), sementara ketinggian kurva pada masing-masing nilai x menyatakan kepadatan peluang pada nilai x . Apa itu kepadatan peluang? Agar mudah memahaminya, ketinggian kurva tersebut pada suatu nilai x dapat dibayangkan sebagai proporsi banyaknya wanita berumur 14 tahun yang memiliki tinggi x pada suatu populasi.



Gambar 2.4. Kurva distribusi normal untuk tinggi badan wanita berumur 14 tahun.

Kurva pada distribusi normal selalu memiliki bentuk seperti gunung, di mana kurva distribusi normal berpuncak pada nilai rata-rata dan semakin mengecil pada kiri dan kanan secara simetri. Pada kumpulan data yang memiliki distribusi normal, simpangan baku memiliki peran penting. Perhatikan Gambar 2.5 untuk melihat peran dari simpangan baku.



Gambar 2.3. Ilustrasi rata-rata dan simpangan baku dengan distribusi untuk tinggi badan wanita berumur 14 tahun ($\mu = \text{rata-rata}$, $\sigma = \text{simpangan baku}$).

Sekitar 68% wanita akan memiliki tinggi badan pada rentang rata-rata-simpangan baku hingga rata-rata+simpangan baku. Sekitar 95% wanita akan memiliki tinggi badan pada rentang rata-rata \pm 2·simpangan baku hingga rata-rata \pm 2·simpangan baku. Sekitar 99.7% wanita akan memiliki tinggi badan pada rentang rata-rata \pm 3·simpangan baku hingga rata-rata \pm 3·simpangan baku. Dengan mengetahui rata-rata dan simpangan baku, kita dapat mengetahui perkiraan rentang untuk nilai peubah acak yang dimiliki.

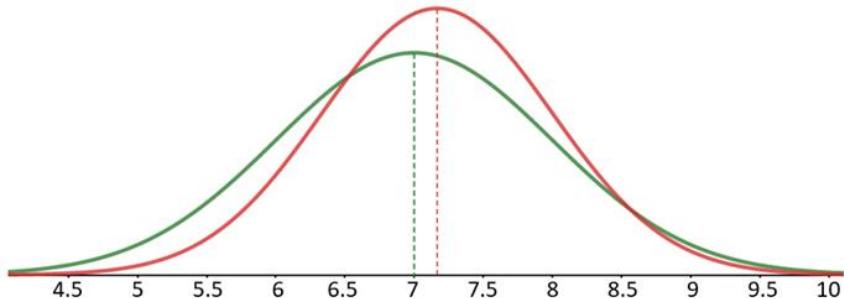
Mengapa kita menyimpan informasi simpangan baku dan tidak nilai maksimum minimum untuk mengetahui rentang dari nilai peubah acak yang dimiliki? Seringkali data yang kita dapatkan memiliki nilai yang cukup ekstrim, misalnya terlalu besar atau terlalu kecil dibandingkan dengan nilai peubah acak yang lain. Misalkan terdapat satu anak dengan tinggi 190 cm atau satu anak dengan tinggi 120 cm. Adanya nilai yang ekstrim ini akan membuat rentang nilai dari peubah acak terlihat besar padahal tidak banyak yang memiliki nilai sebesar atau sekecil itu (seringkali hanya sekian persen dari populasi). Oleh karena itu, kita biasa mengambil rentang rata-rata \pm 2·simpangan baku yang memuat sekitar 95% data.

Statistically Significant

Pernahkah pembaca mendengar pernyataan-pernyataan seperti “minuman manis tingkatkan risiko kematian dini”? Pernyataan tersebut bukan hasil penelitian terhadap kandungan zat pada minuman dan efeknya pada biologis tubuh seseorang. Tetapi pernyataan tersebut ternyata hasil dari penelitian terhadap data gaya hidup beberapa orang. Dari orang-orang yang diteliti, didapatkan hasil orang-orang yang punya gaya hidup suka minuman manis umurnya yang lebih pendek dibandingkan umur orang yang tidak meminum minuman manis.

Hasil penelitian seperti ini memanfaatkan data dan statistik untuk mengambil kesimpulan. Tentu penelitian ini tidak dilakukan kepada seluruh orang di dunia (bayangkan berapa banyak biaya dan waktu yang dibutuhkan jika dilakukan penelitian kepada semua orang di bumi). Karena tidak mungkin dilakukan kepada seluruh manusia, maka hampir seluruh penelitian melakukan pengambilan sampel, dimana data dikumpulkan dari sampel ini. Hasil analisis dari sampel ini diharapkan dapat menggambarkan keadaan sesungguhnya dari sebuah populasi.

Karena hasil analisis hanya berdasarkan pengambilan sampel, tentu hasil yang diperoleh tidak 100% tepat. Misalkan saja, ingin diketahui apakah durasi tidur wanita lebih sedikit daripada pria. Untuk itu, dilakukan penelitian terhadap 40 orang wanita dan 40 orang pria yang dipilih secara acak. Mereka diminta untuk mencatat berapa durasi tidur mereka biasanya. Dari hasil survei, diketahui rata-rata durasi tidur 40 wanita tersebut adalah 7 jam dan rata-rata durasi tidur dari 40 pria tersebut adalah 7 jam 10 menit. Apakah dapat dikatakan perbedaan tersebut cukup signifikan sehingga dapat disimpulkan durasi tidur wanita memang lebih sedikit daripada durasi tidur pria?

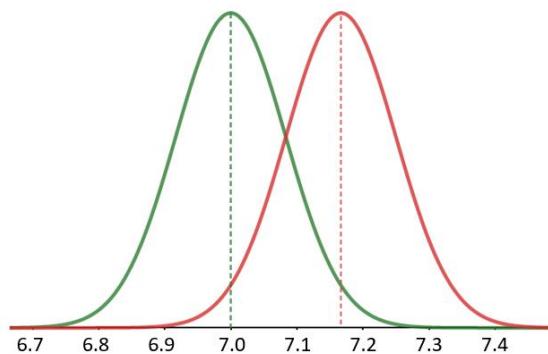


Gambar 2.4. Kurva frekuensi untuk durasi tidur pria (merah) dan wanita (hijau).

Tentu tidak dapat langsung disimpulkan signifikansi hasil penelitian tersebut jika hanya dilihat dari rata-ratanya saja. Bayangkan jika ternyata simpangan baku dari durasi tidur wanita dan pria yang diteliti ternyata besar (1 jam untuk wanita dan 50 menit untuk pria) seperti pada Gambar 2.6.

Pada Gambar 2.6, garis berwarna merah merupakan kurva frekuensi untuk durasi tidur pria sedangkan garis berwarna hijau merupakan kurva frekuensi untuk durasi tidur wanita. Berdasarkan rata-ratanya, durasi tidur pria mungkin lebih banyak daripada durasi tidur wanita. Tetapi jika dilihat dari simpangan baku, perbedaan rata-ratanya tidak terlalu signifikan (hanya 10 menit). Apakah 10 menit perbedaan bisa menjadi signifikan?

Sekarang andaikan simpangan bakunya adalah 5 menit untuk wanita dan pria. Gambar 2.7 memberikan ilustrasi kurva frekuensi terhadap durasi tidur.



Gambar 2.5. Kurva Frekuensi untuk Durasi Tidur Pria (Merah) dan Wanita (Hijau)

Dengan melihat Gambar 2.7, dapat ditarik kesimpulan bahwa perbedaan 10 menit durasi tidur wanita dengan pria cukup signifikan. Tentu saja kita tidak dapat hanya mempercayai penglihatan kita terhadap grafik. Ada teknik dalam statistik untuk menguji seberapa signifikan hasil analisis terhadap hasil sampel

(ANOVA, t-test, dll). Tetapi kita tidak akan membahas detil masing-masing teknik tersebut. Kita akan membahas sebenarnya bagaimana tes tersebut dilakukan secara intuitif.

Untuk menentukan apakah hasil analisis signifikan secara statistik atau tidak, idenya adalah melihat rata-rata dari masing-masing peubah acak, simpangan baku dari masing-masing peubah acak, distribusi dari masing-masing peubah acak, dan banyaknya sampel yang diambil. Semakin besar perbedaan rata-rata tentu akan membuat hasil perbedaan semakin signifikan. Semakin kecil simpangan baku akan membuat hasil pengujian juga semakin signifikan. Banyaknya sampel juga memiliki peran dalam pengujian ini. Semakin besar sampel yang diambil, akan membuat hasil pengujian semakin signifikan.

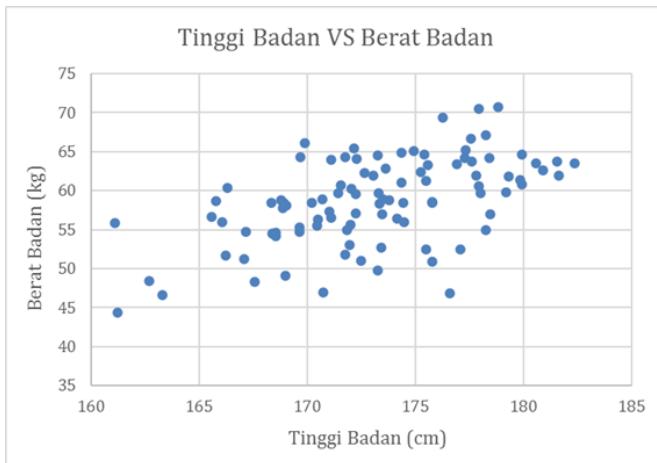
Pada penelitian (Okano, 2019), diberikan nilai dari [p-value](#). Nilai p-value menyatakan persen resiko kesimpulan yang diambil salah. Tentu sana nilai p-value dihitung berdasarkan rata-rata, simpangan baku, banyaknya sampel, dan distribusi nilai peubah acak yang diperoleh dari sampel. Dengan mengetahui p-value, kita dapat menentukan seberapa signifikan kesimpulan yang diambil. Semakin kecil nilai p-value, resiko kesimpulan yang diambil akan salah juga semakin kecil. Oleh karena itu, hasil kesimpulan dapat dikatakan signifikan secara statistik.

Umumnya batas nilai p-value adalah 0.05 (5%). Yang artinya, resiko kesimpulan yang diambil salah kurang dari atau sama dengan 5% saja dan hasil penelitiannya 95% signifikan. Tetapi batas nilai ini dapat berubah sesuai kebutuhan. Jika hasil penelitian memiliki resiko yang besar jika salah, maka tentu nilai batas kesalahannya harus diperkecil (misalnya pada pengujian larutan yang berbahaya).

Pearson Correlation

Korelasi adalah nilai yang digunakan untuk melihat hubungan antara dua buah peubah acak numerik. Misalnya ingin diketahui hubungan antara tinggi badan dengan berat badan seseorang. Kita dapat membuat sebuah visualisasi untuk melihat hubungan antara tinggi badan dengan berat badan. Gambar 2.8 memperlihatkan plot tinggi dan berat badan untuk beberapa orang.

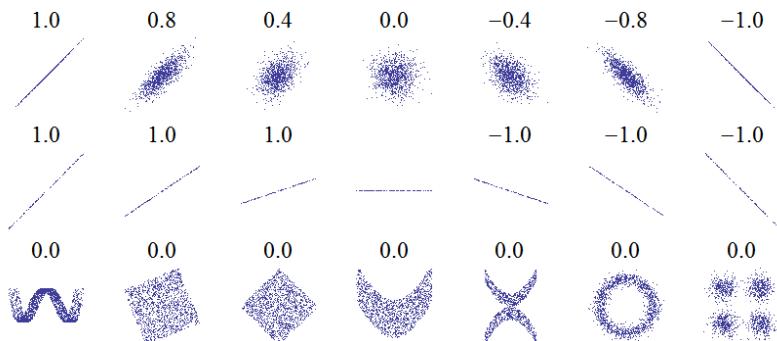
Sebuah titik pada Gambar 2.8 menyatakan tinggi badan dan berat badan dari satu orang. Walaupun titik-titik pada Gambar 2.8 tidak membentuk garis lurus, tetapi kita dapat melihat bahwa terdapat kecenderungan bahwa orang yang memiliki tinggi badan yang lebih tinggi biasanya memiliki berat badan yang lebih berat. Dari titik-titik pada Gambar 2.8 dapat ditarik sebuah garis lurus yang mewakili titik-titik tersebut.



Gambar 2.6. Scatter plot berat badan terhadap tinggi sampel orang⁷.

Seberapa tepat garis lurus mewakili titik-titik tersebut diukur oleh korelasi. Nilai korelasi berada di rentang -1 hingga 1. Nilai korelasi yang positif menyatakan kedua peubah acak numerik memiliki hubungan yang berbanding lurus, maksudnya kedua peubah acak akan saling bertambah besar bersamaan atau bertambah kecil bersamaan. Nilai korelasi yang negatif menyatakan kedua peubah acak numerik memiliki hubungan yang berbanding terbalik, maksudnya adalah saat sebuah peubah acak bertambah besar, peubah acak yang lain bertambah kecil, begitu pula sebaliknya.

Semakin dekat nilai korelasi dengan -1 atau 1 menyatakan titik-titik data semakin cocok dengan garis yang mewakili titik-titik data tersebut. Gambar 2.9 merupakan contoh-contoh nilai korelasi. Semakin tersebar sebuah data, maka korelasi semakin mendekati nilai nol.



Gambar 2.7. Contoh nilai korelasi⁸.

⁷ <https://www.kaggle.com/burnoutminer/heights-and-weights-dataset>

⁸ https://commons.wikimedia.org/wiki/File:Correlation_examples.png

2.3. Pengumpulan Data dari Peserta Kuliah

Hasil penelitian (Okano, 2019) menggunakan data yang dikumpulkan dari 100 mahasiswa/i. Mahasiswa/i yang mengikuti penelitian ini dipilih dari 370 peserta kuliah Pengenalan Kimia Zat Padat di MIT (*Massachusetts Institute of Technology*). Dari 100 sampel peserta (partisipan) tersebut, terdapat 47 mahasiswi (wanita) dan sisanya mahasiswa (pria). Banyaknya wanita dan pria yang dipilih hampir sama karena penelitian (Okano, 2019) juga ingin meneliti relasi gender terhadap kualitas tidur.

Seratus partisipan yang dipilih ini akan diminta untuk menggunakan *smartwatch* Fitbit selama satu semester saat mereka mengikuti kuliah tersebut. Kuliah Pengenalan Kimia Zat Padat terdiri dari kuliah mingguan yang diajar seorang profesor dan terdapat 2 minggu di mana siswa belajar bersama asisten dosen. Terdapat 12 asisten dosen yang mengajar pada kelas ini. Seorang partisipan akan mendapat hanya satu dari 12 asisten dosen selama semester tersebut. Para partisipan ini mengikuti kuis setiap minggu, tiga ujian pada pekan ujian, dan sebuah ujian akhir. Nilai-nilai inilah yang diambil sebagai representasi prestasi akademik pada penelitian (Okano, 2019).

Para partisipan yang dipilih untuk dicatat prestasinya sebisa mungkin mendapat “perlakuan” yang sama agar meminimalisir kemungkinan adanya faktor luar yang mempengaruhi hasil prestasi mereka. Setiap minggu mereka selalu mendapat kuliah dari profesor yang sama dan mengikuti ujian serta kuis-kuis yang sama, hal luar yang membedakan hanyalah 12 asisten dosen. Untuk mendapatkan hasil penelitian yang benar, maka harus dicari tahu apakah “kualitas” mengajar 12 asisten ini cukup sama. Jika kualitas mengajar 12 asisten ini cukup sama, maka perbedaan asisten yang diperoleh siswa tidak menjadi faktor yang mengakibatkan adanya perbedaan prestasi.

Bagaimana cara untuk mengetahui apakah perbedaan asisten dosen akan mempengaruhi prestasi yang diperoleh seorang siswa? Untuk mengetahuinya, penelitian ini melakukan sebuah pengujian terhadap hasil prestasi yang diperoleh siswa-siswi yang diajar oleh 12 asisten dosen tersebut. Jika tidak terdapat perbedaan yang signifikan dari hasil para mahasiswa/i yang diajar oleh asisten dosen yang berbeda, maka dapat disimpulkan perbedaan asisten dosen tidak mempengaruhi prestasi dari mahasiswa/i tersebut.

Dari hasil pengujian terhadap hasil para mahasiswa/i yang diajar oleh 12 asisten tersebut, ternyata tidak terdapat perbedaan yang signifikan dari hasil para mahasiswa/i yang diajar oleh asisten yang berbeda-beda. Dari hasil pengujian tersebut, maka kita dapat yakin para mahasiswa/i yang ikut penelitian ini mendapatkan perlakuan yang mirip sekali dari sisi pengajaran.

Pada akhir penelitian, hanya hasil dari 88 partisipan (mahasiswa/i) yang digunakan untuk penelitian (Okano, 2019). Tujuh partisipan tidak jadi berpartisipasi karena mereka memakai perekam aktivitas yang diberikan kurang dari 80% waktu di semester percobaan tersebut. Tiga partisipan tidak diikutsertakan lagi dalam penelitian ini karena mereka menghilangkan perekam aktivitas yang diberikan kepada mereka. Dua partisipan lainnya tidak diikutsertakan karena keikutsertaan mereka dalam kuis dan ujian di bawah 75% dari banyaknya kuis dan ujian yang diberikan. Kurangnya data dari 12 partisipan ini menyebabkan

analisis relasi kualitas tidur dengan prestasi akademik akan kurang valid jika 12 data ini diikutsertakan. Dari data 88 partisipan yang data aktivitasnya digunakan untuk analisis, terdapat 45 data partisipan wanita dan sisanya adalah pria.

2.4. Hasil Analisis Data

Pada bagian ini, hasil analisis data dengan menggunakan teknik yang sudah dijelaskan sebelumnya, dideskripsikan.

Pengaruh Jam Tidur dan Bangun terhadap Prestasi Akademis

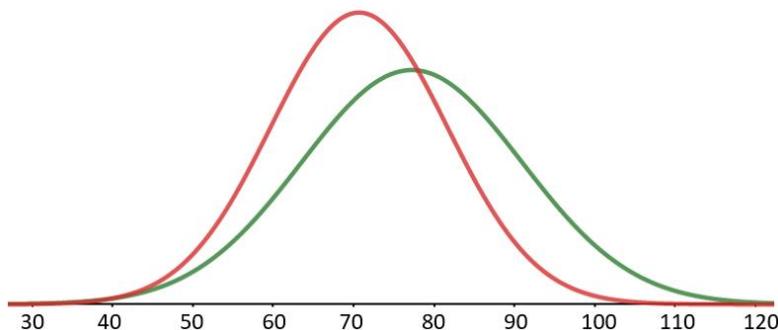
Dari hasil penelitian kepada 88 partisipan, diketahui nilai tengah dari jam tidur para partisipan tersebut adalah 1:47 a.m. Hal ini berarti setengah dari partisipan tidur sebelum atau tepat pukul 1:47 a.m dan setengah lagi tidur setelah atau tepat pada pukul 1:47 a.m. Rata-rata jam tidur dari para partisipan ini adalah pukul 1:54 a.m dengan simpangan baku 2 jam 11 menit. Data waktu tidur 88 partisipan ini diketahui memiliki distribusi normal (sebagian besar siswa tidur di sekitar pukul 1:54 a.m dan memiliki kurva frekuensi berbentuk gunung seperti yang sudah dijelaskan di bagian distribusi normal).

Penelitian (Okano, 2019) melakukan analisis untuk mengetahui apakah terdapat perbedaan yang signifikan dari nilai-nilai siswa yang tidur sebelum pukul 1:47 a.m (nilai tengah) dengan siswa yang tidur setelah pukul 1:47 a.m. Dari data yang dimiliki, dapat ditentukan rata-rata dan simpangan baku dari para partisipan yang tidur sebelum pukul 1:47 a.m dengan partisipan yang tidur setelah pukul 1:47 a.m. Nilai rata-rata dan simpangan baku tersebut diberikan pada Tabel 2.1.

Tabel 2.1. Rata-rata dan simpangan baku partisipan yang tidur sebelum dan setelah pk 1:47 a.m

Waktu tidur	Rata-rata	Simpangan baku
<=1:47 a:m	77.25	13.71
>=1:47 a:m	70.68	11.01

Dari rata-ratanya terlibat partisipan yang tidur sebelum pukul 1:47 a.m mendapat hasil yang lebih baik daripada partisipan yang tidur setelah pukul 1:47 a.m. Perbedaan rata-rata ini cukup signifikan mengingat simpangan baku yang tidak terlalu besar. Perhatikan Gambar 2.10 untuk mendapat ilustrasi tentang data nilai-nilai siswa yang tidur sebelum pukul 1:47 dan yang tidur setelah pukul 1:47.



Gambar 2.8. Ilustrasi kurva frekuensi nilai siswa yang tidur sebelum (hijau) dan setelah pukul 1:47 (merah)

Pada Gambar 2.10, garis merah menyatakan frekuensi dari nilai-nilai siswa yang tidur setelah 1:47, sementara garis hijau menyatakan frekuensi dari nilai-nilai siswa yang tidur sebelum 1:47 (Gambar ini tidak berdasarkan data sesungguhnya, tetapi menggunakan simulasi berdasarkan nilai rata-rata, simpangan baku, dan distribusi datanya. Oleh karena itulah, terdapat nilai yang lebih besar daripada 100). Berdasarkan Gambar 2.10 terlihat kurva merah lebih kiri daripada kurva hijau, sehingga dapat terlihat siswa yang tidur setelah pukul 1:47 mendapat prestasi yang lebih rendah daripada siswa-siswa yang tidur sebelum 1:47. Selain itu, nilai p-value dari perbedaan rata-rata ini adalah 0.01 yang menyatakan bahwa perbedaan nilai rata-rata dari siswa yang tidur sebelum 1:47 a.m dengan siswa yang tidur setelah 1:47 a.m signifikan secara statistik.

Dari hasil analisis nilai para partisipan yang tidur sebelum dan setelah pukul 1:47, dapat disimpulkan jam tidur memiliki pengaruh terhadap hasil prestasi seorang siswa. Untuk lebih meyakinkan hasil ini, penelitian (Okano, 2019) juga melakukan analisis korelasi antara jam tidur dengan prestasi siswa. Berdasarkan analisis korelasi, diketahui jam tidur dengan prestasi siswa memiliki korelasi negatif. Artinya, semakin cepat partisipan tidur, prestasi partisipan tersebut akan semakin bagus. Hasil ini mendukung hasil dari analisis perbedaan prestasi siswa berdasarkan jam tidur sebelum dan setelah pukul 1:47 a.m.

Berdasarkan analisis terhadap jam tidur, dapat disimpulkan jam tidur memiliki pengaruh terhadap prestasi akademik seseorang. Apakah jam bangun seorang siswa juga memiliki pengaruh terhadap prestasi akademik seseorang?

Dari hasil penelitian kepada 88 partisipan, diketahui nilai tengah dari jam bangun para partisipan tersebut adalah 9:12 a.m. Rata-rata jam bangun dari para partisipan ini adalah pukul 9:17 a.m dengan simpangan baku 2 jam 2 menit. Data waktu tidur 88 partisipan ini diketahui memiliki distribusi normal juga. Sama seperti analisis terhadap pengaruh jam tidur terhadap prestasi akademik seseorang, untuk mengetahui apakah jam bangun memiliki pengaruh terhadap prestasi akademik seseorang, penelitian (Okano, 2019) membandingkan prestasi akademik partisipan yang bangun sebelum pukul 9:12 a.m dengan partisipan yang bangun setelah pukul 9:12 a.m.

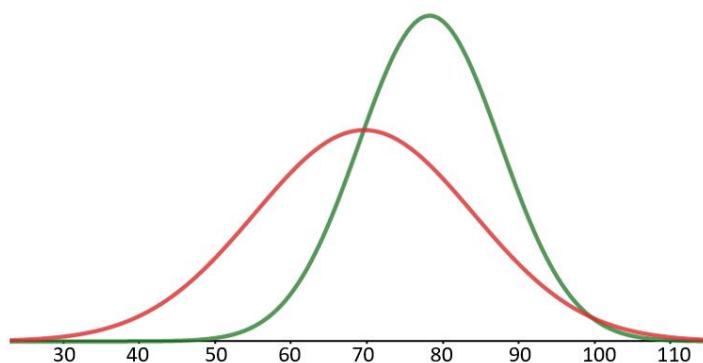
Dari data yang dimiliki, dapat ditentukan rata-rata dan simpangan baku dari partisipan yang bangun sebelum pukul 9:12 a.m dengan partisipan yang bangun setelah pukul 9:12 a.m. Nilai rata-rata dan simpangan baku tersebut diberikan pada Tabel 2.2.

Tabel 2.2. Rata-rata dan simpangan baku siswa yang bangun sebelum dan setelah pk 9:12 a.m

Waktu bangun	Rata-rata	Simpangan baku
<=9:12 a:m	78.28	9.33
>=9:12 a:m	69.63	14.38

Berdasarkan nilai rata-rata, dapat dilihat bahwa partisipan yang bangun lebih pagi (sebelum pukul 9:12) mendapatkan nilai yang lebih tinggi daripada siswa-siswa yang bangun lebih siang (setelah pukul 9:12). Simpangan baku untuk masing-masing kelompok juga cukup kecil sehingga hasil perbedaan rata-rata cukup signifikan secara statistik.

Pada Gambar 2.11, garis merah menyatakan frekuensi dari nilai-nilai partisipan yang bangun setelah 9:12, sementara garis hijau menyatakan frekuensi dari nilai-nilai partisipan yang bangun sebelum 9:12 (Gambar ini tidak berdasarkan data sesungguhnya, tetapi menggunakan simulasi berdasarkan nilai rata-rata, simpangan baku, dan distribusi datanya. Oleh karena itulah, terdapat nilai yang lebih besar daripada 100). Berdasarkan Gambar 2.11, terlihat kurva merah lebih kiri daripada kurva hijau, sehingga dapat terlihat partisipan yang bangun lebih siang mendapat prestasi yang lebih rendah. Selain itu, nilai p-value dari perbedaan rata-rata ini adalah 0.01 yang menyatakan bahwa perbedaan nilai rata-rata dari partisipan yang bangun sebelum 9:12 a.m dengan partisipan yang bangun setelah 9:12 a.m signifikan secara statistik.



Gambar 2.9. Ilustrasi kurva frekuensi nilai partisipan yang bangun sebelum (hijau) dan setelah pukul 9:12 (merah).

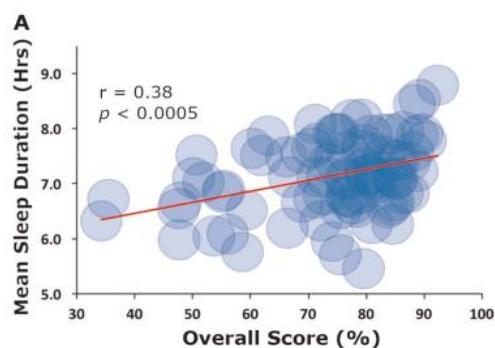
Untuk lebih meyakinkan mengenai apakah benar seseorang yang bangun lebih pagi cenderung mendapat nilai yang lebih baik daripada orang yang bangun lebih siang, penelitian (Okano, 2019) juga melakukan analisis korelasi antara jam bangun dengan prestasi akademik siswa. Jam bangun dengan prestasi akademik ternyata memiliki korelasi negatif. Artinya, semakin cepat seseorang bangun, semakin baik prestasi yang dimiliki orang tersebut dibandingkan dengan orang yang lebih lambat bangun.

Hasil analisis korelasi jam bangun dengan prestasi akademik mendukung hasil analisis dari perbedaan prestasi siswa berdasarkan perbedaan jam bangunnya. Oleh karena itu, dapat disimpulkan bahwa **jam bangun juga memiliki pengaruh terhadap prestasi akademik partisipan**. Dengan melihat korelasi antara jam tidur dan jam bangun, diketahui juga partisipan yang tidur lebih cepat cenderung bangun lebih pagi. Sehingga dapat dikatakan partisipan yang tidur lebih cepat akan bangun lebih pagi dan mendapatkan prestasi akademik yang lebih baik dibandingkan terhadap siswa yang tidur lebih larut.

Pengaruh Durasi Tidur terhadap Prestasi Akademis

Setelah membahas mengenai relasi jam tidur dan jam bangun terhadap prestasi seseorang, sekarang kita akan membahas mengenai pengaruh durasi tidur terhadap prestasi seseorang. Mungkin bagian ini akan menjadi bagian yang disukai oleh para pelajar karena akan disajikan analisis-analisis yang menjawab pertanyaan dari judul pada artikel ini. **Apakah seorang pelajar salah jika memiliki tidur yang cukup?**

Untuk mengetahui apakah durasi tidur seseorang memiliki pengaruh terhadap prestasi akademik atau tidak, data yang digunakan adalah rata-rata durasi tidur seseorang selama satu semester dan total nilai siswa pada mata kuliah yang pernah dijelaskan sebelumnya pada semester tersebut. Gambar 2.12 menunjukkan hubungan rata-rata durasi tidur seseorang dengan total nilai yang diperoleh oleh partisipan tersebut.



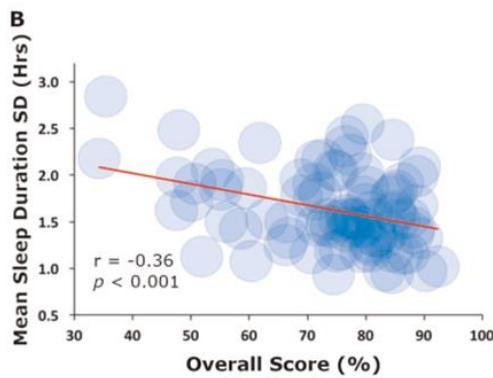
Gambar 2.10. Relasi rata-rata durasi tidur dan nilai peserta kuliah yang menjadi partisipan penelitian (Okano, 2019)

Pada Gambar 2.12, masing-masing titik menyatakan satu orang partisipan. Walaupun ada partisipan yang mendapat nilai bagus walaupun durasi tidurnya kecil, tetapi jika diambil sebuah garis lurus yang mewakili titik-titik tersebut, dapat dilihat bahwa ada kecenderungan partisipan yang memiliki durasi tidur semakin banyak mendapat nilai yang lebih baik. Nilai r yang berada di grafik tersebut menyatakan korelasi antara durasi dengan nilai. Nilai $r = 0.38$ menyatakan korelasi di antara keduanya positif, yaitu semakin tinggi durasi, semakin tinggi pula nilai yang didapat.

Selain durasi, penelitian (Okano, 2019) juga meneliti pengaruh kekonsistenan tidur seseorang terhadap prestasi akademik. Apakah konsistensi tidur seseorang dalam satu semester itu memiliki pengaruh terhadap prestasi akademik yang diperoleh? Konsistensi tidur diukur berdasarkan simpangan baku dari durasi tidur masing-masing partisipan selama satu semester. Semakin besar simpangan baku dari durasi tidur seseorang, artinya orang tersebut tidak konsisten tidurnya.

Setelah mengukur simpangan baku dari durasi tidur masing-masing partisipan, penelitian (Okano, 2019) menyajikan relasi antara kekonsistenan tidur seseorang yang dilambangkan oleh simpangan baku dengan nilai yang didapat oleh orang tersebut melalui plot pada Gambar 2.13.

Dari Gambar 2.13, dapat dilihat bahwa semakin kecil simpangan baku seorang partisipan, semakin tinggi nilai yang diperoleh partisipan tersebut. Hal ini dapat dilihat dari garis lurus yang mewakili keseluruhan data dan nilai korelasi $r = -0.36$. Nilai korelasi yang negatif menyatakan semakin kecil nilai simpangan baku, semakin besar nilai yang diperoleh. Semakin kecil simpangan baku dari durasi tidur seseorang menyatakan orang tersebut memiliki durasi tidur yang konsisten selama semester tersebut. Sehingga dapat disimpulkan ternyata ketidakkonsistenan dalam durasi tidur berpengaruh terhadap kurangnya prestasi akademik.



Gambar 2.11. Relasi Konsistensi Tidur dan Nilai Siswa (Okano, 2019).

Walaupun durasi dan kualitas tidur memiliki pengaruh terhadap prestasi yang diperoleh, tetapi ternyata durasi dan kualitas tidur seseorang di malam sebelum ujian tidak memiliki hubungan dengan prestasi yang diperoleh. Korelasi antara durasi tidur dengan hasil ujian keesokan harinya kurang dari 0.2. Karena

korelasinya sangat kecil, maka tidak dapat disimpulkan bahwa durasi tidur malam sebelum ujian dengan hasil ujian memiliki hubungan. Begitu pula dengan kualitas tidur di malam sebelum ujian.

Pola Tidur Pria dan Wanita Berpengaruh terhadap Perbedaan Prestasi

Berdasarkan hasil pencatatan dari Fitbit, didapatkan bahwa wanita memiliki kualitas yang tidur yang lebih baik. Berdasarkan hasil pengujian, p-value untuk pernyataan ini adalah 0.01, sehingga perbedaan kualitas wanita dan pria cukup signifikan. Begitu pula dengan konsistensi tidur, wanita memiliki konsistensi tidur yang lebih baik pula sepanjang semester. Tetapi kelompok pria dan wanita tidak menunjukkan perbedaan yang besar dalam hal durasi tidur.

Kedua kelompok gender dalam penelitian ini menunjukkan korelasi yang kuat antara kualitas dan durasi tidur; korelasi pada kelompok pria antara kualitas dan durasi tidur adalah 0.85, korelasi pada kelompok wanita antara kualitas dan durasi tidur adalah 0.64. Korelasi yang lebih kuat pada pria dapat memberikan saran kepada kaum pria untuk tidur dalam durasi yang lebih lama agar mendapatkan kualitas tidur yang lebih baik. Ketidakkonsistenan waktu tidur dan kualitas tidur berkorelasi negatif pada pria ($r = -0.51$), namun berkorelasi positif pada wanita ($r = 0.29$). Hal tersebut dapat memberikan saran kepada para pria untuk membuat jadwal tidur yang lebih konsisten agar mendapatkan kualitas tidur yang lebih baik.

Secara umum, nilai yang didapatkan oleh wanita lebih baik daripada pria ($p = 0.01$). Berdasarkan analisis pengujian untuk melihat seberapa signifikan perbedaan kelompok pria dan wanita, didapatkan hasil bahwa baik pria maupun wanita tidak menunjukkan perbedaan signifikan pada nilai yang mereka dapatkan ketika dilakukan kontrol kualitas tidur ($p = 0.14$). Ketidakkonsistenan tidur dan nilai secara keseluruhan pada pria, berkorelasi negatif ($r = -0.44$), sedangkan pada wanita nilai korelasinya juga negatif tetapi cenderung kecil ($r = -0.13$). Dengan demikian, dapat ditarik saran agar kaum pria mengusahakan jadwal tidur yang lebih konsisten agar mendapatkan hasil akademis yang lebih baik. Selain konsistensi tidur, tidak ada perbedaan lagi yang ditunjukkan oleh dua kelompok gender tersebut.

2.5. Kesimpulan

Penelitian ini mendukung beberapa penelitian terdahulu yang menyatakan bahwa ada kaitan antara kualitas tidur terhadap prestasi akademik. Penelitian ini menunjukkan bahwa durasi tidur yang lebih lama dan konsistensi tidur menyebabkan menghasilkan prestasi akademik yang lebih baik. Kekuatan dari penelitian ini adalah telah digunakannya alat ukur objektif, yaitu Fitbit. Dua penelitian sejenis mengatakan bahwa durasi tidur yang lebih lama selama satu minggu sebelum ujian dan lima hari berturut-turut sebelum ujian menghasilkan peningkatan performa siswa dalam ujian.

Namun masih ada hal yang perlu menjadi perhatian dalam penelitian ini, bahwa nilai 1-10 yang diberikan oleh Fitbit, untuk menggambarkan kualitas tidur seseorang, belum pernah dinyatakan secara ilmiah

sebagai nilai pengukuran kualitas tidur yang valid. Selain itu, masih terdapat peluang yang mungkin dapat mempengaruhi performa akademik seseorang, yaitu faktor stress, kegelisahan, motivasi, dan masalah kepribadian yang lain. Masih terdapat begitu banyak peluang untuk menyempurnakan penelitian pada bidang pengukuran prestasi akademik seseorang.

Referensi

- (Okano, 2019) Okano, Kana. et al., "Sleep quality, duration, and consistency are associated with better academic performance in college students", npj Science of Learning, 2019.
- (Frost, 2020) Frost, Jim., "Normal Distribution in Statistics", <https://statisticsbyjim.com/basics/normal-distribution/#:~:text=The%20normal%20distribution%20is%20a,off%20equally%20in%20both%20directions> (diakses 20 Juni 2020)

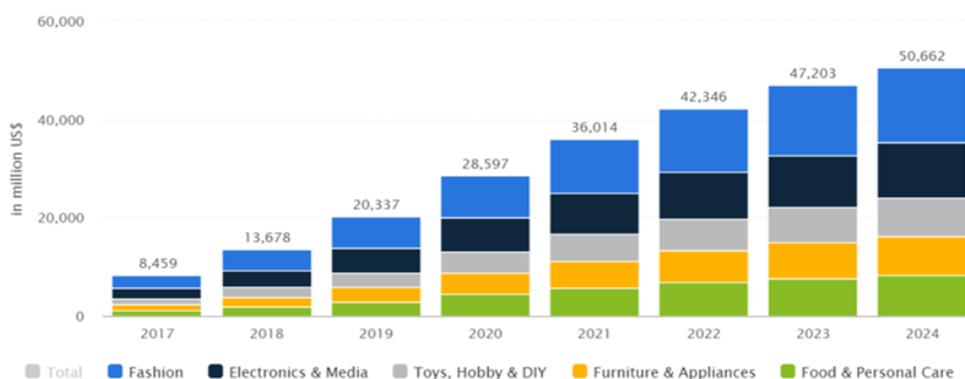
Bab 3 Pengenalan Sistem Rekomendasi pada e-Commerce

Oleh:

Mariskha Tri Adithia

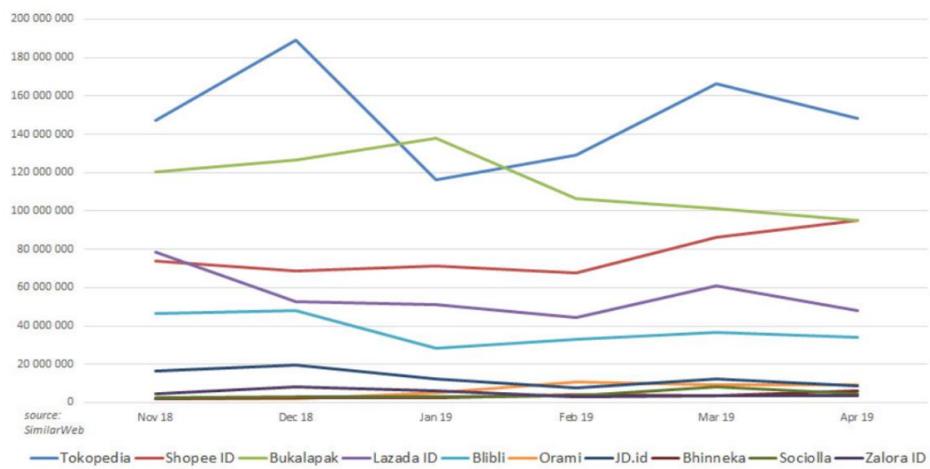
3.1. Pendahuluan

Belanja *online* saat ini sudah menjadi kebiasaan masyarakat dunia, termasuk Indonesia. Pendapatan perusahaan e-commerce di Indonesia pada tahun 2020 (data sampai bulan Juni 2020) mencapai 28,597,000 dollar [Statista, 2020]. Pendapatan ini terus meningkat sejak tahun 2017, seperti ditunjukkan Gambar 3.1. Jumlah masyarakat Indonesia yang sudah belanja *online* juga terus meningkat setiap tahunnya; mencapai 134.09 juta orang pada tahun 2020 ini, atau 50.5% seluruh penduduk Indonesia [Statista, 2020]. Di Indonesia, sudah banyak *platform e-commerce* lokal dengan *traffic* sangat tinggi, misalnya, Tokopedia, Shopee, dan Lazada. Pada Gambar 3.2, diberikan 10 *platform e-commerce* Indonesia, dengan perkiraan *traffic* bulanan tertinggi [Aseanup, 2019]. Posisi pertama dipegang oleh Tokopedia, dengan rata-rata jumlah kunjungan sebesar 148,500,000, pada periode bulan November 2018 sampai April 2019.



Gambar 3.1. Pendapatan perusahaan e-commerce di Indonesia sampai tahun 2020, dan prediksinya sampai tahun 2024 [Statista, 2020].

Belanja *online* dipilih karena kemudahannya, di mana konsumen tidak perlu pergi ke luar rumah, melalui kemacetan, dan mencari tempat parkir untuk dapat berbelanja. Selain itu, dengan berbelanja *online*, barang yang susah didapat di suatu kota, dapat dengan mudah dibeli di kota lain, hanya dengan menambah ongkos kirim saja. Belanja *online* juga sudah terasa seperti belanja di toko fisik karena kemudahan melihat berbagai barang berbeda dan pilihan metode pembayaran yang beragam.



Gambar 3.2. Top 10 platform e-commerce dengan perkiraan traffic bulanan tertinggi [Aseanup, 2019]

Sadar atau tidak, saat berbelanja *online*, toko *online* atau *platform e-commerce*, biasanya merekomendasikan beberapa barang yang sesuai dengan ketertarikan konsumen. Barang-barang yang direkomendasikan, biasanya terkait dengan barang yang sedang dicari konsumen saat itu, atau yang pernah dibeli pada masa sebelumnya. Bagaimana *platform e-commerce* dapat memberikan rekomendasi barang ini, dan lebih lagi, bagaimana *platform e-commerce* dapat memberikan rekomendasi yang tepat? Jawabannya adalah sistem rekomendasi.

Sistem rekomendasi adalah sistem yang memfilter informasi untuk memprediksi preferensi konsumen terhadap suatu barang. Di bidang e-commerce, sistem rekomendasi ini digunakan untuk mempersonalisasi *platform e-commerce* untuk setiap konsumen. Rekomendasi ini misalnya dilakukan berdasarkan barang yang pernah dibeli sebelumnya, barang yang pernah dilihat, dan informasi demografis konsumen.

Algoritma yang digunakan pada sistem rekomendasi umumnya menentukan rekomendasinya dengan mencari sekumpulan konsumen lain yang pembelian dan penilaian barangnya sesuai atau mirip dengan pembelian dan penilaian barang dari pengguna. Data barang yang dibeli sekumpulan konsumen tadi, dikurangi dengan barang-barang yang sudah pernah dibeli oleh pengguna tersebut, dijadikan rekomendasi bagi pengguna.

Salah satu algoritma yang popular digunakan untuk sistem rekomendasi ini adalah *collaborative filtering*. *Collaborative filtering* adalah metode untuk membuat prediksi otomatis tentang ketertarikan seorang pengguna dengan mengumpulkan informasi preferensi atau ketertarikan banyak pengguna lain (kolaborasi) yang profilnya mirip dengan si pengguna.

Pada artikel ini, penggunaan *collaborative filtering* pada sistem rekomendasi akan dibahas. Pembahasan akan dilengkapi dengan contoh kasus, data, dan rekomendasi yang diberikan. Paparan di sini merupakan pengantar untuk mendapatkan pemahaman awal tentang komputasi rekomendasi pada sistem e-commerce yang nyata.

3.2. Sistem Rekomendasi dan Collaborative Filtering

Pada saat berbelanja *online*, *platform e-commerce* sering memberikan rekomendasi barang untuk masing-masing konsumen. Contohnya dapat dilihat pada Gambar 3.3, di mana sebuah *platform e-commerce* memberikan rekomendasi makanan ikan, lap tangan, dan polybag, kepada konsumen. Rekomendasi ini diberikan berdasarkan barang-barang yang banyak dibeli maupun dicari oleh konsumen tersebut. *Platform e-commerce* dapat memberikan rekomendasi yang tepat, yang sesuai dengan kebutuhan dan ketertarikan konsumen dengan menggunakan sistem rekomendasi, seperti yang dijelaskan di Bab 1.



Gambar 3.3. Contoh rekomendasi yang diberikan sebuah platform toko online⁹.

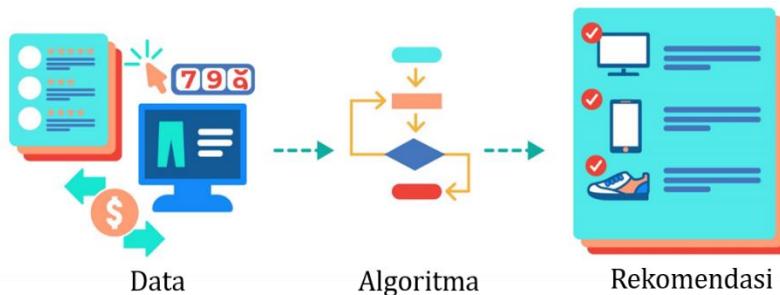
Sistem rekomendasi digunakan oleh *platform e-commerce* untuk membangun daftar rekomendasi untuk konsumennya, berdasarkan perilaku dan ketertarikan konsumen saat berbelanja atau mencari produk [Schafer, 2001]. Selain itu, rekomendasi ini misalnya dilakukan berdasarkan produk yang pernah dibeli sebelumnya, produk yang pernah dilihat, dan hal spesifik terkait pencarian, misalnya, artis favorit [Linden, 2003].

⁹ Gambar diambil dari screenshot akun Tokopedia penulis

Berdasarkan [Sivaplan, 1999], sistem rekomendasi pada *platform e-commerce* secara umum menggunakan data berikut:

- Rating suatu barang yang dibeli
- Pola perilaku, misalnya, durasi browsing dan jumlah klik
- Transaksi, misalnya, tanggal pembelian, kuantitas, dan harga
- Produksi, misalnya, merk barang, artis favorit, dan topik favorit

Data ini akan menjadi input bagi sistem rekomendasi dan diolah dengan menggunakan algoritma tertentu, sampai menghasilkan daftar rekomendasi untuk konsumen tertentu. Sebagai gambaran cara kerja sistem rekomendasi, lihat Gambar 3.4.



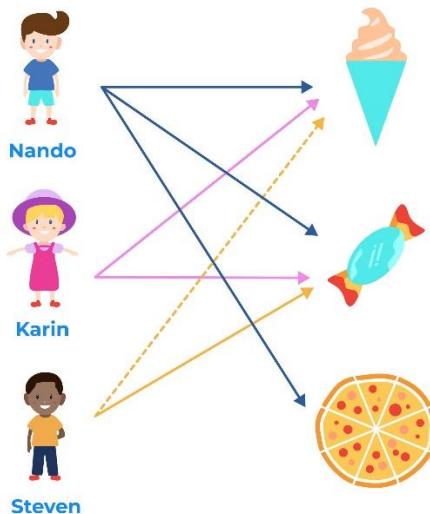
Gambar 3.4. Cara kerja sistem rekomendasi.

Ada beberapa jenis sistem rekomendasi. Misalnya, *personalized*, *non-personalized*, dan *people-to-people correlation*. Pada jenis sistem rekomendasi *personalized*, rekomendasi diberikan secara otomatis, berdasarkan kesukaan konsumen, misalnya, warna favorit, genre musik, dan genre film [Schafer, 2001]. Rekomendasi *non-personalized* diberikan hanya berdasarkan *rating* produk yang diberikan oleh konsumen lain [Schafer, 2001].

People-to-people correlation memberikan rekomendasi berdasarkan produk atau *rating* yang diberikan oleh konsumen lain, yang dianggap mirip dengan konsumen penerima rekomendasi [Sarwar, 2000]. Sistem jenis ini banyak menggunakan algoritma *collaborative filtering*, yaitu algoritma yang memprediksi ketertarikan seorang konsumen, berdasarkan kesukaan dan selera banyak konsumen lain. Kelebihan algoritma ini adalah pembangunan rekomendasi dilakukan dengan menggunakan data konsumen aktif saat ini, yang kesukaan dan karakternya mirip [Sivaplan, 2001]. Salah satu metode pemfilteran pada algoritma *collaborative filtering* adalah metode *item-based* [Sarwar, 2001].

Metode *item-based* menggunakan nilai *rating* yang diberikan oleh beberapa konsumen untuk dua produk yang sama, sebagai basis untuk merekomendasikan barang tersebut untuk konsumen lain. Metode ini bekerja berdasarkan asumsi, jika suatu produk mendapatkan *rating* yang baik dari seorang konsumen, maka konsumen lain dengan profil yang mirip, akan memberi *rating* yang baik pula untuk produk tersebut.

Gambar 3.5 memberikan gambaran tentang konsep *collaborative filtering item-based*. Pada gambar tersebut, dapat dilihat bahwa Nando suka permen, es krim, dan pizza. Sedangkan, Karin suka permen dan es krim. Steven menyukai permen. Karena Nando dan Karin, yang menyukai es permen, juga suka es krim, maka seharusnya Steven juga menyukai es krim.



Gambar 3.5. Ilustrasi collaborative filtering item-based.

Cara kerja sistem rekomendasi dengan algoritma *collaborative filtering item-based* adalah sebagai berikut [Sarwar, 2001]:

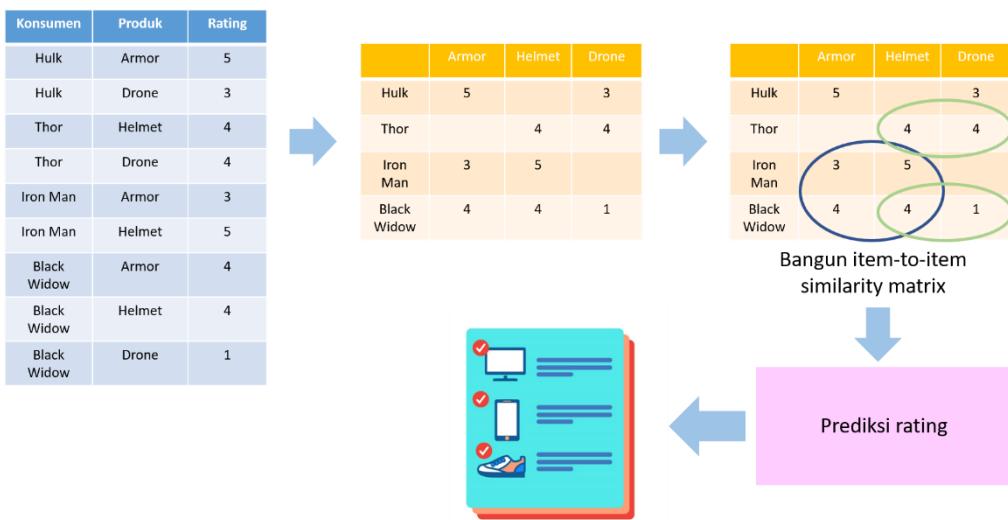
1. Mengubah data *rating* konsumen menjadi matriks *rating*. Baris matriks ini mewakili identitas konsumen, sedangkan kolomnya mewakili produk yang diberi *rating*.
2. Membuat *item-to-item similarity matrix*, yaitu matriks yang berisikan nilai kemiripan antara produk satu dengan lainnya. Nilai kemiripan ini, misalnya, didapatkan dengan menghitung korelasi Pearson, jarak Euclidean, atau *cosine similarity measure* antar produk. Hal ini dilakukan untuk mengukur kemiripan produk.
3. Menghitung prediksi *rating* konsumen untuk produk terkait, berdasarkan *similarity matrix* yang sudah didapat sebelumnya.
4. Membuat rekomendasi produk, berdasarkan prediksi *rating* produk, yang telah dihitung sebelumnya.

Langkah kerja ini secara sederhana ditunjukkan pada Gambar 3.6.

Algoritma *collaborative filtering item-based* ini, akan digunakan pada studi kasus di Bagian 3.4. Perhitungan dan contoh lebih rinci diberikan pada bagian tersebut.

3.3. Data e-Commerce

Platform e-commerce merekam berbagai data terkait perilaku konsumen saat *browsing* di *platform* tersebut. Perilaku konsumen dapat dilihat misalnya dari durasi *browsing*, berapa lama suatu produk dilihat, produk yang dilihat, dan pada jam berapa saja seorang konsumen aktif. Data transaksi pada *platform e-commerce* juga direkam. Data ini meliputi identitas konsumen, produk yang dibeli, harga produk, kuantitas, sampai *rating* konsumen terhadap produk tersebut. Semua perekaman ini dilakukan untuk menentukan strategi bisnis di masa mendatang, maupun penentuan strategi perbaikan layanan *platform* sendiri.



Gambar 3.6. Langkah kerja algoritma collaborative filtering item-based.

Terkait pemberian rekomendasi, seperti yang dijelaskan pada Bagian 1.2, data yang dibutuhkan adalah identitas konsumen, produk yang dibeli, dan *rating* yang diberikan konsumen untuk produk tersebut. Pada artikel ini, digunakan dataset dari toko *online* terkenal di Indonesia, yaitu Lazada Indonesia [Kaggle, 2019].

Dataset Lazada Indonesia ini terdiri atas 3 buah *file*, yaitu keterangan kategori produk, data keterangan tiap produk, dan data transaksi konsumen. Data ini hanya meliputi transaksi produk elektronik. Masing-masing data ini dijelaskan sebagai berikut:

- Pada keterangan kategori produk, diberikan kategori yang digunakan untuk mengelompokkan produk pada data lainnya. Terdapat lima kategori pada keterangan tersebut yaitu:
 - beli-harddisk-eksternal
 - beli-laptop
 - beli-smart-tv
 - jual-flash-drives

- shop-televi-digital
- Pada data keterangan tiap produk, termuat informasi berikut: identitas produk, kategori (sesuai dengan poin sebelumnya), nama produk, merk, URL, harga, *rating* rata-rata, jumlah *reviewer*, dan tanggal. Pada data ini, terdapat 4,426 *record*, atau dapat diartikan sebagai banyaknya jenis produk yang terjual. *Record* adalah tiap baris pada suatu data.
- Pada data transaksi, informasi yang terdapat di dalamnya, antara lain: identitas produk, kategori, identitas konsumen, *rating*, *review*, dan tanggal pembelian. Pada data ini, terdapat 203,787 *record*, yang direkam mulai tanggal 19 April 2014 sampai 2 Oktober 2019. Data ini adalah data yang akan digunakan untuk membuat rekomendasi nantinya. Beberapa keterangan khusus yang terkait sebagian informasi ini adalah:
 - Identitas produk berupa angka, dengan panjang 4-9 digit
 - Identitas konsumen berupa nama yang digunakannya pada *platform* tersebut.
 - *Rating* memiliki kisaran nilai 1 sampai 5.

Sebelum diolah lebih lanjut, seperti yang telah dijelaskan di Bab 1, harus diterapkan *data cleaning*, atau pembersihan data, terlebih dahulu terhadap data transaksi di atas. Dalam hal ini, *data cleaning* yang diterapkan adalah

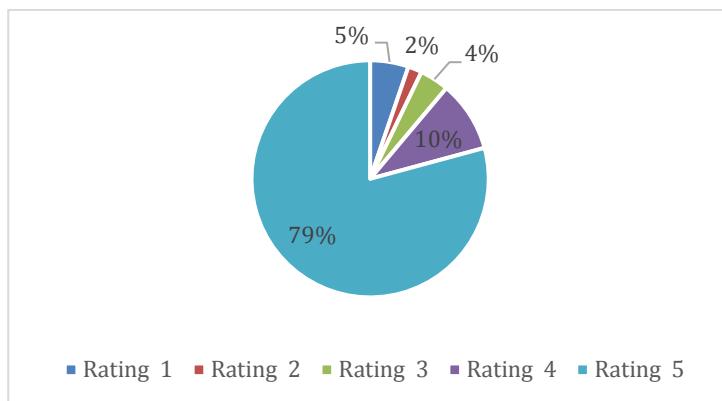
- Menghilangkan duplikasi *record*. Sebagai gambaran, lihat Gambar 3.7. Data transaksi ini mengandung banyak *record* yang sama, namun dituliskan berulang. Kesamaan dalam hal ini mencakup identitas konsumen, produk yang dibeli, isi review, sampai tanggal pembelian. Oleh karena itu, duplikasi *record* ini dihilangkan.
- Menghilangkan duplikasi produk. Suatu nama produk dapat dituliskan dengan banyak cara berbeda. Agar data akurat, maka duplikasi nama produk ini dihilangkan dari data transaksi.

1	A	B	C	D	E	F	G	M	N
itemID	category	name	rating	originalRa	reviewTitle	reviewContent	boughtDate		clientType
2	25850 beli-harddisk-eksternal	Riska	3	OKE	Everthing is OK..		19-Apr-14		desktop
3	25850 beli-smart-tv	Riska	3	OKE	Everthing is OK..		19-Apr-14		desktop
4	25850 jual-flash-drives	Riska	3	OKE	Everthing is OK..		19-Apr-14		desktop
5	25850 shop-televi-digital	Riska	3	OKE	Everthing is OK..		19-Apr-14		desktop
6	55752 beli-harddisk-eksternal	taufik n.	5	null	null		25-Mei-14		androidApp
7	55752 jual-flash-drives	taufik n.	5	null	null		25-Mei-14		androidApp
8	55752 beli-harddisk-eksternal	Sindy N.	5		sampe skrng di pi		16-Jun-14		androidApp
9	55752 jual-flash-drives	Sindy N.	5		sampe skrng di pi		16-Jun-14		androidApp
10	71175 beli-harddisk-eksternal	JUSTINUS H.	5	null	null		19-Jun-14		iosApp
11	71175 jual-flash-drives	JUSTINUS H.	5	null	null		19-Jun-14		iosApp
12	55752 beli-harddisk-eksternal	Hermulia S.	5	null	null		20-Jun-14		androidApp
13	55752 jual-flash-drives	Hermulia S.	5	null	null		20-Jun-14		androidApp
14	71175 beli-harddisk-eksternal	Justinus H.	5	null	null		25-Jun-14		iosApp
15	71175 jual-flash-drives	Justinus H.	5	null	null		25-Jun-14		iosApp
16	71175 beli-harddisk-eksternal	Andre H.	5	null	null		28-Jun-14		androidApp
17	71175 jual-flash-drives	Andre H.	5	null	null		28-Jun-14		androidApp
18	79301 beli-harddisk-eksternal	Agung L.	5	null	null		29-Jun-14		mobile
19	79301 jual-flash-drives	Agung L.	5	null	null		29-Jun-14		mobile

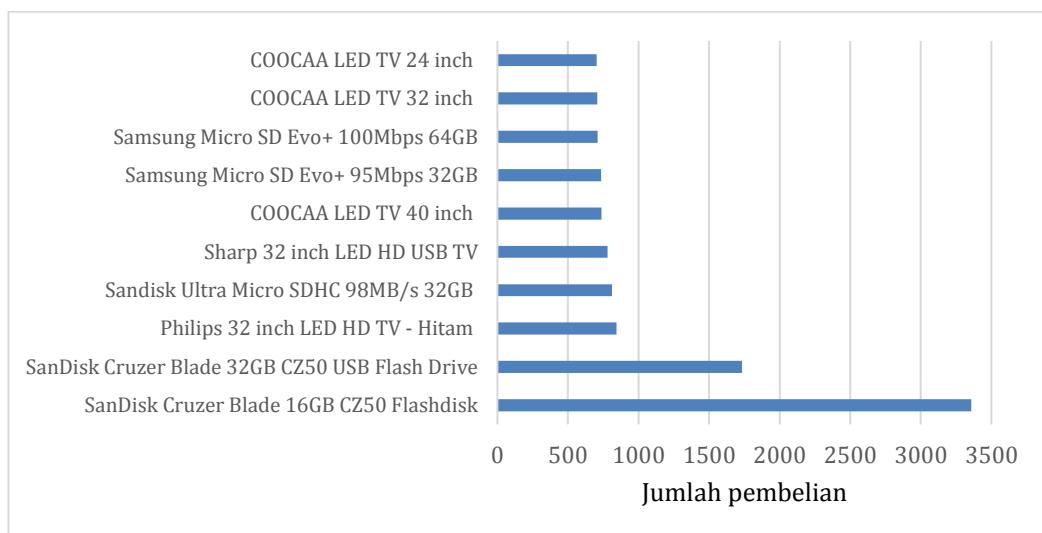
Gambar 3.7. Contoh duplikasi *record* pada data.

Setelah *data cleaning* dilakukan, didapatkan 64,994 *record* yang sudah bersih. Data bersih ini, sesuai dengan data aslinya, tidak memuat nama produk; hanya memuat nomor identitas produk saja. Nomor identitas produk ini merujuk pada nama produk, yang terdapat pada data keterangan tiap produk. Oleh karena itu, langkah selanjutnya adalah mengubah nomor identitas produk menjadi nama produk.

Dari data transaksi, dapat disimpulkan bahwa sebagian besar produk mendapatkan *rating* 5. Lihat Gambar 3.8. Selain itu, didapat pula informasi terkait 10 produk elektronik yang paling banyak dibeli, seperti diberikan pada Gambar 3.9.



Gambar 3.8. Persentase masing-masing rating.



Gambar 3.9. Sepuluh produk elektronik yang paling banyak dibeli.

3.4. Studi Kasus

Pada bagian ini, diberikan studi kasus pembuatan rekomendasi bagi konsumen, dengan menggunakan metode yang sudah dijelaskan pada Subbab 3.2. Data yang akan digunakan pada studi kasus ini adalah data transaksi dari Lazada Indonesia, seperti yang sudah dideskripsikan pada Subbab 3.3.

Sebelum algoritma *collaborative filtering item-based* diterapkan, proses *data cleaning* dilakukan lebih jauh, untuk menghilangkan *record* dengan identitas konsumen yang merupakan identitas default jika konsumen tidak memberikan identitasnya. Lihat Gambar 3.10. Misalnya, banyak *record* dengan identitas Lazada Member atau Lazada Guest. Identitas seperti ini tidak bisa dibedakan satu dengan lainnya dan tidak mewakili individu secara unik. Dari proses *data cleaning* ini, tersisa 60,942 *record*.

C	D	E	F	G	M	N	I
Lazada Customer	5	null	seller nya cepat k	24-Sep-19	androidApp		
Lazada Customer	5	null	pengiriman mant	24-Sep-19	androidApp		
Lazada Customer	5	null	pengiriman sangat	24-Sep-19	androidApp		
Lazada Customer	5	null	Selasa order kam	24-Sep-19	androidApp		
Lazada Customer	5	null	barang ok „ peng	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		
Lazada Customer	5	null	flasdis nya gk bisa	24-Sep-19	androidApp		
Lazada Customer	5	null	pengirimannya ce	24-Sep-19	androidApp		
Lazada Customer	5	null	pelayanan cepat,	24-Sep-19	androidApp		
Lazada Customer	1	null	Buruk	24-Sep-19	androidApp		
Lazada Customer	5	null	mantul bosQ	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		
Lazada Customer	5	null	Lumayan lah bua	24-Sep-19	androidApp		
Lazada Customer	5	null	sesuai ekspektasi l	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		
Lazada Customer	5	null	Mantap sesuai c	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		
Lazada Customer	5	null	null	24-Sep-19	androidApp		

Gambar 3.10. Contoh identitas default yang banyak digunakan.

Untuk studi kasus ini, agar kasus tidak terlalu besar, kita tentukan dulu produk yang akan direkomendasikan. Dalam hal ini, misalkan kita ambil saja tiga produk, yaitu Philips 32 inch LED HD TV, Sandisk Ultra Micro SD 32GB, dan Sharp 32 inch LED HD USB TV. Lalu kita lakukan langkah berikut:

1. Kumpulkan *record*, yang mengandung pembelian setidaknya dua dari tiga produk di atas, dari data yang sudah bersih. Mengapa setidaknya dua? Karena nanti, saat menghitung kemiripan produk, akan dibandingkan dua produk, yang keduanya sudah diberi *rating*.
2. Dari hasil *record* yang sudah dikumpulkan, hilangkan informasi yang tidak dibutuhkan. Sehingga hanya tersisa yang dibutuhkan saja untuk membuat rekomendasi, yaitu, identitas konsumen, produk, dan *rating*nya.

Setelah dua langkah di atas dilakukan, tersisa 168 *record* yang akan dijadikan data *rating* kita. Pada Tabel 3.1 diberikan sebagian *record* dari data *rating* yang dihasilkan.

Tabel 3.1. Sebagian data rating yang dihasilkan

Konsumen	Produk	Rating
Agus W.	Sharp 32 inch LED HD USB TV	5
Agus W.	Philips 32 inch LED HD TV	5
Agus W.	Sandisk Ultra Micro SD 32GB	5
Ahmad	Sharp 32 inch LED HD USB TV	5
Ahmad	Philips 32 inch LED HD TV	4
Ahmad F.	Sharp 32 inch LED HD USB TV	5
Ahmad F.	Philips 32 inch LED HD TV	5
Ahmad F.	Sandisk Ultra Micro SD 32GB	5

Selanjutnya, untuk kemudahan penulisan, nama produk akan dikodekan menjadi sebagai berikut:

- p1: Sharp 32 inch LED HD USB TV
- p2: Philips 32 inch LED HD TV
- p3: Sandisk Ultra Micro SD 32GB.

Dari data *rating* yang sudah didapat, selanjutnya kita bangun matriks *rating*. Pada langkah ini, yang dilakukan adalah menuliskan *rating* yang oleh setiap konsumen, pada satu baris yang sama. Sebagian matriks *rating* diberikan pada Tabel 3.2. Misalkan, pada Tabel 3.1, konsumen dengan identitas Agus S. membeli 3 produk tersebut, dan memberi *rating* 5 untuk masing-masing produk. Tiga baris pertama pada Tabel 3.1 tersebut kita ubah menjadi baris pertama pada Tabel 3.2.

Pada Tabel 3.2, ada *rating* yang diberi "?". Notasi ini dipilih untuk menunjukkan bahwa konsumen belum memberi *rating* untuk produk tersebut. Contohnya, konsumen dengan identitas Arif A. sudah membeli dan memberikan *rating* untuk produk p1 dan p3, tapi belum untuk produk p2.

Tabel 3.2. Sebagian matriks rating

Konsumen	p1	p2	p3
Agus W.	5	4	5
Ahmad F.	5	5	5
Ahmad R.	5	3	5
Ahmad S.	5	5	5
Anis	5	4	5
Arif S.	3	4	5
Dian A.	5	5	5
Indra	1	5	5
Muhammad A.	5	1	5
Arif A.	1	?	3
Yuni A.	?	5	5
Budi P.	5	4	?

Dari matriks *rating* yang dihasilkan, kita bangun *item-to-item similarity matrix*. Matriks ini adalah matriks dengan jumlah kolom dan baris yang sama, sejumlah produk yang ingin direkomendasikan. Dalam kasus ini, karena terdapat tiga produk yang akan direkomendasikan, maka matriks memiliki tiga baris dan tiga kolom. Tiap anggota matriks ini, menunjukkan nilai kemiripan *rating* antara dua buah produk. Misalnya, baris 1 dan kolom 1, menunjukkan nilai kemiripan produk p1 dan p1. Karena kedua produk ini adalah produk yang sama, maka nilai kemiripannya adalah 1. Begitu juga dengan produk p2 dan p2, serta p3 dan p3. Lihat Gambar 3.11.

Untuk menghitung nilai kemiripan produk p1 dan p2, pada studi kasus ini, kita hitung jarak Euclidean [Rosalind, 2020] antara *rating* produk p1 dan p2, yang diberikan pada Tabel 2 kolom p1 dan p2. Namun, dalam perhitungan ini, *record* yang diambil hanyalah *record* di mana kedua barang sudah diberi *rating* oleh konsumen. Contohnya, pada Tabel 2, *record* dengan identitas Arif A. dan Yuni A., tidak diikutsertakan, karena mereka tidak memberi *rating* untuk p2 dan p1. Dari hasil perhitungan, nilai kemiripan produk p1 dan p2 adalah 11.09. Nilai ini dimasukkan ke baris 1 kolom 2 dan baris 2 kolom 1 pada matriks. Dengan cara yang sama, nilai kemiripan p1 dan p3, serta p2 dan p3 dapat dihitung. Item-to item similarity matrix yang lengkap diberikan pada Gambar 3.11.

	p1	p2	p3
p1	1		
p2		1	
p3			1

	p1	p2	p3
p1	1	11.09	7.75
p2	11.09	1	8.19
p3	7.75	8.19	1

Gambar 3.11. Pembangunan *item-to-item similarity matrix*.

Selanjutnya, dengan menggunakan *item-to-item similarity matrix* yang dihasilkan, kita hitung prediksi *rating* dengan menggunakan rumus *simple weighted average* [Saluja, 2018]. Dalam hal ini, akan dihitung prediksi *rating* konsumen (lihat Tabel 2):

- Arif A. untuk produk p2
- Yuni A. untuk produk p1
- Budi P. untuk produk p3

Perhitungan *simple weighted average* ini, misalkan untuk konsumen Arif A., memanfaatkan nilai *rating* Arif A. yang sudah diberikan untuk produk p1 dan p3, serta nilai kemiripan antara produk p1 dan p2, dan p3 dan p2.

Dari hasil perhitungan ini, didapatkan prediksi *rating*:

- Arif A. untuk produk p2 = 1.9
- Yuni A. untuk produk p1 = 5
- Budi P. untuk produk p3 = 4.9

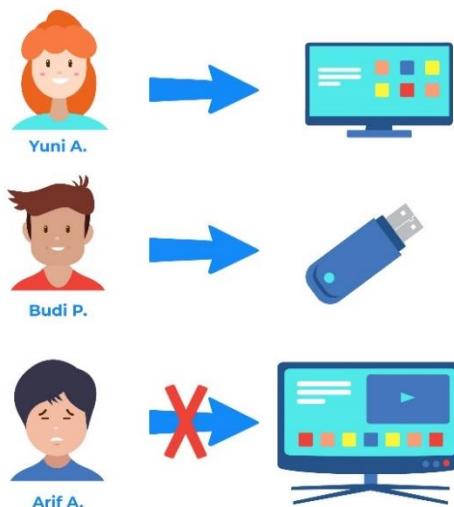
Hasil perhitungan ini ditunjukkan pada Tabel 3.3.

Tabel 3.3. Prediksi nilai rating

Konsumen	p1	p2	p3
Agus W.	5	4	5
Ahmad F.	5	5	5
Ahmad R.	5	3	5
Ahmad S.	5	5	5
Anis	5	4	5
Arif S.	3	4	5
Dian A.	5	5	5
Indra	1	5	5
Muhammad A.	5	1	5
Arif A.	1	1.9	3
Yuni A.	5	5	5
Budi P.	5	4	4.9

Berdasarkan hasil ini, dapat dilihat bahwa prediksi *rating* konsumen Yuni A. untuk produk p1 sangat tinggi, maka kita dapat merekomendasikan produk p1 untuk konsumen ini. Nilai *rating* tinggi yang diberikan konsumen Yuni A. untuk produk p2 dan p3 menunjukkan bahwa konsumen ini menyukai dua produk tersebut. Karena itu, algoritma menyimpulkan bahwa pasti konsumen Yuni A. juga menyukai produk p1. Hal yang sama juga berlaku untuk konsumen Budi P., di mana prediksi *rating* yang didapat untuk produk p3 cukup tinggi, sehingga produk ini dapat direkomendasikan untuk konsumen Budi P.

Namun, hal berbeda dapat dilihat pada hasil prediksi *rating* konsumen Arif A. untuk produk p2. Hasil prediksi ini rendah, yaitu hanya 1.9. Konsumen Arif A. tidak menyukai produk p1 dan p3; ini terlihat dari *rating* yang diberikannya. Oleh karena itu, algoritma menyimpulkan bahwa konsumen ini pasti juga tidak menyukai produk p2. Sehingga, produk p3 tidak dapat direkomendasikan untuk konsumen Arif A. Rekomendasi yang dihasilkan ini dapat juga dilihat pada Gambar 3.12.



Gambar 3.12. Hasil rekomendasi.

Hasil ini mungkin saja berbeda jika kombinasi produk yang dipilih juga berbeda. Artinya, jika produk p1 diolah bersama dengan dua produk lain (bukan p2 dan p3), maka hasil rekomendasi dapat berbeda. Selain itu, pilihan rumus perhitungan nilai kemiripan yang berbeda, juga dapat menyebabkan hasil rekomendasi yang berbeda, walupun perbedaannya tidak signifikan.

3.5. Penutup

Pada bab ini, telah dideskripsikan bagaimana sistem rekomendasi bekerja, untuk memberikan rekomendasi suatu produk untuk konsumen. Algoritma yang digunakan pada bab ini adalah algoritma *collaborative filtering item-based*, yang cara kerjanya cukup sederhana. Studi kasus juga diberikan, dengan menggunakan data transaksi dari *platform e-commerce* Lazada Indonesia, untuk memberikan gambaran lebih rinci tentang cara kerja sistem rekomendasi ini.

Selain memberikan cara kerja sistem rekomendasi, bab ini juga memberikan pengetahuan seputar *data cleaning*, algoritma, dan visualisasi data, yang merupakan hal krusial di bidang data science.

Sistem e-commerce yang nyata merupakan sistem yang besar dan kompleks. Data yang terkumpul dapat dikategorikan menjadi big data (lihat bahasan pada Bab 10), sehingga perlu ditangani dengan teknologi big data beserta algoritma-algoritma yang sesuai untuk big data. Namun melalui paparan pada bab ini, diharapkan wawasan pembaca bertambah, khususnya tentang bagaimana data dapat dimanfaatkan untuk meningkatkan penjualan pada sistem *e-commerce*.

Referensi

- (Adomavicius, 2005) G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions*, Vol. 17, No. 6, pp. 734-749, 2005.
- (Aseanup, 2019) <https://aseanup.com/top-e-commerce-sites-indonesia/> (diakses 23 Juni 2020)
- (Kaggle, 2019) <https://www.kaggle.com/grikomsn/lazada-indonesian-reviews> (diakses 12 Juni 2020)
- (Linden, 2003) G. Linden, B. Smith, and J. York, "Amazon.com Recommendations Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, Volume: 7, Issue: 1, Jan.-Feb. 2003.
- (Rosalind, 2020) <http://rosalind.info/glossary/euclidean-distance/> (diakses 23 Juni 2020)
- (Saluja, 2018) <https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1> (diakses 10 Juni 2020)
- (Sarwar, 2000) B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," *Proceedings of the 2nd ACM conference on Electronic commerce*, pp. 158–167, Oct. 2000.
- (Sarwar, 2001) B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "*Item-based Collaborative filtering Recommendation Algorithms*," : *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, April 2001.

(Schafer, 2001) J.B. Schafer, J.A. Konstan, and J. Reidl, "E-Commerce Recommendation Applications," Data Mining and Knowledge Discovery, Kluwer Academic, pp. 115-153, 2001.
(Sivapalan, 1999) S. Sivapalan, A. Sadeghian, H. Rahanam, and A. M. Madni, "Recommender Systems in E-Commerce," Proceedings of the 1st ACM conference on Electronic commerce Nov. 1999, pp. 158-166
(Statista, 2020) https://www.statista.com/outlook/243/120/ecommerce/indonesia#_market-revenue (diakses 17 Juni 2020)

Bab 4 Pencarian Keterkaitan Bahan Masakan dengan Teknik *Clustering*

Oleh:

Kritopher D. Harjono dan Mariskha Tri Adithia

4.1. Pendahuluan

Apakah para pembaca pernah merasakan kebingungan saat memilih masakan di restoran baru? Kita, yang umumnya tidak mengenal rasa bahan dan bumbu tertentu, pastinya bingung walaupun menu sudah menjelaskan dengan rinci tentang suatu masakan pada menu. Kebingungan memilih ini juga terjadi terutama saat kita mengunjungi kota baru, apalagi negara baru (Gambar 4.1).

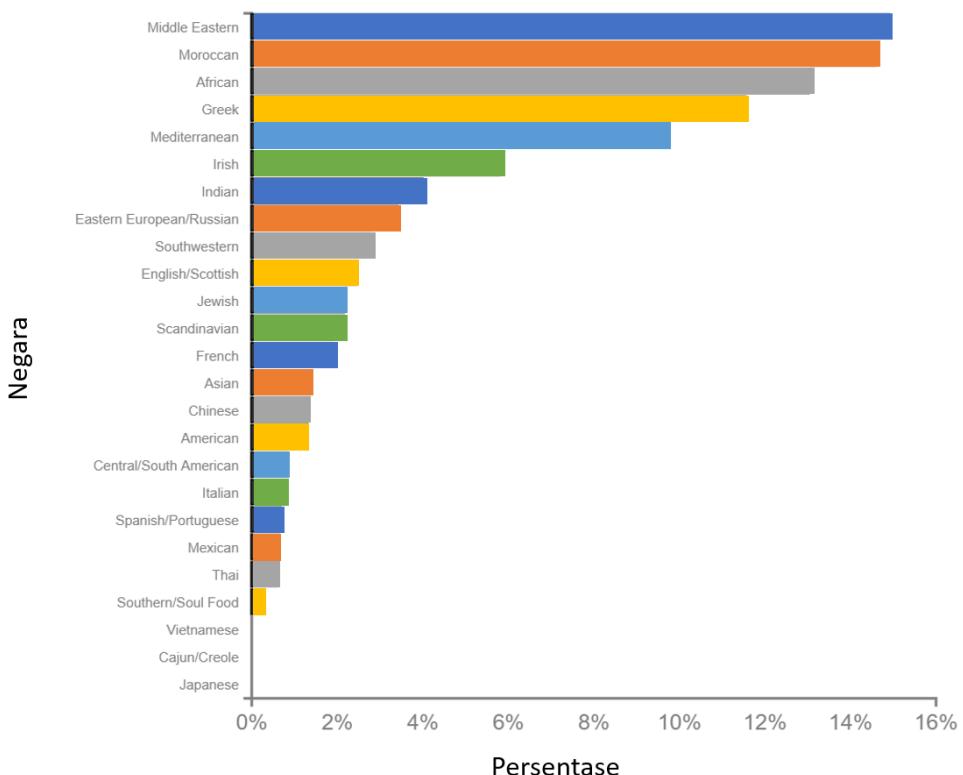


Gambar 4.1. Kebingungan memilih masakan.

Masakan dari tempat yang berbeda, memang memiliki rasa khas yang berbeda-beda pula. Misalnya, masakan dari negara Eropa tengah dan utara, secara umum, seperti masakan yang kurang bumbu, jika dibandingkan dengan masakan Asia. Bagi orang Asia, Masakan Eropa ini hambar. Sedangkan bagi orang Eropa, masakan Asia terlalu berbumbu dan pedas. Apakah semua masakan hambar pasti adalah masakan Eropa atau Amerika Utara? Atau masakan pedas pasti dari India? Walaupun tidak bisa digeneralisir,

memang tidak dapat dipungkiri bahwa rasa khas masakan tertentu berkaitan dengan negara atau kawasan tertentu di dunia (Independent, 2015).

Suatu rasa khas pada masakan tertentu, dihasilkan oleh bahan-bahan yang digunakan untuk memasak masakan tersebut. Berdasarkan riset yang dilakukan oleh Priceconomics (Priceconomics, 2015) terhadap 13 ribu resep masakan dari seluruh dunia, bahan-bahan tertentu memang dominan digunakan pada banyak masakan dari suatu kawasan atau negara. Misalnya, 30% resep masakan Asia menggunakan bahan minyak wijen, 16% resep masakan Cina menggunakan bahan minyak kacang, dan 15% resep masakan Jerman menggunakan bahan kol asam. Bagaimana dengan daging kambing? Negara atau bagian dunia manakah yang paling banyak menggunakan daging kambing dalam masakan? Ternyata, jawabannya adalah kawasan Timur Tengah. Sedangkan masakan Jepang, tidak menggunakan daging kambing sama sekali. Lihat Gambar 4.2.



Gambar 4.2. Persentase resep masakan negara atau kawasan tertentu, yang menggunakan daging kambing (Priceconomics, 2015).

Apakah mungkin kita dapat mengaitkan bahan-bahan masakan dengan negara asal masakan tersebut? Lebih jauh lagi, apakah mungkin kita mengaitkan bahan masakan satu negara dengan lainnya, sehingga memudahkan kita memilih masakan saat mengunjungi tempat baru? Hal ini mungkin dilakukan dengan menggunakan teknik *clustering* atau pengelompokan. *Clustering* adalah salah satu teknik data mining

untuk mengelompokkan data ke dalam beberapa grup di mana anggota-anggota dalam suatu grup memiliki kemiripan sifat (Seif, 2018). Dengan menggunakan teknik *clustering* ini, bahan-bahan masakan dikelompokkan ke dalam grup-grup, dan tiap grup ini akan dianalisis, apakah mewakili negara tertentu.

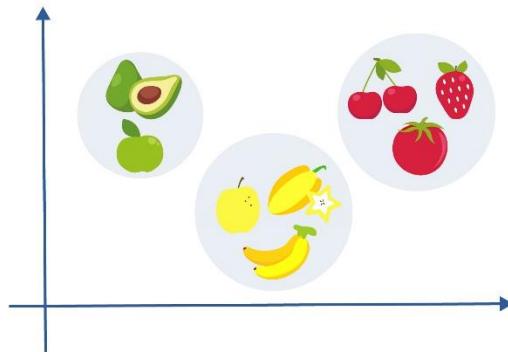
Pada artikel ini, teknik *clustering* akan dijelaskan dan diaplikasikan untuk mengaitkan bahan-bahan masakan dengan negara asalnya. Teknik *clustering* juga digunakan untuk mengaitkan bahan masakan suatu negara dengan negara lain. Penjelasan akan juga dilengkapi dengan studi kasus dan hasilnya.

4.2. Teknik *Hierarchical Clustering*

Clustering adalah proses mengelompokkan data menjadi grup-grup (Han, 2012). Grup-grup ini disebut *cluster*. Anggota di dalam suatu *cluster* memiliki tingkat kesamaan sifat atau fitur yang tinggi dengan anggota lainnya. Sebaliknya, tingkat kesamaan sifat atau fitur anggota suatu *cluster*, bernilai rendah dengan sifat atau fitur anggota *cluster* lain. Pada *data mining*, *clustering* digunakan untuk mendapatkan pemahaman terkait distribusi data, mengobservasi karakteristik tiap *cluster*, dan berfokus pada *cluster* tertentu saja untuk analisis lebih lanjut.

Gambar 4.3 menunjukkan suatu contoh hasil *clustering* sederhana. Pada kasus ini, sekelompok buah dikelompokkan ke dalam *cluster*. Gambar ini menunjukkan bahwa buah yang memiliki kesamaan sifat, dikelompokkan menjadi satu *cluster*. Apakah kemiripan sifat buah pada suatu *cluster* di kasus ini? Jawabannya adalah warna. Pada kasus ini, buah dengan warna mirip dikelompokkan menjadi satu *cluster*. Misalnya, alpukat dan apel hijau dikelompokkan menjadi satu *cluster* karena berwarna hijau. Tentu saja jika dikelompokkan berdasarkan sifat atau fitur lain, misalnya ukuran, hasil *cluster* yang didapat, akan berbeda.

Teknik *clustering* dapat dilakukan dengan menggunakan banyak algoritma, misalnya *k-Means*, *mean-shift*, and *agglomerative hierarchical clustering* (Seif, 2018). Pada artikel ini, algoritma yang akan digunakan adalah *agglomerative hierarchical clustering*. *Agglomerative hierarchical clustering* adalah algoritma *clustering* yang mengelompokkan data secara hirarkis, yang bekerja secara *bottom-up*. Algoritma bekerja dengan menjadikan tiap objek pada data menjadi *satu cluster*. Lalu, *cluster* digabungkan dengan *cluster* lain yang mirip. Dalam hal ini, kemiripan ditentukan berdasarkan jarak antara kedua *cluster* tersebut. Proses ini terus dilakukan sampai akhirnya hanya terbentuk satu *cluster* saja.



Gambar 4.3.. Ilustrasi clustering sederhana.

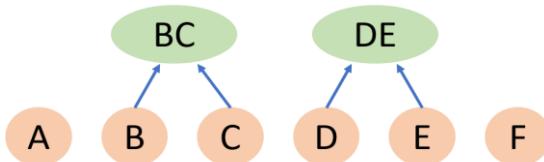
Misalkan kita memiliki data yang terdiri atas 6 objek, yaitu A, B, C, D, E, dan F. Langkah-langkah algoritma *agglomerative hierarchical clustering* berikut ilustrasinya pada contoh data yang diberikan adalah sebagai berikut (Seif, 2018):

1. Bentuk cluster sebanyak objek pada data. Dalam hal ini, kita bentuk cluster sebanyak 6 buah, di mana masing-masing cluster beranggotakan satu objek. Lihat **Error! Reference source not found.** Gambar 4.4.



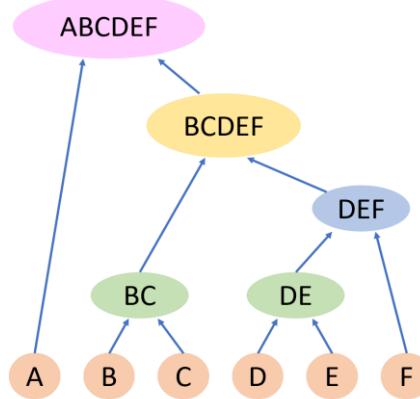
Gambar 4.4. Cluster yang terbentuk dari langkah 1.

2. Hitung jarak antar cluster, lalu gabungkan cluster yang jaraknya paling dekat. Misalkan dalam hal ini, cluster B dan C, dan D dan F, berjarak paling dekat, maka gabungkan cluster B dan C, menjadi cluster BC, dan D dan F, menjadi cluster DF. Lihat Gambar 4.5.



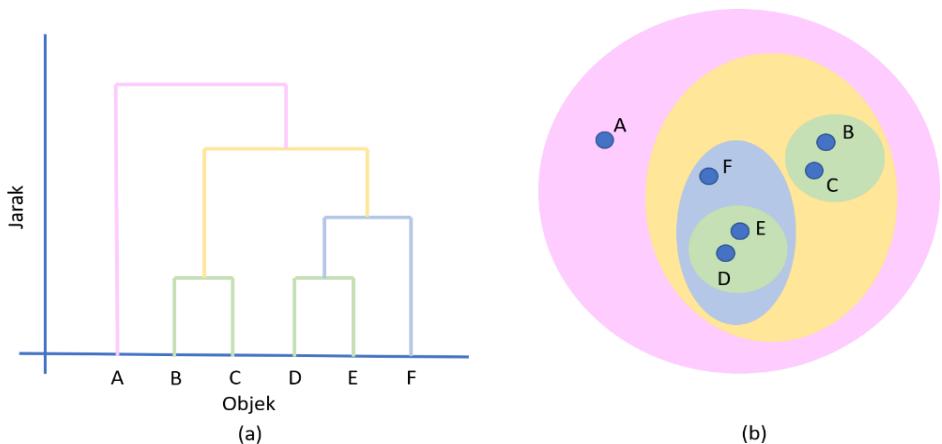
Gambar 4.5. Cluster yang terbentuk dari langkah 2 setelah dijalankan satu kali.

3. Ulangi lagi langkah 2, sampai terbentuk hanya satu cluster. Misalkan, didapat cluster DEF, karena cluster DE berjarak dekat dengan F sehingga kedua cluster ini bisa digabungkan. Lalu pada tahap selanjutnya, didapat cluster BCDEF, dan yang terakhir cluster ABCDEF. Lihat Gambar 4.6.



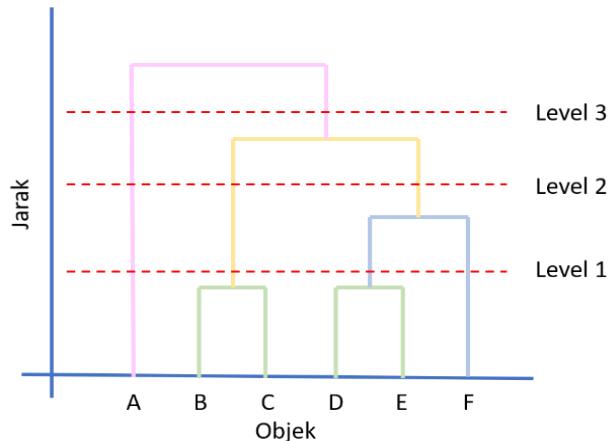
Gambar 4.6. Cluster yang dihasilkan.

Proses di atas dapat digambarkan dalam bentuk dendrogram dan diagram Venn, seperti yang ditunjukkan pada Gambar 4.7. Pada gambar ini, warna hijau menunjukkan hasil *clustering* pada tahap pertama, warna biru menunjukkan hasil *clustering* pada tahap kedua, dan seterusnya.



Gambar 4.7. (a) Dendrogram dan (b) diagram Venn yang menggambarkan algoritma agglomerative hierarchical clustering.

Lalu, pada contoh di atas, yang manakah cluster yang dihasilkan? Hal ini bergantung pada berapa jumlah cluster yang ingin dibangun. Jika kita ingin membentuk empat cluster, maka potong dendrogram pada level pertama, sehingga didapat cluster A, BC, DE, dan F (lihat Gambar 4.8.). Namun, jika kita ingin mendapatkan jumlah cluster yang lebih kecil, maka kita potong dendrogram, misalnya di level dua, sehingga didapatkan tiga cluster, yaitu cluster A, BC, dan DEF.



Gambar 4.8. Pemotongan dendogram untuk mendapatkan jumlah cluster yang diinginkan.

4.3. Data Resep Masakan

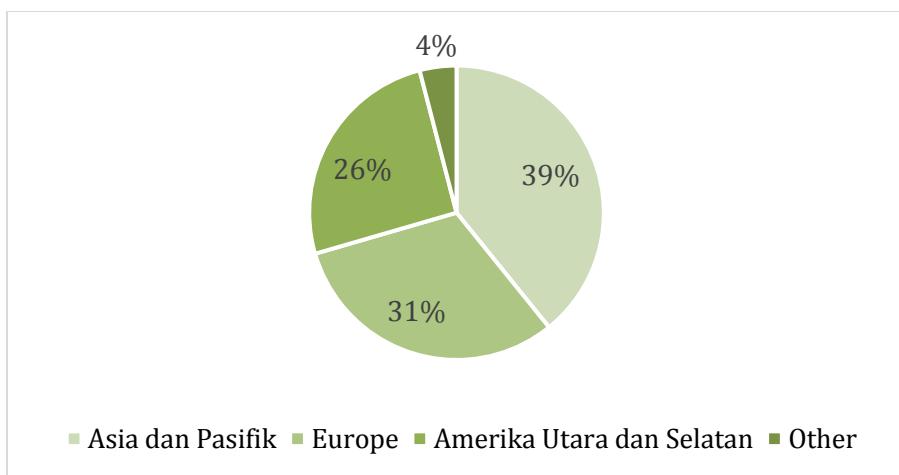
Pada artikel ini, studi kasus dilakukan berdasarkan makalah yang berjudul “Tell Me What You Eat, and I Will Tell You Where You Come From: A Data Science Approach for Global Recipe Data on the Web” (Kim, 2016). Pada makalah tersebut, digunakan data resep masakan dari Recipe Source (Recipe, 2020). Website ini adalah salah satu website resep masakan tertua di dunia, yang dibangun oleh Jennifer Snider pada tahun 1993. Awalnya, terkumpul 10,000 resep, sekarang Recipe Source sudah memiliki sekitar 70,000 resep dari berbagai negara dan kawasan di dunia. Contoh resep masakan pada website Recipe Source dapat dilihat pada Gambar 4.9.. Resep ini berasal dari Indonesia, dan berbahan dasar *seafood*.

CUMI-CUMI ISI (STUFFED SQUID)			
Recipe By	:	Serving Size	: 4 Preparation Time :0:00
Categories	:	Seafood	Indonesian
Amount Measure		Ingredient -- Preparation Method	
1	lb	Squid, fresh	
3/4	lb	Snapper fillets	
1		Garlic clove	
1		Egg white	
1/2	t	Salt	
1/4	t	Pepper, white	
		Nutmeg -- dash	
2		Shallot	
2		Thai chile, fresh	
3		Candlenut	
2		Lemon grass, stem	
		Oil -- for frying	
1	c	Coconut milk	
Fat grams per serving:		Approx. Cook Time: 1:20	
Clean the squid. Wash under cold running water and dry thoroughly. Remove the skin from the snapper (ensure no bones remain) and cut the meat into tiny pieces. Crush the garlic. Beat the egg whites lightly, add the snapper and garlic and season with salt, white pepper and nutmeg. Stir to blend thoroughly, then stuff the mixture into the squid. Chop the shallots, chiles, candlenuts, and lemon grass, then saute in very hot oil for three to four minutes. Add the coconut milk and bring to the boil, then lower heat and add the stuffed squid. Allow to simmer until the squid is very tender, approximately one hour, then transfer to a serving dish and pour the sauce on top.			

Gambar 4.9. Contoh resep dari Recipe Source.

Dari Gambar 4.9, dapat dilihat bahwa resep masakan pada website Recipe Source memuat informasi, di antaranya, pembuat resep, jumlah penyajian, kategori, bahan dan jumlah yang dibutuhkan, dan cara memasak.

Dari 70,000 resep si Recipe Source ini, hanya sekitar 10% resep yang dikategorikan berdasarkan negara atau Kawasan asalnya. Oleh karena itu, kita hanya akan menggunakan 5,917 resep dari 22 negara atau bangsa. Rangkuman resep yang digunakan diberikan pada Gambar 4.10 dan *Tabel 4.1*.



Gambar 4.10. Rangkuman resep masakan berdasarkan kawasan asalnya

Tabel 4.1. Rangkuman resep masakan

Kawasan	Negara/bangsa	Jumlah resep per negara/bangsa
Asia dan Pasifik	Cina	892
	Filipina	54
	India	589
	Indonesia	112
	Jepang	122
	Korea	104
	Thailand	350
	Vietnam	96
Europe	Inggris	92
	Prancis	110
	Jerman	232
	Yunani	407
	Irlandia	101
	Italia	657
	Polandia	88
	Rusia	105
	Skotlandia	61
Amerika Utara dan Selatan	Cajun	540
	Kanada	111
	Karibia	87
	Meksiko	768
	Other	239

Data resep inilah yang selanjutnya digunakan pada studi kasus kita, untuk mengaitkan bahan masakan dengan negara asalnya.

4.4. Studi Kasus

Sebelum data kita proses dengan menggunakan algoritma *agglomerative hierarchical clustering*, yang sudah dijelaskan sebelumnya, data resep perlu diolah terlebih dahulu. Pengolahan yang dilakukan adalah sebagai berikut:

1. Ambil hanya informasi yang dibutuhkan, dalam hal ini adalah asal negara atau bangsa dan bahan masakan.
2. Pilih bahan pembeda antara masakan dari satu negara dengan lainnya. Misalnya, cabai dapat dipilih sebagai bahan masakan pembeda antara masakan dari Indonesia dengan masakan dari Kanada, karena cabai sangat banyak digunakan pada masakan Indonesia, namun sedikit pada masakan Kanada.
3. Hitung persentase penggunaan bahan pembeda pada langkah sebelumnya, dari seluruh resep negara tertentu. Untuk singkatnya, selanjutnya persentase ini kita sebut persentase bahan.

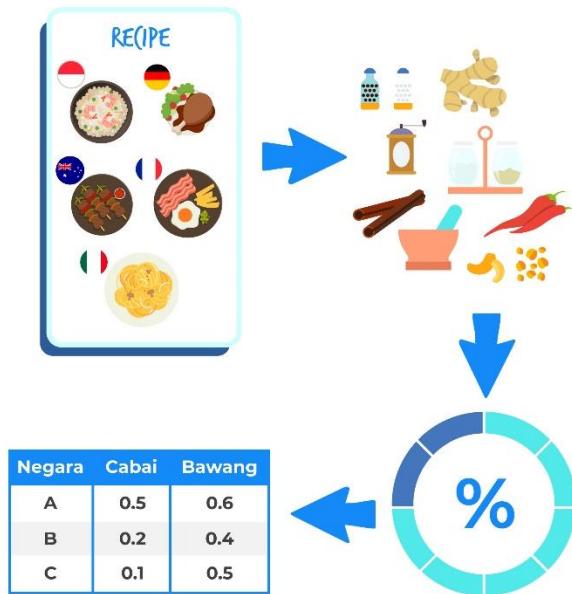
Misalnya, kita ingin menghitung persentase bahan cabai pada resep masakan Indonesia, dan misalkan terdapat tiga resep masakan Indonesia. Masing-masing bahan masakan ketiga resep ini diberikan pada *Tabel 4. 2*. Berdasarkan tabel ini, terdapat cabai pada dua dari tiga resep bahan masakan ini, yaitu pada resep Cumi-cumi Isi dan Terong Balado. Jika cabai dipilih sebagai bahan pembeda masakan Indonesia, maka nilai persentase bahan ini adalah $\frac{2}{3}$ atau 67%.

Tabel 4. 2. Bahan masakan Cumi-cumi Isi, Terong Balado, dan Opor Ayam

Cumi-cumi Isi	Terong Balado	Opor ayam
Cumi-cumi	Terong	Ayam
Ikan kakap	Bawang putih	Garam
Bawang putih	Bawang merah	Bawang merah
Telur	Tomat	Bawang putih
Garam	Gula	Minyak goreng
Merica putih	Garam	Ketumbar
Pala	Cabai	Jahe
Bawang merah	Air	Serai
Cabai	Minyak goreng	Santan
Kemiri		Daun salam
Serai		
Minyak goreng		
Santan		

4. Bangun matriks persentase bahan pembeda, dengan baris berisikan negara asal masakan, kolom berisikan nilai persentase per bahan masakan pembeda.

Langkah pemrosesan data ini juga diilustrasikan pada Gambar 4.11.



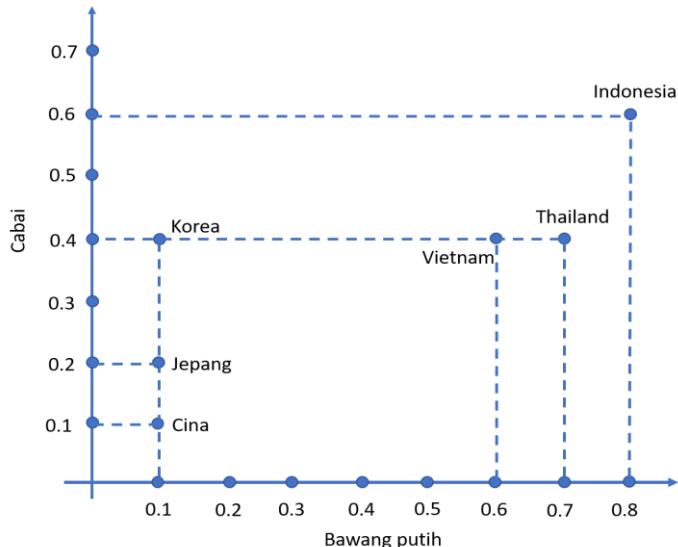
Gambar 4.11. Pengolahan data sebelum diproses lebih lanjut.

Hasil matriks persentase bahan pembeda inilah yang selanjutnya diproses dengan menggunakan algoritma *agglomerative hierarchical clustering*. Dalam hal ini, negara akan dikelompokkan berdasarkan nilai persentase bahan-bahan pembedanya.

Agar mudah dimengerti, sebagai ilustrasi, kita ambil resep masakan dari enam negara, yaitu Cina, Indonesia, Jepang, Korea, Thailand, dan Vietnam. Dari enam negara ini, misalkan kita ambil dua bahan pembeda saja, yaitu bawang putih dan cabai. Mengapa hanya diambil dua bahan pembeda saja? Karena dengan cara ini, tiap negara dapat direpresentasikan pada bidang Kartesius, dengan nilai persentase bahan pembeda menjadi koordinatnya. Misalkan, didapat matriks persentase bahan pembeda pada *Tabel 4.3*. Maka, negara Indonesia, kita tempatkan pada koordinat (0.8, 0.6) pada bidang Kartesius. Ini kita lakukan untuk semua negara, dan hasilnya dapat dilihat pada Gambar 4.12.

Tabel 4.3. Matriks persentase dua bahan pembeda negara

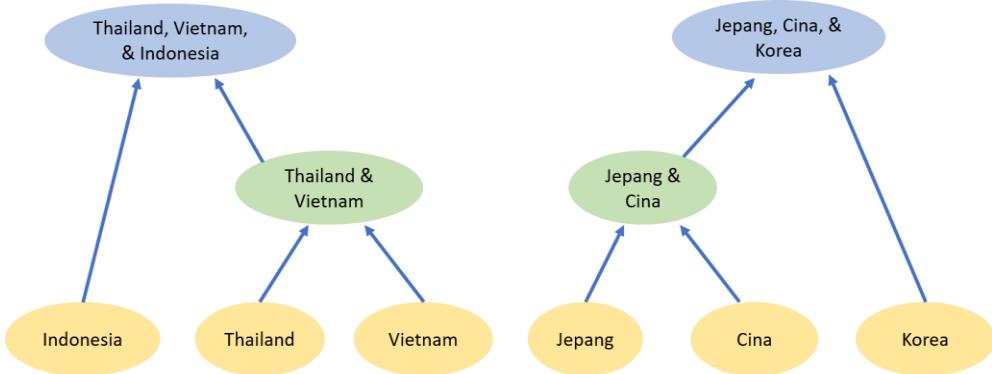
Negara	Bawang Putih	Cabai
Cina	0.1	0.1
Indonesia	0.8	0.6
Jepang	0.1	0.2
Korea	0.1	0.4
Thailand	0.7	0.4
Vietnam	0.6	0.4



Gambar 4.12. Negara pada bidang Kartesius berdasarkan nilai persentase bahan pembedanya.

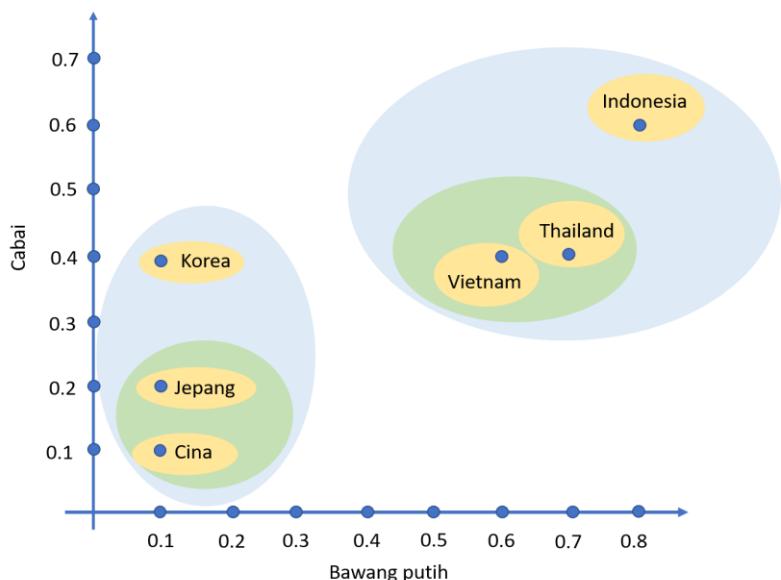
Setelah diletakkan pada bidang Kartesius, kita dapat melihat posisi resep masakan dari tiap negara relatif terhadap negara lain. Contohnya, kita dapat melihat resep masakan Cina memiliki persentase bahan pembeda yang cukup "dekat" dengan resep masakan Jepang. Begitu pula antara resep masakan Thailand dan Indonesia. Namun, dari Gambar 4.12 dapat kita lihat juga bahwa resep masakan Indonesia memiliki

persentase bahan pembeda yang berbeda cukup “jauh” dengan resep masakan Korea. Dengan melihat jarak inilah *clustering* dilakukan. Proses *clustering* ini dapat dilihat pada Gambar 4.13. Pada gambar ini ditunjukkan bahwa di awal Thailand dan Vietnam, serta Jepang dan Cina dijadikan satu cluster karena jarak persentase bahan pembeda negara-negara ini berjarak dekat. Kemudian pada langkah selanjutnya, cluster Thailand, Vietnam, Indonesia, dan Jepang, Cina, Korea dibentuk, karena jarak antar negara ini relatif dekat, jika dibandingkan dengan pilihan negara lain yang ada.



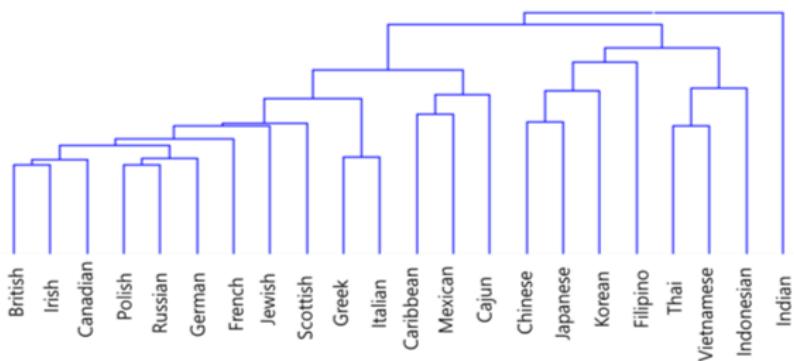
Gambar 4.13. Cluster yang dihasilkan pada tiap langkah.

Hasil cluster dengan representasi diagram Venn diberikan pada Gambar 4.14. Dapat dilihat bahwa di tahap awal *cluster* yang terbentuk ditandai dengan elips hijau. Lalu di tahap selanjutnya terjadi penggabungan *cluster*, sehingga terbentuk *cluster* yang ditandai dengan warna biru muda.



Gambar 4.14. Hasil cluster dalam bentuk diagram Venn pada bidang Kartesius.

Jika proses dilakukan untuk seluruh 5,917 resep, maka didapatlah hasil *clustering* yang ditunjukkan pada Gambar 4.15. Di sini, contohnya bahan masakan dari Inggris dan Irlandia sangat mirip, sehingga mereka dikelompokkan menjadi satu cluster. Pada tahap selanjutnya, terjadi penggabungan cluster antara Inggris dan Irlandia, dengan Kanada. Yang artinya, walaupun tidak mirip sekali, bahan masakan Inggris dan Irlandia dengan bahan masakan Kanada lebih mirip jika dibandingkan dengan masakan Prancis.



Gambar 4.15. Dendrogram proses clustering untuk seluruh resep (Kim, 2016).

Masakan Asia, secara umum dikelompokkan menjadi dua cluster besar, yaitu Cina, Jepang, Korea, Filipina, dan Thailand, Vietnam, Indonesia. Kedua cluster ini dibedakan oleh banyak bumbu yang digunakan pada makanan Asia Tenggara, dalam hal ini diwakili Thailand, Vietnam, dan Indonesia. India, sebagai negara di Asia, tidak digabungkan dengan cluster manapun sampai tahap terakhir, karena variasi bumbu dan bahan yang digunakan, yang meliputi bumbu dan bahan dari Eropa dan Asia.

Dari cluster yang dihasilkan ini, kita dapat menyimpulkan bahwa memang ada keterkaitan antara bahan masakan dengan negara atau bangsa asalnya. Dengan cara ini, jika kita diberikan sekelompok bahan masakan, kita dapat menebak kandidat negara-negara asalnya.

Selain itu, dari cluster yang dihasilkan ini pula, kita dapat terbantu saat memilih makanan. Misalkan, kita sangat menyukai masakan Meksiko, kemungkinan kita juga akan menyukai masakan Karibia dan Cajun, karena Meksiko berada di cluster yang sama dengan masakan Karibia dan Cajun. Sebaliknya, jika kita tidak menyukai suatu masakan, misalnya masakan Jerman, maka kemungkinan besar kita juga tidak menyukai masakan dari negara-negara yang sekelompok dengan Jerman, yaitu Polandia dan Rusia.

Selain pada permasalahan ini, teknik *clustering* banyak diaplikasikan pada bidang lain pula. Misalnya, teknik *clustering* dapat digunakan untuk mengenali berita palsu atau *hoax* (Hosseinimotagh, 2018). Selain itu, teknik clustering juga digunakan untuk mengenali *email spam* (Whittaker, 2019).

Pada bidang *e-Commerce*, teknik *clustering* digunakan untuk melakukan segmentasi kustomer berdasarkan sejarah transaksi, ketertarikan, dan aktifitas pada *platform e-Commerce*, untuk menentukan target promosi yang tepat. Dengan menggunakan teknik clustering pula, produk yang paling banyak dibeli bersamaan dengan produk tertentu dapat dikelompokkan. Informasi ini dapat digunakan untuk menentukan rekomendasi produk bagi kustomer (Le, 2019). Teknik *clustering* juga dapat digunakan

untuk mengelompokkan kustomer berdasarkan lokasi geografisnya. Hasil *cluster* yang dihasilkan ini selanjutnya dapat dimanfaatkan untuk analisis selanjutnya, misalnya dengan teknik klasifikasi, untuk mengaitkan tiap *cluster* dengan produk tertentu yang mungkin digemari.

4.5. Penutup

Pada artikel ini, telah dijelaskan satu teknik yang digunakan pada bidang data science, yaitu teknik *clustering*. Satu algoritma telah dijelaskan dengan ringkas, yaitu algoritma *agglomerative hierarchical clustering*. Penggunaan algoritma ini juga diberikan pada bagian studi kasus, yaitu untuk mencari keterkaitan antara bahan masakan dengan asal negara atau bangsanya.

Selain memberikan cara kerja teknik *clustering*, bab ini juga memberikan pengetahuan seputar persiapan data, yang merupakan hal krusial di bidang data science.

Melalui bab ini, diharapkan pembaca mendapatkan tambahan wawasan terkait data science dan keluasan penggunaannya untuk mendapatkan pola dan insight dari data. Dari berbagai informasi yang dipaparkan pada bab ini, diharapkan pula pembaca dapat menggunakan teknik *clustering*, untuk pencarian pola dan analisis lebih lanjut, untuk masalah lain, sesuai kebutuhan.

Referensi

- (Han, 2012) J. Han, M. Kamber, and J. Pei, "Data Mining Techniques and Concepts," Morgan Kaufmann, USA, 2012.
- (Hosseinimotagh, 2018) S. Hosseinimotagh dan E. E. Papalexakis, "Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles," Proceedings of the Workshop on Misinformation and Misbehavior Mining on the Web (MIS2), 2018.
- (Independent, 2015) <https://www.independent.co.uk/life-style/food-and-drink/news/national-cuisines-what-ingredients-make-dishes-from-different-cultures-distinctive-10404837.html> (diakses 13 Juli 2020).
- (Kim, 2016) K. J. Kim dan C. H. Chung, "Tell Me What You Eat, and I Will Tell You Where You Come From: A Data Science Approach for Global Recipe Data on the Web," in *IEEE Access*, vol. 4, pp. 8199-8211, 2016.
- (Le, 2019) <https://lucidworks.com/post/clustering-classification-supervised-unsupervised-learning-ecommerce/#:~:text=Importance%20of%20Clustering%20in%20Ecommerce&text=Clustering%20is%20sometimes%20called%20unsupervised,the%20better%20our%20clusters%20are.> (diakses 21 Juli 2020).
- (Priceconomics, 2015) <https://priceconomics.com/what-are-the-defining-ingredients-of-a-cultures/> (diakses 13 Juli 2020).
- (Recipe, 2020) <https://recipesource.com/> (diakses 15 Juli 2020)
- (Seif, 2018) <https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68> (diakses 13 Juli 2020).
- (Whittaker, 2019) <https://datafloq.com/read/7-innovative-uses-of-clustering-algorithms/6224> (diakses 21 Juli 2020)

Halaman ini sengaja dikosongkan

Bab 5 Analisis Data Penginderaan Jauh Satelit, Kasus: Prediksi Panen Padi

Oleh:

Veronica S. Moertini

5.1. Pendahuluan

Lebih dari seribu satelit buatan manusia mengorbit di ruang angkasa yang berputar mengikuti rotasi bumi dan berstatus masih aktif atau dimanfaatkan manusia (Gambar 5.1). Satelit dimanfaatkan untuk keperluan di berbagai bidang, misalnya (Ritter, 2014; Bitar, 2019; Bitar, 2020):

- Bidang meteorologi dan klimatologi: peramalan cuaca dan bencana alam yang terkait dengan cuaca, seperti badai, puting beliung dan banjir.
- Bidang hidrologi: pemetaan daerah aliran sungai (DAS) terkait dengan potensi banjir.
- Bidang kelautan: pengamatan gelombang laut dan pemetaan perubahan pantai akibat erosi dan sedimentasi.
- Bidang pertanian dan kehutanan: pengenalan dan klasifikasi jenis tanaman, evaluasi kondisi tanaman, perkembangan luas hutan dan perkiraan produksi tanaman.

Selain contoh di atas, satelit juga dimanfaatkan untuk komunikasi, astronomi, navigasi, keperluan militer, dll.



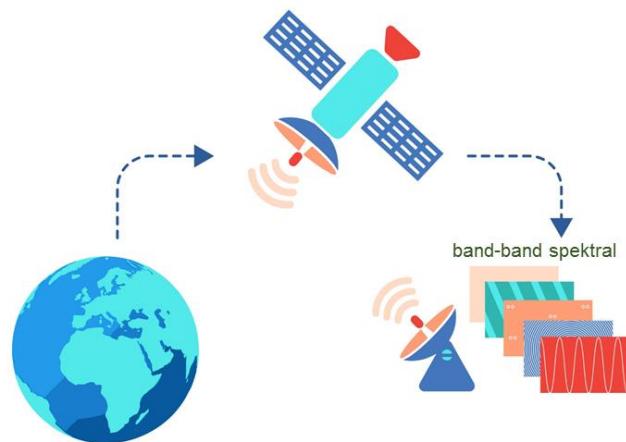
Gambar 5.1. Satelit buatan yang mengelilingi bumi¹⁰.

¹⁰ <https://news.itu.int/world-space-week-itus-contribution-to-world-united-by-space/>

5.2. Data Penginderaan Jauh Satelit

Penginderaan Jauh (*Remote Sensing*) adalah ilmu, seni dan teknik untuk memperoleh informasi suatu objek, daerah, dan/atau fenomena melalui analisis data yang diperoleh dengan suatu alat tanpa harus kontak langsung dengan objek, daerah, atau fenomena yang dikaji (UGM, 2017). Dengan menggunakan data penginderaan jauh, wilayah yang sulit untuk diakses oleh manusia sekalipun (contohnya pegunungan terjal) dapat dianalisis dan menghasilkan informasi yang dapat dipercaya. Alat pengindera (yang tidak berhubungan langsung dengan objeknya sendiri) tersebut adalah alat yang pada waktu perekaman objek tidak berada di permukaan bumi, tetapi berada di angkasa maupun luar angkasa. Alat tersebut misalnya adalah satelit, pesawat udara dan balon udara.

Sebagai alat pengindera jauh, satelit mengirim gelombang elektromagnetik sebagai “pengindera” ke bumi, lalu menangkap pantulan dari gelombang tersebut dan mengirimnya kembali ke stasiun di bumi. Kiriman satelit tersebut direkam sebagai data dalam format band-band spektral (lihat Gambar 5.2). Tiap band dapat berupa citra foto maupun non-foto dan berisi data hasil penginderaan dengan gelombang tertentu (Bitar, 2018).



Gambar 5.2. Ilustrasi data band-band spektral dari satelit.

Berdasarkan spektrum elektromagnetiknya, citra foto dapat dibedakan menjadi:

- Foto ortokromatik yang dibuat dengan menggunakan spektrum tampak dari band (saluran) biru hingga sebagian hijau (0,4 – 0,56 mikrometer).
- Foto ultraviolet yang dibuat dengan menggunakan spektrum ultra-violet dengan panjang gelombang 0,29 mikrometer.

- Foto pankromatik yang dibuat menggunakan spektrum tampak mata. Foto pankromatik dibedakan menjadi 2 yaitu pankromatik hitam putih dan foto infra merah.

Foto pankromatrik hitam-putih digunakan dalam berbagai bidang, misalnya:

- Di bidang pertanian, digunakan untuk pengenalan dan klasifikasi jenis tanaman, evaluasi kondisi tanaman, dan perkiraan jumlah produksi tanaman.
- Di bidang kehutanan, digunakan untuk identifikasi jenis pohon, perkiraan volume kayu, dan perkembangan luas hutan.
- Di bidang sumber daya air, digunakan untuk mendeteksi pencemaran air, evaluasi kerusakan akibat banjir, juga agihan air tanah dan air permukaan.

Sedangkan contoh penggunaan foto inframerah berwarna di bidang pertanian dan kehutanan adalah untuk mendeteksi atau membedakan tanaman yang sehat dan tanaman yang terserang penyakit.

Berdasarkan spektrum elektromagnetiknya, citra non-foto dapat dibedakan menjadi:

- Citra infra merah termal, yaitu citra yang dibuat dengan spektrum infra merah thermal.
- Citra radar dan citra gelombang mikro, yaitu citra yang dibuat dengan spektrum gelombang mikro.

Tiap jenis satelit buatan, sesuai dengan fungsinya, menghasilkan rekaman data hasil penginderaan dengan ciri-ciri khusus (Selfa, 2017). Data ini diberi nama tertentu yang terkait dengan fungsi maupun nama satelit, misalnya:

- Citra satelit cuaca: TIROS-1, ATS-1, GOES, NOAA AVHRR, MODIS dan DMSP;
- Citra satelit geodesi dan sumber daya alam dengan resolusi rendah: SPOT, LANDSAT, dan ASTER;
- Citra satelit geodesi dan sumber daya alam dengan resolusi tinggi: IKONOS dan QUICKBIRD;
- Citra satelit sumber daya alam dan lingkungan hidup generasi terbaru dengan resolusi spectral yang lebih lengkap: WorldView.

Sebagai contoh, di bawah ini diberikan penjelasan singkat tentang satelit Landsat dan SPOT dan band-band spektral yang dihasilkan:

Satelit Landsat-7 diluncurkan dari California AS pada April 1999 (Masek, 2020). Citra Landsat-7 terdiri dari 8 *band* atau lapis data. Data pada tiap band merupakan hasil penginderaan dengan panjang gelombang tertentu. Delapan band tersebut adalah: *blue, green, red, NIR (near infra red), SWIR 1* (terkait dengan kelembab tanah dan vegetasi), *thermal* (pemetaan termal dan kelembaban pada tanah), *SWIR 2 (hydrothermally altered rocks)* yang terkait dengan kandungan mineral pada tanah) dan pankromatik. Luas liputan Landsat-7 per *scene* adalah 185 km x 185 km. Landsat mempunyai kemampuan untuk meliputi daerah yang sama pada permukaan bumi pada setiap 16 hari, pada ketinggian orbit 705 km. Contoh citra Landsat 7 yang sudah dikalibrasi diberikan pada Gambar 5.3.

Satelit SPOT-4 (*Systeme Pour l'Observation de la Terre*) merupakan satelit milik Perancis yang diluncurkan pada 1986 dan diperbarui pada 1998 dengan menambahkan kemampuan baru¹¹. Setelah merekam 6.811.918 citra, SPOT-4 diterminasi pada 2013. Data SPOT-4 sampai sekarang masih tersedia dan dapat

¹¹ http://spot4.cnes.fr/spot4_gb/satellit.htm (diakses 20 Juli 2020)

digunakan. Data hasil penginderaan SPOT-4 terdiri dari 5 band, yaitu: *blue, green, red, SWIR* dan pankromatik.



Gambar 5.3. Contoh citra satelit Landsat 7¹².

5.3. Analisis Data Satelit SPOT-4 untuk Prediksi Panen Padi

Sebagaimana telah disebutkan sebelumnya, salah satu manfaat analisis data penginderaan jauh satelit adalah untuk memprediksi atau memperkirakan jumlah produksi tanaman, misalnya hasil panen padi.

Prediksi hasil panen padi beberapa bulan sebelum masa panen (pada saat sawah masih hijau) penting untuk menjamin ketersediaan beras. Hasil prediksi dapat dimanfaatkan oleh pemerintah (atau pihak pengambil keputusan lainnya) untuk memutuskan berapa banyak beras yang harus diimpor (jika panen tidak mencukupi) atau dieksport (jika panen surplus).

Jika tanpa menggunakan data pengindera jauh satelit, secara tradisional, prediksi panen padi dihitung dari data hasil survei di sawah-sawah (*ground-based*). Cara ini cukup bias, tingkat kesalahan dapat besar dan membutuhkan biaya besar. Selain itu, proses pengumpulan data dapat makan waktu lama, sehingga hasil prediksi yang dihitung berdasar data tersebut sudah terlambat untuk mengantisipasi kemungkinan buruk (misalnya kekurangan beras bagi penduduk).

Untuk keperluan perhitungan potensi panen padi, data hasil pengindera jauh memiliki kelebihan, yaitu: tersedia secara *real time* (waktu nyata), data dapat berkualitas bagus dan memberikan informasi yang memadai yagn terkait dengan pertumbuhan padi (Noureldin, 2013). Dengan demikian, perhitungan potensi panen dapat dilakukan pada saat tanaman padi masih hijau, misalnya 2-3 bulan sebelum masa panen (Gambar 5.4).

¹² https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LE07_C01_T1_TOA

Beberapa metoda untuk memprediksi hasil panen sawah dengan memanfaatkan data satelit telah dikembangkan. Sebagai contoh, pada bab ini dibahas metoda yang dibahas pada (Noureldin, 2013), yang memanfaatkan data satelit pengindera jauh SPOT-4.



Gambar 5.4. Berapa ton gabah/hektar akan dipanen dari sawah yang masih hijau ini?

5.3.1. Konsep Dasar

Sebagaimana telah dibahas sebelumnya, sistem satelit mengukur (merekam) berbagai band spektral pada rentang tengah infra-red yang nampak pada spektrum elektromagnetik. Penyerapan spektral biasanya terjadi pada panjang gelombang elektromagnetik dengan rentang 670 sampai 780 nm. Terkait dengan tumbuhan, klorofil pada daun terindikasi “menyerap” banyak gelombang pada rentang 0.45 μm sampai 0.67 μm dan memiliki pantulan yang tinggi terhadap gelombang near infrared (0.7-0.9 μm). Gelombang near infrared bermanfaat untuk survei dan pemetaan vegetasi karena “steep gradient” 0.7-0.9 μm tersebut hanya diproduksi oleh vegetasi atau tumbuh-tumbuhan.

Tumbuhan yang sehat memiliki nilai Normalized Difference Vegetation Index (NDVI) yang tinggi karena sifatnya yang hanya relatif sedikit memantulkan spektrum merah (*red*). Vegetation Index (VI) menyatakan besaran ukuran optikal terhadap “kehijauan” canopy (lapisan terluar daun) dari tumbuhan, dalam hal ini padi. Nilai VI mengindikasikan potensi fotosintesis dari klorofil daun, luas daun, penutup dan struktur “canopy” daun. VI dapat diperoleh dari data pengindera jauh satelit maupun dari alat observasi di tanah.

NDVI terkait dengan beberapa parameter tumbuhan yang berpengaruh terhadap jumlah panen (produksi “buah”) yang akan dihasilkan tumbuhan tersebut. Parameter-parameter tersebut dipengaruhi oleh kesuburan tanah, kelembaban tanah dan densitas (kerapatan) tumbuhan. Dengan demikian, NDVI dapat digunakan untuk mengestimasi hasil panen sebelum panen benar-benar dilakukan.

Data lain yang dibutuhkan untuk memprediksi panen adalah Leaf Area Index (LAI) yang merepresentasikan parameter biofisis terkait dengan pertumbuhan tanaman. LAI memberikan informasi

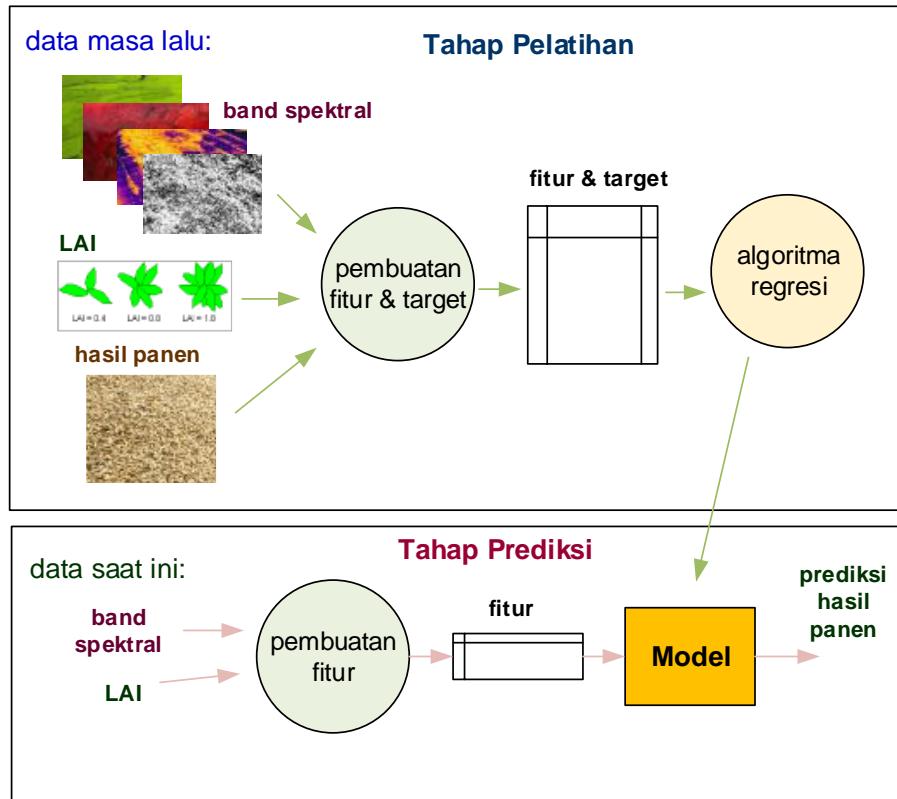
densitas (tingkat kerapatan) dari foliage yang berhubungan erat dengan kapasitas atau kemampuan tanaman dalam melakukan fotosintesis dan evapo-transpirasi.

Dengan demikian, VI dan LAI memiliki korelasi yang tinggi terhadap kondisi fisik tanaman dan produktivitas tamanan.

Salah satu algoritma Machine Learning yang dapat dimanfaatkan untuk membuat model prediksi adalah regresi. Algoritma ini menerima masukan himpunan data yang memiliki variabel-variabel prediktor (yang digunakan untuk memprediksi sebuah nilai) dan target (yang akan diprediksi) untuk membuat model. Ada dua tipe regresi, yaitu linear dan non-linear. Komputasi pada regresi non-linear lebih kompleks dibandingkan yang linear.

5.3.2. Rancangan Model Regresi untuk Prediksi Panen Padi

Secara umum, pemanfaatan algoritma regresi untuk membuat model terdiri dari dua tahap, yaitu tahap pelatihan dan tahap prediksi ([lihat Gambar 5.5](#)). Pada tahap pelatihan, model dibuat berdasar himpunan data (*dataset*) pelatihan dan target yang diumpulkan ke algoritma. Model tersebut lalu diuji menggunakan data uji. Apabila dari hasil pengujian didapat bahwa model tersebut berkualitas bagik (misalnya tingkat akurasi tinggi), maka pada tahap selanjutnya model dapat dimanfaatkan untuk memprediksi nilai target. Dalam konteks ini, nilai target yang akan diprediksi adalah hasil panen padi dalam satuan ton/hektar. Tentang data yang dibutuhkan untuk membuat model, akan dibahas lebih lanjut di bawah.



Gambar 5.5. Tahap pelatihan dan pemanfaatan model untuk prediksi hasil panen padi.

5.3.3. Penyiapan Data untuk Pembuatan Model

Sebelumnya sudah diidentifikasi bahwa VI, NDVI dan LAI merupakan data penting yang dapat mengindikasikan hasil panen padi. Nilai VI dan NDVI terkait dengan band-band spektral satelit, sedangkan nilai LAI diukur dengan alat di lokasi sawah. Pertanyaannya adalah: Bagaimana memperoleh nilai-nilai VI dan NDVI (yang benar atau representatif) dari data satelit? Bagaimana mendapatkan nilai LAI? Bagaimana menyiapkan himpunan data yang siap untuk diumpulkan ke algoritma regresi, sehingga model prediksi dapat dihitung?

Untuk dapat mencari solusi terhadap pertanyaan-pertanyaan tersebut, dibutuhkan langkah yang cukup panjang, melibatkan beberapa tahap kegiatan dan komputasi yang cukup kompleks. Bahkan, untuk keperluan pembuatan model prediksi dengan algoritma regresi, “pekerjaan besar dan kompleks” dilakukan pada tahap penyiapan data. Karena buku ini dimaksudkan hanya untuk memperkenalkan ilmu sains data (data science), di bab ini langkah-langkah dan komputasinya ditunjukkan secara garis besar saja dan tidak dipaparkan dengan detil.

Lokasi dan area sawah yang dipilih tentunya haruslah yang diindera oleh satelit SPOT-4. Pada kasus ini, area sawah yang digunakan untuk eksperimen berlokasi di Kafr El-Sheikh Governorate, Mesir. Luas sawah adalah 2.4 ha.

Data satelit dan LAI diambil sembilan puluh hari sebelum masa panen tahun 2008 dan 2009, pengukuran dilakukan pada 60 kotak di area sawah tersebut berdasar pembagian dengan grid ([lihat Gambar 5.6](#)). Tiap kotak melingkup area 400 m^2 ($20 \text{ m} \times 20 \text{ m}$) yang identik dengan sebuah pixel pada tiap band SPOT. Nilai-nilai VI, NDVI dan LAI, yang berperan sebagai bagian dari data pelatihan, dihitung pada tiap kotak. Data panen riil (yang menjadi [target](#) masukan algoritma regresi) diambil pada pada 24 Mei dan 23 Mei tahun 2008 dan 2009 juga dihitung untuk tiap kotak dalam satuan ton/hektar.



Gambar 5.6. Pembagian area sawah menjadi 60 kotak (grid).

Adapun rincian dari data (yang berperan sebagai prediktor) pada tiap kotak dijelaskan lebih lanjut di bawah ini.

Tiga tipe jenis data yang digunakan pada tiap kotak sawah adalah: Data spektral yang dikumpulkan langsung dari satelit SPOT (band green, red dan near-infrared), enam nilai indeks vegetasi hasil kalkulasi dengan rumus tertentu, LAI dan nilai panen padi hasil observasi.

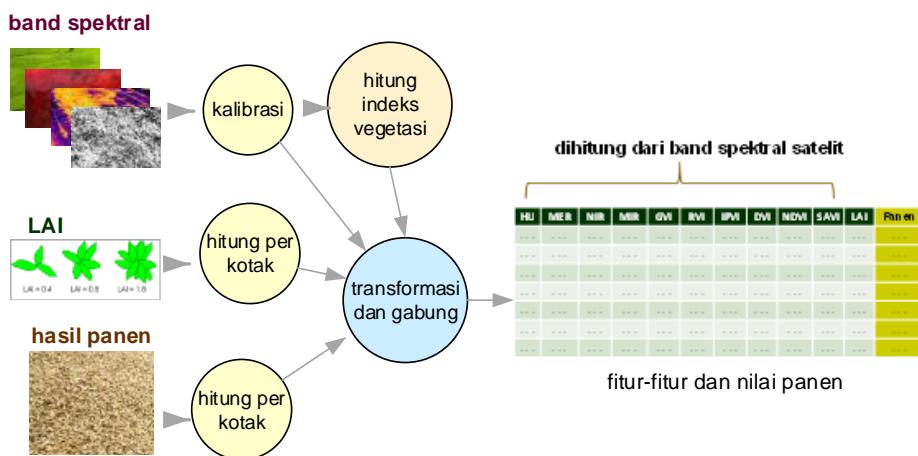
Lima nilai indeks vegetasi dihitung dengan menggunakan rumus matematika tertentu dengan menggunakan nilai band [merah](#) dan [near-infrared](#). Nilai indeks tersebut adalah: *Green Vegetation Index (GVI)*, *Ratio Vegetation Index (RVI)*, *Infrared Percentage Index (IPVI)*, *Difference Vegetation Index (DVI)*, *Normalized Difference Vegetation Index (NDVI)* dan *Soil Adjusted Vegetation Index (SAVI)*. Rumus matematika untuk menghitung nilai-nilai ini dapat [dilihat di](#) (Noureldin, 2013).

Namun perlu disampaikan di sini bahwa data asli dari satelit SPOT ternyata “belum bagus” atau presisi. Sebelum data digunakan, data satelit masih perlu diperbaiki dan dikalibrasi terlebih dahulu. Tahap ini melibatkan komputasi yang cukup kompleks.

Sedangkan *Leaf Area Index (LAI)* diukur di area sawah dengan cara sbb: Pada tiap kotak sawah, lima (sampel) nilai LAI dibaca dengan menggunakan alat penganalisis kanopi tumbuhan LAI-2000. Lima nilai LAI tersebut kemudian dirata-rata. Nilai rata-rata ini yang digunakan sebagai masukan pada pembuatan model.

Pada akhir musim tanam padi, data hasil panen digunakan untuk menghitung panen tiap kotak sawah dan juga digunakan untuk menghitung rata-rata panen (ton/ha) sebagai nilai target.

Pada tiap kotak, band SPOT-4 yang digunakan adalah band hijau (green), merah (red), near infra-red (NIR), middle infra-red (MIR). Dengan demikian, dari seluruh 60 kotak area sawah akan dihasilkan, pada tiap kotak akan dihasilkan 4 nilai band SPOT-4, data indeks vegetasi (6 buah nilai indeks), 1 nilai LAI dan 1 nilai hasil panen padi dalam satuan ton/hektar (lihat Gambar 5.7). Hasil akhir dari penyiapan data ini berupa data dalam format tabular yang terdiri dari 60 baris, sesuai dengan jumlah kotak sawah (sebagaimana ditunjukkan pada Gambar 5.6).



Gambar 5.7. Ilustrasi langkah penyiapan data.

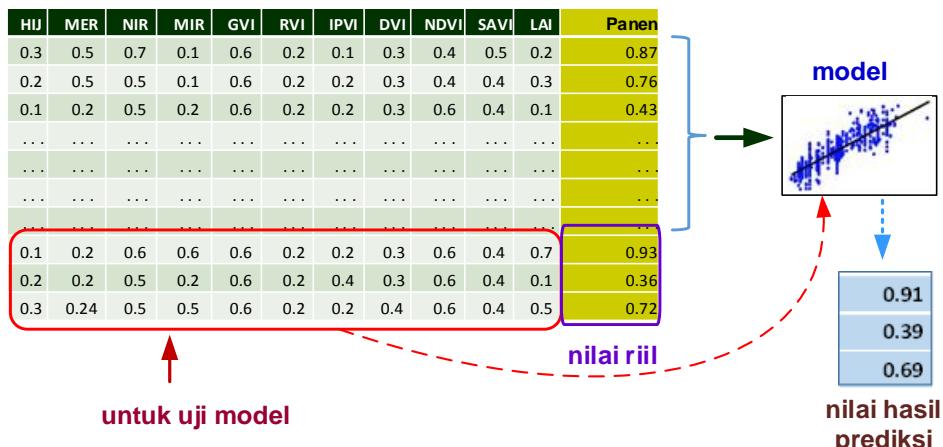
Penyiapan data dilakukan untuk dua tiap musim tanam dan panen pada tahun 2008 dan 2009. Masing-masing himpunan data akan digunakan untuk membuat sebuah model. Jadi akan dibuat 2 model prediksi, satu model berdasarkan data tahun 2008, satu lagi berdasarkan data tahun 2009.

5.3.5. Pembuatan dan Pengujian Model Regresi

Sebagaimana ditunjukkan pada Gambar 5.5, pengembangan model regresi terdiri dari 2 tahap, yaitu pembuatan model dan pemanfaatan model. Pada tahap pembuatan model, harus dipastikan terlebih dahulu bahwa model yang dibuat berkualitas bagus sehingga dapat dimanfaatkan untuk memprediksi

nilai panen. Untuk keperluan pengujian model, dibutuhkan data yang terpisah dari data yang digunakan untuk membuat model.

Pada kasus ini, dari 60 baris pada data tabular yang siap diumpulkan ke algoritma regresi, 50 baris yang dipilih secara acak digunakan untuk pembuatan model prediksi, sedangkan 10 sampel sisanya digunakan untuk menguji model. Tahap pembuatan dan pengujian model ini diilustrasikan pada Gambar 5.8.



Gambar 5.8. Pembuatan dan pengujian model regresi: Untuk regresi sederhana hanya dipilih 1 kolom, untuk multiple-regressi dipilih 2 kolom yaitu LAI dan 1 kolom lainnya.

Sebagaimana ditunjukkan pada Gambar 5.8, setelah model prediksi dibuat dengan 50 baris data (rekord), maka baris demi baris dari data uji “diumpulkan” ke model untuk mendapatkan nilai prediksi panen. Hasil prediksi tersebut dibandingkan dengan nilai panen riil. Dari hasil perbandingan tersebut, dapat dihitung sebuah nilai statistik (koefisien R^2) yang menyatakan kualitas dari model. Makin besar nilai R^2 (mendekati 1), makin bagus model tersebut.

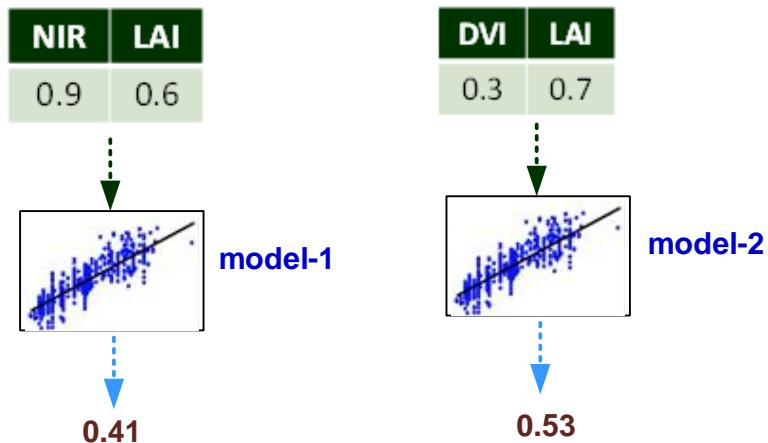
Pada kasus ini, model yang dibuat berdasar algoritma regresi sederhana (simple regression) yang hanya menggunakan 1 kolom prediktor (yaitu HIJ, MDER, NIR, dll.) dan multiple-regression yang menggunakan 2 kolom prediktor (pasangan kolom LAI dengan kolom lainnya, misalnya HIJ-LAI, MER-LAI, NIR-LAI, MIR-LAI, GVI-LAI, dll).

Dari hasil pengujian model, didapatkan hasil penting sebagai berikut:

- Model multiple-regression memiliki kualitas yang baik (R^2 secara umum lebih besar dari 0.89);
- Model dengan nilai R^2 tertinggi adalah model yang dibuat dengan data pelatihan dengan kolom NIR – LAI dan DVI-LAI, dimana nilai R^2 adalah 0.97.

5.3.6. Pemanfaatan Model Regresi

Berdasarkan hasil pengujian tersebut, jika model yang terbaik akan digunakan untuk memprediksi nilai panen padi pada masa yang akan datang (misalnya tahun 2010, 2011, 2012, dst), maka data input yang dibutuhkan adalah nilai NIR, DVI dan LAI. Model akan memberikan keluaran hasil prediksi berupa angka panen dalam satuan ton/hektar (lihat Gambar 5.9).



Gambar 5.9. Ilustrasi pemanfaatan model untuk prediksi.

Model yang dihasilkan hanya dapat dimanfaatkan pada lingkungan dan kondisi tertentu yang dijelaskan pada (Noureldin, 2013).

5.4. Penutup

Bab ini telah memberikan ilustrasi tentang bagaimana analisis data satelit dilakukan khususnya untuk keperluan pembuatan model prediksi panen padi berdasar pada data satelit SPOT yang direkam pada saat sawah masih menghijau.

Hal-hal detil yang terkait dengan bagaimana menyiapkan data dan menguji model tersebut tidak diberikan (dapat dilihat pada (Noureldin, 2013)). Pada kasus ini, data scientist haruslah menguasai ilmu yang memadai di bidang penginderaan jauh (*remote sensing*) satelit, bagaimana mendapatkan data satelit dan memahami data tersebut, bidang biologi atau pertanian yang terkait dengan tanaman padi dan produksinya, bidang statistik untuk menyiapkan data maupun menguji model dan teknik pada Machine Learning khususnya untuk pemodelan prediksi.

Referensi

- (Bitar, 2018) Bitar, *Penginderaan Jauh: Pengertian, Sistem, Jenis, Manfaat & Cara Kerjanya Lengkap*, <https://seputarilmu.com/2018/12/penginderaan-jauh.html>, Desember 21, 2018 [Diakses: 25 Januari 2020]
- (Bitar, 2019) Bitar, *Pengertian, Fungsi Dan Macam-Macam Satelit Beserta Contohnya Terlengkap*, <https://www.gurupendidikan.co.id/satelit/>, 29/11/2019 [Diakses: 26 Jan 2019].
- (Bitar, 2020) Bitar, *Penginderaan Jauh: Pengertian, Sistem, Jenis, Manfaat & Cara Kerjanya Lengkap*, <https://seputarilmu.com/2018/12/penginderaan-jauh.html>, Desember 21, 2018 [Diakses: 25 Januari 2020]
- (Masek, 2020) J.G. Masek, *Landsat 7*, Landsat Science, NASA, <https://landsat.gsfc.nasa.gov/landsat-7/> [Diakses 20 Juli 2020].
- (Noureldin, 2013) N.A. Noureldin, M.A. Aboelghar, H.S. Saudy, A.M. Ali, "Rice yield forecasting models using satellite imagery in Egypt", *The Egyptian Journal of Remote Sensing and Space Sciences* (2013) 16, 125–131.
- (Ritter, 2014) Malcolm Ritter, *How Many Man-Made Satellites Are Currently Orbiting Earth?*, 28 Maret 2014, <https://talkingpointsmemo.com/idealab/satellites-earth-orbit> [Diakses: 2 Feb 2020]
- (Selfa, 2017) *Macam-Macam Jenis Citra Satelit dan Penggunaannya Serta Menggabungkan Band Pada Landsat*, <https://selfaseptianiaulia.wordpress.com/2013/05/17/pertemuan-1-macam-macam-jenis-citra-satelit-dan-penggunaannya-serta-menggabungkan-band-pada-landsat/> [Diakses: 25 Januari 2020]
- (UGM, 2017) Teknik Geologi UGM, *Pemanfaatan Citra Penginderaan Jauh Sebagai Informasi Permukaan Bumi, Ilmu Geologi dan Mitigasi Bencana Alam*, 23 May 2017, <https://mitgeo.ft.ugm.ac.id/2017/05/23/pemanfaatan-citra-penginderaan-jauh-sebagai-informasi-permukaan-bumi-ilmu-geologi-dan-mitigasi-bencana-alam/>, [Diakses: 25 Jan 2020].

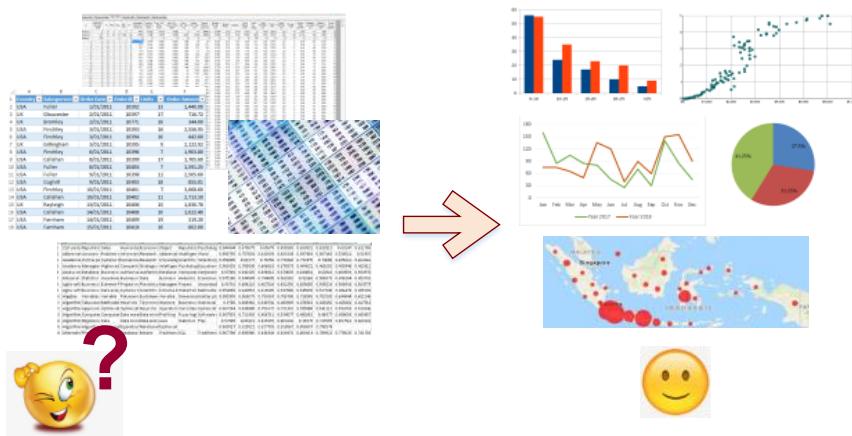
Bab 6 Penggalian Insights dari Data COVID-19 dengan Visualisasi, Studi Kasus: Data Korea Selatan

Oleh:

Veronica S. Moertini dan Kristopher D. Harjono

6.1. Pendahuluan

Pada Bab 1 telah dipaparkan bahwa data scientist (ilmuwan data) harus memiliki *curiosity, hacker-mind*, juga menguasai teknologi-teknologi yang dibutuhkan. Bab 1 juga telah membahas bahwa visualisasi data merupakan kegiatan penting pada tahap eksplorasi (mempelajari) data maupun untuk memaparkan **insights** yang merupakan informasi berharga atau menarik dari data. Jadi, salah satu pekerjaan penting seorang data scientist adalah membuat visualisasi dari data yang efektif dalam menjawab tujuan yang disasar.



Gambar 6.1. Ilustrasi visualisasi data.

Ketika orang dihadapkan dengan data, misalnya berupa angka-angka yang tertuang di tabel-tabel, apalagi dengan jumlah baris yang banyak (ratusan hingga jutaan), akan sulit bagi orang untuk memahami informasi dari data itu. Untuk itu, dibutuhkan bantuan berupa visualisasi yang merepresentasikan

ringkasan data (Gambar 6.1). Agar dapat menghasilkan visualisasi yang tepat (bagi pembaca atau audiens pada presentasi atau seminar) dan dapat menyampaikan informasi yang diinginkan, data scientist perlu melakukan langkah-langkah (Gambar 6.2):

Pertama, merumuskan insights apa saja yang ingin “digali” dan disampaikan dari data yang dimiliki. Namun, sebelum dapat merumuskannya, semua elemen data harus dipelajari dengan seksama dan teliti dulu sehingga data benar-benar dipahami dan dikuasai. Di sini, seringkali data scientist perlu menghitung ringkasan-ringkasan data dan kadang membuat “visualisasi dengan cepat” untuk menginterpretasikannya. *Curiosity* (rasa ingin tahu) yang kuat menjadi dasar bagi data scientist dalam merumuskan *insights* yang akan digali dari data. *Curiosity* dapat diterjemahkan menjadi pertanyaan yang akan dicari jawabnya.

Kedua, menentukan bentuk visualisasi, apakah itu grafik, text atau tabel. Bentuk perlu dipilih yang sesuai dan efektif untuk menyampaikan tiap informasi dan audiens atau pembaca yang ditarget.

Ketiga, memilih *tools, software* atau perangkat lunak yang tepat untuk tiap bentuk visual yang akan dibuat. Belum tentu satu tools dapat digunakan untuk membuat semua visualisasi yang diinginkan, jadi harus dicari dua atau lebih tools yang sesuai. Jika tools tidak tersedia atau mahal untuk dibeli, pilihan lain: membuat program atau “ngoding”, misalnya dengan bahasa Python yang gratis.

Keempat, menyiapkan data dengan format sedemikian rupa, sehingga dapat ditangani atau diproses oleh tools yang dipilih. Jika penyiapan data tidak dapat dilakukan dengan tools itu, maka perlu merancang algoritma dan dilanjutkan dengan pembuatan program dengan Python atau bahasa pemrograman lainnya. Jika visualisasi data akan dilakukan dengan membuat program, umumnya penyiapan data menjadi bagian dari program itu.

Kelima, membuat visualisasi dari data (dengan tools atau program) yang telah disiapkan. Ini biasanya tidak “sekali jadi”. Setelah bentuk visual ada, harus dievaluasi apakah sudah jelas, bagus, dan informasi tersampaikan. Jika masih ada yang kurang atau tidak tepat dalam merepresentasikan insights, visualisasi diperbaiki lagi.



Gambar 6.2. Tahap pembuatan visualisasi dari data.

Hacker mind dan penguasaan teknologi (tools maupun bahasa pemrograman) dibutuhkan di tahap ketiga, keempat maupun kelima. Seorang hacker memiliki kegigihan yang tinggi dalam “mengulik” hal-hal yang ingin dicarinya. Dalam konteks ini, kegigihan dibutuhkan dalam mencari dan mencoba-coba tools untuk mendapatkan yang paling sesuai, selama merancang algoritma dan membuat program, juga dalam mencoba-coba bentuk visual yang tepat, yang benar-benar dapat merepresentasikan insights yang ingin disampaikan.

Isi bab ini dimaksudkan untuk memberikan contoh penerapan dari langkah-langkah di atas disertai dengan contoh hasilnya.

6.2. Data COVID-19 di Korea Selatan

Pada saat bab ini ditulis, dunia sedang mengalami pandemi akibat Coronavirus Disease-2019 (COVID-19) yang disebabkan virus Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV2). Banyak negara dan wilayah di bawahnya (setting provinsi, kota, maupun kabupaten) menyediakan website yang menampilkan data maupun berbagai visualiasi yang terkait dengan paparan virus itu. Para pembaca mungkin sudah melakukan “browsing” di website-website tersebut dan memahami informasinya. Namun bagi yang ingin tahu, di “belakang” website-website itu, apa yang dikerjakan oleh si data scientist? Jawabannya tidak dapat ditemukan di situ.

Setelah mencari-cari data yang tersedia di Internet, ternyata penulis tidak berhasil mendapatkan data dari Indonesia yang memadai untuk dijadikan sebagai studi kasus. Yang didapatkan dari website penyedia data Kaggle, adalah data COVID-19 dari Korea Selatan (Korsel). Maka studi kasus dipilih untuk negara

Korsel. Maka, bab ini dimaksudkan untuk memberikan contoh atau salah satu opsi jawaban tentang hal-hal yang dilakukan data scientist dalam menyiapkan visualisasi terkait COVID-19.

Data yang tersedia merupakan hasil rekaman kasus-kasus mulai 20 Januari s/d 30 April 2020 (Kaggle, 2020). Setiap data berupa tabel, yang dapat dibuka dengan Excel. Sebagian tabel pada data COVID-19 tersebut adalah:

Data kasus:

- Case.csv (112 baris): Kasus-kasus terpapar COVID-19 dengan kolom *case_id, province, city, group, infection_case, confirmed, latitude* dan *longitude*.

Data pasien:

- PatientInfo.csv (3.388 baris): Data epidemis pasien COVID-19 dengan kolom *patient_id, global_num, sex, birth_year, age, country, province, city, disease, infection_case, infection_order, infected_by, contact_number, symptom_onset_date, confirmed_date, released_date, deceased_date, state* dan *confirm_released*.

Data time series:

- Time.csv (102 baris): data untuk status COVID-19 dengan kolom *date, time, test, negative, confirmed, released* dan *deceased*.
- TimeAge.csv (540 baris): data untuk status COVID-19 berdasar umur dengan kolom *date, time, age, confirmed*, dan *deceased*.
- TimeGender.csv (120 baris): data untuk status COVID-19 berdasar gender dengan kolom *date, time, sex, confirmed* dan *deceased*.
- TimeProvince.csv (1.734 baris): data untuk status COVID-19 untuk tiap provinsi dengan kolom *date, time, province, confirmed, released* dan *deceased*.

Contoh sebagian isi dari file-file di atas diberikan di Appendiks.

6.3. Bentuk-bentuk Visualisasi

Terdapat bermacam-macam bentuk visualisasi, namun mayoritas kebutuhan untuk memvisualisasikan data dapat dipenuhi dengan menggunakan beberapa bentuk saja. Di bawah ini diberikan bahasan singkat tentang beberapa bentuk visualisasi dan kapan cocok digunakan yang digunakan pada bab ini (ulasan lebih lengkap dan detil dapat ditemukan di (Knaflc, 2015)):

- Garis (line): Cocok untuk data “time-series” dan memberikan *trend*, misalnya harga satu atau lebih saham dari tanggal ke tanggal.
- Plot tersebar (*scatter plot*): Cocok untuk menunjukkan hubungan antara dua nilai variabel, misalnya berat terhadap tinggi badan dari para pemain sepakbola.
- Bar vertikal: Cocok digunakan ketika ingin ditunjukkan nilai-nilai beberapa variabel atau kategori agar terlihat perbandingannya.
- Bar horizontal: Sama dengan bar vertikal, namun lebih cocok digunakan ketika nama variabel atau kategori dari data panjang (misalnya, nama provinsi).

- Teks sederhana: Jika terdapat satu atau dua angka penting yang akan dibagikan, visualisasi ini pas untuk digunakan.

Contoh dari pemanfaatan mayoritas bentuk visualisasi di atas diberikan di sub-bab berikut ini.

Selain bentuk-bentuk di atas, terdapat bentuk visualisasi untuk kebutuhan khusus lainnya, misalnya:

- Visualisasi data pada peta: Visualisasi data pada titik-titik tertentu (misalnya kota, kecamatan, rumah sakit, dll.) dilakukan dengan membuat simbol (misalnya lingkaran) yang proporsional dengan nilai data. Pada peta area wilayah dengan batas-batas tertentu (propinsi atau wilayah yang lebih kecil), visualisasi data dapat dinyatakan dengan isi warna area yang berdegradasi sesuai nilai data, misalnya makin gelap makin besar (Kraak, 2005). Dengan visualisasi data pada peta, orang dengan cepat dapat mengaitkan data dengan lokasi untuk tujuan tertentu.
- Visualisasi graf: Yang dimaksud graf di sini adalah sekumpulan simpul (*vertices*) dan sisi (*edges*). Satu simpul merepresentasikan sebuah objek pada data tertentu (misalnya orang, dokumen, nomor telpon, transaksi, dll), sedangkan sisi merepresentasikan hubungan antar simpul. Visualisasi graf dimaksudkan untuk merepresentasikan graf dalam bentuk visual¹³. Bila objek-objek pada data saling terkait, visualisasi graf dapat dimanfaatkan untuk melihat hubungan antar objek tersebut¹⁴. Visualisasi graf sudah dimanfaatkan pada masalah kejahatan, contohnya di bidang keuangan (untuk mendeteksi adanya pola hubungan yang mencurigakan) dan keamanan jaringan (untuk mendeteksi aktivitas yang mencurigakan). Dalam konteks data COVID-19, pasien dapat dijadikan simpul graf, sedangkan penularan virus dari pasien ke pasien lain direpresentasikan sebagai sisi graf. Berdasar visualisasi graf COVID-19, akan dapat diidentifikasi adanya komunitas atau klaster COVID-19.

6.4. Penggalian Insights

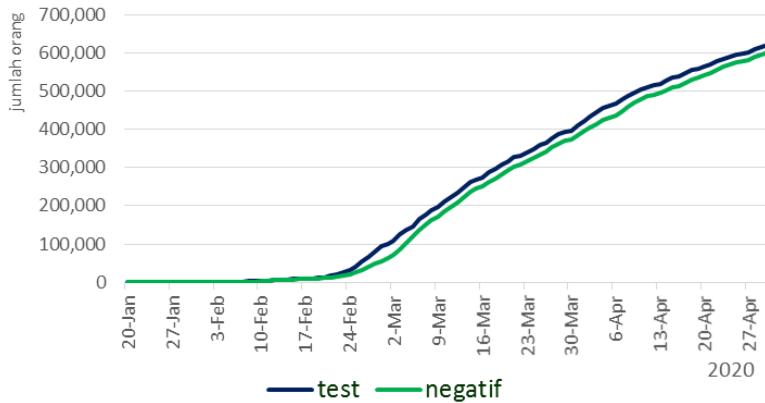
Pada subbab ini akan diberikan 14 contoh pertanyaan yang timbul yang didasari karena adanya curiositas terhadap data, apa yang dilakukan untuk menjawab pertanyaan, sampai mendapatkan visualisasi dan insights yang tersampaikan melalui visualisasi itu.

Pertanyaan-1: Bagaimana trend test COVID-19 dilakukan di Korsel dari waktu ke waktu? Apakah banyak orang yang “terbebas”?

Bentuk visual yang cocok adalah garis, yang merepresentasikan jumlah (test dan yang negatif) terhadap waktu. Pada file Time.csv, data sudah tersedia. Hanya saja, pada data asli, tanggal ditulis dengan format MM/DD/YYYY. Karena periode sudah diketahui (Januari s/d April 2020), format tanggal perlu diubah ke DD-nama bulan agar grafik lebih singkat dan cepat dibaca. Selanjutnya, dengan Excel dibuat grafik garis, warna garis diubah ke biru dan hijau. Hasilnya ditampilkan pada Gambar 6.3.

¹³ What is graph visualization, <https://linkurio.us/blog/why-graph-visualization-matters/> (diakses 15/8/20)

¹⁴ <https://dzone.com/articles/the-importance-of-graph-visualization-tools-explor> (diakses 15/8/20)

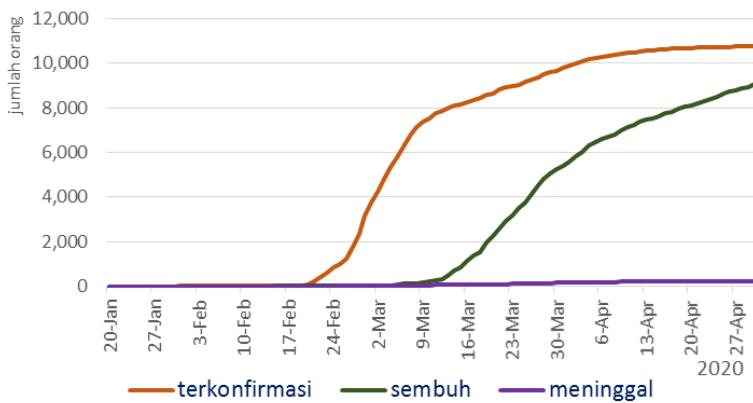


Gambar 6.3. Trend jumlah test dan hasil yang negatif.

Insights dari data: Test dilakukan dengan cepat (grafik naik eksponensial dari Februari ke akhir April) dan dari waktu ke waktu, hasilnya sebagian besar negatif.

Pertanyaan-2: Bagaimana trend akumulasi terkonfirmasi (positif), yang sembuh dan meninggal dari waktu ke waktu?

Sama dengan trend test, visualisasi yang cocok adalah grafik garis. Data tersedia pada file Time.csv, kolom date, confirmed, released dan deceased. Seperti sebelumnya, format tanggal perlu diubah, lalu grafik dibuat dengan Excel. Agar tiap garis merepresentasikan informasi yang berbeda, warna dibedakan dengan garis test dan negatif pada Gambar 6.4.

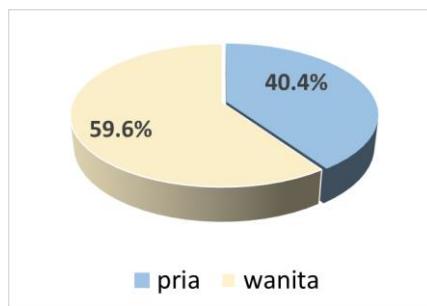


Gambar 6.4. Akumulasi terkonfirmasi, sembuh dan meninggal.

Insights dari data: Penyebaran COVID-19 di Korsel segera terkendali (grafik naik dari pertengahan Februari sampai akhir Maret, selanjutnya landai). Bagi yang terpapar, proses penyembuhan juga relatif cepat (grafik naik secara tajam dari 9 Maret sampai akhir April).

Pertanyaan-3: Jika di banyak negara, pria lebih banyak yang terinfeksi COVID-19, bagaimana dengan di Korsel?

Untuk menjawab pertanyaan tersebut, data dapat diperoleh dari file TimeGender.csv pada dua baris terakhir, yang berisi jumlah wanita dan pria yang terkonfirmasi terpapar COVID-19 dan yang meninggal pada tanggal 30 April 2020. Nilai kolom sex dan confirmed lalu digunakan untuk membuat pie-chart di Excel atau dengan Matplotlib Python. Pemilihan pie-chart dimaksudkan untuk menunjukkan “porsi kue” untuk pria dan wanita yang terinfeksi (Gambar 6.5).



Gambar 6.5. Persentase terinfeksi COVID-19 berdasarkan gender.

Insights dari data: Di Korsel lebih banyak wanita, sekitar 2/3 dari total, yang terinfeksi.

Pertanyaan-4: Bagaimana tingkat kematian dari yang terinfeksi? Apakah wanita, yang lebih banyak terinfeksi, memiliki resiko kematian yang lebih tinggi pula?

Untuk menjawabnya, digunakan data dua baris terakhir dari file TimeGender.csv yang digunakan untuk menjawab Pertanyaan-3. Persentase meninggal wanita dan pria dihitung dari jumlah per gender dan dari total yang terinfeksi dari kedua gender. Agar nilai dan perbandingan jelas, dipilih visualisasi teks (Gambar 6.6).

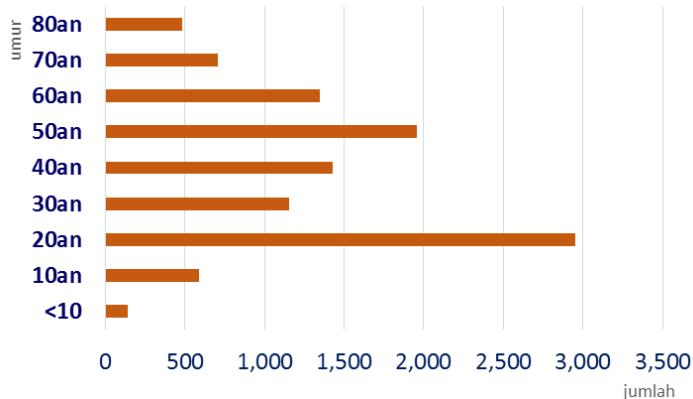


Gambar 6.6. Percentase meninggal berdasarkan gender.

Insights dari data: Dibanding banyak negara lain (misalnya USA, Italia, UK dan Perancis, dimana resiko kematian mencapai lebih dari 5%¹⁵), tingkat kematian akibat COVID-19 di Korsel lebih rendah. Pria memiliki resiko hampir dua kali dibanding wanita.

Pertanyaan-5: Berbagai hasil analisis data COVID-19 berdasar umur menunjukkan hasil bahwa dari satu negara ke negara lain, distribusi orang yang terserang COVID-19 berbeda-beda. Ada orang-orang yang mengira bahwa COVID-19 lebih banyak “menyerang kaum tua”. Bagaimana dengan di Korsel? Bagaimana persentase tiap kelompok umur?

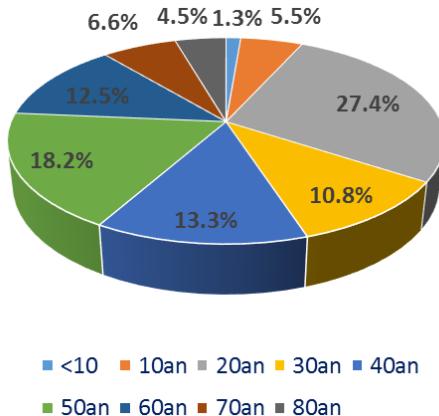
Untuk menjawabnya, data tersedia di file TimeAge.csv. Namun harus dipilih jumlah per kelompok umur pada tanggal terakhir, yaitu 30 April 2020. Untuk mevisualisasikan jumlah terinfeksi pada tiap kelompok umur, dipilih grafik bar horizontal agar perbandingan terlihat jelas. Dengan menggunakan Excel, hasil perhitungan jumlah per kelompok umur, digunakan untuk membuat grafik bar seperti ditunjukkan pada Gambar 6.7.



Gambar 6.7. Distribusi terkonfirmasi COVID-19 berdasar kelompok umur.

Setelah mendapatkan jumlah terinfeksi per kelompok umur, dapat dihitung persentasenya. Tiap jumlah dibagi dengan total terinfeksi (10.765). Untuk menunjukkan “porsi kue” (dari total 100%) per kelompok umur, dipilih visualisasi pie-chart dengan menyertakan angka persentase (Gambar 6.8).

¹⁵ <https://www.worldometers.info/coronavirus/#countries>

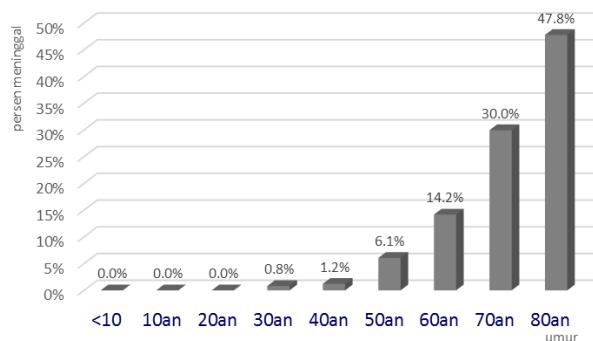


Gambar 6.8. Persentase terkonfirmasi COVID-19 berdasar umur.

Insights dari data: Yang terpapar COVID-19, terbanyak di umur 20-an, kedua di 50-an, ketiga di 40-an. Jadi, berbeda dengan anggapan banyak orang, di Korsel ternyata umur 20-an memiliki resiko tertinggi terinfeksi COVID-19.

Pertanyaan-6: Hasil analisis dari berbagai negara mengindikasikan bahwa semakin tua pasien, resiko kematian semakin tinggi. Untuk Indonesia, berdasar informasi pada website Peta Sebaran¹⁶, mulai umur 45 persentase meninggal di atas 40%. Bagaimana dengan pasien di Korsel?

Untuk menjawabnya, data harus disiapkan dari file TimeAge.csv. Data jumlah orang meninggal dipilih per kelompok umur pada tanggal terakhir, yaitu 30 April 2020. Lalu persentase dihitung untuk tiap kelompok umur dengan membaginya dengan jumlah total meninggal. Di sini, dipilih grafik bar vertikal agar kenaikan dari umur <10 sampai 80-an terlihat jelas. Dengan menggunakan Excel, hasil perhitungan persentase per kelompok umur digunakan untuk membuat grafik bar vertikal seperti ditunjukkan pada Gambar 6.9 .



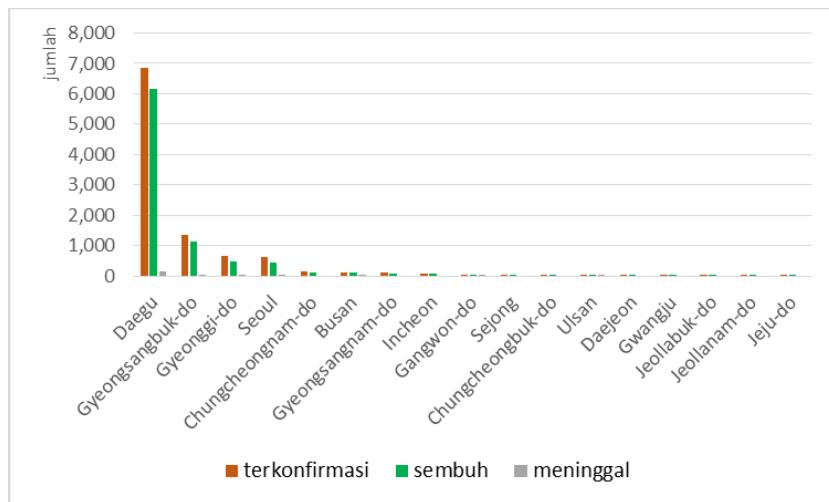
Gambar 6.9. Persentase meninggal karena COVID-19 berdasar umur.

¹⁶ <https://covid19.go.id/peta-sebaran>

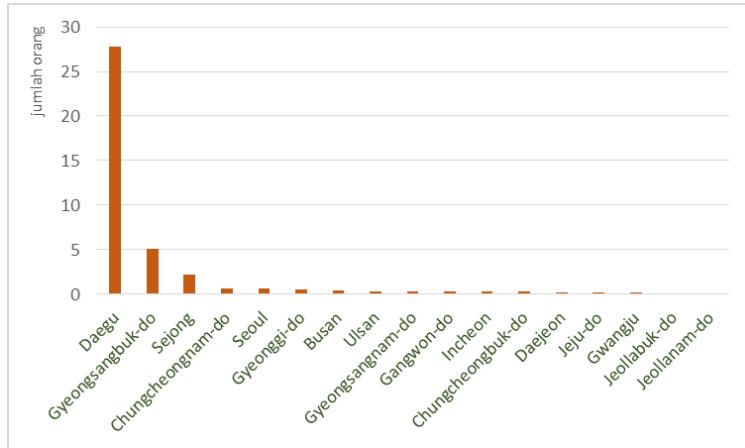
Insights dari data: Makin tua umur orang yang terinfeksi COVID-19 makin besar resiko kematianya. Resiko meningkat tajam sejak umur 50-an.

Pertanyaan-7: Korsel memiliki 17 provinsi. Apakah seluruh provinsi sudah terpapar? Bagaimana tingkat paparan terhadap jumlah penduduk? Bagaimana perbandingan terinfeksi (terkonfirmasi), sembuh dan meninggal di tiap provinsi?

Untuk menjawabnya, data diambil dari 17 baris terakhir dari file TimeProvince.csv. Hasilnya lalu diurutkan dari terbesar ke lebih kecil dan digunakan untuk membuat grafik bar vertikal pada Gambar 6.10, sedangkan perbandingan jumlah terkonfirmasi per 10.000 penduduk diberikan pada Gambar 6.11.



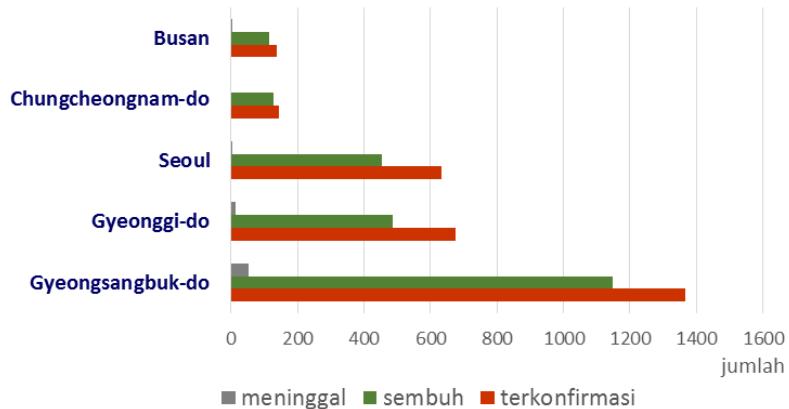
Gambar 6.10. Perbandingan jumlah terkonfirmasi, sembuh dan meninggal di seluruh provinsi.



Gambar 6.11. Jumlah terkonfirmasi per 10.000 penduduk di semua provinsi.

Insights dari data: Jumlah terinfeksi di provinsi Daegu, jauh melampaui yang lain, disusul Gyeongsakbuk-do, Gyeonggi-do, dan Seuol. Setelah itu, jumlah relatif sedikit.

Karena bar Daegu terlalu tinggi, perbandingan terkonfirmasi – sembuh – meninggal di provinsi lainnya tidak jelas. Maka, dibuat juga grafik bar untuk top-5 provinsi di bawah Daegu (Gambar 6.12).

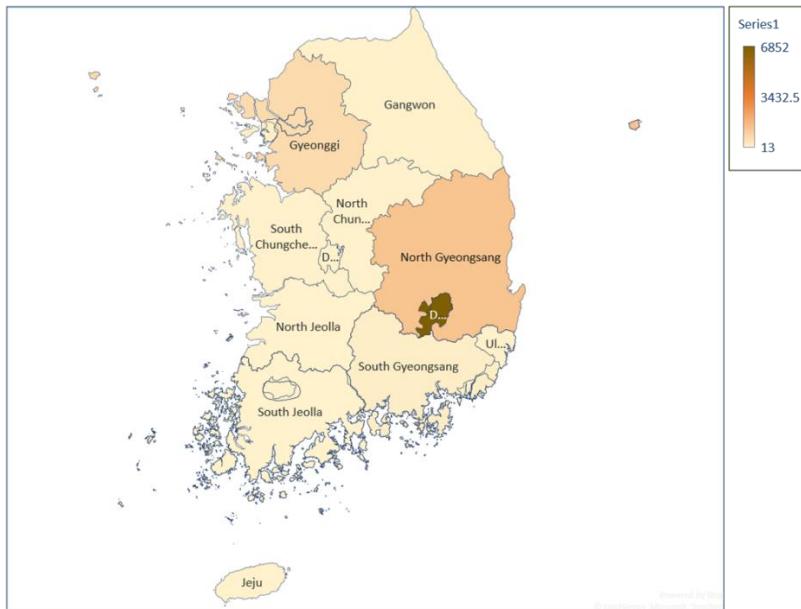


Gambar 6.12. Top-5 provinsi (di bawah Daegu).

Pertanyaan-8: Jika pada Gambar 6.12 ditunjukkan bahwa pada beberapa provinsi memiliki angka paparan yang tinggi, apakah lokasi mereka berdekatan?

Untuk menjawab pertanyaan itu, perlu dicari tools yang dapat memaparkan peta distribusi per provinsi. Excel versi 2016 ke atas sudah memiliki kemampuan untuk membuat visualisasi distribusi pada pada peta. Namun, pada saat membuatnya harus terkoneksi ke Internet untuk mendapatkan dengan peta.

Pada Gambar 6.13 diberikan hasil visualisasi yang dibuat dengan Excel. Opsi lain adalah membuat program dengan Python dengan menggunakan library Geopandas yang instalasinya tidak mudah karena membutuhkan kecocokan berbagai library. Program lalu dibuat dengan masukan data paparan tiap provinsi di atas dan peta Korsel.

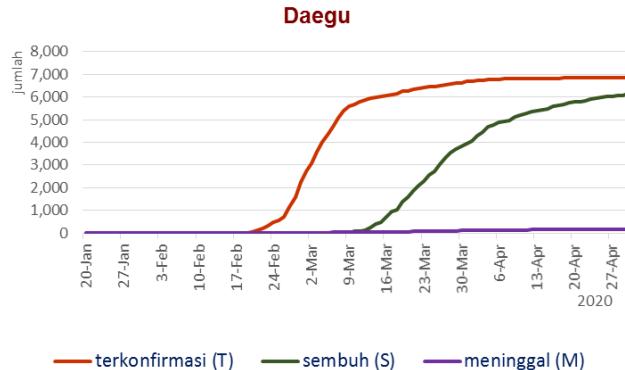


Gambar 6.13. Tingkat paparan pada tiap provinsi di Korsel.

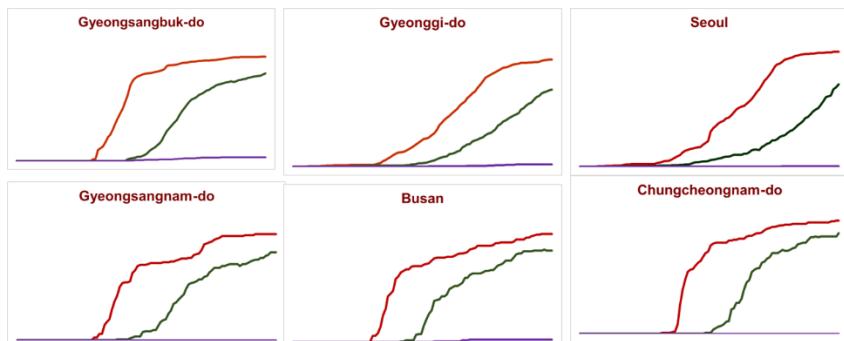
Insights dari data: Di sekitar provinsi Daegu, paparan cukup tinggi. Jadi, Daegu menjadi provinsi episentrum COVID-19. Episentrum lainnya terletak di sebelah utara, provinsi Gyeonggi dan Seoul yang berdekatan.

Pertanyaan-9: Bagaimana trend atau pola terkonfirmasi dan sembuh di tiap provinsi berdasarkan waktu?

Data tersedia di file TimeProvince.csv, namun harus dipilih dulu. Pemilihan data untuk tiap provinsi dapat dengan mudah dilakukan dengan Excel (fitur filter). Tanggal perlu diubah, lalu dibuat grafik garis yang menunjukkan trend. Untuk menghemat tempat di buku ini, grafik tunggal dibuat untuk provinsi Daegu yang memiliki kasus terkonfirmasi/terinfeksi terbanyak (Gambar 6.14), sedangkan provinsi-provinsi lain digabung dalam satu gambar dengan hanya menunjukkan garis trend (Gambar 6.15).



Gambar 6.14. Grafik akumulasi di provinsi Daegu yang memiliki jumlah terinfeksi terbanyak.

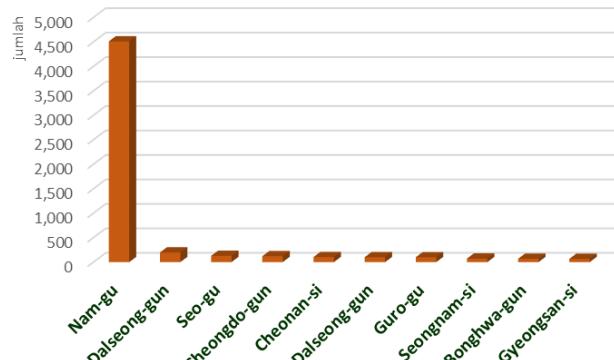


Gambar 6.15. Trend akumulasi terkonfirmasi, sembuh dan meninggal di 6 provinsi terbanyak (selain Daegu).

Insights dari data: Di semua provinsi, menjelang akhir April jumlah penambahan terinfeksi sudah mendekati nol. Penyebaran berhasil ditangani dengan baik. Selain itu, trend kesembuhan juga bagus, meningkat cepat dari Maret sampai akhir April.

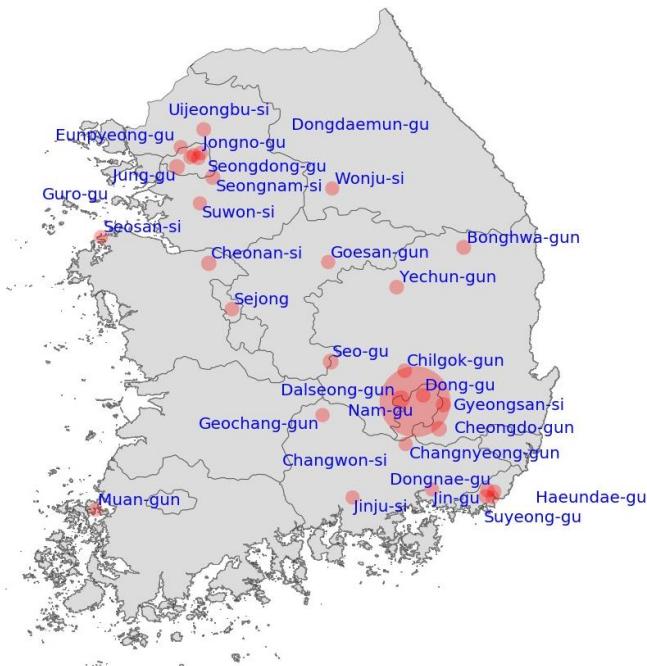
Pertanyaan-10: Bagaimana sebaran terinfeksi di kota-kota Korsel? Apakah terpusat di kota-kota tertentu dan terdapat episentrum?

Untuk menjawabnya, data belum tersedia. Namun, jumlah terinfeksi di tiap kota dapat dihitung dari file Case.csv. Pada tiap kota, dilakukan penjumlahan (sum) dari kolom confirmed pada semua baris untuk kota tersebut. Komputasi dilakukan dengan melakukan group-by berdasar kota untuk menjumlah nilai kolom confirmed. Ini dapat dilakukan di Excel, dengan membuat program menggunakan library Pandas pada Python, atau SQL pada basisdata relasional. Hasilnya lalu diurutkan dari terbesar ke lebih kecil dan digunakan untuk membuat grafik bar vertikal untuk 10 kota dengan jumlah terbanyak seperti ditunjukkan pada Gambar 6.16.



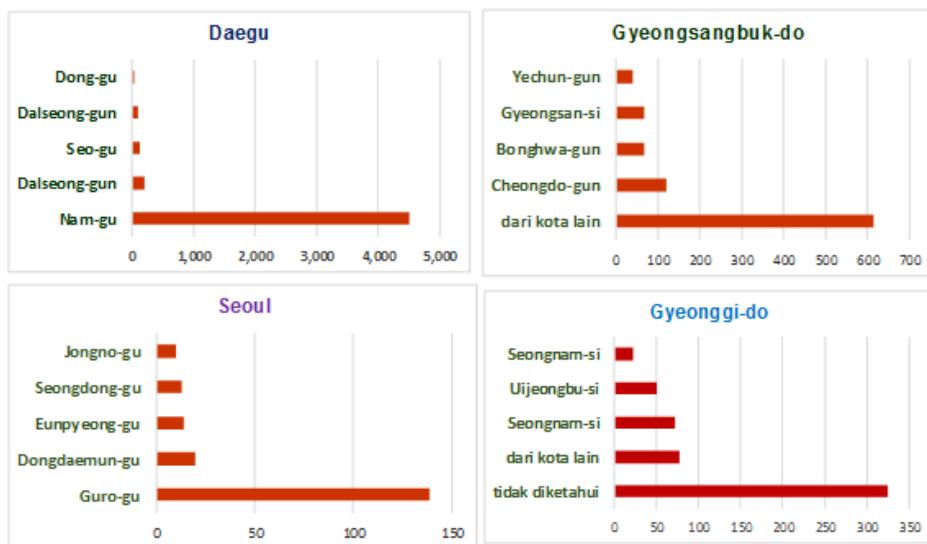
Gambar 6.16. Sepuluh kota dengan jumlah terinfeksi terbanyak di Korsel.

Pada Gambar 6.16 terlihat bahwa kota Nam-gu merupakan episentrum, dimana jumlah terinfeksi jauh melampaui kota-kota lainnya. Namun, Gambar 6.16 belum menjawab sebaran di kota-kota. Untuk itu, perlu dibuat visualisasi kota-kota dengan ukuran “tanda” yang sesuai dengan jumlah terinfeksinya. Kode program dapat dibuat dengan library Geopandas pada Python dimana dibuat lingkaran-lingkaran merah di kota-kota terinfeksi dimana diameter dibuat sebanding dengan jumlah terinfeksi. Data yang disiapkan untuk masukan program adalah: nama kota beserta jumlah paparannya, dan koordinat GPS (latitude dan longitude) yang dapat diambil dari Case.csv. Hasilnya ditunjukkan pada Gambar 6.17. Pada peta, terlihat sebaran COVID-19 di kota-kota Korsel dengan episentrum di Namgu dan sekitarnya.



Gambar 6.17. Peta sebaran paparan COVID-19 di kota-kota Korsel (makin besar lingkaran, makin banyak yang terpapar).

Untuk melengkapi peta pada Gambar 6.17, pada masing-masing provinsi lalu dihitung jumlah terinfeksi di tiap kota, hasilnya diurutkan dari terbesar ke terkecil. Proses dilakukan pada file Case.csv. Cara yang digunakan adalah filter (menyaring data untuk provinsi tertentu), group-by berdasar kota, lalu sort data dan dipilih lima teratas. Grafik bar horizontal dibuat dengan Excel dan hasilnya diberikan pada Gambar 6.18.

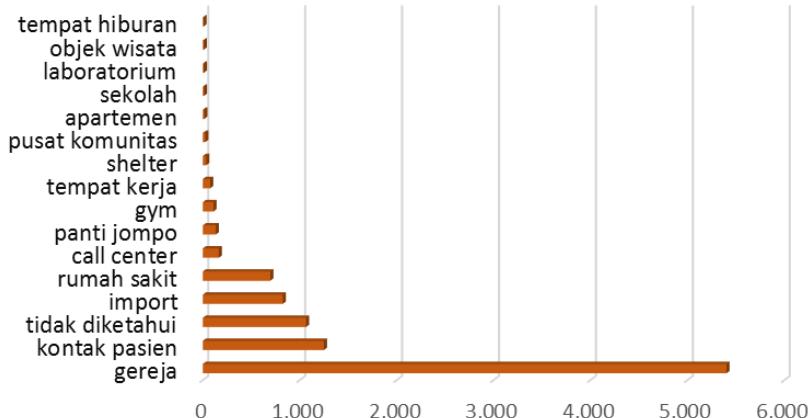


Gambar 6.18. Top-5 kota dengan sebaran paparan terbanyak di 4 provinsi.

Insights dari data: Penyebaran COVID-19 di Korsel hanya terjadi di beberapa kota dengan episentrum di Nam-gu, provinsi Daegu. Untuk provinsi dengan paparan terbanyak lainnya, hanya Seoul yang memiliki kota episentrum. Di Gyeonggi-do dan Gyeongsangbuk-do, kasus terbanyak berasal dari kota lain.

Pertanyaan-11: Bagaimana dengan asal paparan? Tempat-tempat mana saja yang paling banyak menjadi ajang penularan COVID-19?

Untuk menjawab, data belum tersedia namun dapat disiapkan dari file Case.csv dengan memanfaatkan kolom infection_case dan confirmed. Di sini perlu dibuat sebuah kolom baru, place_group, yang diisi dengan kategori tempat (sekolah, gereja, gym, dll.). Nilai kolom place_group ditentukan berdasar isi kolom infection_case. Perhitungan dengan group-by dilakukan untuk menjumlahkan nilai-nilai confirmed untuk tiap nilai di place_group. Hasilnya lalu diurutkan dari terbesar ke terkecil dan digunakan untuk membuat grafik bar horizontal pada Gambar 6.19.

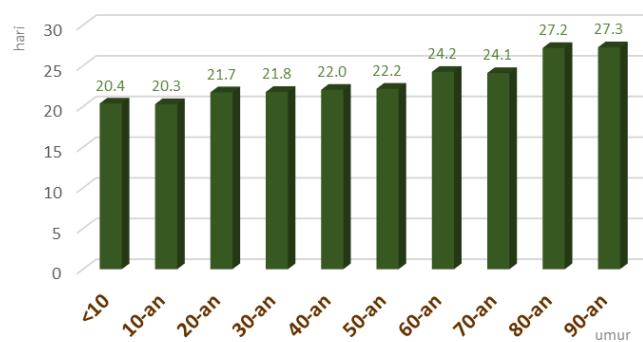


Gambar 6.19. Distribusi asal penularan COVID-19 di Korsel.

Insights dari data: Gereja dan rumah sakit merupakan tempat-tempat dimana mayoritas orang terpapar. Selain itu, orang dapat terpapar dari kontak dengan pasien dan berasal dari luar Korsel (import). Namun, terdapat lebih dari 1000 kasus yang tidak dapat diketahui darimana mereka tertular.

Pertanyaan-12: Berapa lama orang terinfeksi COVID-19 akan sembuh? Apakah umur berpengaruh terhadap lama sakit (dan dirawat di rumah sakit)?

Data belum tersedia, namun lama kesembuhan dapat dihitung dari file PatientInfo.csv (yang berisi data cukup detil dari 3.388 sampel pasien). Lama pasien sembuh dihitung dengan cara mengurangi nilai released_date dengan confirmed_date menggunakan Excel. Setelah itu, dengan group-by dihitung rata-rata kesembuhan tiap kelompok umur. Hasilnya digunakan untuk membuat grafik bar horisontal pada Gambar 6.20.

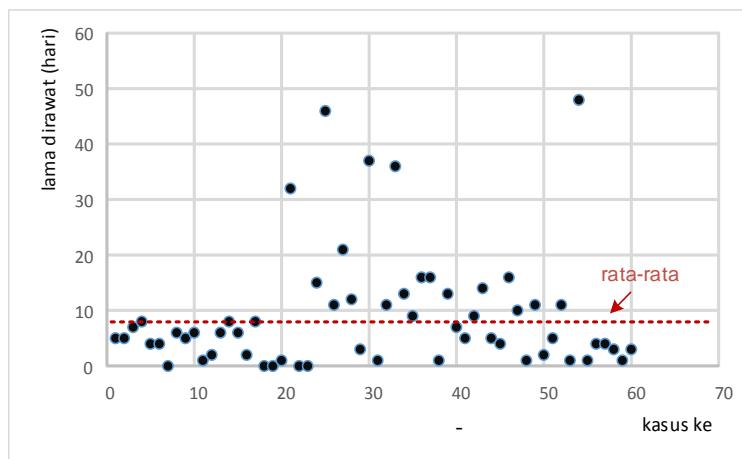


Gambar 6.20. Rata-rata lama sembuh berdasarkan umur.

Insights dari data: Rata-rata lama pasien sembuh lebih dari 20 hari dan secara umum naik berdasar umur. Peningkatan secara signifikan terjadi mulai umur 60.

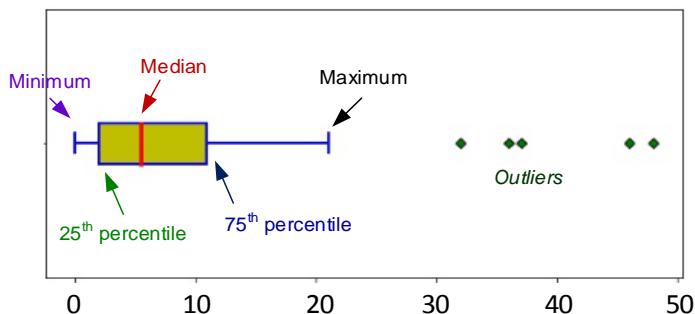
Pertanyaan-13: Untuk pasien yang meninggal, berapa lama pasien dirawat?

Data belum tersedia, namun lama kesembuhan dapat dihitung dari file PatientInfo.csv. Sebagaimana ditunjukkan pada Gambar 6.6, jumlah pasien meninggal di Korea relatif rendah. Kasus-kasus pada PatientInfo.csv harus dipilih dulu untuk mendapatkan kasus-kasus meninggal. Pemilihan dilakukan dengan filter dimana kolom state bernilai deceased (meninggal). Dari sini, hanya ditemukan 60 kasus. Kemudian, lama pasien dirawat (sampai meninggal) dihitung dengan mengurangi nilai deceased_date dengan confirmed_date. Setelah dilihat, ternyata jumlah hari pada 60 kasus bervariasi. Untuk menunjukkan variasi tersebut dibuat visualisasi dengan menggunakan scatter-plot pada tiap kasus (Gambar 6.21).



Gambar 6.21. Distribusi lama pasien dirawat untuk 60 pasien yang meninggal.

Karena berdasar data dari 60 kasus tersebut, jumlah hari dirawat bervariasi, perlu dibuat visualisasi berbasis statistik, yaitu boxplot, yang memberikan ukuran-ukuran sebaran jumlah hari dengan lebih rinci. Boxplot dibuat dengan library Matplotlib pada Python, dengan data masukan untuk 60 kasus di atas. Hasilnya ditunjukkan pada Gambar 6.22. Pada gambar ditunjukkan bahwa nilai minimum adalah 0 hari, 25th percentile (Q1) 2 hari, median (nilai tengah dari keseluruhan lama hari) 5.5 hari, 75th percentile (Q3) 11 hari, maksimum (Q3 + 1.5xIQR) 21 hari. Selain itu, terdapat kasus "pencilan" (outlier) dimana 5 pasien meninggal setelah dirawat lebih lama dari 21 hari.



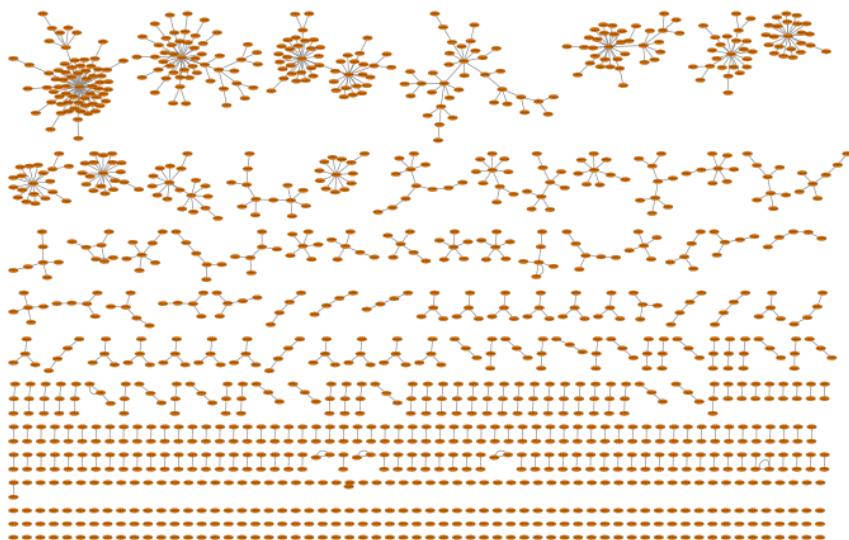
Gambar 6.22. Boxplot dari data lama pasien dirawat sebelum meninggal.

Insights dari data: Lama pasien dirawat sebelum meninggal bervariasi, terbanyak berada di rentang 2 sampai 11 hari, dengan median (nilai tengah) 5.5 hari. Angka 0 (nol) mengindikasikan bahwa kasus tersebut terkonfirmasi pada tanggal yang bersamaan dengan terkonfirmasi terinfeksi.

Pertanyaan-14: Bagaimana penyebaran Covid-19 di Korea, apakah terdapat klaster-klaster? Jika ada, bagaimana klaster-klaster di tiap provinsi?

Untuk menjawabnya, digunakan sampel kasus pada file PaintentInfo.csv. Pada file, terdapat kolom patient_id dan infected_by, dimana kolom terakhir ini berisi id dari kasus (lain) yang menginfeksi. Siapa menginfeksi siapa saja dapat divisualisasi dengan bentuk "graf". Jadi, perlu dicari tools atau software apa yang dapat memberikan visualisasi yang dapat dipahami. Setelah penulis melakukan eksperimen membuat visualisasi graf dengan beberapa software, akhirnya didapatkan Cytoscape¹⁷ yang dapat dimanfaatkan. Software ini menerima input data berformat csv. Setelah data csv dibaca, dipilih kolom yang digunakan (dalam hal ini patient_id dan infected_by) yang merepresentasikan node sumber dan target. Hasil visualisasi yang merepresentasikan "jaringan penularan" COVID-19 antar kasus di seluruh Korsel ditunjukkan pada Gambar 6.23 (gambar asli dipotong pada bagian bawah yang mengindikasikan tidak ada hubungan antar pasien). Pada gambar itu, sebuah ellip merepresentasikan satu kasus dan garis antar ellip menyatakan penularan. Di bagian atas terlihat ada "gerombolan" kasus-kasus yang saling terhubung, mengindikasikan adanya klaster-klaster. Selanjutnya, klaster-klaster di provinsi dapat dicari.

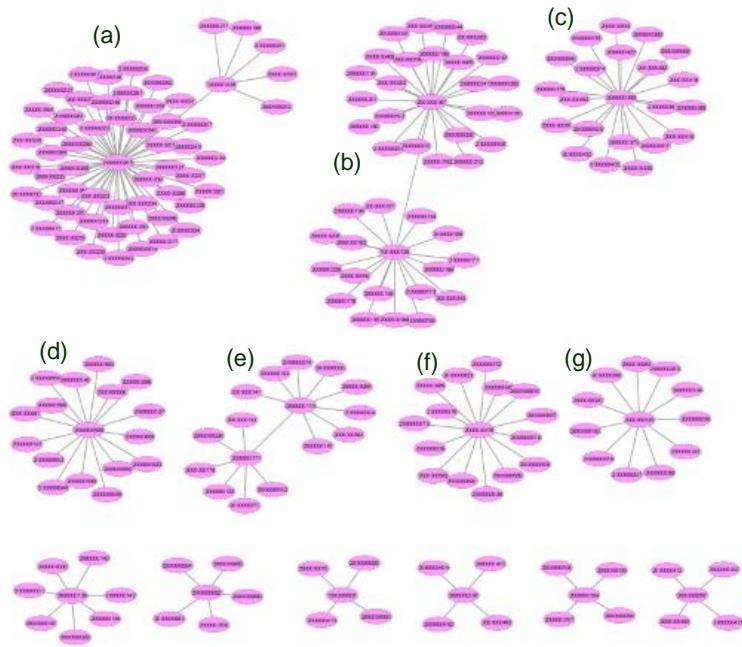
¹⁷ <https://cytoscape.org/>



Gambar 6.23. "Jaringan" penularan pada 3.388 kasus di Korsel, dimana terdapatnya banyak ellip-ellip yang terhubung (di bagian atas) mengindikasikan adanya klaster-klaster.

Dalam rangka mencari klaster di provinsi, dilakukan filter data pada PaintentInfo.csv untuk tiap provinsi yang memiliki kasus banyak (lihat Gambar 14 dan 15). Dari pemeriksaan hasil filter, ternyata di provinsi Daegu, dimana jumlah terinfeksi terbanyak, tidak terdapat klaster (ada kemungkinan sampel kasus di Daegu tidak lengkap). Klaster-klaster ditemukan di provinsi Gyeonggi-do, Chungcheongnam-do dan Gyeongsangnam-do. Sebagai contoh, berikut ini diberikan visualisasi graf untuk provinsi Gyeonggi-do dan Chungcheongnam-do. Hasil visualisasi dalam bentuk graf antar kasus diberikan pada Gambar 6.24 s/d 6.25. Pada gambar-gambar itu, nomor di tengah ellip menyatakan Id dari kasus.

Pada tiap klaster lalu dicari jumlah kasusnya dan "pusat penularnya". Untuk keperluan ini, perlu dibuat program Python dengan menggunakan *library* Panda. Fungsi utama program adalah untuk menghitung kemunculan kasus dan mencari Id yang paling banyak menginfeksi Id lainnya.



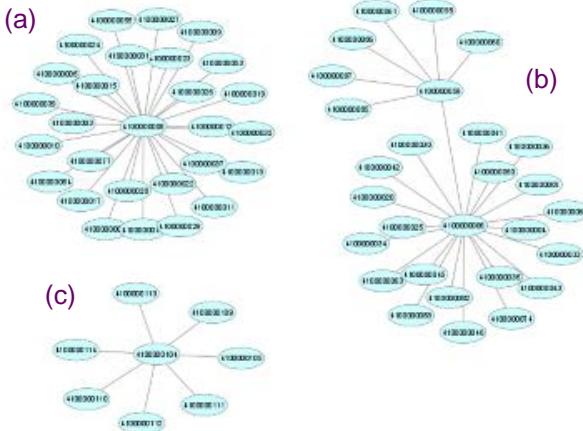
Gambar 6.24. Klaster-klaster penyebaran COVID-19 di provinsi Gyeonggi-do.

Pada Gambar 6.24 terdapat 7 klaster (a s/d g) ukuran sedang sampai besar dan selebihnya ada 6 klaster kecil (gambar bawah).

Klaster yang terjadi di Gyeonggi-do beserta informasi kota dan pusat penularnya diberikan di bawah ini:

- Klaster (a): 51 kasus di kota Seongnam-si dan Namyangju-si dengan pusat penular Id 20000000205.
- Klaster (b-1): 24 kasus di kota Bucheon-si dengan pusat penular Id -20000000167, dan Klaster (b-2): 18 kasus dengan pusat penular di kota Anyang-si, Bucheon-si, Gimpo-si, Gwangmyeong-si, Uijeongbu-si dan Pyeongtaek-si dengan pusat penular Id 1000000125.
- Klaster (c): 21 kasus di kota Gunpo-si, Seongnam-si dan Anseong-si dengan pusat penular Id 20000000309.
- Klaster (d): 16 kasus di kota Pyeongtaek-si dan Osan-si dengan penular Id 200000000508.
- Klaster (e-1): 8 kasus dengan penular Id 20000000114 dan Klaster (e-2) 7 kasus dengan penular Id 20000000111 di kota Gwangju-si, Ansan-si dan Seongnam-si.
- Klaster (f): 15 kasus di kota Uijeongbu-si, Dongducheon-si, Pocheon-si, Dongducheon-si, Yangju-si, dan Namyangju-si dengan penular Id 20000000476.
- Klaster (g): 11 kasus di kota Seongnam-si, Gwangju-si dan Uijeongbu-si dengan penular Id 2000000125.

Klaster terbanyak kedua ditemukan di provinsi Chungcheongnam-do (Gambar 6.25).



Gambar 6.25. Tiga klaster penyebaran COVID-19 di provinsi Chungcheongnam-do.

Adapun klaster yang terjadi di Chungcheongnam-do beserta informasi kota dan pusat penularnya diberikan di bawah ini:

- Klaster (a): 27 kasus di kota Cheonan-si dengan penular Id 410000008.
- Klaster (b-1): 21 kasus dengan penular Id 410000006 di kota Cheonan-si dan Asan-si dan Klaster (b-2) 6 kasus dengan penular Id 410000059 di kota Cheonan-si.
- Klaster (c): 7 kasus dengan penular Id 41000000104 di kota Seosan-si.

Berdasar data sampel tersebut, provinsi-provinsi lainnya tidak memiliki klaster berukuran besar. Seandainya didapatkan data detil dari seluruh pasien di Korsel, mungkin klaster-klaster dapat ditemukan.

Insights dari data: Klaster-klaster di 4 provinsi Korsel yang memiliki jumlah terinfeksi terbanyak yang sudah dijelaskan di atas. Selain itu, seseorang dapat menulari virus hingga mencapai 51 orang.

6.5. Penutup

Dengan telah diberikan contoh-contoh penerapan langkah-langkah pada penggalian insights atau informasi penting/berharga dari data dengan teknik visualisasi, diharapkan para pembaca mendapatkan gambaran tentang salah satu pekerjaan penting yang dikerjakan oleh data scientist atau ilmuwan data.

Pada bab ini, langkah-langkah pembuatan visualisasi data hanya diberikan inti-inti kegiatannya saja, tidak dipaparkan dengan detil. Pemaparan detil akan membutuhkan penjelasan langkah-langkah pemakaian tools yang digunakan atau algoritma program untuk yang dikerjakan dengan program. Hal ini akan membuat konten bab ini menjadi panjang dan kurang fokus. Bagi pembaca yang sedang mencari informasi tentang data science dan gambaran apa saja yang dilakukan oleh data scientist, bahasan teknis yang detil tersebut juga belum dibutuhkan. Pemanfaatan tools, perancangan algoritma dan pemrograman umumnya menjadi bagian dari kurikulum penyelenggara pendidikan di bidang data science.

Referensi

- (Kaggle, 2020) <https://www.kaggle.com/kimjihoo/ds4c-what-is-this-dataset-detailed-description> (diakses 16 Mei 2020)
- (Knaflic, 2015) Knaflic, C. Nussbaumer, "Story Telling with Data", Wiley Publ., 2015.
- (Kraak, 2005) Kraak, M. J., "Visualising Spatial Distributions", bab pada buku P. A. Longley, et al (Eds.), *Geographical information systems: principles, techniques, management and applications*, pp. book 49-65, Hoboken: Wiley & Sons, 2005.

Apendiks

Di bawah ini diberikan beberapa baris pada 4 file (berisi data "mentah") sebagai contoh.

Case.csv:

case_id	provin-ce	city	group	infection_case	con-firmed	latitude	longi-tude
1000001	Seoul	Guro-gu	TRUE	Guro-gu Call Center	98	37.50816	126.8844
1000002	Seoul	Dongdaemun-gu	TRUE	Dongan Church	20	37.59289	127.0568
1000003	Seoul	Guro-gu	TRUE	Manmin Central Church	41	37.48106	126.8943
1000004	Seoul	Eunpyeong-gu	TRUE	Eunpyeong St. Mary's Hospital	14	37.63369	126.9165

PatientInfo.csv (sebagian kolom dihapus agar contoh isi tabel dapat ditampilkan di sini):

patient_id	sex	age	country	province	city	infection_case	infected_by	symptom_onset	confirmed_date	released_date	ceased_date	state
1000000001	male	50s	Korea	Seoul	Gangseo-jo	overseas inflow		1/22/2020	1/23/2020	2/5/2020		released
6001000285	male	60s	Korea	Gyeongsan-si	Gyeongsan-si				3/3/2020		3/4/2020	deceased
6001000286	female	80s	Korea	Gyeongsan-si	Gyeongsan-si				3/3/2020	3/26/2020		released
1100000019	female	30s	Korea	Busan	Seo-gu	Onchun Church	1100000016		2/23/2020			released
1100000020	female	50s	Korea	Busan	Seo-gu	contact with pat	1100000013	2/20/2020	2/23/2020			released

Time.csv:

date	time	test	negative	confirmed	released	deceased
2/21/2020	16	16400	13016	204	17	2
2/22/2020	16	21586	15116	433	18	2
2/23/2020	16	26179	17520	602	18	6
2/24/2020	16	32756	20292	833	24	8

TimeProvince.csv:

date	time	province	confirmed	released	deceased
2/21/2020	16	Jeju-do	1	0	0
2/22/2020	16	Seoul	30	6	0
2/22/2020	16	Busan	11	0	0
2/22/2020	16	Daegu	193	0	0

Bab 7 Prediksi Kualitas Tidur dari Data Wearable Device

Oleh:

Chandra Wijaya dan Raymond Chandra Putra

7.1. Pendahuluan

Siapa yang tidak ingin selalu sehat dan merasa bugar? Secara umum, semua orang ingin selalu sehat agar tetap dapat beraktivitas normal dan tidak berurusan dengan dokter atau rumah sakit. Beruntungnya pada jam *now*, sudah tersedia berbagai aplikasi di ponsel untuk membantu agar orang selalu sehat. Misalnya aplikasi untuk melacak makanan yang kita konsumsi dan olahraga yg kita lakukan. Berdasar hasil lacakan tersebut, aplikasi lalu memberikan rekomendasi makanan untuk kita¹⁸. Pada aplikasi itu, rekomendasi diberikan berdasar hasil analisis data yang dikumpulkan aplikasi. Di balik pemberian rekomendasi itu, ada teknologi-teknologi yang dimanfaatkan untuk pengumpulan data. Juga ada proses penyiapan data dan analisis data yang memanfaatkan teknik-teknik atau algoritma-algoritma yang kompleks. Intinya, tahapan Data Science (lihat Bab 1) diterapkan pada kasus ini sampai hasilnya, yang berupa rekomendasi, dapat diberikan melalui aplikasi.

Selain makanan dan olah-raga, hal penting lain yang membuat kita sehat adalah tidur nyenyak (berkualitas baik) dalam waktu yang cukup. Kualitas tidur kita dapat diprediksi berdasarkan aktivitas kita sehari-hari. Dengan mengetahui kualitas tidur, jika ternyata jelek, kita lalu dapat melakukan hal-hal yang memang perlu dilakukan untuk memperbaiki tidur kita.

Bab ini membahas *wearable device*, termasuk *smartwatch*, konsep klasifikasi data, Jaringan Syaraf Tiruan (JST), dan hasil penelitian tentang bagaimana memproses data dari smartwatch agar dapat dihasilkan model berbasis JST untuk memprediksi kualitas tidur pemakai smartwatch. JST merupakan dasar dari sistem *deep learning* yang saat ini banyak dimanfaatkan untuk menganalisis data. Dari paparan pada bab ini, diharapkan para pembaca mendapatkan pengetahuan awal (yang tidak kompleks) tentang bagaimana analisis data dari smartwatch dilakukan dengan memanfaatkan JST, hingga menghasilkan model prediksi.

¹⁸ <https://www.androidauthority.com/best-health-apps-for-android-668268/> (diakses 20 Agustus 20)

7.2. Wearable Device

Wearable devices, yang merupakan salah satu perangkat *Internet of Things* (IoT), adalah salah satu alat yang dipasang pada satu bagian tubuh tertentu. Alat tersebut bekerja untuk mendeteksi aktivitas atau kejadian pada bagian tubuh tertentu. Salah satu jenis dari alat tersebut dapat dipasang di pergelangan tangan dan memiliki kemampuan untuk mengukur detak jantung seseorang untuk kepentingan tertentu, misalnya untuk mendeteksi adanya gangguan kesehatan. Detak jantung yang beraturan menandakan bahwa jantung berfungsi dengan baik atau tidak mengalami gangguan. Namun jika detak jantung tidak beraturan, kadang lambat dan beberapa saat kemudian cepat, berarti jantung mengalami gangguan. (Orang tersebut lalu perlu memeriksakan kesehatan ke rumah sakit atau dokter).

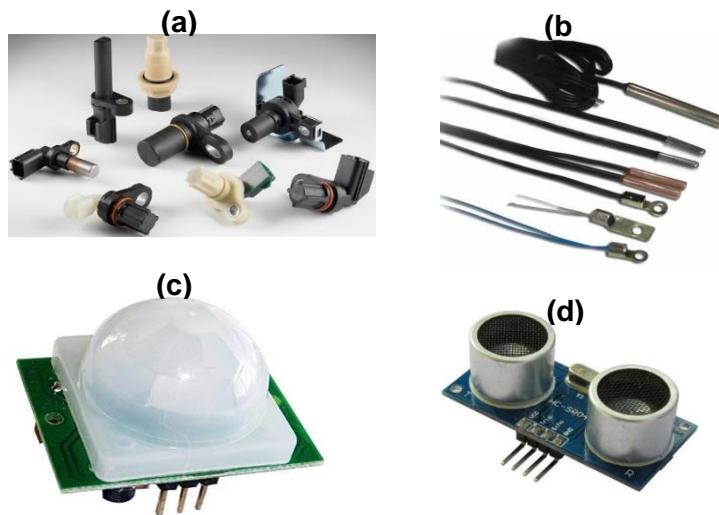
Wearable device memiliki beberapa sensor dan microcontroller yang bekerja untuk tujuan tertentu. Sensor pada wearable device bekerja untuk mengukur nilai tertentu (misalnya detak jantung, suhu tubuh, dll). Sensor dapat dikelompokkan berdasarkan karakteristik dan tipenya.

Berikut ini adalah pengelompokan sensor berdasarkan karakteristiknya:

- Aktif vs pasif: Sensor aktif bekerja dengan catu daya yang ditambahkan pada sistem sensor, sedangkan sensor pasif bekerja dengan catu daya dari energi sinyal yang dideteksi.
- Digital vs analog: Sensor digital menghasilkan sinyal bernilai biner (nyala/true atau mati/false), sedangkan sensor analog menghasilkan nilai numerik/kontinyu atau bilangan bulat.

Berikut ini adalah klasifikasi sensor berdasarkan tipenya (watelectronics, 2020):

- Sensor kecepatan: Sensor yang digunakan untuk mendeteksi kecepatan dari sebuah objek ataupun kendaraan. Beberapa contoh sensor dengan tipe ini adalah wheel speed sensor, speedometer, Light Detection and Ranging (LIDAR), ground speed radar, pitometer, doppler radar, dll. Contoh dari sensor kecepatan dapat dilihat pada Gambar 7.1(a).
- Sensor suhu: Sensor yang mendapatkan nilai suhu dalam bentuk sinyal elektrik. Sensor ini dikelompokkan menjadi sensor berkontak dan tidak-berkontak (dengan objeknya). Pada Gambar 7.1(b) ditunjukkan beberapa contoh sensor suhu berkontak, dimana dalam pemanfaatannya sensor harus berkontak langsung dengan objek untuk mengukur suhunya.
- Passive Infra Red (PIR) Sensor: Sensor PIR adalah sensor yang digunakan untuk mengukur pancaran radiasi cahaya infra merah dari sebuah objek. Setiap objek yang memiliki suhu diatas 0 akan mengirimkan energi panas dalam bentuk radiasi gelombang infra merah. Gelombang ini tidak dapat dilihat oleh mata manusia, namun dapat ditangkap oleh sensor seperti sensor PIR motion detector. Contoh dari sensor PIR dapat dilihat pada Gambar 7.1(c).
- Sensor ultrasonik: Cara kerja sensor ultrasonik sama dengan sonar atau radar, dimana sensor memancarkan gelombang suara frekuensi tinggi ke arah objek, lalu menginterpretasikan gelombang pantulan dari suatu objek. Contoh sensor ultrasonik dapat dilihat pada Gambar 7.1(d).



Gambar 7.1. (a) Contoh sensor kecepatan¹⁹, (b) contoh sensor suhu²⁰, (c) sensor infra red pasif²¹, (d) sensor ultrasonik²².

Wearable device umumnya hanya beroperasi untuk mendapatkan nilai dari sensor. Berbagai data yang didapatkan oleh *wearable device* akan dikirimkan ke smartphone untuk diproses lebih lanjut. Ini dilakukan karena ukuran media penyimpanan di *wearable device* relatif kecil, selain itu, prosesor pada alat ini juga tidak memiliki kecepatan proses yang tinggi agar tidak membutuhkan daya listrik banyak.

Komunikasi antara *wearable device* dengan smartphone umumnya dilakukan dengan Bluetooth. Bluetooth adalah sebuah standar komunikasi nirkabel dengan jarak jangkauan layanan terbatas, maksimal sekitar 10 meter. Konektifitas antara kedua alat tersebut sangat bergantung dengan ketersediaan bluetooth. Apabila bluetooth tidak aktif, maka telepon genggam tidak dapat menerima data yang dikirimkan oleh *wearable devices*. Namun umumnya *wearable devices* memiliki kemampuan untuk menyimpan data pada tempat penyimpanan internal, sehingga setelah hubungan antara *wearable devices* dengan telepon genggam tersedia, seluruh data pada *wearable devices* akan dikirimkan ke telepon genggam dan dapat diproses lebih lanjut.

¹⁹ <https://www.watelectronics.com/different-types-of-sensors-with-applications/>

²⁰ <https://www.watelectronics.com/different-types-of-sensors-with-applications/>

²¹ <https://www.elprocus.com/passive-infrared-pir-sensor-with-applications/>

²² <https://www.watelectronics.com/different-types-of-sensors-with-applications/>

7.3. Konsep Dasar

Pada bagian ini dibahas pengantar konsep-konsep yang dimanfaatkan pada kasus ini. Konsep dibahas dengan disederhanakan agar dapat diikuti pembaca awam.

7.3.1. Klasifikasi Data

Sebagai teknik analisis data, teknik klasifikasi data dikategorikan ke dalam teknik prediksi. Dengan meproses data dengan format tertentu, teknik ini akan menghasilkan model yang dapat digunakan untuk memprediksi nilai kategorial atau diskret. Data masukan yang dibutuhkan untuk membuat model diilustrasikan pada Gambar 7.2. Pada data yang berformat tabular tersebut, terdapat kolom-kolom (atribut-atribut) prediktor dan kolom/atribut kelas. Jika model dilatih dengan data tersebut, nantinya model akan dapat digunakan untuk memprediksi jenis binatang jika kita memiliki nilai-nilai dari atribut prediktor, yaitu jumlah kaki, punya sayap atau tidak, tinggi tubuh, jenis makanan dari binatang. Adapun hasil prediksi jenis binatang yang kita dapatkan, akan bernilai salah satu dari yang tercantum pada kolom Jenis, yaitu burung kutilang, kucing, sapi, dll. Di sini perlu disampaikan bahwa data yang digunakan untuk membuat model klasifikasi dapat saja memiliki semua atribut prediktor bertipe numerik, misalnya jumlah kaki, berat, tinggi, umur, dll.

Jumlah Kaki	Punya Sayap?	Tinggi (Cm)	Makanan	Jenis
2	Y	12.3	buah	burung kutilang
4	T	23	campuran ikan	kucing
4	T	134	campuran tumbuhan	sapi
2	Y	3	biji-bijian	burung gereja
0	T	2	binatang kecil	ular
4	T	0.6	serangga	cicak

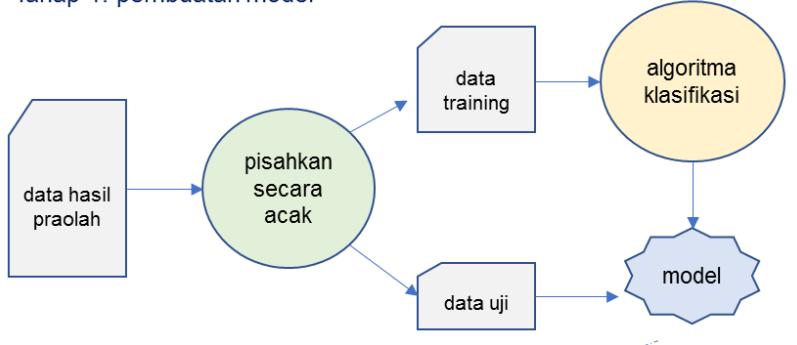
Gambar 7.2. Ilustrasi data untuk pembuatan model klasifikasi.

Dalam praktek pembuatan model klasifikasi yang sebenarnya, seringkali data yang siap diumpulkan ke algoritma klasifikasi belum tersedia. Dari hasil kegiatan pengumpulan data (lihat Bab 1), dihasilkan data mentah yang harus disiapkan/dipraolah terlebih dahulu sedemikian rupa agar diterima oleh algoritma klasifikasi. Untuk membangun model akan dibutuhkan data hasil praolah perlu berukuran relatif besar dengan jumlah rekord/baris yang banyak, misalnya lebih dari 1000, dan memenuhi kriteria tertentu (misalnya tiap nilai kelas direpresentasikan oleh jumlah baris yang seimbang, tidak mengandung nilai atribut yang salah, dll.). Hal tersebut dimaksudkan agar dapat dihasilkan model dengan tingkat akurasi yang baik.

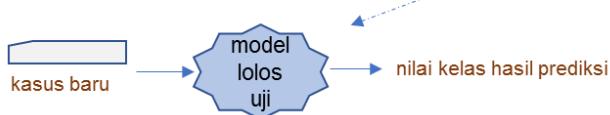
Secara umum, klasifikasi data terdiri dari dua tahap yang utama, yaitu (lihat Gambar 7.3):

- Pertama, pemisahan data masukan (hasil praolah data) secara acak menjadi data training (misalnya 80% dari keseluruhan data) dan uji (misalnya 20%). Kemudian data training diumpulkan ke algoritma klasifikasi untuk mendapatkan keluaran berupa model. Terdapat berbagai algoritma klasifikasi yang sudah dikembangkan para peneliti. Seorang data scientist perlu memilih yang paling tepat berdasar data yang diolah, kinerja dan pertimbangan lain yang perlu. Setelah dilatih, model yang dihasilkan oleh algoritma klasifikasi belum tentu berkualitas baik dapat dapat dimanfaatkan, karena itu harus diuji dulu dengan data uji. Salah satu cara untuk mengevaluasi model adalah dengan menghitung akurasi dari model berdasar masukan data uji. Akurasi dihitung dari jumlah baris/rekord yang diprediksi benar dibagi dengan total jumlah rekord. Jika model lolos uji, maka dapat dimanfaatkan di tahap kedua.
- Kedua, penggunaan model untuk mengklasifikasi data baru. Di sini, sebuah rekord yang belum diketahui kelasnya “diumpulkan” ke model, yang lalu akan memberikan jawaban “kelas” hasil perhitungannya. Dalam konteks klasifikasi kemampuan ekonomi orang, misalnya rekord itu memiliki nilai kolom/variabel jumlah penghasilan, kondisi tempat tinggal, jumlah tanggungan, lingkungan tempat tinggal, dll. Hasil prediksi, misalnya miskin, penghasilan menengah atau kaya.

Tahap-1: pembuatan model



Tahap-2: pemanfaatan model



Gambar 7.3. Proses klasifikasi data.

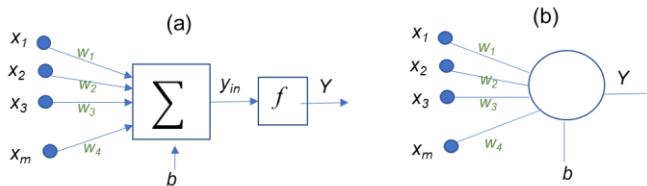
Terdapat berbagai algoritma klasifikasi, namun sebagai pengenalan, di sini hanya akan dibahas Jaringan syaraf tiruan (JST) yang dimanfaatkan pada penelitian (A Sathyaranayana, 2016). Pada penelitian itu, model klasifikasi yang berbasis JST dibandingkan dengan model *Logistic Regression* dan didapatkan hasil bahwa model yang berbasis JST berkinerja lebih baik.

7.3.2. Jaringan Syaraf Tiruan dan Multilayer Perceptrons

Jaringan syaraf tiruan (JST) merupakan salah satu dari *tools* dan pendekatan yang digunakan pada algoritma-algoritma Machine Learning. JST banyak dimanfaatkan pada kehidupan sehari-hari, misalnya untuk mengenali bentuk-bentuk gambar/citra, mengenali kata-kata (hasil tulisan tangan), penyortiran email spam, diagnosis penyakit²³, dll.

JST merupakan sistem yang dapat “belajar” (dari data) melalui serangkaian komputasi. JST menggunakan jaringan fungsi-fungsi untuk memahami dan menterjemahkan data masukan dalam format tertentu menjadi keluaran yang diinginkan, yang biasanya dalam bentuk berbeda (dibanding data masukan). Konsep JST ini diinspirasi oleh otak manusia dan cara kerja jaringan yang menghubungan berjuta-juta neuron pada otak (Han, J., Pei, J., & Kamber, M., 2012). Pada jaringan itu, neuron-neuron bekerja bersama-sama dalam rangka memahami masukan-masukan dari indera manusia.

Pada JST, sebuah neuron dimodelkan sebagai model matematika dan dinamakan perceptron yang ditunjukkan pada Gambar 7.4.



Gambar 7.4. Model sebuah perceptron: (a) versi detil, (b) versi yang disederhanakan.

Pada Gambar 7.4, y_{in} dan Y direpresentasikan dengan rumus-rumus di bawah ini.

Untuk y_{in} ,

$$y_{in} = x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 \dots x_m \cdot w_m + b \quad (1)$$

dimana w_i menyatakan bobot (untuk variabel masukan, x_i) dan b adalah bias. Secara umum, y_{in} dapat dituliskan sebagai:

$$y_{in} = \sum_{i=0}^m x_i \cdot w_i + b \quad (2)$$

Sedangkan keluaran dari perceptron, Y :

$$Y = f(y_{in}) \quad (3)$$

dimana f biasa dinamakan fungsi aktivasi. Fungsi aktivasi tersebut bermacam-macam, diantaranya adalah fungsi linear, hyperbolic tangent, fungsi logistik dan rectified linear activation. Sebagai contoh di bawah ini diberikan persamaan dari fungsi logistik

$$f(x) = \frac{L}{1+e^{-k(x-x_0)}} \quad (4)$$

dimana

²³ <https://deeppai.org/machine-learning-glossary-and-terms/neural-network>

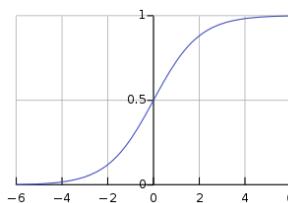
x = variabel dengan domain bilangan riil dengan rentang minimum tak terhingga sampai positif tak terhingga

L = nilai kurva ($f(x)$) maksimum

x_0 = nilai x yang memberikan titik tengah kurva

k = *steepness* pada kurva.

Jika $L = 1$, $k = 1$ dan $x_0 = 0$, maka fungsi tersebut dinamakan fungsi sigmoid logistik standard dan kurvanya diberikan pada Gambar 7.5, dimana sumbu horizontal menyatakan nilai x , sedangkan sumbu vertikal adalah nilai $f(x)$.

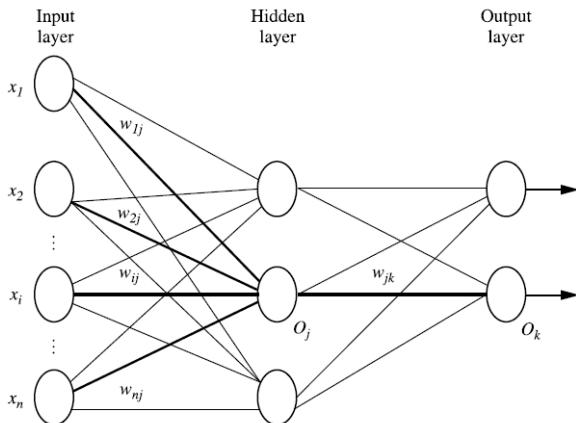


Gambar 7.5. Kurva fungsi sigmoid logistik standar.

Dari persamaan 1, 2 dan 3, dapat diterjemahkan bahwa sebuah perceptron akan menghasilkan sebuah nilai keluaran yang merupakan fungsi dari hasil penjumlahan semua variabel masukan ($x_1 \dots x_m$) dikalikan dengan bobot-bobot tiap variabel ($w_1 \dots w_m$) ditambah dengan bias (b). Jika fungsi aktivasi yang digunakan adalah fungsi sigmoid logistik standar (Gambar 7.5), maka Y akan bernilai 0 sampai dengan 1.

Sebagaimana otak manusia yang terdiri dari berjuta-juta neuron yang saling tersambung dan membentuk jaringan, untuk keperluan analisis data, JST juga umumnya dirancang dengan menggunakan banyak perceptron yang tersambung dan membentuk jaringan. Pada jaringan itu, keluaran dari sebuah perceptron dapat menjadi masukan bagi perceptron di belakangnya. Dalam hal perceptron menerima masukan yang berupa keluaran dari perceptron lain, maka persamaan 3 tetap berlaku, hanya saja nilai y_{in} diperoleh dari nilai-nilai Y pada perceptron di depannya.

Salah satu contoh JST adalah *Multilayer Perceptrons* yang disingkat menjadi MLP. MLP termasuk teknik pada Machine Learning yang tergolong ke dalam kelompok *deep learning* yang sederhana. Contoh MLP diberikan pada Gambar 7.6 (untuk penyederhanaan, b tidak digambarkan pada jaringan tersebut).



Gambar 7.6. Model jaringan syaraf tiruan (Han, J., Pei, J., & Kamber, M., 2012).

Sebagaimana ditunjukkan pada Gambar 7.6, MLP memiliki tiga komponen utama yaitu *input layer* (lapis masukan), *hidden layer* (lapis tersembunyi), dan *output layer* (lapis keluaran) dengan penjelasan sebagai berikut:

- *Input layer* merupakan layer yang menerima data berformat vektor dengan jumlah elemen sesuai dengan jumlah atribut prediktor yang akan diproses. Pada model klasifikasi, tiap elemen vektor (x_i) disambungkan ke tiap atribut prediktor. Jika misalnya terdapat 4 atribut prediktor, akan terdapat 4 elemen vektor pada lapis masukan.
- *Hidden layer* dapat terdiri dari satu atau lebih lapis. Tiap lapis berisi sejumlah perceptron. Jika lapis tersembunyi hanya terdiri dari satu lapis (Gambar 7.6), masukan tiap perceptron tersambung ke elemen vektor (x_i) pada lapis masukan, sedangkan luaran tiap perceptron tersambung ke lapis luaran. Tiap hubungan dari perceptron ke lapis masukan maupun lapis luaran memiliki bobot (w_{ij} atau w_{jk}) tersendiri. Jumlah lapis tersembunyi dan jumlah perceptron pada tiap lapis yang tepat biasanya didapatkan melalui serangkaian eksperimen (dari beberapa konfigurasi yang diuji-coba, dapat dipilih MLP yang memberikan akurasi terbaik dengan komputasi yang cepat). Pada tahap pelatihan, bobot-bobot pada semua perceptron akan dihitung dari data pelatihan sedemikian rupa sehingga pada akhir pelatihan dihasilkan MLP dengan nilai bobot-bobot tertentu.
- *Output layer* terdiri dari satu atau lebih perceptron. Penentuan jumlah perceptron ini biasanya juga didasarkan pada eksperimen (dari beberapa konfigurasi yang diuji-coba, dapat dipilih MLP dengan kinerja terbaik). Jika nilai yang akan diprediksi terdiri dari dua nilai (0 atau 1), pada lapis keluaran dapat digunakan satu perceptron yang menghasilkan luaran dengan nilai yang mendekati 0 atau 1.

Pada MLP itu, nilai keluaran tiap perceptron pada output layer (O_k) dihitung dari keluaran perceptron-perceptron pada hidden layer. Tiap perceptron pada hidden layer sendiri, memproses masukan dari data berformat vektor (x_1, \dots, x_n) untuk menghasilkan nilai keluarannya.

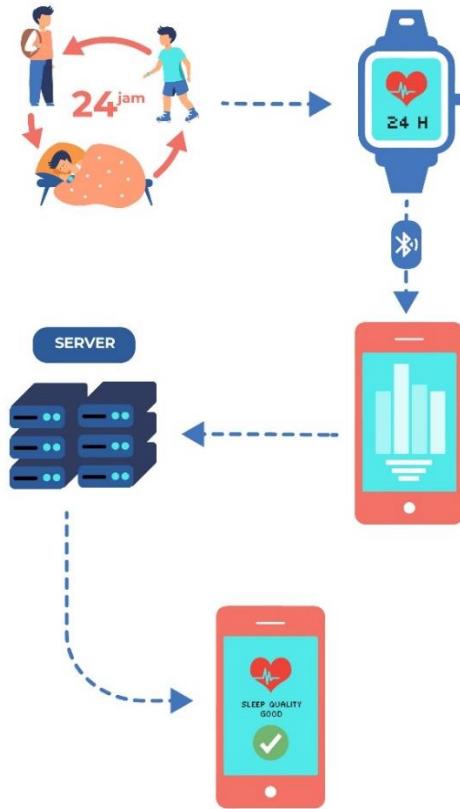
Jika MLP digunakan untuk klasifikasi data, sebagaimana ditunjukkan pada Gambar 7.3, MLP perlu dilatih dengan menggunakan data training. Pada MLP, tujuan dari pelatihan adalah untuk menghitung bobot-bobot pada hidden layer maupun output layer. Proses pelatihan tersebut diilustrasikan pada Subbab 7.4.3. Setelah model terbentuk, model juga perlu diuji tingkat akurasinya dengan data uji.

7.4. Klasifikasi Data Wearable Device

Setelah pada bagian sebelumnya dibahas mengenai wearable device, teknik klasifikasi beserta MLP, pada subbab ini akan dibahas pembuatan model klasifikasi dengan memanfaatkan MLP. Model akan dapat digunakan untuk memprediksi kualitas tidur seseorang bedasarkan data yang direkam oleh salah satu jenis wearable device, yaitu smartwatch.

Salah satu opsi tahapan utama pada sistem yang memanfaatkan teknik klasifikasi untuk memprediksi kualitas tidur seseorang dibiberikan pada Gambar 7.7, dengan keterangan sebagai berikut:

- a. Detak jantung para pengguna smartwatch ketika melakukan berbagai kegiatan (jalan, belajar, bekerja, olah-raga, dll.) dideteksi oleh sensor pada smartwatch, lalu hasil deteksi tersebut direkam secara periodik oleh smartwatch selama 24 jam.
- b. Ketika smartwatch sudah mendeteksi adanya koneksi dengan smartphone (umumnya via bluetooth), maka data detak jantung tersebut dikirim ke smartphone (pemilik smartwatch).
- c. Melalui Internet, aplikasi pada smartphone lalu mengirimkan data detak jantung ke server di cloud (awan). Pengumpulan data dari banyak (mencapai jutaan) smartphone, dimana tiap smartphone bisa sering mengirim data ke server di cloud, umumnya dilakukan dengan memanfaatkan teknologi big data (lihat Bab 10 yang membahas big data dan teknologinya).
- d. Program di server di cloud menggunakan data dari banyak orang (yang berukuran sangat besar) untuk membangun model klasifikasi. (Keterangan: Karena data yang diproses berukuran besar dan bertambah dengan cepat, umumnya program dirancang dan diimplementasi dalam lingkungan sistem big data.) Jika model sudah diuji dan terbukti akurat untuk memprediksi kualitas tidur orang, maka setidaknya terdapat dua cara untuk memanfaatkan model tersebut, yaitu:
 - Ketika program di server menerima data detak jantung seseorang (misalnya, data selama kurun waktu 24 jam), maka program dapat melakukan prediksi (apakah tidur orang tersebut berkualitas atau tidak) dan mengirim hasilnya ke smartphone yang tadinya mengirim data itu.
 - Model klasifikasi prediksi (yang berukuran relatif kecil) dikirim oleh server ke smartphone, dengan demikian jika smartphone menerima data kegiatan pemilik smartwatch selama periode tertentu, maka aplikasi smartphone dapat memberikan prediksi kualitas tidur pemilik smartphone.



Gambar 7.7. Tahapan utama pada sistem yang memanfaatkan data aktivitas untuk memprediksi kualitas tidur seseorang.

Berikut ini diberikan ilustrasi lebih detil pada tahap pengumpulan data, penyiapan data, pelatihan model klasifikasi dan pemanfaatan model untuk melakukan prediksi kualitas tidur.

7.4.1 Pengumpulan Data

Salah satu contoh *wearable device* berupa smartwatch yang dijual di pasaran dapat dilihat pada Gambar 7.8. Beberapa data yang direkam dan disediakan oleh alat ini adalah detak jantung, jarak dan langkah yang telah ditempuh, tekanan darah dan kadar oksigen dalam darah.

Smartwatch itu memiliki beberapa sensor yang terintegrasi dengan sebuah microprosesor. Data hasil deteksi sensor disimpan secara local di media penyimpanan pada smartwatch.



Gambar 7.8. Contoh wearable device berbentuk smartwatch²⁴.

Sensor akselerometer pada smartwatch digunakan untuk mendeteksi pergerakan tangan pengguna. Sensor itu menangkap 3 buah nilai, yang merepresentasikan pergerakan horizontal (sumbu x), vertikal (sumbu y) dan orthogonal (sumbu z). Sensor tersebut biasanya sangat presisi, sehingga pergerakan tangan sedikit saja akan terdeteksi.

Selain tiga nilai tersebut, smartwatch juga dapat merekam data lainnya. Gambar 7.9 memaparkan salah satu contoh data yang ditangkap oleh smartwatch. Pada tabel di gambar tersebut, Epoch adalah urutan penangkapan data, SpO2 adalah kadar oksigen dalam darah, HR adalah detak jantung, BPOS adalah tangan yang mengenakan smartwatch (kiri/L atau kanan/R). Kemudian tiga kolom berikutnya adalah nilai yang ditangkap oleh sensor accelerometer untuk sumbu x, y dan z. (Keterangan: Pada tabel terdapat SpO2 yang bernilai 0. Hal ini mengindikasikan adanya hasil yang tidak benar. Untuk itu, sebagaimana dibahas pada Bab 1, nantinya pada tahap penyiapan data perlu dilakukan pembersihan data terhadap data mentah tersebut.)

Epoch	SpO2	HR	BPOS	Acc-x	Acc-y	Acc-z
1	0	67	R	0,010105227	0,130185094	0,08990695
2	97	67	R	0,010829214	0,139301211	0,07415258
3	97	67	R	0,011063745	0,148386737	0,06133495
4	97	67	R	0,01182852	0,15704399	0,05478848
5	97	67	R	0,012909402	0,164763118	0,05194352
6	97	67	R	0,013592601	0,170881317	0,04904757
7	97	67	R	0,013837329	0,18415781	0,04749762
8	95	67	R	0,013490631	0,182832201	0,04557039
9	97	67	R	0,013021569	0,172839142	0,04390828
10	96	67	R	0,012328173	0,160521166	0,04534606
11	0	67	R	0,012134443	0,14419577	0,04771176
12	96	67	R	0,012603492	0,113696545	0,05360563
13	97	67	R	0,013205115	0,122231434	0,05017944
14	96	67	R	0,013409055	0,126983236	0,04542763

Gambar 7.9. Contoh data yang ditangkap dan direkam smartwatch.

²⁴ <https://www.lazada.co.id/products/jam-kesehatan-pengukur-detak-jantung-smart-watch-m3-i1064644585-s1649638353.html>

Apabila smartwatch telah dihubungkan dengan smartphone (telepon genggam) via Bluetooth, maka seluruh data tersebut akan dikirimkan ke telepon genggam.

7.4.2 Penyiapan Data

Pada bagian ini akan diterangkan bagaimana data yang sudah dikumpulkan dari *wearable device* disiapkan agar dapat diumpulkan ke MLP.

Data yang diambil dari sensor gerak (Gambar 7.9), masih berupa data mentah yang belum dapat digunakan untuk melatih MLP. Dari data mentah tersebut, harus disiapkan dahulu data masukan untuk kedua teknik tersebut. Untuk keperluan ini, perlu didefinisikan terlebih dahulu tentang kualitas tidur dan variabel-variabel yang dapat digunakan untuk menentukan apakah tidur seseorang berkualitas atau tidak. Setelah itu, disiapkan data pelatihan yang mengandung nilai dari variabel-variabel tersebut termasuk dengan labelnya (tidur berkualitas/tidak).

Pada bagian ini dibahas definisi kualitas tidur, representasi data dan data hasil penyiapan.

Definisi Kualitas Tidur

Kualitas tidur dapat ditentukan berdasarkan efisiensi tidur yang dapat dihitung dengan membandingkan waktu tidur seseorang dengan lamanya seseorang berada pada kasur (A Sathyanarayana, 2016). Efisiensi tidur ini dapat dituliskan dengan persamaan berikut:

$$\begin{aligned} \text{Efisiensi Tidur} &= \frac{\text{Total waktu tidur}}{\text{Total waktu di kasur}} \\ &= \frac{\| \text{Periode tidur} \| - \text{WASO}}{\| \text{Periode tidur} \| + \text{Latensi}} \end{aligned}$$

Pada rumus di atas, total waktu di kasur (tempat tidur) dihitung berdasar periode tidur ditambah latensi. Ada kalanya, seseorang sudah berbaring di kasur, namun belum masuk ke dalam tahap tidur. Jeda waktu itulah yang disebut dengan latensi.

Ketika kita tidur, terkadang kita dapat terbangun, baik disengaja/disadari maupun tidak. Hal ini akan mempengaruhi total waktu tidur. Oleh karena itu total waktu tidur yang sebenarnya dihitung dari periode tidur dikurangi dengan durasi terbangun, yang disebut dengan *Wake After Sleep Onset* (WASO). Karena WASO ini dihitung dari pergerakan sensor akselerometer, cukup sulit membedakan pergerakan kecil sesaat dengan memang betul-betul bangun. Oleh karena itu, kita hanya mengambil periode bangun yang melebihi 5 menit saja. Dari penjelasan tersebut, WASO dapat dituliskan dengan persamaan berikut.

$$\text{WASO} = \sum \begin{cases} \| \text{Periode Bangun} \|, & \| \text{Periode Bangun} \| > 5 \text{ menit} \\ 0, & \text{lainnya} \end{cases}$$

Dengan menghitung efisiensi dengan rumus di atas kita dapat menentukan apakah tidur seseorang sudah berkualitas atau belum. Tidur seseorang dikatakan berkualitas jika efisiensi tidurnya mencapai minimal 85% (DL Reed, 2016). Dengan kata lain, jika efisiensi tidur kurang dari 85% maka kualitas tidurnya buruk.

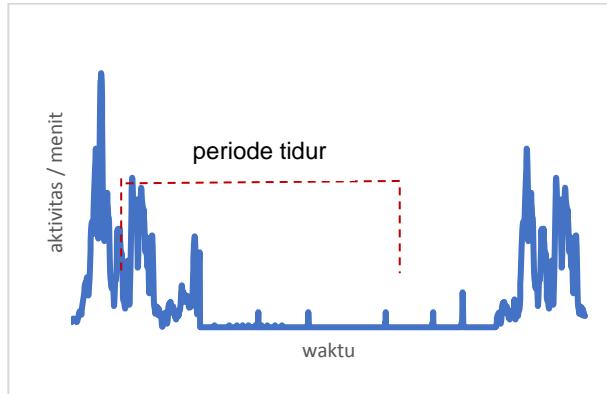
Data Mentah

Data mentah dari akselerometer untuk sebuah sumbu, misalnya x, berupa data terturut berdasar waktu. Sebagai contoh pada Tabel 7.1 diperlihatkan data sebuah sumbu selama 0,25 detik dengan interval 0,02 detik (interval ini dapat diatur sesuai kebutuhan dan akurasi yang diinginkan). Dengan interval pengambilan data itu, untuk periode 1 hari, data yang ditangkap sudah melebihi 1 juta baris atau rekord. (Padahal untuk membangun model prediksi kualitas tidur, tidak cukup hanya menggunakan data selama 1 hari saja.)

Tabel 7.1. Contoh data akselerometer pada 1 sumbu selama 0,25 detik.

Timestamp	Nilai
0	-0.0014
0.02	-0.011
0.04	-0.0103
0.06	-0.009
0.08	-0.0097
0.1	-0.0122
0.12	-0.0145
0.14	-0.0131
0.16	-0.0112
0.18	-0.0087
0.2	-0.0087
0.22	-0.0134
0.24	-0.0179

Karena jumlah baris pada data yang terkumpul sangat banyak (berjuta-juta), data perlu diubah (ditransformasi) untuk mengecilkan ukuran baris. Sebagai contoh, kita dapat mengambil nilai rata-rata per 1 menit. Hasil visualisasi dari contoh hasil rata-rata tersebut dapat dilihat pada Gambar 7.10.



Gambar 7.10. Contoh grafik aktivitas seseorang dalam satu hari.

Dengan menginterpretasikan grafik pada Gambar 7.8, kita dengan mudah dapat mengetahui kapan seseorang itu tidur. Secara umum, ketika seseorang sedang tidur maka aktivitas yang dilakukan sangat minim tetapi tidak 0 (akselerometer mempunyai kemampuan mendeteksi aktivitas pergerakan mikro atau sangat kecil ketika *wearable device* digunakan). Waktu dimana data akselerometer menunjukkan angka rendah (terletak di tengah grafik) dapat ditandai sebagai periode tidur. Aktivitas yang menaik sedikit di tengah periode tidur, kemungkinan besar mengindikasikan saat seseorang terbangun di tengah tidur.

Agar data dapat diumpulkan ke algoritma Logistic Regression maupun MLP untuk melatih model, kita perlu menyiapkan data berisi rekord-rekord (baris-baris) dimana dalam satu rekord berisi nilai-nilai variabel yang berpengaruh terhadap kualitas tidur dan nilai kualitas tidur (bagus atau tidak). Komputasi untuk menyiapkan nilai-nilai fitur ini cukup kompleks dan menggunakan algoritma yang cukup rumit. Karena itu, di sini hanya diberikan contoh hasil akhir tahapan ini.

Contoh Hasil Penyiapan Data

Himpunan data yang diolah atau disiapkan dari data mentah (Gambar 7.7) yang sekarang dapat diumpulkan ke MLP ditunjukkan pada Tabel 7.2.

Keterangan tiap kolom pada tabel tersebut diberikan di bawah ini:

- Vektor Bangun: berisi sekumpulan data kontinyu dan terurut menurut waktu (rata-rata per satuan waktu dari nilai pada sumbu, x, y dan z) pada saat seseorang tidak tidur
- Max: Nilai maksimum pada Vektor Bangun
- Min: Nilai minimum pada Vektor Bangun
- Rata-Rata: Nilai rata-rata pada Vektor Bangun
- Vektor Tidur: berisi sekumpulan data kontinyu dan terurut menurut waktu (rata-rata per satuan waktu dari nilai pada sumbu, x, y dan z) pada saat seseorang dinyatakan tidur

- Banyak Gerakan: berisi nilai yang merepresentasikan banyak gerakan yang dihitung dengan rumus/algoritma tertentu dari Vektor Bangun dan Vektor Tidur dan dinormalisasi
- Efisiensi Tidur: berisi nilai yang dihitung dari Vektor Bangun dan Vektor Tidur dengan rumus tertentu
- Kategori/Kelas: berisi nilai yang menyatakan kualitas tidur seseorang dan ditentukan berdasarkan nilai efisiensi tidur.

Tabel 7.2. Contoh hasil penyiapkan data.

Hari	Vektor Bangun	Max	Min	Rata-Rata	Vektor Tidur	Banyak Gerakan	Efisiensi Tidur	Kategori/Kelas
1	[1.6 ,1.44, ...]	2.5	0.01	1.52	[0.1,0.02, ...]	0.8	88	Berkualitas
2	[2.1,2.3, ...]	2.1	0.011	2.2	[0.12, 0.03, ...]	0.86	93	Berkualitas
3	[0.95, 0.93, ...]	1.7	0.014	0.8	[0.6, 0.8, ...]	0.78	51	Tidak Berkualitas
4	[2.5, 2.6, ...]	3.2	0.015	2.5	[0.09 ,0.1, ...]	0.34	90	Berkualitas
5	[1.08, 1.23, ...]	3.1	0.016	1.1	[0.9 ,1.1, ...]	0.45	75	Tidak Berkualitas
dst

7.4.3 Pelatihan MLP

Setelah hasil penyiapan data didapatkan, biasanya masih perlu dilakukan pemilihan data lagi. Sebagai ilustrasi sederhana, pada contoh MLP di sini, dipilih kolom Max, Min, Rata-rata dan Banyak Gerakan sebagai kolom prediktor (sebagai informasi, kolom prediktor untuk MLP dapat mencapai ratusan bahkan ribuan). Sedangkan Kategori/Kelas dijadikan kolom kelas. Ilustrasi pelatihan MLP dan model hasilnya dibahas di bawah ini.

Untuk membuat model klasifikasi dengan MLP, mula-mula perlu dirancang strukturnya terlebih dahulu, yang melingkup: jumlah elemen/node pada lapis masukan (input layer), jumlah lapis tersembunyi (hidden layer) dan tiap lapis memiliki berapa perceptron/neuron, fungsi aktivasi pada tiap neuron dan berapa jumlah elemen/node pada lapis luaran (output layer).

Jumlah elemen pada input layer disesuaikan dengan jumlah kolom prediktor pada data training. Jumlah elemen pada output layer disesuaikan dengan nilai kelas yang akan diprediksi.

Dalam kasus ini, karena data training memiliki 4 atribut prediktor, pada MLP dirancang 4 elemen pada input layer. Pada output layer dirancang memiliki satu elemen karena hanya digunakan untuk memprediksi dua nilai kelas, yaitu "berkualitas" dan "tidak berkualitas" (lihat Gambar 7.11). Jumlah hidden layer, perceptron dan fungsi aktivasi biasanya dicari yang paling optimal melalui serangkaian eksperimen. Konfigurasi MLP ini akan mempengaruhi lamanya waktu proses pelatihan (sampai komputasi konvergen dan model terbentuk) dan tingkat akurasi model. Pada contoh desain yang dipresentasikan pada Gambar 7.11, hidden layer terdiri dari 1 lapis dengan 15 perceptron.

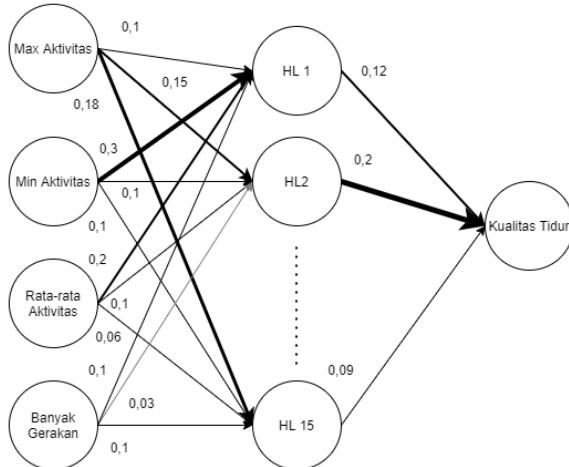
Setelah konfigurasi MLP dirancang, tiap bobot (pada Gambar 6 adalah w_{ij} dan w_{jk}) diberi nilai inisial atau nilai awal, lalu MLP tersebut dilatih. (Algoritma pelatihan pada JST secara umum cukup kompleks, karena itu di sini hanya akan diberikan inti langkah-langkahnya saja.) Secara umum, tahap pelatihan ini terdiri dari dua langkah utama, yaitu:

- *Feed-forward* (pengumpanan ke depan): keluaran (Y pada persamaan 3) pada tiap perceptron dihitung berdasar data masukan yang diterima dan bobot-bobot pada jaringan, dengan urutan dari lapis terdepan ke belakang.
- *Back-propagation* (propagasi balik): dengan menggunakan turunan fungsi aktivasi, Y , sebuah nilai *learning rate* dan bobot-bobot saat sekarang, dilakukan perbaikan nilai-nilai bobot dari lapis terbelakang (bobot-bobot pada output layer) ke depan (bobot-bobot pada hidden layer, lalu input layer).

Dua langkah di atas dilakukan secara bergantian (feed-forward lalu back-propagation) untuk tiap baris/rekord pada data training/pelatihan. Jadi, perbaikan bobot-bobot dilakukan pada pemrosesan tiap baris/rekord. Satu siklus pelatihan, dimana seluruh baris pada data training sudah diumpulkan ke MLP, dinamakan *epoch*. Setelah satu epoch selesai, jika nilai keluaran belum mendekati (atau sama dengan) nilai kelas pada data training, siklus pelatihan akan diulangi lagi. Demikian seterusnya sampai didapatkan bobot-bobot yang paling baik atau jumlah maksimum epoch yang ditetapkan telah dicapai. Bergantung dari data training dan konfigurasi MLP, pelatihan dapat membutuhkan puluhan sampai ribuan epoch.

Hal yang perlu diketahui, untuk melatih MLP, semakin banyak kasus (baris) pada data training yang “mewakili” tiap kelas, umumnya bobot-bobot akan semakin baik. Dengan kata lain, model MLP akan semakin akurat (dalam melakukan prediksi).

Setelah proses pelatihan selesai, akan diperoleh bobot-bobot final. Sebagai ilustrasi, pada Gambar 7.11 ditunjukkan MLP yang sudah melewati tahap pelatihan. Garis yang bergaris tebal menggambarkan bahwa sisi tersebut mempunyai bobot yang lebih besar dan merupakan kriteria yang lebih mempengaruhi *perceptron* di layer selanjutnya. Sebaliknya, untuk garis yang tipis atau buram menunjukkan bahwa bobot pada sisi tersebut sangat kecil atau kurang berpengaruh.



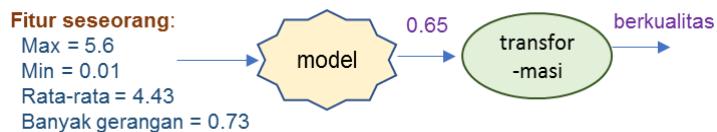
Gambar 7.11. Ilustrasi bobot-bobot MLP setelah pelatihan selesai.

Setelah model MLP dihitung dari data training, selanjutnya model diuji akurasinya menggunakan data uji. Jika akurasi model MLP dinilai cukup baik (misalnya di atas 70%), maka model siap dimanfaatkan untuk memprediksi kualitas tidur seseorang.

7.4.4 Pemanfaatan Model untuk Prediksi

Berdasarkan penelitian yang dilaporkan pada (A Sathyanarayana, 2016) disimpulkan bahwa model MLP cocok dimanfaatkan untuk prediksi kualitas tidur karena memiliki tingkat akurasi yang baik.

Pada contoh di sini, cara pemanfaatan model MLP untuk memprediksi kualitas tidur seseorang dilakukan dengan mengumpulkan sebuah rekord (baris) berisi nilai-nilai fitur (pada contoh di atas: max, min, rata-rata aktivitas dan banyak gerakan). Model lalu akan menghitung nilai keluaran (kualitas tidur) berdasar data input dan bobot-bobot pada input layer dan output layer. Jika angka pada node output menghasilkan nilai diantara 0 – 0,5 berarti prediksinya adalah “Tidak Berkualitas” sedangkan jika nilainya diantara 0,5 – 1 berarti prediksinya adalah “Berkualitas”. Pada program aplikasi yang memanfaatkan model, dapat ditambah dengan fungsi untuk mengubah nilai numerik menjadi biner, dengan nilai berkualitas/tidak berkualitas, sehingga dapat dihasilkan hasil prediksi bernilai biner (lihat Gambar 7.12).



Gambar 7.12. Ilustrasi prediksi kualitas tidur seseorang.

7.5. Penutup

Bab ini telah memberikan gambaran penerapan tahapan data science pada kasus [klasifikasi data](#), dimana MLP digunakan pada pembuatan model, yang lalu dapat digunakan memprediksi kualitas tidur seseorang. Data yang dikumpulkan dan dianalisis berasal dari wearable device (smartwatch).

Pada sistem nyata (riil), data yang dikumpulkan di server dapat berasal dari berjuta-juta smartwatch. Karena itu, pengumpulan data perlu ditangani oleh sistem big data. Karena data terkumpul dengan cepat dan berukuran sangat besar, algoritma untuk membuat model juga algoritma untuk big data dan komputasi model dilakukan dengan memanfaatkan teknologi big data. Sistem dan komputasi tersebut kompleks. Agar seorang data scientist dapat melakukan pekerjaan semacam ini, data scientist perlu memiliki berbagai skill dan keahlian yang dibahas pada Bab 1.

Referensi

- (A Sathyanarayana, 2016) Sleep Quality Prediction From Wearable Data Using Deep Learning, *JMIR Mhealth Uhealth*, Vol. 4, No. 4
- (DL Reed, 2016) Measuring Sleep Efficiency: What Should the Denominator Be?, *Journal of Clinical Sleep Medicine*, Vol 12., No. 2.
- (Han, J., Pei, J., & Kamber, M., 2012) *Data Mining: Concepts and Techniques 3rd Ed.*, Morgan Kauffman Publ., USA.
- (Watelectronics, 2020) <https://www.watelectronics.com/different-types-of-sensors-with-applications/> (diakses 20 Juni 2020)

Bab 8 Rekomendasi Film dengan Fuzzy Collaborative Filtering

Oleh:

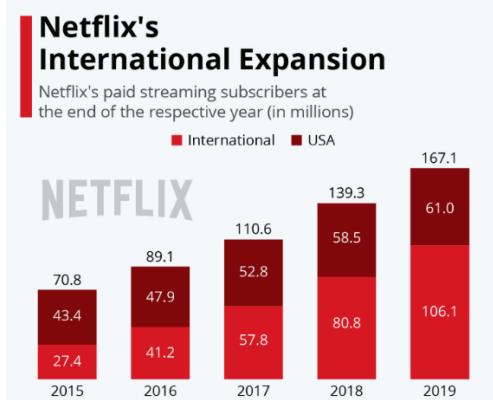
Husnul Hakim

8.1. Pendahuluan

Bagi para pembaca yang suka nonton film, tentunya sudah tidak asing lagi dengan istilah *rating* atau penilaian film. Penilaiai suatu film ada yang diberikan oleh kritikus film, dan ada pula yang diberikan oleh para penonton yang sudah menonton film tersebut. Umumnya, para penonton hanya dapat melihat rangkuman hasil penilaian dari seluruh penonton lain. Rangkuman ini yang nantinya kita gunakan untuk membantu pengambilan keputusan apakah suatu film akan kita tonton atau tidak.

Kita juga sudah tidak asing dengan layanan pemutar film daring, seperti Netflix, Amazon Prime Video, dan HBO Go. Layanan pemutar film daring ini semakin banyak dipilih oleh masyarakat, lagi-lagi karena kemudahannya. Sekarang, kita tidak perlu pergi ke bioskop untuk menonton film yang berkualitas; cukup dari rumah saja, dengan bermodalkan koneksi internet yang baik. Dari segi biaya, menggunakan layanan pemutar film daring juga tidak mahal. Sebagai ilustrasi, Gambar 8.1 menunjukkan peningkatan pengguna salah satu platform layanan film daring terbesar, yaitu Netflix.

Untuk meningkatkan kepuasan pengguna, layanan pemutar film daring memberikan rekomendasi film, yang dianggap cocok untuk pengguna tertentu. Gambar 8.2 memberikan contoh rekomendasi yang diberikan oleh layanan pemutar film daring.



Gambar 8.1. Pengguna Netflix tahun 2015-2019²⁵.



Gambar 8.2. Rekomendasi film yang diberikan oleh Netflix²⁶.

²⁵ <https://www.businessofapps.com/data/netflix-statistics/#1>

²⁶ Gambar diambil dari akun layanan pemutar film daring pengguna

Namun, bagaimana suatu rekomendasi film diberikan? Misalnya, suatu hari, Andy ingin menonton X-Men, namun ia tidak yakin apakah ia akan menyukai film tersebut. Karena itu, ia bertanya kepada seorang temannya, yaitu Citra, apakah X-Men adalah film yang bagus atau tidak. Namun apakah benar jika Citra menyukai film X-Men maka Andy juga akan menyukai film itu? Belum tentu. Secara intuitif kita dapat memprediksi, pendapat Citra tersebut akan bermanfaat bagi Andy jika judul-judul film yang disukai oleh Citra mirip dengan judul-judul film yang disukai oleh Andy. Akan lebih baik lagi apabila judul-judul film yang tidak disukai oleh Citra juga mirip dengan judul-judul film yang tidak disukai oleh Andy. Dengan demikian, pendapat Citra tentang film X-Men dapat mewakili pendapat Andy tentang film tersebut.

Andy dapat lebih yakin apakah ia akan menyukai film X-Men ketika menontonnya, jika ia tidak hanya bertanya atau meminta pendapat dari Citra. Tapi dia perlu bertanya juga kepada teman-temannya yang memiliki selera yang mirip dengan Andy dalam hal kesukaan dan ketidaksukaan terhadap film-film. Dari mereka semua kemudian Andy, dengan lebih pasti, dapat menentukan apakah ia akan menyukai film X-Men atau tidak.

Cara yang digunakan di atas, sebenarnya adalah cara yang digunakan berbagai *website* penjual berbagai produk atau jasa untuk memberikan berbagai rekomendasi produk/jasa kepada setiap pengguna *website*. Rekomendasi yang diberikan kepada seseorang dapat didasari oleh kemiripan dia dengan para pengguna lainnya. Sistem seperti ini dikenal dengan *user-based collaborative filtering recommendation system* atau sistem rekomendasi yang memanfaatkan algoritma *user-based collaborative filtering*. Gambar 8.3 memberikan ilustrasi dari sistem rekomendasi *user-based collaborative filtering*.



Gambar 8.3. Ilustrasi sistem rekomendasi User-based Collaborative Filtering.

Pada Gambar 8.1, kita dapat melihat bahwa Andy menyukai film dengan judul Harry Potter, Frozen, The Conjuring dan The Avengers. Sementara itu, Citra menyukai film Harry Potter, Frozen, X-men, The Conjuring, dan The Avengers. Dari sini, kita dapat mengetahui bahwa ada kemiripan antara Andy dan Citra berdasarkan kesukaannya dan ketidaksukaannya terhadap film. Karena Citra menyukai X-Men, maka dapat diduga bahwa Andy juga akan menyukai X-Men. Dengan demikian, film dengan judul X-Men akan direkomendasikan kepada Andy.

Untuk dapat memberikan rekomendasi film berdasar film-film yang disukai penonton lain (yang memiliki kemiripan selera film) kepada seseorang, pertama-tama harus dikumpulkan terlebih dahulu data penilaian (rating) terhadap berbagai film yang dilakukan oleh para penonton. Setelah data tersebut diperoleh, penonton-penonton ini akan dikelompokkan. Penonton-penonton yang memberikan penilaian yang mirip terhadap berbagai jenis film akan berada pada kelompok yang sama.

Film-film yang akan direkomendasikan kepada calon penonton ini adalah film-film yang belum pernah dia tonton, yang diberi nilai yang baik oleh anggota-anggota lain dalam kelompok tersebut. Karena penonton-penonton di dalam satu kelompok memiliki kemiripan, maka dapat diprediksi bahwa film-film yang dinilai baik oleh anggota lain akan dinilai baik pula oleh calon penonton ini. Dengan demikian, rekomendasi yang diberikan merupakan rekomendasi yang tepat.

Pada bab ini akan dibahas suatu algoritma dalam sistem rekomendasi yang dikenal dengan nama *collaborative filtering*. Algoritma ini lalu digabungkan dengan algoritma pengelompokan yang dikenal dengan algoritma *c-Means* untuk menghasilkan rekomendasi yang lebih baik dibandingkan dengan *collaborative filtering* biasa.

8.2. User-based Collaborative Filtering

User-based collaborative fitlering adalah algoritma pemberi rekomendasi yang bekerja berdasarkan kemiripan sekelompok orang. Kata *user* sendiri mengacu kepada orang yang menjadi pengguna sistem rekomendasi. Sebagai contoh, untuk dapat memberikan rekomendasi film kepada seorang pengguna, misalnya Andy, maka akan dicari sekelompok pengguna lainnya yang menyukai film-film yang sama dengan yang disukai Andy. Film yang akan direkomendasikan kepada Andy adalah film-film yang belum pernah ditonton oleh Andy, namun disukai oleh pengguna lain di dalam kelompoknya.

Untuk dapat bekerja dengan benar, algoritma *user-based collaborative filtering* membutuhkan *dataset* (himpunan data) yang akan menjadi masukan. Dataset tersebut adalah tabel berisi penilaian untuk tiap produk dari tiap pengguna. Selanjutnya, tabel ini akan disebut sebagai tabel penilaian. Hasil dari *user-based collaborative filtering* adalah prediksi apakah seorang pengguna akan menyukai atau tidak menyukai suatu produk. Selain itu, hasil algoritma tersebut juga dapat berupa daftar produk yang direkomendasikan kepadanya (Jannach, Zanker, Felfernig, & Friedrich, 2011). Pada Tabel 8.1, kita dapat melihat contoh tabel penilaian untuk suatu film.

Tabel 8.1. Contoh Tabel Penilaian

Pengguna	The Usual Suspects	7even	Back to The Future	The Hobbit
Andy	?	4	3	2
Bobby	2	4	2	1
Citra	1	5	3	3
Dodo	3	3	3	2
Ernie	4	2	3	4

Pada Tabel 8.1, terdapat lima orang pengguna yaitu Andy, Bobby, Citra, Dodo, dan Ernie. Angka 1 sampai dengan 5 menunjukkan penilaian yang diberikan oleh para pengguna terhadap film yang terdapat pada tiap kolom. Angka 1 menunjukkan bahwa pengguna sangat tidak menyukai suatu film, sedangkan angka 5 menunjukkan bahwa pengguna sangat menyukai suatu film. Tanda tanya menunjukkan bahwa seorang pengguna belum menilai atau belum pernah menonton suatu film. Pada tabel tersebut terlihat bahwa Andy belum pernah menonton film dengan judul The Usual Suspect. Sistem rekomendasi dengan *user-based collaborative filtering* dapat memprediksi nilai yang akan diberikan oleh Andy terhadap film dengan judul The Usual Suspect.

Setelah tabel penilaian didapatkan, maka proses pemberian rekomendasi dapat dilakukan. Untuk lebih jelasnya, langkah-langkah pemberian rekomendasi ini diilustrasikan oleh Gambar 8.4. Pada gambar itu kita dapat melihat bahwa terdapat tiga langkah untuk mendapatkan hasil prediksi, yaitu tahap perhitungan rata-rata, perhitungan nilai kemiripan, dan perhitungan prediksi.



Gambar 8.4.. Tahap-tahap pada algoritma User-based Collaborative Filtering.

Pertama-tama, dari tabel penilaian akan dihitung rata-rata penilaian yang diberikan oleh setiap pengguna untuk semua produk. Dalam perhitungan rata-rata, produk yang belum pernah diberi nilai oleh seorang pengguna akan dianggap bernilai 0. Sebagai contoh, dari Tabel 8.1 dapat diperoleh rata-rata nilai yang

diberikan oleh tiap pengguna. Hasilnya dapat dilihat pada Tabel 8.2. Nilai rata-rata ini akan digunakan pada perhitungan kemiripan pada langkah kedua.

Tabel 8.2. Contoh Perhitungan Rata-rata Nilai untuk Tiap Film

Pengguna	The Usual Suspects	7even	Back to The Future	The Hobbit	Rata-rata
Andy	0	4	3	2	3
Bobby	2	4	2	1	2.25
Citra	1	5	3	3	3
Dodo	3	3	3	2	2.75
Ernie	4	2	3	4	3.25

Dengan rata-rata penilaian pengguna, kita kemudian dapat menghitung kemiripan antara dua orang pengguna. Pengguna pertama adalah pengguna yang akan diberi rekomendasi, sedangkan pengguna kedua adalah pengguna lainnya. Nilai ini dapat diperoleh dengan menggunakan sebuah persamaan yang dikenal dengan *Pearson's Correlation Coefficient*. Nilai kemiripan ini berada pada rentang -1 sampai dengan +1. Nilai -1 menunjukkan bahwa dua orang pengguna sangat bertolak belakang preferensinya. Nilai +1 menunjukkan bahwa dua orang pengguna sangat mirip. Sementara itu, nilai 0 menunjukkan bahwa dua orang pengguna tidak memiliki kemiripan.

Pada contoh kasus sebelumnya, yang akan diprediksi adalah nilai yang akan diberikan oleh Andy untuk film berjudul The Usual Suspect. Oleh karena itu, akan dihitung kemiripan antara Andy dengan semua pengguna lainnya. Contoh hasil perhitungan ini ditunjukkan pada Tabel 8.3.

Tabel 8.3. Contoh Nilai Kemiripan antara Andy dengan Para Pengguna Lainnya

Pengguna	Kemiripan dengan Andy
Bobby	0.98
Citra	0.86
Dodo	0.86
Ernie	-1.00

Setelah diperoleh nilai kemiripan dari seorang pengguna terhadap pengguna lainnya, maka dapat dilakukan prediksi. Yang dimaksud dengan prediksi adalah perkiraan nilai yang akan diberikan oleh seorang pengguna untuk sebuah produk. Sebelum melakukan perhitungan, perlu ditentukan banyaknya pengguna yang paling mirip dengan pengguna yang akan diberi rekomendasi. Sebagai contoh, untuk memprediksi nilai yang akan diberikan oleh Andy untuk film The Usual Suspect, akan dipilih dua orang yang paling mirip dengan Andy, yaitu Bobby dan Citra. Perhitungan rata-rata ini dilakukan dengan menggunakan rata-rata terbobot dari para pengguna yang mirip ini. Dari perhitungan tersebut dapat

terlihat bahwa nilai yang mungkin akan diberikan oleh Andy untuk film berjudul The Usual Suspect adalah 1.035.

Langkah terakhir dari algoritma ini adalah menghitung nilai prediksi berdasarkan beberapa orang yang mirip. Banyaknya orang yang mirip ini akan mempengaruhi ketepatan hasil rekomendasi (Jannach dkk, 2011). Sayangnya, belum ada penelitian yang dapat menentukan dengan pasti berapa banyak pengguna yang mirip yang harus dimasukkan dalam perhitungan prediksi. Namun, (Herlocker Jon, Konstan, & Riedl, 2002) menyatakan bahwa dengan menggunakan data dari 20 sampai 50 orang sudah akan dapat dihasilkan prediksi yang dengan tingkat ketepatan yang baik.

Cara lain yang dapat digunakan untuk menentukan pengguna-pengguna yang memiliki kemiripan tinggi adalah dengan melakukan *clustering* (Koohi & Kiani, 2016). Dengan *clustering*, para pengguna yang mirip akan dikelompokkan ke dalam kelompok yang sama. Perhitungan prediksi akan dilakukan berdasarkan anggota dari kelompok ini.

8.3. Algoritma Clustering Fuzzy c-Means

Pengelompokan atau *clustering* adalah salah satu algoritma dari Data Mining (penambangan data) yang berbasis Machine Learning. Clustering digunakan untuk mengelompokkan objek-objek, sehingga objek-objek di dalam kelompok yang sama akan memiliki kemiripan satu sama lainnya, sekaligus memiliki perbedaan yang signifikan atau relatif besar dengan objek-objek yang menjadi anggota pada kelompok lainnya (Tan, Steinbach, & Kumar, 2005). Terdapat sejumlah algoritma untuk melakukan clustering terhadap dataset (himpunan data), salah satunya adalah *Fuzzy c-means*.

Dalam kasus rekomendasi film yang dibahas sebelumnya, yang dimaksud dengan objek adalah para pengguna, yaitu Andy, Bobby, Citra, Dodo, dan Ernie. Agar objek-objek dapat dikelompokkan, harus dipilih kriteria pengelompokannya. Kriteria ini kita kenal dengan istilah atribut. Dalam rekomendasi film, kriteria pengelompokan adalah nilai-nilai yang telah diberikan oleh untuk semua film yang ada. Dengan demikian, pada contoh kasus sebelumnya, masing-masing pengguna memiliki empat buah atribut, yaitu nilai untuk film The Usual Suspect, 7even, Back to The Future, dan The Hobbit. Untuk kemudahan, akan diberikan notasi Andy = {0, 4, 3, 2} yang mewakili nilai yang diberikan oleh Andy untuk keempat film tersebut secara berturut-turut.

Pada *Fuzzy c-means*, setiap objek dapat dimasukkan ke dalam lebih dari satu buah kelompok, dengan kadar atau derajat keanggotaan yang tertentu. Kadar atau derajat ini dikenal dengan derajat keanggotaan. Derajat keanggotaan sebuah objek di dalam sebuah kelompok berada pada nilai 0 hingga 1. Derajat keanggotaan yang bernilai 0 menandakan bahwa sebuah objek tidak menjadi anggota dari suatu himpunan, sedangkan 1 berarti anggota penuh dari suatu himpunan.

Untuk dapat mengelompokkan objek-objek dengan *Fuzzy c-Means*, pertama-tama kita perlu menentukan banyaknya kelompok. Sebagai contoh, pada kasus ini, banyak kelompok adalah 2. Ini berarti, lima orang pengguna yaitu Andy, Bobby, Citra, Dodo, dan Ernie akan dibagi menjadi dua buah kelompok.

Algoritma *Fuzzy c-Means* lalu akan menghitung nilai derajad keanggotaan pada tiap objek. Perhitungan ini dilakukan secara iteratif atau berulang-ulang. Pada iterasi yang pertama, setiap objek harus diberi derajad keanggotaan awal (inisial). Nilai ini diberikan secara acak, namun jumlah derajat keanggotaan suatu objek di semua kelompok harus sama dengan 1. Setelah itu, berdasarkan nilai-nilai atribut objek, derajad keanggotaan tersebut akan diperbarui (dihitung ulang) beberapa kali sampai nilai derajad keanggotaan (hampir) tidak berubah. Berikut ini diberikan ilustrasi proses clustering terhadap himpunan data pada Tabel 8.1.

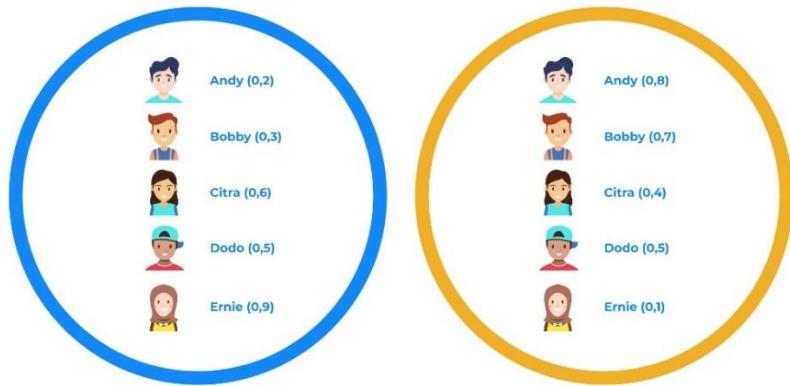
Pada tahap awal, nilai derajat keanggotaan inisial (awal) untuk tiap objek pada tiap kelompok ditunjukkan pada Tabel 8.4.

Tabel 8.4. Contoh Pemberian Nilai Awal untuk Derajat Keanggotaan

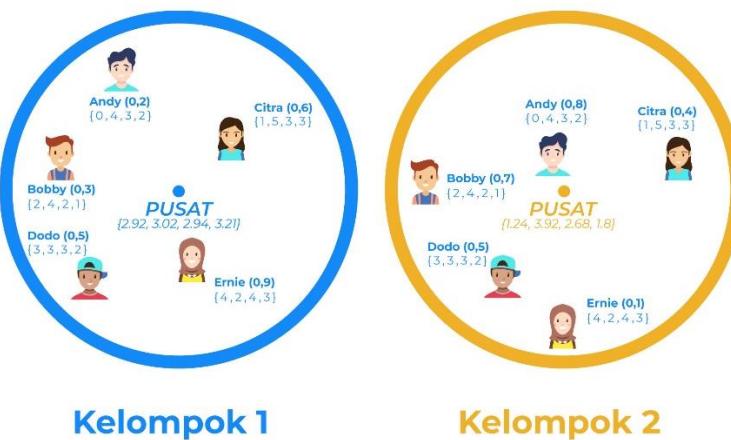
Pengguna	Derajat Keanggotaan di Kelompok 1	Derajat Keanggotaan di Kelompok 2
Andy	0.2	0.8
Bobby	0.3	0.7
Citra	0.6	0.4
Dodo	0.5	0.5
Ernie	0.9	0.1

Visualisasi dari Tabel 8.4 ditunjukkan pada

Gambar 8.5. Pada gambar ini dapat kita lihat bahwa tiap pengguna masuk ke masing-masing kelompok 1 dan 2 dengan derajat keanggotaan tertentu. Sebagai contoh, derajat keanggotaan Andy di kelompok 1 adalah sebesar 0.2 dan di kelompok 2 adalah sebesar 0.8. Total derajat keanggotaan Andy di semua kelompok adalah 1. Hal ini juga berlaku untuk setiap pengguna lainnya.

**Kelompok 1****Kelompok 2***Gambar 8.5. Visualisasi derajat keanggotaan tahap awal.*

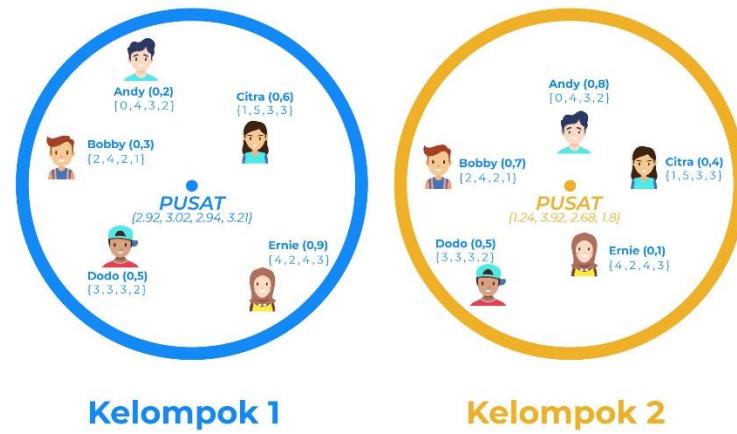
Sekarang, tiap kelompok sudah memiliki anggota dengan derajat keanggotaan tertentu. Langkah selanjutnya adalah menghitung titik pusat dari setiap kelompok. Titik pusat dari tiap kelompok diperoleh dengan menghitung rata-rata terbobot dari nilai-nilai atribut pada objek-objek yang terdapat di dalam suatu kelompok. Ilustrasi tahap ini ditunjukkan pada Gambar 8.6. Pada ini titik pusat tiap kelompok ditandai dengan titik berwarna biru untuk kelompok 1, dan berwarna merah untuk kelompok 2. Karena tiap objek menyimpan empat buah nilai atribut yang menyatakan nilai untuk empat buah film, maka titik pusat juga menyimpan empat buah nilai atribut.

*Gambar 8.6. Ilustrasi perhitungan titik pusat dari tiap kelompok.*

Pada iterasi berikutnya, algoritma *Fuzzy c-Means* menghitung kembali derajat keanggotaan tiap objek di dalam tiap kelompok. Perhitungan derajat keanggotaan ini didasarkan prinsip bahwa objek-objek yang dekat dengan titik pusat kelompok, akan memiliki derajat keanggotaan yang lebih tinggi dibandingkan dengan objek-objek yang berjauhan dengan titik pusat kelompok. Oleh karena itu perhitungan derajat keanggotaan ini akan melibatkan perhitungan jarak antara setiap objek dengan titik pusat dari kelompok.

Gambar 8.7 merupakan ilustrasi dari perubahan derajat keanggotaan. Pada gambar itu terlihat bahwa objek-objek dengan derajat keanggotaan yang lebih tinggi adalah objek-objek yang berada lebih dekat dari titik pusat kelompok. Sebaliknya, derajat keanggotaan yang rendah akan diberikan kepada objek-objek yang jauh dari titik pusat kelompok.

Proses perhitungan titik pusat kelompok dan derajat keanggotaan ini akan terus-menerus dilakukan pada iterasi-iterasi berikutnya sampai tidak ada lagi derajat keanggotaan yang berubah nilainya. Sebagai contoh, setelah beberapa kali dilakukan perhitungan ulang terhadap derajat keanggotaan dan titik pusat, diperoleh hasil akhir seperti yang ditunjukkan oleh Gambar 8.. Berdasar hasil akhir tersebut, maka tiap objek sudah dapat ditentukan kelompoknya. Pemilihan kelompok yang tepat untuk tiap objek, dilakukan dengan memilih kelompok di mana objek yang ditelaah itu memiliki derajat keanggotaan tertinggi. Sebagai contoh, pada Gambar 8.8, Andy akan masuk ke kelompok 1 karena derajat keanggotaan Andy di kelompok 1 lebih tinggi daripada derajat keanggotanya di kelompok 2. Tabel 8.5 menunjukkan hasil pengelompokan para pengguna yang dihitung oleh algoritma *Fuzzy c-means*. Jika kita lihat pada Tabel 8.5, hanya Ernie yang masuk ke kelompok 2, sementara pengguna lainnya berada di dalam kelompok 1.



Gambar 8.7. Ilustrasi pergantian nilai derajat keanggotaan.



Gambar 8.8. Ilustrasi pengelompokan setelah dilakukan pergantian nilai derajat keanggotaan dan titik pusat berkali-kali.

Tabel 8.5. Hasil Pengelompokan Pengguna Menggunakan Fuzzy c-Means

Pengguna	Derajat Keanggotaan di Kelompok 1	Derajat Keanggotaan di Kelompok 2	Kelompok
Andy	0.81	0.19	1
Bobby	0.90	0.10	1
Citra	0.85	0.15	1
Dodo	0.85	0.15	1
Ernie	0.45	0.55	2

8.4. Hasil Penelitian Rekomendasi Film dengan Fuzzy Collaborative Filtering

Pada dua sub-bab sebelumnya telah dibahas algoritma Collaborative Filtering dan Fuzzy c-Means. Algoritma Collaborative Filtering berfungsi untuk menghitung penilaian pengguna berdasar penilaian pengguna-pengguna yang saling mirip. Algoritma Fuzzy c-Means digunakan untuk mengelompokkan pengguna-pengguna berdasar atribut-atribut penilaian dari para pengguna tersebut. Fuzzy c-Means akan menghasilkan kelompok-kelompok pengguna, dimana para pengguna yang tergabung dalam satu kelompok saling mirip satu terhadap lainnya. Rekomendasi kepada seorang pengguna, dapat diberikan berdasar perhitungan penilaian dari pengguna-pengguna lain yang berada di dalam kelompok yang sama.

Pada sub-bab ini dibahas sebuah hasil penelitian (Koohi & Kiani, 2016) yang membuktikan bahwa kedua algoritma tersebut, yaitu Collaborative Filtering dan Fuzzy c-Means dapat digabungkan agar dapat dihasilkan model pemberi rekomendasi yang tepat. Fuzzy c-Means digunakan untuk mengelompokkan para pengguna (dalam konteks ini penonton film-film), sedangkan Collaborative Filtering berfungsi untuk menghitung dan memberikan rekomendasi film kepada pengguna (calon penonton film).

Data mentah yang digunakan dalam penelitian ini adalah data yang berisi penilaian yang diberikan oleh penonton untuk berbagai judul film. Data ini diambil dari *movielens dataset* (<https://grouplens.org/datasets/movielens/>). Data penilaian dikumpulkan oleh GroupLens Research Project dari Universitas Minnesota. Data ini berisi 100.000 penilaian yang diberikan oleh 943 pengguna untuk 1682 film. Setiap pengguna memberikan paling sedikit penilaian terhadap 20 film. Data ini dikumpulkan sejak September 1997 sampai April 1988. Data yang digunakan ini berisi empat kolom, yaitu kolom identitas pengguna, identitas film yang dinilai, nilai untuk film tersebut, serta *timestamp*.

Sebagaimana dibahas pada Bab 1, mula-mula data mentah di atas perlu untuk disiapkan terlebih dahulu agar dapat diumpulkan ke algoritma *fuzzy collaborative filtering*. Tahap persiapan yang pertama adalah pemilihan kolom-kolom yang relevan. Dalam hal ini, kolom *timestamp* bukanlah kolom yang relevan. Karena itu kolom ini bisa diabaikan. Selanjutnya, untuk dapat digunakan dalam rekomendasi, data tersebut juga perlu ditransformasi sehingga bentuknya sama seperti Tabel 8.1. Setelah itu barulah data dapat diumpulkan ke algoritma *fuzzy collaborative filtering* untuk diproses. Hasil atau keluaran dari sistem rekomendasi ini adalah daftar film-film yang direkomendasikan kepada seorang pengguna.

Dengan menggunakan data yang telah disiapkan di atas, dilakukan berbagai eksperimen guna mengetahui kinerja dari *fuzzy collaborative filtering*. Data itu dipisah, 80% digunakan untuk pembuatan model rekomendasi dengan menggunakan algoritma *fuzzy collaborative filtering*. Sisanya, yang 20% digunakan untuk data uji. Salah satu ukuran apakah model dapat dimanfaatkan atau tidak adalah akurasi. Akurasi dihitung dengan membandingkan keluaran atau hasil rekomendasi dengan penilaian yang sebenarnya yang diberikan oleh pengguna pada data masukan. Pada penelitian ini, hasil penilaian pengguna pada data uji dibandingkan dengan hasil rekomendasi untuk menghitung akurasi dari *fuzzy collaborative filtering*. Cara menghitung akurasi adalah dengan menjumlahkan *true positive* dengan *true negative* kemudian membaginya dengan banyaknya film.

True positive dihitung dari banyaknya *film* yang direkomendasikan oleh *fuzzy collaborative filtering* yang memang disukai oleh pengguna yang diuji. Sementara itu, *true negative* dihitung dari banyaknya *film* yang tidak direkomendasikan oleh *fuzzy collaborative filtering* yang memang tidak disukai oleh pengguna yang diuji. Sebagai contoh, dari 1682 film yang ada pada data masukan, diberikan 5 rekomendasi untuk Andy. Dari lima film tersebut, hanya 4 yang benar-benar disukai oleh Andy berdasarkan penilaian yang terdapat pada data masukan. Dengan demikian nilai *true positive* adalah 4. Selanjutnya, untuk menghitung nilai *true negative*, perlu dihitung banyaknya film yang tidak direkomendasikan kepada Andy yang benar-benar tidak ia sukai. Sebagai contoh, dari 1677 film yang tidak direkomendasikan, ada 1600 film yang memang tidak disukai oleh Andy berdasarkan data masukan. Dengan demikian nilai *true negative* adalah sebesar 1600. Dari contoh tersebut, maka nilai akurasi adalah sebesar $\frac{4+1600}{1682} = 95.36\%$.

Karena menggunakan *Fuzzy c-means* untuk pengelompokan pengguna sebelum penilaian terhadap film dihitung (dari para pengguna yang berada di dalam satu kelompok), sedangkan jumlah kelompok tersebut harus ditentukan di depan, maka pada penelitian ini dilakukan eksperimen untuk mencari jumlah kelompok yang menghasilkan akurasi terbaik. Tabel 8.6 menunjukkan hasil eksperimen ini. Pada tabel tersebut terlihat bahwa akurasi terbaik dihasilkan jika banyaknya kelompok adalah 3. Hal ini berarti

bawa hasil prediksi dan rekomendasi menjadi akurat jika banyak kelompok adalah sebesar 3. Dari Tabel 8.6 juga dapat dilihat bahwa akurasi cenderung menurun seiring dengan pertambahan banyak kelompok.

Selain meneliti tentang pengaruh banyaknya kelompok terhadap akurasi dari sistem rekomendasi, Koohi dan Kiani juga membandingkan penggunaan *Fuzzy c-Means* dengan algoritma pengelompokan yang lain, yaitu *k-Means*. Pengelompokan k-means adalah suatu algoritma pengelompokan yang mirip dengan *c-Means*. Perbedaannya adalah pada derajat keanggotaan. Pada *k-Means*, sebuah objek memiliki derajat keanggotaan pada satu buah kelompok dengan nilai 0 atau 1. Dengan kata lain, sebuah objek hanya bisa secara utuh menjadi anggota dari satu kelompok, dan tidak menjadi anggota kelompok yang lain. Pada Gambar 8.9 ditunjukkan perbandingan nilai akurasi algoritma *Fuzzy c-Means* terhadap *k-Means*.

Tabel 8.6. Hasil Eksperimen Akurasi untuk beberapa Kelompok (Koohi & Kiani, 2016)

Banyak Kelompok	Akurasi
3	80.44%
5	80.33%
7	80.17%
9	80.12%
11	79.92%
13	79.82%
15	79.91%



Gambar 8.9. Perbandingan akurasi Fuzzy C-means dan k-Means (Koohi & Kiani, 2016).

8.5. Penutup

Pada bab ini telah dibahas sistem rekomendasi dengan memanfaatkan algoritma *user-based collaborative filtering*. Rekomendasi yang diberikan oleh algoritma ini akan bergantung pada sejumlah pengguna lain yang saling memiliki kemiripan yang tinggi. Agar dihasilkan rekomendasi yang tepat, perlu dicari kelompok dengan anggota pengguna-pengguna yang mirip, lalu dari kelompok pengguna ini rekomendasi diberikan terhadap pengguna-pengguna di kelompok tersebut.

Untuk mengatasi hal tersebut, telah dibahas hasil penelitian yang menggabungkan algoritma pengelompokan Fuzzy c-Means dengan User-Based Collaborative Filtering. Peranan Fuzzy c-means adalah untuk mendapatkan kelompok-kelompok pengguna. Pengguna-pengguna yang mirip akan masuk ke dalam kelompok yang sama, sehingga rekomendasi terhadap seorang pengguna didasarkan pada penilaian dari para pengguna lain yang berada di kelompok yang sama. Pada penelitian tersebut, hasil pengujian dengan data penilaian film yang riil telah memberikan hasil bahwa pemberian rekomendasi dengan algoritma tersebut memiliki akurasi di atas 80%. Artinya, lebih dari 80% rekomendasi yang

diberikan adalah rekomendasi yang tepat. Dengan demikian, model yang dihasilkan pada penelitian ini dapat dimanfaatkan untuk pemberian rekomendasi film yang layak ditonton.

Referensi

- (Herlocker Jon, Konstan, & Riedl, 2002) Herlocker Jon, Konstan, J. A., & Riedl, J., "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. Information Retrieval," 287-310, 2002.
- (Jannach dkk, 2011) Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G., "Reccomender System: An Introduction," New York: Cambridge University Press. 2011.
- (Koohi & Kiani, 2016) Koohi, H., & Kiani, K., "User Based Collaborative Filtering using Fuzzy C-Means," Measurement, Volume 91, Pages 134-139, 2016.
- (Tan, Steinbach, & Kumar, 2005) Tan, P.-N., Steinbach, M., & Kumar, V., "Introduction to Data Mining," Pearson, 2005.

Bab 9 Urun Daya Data Kepadatan Lalu Lintas

Oleh:

Pascal Alfadian

9.1. Pendahuluan

Aplikasi Google Maps dan Waze semakin popular beberapa tahun belakangan ini. Dengan semakin padatnya lalu lintas dunia, orang-orang menggunakan kedua aplikasi ini untuk memantau kepadatan lalu lintas dan penentu rute terbaik, dengan jarak optimal dan kepadatan minimal, dari satu tempat ke tempat lain. Tak jarang, aplikasi ini juga digunakan untuk mencari suatu lokasi, dan cara untuk menuju lokasi tersebut.

Jika pembaca pernah menggunakan aplikasi Google Maps atau Waze selama berkendara, pembaca pasti sudah memahami bahwa kedua aplikasi tersebut mampu beradaptasi atau memberikan informasi yang terkini tentang kepadatan lalu lintas yang terjadi di ruas-ruas jalan tertentu yang akan pembaca lalui. Suatu ruas jalan dapat berwarna hijau, misalnya jika lancar, oranye jika agak padat, dan merah tua jika sangat macet. Lihat Gambar 9.1 sebagai ilustrasi. Warna-warna ini dapat berubah sesuai dengan kondisi *real time* ruas jalan terkait.

Bagaimana kedua aplikasi tersebut dapat menampilkan informasi kepadatan tersebut? Prosesnya cukup kompleks, dengan memanfaatkan teknologi big data dengan sumber daya yang masif, dan melalui tahap-tahap data science yang sudah dibahas di Bab 1.

Bab ini memberi gambaran umum dan sederhana tentang bagaimana informasi kepadatan tersebut dapat disediakan bagi para pengguna aplikasi. Setelah itu, di bagian akhir, para pembaca akan diajak menjadi *data scientist* kecil-kecilan dengan memanfaatkan data yang disediakan oleh Google Maps.



Gambar 9.1. Tampilan Google Maps pada suatu daerah di Bandung²⁷.

9.2. Pengukuran Kepadatan Lalu Lintas oleh Google Maps

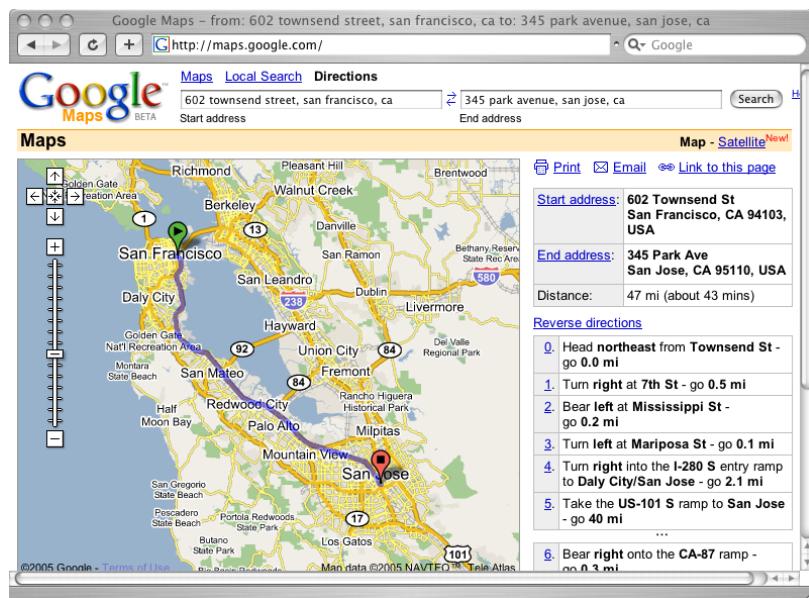
Deteksi kepadatan lalu lintas sudah dilakukan sejak lama. Laporan dari Federal Highway Administration di tahun 2006 menyebutkan bahwa ide ini muncul di tahun 1920an, saat kontrol sinyal otomatis (“lampa merah”, atau APILL / Alat Pemberi Isyarat Lalu Lintas) mulai menggantikan persinyalan manual (FHWA, 2006). Pada tahun 1928, Charles Adler, Jr. mengembangkan sensor yang teraktivasi saat pengemudi membunyikan klakson kendaraan. Pada saat yang hampir bersamaan, Henry A. Haugh mengembangkan sensor tekanan yang diletakkan di dalam jalan yang dilalui kendaraan. Metode ini digunakan selama hampir 30 tahun untuk mendeteksi keberadaan kendaraan maupun kepadatan lalu lintas. Untuk metode-metode lainnya, pengukuran kepadatan lalu lintas antara lain dilakukan dengan memanfaatkan:

- Suara (dengan sensor akustik)
- *Opacity* (dengan sensor optik, inframerah, dan pemroses gambar video)
- *Geomagnetism* (dengan sensor magnet, magnetometer)
- Refleksi dari energi yang dipancarkan (dengan radar laser inframerah, sensor ultrasonik, sensor radar gelombang mikro)
- Induksi elektromagnetik (dengan detektor *inductive-loop*)
- Getaran (dengan triboelektrik, seismik, dan sensor *inertia switch*).

Google Maps diluncurkan pertama kali pada tanggal 8 Februari 2005 (GMaps, 2020), ditandai dengan sebuah *blog post* sederhana dari Google sendiri (Taylor, 2005). Sebelumnya, aplikasi ini hanya berupa program komputer untuk *desktop* yang dikembangkan oleh Where 2 Technologies. Pada Oktober 2004, Google mengakuisisi perusahaan tersebut lalu program dikembangkan sebagai aplikasi web, seperti dapat

²⁷ Tampilan ini diambil dari akun Google Maps penulis

dilihat pada *Gambar 9.. Selanjutnya, pada tahun 2007, Google merilis fitur yang menampilkan informasi kepadatan lalu lintas pada ruas-ruas jalan di lebih dari 30 kota besar di Amerika Serikat (Wang, 2007). Pada saat fitur tersebut dirilis, Google tidak memberikan informasi bagaimana mereka mendapatkan informasi kepadatan lalu lintas untuk ditampilkan. Walaupun begitu, salah satu alternatif yang mungkin adalah kerja sama dengan pemerintah setempat, mengingat Federal Highway Administration sudah melakukan pengukuran kepadatan lalu lintas sejak lama dan tentu saja memiliki datanya.*



Gambar 9.2. Tampilan Google Maps pada tahun 2005 (dari <http://digital-archaeology.org>).

Pada tahun 2009, Google mengumumkan bahwa mereka menggunakan cara baru untuk mendapatkan informasi kepadatan lalu lintas, yaitu dengan mengumpulkan informasi dari pengguna aplikasi ponsel pintar Google Maps yang menyalakan fitur "My Location" (Barth, 2009). Secara sederhana dan seperti dijelaskan pada blog tersebut, teknik pengumpulan data tersebut dapat dijelaskan sebagai berikut: Setiap ponsel yang digunakan seorang pengendara mengirimkan informasi kecepatan berkendara kepada pusat data Google yang memiliki beribu-beribu komputer server (Google memiliki pusat-pusat data di beberapa negara, sebagai contoh pada *Gambar 9.4* diberikan foto pusat data di Belgia).

Pada tahun 2020 ini, penghuni bumi sudah lebih dari 7 miliar orang. Jika 10% dari jumlah penduduk tersebut memiliki ponsel pintar dan menyalakan fitur My Location, maka terdapat lebih dari 700 juta ponsel yang mengirimkan kecepatan dan lokasi ponsel itu, dari berbagai ruas-ruas jalan di banyak negara, ke server Google! Data dari jutaan ponsel tersebut dikirimkan secara real time (waktu nyata) ketika penggunanya bergerak, maka dapat dibayangkan bahwa data yang dikirimkan ke server Google "terus mengalir" dari waktu ke waktu dari ratusan juta ponsel. Data yang dikirimkan tersebut tentu saja dianonimisasi untuk menjaga privasi penggunanya. Data yang mengalir dengan kecepatan tinggi ini termasuk "big data stream" (bahasan tentang ini dapat dilihat pada Bab 10) dan membutuhkan teknologi

khusus (dengan menggunakan beribu-ribu komputer server), untuk menanganinya. Selanjutnya, data kecepatan dari seluruh ponsel dianalisis dengan memanfaatkan algoritma khusus untuk mengangani big data stream, sehingga untuk tiap ruas jalan tertentu (di dunia!), dapat dihitung kecepatan rata-ratanya (lihat *Gambar 9.*). Google lalu menampilkan informasi kepadatan lalu lintas secara “instant” di ruas-ruas jalan di berbagai negara kepada penggunanya, seperti dapat dilihat pada *Gambar 9.*

Teknik pengumpulan data kecepatan dari berbagai ponsel tersebut termasuk “urun daya” (*crowdsourcing*). Sebagaimana didefinisikan Wikipedia, *crowdsourcing* merupakan “proses untuk memperoleh layanan, ide, maupun konten tertentu dengan cara meminta bantuan dari orang lain secara massal, secara khusus melalui komunitas daring” (Crowd, 2018).



Gambar 9.3. Ilustrasi pengukuran kecepatan oleh Google Maps.



Gambar 9.4. Rak-rak berisi ribuan komputer server di pusat data Google di Belgia²⁸.



Gambar 9.5. Aplikasi Ponsel Google Maps di tahun 2009²⁹.

²⁸ <https://www.google.com/about/datacenters/gallery/>

²⁹ <https://googleblog.blogspot.com>

Pertanyaan yang menurut penulis cukup menarik adalah: Bagaimana Google menjaga agar cukup banyak pengguna Google Maps dengan sukarela berkontribusi ke data kepadatan tersebut? Perlu ada insentif bagi pengguna untuk melakukannya, dan Google berusaha memudahkan hal tersebut. Pada tahun 2014, Google mengumumkan bahwa mereka akan meluncurkan ponsel pintar Android versi murah dengan nama Android One (Pichai, 2014). Dengan harga yang dijaga di bawah USD 100, ponsel ini sangat terjangkau bagi kalangan menengah ke bawah, hanya sedikit *upgrade* dari ponsel jenis non-pintar (*featured phone*). Belum lagi Google Maps yang disediakan secara gratis. Tanpa disadari para pemilik ponsel tersebut telah “membayar” harga murah tadi dengan data.

Sistem Google dalam menentukan kepadatan lalu lintas di atas masih memiliki kelemahan. Sebuah cerita menarik yang menunjukkan kelemahan itu: Pada tahun 2020 seorang seniman bernama Simon Weckert “mengelabui” sistem Google Maps ini, dengan bermodalkan 99 ponsel dan kereta kecil (Weckert, 2020). Beliau menaruh 99 ponsel tersebut ke dalam kereta kecil, masing-masing menjalankan aplikasi Google Maps seperti dapat dilihat pada Gambar 9.6 and Gambar 9.7. Kemudian, kereta tersebut ditarik sambil berjalan kaki melewati sebuah jalan kecil yang relatif sepi di Berlin. Server Google mengira kecepatan berjalan yang relatif perlahan tersebut mewakili kecepatan berkendara, sehingga menyimpulkan bahwa di jalan sepi tersebut sebenarnya terjadi kemacetan. Walaupun terdengar lucu dan sederhana, implikasinya bisa bermacam-macam, apalagi jika dimanfaatkan oleh orang yang tidak bertanggung jawab. Fitur “Directions” yang dimiliki Google Maps secara bawaan menghindari jalan dengan kepadatan lalu lintas yang tinggi, sehingga dapat dipengaruhi juga untuk menghindari jalan-jalan yang secara spesifik “diakali” tersebut.



Gambar 9.6. Kereta dengan 99 ponsel yang menjalankan aplikasi Google Maps³⁰.

³⁰ <http://www.simonweckert.com>

Kesalahan Google dalam memberikan informasi kepadatan lalu lintas tersebut dapat terjadi, karena Google memberikan kepercayaan penuh kepada penggunanya yang relatif anonim sebagai kontributor data. Di satu sisi, metode ini mampu mengumpulkan sampel dalam jumlah besar. Di sisi lain, kejujuran dari setiap pengguna berpengaruh ke kualitas prediksi. Bagaimana jika seluruh kontributor data bisa dipercaya? Hal inilah yang sepertinya dimanfaatkan oleh *platform* Trafi, mitra resmi dari Jakarta Smart City (Trafic, 2017). Menurut informasi pada situs web Trafic, prediksi kemacetan didapatkan dari kecepatan armada transportasi public (Trafic, 2020). Dari sisi jumlah sampel, tentu saja jauh di bawah pengguna Google Maps. Namun, validitas data yang dikirimkan armada transportasi publik juga lebih bisa dipercaya dibandingkan dengan pengguna yang anonim.



Gambar 9.7. "Kemacetan" yang ditimbulkan oleh 99 ponsel (kiri), serta foto seniman tersebut bersama ponselnya³¹.

9.3. Pemanfaatan Google Traffic untuk Penentuan Waktu Pergi dan Pulang

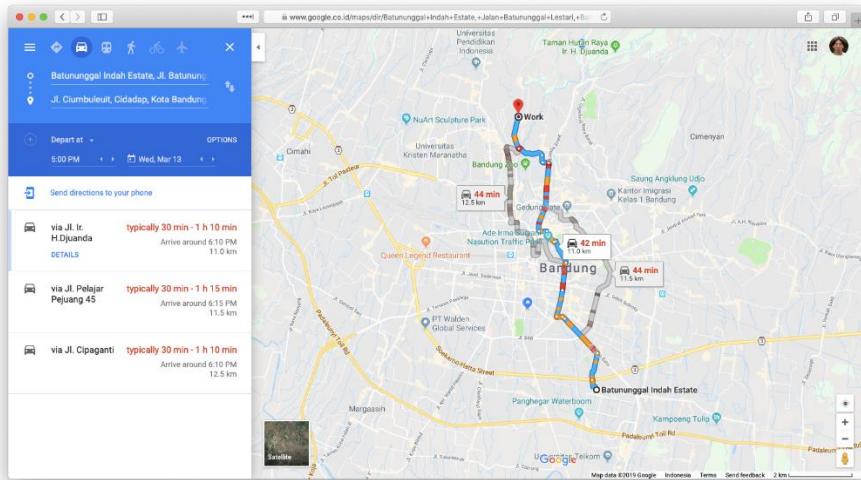
Mengumpulkan data kepadatan lalu lintas, mengolahnya lalu memberikan hasilnya kepada kita adalah pekerjaan Google Traffic. Sebagai pengguna, apa yang bisa kita manfaatkan dari sana? Salah satunya tentu saja dengan menggunakan fitur dasar yang sudah tertanam di aplikasi tersebut, misalnya untuk mencari jalur tercepat dari satu lokasi ke lokasi lain. Namun, lebih dari itu, kita juga bisa praktik menjadi *data scientist* amatir, dengan melakukan sedikit praktik pengumpulan dan analisis data untuk membantu kita lebih lanjut: menentukan waktu yang tepat untuk berangkat.

Sebagian besar dari kita memiliki rutinitas bepergian ke luar rumah di pagi hari, dan pulang kembali ke rumah di siang atau sore hari. Dengan banyaknya pengguna jalan yang memiliki rutinitas yang sama, akan

³¹ <http://www.simonweckert.com>

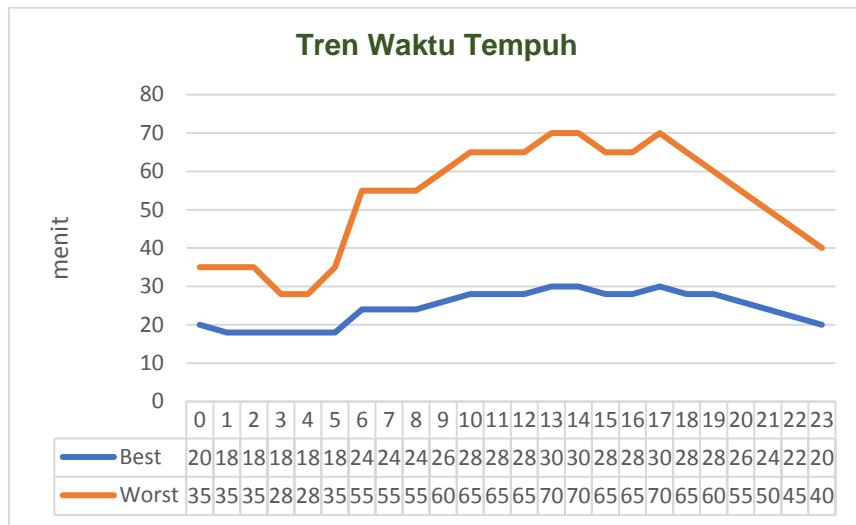
ada jam-jam di mana kepadatan terjadi, dan menambah waktu yang terbuang di perjalanan. Lalu, kapan sebenarnya waktu yang tepat untuk memulai perjalanan pergi ataupun pulang ke rumah?

Jawaban atas pertanyaan tersebut, tentu saja berbeda untuk setiap individu, karena bergantung pada posisi rumah serta tempat kerja atau sekolah yang dituju. Walau begitu, jika pembaca tinggal di kota besar, kemungkinan jawabannya bisa dicari dengan memanfaatkan data yang dimiliki oleh Google Maps, ditambah sedikit teknik pengolahan data.



Gambar 9.8. Antarmuka situs web Google Maps.

Untuk mendapatkan jawaban dari pertanyaan di atas, caranya cukup mudah, yaitu: Pengguna mengakses Google Maps versi web <https://www.google.com/maps>, memilih asal dan tujuan, serta memilih waktu keberangkatan seperti tangkap layar pada Gambar 9.8. Kemudian, langkah tersebut diulangi sebanyak 24 kali dalam sehari, data diambil setiap jam. Waktu tempuh dicatat. Berdasarkan data yang dikumpulkan, pembaca akan dapat membuat grafik tren waktu tempuh yang berguna untuk menentukan kapan waktu terbaik untuk berangkat ataupun. Sebagai contoh, grafik pada Gambar 9.9 menunjukkan tren waktu tempuh dalam satu hari penuh dari sebuah kompleks perumahan di selatan kota Bandung, ke kampus UNPAR yang berada di Jalan Ciumbuleuit. Dari grafik tersebut, kita dapat menyimpulkan atau mendapatkan *insights* dari data, bahwa waktu terbaik untuk berangkat ke UNPAR di pagi hari adalah sekitar pukul 4-5 pagi, sebelum waktu tempuh mulai bertambah. Kalau pembaca tidak memiliki masalah untuk bangun pagi, temuan tersebut lalu dapat dimanfaatkan.

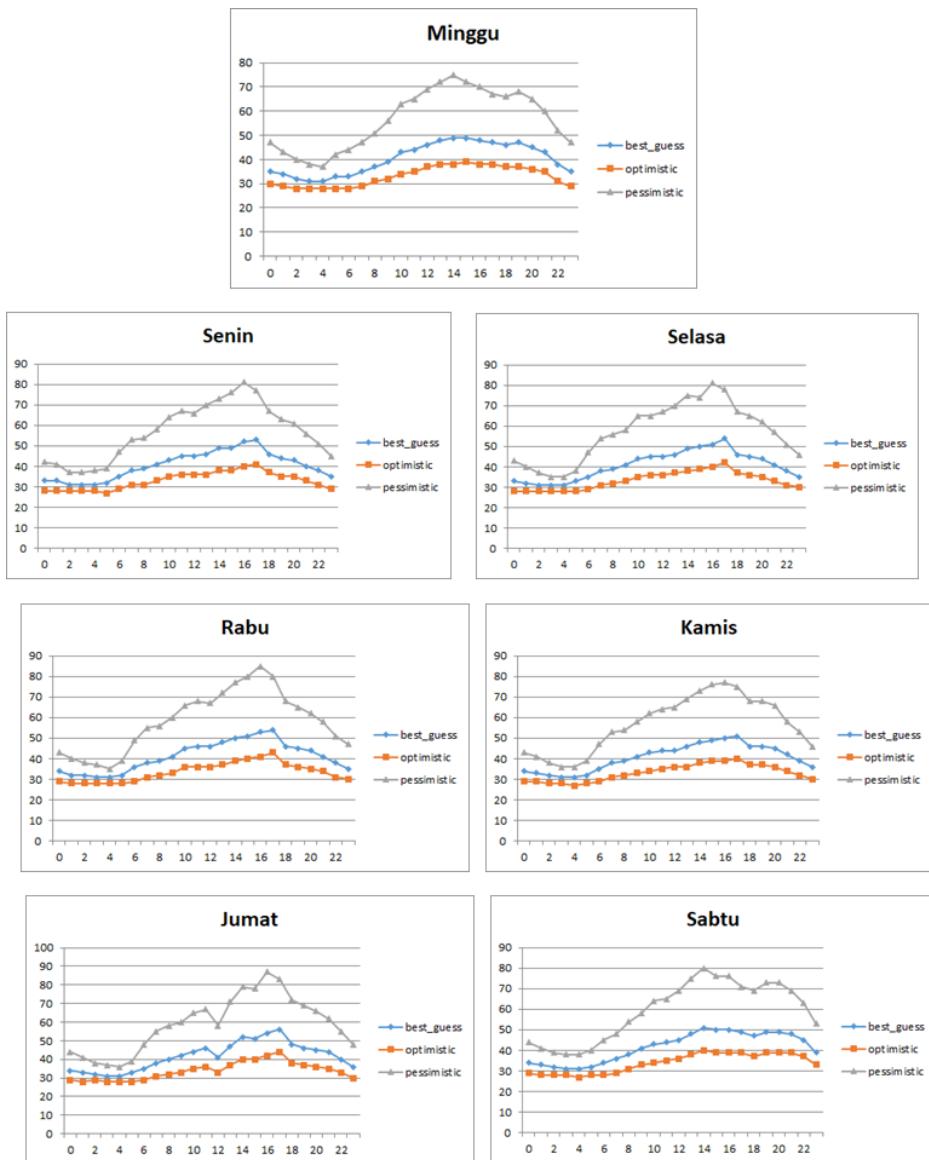


Gambar 9.9. Grafik waktu tempuh dalam rentang 24 jam.

Namun jika kita ingin mendapatkan tren waktu tempuh tersebut untuk 7 hari dalam seminggu, dan mengumpulkan data waktu tempuh secara manual, maka cara tersebut tidak lagi masuk akal untuk dilakukan. Pada tahun 2018, seorang mahasiswa Teknik Informatika UNPAR, Frasetiawan Hidayat, mengerjakan penelitian pada skripsinya, untuk mengotomatisasi langkah-langkah tersebut. Pada penelitiannya, data diambil melalui Google Directions API (GRoutes, 2020) dengan menggunakan program komputer. Eksperimen dilakukan untuk mendapatkan waktu tempuh antara dua titik lokasi selama 24 jam per hari, dan 7 hari seminggu. Hasilnya lalu dibuat visualisasi dalam bentuk grafik-grafik tren yang ditunjukkan pada Gambar 9.10.

Dengan representasi visual pada Gambar 9.10, kita akan lebih mudah untuk mendapatkan *insights* dari data, berupa waktu terbaik untuk pergi tiap hari, mulai hari Minggu sampai Sabtu. Contohnya, jika pembaca perhatikan pada hari Sabtu dan Minggu, waktu tempuh tidak cepat turun setelah sekitar jam 18.00. Hal ini berbeda dengan hari Senin sampai Jumat, yang grafiknya relatif turun setelah jam 18.00. Ini menunjukkan peningkatan aktivitas pengguna jalan, yang menghabiskan akhir minggunya (sampai petang hari) di luar rumah³². Masih ada satu lagi hal yang menurut penulis menarik pada grafik di atas. Di hari Jumat, terdapat penurunan yang signifikan di tengah hari, yang tidak terjadi pada hari-hari lain. Dapatkan pembaca menjelaskan mengapa demikian?

³² Pengambilan sampel dilakukan sebelum masa Pembatasan Sosial Berskala Besar, yang mengurangi kepadatan lalu lintas secara signifikan.



Gambar 9.10. Grafik waktu tempuh 7 hari x 24 jam.

Referensi

- (Barth, 2009) <https://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html> (diakses 16 Juni 2020).
- (Crowd, 2018) https://id.wikipedia.org/wiki/Urun_daya (diakses 17 Juni 2020).
- (FHWA, 2006) <https://www.fhwa.dot.gov/publications/research/operations/its/06108/01.cfm> (diakses 16 Juni 2020).
- (GMaps, 2020) https://en.wikipedia.org/wiki/Google_Maps (diakses 16 Juni 2020)
- (GRoutes, 2020) <https://cloud.google.com/maps-platform/routes/> (diakses 16 Juni 2020).
- (Pichai, 2014) <https://googleblog.blogspot.com/2014/06/google-io-2014-keynote.html> (diakses 17 Juni 2020).
- (Taylor, 2005) <https://googleblog.blogspot.com/2005/02/mapping-your-way.html> (diakses 16 Juni 2020)
- (Trafi, 2017) <https://smartcity.jakarta.go.id/blog/180/trafi-aplikasi-mitra-jakarta-smart-city> (diakses 16 Juni 2020).
- (Trafi, 2020) <https://info.trafi.com/site/platform> (diakses 16 Juni 2020).
- (Wang, 2007) <https://googleblog.blogspot.com/2007/02/stuck-in-traffic.html> (diakses 16 Juni 2020).
- (Weckert, 2020) <http://www.simonweckert.com/googlemapshacks.html> (diakses 16 Juni 2020).

Halaman ini sengaja dikosongkan

Bagian Kedua

Paparan Teknis

Halaman ini sengaja dikosongkan

Bab 10 Teknologi Big Data

Oleh:

Gede Karya

10.1. Pendahuluan

Pada Bab 1 telah disampaikan bahwa *data scientist* harus menguasai teknologi-teknologi yang dibutuhkan. Salah satu teknologi yang penting dikuasai adalah teknologi *big data*, jika tugas data scientist termasuk menganalisis data yang dapat dikategorikan sebagai *big data*.

Sebelum dibahas teknologi *big data*, pada bagian awal akan diulas tentang seputar *big data*, yang mencakup definisi, ciri, masalah dan gambaran teknologi sebagai solusinya. Setelah itu pada subbab berikutnya dibahas teknologi big data dengan detil, mulai dari arsitektur teknologi, dan teknologi yang berbasis *free open source software* (FOSS). Paparan FOSS melingkup ekosistem Hadoop (*Hadoop ecosystem*) dan penjelasan komponen dari ekosistem tersebut, seperti HDFS, MapReduce, HBase, Hive, Spark, Sqoop, Flume, Kafka dan R. Agar lebih lengkap, maka dibahas juga teknologi yang sifatnya komersial dan berbasis *cloud computing*. Pada subbab terakhir juga dibahas contoh penggunaan teknologi *big data* pada kasus sistem anti hoax pada situs www.antihoax.id.

10.2. Seputar Big Data

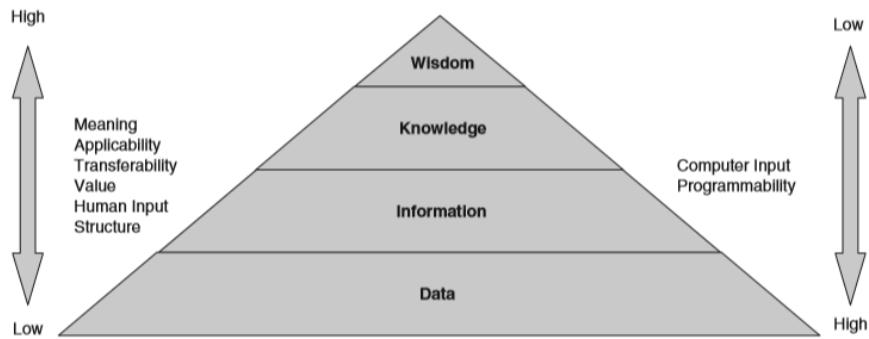
Apa itu big data?

Dari berbagai pendapat yang dikumpulkan pada penelitian (De Mauro dkk., 2016) *big data* didefinisikan sebagai aset informasi yang dicirikan oleh karakteristik 3v, yaitu: *volume*, *velocity* dan *variety* yang tinggi, yang memerlukan metode dan teknologi tertentu untuk memprosesnya menjadi pengetahuan (*knowledge*) yang bernilai (*value*) dalam pengambilan keputusan. Aset informasi bermakna penting, karena data dianggap memiliki nilai yang tinggi bagi organisasi seperti aset lain (mesin, material, orang, modal, dan metode) dan dapat divaluasi (dinalai dalam satuan uang).

Mengapa fenomena big data berkembang pesat?

Konsep hirarki *data-information-knowledge-wisdom* (DIKW) (Rowley, 2007) atau sering disebut sebagai *wisdom hierarchy* (Gambar 10.1) memberikan alasan yang masuk akal mengapa fenomena *big data* begitu berkembang. Dengan besarnya potensi data yang ada saat ini dan di masa depan, maka besar

juga potensi informasi yang tersedia untuk ditransformasi menjadi pengetahuan (*knowledge*) sehingga dapat mengoptimalkan pengambilan keputusan (*wisdom*).



Gambar 10.1. Hirarki Wisdom (Rowley, 2007).

Dengan demikian, jika *big data* dapat ditangani dengan baik akan memberikan manfaat besar bagi organisasi, khususnya semakin bijaksana dalam mengambil keputusan yang didasarkan atas data (bersifat *data driven*), sehingga lincah dalam mengambil keputusan dalam perubahan kondisi lingkungan yang cepat berubah.

Apa saja ciri dan batasan big data?

Big data dicirikan dengan karakteristik *big data* menurut (De Mauro dkk., 2016) adalah 3v yaitu: *volume*, *velocity* dan *variety* yang tinggi. Secara umum batasan tinggi dalam konteks *big data* mengikuti hukum Moore (Moore, 2006). Namun demikian, saat ini karakteristik *big data* digambarkan seperti pada Gambar 10.2.

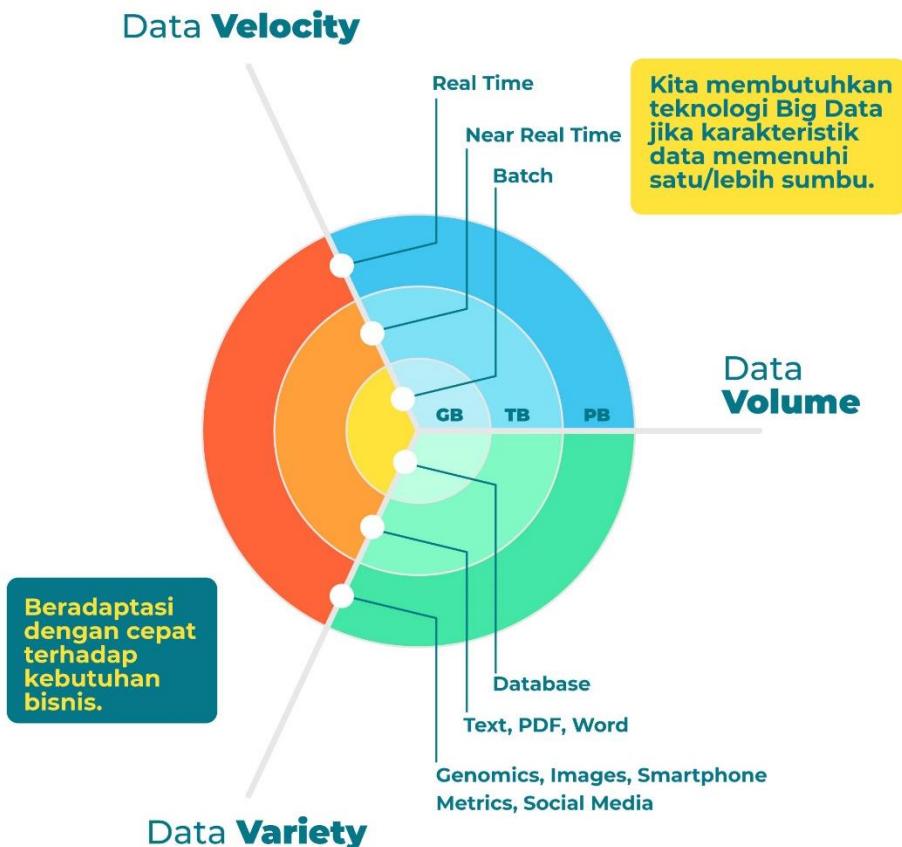


Gambar 10.2. Karakteristik big data.

Big data memiliki karakteristik *volume* yang tinggi, dari *terabytes* ke *zettabytes*. Hal ini berkonsekuensi pada kapasitas penyimpanan dan kapasitas pemrosesan data yang tidak dapat ditangani oleh metode dan teknologi informasi konvensional saat ini. Metode dan teknik penyimpanan yang diterapkan hingga saat

ini mengarah pada pemrosesan secara paralel pada lingkungan sistem terdistribusi, baik dari sisi media penyimpanan maupun pemrosesannya. Karakteristik *big data* lebih detail dapat dilihat pada Gambar 10.3.

Karakteristik *velocity* pada *big data* mengubah sudut pandang pemrosesan data secara *batch*, menjadi pemrosesan data secara dinamis. Dengan demikian data tidak lagi dilihat secara statis, namun secara dinamis sebagai *stream*. Selain sebagai *data stream*, *big data* juga berkaitan dengan pergerakan data dalam jumlah besar (*high volume movement*) seperti data spasial, citra, dan lainnya.

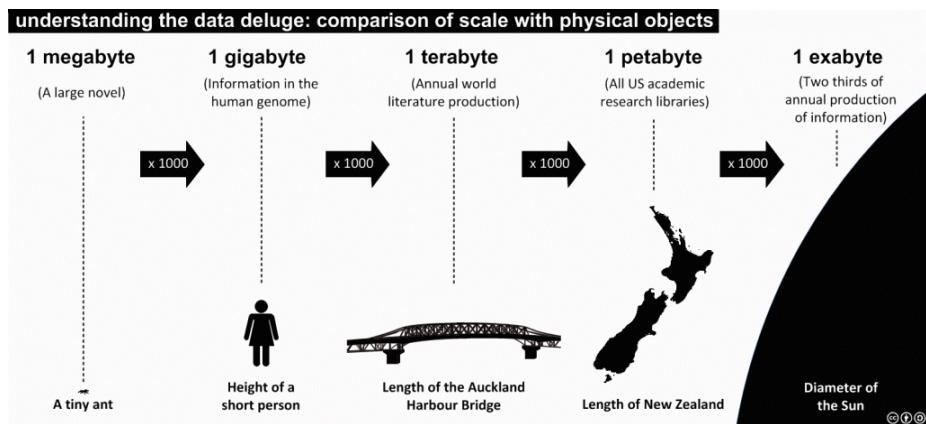


Gambar 10.3. Karakteristik detail big data.

Big data bersumber dari berbagai *event*. Semua kegiatan kita yang menggunakan komputer, gadget, sensor dan peralatan lainnya menghasilkan *big data*. Selain sumber yang beraneka ragam, dari sisi struktur juga beraneka ragam, mulai dari yang terstruktur, seperti: data transaksi (pasar uang, e-commerce, dll), semi terstruktur, maupun yang tidak terstruktur, seperti: image, text opini pada media sosial maupun halaman web di internet. Untuk itu diperlukan metode dan teknologi untuk mengintegrasikan *big data* dari berbagai sumber dan dari format yang berbeda-beda tersebut.

Apa masalah utama dari big data?

Masalah utama *big data* dikenal dengan istilah fenomena *data deluge*, suatu fenomena dimana laju pertumbuhan data lebih tinggi dari pada laju kemampuan memproses dan menganalisis data suatu organisasi. Pada Gambar 10.4 dapat dilihat besarnya volume data dibandingkan dengan objek fisik. Oleh karena itu dalam memproses dan menganalisis data, kita memerlukan teknologi yang tidak konvensional lagi. Kita memerlukan teknologi yang dapat mengimbangi laju pertumbuhan data yang meningkat seiring dengan waktu dan peningkatan penggunaan teknologi informasi dan komunikasi.



Gambar 10.4 Fenomena Data Deluge³³

Jika data diibaratkan seperti hujan lebat (Gambar 10.5), maka kita bisa menangkap air dengan laju yang sesuai, kemudian mengumpulkannya untuk menyiram tanaman. Untuk menangkap semua data, tentu memerlukan banyak, yang diumpamakan dengan banyak payung atau payung yang sangat besar.

³³ [https://ritholtz.com/2016/09/162347/Understanding the Data Deluge: Comparison of Scale with Physical Objects](https://ritholtz.com/2016/09/162347/Understanding%20the%20Data%20Deluge%3A%20Comparison%20of%20Scale%20with%20Physical%20Objects)



Gambar 10.5. Fenomena Data Deluge³⁴.

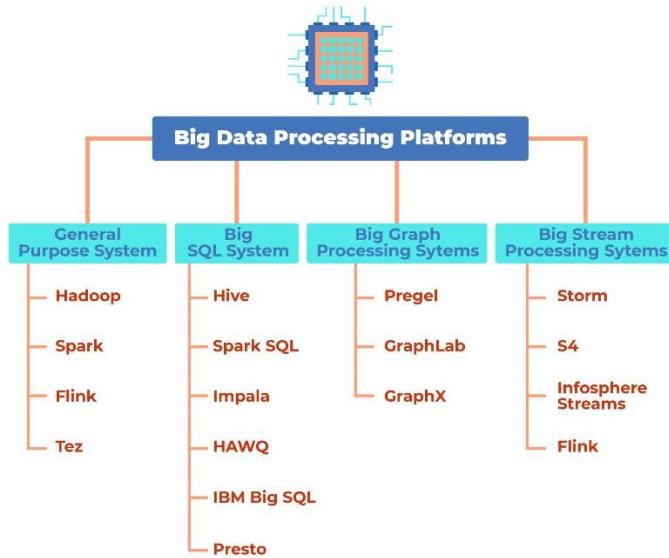
Apa itu teknologi big data?

Teknologi *big data* merupakan teknologi khusus yang diperuntukkan untuk menangani masalah *big data*. Untuk menangani masalah volume, teknologi *big data* menggunakan teknik penyimpanan dan pemrosesan data terdistribusi. Masalah *velocity* ditangani dengan menggunakan pemrosesan stream dan terdistribusi. Sedangkan masalah *variety* ditangani menggunakan teknik integrasi data dan penyimpanan data tidak terstruktur (*on write*). Penentuan struktur dilakukan pada saat proses pembacaan data tersebut (*on read*). Pada Gambar 10.6 dapat kita lihat berbagai platform teknologi pemrosesan *big data* yang telah dikumpulkan pada penelitian (Bajaber dkk., 2016).

Berdasar platorm pemrosesan big data, teknologi tersebut dapat dikelompokkan menjadi (Gambar 10.6):

- umum (*general purpose*), seperti Hadoop, Spark, Flink, Tez
- pemroses query (*big SQL*), seperti: Hive, Spark SQL, Implala, HawQ, IBM Big SQL
- pemroses *big graph*, seperti: pregel, graphLab, GraphX, dan
- pemroses *big stream*, seperti: Storm, S4, Infosphere stream, flink, dan Spark Stream.

³⁴ <https://www.economist.com/leaders/2010/02/25/the-data-deluge>

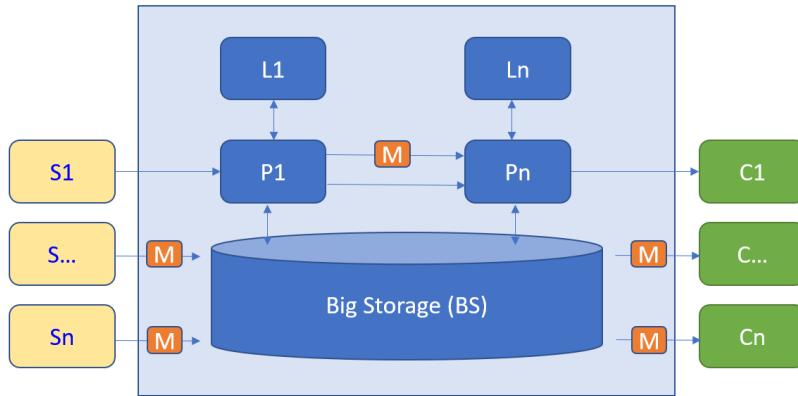


Gambar 10.6. Platform pemrosesan big data.

Teknologi *big data* yang populer digunakan saat ini adalah teknologi Hadoop. Hadoop dikembangkan pada awalnya oleh Google (Ghemawat dkk., 2003), kemudian menjadi proyek Apache yang berdiri sendiri. Prinsip utama dari teknologi Hadoop adalah penyimpanan dan pemrosesan terdistribusi pada komputer-komputer komoditas yang terhubung dalam jaringan (sering disebut *cluster*). Inti dari teknologi Hadoop adalah *Hadoop Distributed File System* (HDFS) untuk menangani penyimpanan data terdistribusi dan Map Reduce untuk pemrosesan data terdistribusikan yang dilakukan pada komputer (*node of cluster*) tempat data disimpan. Untuk menyelesaikan berbagai persoalan komputasi, Hadoop didukung oleh berbagai teknologi yang secara keseluruhan sering disebut sebagai Hadoop *ecosystem*.

10.3. Arsitektur Teknologi Big Data

Untuk mengatasi masalah *big data*, maka teknologi *big data* bukanlah suatu teknologi tunggal. Oleh karena itu, kita perlu melihatnya secara menyeluruh dan secara detail melalui arsitektur teknologi *big data* seperti pada Gambar 10.7. Arsitektur teknologi *big data* menggambarkan komponen-komponen yang diperlukan, dan interaksi antar komponen sehingga sistem *big data* secara keseluruhan dapat berjalan.



Gambar 10.7. Arsitektur teknologi big data.

Pada Gambar 10.7, S1 sampai Sn adalah sumber (*source*) dari *big data*, seperti sensor, aplikasi media sosial (Twitter stream, Facebook, Instagram, dll), web berita (news), satelit, basis data, dan sumber lainnya. Kita memerlukan teknologi untuk mengumpulkan data dari sumber *big data*.

Pengumpulan data dapat dilakukan dengan 2 cara, yaitu:

- menggunakan aplikasi yang dibuat khusus untuk memproses *big data* (*process* – P1 sampai Pn), misalnya: untuk mengambil data Twitter, kita dapat membuat aplikasi penangkap data Twitter (*Twitter stream capture*), atau
- menggunakan aplikasi yang sudah jadi sebagai mediator (*middleware* - M).

Pemroses (P) terdiri atas 2 jenis atas dasar tingkat interaktifitas pemrosesan dalam menghasilkan output, yaitu: *batch processing* dan *online (real time) processing*.

Sedangkan jika dilihat dari data yang diproses dapat dikelompokkan menjadi 2 juga, yaitu *data in rest* (pemrosesan terhadap data yang sudah tersimpan di media penyimpanan) dan *data in motion (stream) processing* (pemrosesan terhadap data yang belum tersimpan pada media penyimpanan).

Pemrosesan terhadap *data stream* berasosiasi dengan *online/real time processing*, sedangkan pemrosesan terhadap data *in rest* berasosiasi dengan pemrosesan *batch*. Untuk mengembangkan aplikasi pemroses *big data* (P) diperlukan pustaka teknologi (*library* – L1 sampai Ln) dari platform teknologi pemrosesan *big data* (yang telah dibahas di bagian 2). Komponen pemroses (P) atau mediator (M) berkomunikasi dengan media penyimpanan (*big storage* - BS). Media penyimpanan digunakan untuk menyimpan data mentah (*raw data*) maupun data hasil pengolahan/pemrosesan (*result*). Hasil pengolahan/pemrosesan tersebut perlu dikirim ke konsumen (*consument* – C1 sampai Cn) baik melalui pemroses secara langsung (P) maupun melalui mediator (M) secara berkala sebagai proses propagasi atau sinkronisasi. Contoh dari aplikasi C adalah web site tempat publikasi, seperti pada web anti *hoax* yang dibahas sebagai contoh kasus pada subbab 10.6. M juga dapat digunakan sebagai mediator komunikasi antar aplikasi pemroses (P).

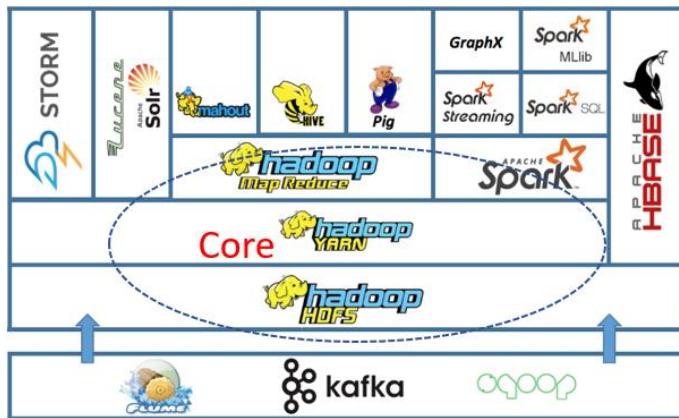
Pada proses pengambilan data dari S di atas, karena sumbernya banyak dengan format yang beranekaragam, maka teknologi yang diperlukan tidak hanya membaca/mengambil (*capture*), namun juga mengintegrasikan (*integrator*) baik melalui perubahan format, jenis, pewaktuan dan pra olah lainnya, sehingga dapat dihasilkan data yang siap untuk diproses lebih lanjut agar menjadi pengetahuan (*knowledge*).

Contoh teknologi yang berperan pada komponen arsitektur pada Gambar 7 lebih lanjut dijelaskan pada bagian 4 ekosistem Hadoop.

10.4. Ekosistem Hadoop

Seperti telah dijelaskan pada subbab 10.2, teknologi *big data* yang sangat populer saat ini adalah teknologi Hadoop. Kepopuleran Hadoop didukung oleh ketersediaanya sebagai proyek *Free Open Source Software* (FOSS) yang dikelola oleh Apache, sehingga tersedia bagi semua kalangan dan mendapat masukan dari para *developer* di seluruh dunia.

Kumpulan teknologi yang bekerja di atas platform teknologi Hadoop sering disebut sebagai Ekosistem Hadoop (*Hadoop Ecosystem*). Teknologi ini saling bekerjasama untuk memberikan solusi pemrosesan *big data*. Ekosistem Hadoop secara skematis dapat dilihat pada Gambar 10.8.



Gambar 10.8. Ekosistem Hadoop³⁵

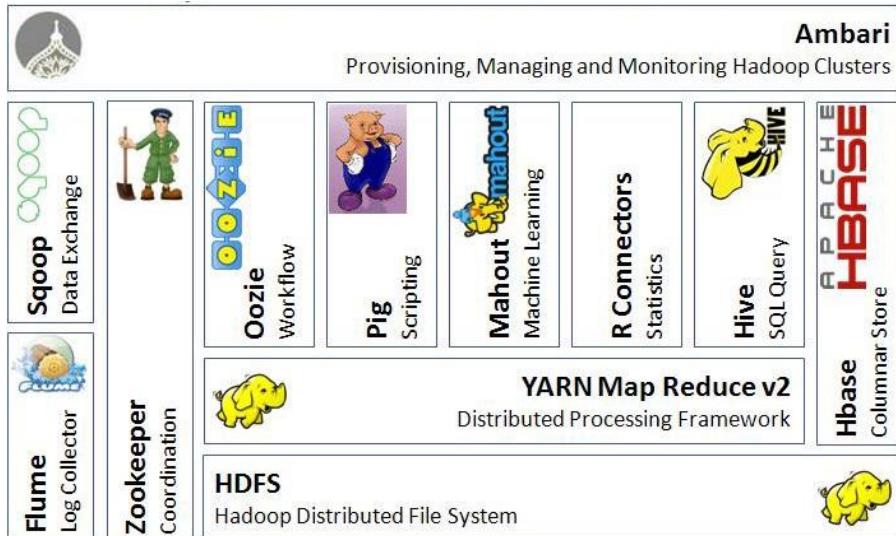
Banyak pihak menggambarkan ekosistem Hadoop secara berbeda untuk berbagai penekanan, seperti pada Gambar 10.9 dan 10.10.

Pada Gambar 10.8 dapat dilihat bahwa komponen dasar dari ekosistem Hadoop adalah teknologi Hadoop. Hadoop menyediakan teknologi penyimpanan dan pemrosesan terdistribusi (*parallel*) pada komputer-

³⁵ <http://blog.newtechways.com/2017/10/apache-hadoop-ecosystem.html>

komputer komoditas yang terhubung dalam jaringan (sering disebut *cluster*). Inti (*core*) dari Hadoop adalah *Hadoop Distributed File System* (HDFS) untuk menangani penyimpanan data terdistribusi dan Map Reduce untuk pemrosesan data terdistribusi yang dilakukan pada komputer (*node of cluster*) tempat data disimpan, dan Yarn (*Yet Another Resource Negotiator*) untuk mengelola sumberdaya (*resources*) termasuk penjadwalan job (Holmes, 2012).

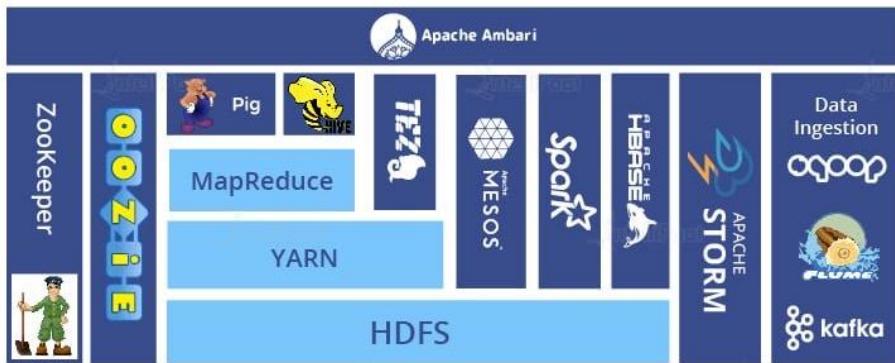
Untuk menyelesaikan berbagai persoalan komputasi, Hadoop didukung oleh berbagai teknologi yang berhubungan/ memanfaatkan teknologi inti tersebut, seperti Flume, Hbase, Hive, Zookeeper, R, Mahout, Pig, Oozie, Sqoop dan lainnya.



Gambar 10.9. Ekosistem Hadoop³⁶.

Dalam banyak kasus juga bekerjasama dalam satu ekosistem dengan teknologi-teknologi pemrosesan *big data* yang telah dijelaskan pada Gambar 10.6, seperti dengan Spark, Storm pada skema Gambar 10.10.

³⁶ <https://medium.com/@theinternetbae/big-data-dengan-hadoop-apache-hadoop-ecosystem-part-2-f01a47453cfb>



Gambar 10.10. Ekosistem Hadoop bersama Spark dan Storm³⁷.

Agar lebih generik, mari kita coba petakan ekosistem Hadoop atas dasar arsitektur teknologi *big data* pada Subbab 10.3. Yang termasuk katagori perantara atau *middleware* (M) adalah Flume, Sqoop, Kafka. Katagori pemroses atau *processor* (P) adalah Spark, R, Storm, Mahout. Sedangkan yang termasuk pada katagori pustaka atau *library* (L) adalah Spark MLLIB, Spark SQL, Spark Stream, dan sejenisnya. HDFS, Hive, Hbase termasuk pada katagori *big storage* (BS).

Beberapa teknologi kita bahas lebih detail terutama fungsinya dalam pemrosesan *big data*, antar lain: HDFS, Map Reduce, Hbase, Hive, Spark, Flume, Sqoop, R dan Kafka. Penjelasan dari teknologi lain dapat dilihat lebih jauh pada situs offisial masing-masing aplikasi.

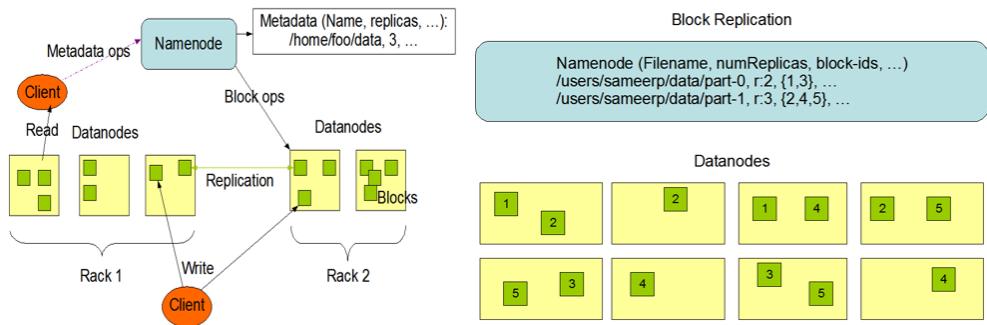
HDFS



HDFS merupakan komponen dari Apache Hadoop yang paling mendasar. HDFS singkatan Hadoop Distributed File System, merupakan sistem file terdistribusi.

Arsitektur HDFS dapat dilihat pada Gambar 10.11. Pada HDFS, data disimpan dalam bentuk file, kemudian setiap file dipecah menjadi potongan data dalam bentuk blok-blok yang ukurannya ditentukan (dari konfigurasi). Misalkan pada Gambar 10.11, sebuah file dipecah menjadi 5 blok (blok 1 sampai 5). HDFS terdiri atas 2 komponen, yaitu: Namenode yang bertugas untuk menyimpan meta data (berupa nama file, banyaknya replika, blok, lokasi penyimpanan blok) dan informasi lain tentang data), dan beberapa Datanode yang menyimpan blok data tersebut. Untuk menjaga ketersediaan data, maka suatu blok data dapat disimpan pada beberapa Datanode, misalnya blok 1 disimpan pada Datanode 1 dan 3.

³⁷ <https://intellipaat.com/blog/tutorial/hadoop-tutorial/hadoop-ecosystem/>



Gambar 10.11. Arsitektur HDFS³⁸.

Aplikasi yang ingin mengakses data yang disimpan pada HDFS (pada Gambar 10.11 disebut Client) terlebih dahulu menghubungi Namenode untuk mendapatkan informasi lokasi data, kemudian mengakses data tersebut langsung ke Datanode tempat data tersimpan. Hal ini juga berlaku dalam skenario penulisan dan pembacaan data.

Map Reduce



Map Reduce merupakan framework pemrograman terdistribusi yang dapat diterapkan terhadap data yang tersimpan pada HDFS. Pemrograman native pada Map Reduce menggunakan bahasa Java. Program dieksekusi pada Datanode dimana data yang diproses disimpan. Namun demikian Map Reduce hanya mendukung pemrograman paralel model batch (*Paralell Batch Processing*). Map Reduce membutuhkan *ResourceManager* yang bertugas mengelola sumberdaya (berpasangan dengan Namenode), dan *NodeManager* yang mengelola eksekusi pada setiap Datanode cluster (berpasangan dengan Datanode). Versi terakhir dari Map Reduce menggunakan Yarn sebagai *ResourceManager*.

HBase



HBase merupakan sistem basis data berbasis kolom (*columnar database system*). Pada basis data ini, data disimpan dalam bentuk pasangan *key-value*. Hbase berjalan di atas HDFS dan memiliki komponen pemroses terdistribusi, yaitu: Hbase Master (Hmaster) yang menyimpan meta data dan Regionserver yang menyimpan data nyata. Oleh karena itu, Hbase dapat menyimpan data dalam jumlah besar dengan akses yang cepat. Dengan model *key-value* (*columnar*) tersebut, maka Hbase dapat digunakan untuk menyimpan data yang strukturnya berbeda untuk setiap record. Model Hbase ini dikembangkan oleh Google menjadi BigTable pada Google Cloud Platform (GCP).

³⁸ <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>

Hive



Hive merupakan sistem pemroses data yang berjalan di atas Map Reduce namun memiliki antarmuka akses berbasis *Structure Query Language* (SQL). Data yang tersimpan pada HDFS maupun HBase dapat diakses menggunakan perintah-perintah SQL layaknya basis data relasional, dan memiliki antarmuka native pada pemrograman Java Map Reduce. Dengan kemampuan ini, Hive menjadi dasar implementasi Data Warehouse dan Data Mart di atas HDFS dan HBase.

Spark



Spark merupakan pemroses data berbasis memori. Jika Map Reduce cocok untuk pemrosesan batch, maka untuk pemrosesan online yang menggunakan iterasi berulang-ulang, Spark sangat cocok digunakan. Spark menjalankan aplikasi menggunakan *Directed Acyclic Graph* (DAG). DAG terdiri atas simpul (*vertices*) yang menyatakan *Resilient Distributed Dataset* (RDD) berupa koleksi berbentuk array yang terdistribusi, dan sisi (*edge*) yang menyatakan operasi yang diterapkan terhadap RDD tersebut. Spark juga dilengkapi dengan berbagai pustaka (library) untuk pemrosesan *big data*, di antaranya: Spark Stream (pemrosesan data stream), Spark MLLib (menyediakan algoritma Machine Learning seperti clustering, klasifikasi, dan rekomendasi), Spark GraphX (pemrosesan big graph) dan Spark SQL (untuk mengakses big data dengan perintah-perintah SQL).

Sqoop



Sqoop merupakan kakas (*tool*) yang sangat efisien untuk mentransfer data dari Hadoop (HDFS dan HBase) ke penyimpanan data terstruktur seperti basis data relasional (Relational Database Management System - RDBMS) dan sebaliknya (atau ekspor/ impor data Hadoop-RDBMS). Sqoop dapat mengakses RDBMS menggunakan antarmuka *Java Database Connectivity* (JDBC) sehingga dapat berhubungan dengan semua basis data yang menyediakan antarmuka JDBC tersebut.

Flume



Flume merupakan kakas (*tool*) yang menyediakan layanan pengumpulan (*collecting*), agregasi (*aggregating*), dan pemindahan (*moving*) data log sekala besar. Flume dapat berjalan secara terdistribusi, reliabel dan memiliki tingkat ketersediaan (*availability*) yang tinggi. Arsitektur Flume didasarkan pada *streaming data flow*, sehingga sangat mendukung aplikasi analisis online (*online analytical application*). Oleh karena itu, Flume juga cocok untuk mengumpulkan data stream dari berbagai sumber seperti: Twitter, Web News, dan sistem berbasis *Internet of Things* (IoT).

R Hadoop & Spark

 R merupakan bahasa pemrograman yang kaya akan fungsi dan pustaka (library) statistik. R sangat powerfull digunakan untuk analisis data berbasis statistik. R telah memiliki versi yang dapat berjalan di atas Hadoop (R-Hadoop) dan dapat berjalan memanfaatkan Spark (Spark-R). Dengan menggunakan R di atas Hadoop maupun Spark dapat memaksimalkan ketersediaan fungsi statistik dan kemampuan pemrosesan data sekala besar, baik batch processing maupun online iteratif.

Kafka

 Kafka merupakan platform *stream* yang memiliki kemampuan menangani pengelolaan distribusi pesan berbasis protokol *publish-subscribe* untuk *data stream* dalam skala besar. Kafka cocok digunakan untuk aplikasi yang memiliki aksi tertentu yang dipicu oleh event *data stream*. Kafka dapat bekerja dengan konfigurasi cluster beberapa komputer sehingga memiliki kemampuan menangani data stream berskala besar. Dengan demikian Kafka dapat digunakan untuk menangani koleksi dan respon terhadap data media sosial seperti Twitter dengan kecepatan dan volume yang besar.

10.5. Teknologi Big Data Komersial

Selain teknologi *open source* yang tergabung pada ekosistem Hadoop, telah tersedia beberapa platform teknologi komersial yang berbasis *cloud computing*. Beberapa teknologi tersebut dapat menjalankan ekosistem Hadoop di dalamnya baik secara langsung atau dengan modifikasi sehingga memiliki fungsi yang sejenis. Pada Gambar 10.12 dapat dilihat peta platform teknologi *cloud computing* berdasarkan survey Gartner tahun 2019 (*Gartner Magic Quadrant for Cloud Infrastructure as Services*).

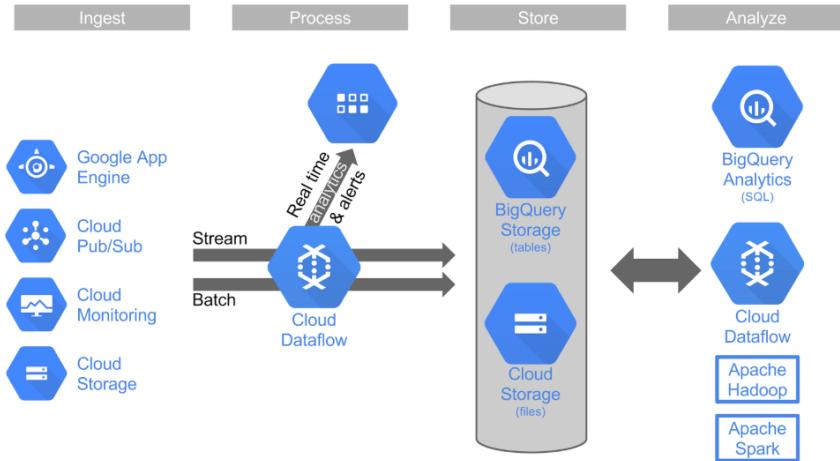


Gambar 10.12. Gartner Magic Quadrant for Cloud Infrastructure as Services 2019³⁹.

Pemain utama (*leader*) teknologi *big data* komersial tersebut adalah: (1) Amazon Web Service (AWS), Microsoft Azure, (3) Google Cloud Platform (GCP). Sedangkan pemain dengan kapabilitas menengah ke bawah (*nice player*) adalah IBM, Oracle dan Alibaba Cloud.

Pada Gambar 10.13 dapat dilihat bahwa Google Cloud Platform (GCP) memiliki produk-produk teknologi yang memiliki fungsi yang sama (bahkan nama lain saja) dengan ekosistem Hadoop. Seperti Cloud Storage (setara dengan HDFS), Compute Engine (setara Map Reduce). Sedangkan BigQuery Analytics (SQL) setara dengan Hive, dan BigQuery Storate setara Hbase. Demikian juga dengan Cloud Pub/Sub dan Data Flow secara Flume.

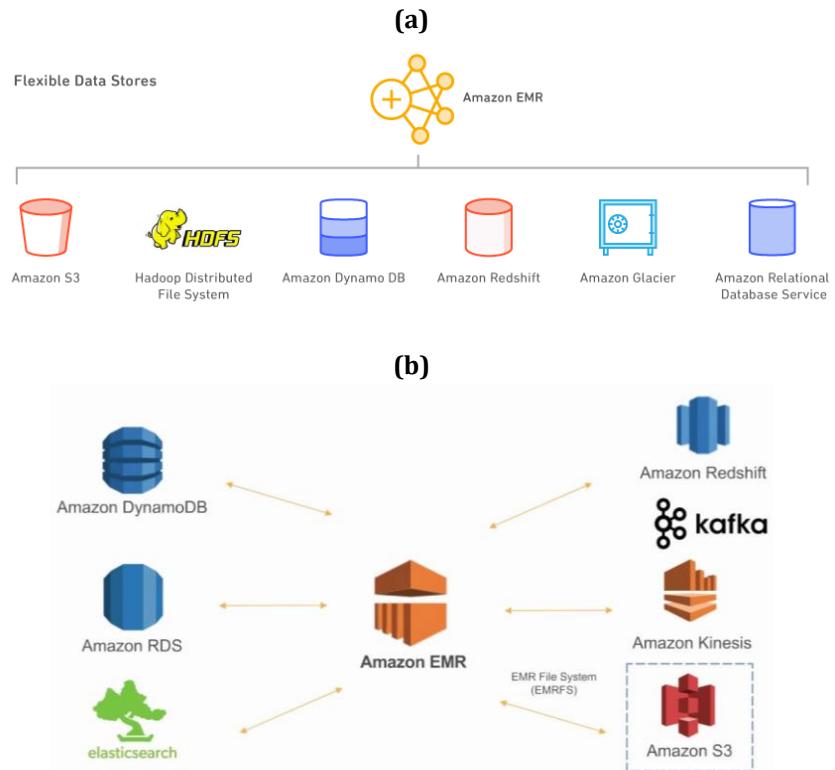
³⁹ https://blogs.gartner.com/olive-huang/files/2019/11/365830_0001.png



Gambar 10.13. Google Cloud Platform ⁴⁰.

Demikian pula dengan AWS, pada Gambar 10.14(a) dapat kita lihat juga bahwa AWS menggunakan teknologi *big data* pada ekosistem Hadoop, sedangkan pada Gambar 10.14(b) ditunjukkan hubungannya dengan Kafka.

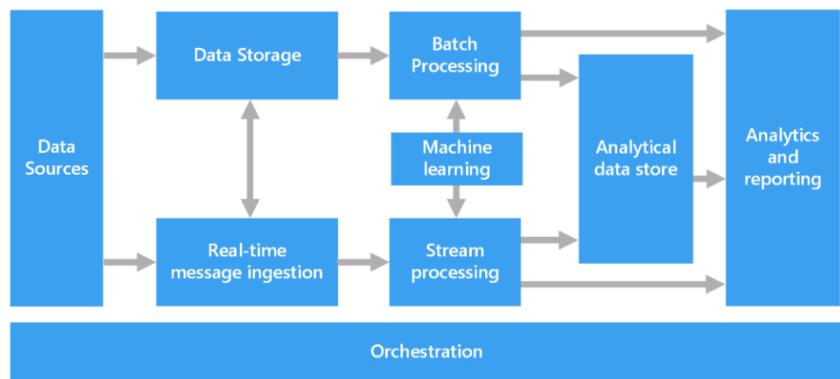
⁴⁰ <https://cloudplatform.googleblog.com/2015/04/big-data-cloud-way.html>



Gambar 10.14 Amazon Web Service (AWS)⁴¹.

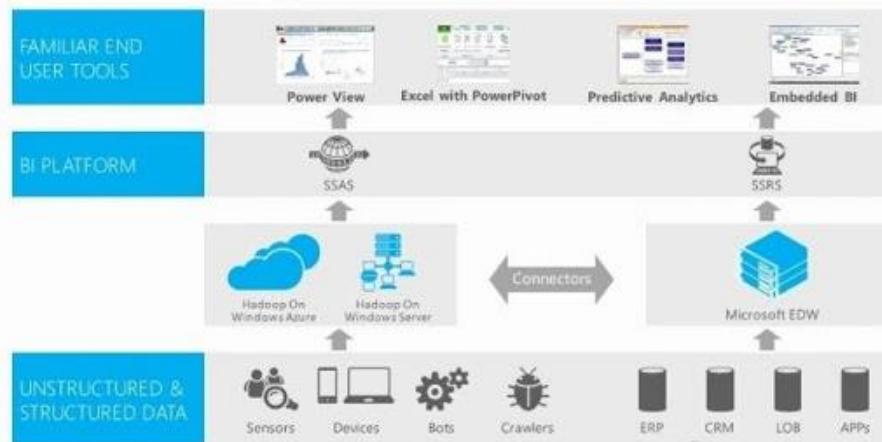
Pada Gambar 10.15 juga dapat dilihat secara sekamatis arsitektur *big data* dari Microsoft, Azure.

⁴¹ <https://aws.amazon.com/blogs/big-data/>



Gambar 10.15. Arsitektur Big data - Azure⁴².

Lebih lanjut solusi *big data* dalam sistem Microsoft Azure dapat dilihat pada Gambar 10.16. Pada gambar tersebut dapat dilihat bahwa solusi andalannya juga berupa Hadoop yang berjalan di dalam cloud platform Azure.



Gambar 10.16. Solusi Big data Microsoft⁴³.

Demikianlah penjelasan tentang teknologi *big data* berbasis cloud computig yang bersifat komersial.

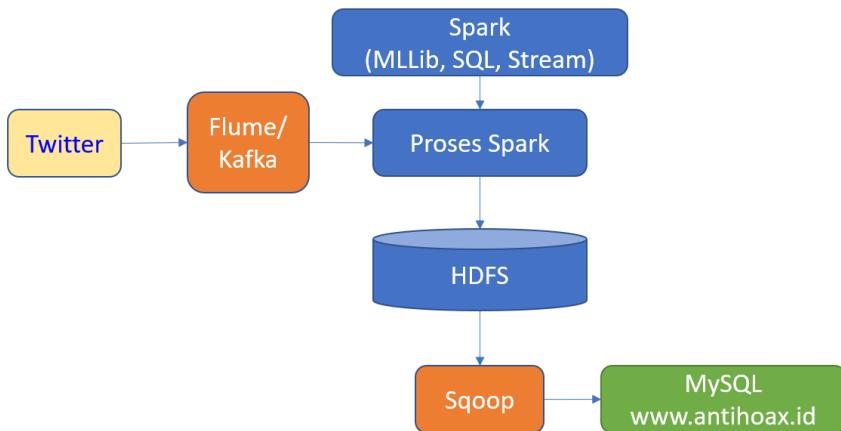
⁴² <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/>

⁴³ <http://princetondatalabs.com/intro-to-microsoft-azure-hdinsight/>

10.6. Contoh Penggunaan Teknologi Big Data

Berikut ini diberikan contoh penggunaan teknologi *big data* untuk mengantisipasi penyebaran berita bohong (*hoax*). Misalkan berita yang ditangani bersumber dari media sosial Twitter. Analisis *hoax* atau tidak dilakukan menggunakan algoritma *machine learning* berupa algoritma clustering dan klasifikasi. Hasil analisis dikirimkan ke situs www.antihoax.id yang memiliki basis data MySQL enterprise.

Arsitektur teknologi dari contoh kasus di atas dapat dilihat pada Gambar 10.17.



Gambar 10.17. Arsitektur teknologi big data untuk kasus anti hoax.

Pada Gambar 10.17 dapat dilihat bahwa data Twitter dikumpulkan menggunakan *middleware* Flume atau Kafka. Hasil pengumpulan data ini diproses secara *real time* (waktu nyata) menggunakan aplikasi yang berjalan di atas Spark. Aplikasi ini mengakses pustaka (*library*) Spark Stream untuk menangani *data stream* yang dipropagasi dari Flume/Kafka, kemudian dilakukan analisis menggunakan algoritma *Machine Learning* dari Spark MLLib. Hasil analisis tersebut dikumpulkan dan dianalisis lebih lanjut menggunakan Spark SQL, lalu hasilnya disimpan di HDFS. Kemudian, dengan memanfaatkan Scoop, hasil analisis di HDFS itu pada setiap waktu tertentu diekspor ke basisdata MySQL yang menjadi bagian dari sistem situs www.antihoax.id. Dengan tersedianya hasil analisis berita Twitter yang senantiasa *up to date* di basisdata itu, situs www.antihoax.id dapat mempublikasikan berita-berita dari Twitter mana saja yang termasuk *hoax*.

Dengan menggunakan teknologi *big data*, maka data Twitter yang mengalir dengan sangat cepat dan menumpuk hingga berukuran sangat besar dapat ditangani. Pemrosesan data secara *real time* dan sinkronisasi hasil analisis ke situs www.antihoax.id juga dimungkinkan.

10.7. Kesimpulan

Big data merupakan aset informasi yang penting yang jika ditangani dengan baik akan memberikan manfaat penting dalam pengambilan keputusan organisasi dan bersifat *data driven*. Karakteristik volume yang besar, kecepatan yang tinggi dan sumber serta format yang beragam memerlukan teknologi pemrosesan yang khusus agar tidak menimbulkan masalah *data deluge*.

Pada bab ini telah dibahas teknologi *big data*, yang melingkup arsitektur, teknologi populer yang *free* dan *open source* serta teknologi komersial yang berbasis *cloud computing*. Pada bagian akhir juga sudah diberikan contoh kasus pemanfaatan teknologi *big data*. Dengan paparan pada bab ini, diharapkan para pembaca memiliki pemahaman awal tentang teknologi *big data* beserta pemanfaatannya.

Referensi

- Bajaber, F., Elshawi, R., Batarfi, O., Altalhi, A., Barnawi, A., dan Sakr, S. (2016): *Big data 2.0 Processing Systems: Taxonomy and Open Challenges*, *Journal of Grid Computing*, 14(3), 379–405, diperoleh melalui situs internet: <https://doi.org/10.1007/s10723-016-9371-1>.
- De Mauro, A., Greco, M., dan Grimaldi, M. (2016): A formal definition of *Big data* based on its essential features, *Library Review*, 65(3), 122–135, diperoleh melalui situs internet: <https://doi.org/10.1108/LR-06-2015-0061>.
- Ghemawat, S., Gobioff, H., dan Leung, S.-T. (2003): The Google file system, *Proceedings of the nineteenth ACM symposium on Operating systems principles - SOSP '03*, 29, diperoleh melalui situs internet: <https://doi.org/10.1145/945449.945450>.
- Holmes, A. (2012): *Hadoop In Practice - MEAP*, *Hadoop In Practice*, diperoleh melalui situs internet: <http://dl.acm.org/citation.cfm?id=2543981>.
- Moore, G. E. (2006): Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff, *IEEE Solid-State Circuits Newsletter*, 20(3), 33–35, diperoleh melalui situs internet: <https://doi.org/10.1109/N-SSC.2006.4785860>.
- Rowley, J. (2007): The wisdom hierarchy: Representations of the DIKW hierarchy, *Journal of Information Science*, 33(2), 163–180, diperoleh melalui situs internet: <https://doi.org/10.1177/0165551506070706>.

Bab ini sengaja dikosongkan

Bab 11 Pengumpulan Data Twitter dengan Teknologi Big Data

Oleh:

Muhammad Ravi dan Veronica S. Moertini

11.1. Pendahuluan

Sebagaimana telah dibahas pada Subbab 10.2, salah satu *V* yang mencirikan big data adalah *velocity*, dimana big data bertambah dengan sangat cepat, atau “mengalir” dengan kecepatan yang tinggi. Dalam konteks big data, data yang demikian disebut *data stream*. Sistem untuk menangkap lalu mengumpulkan big data yang berupa *stream* harus mampu menangani pertambahan data dengan kecepatan tinggi tersebut. Jika data dibutuhkan untuk dianalisis secara dinamis atau waktu nyata (*real time*), tidak dikumpulkan dulu lalu diproses secara *batch*, maka sistem itu juga harus disertai dengan fungsi untuk menganalisis *data stream*.

Salah satu sumber big data yang menghasilkan aliran data dengan kecepatan tinggi adalah Twitter. Media sosial ini memiliki pengguna sekitar 330 juta. Rata-rata per hari terdapat 500 juta *posting* (twit) atau sekitar 6000 twit per detik. Sebagaimana dipaparkan pada Subbab 10.4, Spark telah menyediakan kemampuan untuk memproses data *stream*. *Data stream* berupa *twits* dapat ditangani dengan memanfaatkan library Spark Streaming. Selain itu, penanganan *data stream* juga dapat dilakukan dengan teknologi lain, yaitu Kafka dan Zookeeper.

Bagaimana memanfaatkan Spark Streaming, dan pasangan Kafka pada sistem big data untuk menangani *data stream* twit? Apa perbedaan kedua pendekatan tersebut? Apakah masing-masing cocok digunakan untuk menjawab kebutuhan tertentu? Penelitian yang dilakukan dimaksudkan untuk menjawab pertanyaan-pertanyaan tersebut dan hasilnya dipaparkan pada bab ini.

Bab ini memaparkan konsep Hadoop, penanganan *data stream*, Spark, Spark Streaming, Kafka beserta Zookeeper (dengan pembahasan yang lebih detil dibandingkan Bab 10). Selanjutnya dibahas rancangan arsitektur sistem pengumpul *data stream* Twitter dengan memanfaatkan Spark Streaming maupun Kafka-Zookeeper, implementasi kedua sistem di lab big data, hasil eksperimen pengumpulan data, dan perbandingan kedua sistem tersebut.

11.2. Studi Literatur

11.2.1. Hadoop

Hadoop adalah *framework* untuk mengelola dan memproses big data secara terdistribusi. Jaringan klaster Hadoop dapat terdiri dari puluhan, ratusan atau ribuan komputer komoditas yang biasa dinamakan node. Ada node yang berperan sebagai NameNode (*master*), ada yang sebagai DataNode (*slave*). Dalam satu klaster, terdapat satu atau lebih NameNode (*main* dan *secondary* NameNode) dan sisanya sebagai DataNode. Banyaknya DataNode pada klaster tersebut dikonfigurasi berdasarkan kebutuhan. Hadoop memiliki dua komponen penting, yaitu HDFS (Hadoop Distributed File System) dan MapReduce⁴⁴.

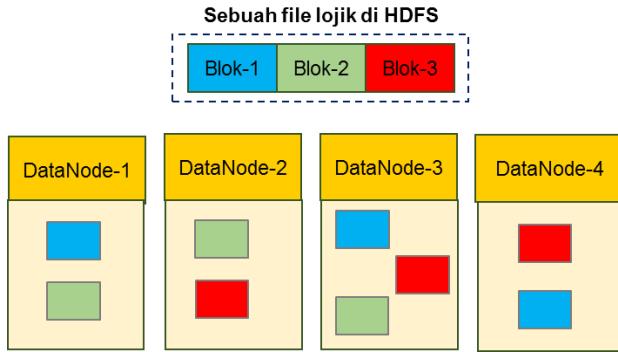
HDFS adalah sistem file terdistribusi yang dirancang untuk jaringan komputer. Pada HDFS, sebuah file dipecah-pecah menjadi blok-blok yang berukuran sama (kecuali blok terakhir). Default ukuran blok adalah 128 Mb. Blok-blok tersebut lalu direplikasi (*default*-nya 3 kali) dan disimpan di disk di komputer-komputer DataNode (lihat Gambar 11.1). HDFS membutuhkan NameNode dan DataNode. NameNode bertugas untuk mengatur operasi-operasi seperti membuka, menutup, dan menamai kembali file atau directory pada sistem file. NameNode meregulasi akses pengguna terhadap file dan mengatur blok mana yang akan diolah oleh DataNode. NameNode membuat semua keputusan terkait replikasi blok. Secara berkala, NameNode menerima *heartbeat* dari setiap DataNode di klaster. *Heartbeat* merupakan sinyal yang dikirim secara rutin dari DataNode ke NameNode untuk menandakan bahwa DataNode masih “hidup” dan berfungsi dengan benar⁴⁵.

DataNode atau *worker node* bertanggung jawab untuk menjalankan perintah membaca dan menulis untuk file sistem Hadoop. DataNode dapat membuat, menghapus, dan mereplikasi blok ketika diberi instruksi oleh NameNode. DataNode menyimpan dan mengambil blok ketika diperintahkan oleh NameNode.

MapReduce adalah sebuah model pemrograman untuk memproses data berukuran besar secara terdistribusi dan paralel pada klaster Hadoop. Dalam memproses data, sebuah job MapReduce terdiri dari dua fase yaitu map dan reduce. Map dan reduce menangani pasangan *key-value* sebagai *input* dan *output*. Fungsi map dijalankan pada DataNode dan bertugas untuk membaca data dari blok-blok (pecahan-pecahan) data yang umumnya tersimpan di disk komputer DataNode dan memprosesnya sesuai dengan algoritma yang dirancang pengembang. Keluaran map akan diserahkan kepada *reduce* untuk diproses lebih lanjut. Keluaran dari *reduce* menjadi hasil akhir² atau masukan pada job MapReduce lainnya.

⁴⁴ (White, 2008)

⁴⁵ (Miner & Shook, 2013)



Gambar 11.1. Ilustrasi pemecahan dan replikasi blok-blok pada file HDFS.

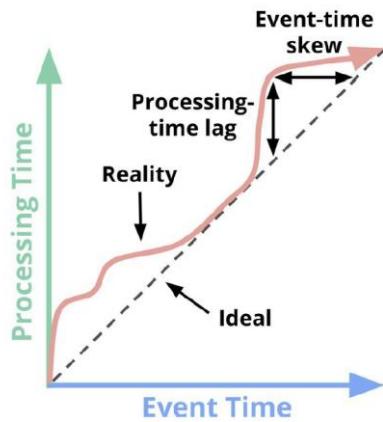
11.2.2. Big Data Stream

Data stream adalah data yang datang secara terus menerus “datang”, seperti aliran air dan ibaratnya tidak pernah berhenti. Potongan data yang mengalir itu dapat memiliki format yang tertentu (numerik, teks, suara atau video) dan tidak bersih (terdapat nilai salah atau kosong). Walaupun data yang dihasilkan per detik berukuran kecil namun karena data tersebut terus dihasilkan secara terus menerus, data dapat dikategorikan sebagai big data⁴⁶.

Sebelum *data stream* dapat dianalisis untuk mendapatkan *insights*, umumnya data harus disiapkan terlebih dahulu. Salah satu caranya adalah dengan ditransformasi terlebih dahulu. Transformasi ini dilakukan untuk mengubah data yang tidak terstruktur menjadi semi-terstruktur atau terstruktur. Proses transformasi ini dinamakan *stream preprocessing* dan metode yang sering digunakan adalah *filtering* dan *windowing*¹.

Stream preprocessing (penyiapan data aliran) harus bisa dilakukan dengan *latency* yang sangat rendah agar waktu (*time-stamp*) yang direkam (pada tiap potongan data) akurat. Pada streaming data, terdapat dua *time-stamp* yang perlu diperhatikan, yaitu *time-stamp* saat data dihasilkan atau datang dan *time-stamp* saat data masuk atau disimpan ke sistem. Jika *latency* pada proses penyiapan data tinggi, maka akan banyak *time-stamp* pada data yang disimpan tidak akurat. Kedua waktu tersebut dapat di gambarkan hubungannya dengan grafik pada Gambar 11.2.

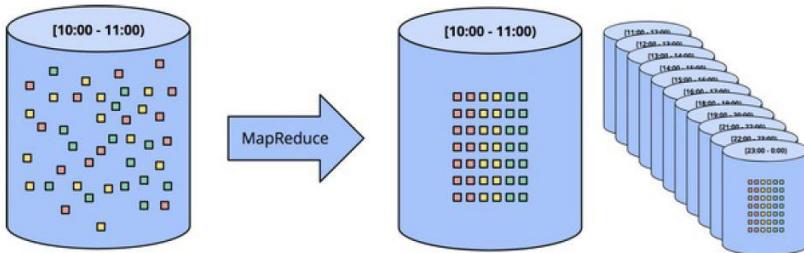
⁴⁶ (Akidau, Chernyak, & Lax, Streaming 101, 2018)



Gambar 11.2. Pemetaan waktu pada data stream (Akida, Chernyak, & Lax, 2018).

Pada Gambar 11.2, *event time* adalah waktu saat data dihasilkan/datang. *Processing time* adalah waktu *stream preprocessing*. Idealnya, *event time* sama dengan *processing time* (divisualisasikan dengan garis lurus). Tetapi, pada nyatanya tidak demikian, *processing time* membutuhkan komputasi yang menyebabkan *latency* bernilai (relatif) tinggi (diilustrasikan dengan garis berwarna pink), sehingga menyebabkan keterlambatan dalam penanganan data yang datang. Hal ini akan menimbulkan masalah, karena dapat berpengaruh terhadap analisis data stream secara keseluruhan⁴⁷.

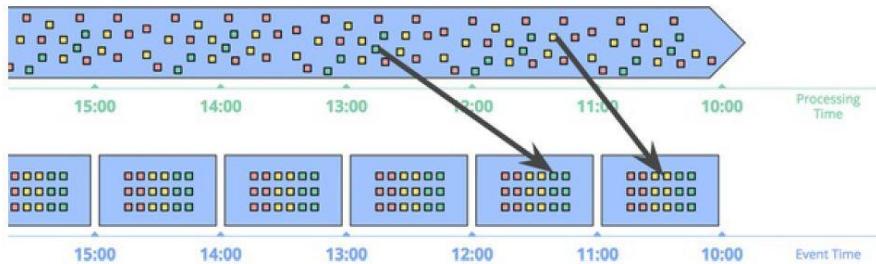
Di atas sudah disebut bahwa proses transformasi dapat dilakukan dengan *windowing*, yaitu membagi aliran data menjadi beberapa bagian berdasar waktu kedatangan. Proses *windowing* dapat dikelompokkan menjadi dua, yaitu *bounded processing* yang membagi aliran data berdasarkan *processing time* dan *unbounded processing* yang mengelompokkan data pada aliran berdasarkan *event time*.



Gambar 11.3. Windowing: Bounded Processing (Akida, Chernyak & Lax, 2018).

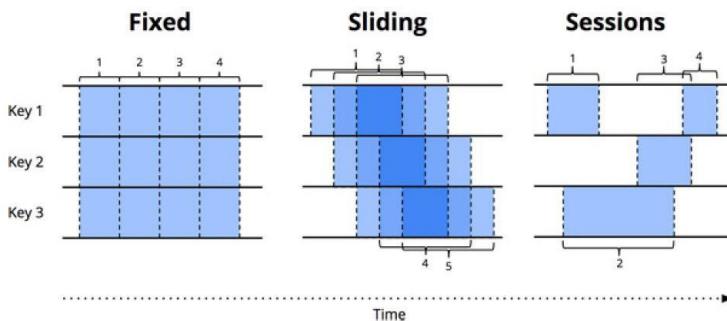
⁴⁷ (Akida, Chernyak, & Lax, The What, Where, When, and How of Data Processing, 2018)

Seperti diperlihatkan pada Gambar 11.3, proses *windowing* pada *bounded processing* dilakukan dengan cara mengelompokkan data sesuai kedatangan⁴⁸. Pada contoh di gambar, data dikelompokkan berdasar interval 10.00-11.00. Kelemahan dari pemodelan ini adalah tidak bisa mengatasi data yang terlambat karena data dikelompokkan sesuai waktu kedatangan pada sistem. Kelemahan dari pemodelan *bounded processing* bisa diatasi dengan *unbounded processing* (Gambar 11.4).



Gambar 11.4. Windowing: Unbounded Processing (Akidau, Chernyak & Lax, 2018).

Pada *unbounded processing*, aliran data yang masuk akan langsung dikelompokkan berdasarkan label waktu pada *event time*. Selanjutnya, proses transformasi akan dilakukan terhadap data yang telah dikelompokkan tersebut⁴⁹. Adapun teknik *windowing* dapat dibedakan menjadi tiga, yaitu *fixed*, *sliding* dan *sessions*, seperti ditunjukkan pada Gambar 11.5.



Gambar 11.5. Teknik-teknik windowing (Akidau, Chernyak & Lax, 2018).

Fixed window adalah window yang memiliki interval yang tidak *overlap*. Jadi, tiap data pada sebuah window tidak beririsan dengan data pada window yang lain. *Sliding window* adalah window yang memiliki interval yang *overlap*. Di sini, window terdiri dari beberapa interval atau periode dan window yang selanjutnya diperoleh dari mengikutsertakan data yang diperoleh pada satu atau lebih periode di window sebelumnya⁵⁰. *Sessions* hampir sama dengan *fixed*, tapi ukuran window-nya tidak tetap.

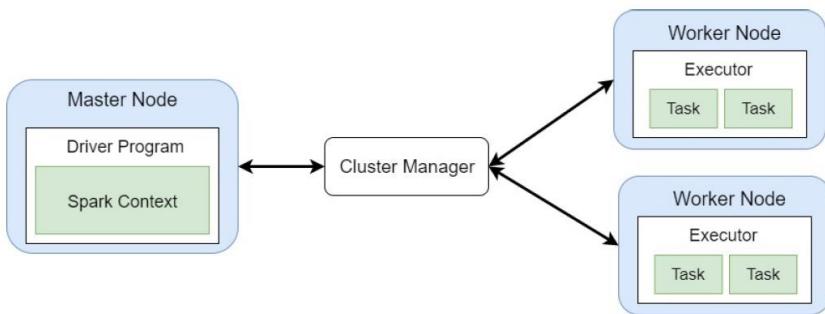
⁴⁸ (Akidau, Chernyak, & Lax, Streaming 101, 2018)

⁴⁹ (Akidau, Chernyak, & Lax, The What, Where, When, and How of Data Processing, 2018)

⁵⁰ (Akidau, Chernyak, & Lax, The What, Where, When, and How of Data Processing, 2018)

11.2.3. Spark

Data Stream tidak cocok diolah menggunakan *Hadoop*, karena Hadoop tidak dirancang untuk memproses data stream. Spark memfasilitasi *in-memory processing*, yaitu menjalankan komputasi di memori paralel dengan DAG (Directed Acyclic Graph). DAG mengoptimasi langkah-langkah eksekusi tugas-tugas pada aplikasi atau *workflow*⁵¹. Melalui DAG, Spark mencari langkah-langkah yang optimal untuk mengerjakan tugas-tugas tersebut. Arsitektur dari Spark terdiri dari beberapa bagian yang ditunjukkan pada Gambar 11.6.



Gambar 11.6. Arsitektur Spark (Karau, et al, 2015).

Komponen pada arsitektur Spark (Gambar 11.6) adalah:

- Driver Program bertugas menjalankan fungsi main pada node master dan tempat dimana Spark Context dibuat. Kode program akan diterjemahkan menjadi task dan akan dijadwalkan untuk dikerjakan oleh eksekutor, yang lalu akan berkomunikasi dengan klaster manager untuk mengatur sumber daya.
- Spark Context menghubungkan aplikasi client dengan klaster manager seperti *YARN* atau *MESOS* dan digunakan untuk membuat RDD, accumulator, dan *broadcast variable*.
- Cluster Manager mengatur sumber daya pada sebuah klaster Spark.
- Executor adalah proses-proses yang berjalan pada worker node dan bertanggung jawab untuk mengerjakan task yang diberikan.
- Task adalah satuan kerja pada Spark yang berisi perintah-perintah untuk dijalankan. Task akan dikirim oleh Driver Program ke Executor untuk dijalankan. Pada umumnya task akan dibuat untuk setiap partisi dari objek Resillient Distributed Dataset. Partisi merupakan potongan data yang terdistribusi dan tersimpan di memori pada komputer-komputer di klaster Spark⁵².

⁵¹ (Karau, Kowinski, Wendell, & Zaharia, 2015; Karau, Kowinski, Wendell, & Zaharia, 2015)

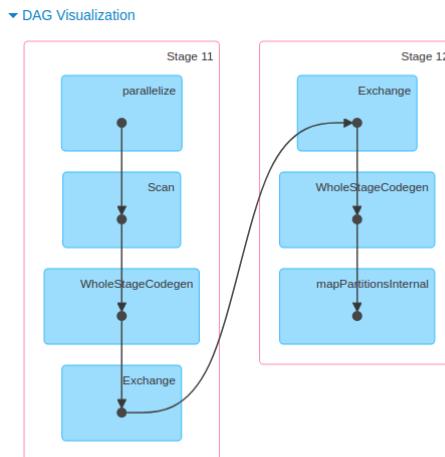
⁵² (Karau, Konwinski, Patrick, & Zaharia, Programming with RDDs, 2015)

Spark memiliki struktur data khusus, yang dinamakan **Resillient Distributed Dataset (RDD)**, yang berisi koleksi objek-objek yang didistribusikan. Setiap dataset yang dibuat menjadi objek RDD dibagi menjadi beberapa partisi yang direplikasi dan didistribusikan di komputer-komputer Worker Node.

Secara umum, RDD dapat dibuat dengan dua cara yaitu dengan memuat dataset eksternal dan mentransformasi RDD yang telah dibuat sebelumnya. Elemen-elemen yang disimpan pada RDD memiliki sifat *fault tolerance*. Ketika terjadi kegagalan dalam pemrosesan karena ada komputer (di klaster) yang mati, data tidak akan hilang dan pemrosesan bisa langsung dilakukan dari titik komputasi terakhir (sebelum mati).

RDD memiliki dua buah operasi yaitu: Transformasi dan Aksi. Transformasi adalah operasi yang menghasilkan RDD baru dari RDD yang telah ada. Jadi, operasi ini menghasilkan RDD baru. Sedangkan operasi Aksi mengembalikan nilai hasil komputasi terhadap RDD, misalnya hasil agregasi dari RDD. Hasil dari Aksi akan dikirim ke *driver program* atau disimpan pada penyimpanan eksternal seperti HDFS. Operasi yang dilakukan pada Spark menggunakan *lazy evaluation*. Spark hanya mengeksekusi perintah ketika bertemu operasi yang berbentuk Aksi (Karau, Konwinski, Patrick, & Zaharia, Programming with RDDs, 2015).

Transformasi yang dilewati oleh Spark sebelum bertemu dengan operasi aksi akan ditulis pada metadata dan akan digunakan untuk mencari langkah optimal pada DAG. Perencanaan tahapan tersebut dimodelkan pada sebuah lapisan DAG Scheduler (Gambar 11.7).



Gambar 11.7. *Directed Acyclic Graph (Spark Web-UI, n.d.).*

DAG Scheduler mengimplementasikan penjadwalan eksekusi perintah berbasis *stage*. Pada DAG, vertex merupakan RDD-RDD dan edges merupakan transformasi yang menghasilkan vertex baru sesuai arah edge sampai mengarah ke vertex terakhir.

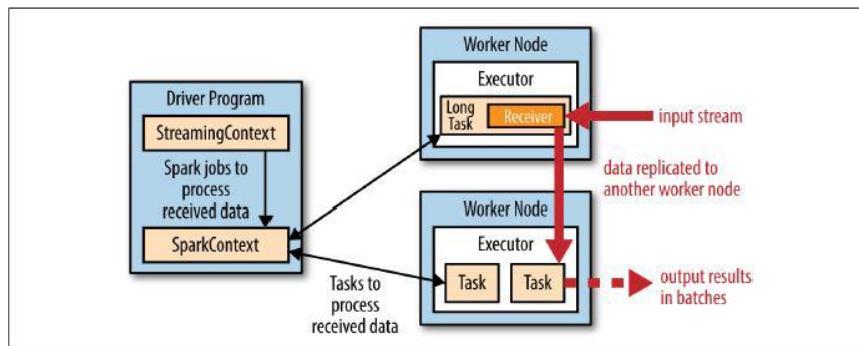
11.2.4. Spark Streaming

Pada Subbab 11.2.2 telah dijelaskan tentang pemodelan pengolahan data stream dengan *unbounded* atau *bounded processing*. Spark Streaming adalah *library* ekstensi dari Spark yang digunakan untuk mengumpulkan dan mengolah data stream dengan *bounded processing*. Dengan demikian, proses *windowing* dilakukan berdasarkan waktu *processing time*⁵³.

Spark Streaming masih memiliki sifat-sifat utama dari Spark, yaitu *in-memory* (data disimpan dan diproses di memori yang terdistribusi), *fault tolerant* (a.l. direalisasi dengan cara mempartisi dari mereplikasi data di memori) dan memiliki sifat tambahan yaitu bisa mengumpulkan data secara *real-time* dari berbagai sumber seperti Twitter, TCP Socket, dan Kafka.

Spark Streaming bekerja dengan cara mengumpulkan data stream dari suatu sumber dan mengubahnya menjadi rangkaian RDD yang disebut dengan Discretized Stream (Dstream). Karena Dstream merupakan rangkaian RDD, operasi transformasi dapat diterapkan pada tiap-tiap RDD. Tapi, karena Spark Streaming bekerja secara paralel operasi transformasi tersebut seolah-olah diterapkan pada seluruh rangkaian RDD. Seperti RDD secara umum, Dstream pun dapat dibuat dengan cara mengambil data eksternal atau mentransformasi Dstream lain yang telah ada.

Spark Streaming bekerja mirip dengan Spark. Namun, ada komponen tambahan pada Driver Program yaitu Streaming Context yang mengatur tugas Spark Context, mengumpulkan atau memproses data.



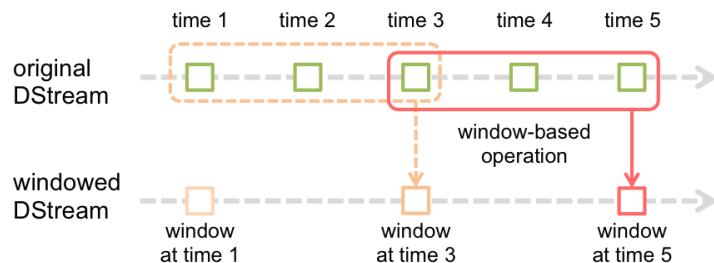
Gambar 11.8. Cara Kerja Spark Streaming (Karau et al, 2015).

Sebagaimana ditunjukkan pada Gambar 11.8, Spark Context bekerja seperti Spark pada umumnya, yaitu mengatur dan menjadwalkan Executor untuk mengumpulkan data atau mengolah data yang telah masuk. Biasanya dari semua node yang ditugaskan hanya beberapa saja yang bertugas mengumpulkan data, selebihnya hanya bertugas untuk menduplikasi data yang telah terkumpul agar tidak ada yang hilang dan mentransformasi data sesuai perintah yang ada pada Spark Context⁸.

⁵³ (Karau, Konwinski, Patrick, & Zaharia, Spark Streaming, 2015)

Operasi transformasi pada Spark Streaming dibedakan menjadi dua, yaitu Stateless dan Stateful. Transformasi Stateless dilakukan pada tiap-tiap RDD pada Dstream. Dengan kata lain, tiap-tiap RDD ditransformasi satu per satu secara independen. Jika Dstream dibuat untuk selang waktu yang sangat singkat, misalnya 1 detik, transformasi stateless mungkin tidak cocok dilakukan karena bisa jadi tidak ada informasi signifikan yang dapat “digali” dari data kecil pada interval waktu sesingkat itu.

Transformasi Stateful digunakan untuk mengakses dan memproses lebih dari satu Dstream sekaligus. Dengan demikian, informasi yang ada pada beberapa Dstream yang telah terbentuk sebelumnya dapat dianalisis lagi. Transformasi Stateful diterapkan dengan metode *windowing* berdasarkan dengan waktu kedatangan data pada sistem (*processing-time*).



Gambar 11.9. Stateful Transformation (Spark Streaming Programming Guide, n.d.)

Pada Gambar 11.9, ukuran *batch window* adalah 1 detik, *sliding window* diatur dengan periode geser setiap 2 detik dan ukuran *window* (kotak merah dan oranye) adalah 3 detik⁵⁴. Operasi transformasi terhadap DStream pada time 5 (DStream pada time 5) mengikutsertakan Dstream pada tme 4 dan time 3.

Dengan cara kerja seperti yang telah dijelaskan di atas, Spark Streaming memiliki kelemahan, yaitu tidak menangani data yang terlambat datang/masuk.

11.2.5. Structured Streaming

Structured Streaming adalah sebuah sistem pemrosesan data stream yang dibangun dengan menggunakan Spark SQL. Berbeda dengan Spark Streaming, sistem ini dirancang untuk mengatasi data stream dengan pemodelan *unbounded processing*. Proses *windowing* pada sistem ini dilakukan dengan mengelompokkan data stream berdasar waktu saat data tersebut dihasilkan¹¹.

Ide dasar dari Structured Streaming ini adalah memperlakukan data seperti data pada tabel basisdata. Setiap data yang baru dihasilkan akan dianggap sebagai baris baru pada tabel dan akan dimasukkan ke

⁵⁴ (Karau, Konwinski, Patrick, & Zaharia, Spark Streaming, 2015)

tabel yang sudah ada di memori. Karena Spark SQL merupakan ekstensi dari Spark maka Structured Streaming memiliki sifat yang sama dengan Spark yaitu komputasi dilakukan di memori secara terdistribusi dan dapat kegagalan sistem dapat diatasi. Di sini, hal yang berbeda adalah data tidak lagi berbentuk RDD melainkan dataframe. Pada Spark, dataframe merupakan koleksi data terdistribusi yang memiliki sifat-sifat RDD namun telah dioptimalkan dengan SQL engine sehingga, developer dapat melakukan kueri SQL dengan cepat.

Perbedaan utama Spark Streaming terhadap Structured Streaming terletak di bagian pembacaan dan penulisan data. Structured Streaming langsung mentransformasi data yang tidak terstruktur yang diterima menjadi terstruktur dengan bantuan skema pada Spark SQL, sedangkan Spark Streaming membaca data yang tidak terstruktur yang masuk (walaupun berformat semi terstruktur seperti JSON atau AVRO) dan menyimpannya sebagai RDD berbentuk DStream.

Structured Streaming membaca data dengan sistem *trigger* (ini berbeda dengan Spark Streaming yang membuat potongan-potongan RDD berukuran beberapa detik untuk mengumpulkan data stream). Pada sistem trigger, misalkan data yang masuk pada detik ke 1 ada sebanyak 2000 rekord, maka seluruh rekord akan disimpan di tabel (yang tersimpan di memori). Jika pada detik berikutnya dihasilkan 200 rekord, maka seluruhnya akan ditambahkan ke tabel tadi. Jadi, ukuran tabel pada sistem ini akan terus membesar sampai batas maksimal ukuran memori (karena itu disebut *unbounded*).

Dengan cara kerja di atas, sistem Structured Streaming tidak bisa mengumpulkan data langsung dari sumber data yang asli, tapi sistem harus terhubung terlebih dahulu ke teknologi pengumpul big data lain yang lebih stabil.

11.2.6. Kafka

Kafka adalah sistem pengumpul data stream yang juga dapat beroperasi dalam menyederhanakan sistem. Penyederhaan ini dilakukan Kafka dengan menjadi perantara antara *sumber* data stream dengan *target* data stream (*client*). Hubungan antara target dan sumber harus disederhanakan karena sumber dari *data stream* dapat lebih dari satu, bahkan banyak. Misalkan: suatu sistem *e-commerce* memiliki sumber data berupa *website events*, *pricing data*, *financial transaction*, dan *user transaction*. Semua data dari sumber-sumber tersebut dibutuhkan untuk sistem-sistem seperti *database*, *analytics*, *email system*, dan *audit*. Jika sistem pengumpulan data tidak disederhanakan maka analisis data menjadi kompleks. Kafka dikembangkan untuk mempermudah hal tersebut. Kafka juga dapat digunakan sebagai tempat “transit data” sebelum dikirim ke sistem lain.

Komponen-komponen penting pada Kafka adalah Topics, Broker, Producer, dan Consumer⁵⁵ yang dibahas di bawah ini.

⁵⁵ (Narkhede, Shapira, & Palino, Meet kafka, 2017)

Topics adalah suatu aliran data yang dimodelkan mirip dengan tabel pada sistem database. Topics dapat berisi banyak topik yang dibedakan melalui namanya. Tiap topik akan dipecah menjadi partisi, setiap partisi mempunyai ID bertipe integer terurut menaik yang disebut *offset*. Partisi ini akan direplikasi (seperti partisi pada RDD). *Topics* dapat menyimpan data selama satu minggu. Artinya, pada durasi satu minggu tersebut, data tidak akan bisa dihapus. *Offset* bersifat *immutable*, sekali dibuat tidak bisa dihapus dan hanya bisa direset kembali ke urutan awal.

Broker adalah komputer yang menyimpan Topics. Sebuah klaster Kafka terdiri dari beberapa broker. Setiap broker dibedakan oleh ID dengan tipe *integer* dan menyimpan beberapa partisi Topics. Sistem yang kita bangun hanya perlu terhubung ke salah satu broker saja dan akan langsung terhubung ke seluruh broker yang ada pada klaster Kafka.

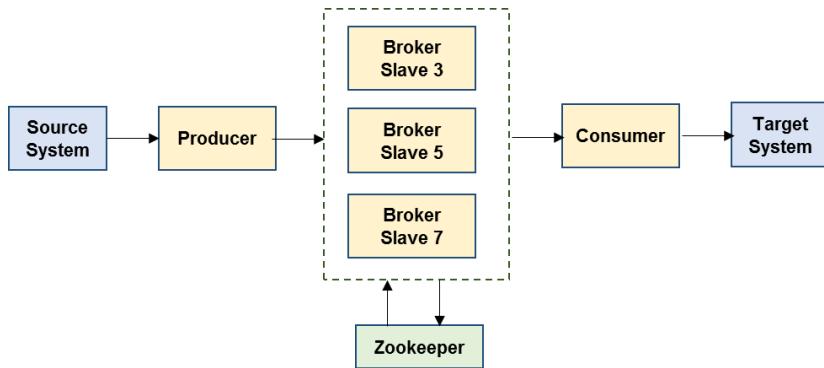
Broker mempunyai dua peran yaitu *leader* dan *follower*. Leader bertugas mengambil data dari sumber untuk salah satu partisi, sedangkan follower akan menduplikasi partisi tersebut dan melakukan sinkronasi. Sebagai contoh: Jika kita memiliki sumber data dari Twitter, kita dapat memecah data dari Twitter menjadi 3 partisi A, B, dan C. Leader partisi A adalah broker nomor 2. Maka, hanya broker nomor 2 yang bisa mengambil data Twitter untuk partisi A. Broker yang lain hanya akan menduplikasi. Hal ini berlaku untuk setiap partisi. Setiap partisi memiliki leader dan follower. Leader dipilih dengan sistem *voting* maka dari itu jumlah broker pada klaster harus ganjil.

Producer adalah komponen Kafka yang menulis data ke Topics. Producer akan langsung tahu ke topik mana yang menjadi bagiannya. Jika salah satu broker gagal, producer akan mengirim data ke broker yang lain. Producer juga memiliki *acknowledgement* atau konfirmasi. Konfirmasi dilakukan untuk menghindari *data loss* dan data duplikasi. Terdapat 3 jenis konfirmasi, yaitu *acks=0*, *acks=1*, dan *acks=all* yang dijelaskan berikut ini:

- *acks=0* berarti producer tidak menunggu konfirmasi dari broker dan hanya akan terus mengirim data dari sumber ke broker. Hal ini akan menyebabkan *data loss* atau duplikasi data karena *producer* tidak tahu apakah data sudah diterima atau belum.
- *acks=1* berarti producer akan menunggu konfirmasi dari leader apakah data sudah diterima atau belum. Jika data sudah diterima maka *producer* tidak akan mengirim lagi data yang sama. Tapi, *acks=1* hanya akan menjamin tidak ada data yang hilang pada *leader* saja.
- *acks=all* berarti tidak ada data yang hilang pada follower.

Consumer adalah komponen Kafka yang membaca data dari *Topics*. Data akan dibaca secara terurut untuk setiap partisi. Tetapi tidak terurut partisi demi partisi. Misalnya: terdapat dua partisi, Partisi 1 dan partisi 2. Setiap *offset* pada partisi 1 dan partisi 2 akan dibaca secara terurut. Tetapi, consumer tidak akan menjamin membaca data dari partisi 1 terlebih dahulu. Data akan dibaca secara acak berdasarkan mana yang duluan selesai dibaca *offset*-nya.

Setelah membahas komponen-komponen Kafka, cara kerja *Kafka* secara keseluruhan dijelaskan di bawah ini (lihat contoh arsitektur sistem pada Gambar 11.10).



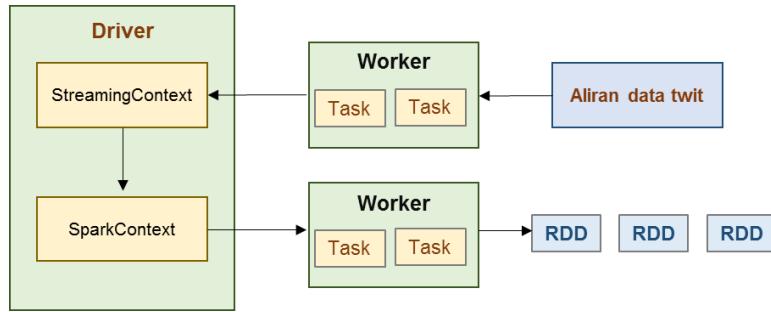
Gambar 11.10. Contoh arsitektur sistem pengumpul data dengan Kafka.

Producer mengambil data dari sumber dan menulisnya ke topik yang disimpan pada broker di Slave 3, 5, dan 7. Broker kemudian mereplikasi data, dan mengirim konfirmasi pada producer untuk memberi tahu apakah data sudah berhasil diterima dan direplikasi pada setiap broker. Jika belum berhasil, maka producer akan mengirim data kembali. Selanjutnya, consumer akan membaca data dari Topics. Consumer hanya perlu terhubung ke salah satu broker saja maka akan lansung terhubung ke seluruh klaster Kafka. Zookeeper digunakan untuk mengatur *leader* dan *follower*. Di sini, Consumer bisa saja berupa sistem lain, seperti Spark Streaming atau Structured Streaming.

11.3. Pengumpul Data Twitter dengan Spark Streaming

11.3.1. Arsitektur Sistem

Arsitektur sistem pengumpul data ini (Gambar 11.11) cukup sederhana karena Spark Streaming langsung mengambil data dari Twitter. Saat sistem mengambil data dari Twitter, sistem mengambilnya dengan cara membuat *batch-batch* data dengan ukuran yang bisa ditentukan sendiri oleh pengguna. Objek dari DStream (berupa RDD) dibuat SparkStreaming berdasarkan ukuran tersebut. Selanjutnya, objek-objek DStream dapat diolah/ditransformasi sesuai dengan fungsi program (menggunakan *stateless* atau *stateful transformation*) yang dibuat pengguna. Keluaran dari program berupa objek-objek RDD yang selanjutkan dapat disimpan di HDFS (karena Spark dijalankan di atas Hadoop).



Gambar 11.11. Arsitektur pengumpul data dengan Spark Streaming.

11.3.2. Perangkat Lunak Pengumpul Data Twitter dengan Spark Streaming

Perangkat lunak yang dibangun digunakan untuk mengumpulkan data twit dan menghitung *trending topics* atau topik yang sering dibicarakan. Hasil perhitungan akan disimpan secara berkala sesuai input dari pengguna, misalkan: memantau perkembangan *trending topics* 1 jam kebelakang, data diambil setiap 30 menit sekali, dengan durasi pemantauan adalah selama 10 jam. Di sini, *trending topics* ditentukan dengan cara menghitung jumlah kata yang diawali dengan symbol *hashtag* (#) pada setiap twit pada interval waktu tertentu.

Contoh satu twit (yang hanya diambil dengan komponen tertentu dan informasi sensitif diedit) dalam format JSON diberikan di bawah ini:

```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2020",
  "id_str": "8550009999921695744",
  "text": "sore ini kircon panas #bandung #jabar",
  "in_reply_to_user_id": null,
  "user": {
    "id": 5555994945,
    "name": "BelaBdg20",
    "statuses_count": 351,
    "location": "Bandung",
    "url": null
  }
}
```



Gambar 11.12. Algoritma komputasi trending topics.

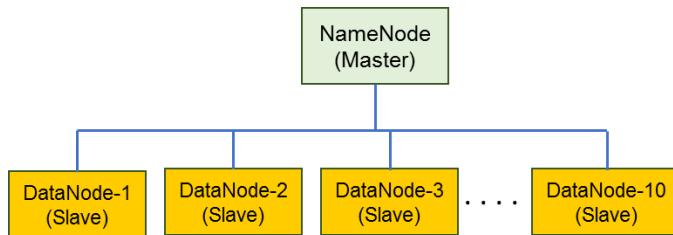
Perangkat lunak ditulis dengan bahasa Scala. Algoritma yang dijalankan pada perangkat lunak tersebut ditunjukkan pada Gambar 11.12, dengan tahapan: Ambil teks twit, ubah ke kata-kata (*words*) dengan menggunakan fungsi transformasi flatMap, pilih kata-kata yang diawali dengan karakter '#', petakan menjadi pasangan *<key, value>* dimana value diganti nilai 1 untuk tiap kata (pasangan menjadi *<kata, 1>*), hitung kemunculan kata (jumlahkan value), urutkan kata berdasarkan jumlah kemunculan, dan simpan di HDFS. Pada perangkat lunak, konfigurasi diatur agar data hasil komputasi disimpan di HDFS tiap perioda 30 menit. Data yang disimpan adalah kata (beserta jumlah kemunculannya) yang paling sering muncul selama satu jam terakhir.

11.3.3. Eksperimen

Tujuan dari eksperimen ini adalah menganalisis performa, apakah sistem dapat mengumpulkan data secara paralel, apakah masih terjadi *delay*, dan bagaimana hasil dari *data stream* yang dikumpulkan oleh *Spark Streaming*.

Eksperimen dilakukan pada klaster Hadoop, Spark (yang juga menjalankan Kafka) dengan 11 komputer. Pada klaster Hadoop, satu komputer bertindak sebagai NameNode atau node master dan sepuluh sisanya sebagai DataNode atau node slave. Untuk klaster Kafka, digunakan 3 dari node slave dimana salah satunya akan dijadikan sebagai leader. Seluruh komputer yang digunakan mempunyai spesifikasi yang sama (kecuali memori), yaitu:

- Memori: 16 Gb (NameNode) dan 8 Gb (DataNode)
- Prosesor: 6 buah Intel Core i3 CPU 550 @3.20GHz
- Harddisk: 1TB



Gambar 11.13. Klaster untuk eksperimen.

Cara melakukan eksperimen adalah dengan menjalankan perangkat lunak (dengan bahasa Scala) pada klaster Spark dengan konfigurasi:

```
spark-submit --class Main --master yarn --num-executors 10 --executor-cores 4 twitter-stream-analysis-assembly-0.1.jar 300 900 1800 /path
```

Spark-submit digunakan untuk menjalankan perangkat lunak pada klaster Spark. Argumen `--class` mendefinisikan kelas utama, `--master` mendefinisikan manajer klaster yang digunakan (di sini YARN), `--num-executors 10` menyatakan jumlah eksekutor (10 komputer) yang dijalankan secara paralel, `--executor-cores 4` menyatakan jumlah cores yang digunakan (pada tiap komputer) dan berikutnya adalah nama file jar, angka berikutnya secara terurut menyatakan *batch size* = 300 detik, *sliding-window* = 900 detik, dan ukuran *window* = 1800 detik atau setengah jam. Path adalah alamat (direktori) tempat hasil komputasi (output) disimpan.

Setelah program dijalankan, kinerja dari node-node Master dan Slave dapat diobservasi dari laporan yang diberikan Spark, contohnya seperti terlihat pada Gambar 11.14. Kolom RDD Blocks menyatakan jumlah RDD yang diproses, Storage Memory menampilkan memori pada node eksekutor yang digunakan, Active Task, Failed Task, dan Complete Task masing-masing menyatakan tugas yang sedang dieksekusi, tugas yang gagal, dan tugas yang telah selesai. Pada gambar terlihat hanya Slave 2 yang membuat DStream dan terus-menerus aktif mengambil data Twits. Node-node yang lain hanya menerima RDD dari Slave 2 dan mengolahnya secara paralel. Namun tidak semua node slave (worker) memproses RDD secara seimbang. Hal ini dapat dilihat pada kolom RDD Blocks, dimana nilai RDD untuk Slave 3, 7, 8, 9 dan 10 adalah 0 (nol) dan sisanya ada di kisaran 26-28. Ini menunjukan bahwa data-data yang telah ditangkap dan diubah menjadi RDD hanya tersebar dan terkonsentrasi pada komputer tertentu saja.

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	GC Time	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
driver	master:36631	Active	0	11.3 MB / 384.1 MB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B		Thread Dump
1	slave2:33515	Active	28	39 MB / 384.1 MB	51 KB	4	1	0	52198	52199	6.3 h (1.3 min)	780.8 MB	411.3 MB	144.7 MB	stdout stderr	Thread Dump
2	slave10:44451	Active	26	29.6 MB / 384.1 MB	0.0 B	4	0	0	49003	49003	6.0 min (19 s)	545.4 MB	432.2 MB	232.8 MB	stdout stderr	Thread Dump
3	slave5:42785	Active	0	888.9 KB / 384.1 MB	0.0 B	4	0	0	41316	41316	4.7 min (3 s)	0.0 B	0.0 B	19.3 MB	stdout stderr	Thread Dump
4	slave4:43527	Active	26	6.2 MB / 384.1 MB	0.0 B	4	0	0	50776	50776	4.6 min (3 s)	61.2 MB	49.3 MB	44.8 MB	stdout stderr	Thread Dump
5	slave6:40987	Active	26	29.4 MB / 384.1 MB	31.9 KB	4	0	0	45932	45932	2.7 h (30 s)	539.9 MB	432.3 MB	224.1 MB	stdout stderr	Thread Dump
6	slave1:46575	Active	26	6.1 MB / 384.1 MB	0.0 B	4	0	0	49295	49295	4.8 min (3 s)	59.7 MB	55.6 MB	46 MB	stdout stderr	Thread Dump
7	slave7:36525	Active	0	905.3 KB / 384.1 MB	0.0 B	4	0	0	50038	50038	4.2 min (3 s)	0.0 B	1.7 KB	23.6 MB	stdout stderr	Thread Dump
8	slave3:43433	Active	0	888.9 KB / 384.1 MB	0.0 B	4	0	0	46768	46768	4.5 min (3 s)	0.0 B	0.0 B	21.5 MB	stdout stderr	Thread Dump
9	slave9:43367	Active	0	888.9 KB / 384.1 MB	0.0 B	4	0	0	37745	37745	4.7 min (5 s)	0.0 B	0.0 B	16.8 MB	stdout stderr	Thread Dump
10	slave8:36263	Active	0	888.9 KB / 384.1 MB	0.0 B	4	0	0	40033	40033	4.8 min (2 s)	0.0 B	0.0 B	18.6 MB	stdout stderr	Thread Dump

Gambar 11.14. Performa Spark Streaming.

Setelah melakukan eksperimen penangkapan data Twits selama 10 jam dan melakukan pengamatan pada jam ke-2, 5 dan 10, hasilnya diringkas dan dipaparkan pada Tabel 11.1. Input rate menyatakan jumlah batch (masing-masing berukuran 5 menit) dari data twit yang diproses per detik. Scheduling delay adalah waktu tunggu eksekutor untuk menjalankan task. Processing time menyatakan lama waktu yang dibutuhkan untuk melakukan transformasi terhadap batch-batch. Total delay menyatakan keseluruhan waktu yang dibutuhkan untuk mengambil data twit sampai menulis hasilnya ke HDFS. Sedangkan Total Tweet menyatakan jumlah twit yang diproses.

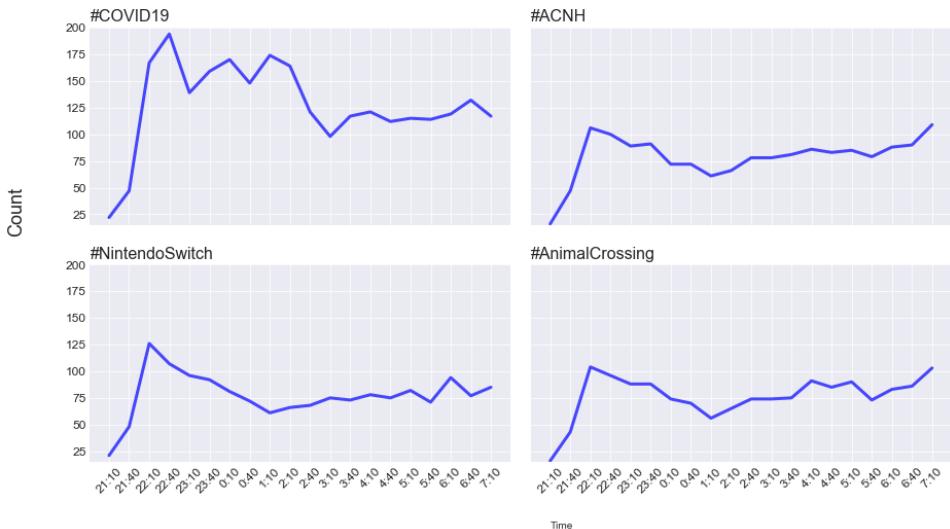
Table 11.1. Performa Spark Streaming

		Jam ke-2	Jam ke-5	Jam ke-10
Input rate		63.25	63.11	55.28
Scheduling delay		0	0	0
Processing time		1.238s	3.691s	5.807s
Total delay		1.239s	3.692s	5.808s
Total Tweet		338,405	1,082,065	2,039,922

Pada Tabel 11.1, nilai input rate terus menurun sampai jam ke-10. Scheduling delay bernilai 0, berarti tidak ada data twit yang harus menunggu untuk diproses. Processing time meningkat dari jam ke jam seiring dengan makin banyaknya data twit yang masuk. Pada, total delay dimana nilainya merupakan processing time ditambah 1 detik, mengindikasikan adanya keterlambatan penulisan data ke HDFS.

Pada eksperimen ini, data yang berhasil terkumpul selama 10 jam adalah 2 juta rekord dengan ukuran 2.2 Gb. Hasil analisis berupa *trending topics* diberikan pada Gambar 11.15.

Apa yang Sering Dibicarakan 10 jam yang lalu?



Gambar 11.15. Perkembangan trending topics selama 10 jam awal Mei 2020.

Data twit yang terkumpul dari waktu ke waktu dapat langsung dianalisis dan divisualisasi seperti ditunjukkan pada Gambar 11.15. Pada gambar tersebut ditunjukkan bahwa empat topik yang paling sering dibicarakan adalah COVID19, ACNH, Nintendo Switch, dan Animal Crossing. Eksperimen ini dilakukan pada awal Mei 2020 dimana pada bulan tersebut negara-negara sedang kewalahan menghadapi pandemi COVID19, sehingga tidak mengherankan jika banyak penduduk dunia yang membicarakan topik tersebut. Sedangkan ketiga trending topik lainnya berhubungan dengan game, yaitu Animal Crossing atau Animal Crossing New Horizon (ACNH). Di banyak negara diterapkan kebijakan *lockdown*, dan orang-orang mencari kesibukan di rumah dengan bermain game. Pada waktu itu, game yang baru dirilis adalah Animal Crossing garapan Nintendo. Karena itu, tidak mengherankan bahwa topik-topik tersebut banyak dibicarakan di Twitter. Insight lain yang ditemukan adalah COVID19 sering dibicarakan pada jam 9-10 malam.

Dengan eksperimen di atas, bisa disimpulkan bahwa Spark Streaming dapat mentransformasi data dan menganalisis data secara *real-time*. Dengan pengaturan ukuran batch (5 menit) dan algoritma komputasi yang sederhana (Gambar 11.12), setiap twit yang datang/masuk dapat ditangani secara on-time (tidak terjadi *delay* penanganan terhadap aliran data yang masuk).

11.4. Pengumpul Data Twitter dengan Kafka

Pengumpulan data Twitter dengan Spark Streaming relatif mudah karena sudah Spark sudah menyediakan *library* (API) untuk mengubah data stream menjadi Dstream. Namun jika *library* tidak tersedia, fungsi untuk mengubah data tersebut cukup kompleks. Selain itu *Spark Streaming* memiliki

kelemahan dilihat dari format data yang dihasilkan, dimana data yang diterima akan disimpan dalam format tidak terstruktur. Jika dibutuhkan hasil pengumpulan data dalam format semi-terstruktur, Kafka dapat digunakan sebagai perantara untuk mengumpulkan data stream terlebih dahulu, lalu mengubahnya ke bentuk semi-terstruktur.

11.4.1. Perancangan dan Arsitektur Pengumpul Data Twitter dengan Kafka

Pengumpulan data Twitter dengan Kafka dilakukan dengan membuat Twitter Stream API dan membuat Kafka Connect.

Twitter Stream API:

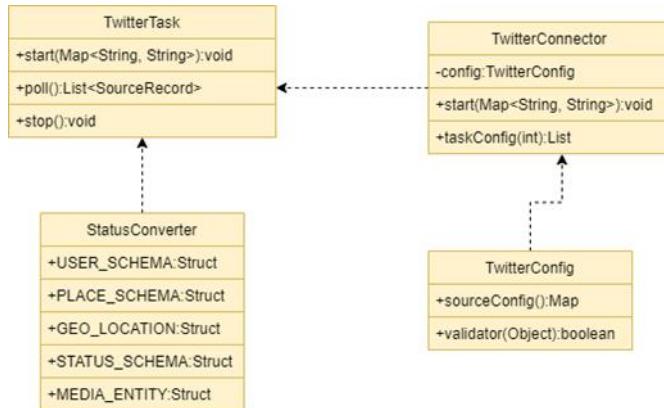
Untuk mengambil data dari Twitter digunakan Twitter Stream API, yaitu API yang melakukan request data ke Twitter secara terus menerus (ketika data dibutuhkan). Dengan membuka koneksi antara program client dengan API, server Kafka akan terus mengirim pesan melalui koneksi tersebut jika ada data masuk yang baru. Karena itu, Stream API ini menyediakan data secara real-time dengan *throughput* yang tinggi. Data yang didapat sendiri berformat JSON.

Rancangan Kafka Connect

Data stream yang berasal dari Twitter-Stream API akan dikumpulkan oleh Kafka dan disimpan di Topik (analogi dengan tabel pada basisdata). Di sini, Kafka Connect berperan menjadi pengumpul data, menjadi source sekaligus producer, yang menulis data yang masuk ke Topik. Data yang ditulis ke Topik dapat difilter terlebih dahulu, misalnya yang hanya mengandung keyword tertentu saja. Parameter yang harus diisi pada saat membuat instansiasi dari Kafka Connect adalah:

- Jumlah task yang akan dieksekusi secara paralel
- Kata yang akan menjadi keywords untuk memfilter data
- Access token untuk otentikasi akses Twitter
- Nama topik yang menjadi tujuan penulisan data.

Pada penelitian ini dirancang empat kelas utama untuk Kafka Connect (Gambar 11.16). Deskripsi ringkas dari tiap kelas diberikan di bawah ini.



Gambar 11.16. Diagram kelas pada program pengambil data dengan Kafka.

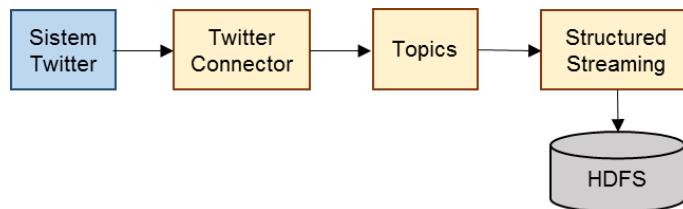
TwitterConfig: digunakan untuk menangai parameter masukan dari pengguna. Di sini terdapat method untuk mengecek apakah parameter tersebut valid atau tidak.

TwitterConnector: digunakan untuk mengambil data dari Twitter secara real-time. config adalah variabel untuk menyimpan konfigurasi dari pengguna. Method start berfungsi untuk mengambil data dari Twitter, sedangkan method taskConfig digunakan untuk menentukan ada berapa task (paralel) yang mengambil data dari Twitter.

StatusConverter: digunakan untuk membuat schema data Twitter yang masuk. Pada perangkat lunak ini, schema dirancang untuk menyimpan data user, tempat dan pesan twit.

TwitterTask: merupakan kelas utama dari Kafka Connect yang digunakan untuk menyimpan data Twitter yang telah ditransformasi ke Topik. Method start bertugas untuk memulai menulis data, method poll berfungsi untuk mengubah data menjadi list dari objek SourceRecord, dan method stop digunakan untuk menghentikan penulisan data.

Secara sederhana, arsitektur sistem pengumpul data Twitter dengan Kafka ditunjukkan pada Gambar 11.17.



Gambar 11.15. Arsitektur pengumpul data Twitter dengan Kafka.

Cara kerja sistem tersebut dapat dijelaskan sebagai berikut: Kafka mengambil data dari Twitter, mentransformasinya menjadi data berformat semi-terstruktur, lalu menuliskannya ke topik tertentu di Topics. Ketika Twitter Connector mengambil data dari Twitter, hasilnya sudah berformat semi-

terstruktur, seperti JSON, yang lalu ditulis ke Topics. Berdasar *event time trigger* (ketika terjadi penambahan data pada Topics), Structured Streaming membaca dan memproses data dari Topics, yang sudah berformat semi-terstruktur, lalu menyimpan hasilnya ke tabel di HDFS dengan cara yang mirip dengan penambahan rekord pada tabel basisdata relasional .

11.4.2. Perangkat Lunak Pengumpul Data Twitter dengan Kafka

Perangkat lunak yang dikembangkan terdiri dari dua modul, yaitu Kafka Connect dan Structured Streaming. Fungsi pada Structured Streaming dirancang untuk melakukan analisis yang lebih detil terhadap data twit (dibandingkan perangkat lunak dengan Spark Streaming). Komputasi yang dilakukan adalah menghitung rata-rata karakter data twit pada interval tertentu, twit yang *di-like* lebih dari 50 kali, twit yang ditulis oleh seorang *social media influencer*, pengguna Twitter dengan pengikut lebih dari 3000 orang, bahasa apa saja yang sering digunakan, dan twit yang viral. Karena data yang tersimpan sudah berformat semi-terstruktur dan tersimpan dengan skema tertentu, komputasi tersebut dapat dilakukan dengan query layaknya pada basisdata konvensional, yaitu dengan menggunakan fungsi-fungsi agregasi seperti *count* atau *average* dan *group-by*.

Rancangan program Kafka Connect diimplementasikan dengan Java, sedangkan Structured Streaming diimplementasikan dengan Scala untuk Spark dan library SparkSQL.

11.4.3. Eksperimen

Tujuan dari eksperimen ini adalah untuk menganalisis performa, dan menguji apakah sistem pengumpul data stream dengan Kafka dan Structured Streaming mampu menangani data yang terlambat.

Program Kafka Connect dijalankan mulai 29 April 2020. Setelah itu, program Structured Streaming dijalankan pada klaster Spark (dan Kafka) dengan cara berikut ini:

```
spark-submit --class Main --master yarn --num-executors 10 --executor-cores 4 --packages org.apache.spark:spark-sql-Kafka-0-10_2.11:2.4.3 twitter-feature-final.jar slave7:9092,slave5:9092,slave3:9092 twitter-train twitter 1 hours
```

`Spark-submit` digunakan untuk menjalankan perangkat lunak pada klaster Spark, argumen `--class` adalah kelas utama pada perangkat lunak, `--master` digunakan untuk menjalankan Spark diatas Hadoop, `--num-executor` digunakan untuk menggunakan 10 komputer secara paralel, `--executor-cores` digunakan untuk menggunakan 4 memori cores, `--packages` adalah library Kafka yang digunakan untuk transformasi data dan agar Spark dapat terhubung ke Kafka. Hal ini dibutuhkan karena Spark tidak menyediakan library Kafka. Argumen berikutnya adalah nama jar dari perangkat lunak, yaitu `twitter-feature-final.jar` dan alamat Kafka pada klaster. Argumen selanjutnya adalah nama topik Kafka dan ukuran window, di sini Structured Streaming menghitung komputasi setiap satu jam sekali.

Program Kafka Connect dan Structured Streaming dijalankan selama 10 hari dan sebagian dari hasil eksperimen ditunjukkan pada Gambar 11.16 dan Tabel 11.17 dan dijelaskan di bawah ini.

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
driver	master:41845	Active	0	2.4 GB / 384.1 MB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B		Thread Dump
1	slave6:35101	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	0	530620	530625	51.4 h (23 min)	0.0 B	31.1 MB	28.2 MB	stdout	Thread Dump
2	slave5:44419	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	0	275133	275138	30.5 h (24 min)	0.0 B	29.1 MB	28.3 MB	stdout	Thread Dump
3	slave8:34479	Active	0	1.6 GB / 384.1 MB	0.0 B	4	5	0	136941	136946	15.8 h (5.1 min)	0.0 B	9.3 MB	28.4 MB	stdout	Thread Dump
4	slave3:42191	Dead	0	1.4 GB / 384.1 MB	0.0 B	4	0	0	116895	116895	11.4 h (7.7 min)	0.0 B	3.9 MB	26.9 MB	stdout	Thread Dump
5	slave2:34909	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	0	454499	454504	46.6 h (22 min)	0.0 B	32.9 MB	28.3 MB	stdout	Thread Dump
6	slave7:37983	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	0	354094	354099	40.0 h (25 min)	0.0 B	26.4 MB	28.4 MB	stdout	Thread Dump
7	slave1:37071	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	0	514961	514966	50.7 h (22 min)	0.0 B	51.1 MB	28.3 MB	stdout	Thread Dump
8	slave10:46529	Active	0	2.3 GB / 384.1 MB	0.0 B	4	0	0	506149	506149	48.8 h (20 min)	0.0 B	39.8 MB	28.3 MB	stdout	Thread Dump
9	slave4:36595	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	1	473550	473556	49.0 h (23 min)	0.0 B	33.4 MB	28.3 MB	stdout	Thread Dump
10	slave9:45001	Active	0	2.4 GB / 384.1 MB	0.0 B	4	5	1	255848	255854	30.3 h (24 min)	0.0 B	26.1 MB	28.3 MB	stdout	Thread Dump

Gambar 11.16. Performa Structured Streaming dengan Kafka.

Keterangan kolom-kolom pada Gambar 11.16 sama dengan Gambar 11.14. Dibandingkan dengan Gambar 11.14 (laporan Spark Streaming), di sini terdapat lebih banyak task yang aktif. Ini terjadi karena pengumpulan data telah ditangani oleh Kafka, dan Spark hanya fokus membaca data (dari Topik) dan memprosesnya saja. Pada gambar untuk tiap node worker yang aktif, terdapat 5 task yang dijalankan (pada Save 4 dan Slave 9 sempat ada task yang gagal). Namun penggunaan Structured Streaming lebih memakai banyak memori (lihat kolom Storage Memory) yang antara lain digunakan untuk menyimpan data pada Topik.

Pada eksperimen, Structured Streaming berhasil melakukan komputasi untuk memproses data twit pada berbagai tanggal dan jam. Contoh hasil komputasi yang dilakukan oleh Structure Streaming disajikan pada Tabel 11.4

Table 11.3. Contoh hasil agregasi data Twitter

Date	Time	Average Characters	Count	Influenced	Viral	Language
5/3/2020	21:00-22:00	129,398	166589	33381	2221	English
5/4/2020	21:00-22:00	138,818	24081	3244	371	Spanish
5/5/2020	21:00-22:00	133,793	6510	756	87	Portugese
5/6/2020	21:00-22:00	132,636	4075	307	39	Indonesian

Pada Tabel 11.4 bisa disimpulkan bahwa pada jam 21:00-22:00, pesan-pesan twit didominasi oleh pengguna berbahasa Inggris. Sebanyak 166 ribu dengan rata-rata karakter yang digunakan sebanyak 129.39 karakter, 2 ribu twit viral yang di-retweet lebih dari 100 orang, dan 30 ribu twit yang ditulis oleh follower lebih dari 3000 orang. Dari tabel tersebut juga dapat diketahui pengguna dari dengan bahasa apa

yang aktif pada jam-jam tertentu dan pada waktu kapan (kisaran jam berapa) twit memiliki peluang besar untuk menjadi viral.

11.5. Kesimpulan

Dari penelitian dengan hasil yang telah dipaparkan dari subbab-subbab sebelumnya dapat ditarik kesimpulan sebagai berikut:

Spark Streaming mengumpulkan dan mengolah data stream dengan pendekatan *bounded processing*. Jika fungsi untuk memproses batch-batch pada data stream kompleks dan membutuhkan waktu komputasi yang relatif lama, dapat terjadi delay pada pengumpulan data, yang dapat menyebabkan adanya data stream tidak tertangani atau terlewatkan.

Spark Streaming cocok untuk menganalisis data stream secara *real-time*, misalnya untuk mendapatkan trend dari waktu ke waktu. Namun dibutuhkan pengawasan saat program pengumpul data dijalankan, karena *delay* dapat terus membesar dan menyebabkan *time-stamp* tidak akurat. Selain itu, karena dijalankan pada beberapa node worker dan memproses data stream dengan *bounded processing*, Spark Streaming cocok dimanfaatkan untuk mengambil data stream dari sumber yang tertentu atau terbatas.

Kafka memproses data stream dengan pendekatan *unbounded processing*. Dengan pendekatan ini, Kafka lebih akurat dalam menangkap data stream atau lebih menjamin bahwa seluruh aliran data yang masuk dapat ditangkap dan disimpan, namun membutuhkan lebih banyak memori (untuk menyimpan data pada tiap Topik) dibandingkan Spark Streaming. Program pengumpul data dengan Kafka bisa dijalankan selama berhari-hari atau berbulan-bulan tanpa pengawasan. Dengan arsitektur Kafka, kelebihan lainnya dari sistem yang memanfaatkan Kafka adalah data stream dapat dikumpulkan dari beberapa sumber (dan disimpan ke berbagai sistem pula).

Pada sistem pengumpul data yang memanfaatkan Kafka, Structured Streaming dapat digunakan untuk menganalisis data dengan menggunakan fungsi-fungsi agregat pada SQL karena hasil pengumpulan data stream disimpan dalam dataframe yang berstruktur menyerupai tabel basisdata.

Ucapan Terima Kasih

Ucapan terima kasih ditujukan kepada Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan yang telah mendanai penelitian ini melalui skema Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT), tahun anggaran 2020, dengan nomor kontrak III/LPPM/2020-04/107-PE-S.

Referensi

Akidau, T., Chernyak, S., & Lax, R. (2018). Streaming 101. In T. Akidau, *Streaming Systems*. Sebastopol: O'Reilly.

- Akida, T., Chernyak, S., & Lax, R. (2018). The What, Where, When, and How of Data Processing. In T. Akida, *Streaming Systems*. Sebastopol: O'Reilly.
- Karau, H., Konwinski, A., Patrick, W., & Zaharia, M. (2015). Programming with RDDs. In H. Karau, A. Konwinski, W. Patrick, & M. Zaharia, *Learning Spark*. Sebastopol: O'Reilly.
- Karau, H., Konwinski, A., Patrick, W., & Zaharia, M. (2015). Spark Streaming. In H. Karau, A. Konwinski, W. Patrick, & M. Zaharia, *Learning Spark*. Sebastopol: O'Reilly.
- Karau, H., Kowinski, A., Wendell, P., & Zaharia, M. (2015). Introduction to Data Analysis with Spark. In H. Karau, *Learning Spark*. Sebastopol: O'Reilly.
- Miner, D., & Shook, A. (2013). *Map Reduce Design Pattern*. Sebastopol: O'Reilly.
- Narkhede, n., Shapira, G., & Palino, T. (2017). Kafka Consumers: Reading Data from Kafka. In N. Narkhede, *Kafka: The Definitive Guide*. Sebastopol: O'Reilly.
- Narkhede, n., Shapira, G., & Palino, T. (2017). Kafka Producer: Writing Messages. In N. Narkhede, *Kafka: The Definitive Guide*. Sebastopol: O'Reilly.
- Narkhede, n., Shapira, G., & Palino, T. (2017). Meet kafka. In H. Karau, A. Konwinski, W. Patrick, & M. Zaharia, *Kafka: Definitive Guide*. Sebastopol: O'Reilly.
- Spark Streaming Programming Guide*. (n.d.). Retrieved from Spark:
<https://spark.apache.org/docs/latest/streaming-programming-guide.html#a-quick-example>
- Spark Web-UI*. (n.d.). Retrieved from spark.apache.org: <https://spark.apache.org/docs/3.0.0-preview/web-ui.html>
- White, T. (2008). *Hadoop: The Definitive Guide*. Sebastopol: O'Reilly.

Halaman ini sengaja dikosongkan

Bab 12 Algoritma Pengelompokan k-Means Paralel untuk Memproses Big Data

Oleh:

Veronica S. Moertini

12.1. Pengelompokan Data

Dalam konteks penambangan data (data mining), klaster didefinisikan sebagai sekelompok objek-objek yang memiliki kemiripan yang tinggi (Han et al., 2012). Objek-objek pada sebuah klaster tertentu memiliki kemiripan yang rendah dengan objek-objek pada klaster-klaster yang lain. Di sini, objek dapat berupa orang, binatang, barang/benda, foto/citra, dokumen, lagu, video, transaksi pengguna di bank, toko, e-commerce, hasil perekaman sensor cuaca pada suatu saat, transaksi pengambilan matakuliah pada satu semester, "klik" pengguna website, dll. Dalam konteks data mining, sebuah objek akan direpresentasikan menjadi "sebuah elemen" pada himpunan data (dataset). Jadi, himpunan data berisi representasi dari objek-objek.

Analisis klaster merupakan salah satu teknik pada data mining yang utama. Tujuannya adalah untuk menemukan klaster-klaster dari himpunan data secara otomatis atau semi-otomatis menggunakan algoritma clustering. Berdasarkan pendekatan dan/atau konsep yang diadopsi, teknik clustering dapat dikategorikan ke dalam metoda yang berbasis partisi, hirarki, dan densitas/kerapatan, grid, model dan konstrain. Setiap kategori teknik memiliki kelebihan dan kekurangan tersendiri. Kelebihan ini terkait dengan himpunan data yang bagaimana yang ditangani, klaster bagaimana yang dibutuhkan juga kecepatan dalam memproses himpunan data. Misalnya, untuk himpunan data yang ditengarai bahwa klaster-klasternya akan "terpisah" satu sama lain, maka yang cocok digunakan adalah teknik partisi. Sedangkan untuk himpunan data yang ditengarai akan menghasilkan klaster yang di "dalamnya" terdapat klaster yang lain, maka yang cocok adalah teknik yang berbasis kerapatan.

Beberapa contoh algoritma pada tiap kategori, diberikan di bawah ini:

- Partisi: k-Means dan k-Medoid
- Hirarki: Diana dan Agnes
- Kerapatan: DBSCAN dan OPTICS
- Grid: WaveCluster dan STING,
- Model: EM, SOM dan COBWEB.

Dari seluruh algoritma-algoritma di atas, k-Means termasuk yang populer dan banyak dimanfaatkan.

12.2. Manfaat Analisis Klaster

Beberapa manfaat dari analisis klaster dari himpunan data antara lain (Han et al., 2012, URL-cluster-1):

1. Di bidang bisnis: Pengelompokan pelanggan berdasarkan pembelian mereka untuk keperluan pemasaran (*targeted marketing*). Hasil pengelompokan yang disasar: Tiap klaster pelanggan memiliki pola atau ciri-ciri tertentu yang lalu cocok untuk ditawari produk-produk tertentu yang cocok atau dibutuhkan pelanggan. Dengan begitu, pemasaran menjadi lebih efektif.
2. Di bidang pemanfaatan lahan: Pengelompokan citra hasil penginderaan jauh satelit dapat menghasilkan area dengan pemanfaatan lahan yang serupa (misalnya: perumahan, sawah, hutan, pertambangan, dll). Hasilnya dapat dimanfaatkan untuk pemantauan pemanfaatan lahan dari waktu ke waktu atau penyusunan kebijakan terkait dengan pemanfaatan lahan.
3. Di bidang pengarsipan: Pengelompokan dokumen-dokumen berdasar isi (semantik) dari dokumen. Hasil dari pengelompokan dapat dimanfaatkan pada pencarian dokumen, misalnya untuk pencarian dengan kata-kunci tertentu, yang dimunculkan adalah dokumen-dokumen pada klaster tertentu.
4. Secara umum, analisis klaster juga dimanfaatkan pada pengenalan pola pada berbagai jenis himpunan data (dokumen, citra, audio, video, dll) dan analisis citra (segmentasi objek-objek yang terdapat di citra), dll.

Pada contoh nomor 1 di atas, disebut bahwa tiap klaster pelanggan memiliki pola atau ciri-ciri tertentu. Pola tersebut contohnya bagaimana? Interpretasi pola dari sebuah klaster dapat berupa deskripsi: "Pengguna band-width internet yang tinggi, umurnya belasan tahun, frekuensi pembelian ayam geprek tinggi dan frekuensi pemakaian ojol tinggi". Adapun deskripsi klaster yang lain: "Pengguna band-width internet yang tinggi, umurnya dua puluhan tahun, frekuensi pembelian bakso tinggi dan pemakaian ojol sedang". Informasi ini lalu dapat digunakan untuk "*targeted marketing*", misalnya memberikan iklan kepada pengguna umur dua puluhan yang tidak sering naik ojol untuk menikmati bakso di restoran/warung tertentu.

Sebagai fungsi pada penambangan data, analisis klaster berperan sebagai alat (*tool*) untuk menggali "informasi yang terpendam" (*insight*) dari distribusi data, misalnya deskripsi klaster di atas.

12.3. Algoritma Pengelompokan k-Means Non-Paralel

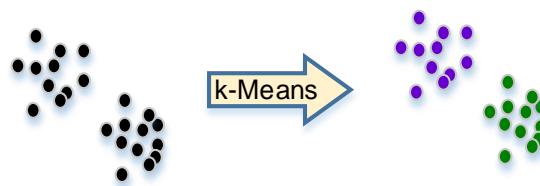
Algoritma k-Means banyak digunakan dan populer, karena itu di bab ini dibahas sebagai salah satu contoh algoritma pengelompokan. Algoritma ini menerima masukan himpunan data berformat vektor, tiap objek memiliki fitur-fitur dan tiap fitur memiliki nilai yang merepresentasikan objek tersebut. Misalnya, objek orang memiliki fitur pemanfaatan-band-widtdh (*pbw*), *umur*, frekuensi-pembelian-bakso (*fpb*), frekuensi-naik-ojol (*fno*). Contoh nilai fitur sebuah objek adalah: $pbw = 15$ (gigabyte/bulan), $umur = 23$, $fpb = 12$ (per bulan), $fno = 20$ (per bulan).

Himpunan data yang berisi objek-objek tersebut dapat dipandang sebagai sebuah “tabel”, dimana kolom-kolom merupakan fitur-fitur dan baris merepresentasikan sebuah objek. Contoh himpunan data diberikan pada Tabel 12.1.

Tabel 12.1. Contoh himpunan data untuk masukan k-Means.

Objek ke	pwb	umur	fpb	fno
1	15	23	12	20
2	11	18	9	14
3	2	45	4	0
4	4	50	1	1
5	15	23	12	12
6	1	15	9	13
7	21	45	4	1
8	14	30	1	2
9	11	19	9	3

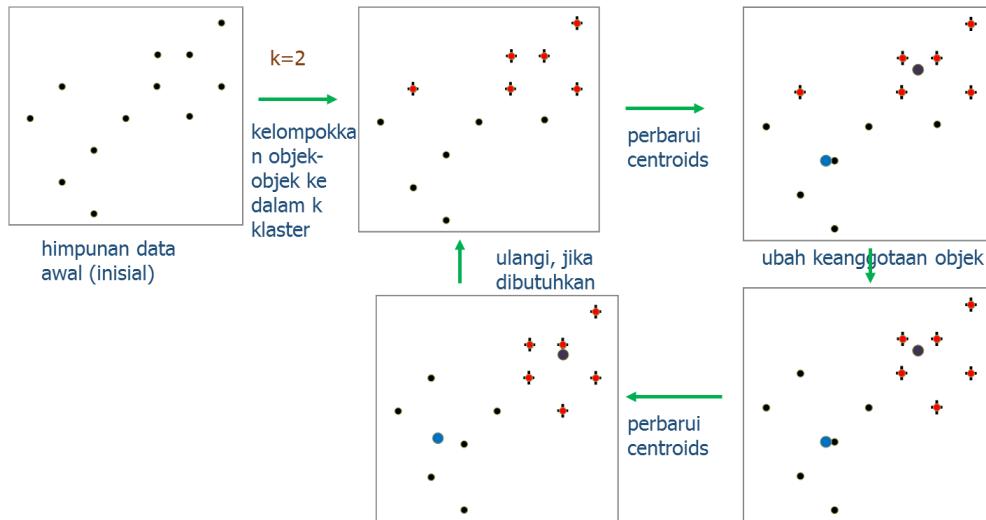
Apabila jumlah fitur hanya dua, misalnya *pwb* dan *umur* saja dan objek-objek dikelompokkan menjadi dua saja ($k = 2$), maka masukan dan keluaran dari algoritma k-Means dapat diilustrasikan sebagaimana ditunjukkan pada Gambar 12.1. Objek-objek yang dikelompokkan berada di sebelah kiri panah terlihat memiliki distribusi yang “terpisah” menjadi 2. Setelah diumpulkan ke k-Means, dihasilkan 2 klaster (ungu dan hijau).



Gambar 12.1. Ilustrasi masukan dan hasil k-Means dengan $k = 2$.

Bagaimana cara kerja algoritma k-Means?

Sejatinya, selain himpunan data, k-Means juga membutuhkan masukan jumlah klaster yang ingin dibentuk, k pada k-Means (selain itu, juga inisial centroids atau titik-titik pusat awal/inisialisasi, jumlah pengulangan eksekusi perintah, yang dikenal dengan istilah iterasi, maksimum dan sebuah nilai konstanta yang digunakan untuk penentuan kapan iterasi dihentikan, ϵ). Untuk ilustrasi algoritma k-Means, pada Gambar 12.2 diberikan contoh beberapa objek yang akan dikelompokkan menjadi dua klaster ($k = 2$).

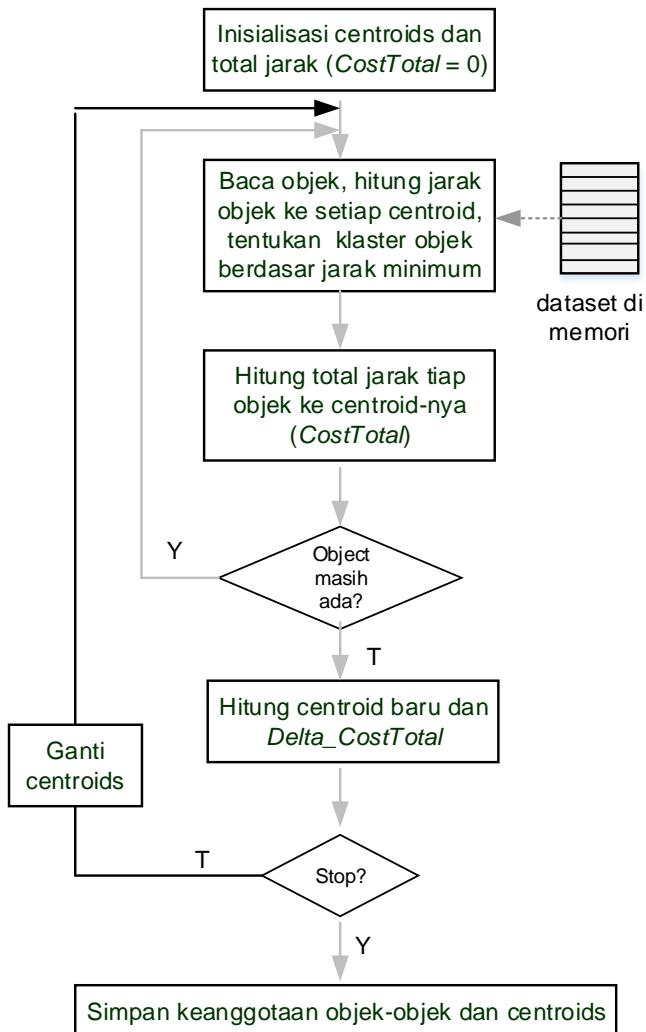


Gambar 12.2. Langkah-langkah komputasi k-Means.

Langkah-langkah k-Means sebagaimana ditunjukkan pada Gambar 12.3 adalah:

1. Untuk setiap objek yang terdapat pada himpunan data: jarak objek ke tiap centroid (titik pusat) dihitung. Objek akan dimasukkan ke klaster dengan jarak terdekat.
2. Jarak tiap objek ke centroid-nya akan dijumlahkan (disimpan pada variabel *CostTotal*).
3. Jika *CostTotal* pada iterasi sekarang dikurangi *CostTotal* dari iterasi sebelumnya (*Delta_CostTotal*) lebih kecil atau sama dengan *eps*, maka iterasi dihentikan. Jika tidak, maka centroids baru dihitung dan langkah kembali ke nomor 1 dengan menggunakan centroids yang baru.

Perhitungan centroids baru dilakukan dengan menghitung rata-rata nilai setiap fitur dari semua objek yang berada di dalam klaster yang sama.



Gambar 12.3. Algoritma k-Means asli (non-paralel).

Cara untuk menghitung jarak antar objek (atau jarak objek ke centroids) bergantung kepada tipe nilai dari atribut-atribut objek. Atribut objek dapat bernilai biner (true/false) atau kategorial (hijau, merah, hitam, dll) atau numerik/angka. Rumus untuk menhitung jarak antar objek harus dipilih yang sesuai. Bahasan yang lengkap tentang cara perhitungan antar objek dapat dipelajari di (Han et al., 2012).

Jika semua atribut objek bertipe numerik, rumus yang sering digunakan adalah jarak Euclidean. Dalam hal ini, objek dapat direpresentasikan menjadi vektor (contoh \mathbf{x} dan \mathbf{y}) yang memiliki atribut ke-1 s/d ke-n. Jika $\mathbf{x} = (x_1, x_2, \dots, x_n)$ dan $\mathbf{y} = (y_1, y_2, \dots, y_n)$, maka jarak \mathbf{x} ke \mathbf{y} :

$$d(x, y) = \sqrt{\sum_{i=1}^{i=n} (x_i - y_i)^2} \quad (1)$$

Adapun $CostTotal (J)$ direpresentasikan sebagai:

$$J = \sum_{i=1}^c J_i = \sum_{i=1}^c (\sum_{k, x_k \in G_i} d(x_k - c_i)) \quad (2)$$

Pada rumus 2 di atas, c = jumlah klaster, x_k = objek ke k , c_i = centroid ke i , G_i = klaster ke i . Sedangkan $d(x_k - c_i)$ = jarak dari objek k ke centroid-nya. Perhitungan c_i dilakukan dengan rumus berikut ini:

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \quad (3)$$

dimana $|G_i|$ menyatakan jumlah objek (anggota) pada klaster ke- i , x_k = objek ke k yang menjadi anggota klaster ke- i . Rumus itu menyatakan bahwa nilai fitur dari objek-objek yang berada dalam klaster yang sama dirata-rata dan disimpan sebagai centroid klaster tersebut.

Evaluasi Hasil Pengelompokan

Lantas bagaimana kita tahu bahwa hasil pengelompokan “sudah bagus” atau objek-objek terkelompok dengan benar (objek-objek yang berdekatan berada dalam satu klaster, dan tiap-tiap klaster saling “berjauhan”)?

Terdapat dua pendekatan dalam mengukur kualitas hasil pengelompokan, yaitu secara internal dan eksternal. Bahasan detil tentang metoda untuk mengevaluasi hasil pengelompokan dapat dipelajari di buku (Han et al., 2012). Prinsip dari 2 metoda yang dapat digunakan diberikan di bawah ini:

1. Pengukuran *internal*: Salah satu pendekatannya dilakukan dengan menghitung koefisien Silhouette untuk tiap objek, $s(i)$. Nilai $s(i)$ ini berada pada rentang -1 s/d 1. Makin dekat ke 1, objek makin terkelompok dengan baik. Jika $s(i)$ bernilai negatif, objek terkelompok “dengan salah”. Untuk mengukur kualitas kelompok secara keseluruhan, nilai $s(i)$ dari semua objek dirata-rata, lalu disimpan sebagai s . Jika s makin mendekati 1, maka kualitas hasil clustering makin baik. Namun demikian, jika jumlah objek tidak terlalu banyak dan semua nilai $s(i)$ masih mungkin untuk divisualisasi (diplot), maka seluruh koefisien dapat diplot untuk dianalisis (misalnya untuk diamati prosentase objek-objek dengan nilai $s(i)$ negatif).
2. Pengukuran *eksternal*: Untuk mengukur kualitas hasil pengelompokan secara eksternal, dibutuhkan himpunan data “ground-truth” yang berisi objek-objek yang sudah “terlabeli” dengan kelompok-kelompok yang sebenarnya. Klaster dari tiap objek hasil dari clustering, dibandingkan dengan label tersebut, lalu dihitung berapa jumlah objek yang terkelompok dengan benar dan salah. Dari sini, lalu dapat hitung *purity* (kemurnian) dari tiap klaster dan/atau akurasinya. Makin besar nilai *purity* dan akurasinya, makin baik hasil pengelompokan. Metoda ini lebih cocok digunakan jika hasil pengelompokan (misalnya, centroids) dijadikan model, dimana model lalu digunakan untuk menentukan kelompok dari objek lain (yang tidak disertakan dalam pembuatan model). Namun demikian, teknik clustering lebih banyak digunakan pada *unsupervised learning*, dimana himpunan

data “ground-truth” tidak tersedia. (Jika data “ground-truth” tersedia, umumnya digunakan teknik klasifikasi.)

Sebagaimana telah dibahas sebelumnya, algoritma k-Means mensyaratkan pengguna untuk menentukan jumlah kelompok (k). Dengan kata lain, pengguna harus memasukkan nilai k beserta himpunan data sebagai masukan dari k-Means. Lalu bagaimana cara memilih k yang tepat? Pemilihan tersebut dapat dilakukan dengan metoda sebagai berikut:

1. Jika ukuran himpunan data tidak terlalu besar (misalnya sampai puluhan megabyte), pengelompokan dapat dilakukan berulang-ulang dengan nilai k yang berbeda-beda. Koefisien Silhoutte (tiap objek dan rata-ratanya) dari hasil pengelompokan, lalu dibandingkan. Jumlah kelompok terbaik dipilih dari k yang memberikan nilai koefisien Silhoutte (rata-rata) yang terbesar.
2. Jika ukuran himpunan data besar dan pengelompokan yang berulang-ulang di atas dinilai tidak efisien (atau tidak *feasible*), dapat dilakukan *sampling* secara acak terhadap himpunan data untuk menghasilkan sub-himpunan data dengan ukuran lebih kecil namun dipandang merepresentasikan himpunan data aslinya. Kemudian, sub-himpunan data ini dikelompokkan berulang-ulang dengan nilai k yang berbeda seperti cara nomor 1. Nilai k yang diperoleh ini lalu digunakan untuk mengelompokkan himpunan data yang sebenarnya.

12.4. Algoritma k-Means Paralel untuk Big Data

Sebagaimana telah disebutkan, himpunan data yang dikelompokkan dapat berukuran besar atau sangat besar dan mencapai ukuran ratusan (atau lebih) gigabyte. Algoritma k-Means non-paralel (yang orisinil) dirancang untuk dijalankan pada sebuah komputer (sebuah core), dengan memanfaatkan memori di komputer itu saja. Kapasitas memori yang dapat digunakan untuk menyimpan himpunan data biasanya ditentukan oleh sistem operasi atau konfigurasi tertentu (misalnya, kalau menggunakan Java Virtual Memory, yang disingkat JVM, maka memori maksimal ditentukan oleh konfigurasi pada JVM). Dengan demikian, k-Means non-paralel tidak dapat menangani himpunan data yang berukuran lebih besar dari ruang memori yang dialokasikan untuk k-Meas.

Untuk menangani data dengan ukuran sangat besar, k-Means non-paralel telah dikembangkan untuk lingkungan sistem tersebar Hadoop dan Spark yang dibahas di bawah ini. Jika pada algoritma k-Means non-paralel yang digunakan untuk mengelompokkan himpunan data berukuran kecil sampai besar (sejauh yang dapat ditangani komputasi pada sebuah komputer) nomor klaster keanggotaan tiap objek direkam dan dijadikan sebagai keluaran algoritma, maka pada pengelompokan big data tidak demikian. Karena jumlah objek dapat mencapai jutaan bahkan milyardan, informasi tentang klaster dari tiap objek menjadi tidak atau kurang bermakna. Hasil akhir utama dari k-Means paralel (untuk big data) adalah centroids dari klaster-klaster. Centroids ini kemudian dapat dijadikan model dan digunakan untuk mengelompokkan objek-objek lain (yang tidak digunakan untuk menghitung centroids).

Pada subbab ini dibahas algoritma k-Means paralel untuk sistem big data Hadoop dan Spark. Bahasan mengenai Hadoop dan Spark dapat dibaca pada Bab 10 dan Bab 11.

12.4.1. Algoritma k-Means Paralel untuk Lingkungan Sistem Hadoop

Sebagaimana ditunjukkan pada Gambar 12.2 (langkah-langkah) dan Gambar 12.3 (*flowchart*), proses pengelompokan objek-objek ke klaster dilakukan berulang-ulang (secara iteratif) sampai “kondisi stabil” dicapai, dimana tidak ada atau mayoritas objek sudah tidak berpindah klaster lagi. Jumlah perulangan atau iterasi ini dipengaruhi oleh centroid inisial, yang pada iterasi ke-1 digunakan untuk menentukan objek-objek masuk ke kelompok yang mana. Makin dekat centroid inisial ke centroid yang sebenarnya, yang dicapai pada kondisi stabil, maka makin sedikit jumlah iterasi yang perlu dijalankan untuk mencapai kondisi stabil. Sehubungan dengan hal ini, beberapa teknik atau algoritma untuk menentukan/menghitung centroid inisial telah dikembangkan.

Salah satu cara penentuan centroid awal adalah dengan mengelompokkan sebagian himpunan data yang diperoleh melalui teknik sampling secara acak. Karena data hasil sampling berukuran jauh lebih kecil daripada data asli, maka komputasi untuk mendapatkan centroid final pada sampel data dapat dilakukan dengan lebih cepat. Centroid dari hasil pengelompokan dengan sampel data ini, kemudian dijadikan centroid inisial untuk mengelompokkan keseluruhan himpunan data.

Untuk mengelompokkan big data, penentuan centroid inisial ini penting, karena tiap iterasi cost/biaya komputasi yang dibutuhkan besar (terkait dengan penggunaan CPU atau core dan memori komputer yang dapat berjumlah sangat banyak).

Algoritma k-Means pada sistem tersebut Hadoop dapat dijelaskan dengan contoh pada Gambar 12.4 sebagai berikut (Zhao, Ma & Q. He, 2009; Moertini & L. Venica, 2017):

Perhitungan centroid inisial dilakukan dengan mengelompokkan sample dari big data yang tersimpan sebagai blok-blok pada HDFS. Centroid hasil pengelompokan ini lalu dijadikan centroid inisial.

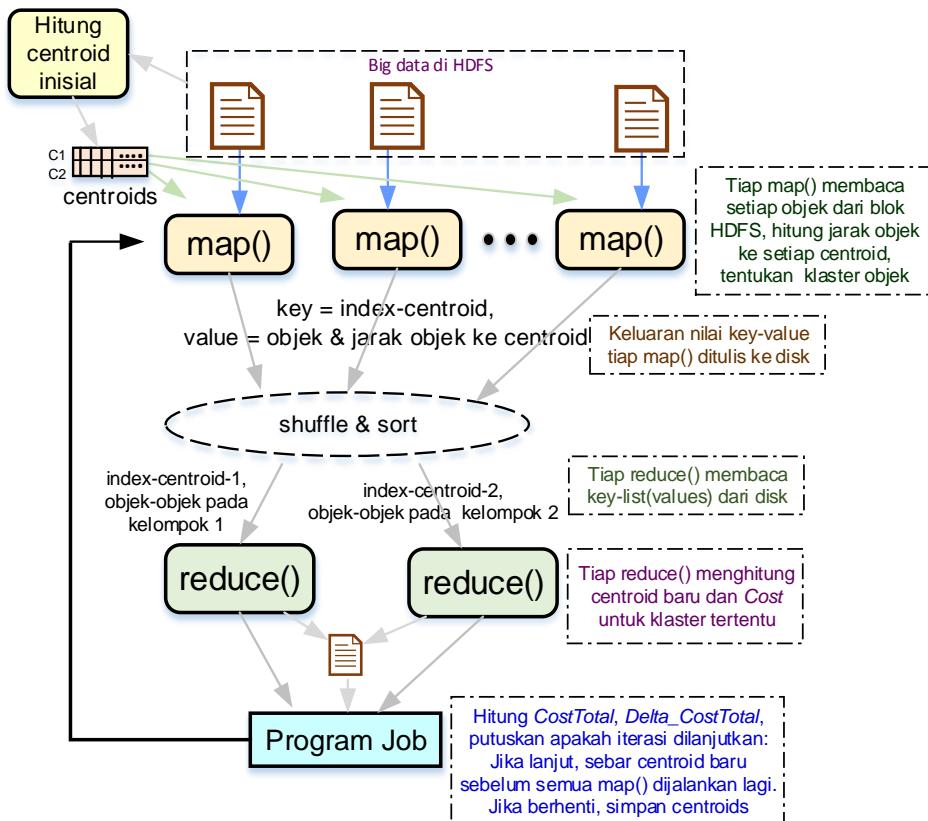
Oleh program Job, nilai-nilai inisial centroid disebar ke semua node data (slave node) yang akan menjalankan `map()`. Setelah itu, `map()` dijalankan untuk membaca baris-demi baris (atau objek demi objek) pada blok data lokal (blok data yang tersimpan di node data tempat `map()` dijalankan). Jarak dari objek ini ke setiap centroid (titik pusat) dihitung. Objek akan “dimasukkan” ke kelompok dengan jarak terdekat. Fungsi `map()` lalu mengeluarkan pasangan key-value, dimana key berisi indeks dari centroid (misalnya 0, 1, 2, dst), value berisi objek (dapat berformat vektor) dan nilai jarak dari objek ke centroid-nya.

Hadoop lalu melakukan proses yang dinamakan “shuffle dan sort”. Keluaran `map()` yang memiliki nilai key yang sama akan dikumpulkan menjadi satu, lalu pasangan nilai value (objek dan jarak) dari key-key tersebut diurutkan dalam bentuk/format list.

Selanjutnya, setiap `reduce()` menerima satu atau lebih dari nilai key beserta pasangannya yang berupa list value, yang berisi objek-objek yang terkelompok pada kelompok bermotor key (di sini, key merepresentasikan nomor kelompok, misalnya kelompok 0, 1, 2, dst).

Setelah reduce() menerima nilai objek-objek dari sebuah key, reduce() akan menghitung titik pusat (centroid) baru untuk kelompok dengan nomor key tersebut dan jumlah total jarak objek-objek ke masing-masing centroid.

Selanjutnya, program Job akan menjumlahkan jarak total keluaran tiap reducer menjadi jarak total (*CostTotal*). Jarak total digunakan untuk mencek apakah kelompok baru yang terbentuk sudah "stabil" (dibandingkan kelompok yang terbentuk pada iterasi sebelumnya). Nilai epsilon akan dihitung dengan mengurangi total jarak pada iterasi sekarang dengan total jarak pada iterasi sebelumnya (*Delta_CostTotal*). Jika nilai *Delta_CostTotal* sudah lebih kecil dari nilai yang didefinisikan atau jumlah iterasi sudah mencapai iterasi maksimum, maka iterasi akan dihentikan dan hasil komputasi centroids yang dilakukan oleh reducer() menjadi centroids versi final lalu "dikeluarkan" atau disimpan sebagai file HDFS. Jika belum, nilai centroids yang baru akan "disebar" ke node-node data, lalu fungsi map() dan reduce() dijalankan lagi.



Gambar 12.4. Ilustrasi algoritma k-Means paralel pada Hadoop dengan jumlah kelompok = 2.

Untuk mengurangi trafik pada jaringan pada proses shuffle dan sort, dapat ditambahkan fungsi `combine()`. Fungsi ini berperan dalam mengumpulkan value-value dengan nilai key yang sama secara lokal (di sebuah

data node). Keluaran `combine()` berupa pasangan key dan value-value yang lalu “diumpulkan” ke proses shuffle dan sort. Dengan memanfaatkan `combine()`, jumlah pasangan key-value yang dikirim ke proses shuffle dan sort berkurang, sehingga mengurangi trafik pada jaringan Hadoop dan dapat mempercepat eksekusi algoritma k-Means.

Hal yang menjadi kelemahan pada framework MapReduce Hadoop adalah:

1. Pada setiap iterasi, setiap `map()` membaca blok HDFS dari disk.
2. Pada tahap shuffle dan sort, keluaran dari setiap `map()` ditulis pada disk. Masing-masing `reduce()` yang menangani nilai key tertentu lalu membaca nilai-nilai value dari disk.
3. Pada akhir iterasi: centroid hasil komputasi pada akhir iterasi (sebagai keluaran `reduce()`) juga selalu disimpan di disk.

Pembacaan dan penulisan data berukuran sangat besar dari dan ke disk secara berulang-ulang tersebut menyebabkan “biaya” yang tinggi pada algoritma k-Means paralel pada Hadoop. Proses clustering big data menjadi tidak efisien.

12.4.2. Algoritma k-Means Paralel untuk Lingkungan Sistem Spark

Sebagaimana telah dijelaskan pada Bab 10, Spark dikembangkan untuk memfasilitasi algoritma-algoritma yang membutuhkan komputasi intensif dan iteratif. Salah satu algoritma tersebut adalah k-Means, yang sudah menjadi fungsi pada library MLlib maupun ML pada Spark (Karau et al., 2015; Karau & Warren, 2017). Pada Spark, himpunan big data (dalam bentuk blok-blok HDFS) dibaca dari disk lalu disimpan di memori-memori komputer yang bertindah sebagai worker dalam bentuk partisi RDD (*Resilient Distributed Dataset*). Jika digunakan konfigurasi default Spark, tiap blok HDFS akan dibuatkan 1 RDD. RDD tersebut dapat disimpan di memori para worker terus selama aplikasi dijalankan (dengan menggunakan fungsi *persist*).

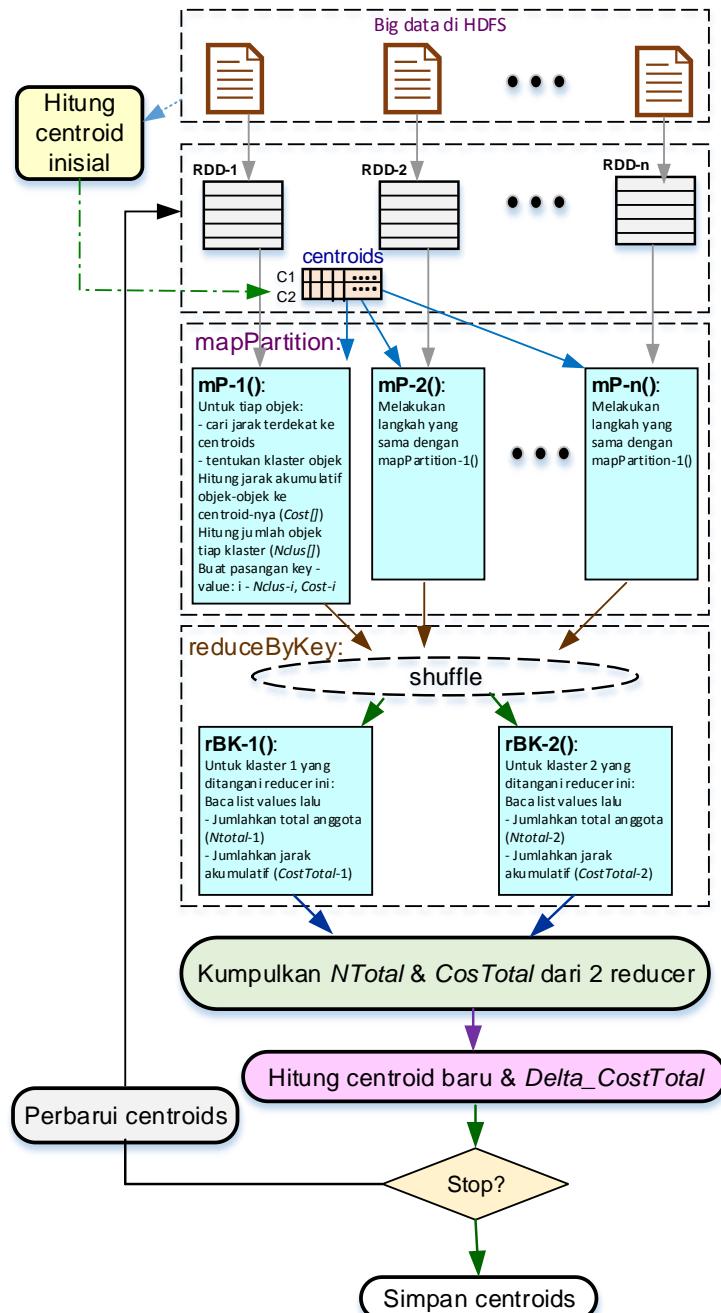
Langkah-langkah algoritma k-Means paralel pada Spark (lihat Gambar 12.5):

Analogi dengan k-Means paralel pada Hadoop, perhitungan centroid inisial dilakukan dengan mengelompokkan sample dari big data (komputasi dapat dilakukan dengan k-Means non-paralel). Centroid hasil pengelompokan ini lalu dijadikan centroid inisial dan “disebar” ke para worker.

Tiap `mapPartition()` pada *worker* membaca tiap objek dari sebuah RDD, menentukan kelompoknya, menghitung jarak akumulasi (*cost*) dari tiap objek ke centroid-nya, lalu mengeluarkan pasangan key-value, dimana key = indeks klaster, value = jumlah objek dan cost.

Pemanggilan `reduceByKey()` pada tiap *worker* mengakibatkan Spark melakukan proses shuffle sedemikian rupa sehingga setiap `reduceByKey()` hanya menangani nilai key tertentu beserta seluruh value untuk key tersebut. Komputasi yang dilakukan `reduceByKey()` adalah menjumlahkan total anggota (*NTotal*) dan jarak akumulatif tiap klaster (*CostTotal*).

Hasil perhitungan tiap `reduceByKey()` pada semua worker lalu dikumpulkan, kemudian centroids yang baru dihitung. Jika *CostTotal* saat ini dikurangi *CostTotal* pada iterasi sebelumnya bernilai lebih kecil atau sama dengan batas minimum (*Eps*) atau iterasi maksimum sudah dicapai, maka iterasi dihentikan. Jika kondisi tersebut tidak dipenuhi, maka centroids diperbarui (diganti dengan hasil perhitungan pada iterasi ini) dan “disebar” ke para worker, lalu iterasi diulangi lagi.



Gambar 12.5. Algoritma k-Means paralel pada Spark dengan kasus jumlah kelompok = 2 (keterangan: $mP = mapPartition$, $rBK = reduceByKey$).

12.5. Pengembangan Algoritma k-Means Paralel

12.5.1. Algoritma k-Means untuk Lingkungan Sistem Hadoop

Pada Subbab 12.4 dibahas algoritma k-Means yang dikembangkan untuk lingkungan Hadoop, dimana pada akhir komputasi menghasilkan model yang berupa centroids atau “titik-titik tengah” dari klaster-klaster. Model tersebut lalu dapat digunakan untuk memprediksi kelompok dari objek yang tidak menjadi bagian dari himpunan data yang digunakan untuk menghitung centroids atau belum digunakan pada proses clustering.

Sebagaimana dipaparkan pada (Han et al., 2012), analisis klaster merupakan salah satu metoda deskriptif. Sesuai dengan namanya, hasil akhir metoda ini adalah “deskripsi dari himpunan data” yang berupa nilai-nilai ringkasan statistik. Nilai-nilai itu lalu dapat dievaluasi dan/atau dibandingkan satu dengan yang lain, dan apabila didapatkan ada nilai-nilai yang “menarik” lalu dapat dijadikan pola-pola klaster yang berharga atau bermanfaat (contoh pola sudah diberikan pada Subbab 12.2).

Nilai-nilai ringkasan statistik yang berpotensi untuk dijadikan pola-pola klaster tersebut apa saja? Pada (Tsitsis dan Chorianopoulos, 2009; Chius dan Tavella, 2011) dirumuskan bahwa nilai ringkasan yang dihitung dari tiap klaster dapat berupa gabungan dari:

- Centroids (nilai rata-rata tiap atribut/fitur)
- Nilai minimum, maximum, standard deviasi dari nilai atribut
- Persentase objek yang memiliki nilai atribut tertentu
- Jumlah objek yang menjadi anggota.

Berikut ini diberikan sebuah contoh ringkasan statistik. Sebuah bank bermaksud meningkatkan pemasaran produk-produk yang dijual (misalnya berbagai kredit) dengan menarget calon-calon nasabah yang potensial. Bank memiliki data nasabah-nasabah beserta kredit yang diambil. Data tersebut dapat disiapkan untuk dikelompokkan. Di sini, tiap objek mewakili satu nasabah. Dari hasil penyiapan data, objek-objek pada himpunan data yang dikelompokkan memiliki 5 fitur, yaitu: Nilai kredit (NK), umur (U), status nikah (SN), jumlah anak (JA) dan pendapatan (P). Setiap nilai fitur “dinormalisasi” dan memiliki nilai antara 0 s/d 1. (Keterangan: Normalisasi, transformasi nilai fitur agar memiliki rentang nilai yang sama ini dimaksudkan agar fitur-fitur berkontribusi setara pada perhitungan jarak objek ke centroidnya atau tidak ada fitur-fitur yang mendominasi).

Klaster-1:

- Centroids (rata-rata nilai fitur): NK = 0.85, U = 0.81, SN = 0.72, JA = 0.79, P = 0.97
- Nilai minimum: NK = 0.79, U = 0.71, SN = 0.62, JA = 0.68, P = 0.88
- Nilai maksimum: NK = 0.95, U = 0.91, SN = 0.89, JA = 0.96, P = 1.0
- Deviasi: NK = 0.09, U = 0.07, SN = 0.12, JA = 0.01, P = 0.08
- Jumlah objek: 302

Klaster-2:

- Centroids (rata-rata nilai fitur): NK = 0.62, U = 0.31, SN = 0.32, JA = 0.09, P = 0.93
- Nilai minimum: NK = 0.41, U = 0.26, SN = 0.0, JA = 0.0, P = 0.68
- Nilai maksimum: NK = 0.75, U = 0.39, SN = 0.39, JA = 0.2, P = 0.99
- Deviasi: NK = 0.18, U = 0.06, SN = 0.2, JA = 0.01, P = 0.18
- Jumlah objek: 191

Dari kedua contoh klaster di atas, Klaster-1 merepresentasikan nasabah-nasabah umur sekitar tengah baya, menikah, punya penghasilan dengan rentang besar dan mengambil kredit yang besar pula. Sebaliknya, Klaster-2, umur sekitar 30-an, sebagian menikah, hanya sedikit yang sudah memiliki anak, memiliki pendapatan besar, dan mengambil kredit bernilai sedang. Untuk tujuan pemasaran produk kredit bank, Klaster-1 dipandang menjadi pola yang berharga. Ditengarai bahwa kredit yang diambil nasabah-nasabah tengah baya tersebut ternyata digunakan untuk membuka dan menjalankan usaha mandiri. Berdasar temuan pola berharga ini, bank lalu dapat menarget orang-orang tengah baya dengan penghasilan tinggi, menikah dan punya anak untuk ditawari produk kredit usaha (untuk berwirausaha).

Pada pengembangan k-Means paralel untuk Hadoop, standar deviasi pada klaster dihitung secara aproksimasi. Perhitungan deviasi membutuhkan komputasi pengurangan nilai variabel dengan rata-rata nilai variabel. Ini membutuhkan 2 iterasi: Iterasi pertama untuk menghitung rata-rata, yang kedua untuk mengurangi selisih nilai variabel dengan rata-rata. Di sini, variabel adalah fitur objek. Dalam konteks big data, jumlah objek mencapai jutaan atau bahkan milyardan sehingga komputasi menjadi mahal, terlebih dengan mempertimbangkan bahwa komputasi k-Means sendiri sudah bersifat iteratif.

Karena itu, komputasi aproksimasi standar deviasi, dilakukan dengan mengambil nilai rata-rata pada iterasi sebelumnya. Dengan demikian, tidak ada iterasi tambahan pada k-Means pada Hadoop.

Pada Gambar 12.4 telah ditunjukkan algoritma k-Means paralel yang berbasis MapReduce untuk Hadoop. Untuk menambahkan komputasi ringkasan statistik di tiap klaster, yang dilakukan adalah menambahkan komputasi pada reduce() dengan rancangan algoritma yang diberikan di bawah ini (Moertini & Venica, 2016).

Algoritma: reduce pada k-Means paralel

Input: *key, listVal, prevcenters* dimana *key* = indeks klaster, *listVal* = list value yang terurut, *prevcenters* = centroids

Output: pasangan *<key', value'*, *key'* = indeks klaster *value'* = string gabungan dari *centers[]* (centroid semua klaster), jumlah objek pada tiap klaster, *countObj*, nilai minimum, maximum, rata-rata, deviasi standar tiap atribut pada klaster, *minAtrVal*, *maxAtrVal*, *StdCluster*; cost untuk tiap klaster, *J*

Step:

1. Inisialisasi *minAtrVal[], maxAtrVal[], SumDiffAtrPrevCenter[], SumAtr[], StdDev[][][], centers[]*
2. *countObj* = 0; *J* = 0;
3. While(*ListVal.hasNext()*)
 4. Ambil nilai-nilai atribut dan *dist* dari *value*
 5. Tiap nilai atribut digunakan untuk menghitung atau memperbarui *minAtrVal[], maxAtrVal[], SumDiffAtrPrevCenter[], SumAtr[], StdDev[][][], centers[]*
 6. *J* = *J* + *dist*;
 7. Hitung centroids baru dengan membagi *SumAtr[]* dengan *countObj* lalu simpan di *centers*

8. Hitung standar deviasi aproksimasi untuk tiap atribut dengan menggunakan *SumDiffAtrPrevCenter* lalu simpan di *StdDev*
9. *value' = gabungan dari centers[], countObj, minAtrVal, maxAtrVal, StdCluster, J*
10. keluarkan *key-value'*

Contoh pemanfaatan algoritma k-Means yang melakukan komputasi ringkasan statistik di tiap klaster diberikan pada (Moertini & L. Venica, 2017). Pada eksperimen di situ, big data yang penulis gunakan adalah data cuaca yang diunduh dari website NOAA (*National Oceanic and Atmospheric Administration*) yang menyediakan big data hasil perekaman sensor cuaca dari berbagai negara.

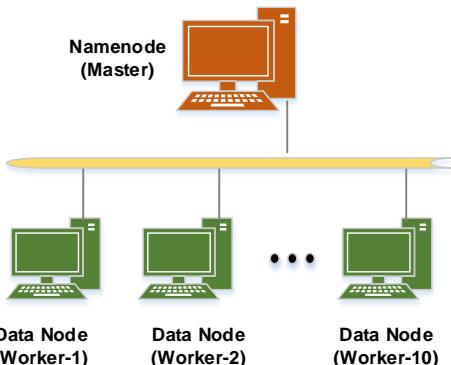
12.5.2. Algoritma k-Means untuk Lingkungan Sistem Spark

Karena k-Means paralel berbasis MapReduce pada Hadoop kurang efisien dalam mengelompokkan big data, penulis telah mengembangkan k-Means pada Spark (Gambar 12.5) untuk menghitung ringkasan statistik di tiap klaster. Pengembangan dilakukan dengan memodifikasi algoritma pada *mapPartition()*, *reduceByKey()* maupun pada program utama, dengan penjelasan di bawah ini:

- Pada *mapPartition()*: Pada pasangan *key-value* yang dikeluarkan, *value* disertai dengan nilai-nilai fitur dari objek. Dengan demikian, *value* yang dikeluarkan adalah: indeks klaster, jarak objek ke centroid-nya dan seluruh nilai fitur objek.
- Pada *reduceByKey()*: Selain menghitung jumlah total anggota dan jarak akumulatif pada sebuah klaster, satu task *reduceByKey()* juga menghitung nilai minimum, maksimum dan standar deviasi aproksimasi dari setiap fitur objek di sebuah klaster.
- Pada program utama (driver): Setelah mengumpulkan keluaran dari semua *reduceByKey()*, menghitung centroid baru dan *Delta_CostTotal*, jika iterasi tidak dilanjutkan lagi maka data yang disimpan (ke disk) adalah ringkasan statistik dari tiap klaster.

12.5.3. Perbandingan Kinerja k-Means pada Hadoop vs Spark

Eksperimen untuk mengelompokan big data studi kasus dan membandingkan kinerja, khususnya kecepatan eksekusi, algoritma k-Means untuk lingkungan Hadoop dan Spark (yang telah dikembangkan penulis) dilakukan pada jaringan dengan 11 komputer. Hadoop dijalankan dengan Yarn yang bertugas untuk memanajemen sumber daya pada komputer-komputer dan menjadwalkan tugas-tugas (tasks) berupa fungsi-fungsi *map()* dan *reduce()*. Satu komputer berperan sebagai master dan sisanya sebagai node data (Gambar 12.6) tempat *map()* dan *reduce()* dijalankan secara paralel dengan mengakses blok-blok HDFS yang tersimpan di disk pada node ini. Spark juga dikonfigurasi untuk berjalan di atas Yarn dan mengakses file-file HDFS pada Hadoop. Bagi Spark, node data pada Hadoop dapat menjadi worker tempat menjalankan tugas-tugas *mapPartition()* dan *reduceByKey()* secara paralel. Dalam membaca file HDFS, (secara *default*) 1 blok HDFS di worker dijadikan 1 RDD.



Gambar 12.6. Jaringan klaster Hadoop untuk eksperimen.

Eksperimen Perbandingan Kinerja

Secara teoritis, algoritma k-Means paralel pada Spark dipastikan lebih cepat daripada k-Means paralel pada Hadoop. Namun, bagaimana perbandingan kecepatan eksekusi keduanya? Untuk mengelompokkan big data tertentu, apakah k-Means Hadoop tetap dapat digunakan dengan cukup efisien? Untuk menjawab pertanyaan ini, penulis eksperimen untuk membandingkan kinerja keduanya.

Data studi kasus yang digunakan untuk eksperimen adalah hasil rekaman penggunaan energi listrik di sebuah rumah, yang diunduh dari <https://archive.ics.uci.edu/ml/datasets/>. Data tersebut terdiri dari 2.075.259 hasil pengukuran (rekord) pada Desember 2006 s/d November 2010 (47 bulan) dan berukuran 132 Mb. Contoh isi data yang berupa rekord-rekord hasil pengukuran adalah:

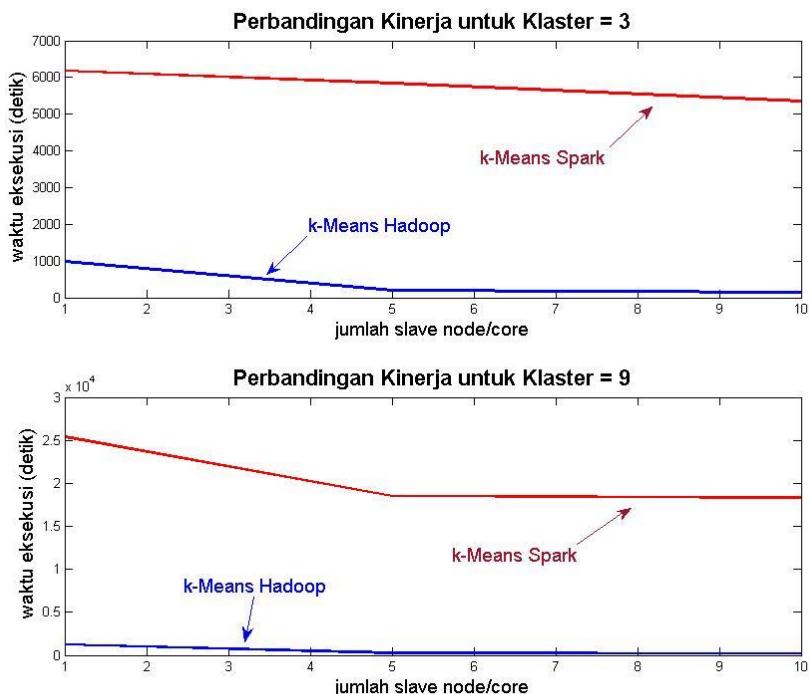
```
8/6/2007;18:39:00;4.072;0.242;236.750;17.200;37.000;1.000;19.000
8/6/2007;18:40:00;3.754;0.222;236.920;15.800;37.000;2.000;17.000
8/6/2007;18:41:00;3.612;0.076;237.640;15.200;38.000;2.000;17.000
8/6/2007;18:42:00;3.612;0.076;237.820;15.200;37.000;1.000;18.000
```

Sebagaimana dipaparkan pada (Moertini & L. Venica, 2017), penulis juga mengelompokkan data tersebut sebagai contoh kasus pemanfaatan k-Means paralel pada Hadoop.

Untuk pengujian kecepatan, himpunan data yang telah di-praolah (sehingga siap untuk diumpulkan ke k-Means) direplikasi beberapa kali sehingga mencapai ukuran 512 Mb dan 1 Gb. Pengelompokan data dilakukan dengan jumlah klaster 3 dan 9 pada jaringan klaster Hadoop dengan berturun-turut menggunakan 1, 5 dan 10 komputer data node atau core. Hasil eksperimen dipaparkan pada Tabel 12.2, Gambar 12.7, Tabel 12.3 dan Gambar 12.8.

Tabel 12.2 Waktu eksekusi k-Means paralel untuk memproses himpunan data dengan 5 fitur dan berukuran 512 Mb.

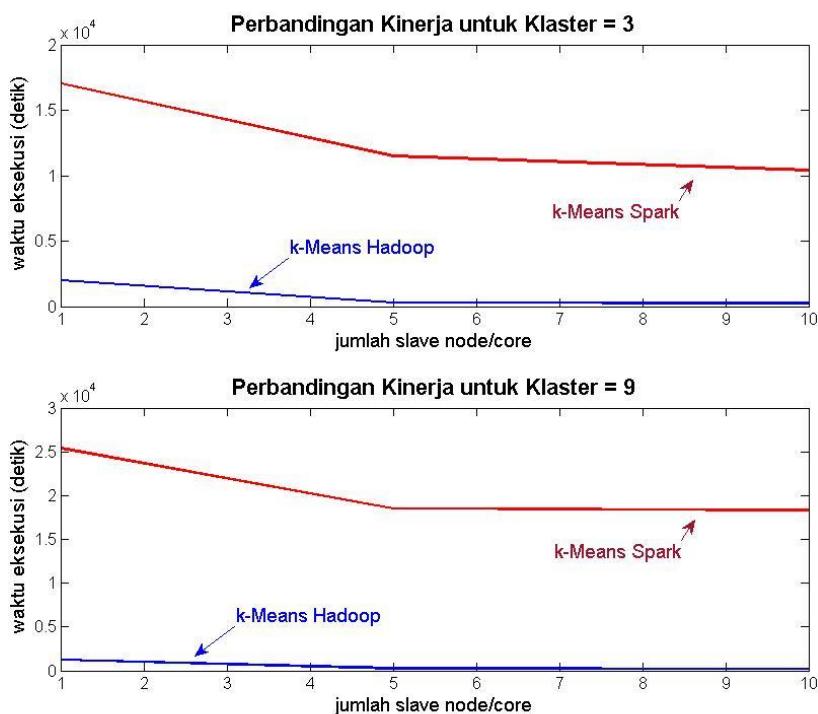
Jumlah Slave/Core	Jumlah klaster = 3		Jumlah klaster = 9	
	k-Means Hadoop (detik)	k-Means Spark (detik)	k-Means Hadoop (detik)	k-Means Spark (detik)
1	6,179	981	25,422	1,244
5	5,842	198	18,479	281
10	5,348	143	18,342	208



Gambar 12.7. Perbandingan waktu eksekusi k-Means Hadoop vs Spark dengan data 512 Mb.

Tabel 12.3. Waktu eksekusi k-Means paralel untuk memproses himpunan data dengan 10 fitur dan berukuran 1 Gb.

Jumlah Slave/Core	Jumlah klaster = 3		Jumlah klaster = 9	
	k-Means Hadoop (detik)	k-Means Spark (detik)	k-Means Hadoop (detik)	k-Means Spark (detik)
1	17,063	2,016	22,126	2,071
5	11,496	304	13,756	292
10	10,415	255	13,492	209



Gambar 12.8.. Perbandingan waktu eksekusi k-Means Hadoop vs Spark dengan data 1 Gb.

Pada dua tabel dan gambar di atas, baik untuk data berukuran 512 Mb maupun 1 Gb, dimana k-Means dijalankan pada jaringan Hadoop dan Spark dengan Yarn, kecepatan eksekusi k-Means paralel Spark berkisar antara 8 sampai 90 kali. Penambahan jumlah core (yang identik dengan *tasks* paralel yang dijalankan) pada Spark berdampak signifikan terhadap peningkatan kecepatan eksekusi. Pada Hadoop, penambahan jumlah worker node hanya sedikit mengurangi waktu eksekusi. "Biaya" proses pembacaan dan penulisan ke disk, juga proses shuffling dan sorting (sebelum pasangan data key-value diproses oleh fungsi `reduce()`) menjadi penyebab dari kelambatan eksekusi k-Means Hadoop.

Dari hasil perbandingan di atas, dapat disimpulkan bahwa pengelompokan big data lebih cocok dilakukan dengan menggunakan k-Means paralel pada Spark.

12.6. Penutup

Bab ini telah membahas cara kerja algoritma k-Means asli (yang dapat digunakan untuk mengelompokan non-big-data) dan pengembangannya menjadi algoritma paralel untuk memproses big data di lingkungan Hadoop dan Spark. Dari hasil eksperimen perbandingan kecepatan eksekusi, ternyata k-Means paralel untuk lingkungan Spark secara umum jauh lebih cepat dibandingkan k-Means pada Hadoop. Dengan demikian, k-Means paralel Spark lebih cocok untuk manajemen big data.

Jika ukuran himpunan data relatif kecil dan jumlah objek-objek yang dikelompokan mencapai ribuan, juga dibutuhkan untuk “melabeli” tiap objek dengan indeks/nomor klasternya, maka dapat dipilih k-Means asli (non-paralel) yang sudah diimplementasikan pada beberapa perangkat lunak (misalnya Matlab, Weka, RapidMiner, dll) dan *library* bahasa Java, Python, dll.

Tujuan pengelompokan big data pada umumnya adalah untuk mendapatkan model atau pola dari tiap klaster. Karena jumlah objek dapat mencapai jutaan bahkan milyaran, maka hasil akhir berupa pelabelan tiap objek menjadi kurang bermanfaat. (Namun, jika dibutuhkan, yang digunakan biasanya teknik klasifikasi yang dapat memberikan hasil pelabelan kelas yang lebih akurat. Dalam hal ini dibutuhkan data training, dimana tiap objek sudah dilabeli dengan kelasnya).

Hal-hal penting untuk diperhatikan ketika memanfaatkan algoritma k-Means:

1. Penyiapan data: Tahap ini merupakan tahap yang krusial dan penting untuk dilakukan dengan benar. Data “mentah” mungkin masih “kotor”, tidak konsisten, ada yang hilang, atau nilai-nilainya ada yang tidak cocok untuk ditangani k-Means. Selain itu, data dapat memiliki banyak atribut/kolom yang jika dikaitkan dengan tujuan pengelompokan, ada yang tidak relevan. Pembersihan, transformasi data dan pemilihan dan/atau pembuatan fitur-fitur perlu dilakukan sedemikian rupa untuk menghasilkan himpunan data berkualitas bagus yang siap untuk diumpulkan ke k-Means (dan diprediksi dapat menghasilkan luaran yang diharapkan).
2. Pemilihan jumlah kelompok: Pada k-Means, untuk dapat menghasilkan klaster-klaster yang bagus (objek-objek dalam satu klaster “berdekatan” dan “berjauhan” dengan objek-objek di klaster yang lain), jumlah kelompok yang tepat atau terbaik harus “dicari” (cara mencari ini sudah dibahas sebelumnya.)
3. Evaluasi dan interpretasi hasil pengelompokan: Hasil pengelompokan (label klaster pada tiap objek, centroids dan komponen-komponen pola klaster lainnya) perlu dievaluasi dan diinterpretasikan apakah sudah dapat menjawab tujuan pengelompokan data. Jika ternyata belum menjawab atau belum memberikan solusi terhadap tujuan, maka proses pengelompokan perlu diulang lagi mulai dari tahap penyiapan data.

Metoda penyiapan data, evaluasi dan interpretasi hasil pengelompokan dapat dicari dari literatur-literatur data mining dan Machine Learning beserta aplikasinya. Jika pengelompokan akan memanfaatkan

library Machine Learning pada Spark (MLLib atau ML), tahapan dapat mengacu ke referensi (Karau & Warren, 2017).

Ucapan Terima Kasih

Ucapan terima kasih ditujukan kepada Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan yang telah mendanai penelitian ini melalui skema Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT), tahun anggaran 2020, dengan nomor kontrak III/LPPM/2020-04/107-PE-S.

Referensi

- (Chius dan Tavella, 2011) S. Chius and D. Tavella; *Data Mining and Market Intelligent for Optimal Marketing Returns*, UK: Routledge Pub., 2011.
- (Han et al., 2012) J. Han, M. Kamber and J. Pei, *Data Mining Concepts and Techniques 3rd Ed.*, USA: The Morgan Kaufmann Publ., 2012.
- (Holmes, 2012) A. Holmes, *Hadoop in Practice*, USA: Manning Publications Co., 2012.
- (Karau et al., 2015) Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia, *Learning Spark*, O'Reilly Media, Inc., 2015
- (Karau & Warren, 2017) Holden Karau and Rachel Warren, *High Performance Spark*, O'Reilly Media, Inc., USA, 2017.
- (Moertini & L. Venica, 2017) V. S. Moertini and L. Venica, Parallel k-Means for Big Data: On Enhancing Its Cluster Metrics and Patterns, *Journal of Theoretical and Applied Information Technology*, Vol. 95, No. 8, 2017, Pp. 1844-1857.
- (Moertini et al., 2018) V. S. Moertini, G. W. Suarjana, L. Venica and G. Karya, Big Data Reduction Technique using Parallel Hierarchical Agglomerative Clustering, *IAENG International Journal of Computer Science*, Vol. 45. No. 1, 2018.
- (Moertini & Venica, 2016) V. S. Moertini, L. Venica, "Enhancing parallel k-means using map reduce for discovering knowledge from big data", in *Proc. of. 2016 IEEE Intl. Conf. on Cloud Computing and Big Data Analysis (ICCCBDA 2016)*, Chengdu China, 4-7 July 2016, pp. 81-87.
- (Sammer, 2012) E. Sammer, *Hadoop Operations*, USA: O'Reilly Media, Inc., 2012.
- (Tsiptsis dan Chorianopoulos, 2009) K. Tsiptsis and A. Chorianopoulos, *Data Mining Techniques in CRM: Inside Customer Segmentation*, UK: John Wiley and Sons, L., 2009.
- (Zhao, Ma dan Q. He, 2009) W. Zhao, H. Ma and Q. He, "Parallel k-means clustering based on mapreduce", *CloudCom 2009*, LNCS 5931, pp. 674–679, Berlin Heidelberg: Springer-Verlag, 2009.
- (URL-cluster-1) Data Mining - Cluster Analysis,
https://www.tutorialspoint.com/data_mining/dm_cluster_analysis.htm (diakses 17 Februari 2020)
- (URL-cluster-2) What is Clustering in Data Mining,
<https://bigdata-madesimple.com/what-is-clustering-in-data-mining/> (diakses 17 Februari 2020)

Halaman ini sengaja dikosongkan

Bab 13 Estimasi Dimensi Tubuh Manusia dengan Kinect

Oleh:

Mariskha Tri Adithia, Hereza Ardhyta dan Kristopher D. Harjono

13.1. Pendahuluan

Pada berbagai bidang, pengukuran dimensi tubuh manusia banyak digunakan. Misalnya, di bidang *fashion*, pengukuran tubuh manusia digunakan untuk keperluan penjahitan pakaian, agar pakaian yang dijahit sesuai dan pas untuk pemesannya. Pengukuran dimensi tubuh manusia juga banyak digunakan di bidang olah raga, untuk meningkatkan potensi optimal seorang atlit pada cabang olah raga tertentu.

Pengukuran dimensi tubuh, misalnya meliputi pengukuran panjang kaki, panjang lengan, dan panjang badan atau torso. Pengukuran ini biasanya dilakukan dengan menggunakan bantuan orang lain yang mengerti tentang pengukuran dimensi tubuh, dan menggunakan meteran. Metode pengukuran seperti ini mengharuskan orang yang akan diukur dan orang yang membantu mengukur berada di tempat yang sama.

Ada kalanya konsumen membutuhkan pengukuran tubuh, namun tidak ada orang lain yang dapat membantunya. Misalnya, saat ingin membeli pakaian. Jika pakaian dibeli di sebuah toko fisik, maka untuk memastikan pakaian tersebut sesuai dengan ukuran tubuh, konsumen hanya perlu mencobanya. Tetapi, saat membeli pakaian secara online pada suatu platform e-commerce, mencoba pakaian yang ingin dibeli tidak mungkin dilakukan. Sehingga, ukuran tubuh konsumen menjadi penting menentukan sesuai tidak pakaian dengan ukuran tubuh. Pada permasalahan seperti inilah, suatu teknologi dibutuhkan, untuk dapat mengestimasi dimensi tubuh konsumen atau manusia pada umumnya.

Salah satu alat bantu yang dapat digunakan untuk mengestimasi dimensi tubuh manusia adalah Microsoft Kinect. Microsoft Kinect, dikembangkan oleh Microsoft, untuk perangkat permainan konsol Xbox mereka. Microsoft Kinect digunakan untuk menangkap dan mengenali gerakan dan gestur tubuh pemain saat sedang bermain menggunakan Xbox. Dengan adanya Microsoft Kinect, pemain tidak perlu lagi menggunakan *gamepad stick* untuk dapat menangkap dan mengenali gerakan dan gestur pemain, Microsoft Kinect, lihat *Gambar 13.1.*, dilengkapi dengan kamera RGB dan sensor infrared untuk menjalankan fungsinya.

Pada artikel ini, Microsoft Kinect akan dimanfaatkan untuk mengestimasi dimensi tubuh manusia. Untuk menghasilkan estimasi ini, manusia yang dimensi tubuhkan akan diukur, berdiri di depan Microsoft Kinect. Selanjutnya, Microsoft Kinect akan menangkap tubuh manusia ini sebagai gambar 3 dimensi dan mengubahnya menjadi data. Data ini tidak serta merta mengeluarkan ukuran tubuh manusia; dibutuhkan

pengolahan data terlebih dahulu. Teknik pengolahan data yang digunakan pada artikel ini adalah Principal Component Analysis (PCA) dan regresi linier.



Gambar 13.1. Microsoft Kinect dan bagian-bagiannya.

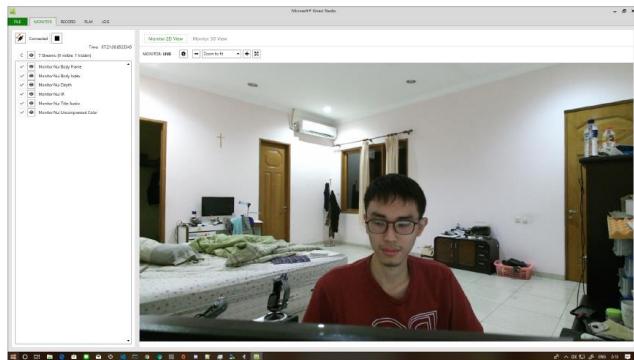
13.2. Microsoft Kinect

Microsoft Kinect merupakan sebuah perangkat pendukung untuk konsol permainan Xbox Kinect yang dikembangkan oleh Microsoft. Xbox Kinect pertama kali diperkenalkan kepada dunia pada saat acara *Electronic Entertainment Expo* (E3) tahun 2009. Pada saat perkenalannya, perangkat ini disebut *Project Natal* [Crecente, 2009]. Tujuan dari Microsoft meluncurkan perangkat ini adalah mereka ingin menciptakan ulang cara manusia berinteraksi dengan komputer. Pada tahun 2010, perangkat ini diluncurkan untuk konsol permainan Xbox 360 dan beranama Microsoft Kinect [Nosowitz, 2010]. Seiring perkembangan Xbox, pada tahun 2013 diluncurkanlah Microsoft Xbox One Kinect dengan berbagai penyempurnaan dari versi sebelumnya.

Microsoft Kinect bekerja dengan menangkap gerakan tubuh pengguna dan menggunakannya sebagai masukan untuk mengendalikan komputer. Perangkat ini menangkap tubuh penggunanya dengan cara memetakan ruang 3 dimensi di depannya dan memutuskan apakah terdapat manusia atau tidak. Pemetaan 3 dimensi ini didapatkan dari sensor-sensor yang terdapat pada Microsoft Kinect (lihat Gambar 13.1) tersebut, yaitu [Jiao, 2017]:

- Sensor Kamera RGB

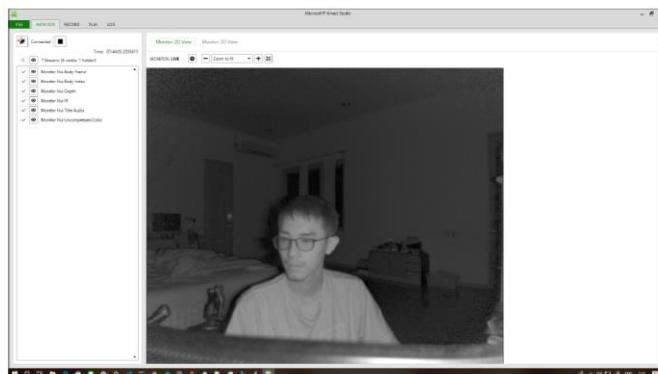
Sensor ini berfungsi untuk menangkap gambar RGB dengan menggunakan kamera. Kamera yang ada pada Microsoft Kinect ini memiliki resolusi sensor sebesar 1920x1080 piksel dengan kemampuan menangkap sebesar 30 *frame per second* dan pandangan horizontal 70 derajat dan vertikal 60 derajat. Kamera ini juga dapat digunakan sebagai *webcam* pada sistem operasi Windows. Contoh hasil gambar dari sensor kamera RGB ini diberikan pada Gambar 13.2.



Gambar 13.2. Contoh hasil tangkapan sensor kamera RGB pada Microsoft Kinect.

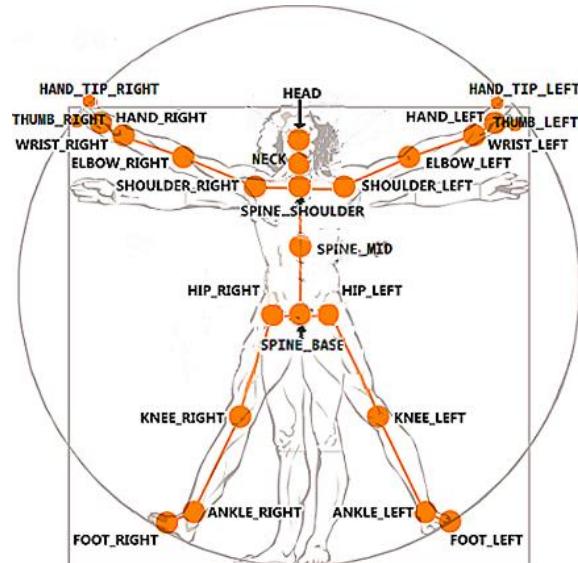
- Sensor Inframerah

Sensor ini berfungsi untuk menangkap gelombang inframerah yang dipantulkan oleh benda di depannya, juga dengan menggunakan kamera. Kamera inframerah ini memiliki resolusi 512x424 piksel dengan kemampuan menangkap sebesar *30 frame per second*. Inframerah yang ditangkap oleh kamera dihasilkan oleh Kinect itu sendiri menggunakan *IR Blaster*. Hasil dari pantulan *IR Blaster* inilah yang akan digunakan untuk memetakan ruang 3 dimensi di depannya. Pemetaan 3 dimensi dari Kinect memiliki batas pengukuran sebesar 0.5 - 4.5 meter. Contoh hasil tangkapan kamera inframerah ini diberikan pada Gambar 13.3.



Gambar 13.3. Contoh hasil tangkapan sensor kamera inframerah Microsoft Kinect.

Microsoft Kinect juga dilengkapi dengan software development kit (SDK) yang disebut KinectSDK. KinectSDK memungkinkan pembangunan software berbasis hasil tangkapan gambar dari Microsoft Kinect tersebut. Dengan menggunakan KinectSDK ini, salah satunya, gambar tubuh manusia dapat disegmentasikan berdasarkan area dan berdasarkan lokasi sendi [Samejima, 2012]. Sendi yang dapat dikenali oleh KinectSDK misalnya sendi pada siku dan lutut. Seluruh sendi yang dikenali oleh KinectSDK diberikan pada Gambar 13.4.



Gambar 13.4. Seluruh sendi yang dikenali KinectSDK.⁵⁶

13.3. Principal Component Analysis

Principal Componen Analysis (PCA) adalah suatu teknik untuk mengurangi dimensi suatu set data dengan banyak variabel [Brems, 2017]. Variabel yang banyak ini belum tentu sesuai dengan kebutuhan analisis selanjutnya. Dengan terlalu banyaknya variabel, model yang dibuat akan overfitting terhadap data, sehingga tidak akurat lagi. Selain itu, model juga menjadi tidak sesuai, karena memuat berbagai variabel yang tidak relevan dengan masalah, misalnya.

Jika diberikan suatu set data dengan variabel bebas $X = \{x_1, x_2, \dots, x_n\}$ dan variabel terikat $Y = \{y_1, y_2, \dots, y_m\}$, langkah-langkah PCA adalah sebagai berikut:

1. Tuliskan set data ke dalam sebuah matriks M , dengan baris mewakili variabel terikat Y , dan kolom mewakili variabel bebas X .
2. Hitung rata-rata masing-masing kolom pada matriks M .
3. Normalisasi tiap entri pada matriks M dengan menggunakan Rumus 1 berikut.

$$m_{ij}^* = \frac{m_{ij} - \mu_j}{\sigma_j} \quad (1)$$

dengan

m_{ij} : entri matriks M pada baris i dan kolom j

m_{ij}^* : entri yang sudah dinormalisasi

μ_j : nilai rata-rata entri pada kolom j

⁵⁶ Sumber gambar: <https://medium.com/@lisajamhoury/understanding-kinect-v2-joints-and-coordinate-system-4f4b90b9df16>

σ_j : standar deviasi entri pada kolom j

4. Bangun matriks kovarian, $K = M^T M$.
5. Hitung nilai eigen dan vektor eigen, yang bersesuaian, dari matriks K . Hasil perhitungan vektor eigen dimuat pada matriks P .
6. Urutkan nilai eigen dan sesuaikan posisi vektor eigen pada matriks P . Namai matriks yang sudah terurut ini sebagai P^* .
7. Bangun matriks data akhir M^* , dengan memilih terlebih dahulu berapa PC yang akan digunakan. Lalu hitung $M^* = M'P^*$, di mana
 - M^* : matriks yang berisikan gabungan kolom PC pada matriks M yang dipilih
 - P^* : matriks yang berisikan gabungan kolom pada matriks P yang dipilih

Tiap kolom pada matriks ini mewakili sebuah principal component (PC).

8. Pilih berapa fitur yang akan digunakan pada analisis selanjutnya dari M^* , misalnya dengan menghitung proporsi varians suatu PC terhadap seluruh data pada M^* . Ini dapat dihitung dengan cara berikut. Misalkan PC_k adalah PC pada kolom k dan λ_k adalah nilai eigen PC_k , maka proporsi varians PC_k diberikan pada Rumus 2.

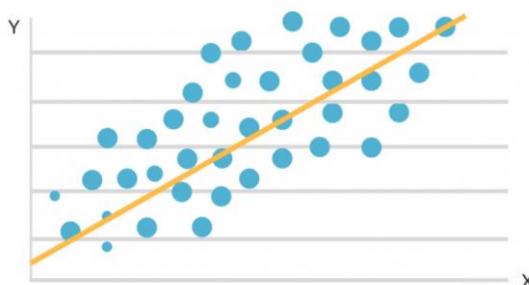
$$\text{Proporsi } PC_k = \frac{\lambda_k}{\sum_{i=1}^n \lambda_i} \quad (2)$$

Proporsi ini juga disebut sebagai kontribusi suatu PC.

13.4. Regresi Linier

Regresi linier adalah suatu pendekatan statistika untuk memodelkan hubungan antara dua variabel, variabel terikat dan bebas, dengan mencocokkan data hasil observasi pada sebuah persamaan linier [Yale, 1997]. Jika persamaan linier sudah didapatkan, persamaan ini nantinya dapat digunakan untuk melakukan prediksi. Regresi linier dapat dilakukan dengan dua cara, yaitu regresi linier univariat dan multivariat.

Ilustrasi terkait regresi linier diberikan pada Gambar 13.5. Pada gambar tersebut, misalkan lingkaran berwarna biru adalah semua data hasil observasi. Garis berwarna kuning, adalah garis, yang merepresentasikan persamaan linier, yang menggambarkan hubungan antara data.



Gambar 13.5. Ilustrasi regresi linier.

Misalkan, y_i adalah variabel terikat, dan x_i variabel bebas, yang nilainya diketahui, maka, hubungan nilai x_i dan y_i dapat dimodelkan dengan menggunakan regresi linier univariat diberikan pada Persamaan 3.

$$y_i = b_0 + b_1 x_i \quad (3)$$

Di mana nilai a dan b akan ditentukan, misalnya dengan menggunakan metode *least square*, pada Persamaan 4 dan Persamaan 5, dengan n adalah banyaknya data.

$$b_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \quad (4)$$

$$b_0 = \bar{y} - b_1 \bar{x} \quad (5)$$

Jika variabel bebas yang terlibat tidak hanya satu, maka harus digunakan regresi linier multivariat, yang mengikuti Persamaan 6 berikut.

$$y_i = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n \quad (6)$$

Nilai $b_0, b_1, b_2, \dots, b_n$ dihitung dengan menyelesaikan persamaan dalam bentuk matriks.

13.5. Metode Estimasi Dimensi Tubuh dan Hasilnya

Pada bagian ini, pengukuran estimasi dimensi tubuh manusia dijelaskan. Pengukuran ini dilakukan dengan menggunakan Microsoft Kinect, yang menangkap gambar manusia dan menggunakan KinectSDK untuk mengeluarkan lokasi sendi (lihat *Gambar 13.4*). Namun, hasil lokasi sendi ini belum berbentuk ukuran dimensi tubuh. Selain itu, ukuran tubuh manusia, seperti lebar pinggul, lingkar perut, dan berat badan, juga tidak dapat diukur dari gambar tangkapan Microsoft Kinect. Oleh karena itu, Pengukuran manual, PCA, dan regresi linier akan dimanfaatkan untuk mengestimasi ukuran-ukuran tersebut. Penjelasan lebih rinci diberikan di sebagai berikut.

Dalam pengukuran dimensi tubuh manusia ini, langkah pertama adalah pengumpulan data, dari 50 sukarelawan berjenis kelamin laki-laki, dengan rentang usia 17-62 tahun. Para sukarelawan ini diminta untuk berdiri di depan Microsoft Kinect, agar gambar seluruh tubuh dapat diambil. Dari gambar ini, didapatkan posisi sendi, dalam koordinat Kartesius 3 dimensi, masing-masing sukarelawan. Dari lokasi sendi, panjang bagian tubuh dapat dihitung, dengan menggunakan rumus jarak Euclidean pada Rumus 7.

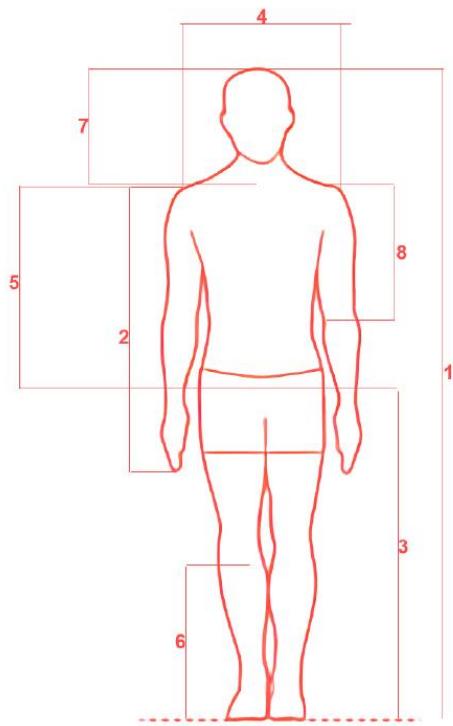
$$D_k = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (7)$$

dengan

D_k : dimensi atau panjang antara joint i dan j .

(x_i, y_i, z_i) : lokasi joint i .

Dimensi tubuh yang dapat diukur dari data ini diberikan pada Gambar 13.6 dan Tabel 13.1.



Gambar 13.6. Dimensi bagian tubuh yang didapatkan langsung dari Microsoft Kinect.

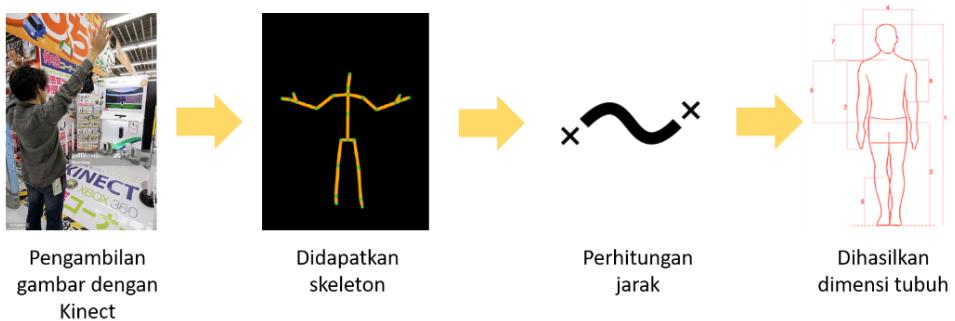
Tabel 13.1. Ukuran yang didapat dari Microsoft Kinect.

Nomor	Ukuran
1	Tinggi badan
2	Panjang lengan
3	Panjang kaki
4	Lebar pundak
5	Panjang torso
6	Tinggi lutut
7	Panjang kepala
8	Panjang lengan atas

Perhitungan D_k ini dilakukan untuk semua bagian tubuh dari semua sukarelawan. Saat semua data sudah didapatkan dan perhitungan D_k sudah dilakukan, maka dimensi tubuh manusia, sesuai Tabel 13.1., sudah didapat. Hasil perhitungan ini dapat direpresentasikan dalam bentuk matriks, misalnya dinotasikan dengan M (lihat matriks 8), seperti diberikan di bawah ini, Baris pada M mewakili sukarelawan, dan kolomnya mewakili masing-masing ukuran tubuh pada Tabel 13.1. .

$$M = \begin{pmatrix} 170 & 75.5 & 86 & 41 & 54.5 & 45.5 & 32 & 29 \\ 171.5 & 78.5 & 100.5 & 41.5 & 58.5 & 50 & 35 & 30.5 \\ 170 & 76 & 94 & 44 & 57.5 & 58 & 35 & 31 \\ \vdots & \vdots \\ 160 & 74.5 & 90.2 & 39 & 49 & 49.8 & 29 & 29 \end{pmatrix} \quad (8)$$

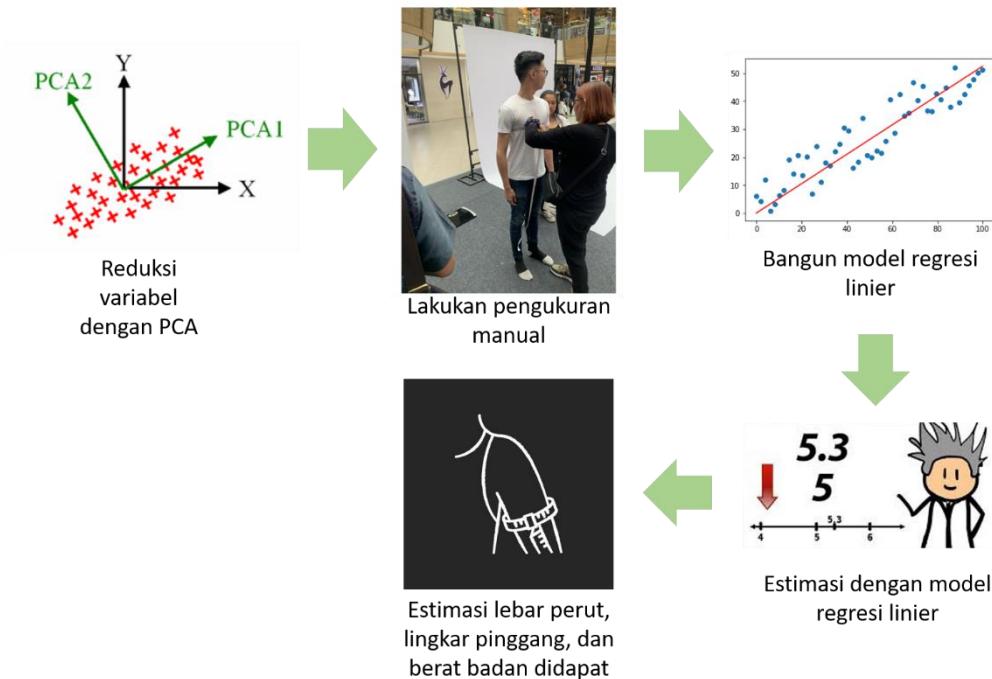
Langkah untuk mendapatkan dimensi tubuh manusia yang dideskripsikan sebelumnya, diberikan pada Gambar 13.7.



Gambar 13.7. Langkah-langkah mendapatkan dimensi tubuh manusia dengan Microsoft Kinect.

Langkah selanjutnya adalah mengestimasi ukuran tubuh yang tidak dapat diukur dengan menggunakan hasil tangkapan gambar Microsoft Kinect, yaitu ukuran lebar pinggul, lingkar perut, dan berat badan. Untuk mengestimasi ukuran ini, dilakukan langkah-langkah berikut (lihat Gambar 13.8):

1. Reduksi variabel pada matriks M dengan menggunakan PCA.
2. Lakukan pengukuran lingkar pinggang, lebar perut, dan berat badan, secara manual dari 50 sukarelawan yang sama.
3. Gunakan regresi linier, dengan menggunakan hasil pengukuran manual dan hasil PCA, untuk mengestimasi ukuran lebar pinggul, lingkar perut, dan berat badan.



Gambar 13.8. Langkah-langkah mendapatkan estimasi dimensi lebar perut, lingkar pinggang, dan berat badan.

Setelah PCA dilakukan, didapatkan proporsi varians PC atau kontribusi masing-masing PC diberikan pada Tabel 13.2.

Tabel 13.2. Hasil PCA

Principal Component (PC)	Variabel/Ukuran	Kontribusi PCA
PC1	Tinggi badan	54.794%
PC2	Panjang lengan	12.482%
PC3	Panjang kaki	10.946%
PC4	Lebar pundak	6.860%
PC5	Panjang torso	5.957%
PC6	Tinggi lutut	4.542%
PC7	Panjang kepala	2.845%
PC8	Panjang lengan atas	1.574%

Langkah 3 di atas dapat dilakukan dengan menggunakan regresi linier univariat. Pada regresi linier univariat ini, hanya digunakan satu PC saja. Misalkan, digunakan PC dengan kontribusi tertinggi, yaitu PC₁. Dari langkah PCA yang sudah dilakukan, didapatkan data akhir dari PCA ini, untuk PC₁, yang diberikan pada matriks M^* (matriks 9).

$$M^* = \begin{pmatrix} -0.714946 \\ 2.129313 \\ 2.008069 \\ \vdots \\ -1.808282 \end{pmatrix} \quad (9)$$

Sebagai ilustrasi, misalkan digunakan PC₁ untuk mengestimasi lebar pinggul. Regresi linier dengan satu PC dilakukan dengan menyelesaikan Persamaan 10 berikut. Persamaan ini dibangun berdasarkan persamaan regresi linier univariat, yang bentuk umumnya diberikan pada Persamaan 3.

$$Y = b_0 + b_1 M^* \quad (10)$$

di mana:

- Y : hasil pengukuran manual lebar pinggul yang sudah dinormalisasi dengan menggunakan Rumus 1.
- M^* : matriks 9 yang diberikan di atas
- b_j : koefisien yang akan dicari nilainya.

Dengan menggunakan hasil yang sudah didapat, Persamaan 10 dapat dituliskan dalam bentuk perkalian matriks, yang diberikan pada Persamaan 11.

$$\begin{pmatrix} 1 & -0.714946 \\ 1 & 2.129313 \\ 1 & 2.008069 \\ \vdots & \vdots \\ 1 & -1.808282 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} -0.775638079 \\ -0.227483959 \\ 0.0465931 \\ \vdots \\ -0.227483959 \end{pmatrix} \quad (11)$$

Solusi Persamaan 11 ini diberikan pada Persamaan 12.

$$\begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} -1.3455 \cdot 10^{-7} \\ -0.1818 \end{pmatrix} \quad (12)$$

Maka, untuk mencari lebar pinggul sukarelawan ke- k digunakan langkah-langkah berikut. Pertama, hitung Y^* dengan Rumus 13 di bawah ini.

$$Y^* = (-1.3455 \cdot 10^{-7}) + (-0.1818 \cdot PC_{1,k}) \quad (13)$$

Pada persamaan 13, $PC_{1,k}$ adalah entri ke- k pada kolom M^* . Hasil Y^* dari Persamaan 10 di atas adalah suatu nilai yang sudah dinormalisasi. Sehingga untuk mengubahnya menjadi ukuran yang sebenarnya, digunakan Rumus 14 berikut. Rumus 14 ini diturunkan dari Rumus 1, yang sudah dijelaskan sebelumnya.

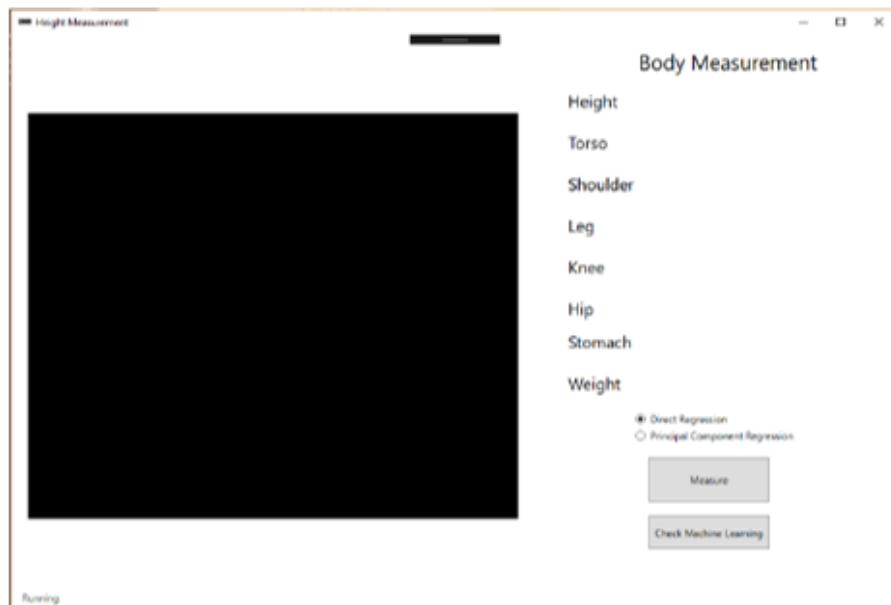
$$Y = (Y^* \cdot \sigma) + \mu \quad (14)$$

Sebagai contoh, misalkan akan dihitung lebar pinggul sukarelawan ke-1, maka gunakan baris pertama pada matriks M^* (matriks 9).

Dengan cara yang sama, dimensi lingkar perut dan berat badan juga dapat diestimasi, sehingga semua dimensi tubuh manusia menjadi lengkap.

13.6. Pembangunan Perangkat Lunak

Untuk melakukan estimasi yang telah dijelaskan, sebuah perangkat lunak telah dibangun dengan menggunakan bahasa C#. Tampilan antarmuka perangkat lunak ini diberikan pada Gambar 13.12.



Gambar 13.12. Tampilan antarmuka perangkat lunak.

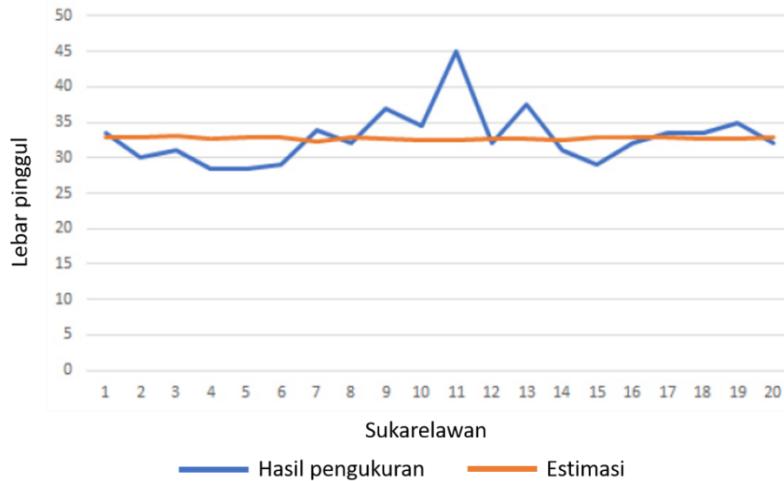
Pada antarmuka ini, hasil tangkapan Microsoft Kinect ditampilkan dalam bentuk skeleton. Dua metode untuk mengestimasi dimensi tubuh dapat dipilih yaitu dengan regresi linier saja, atau dengan menggunakan regresi linier dan PCA. Tombol Measure digunakan untuk menjalankan program. Tombol "Check Machine Learning" digunakan untuk mengukur akurasi hasil perhitungan. Hasil estimasi pengukuran selanjutnya diberikan pada Gambar 13.13.



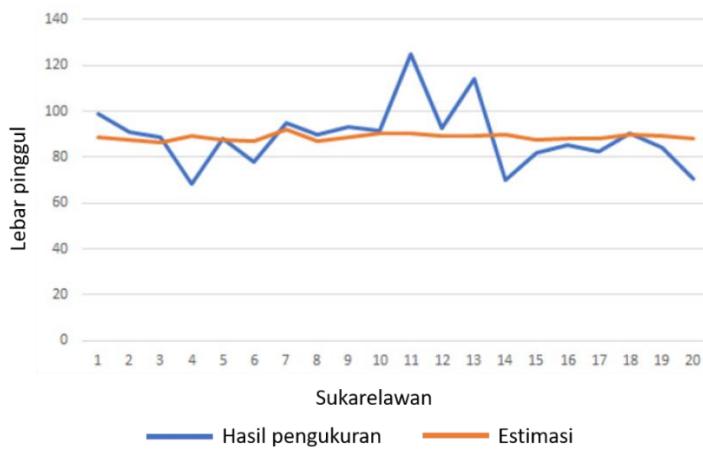
Gambar 13.13. Tampilan antarmuka di mana hasil estimasi sudah diberikan.

13.7. Hasil Eksperimen

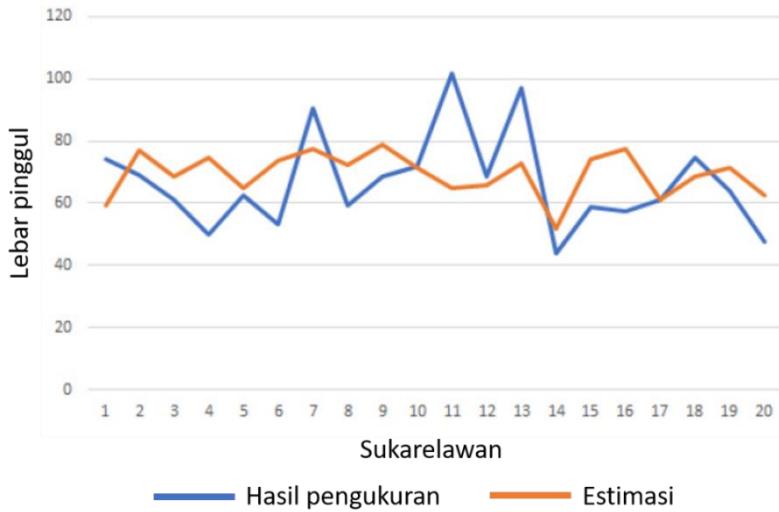
Untuk mengukur akurasi hasil estimasi dimensi lebar pinggul, lingkar perut, dan berat badan dengan PC yang berbeda-beda, jika dibandingkan dengan hasil pengukuran manual. Perbandingan estimasi dimensi lebar pinggul, lingkar perut, dan berat badan dengan PC1 dengan hasil pengukuran manual diberikan pada *Gambar 13.14*, *Gambar 13.15*, dan *Gambar 13.16*.



Gambar 13.14. Perbandingan lebar pinggul asli dengan hasil estimasi, dengan menggunakan PC1.



Gambar 13.15. Perbandingan lingkar perut asli dengan hasil estimasi, dengan menggunakan PC1.



Gambar 13.16. Perbandingan berat badan asli dengan hasil estimasi, dengan menggunakan PC1.

Hasil pengukuran manual dan estimasi dengan menggunakan PC1, untuk enam orang sukarelawan, diberikan pada Tabel 13. 4. Dari tabel ini dan plot yang sebelumnya diberikan, perbedaan hasil pengukuran dan estimasi tidak berbeda jauh.

Tabel 13. 4. Hasil estimasi dengan PC1 dan ukuran aslinya

Sukarelawan	Asli			Estimasi		
	Lebar pinggul	Lingkar perut	Berat badan	Lebar pinggul	Lingkar perut	Berat badan
1	33.5	99	74.2	32.8	88.52	59.35
2	30	91	69.3	32.89	87.71	77.18
3	31	88.5	61.2	33.1	86.36	68.71
4	28.5	68.5	49.8	32.71	89.07	74.48
5	28.5	88	62.7	32.9	87.75	64.67
6	29	78	53	32.97	87.25	73.6

Dari hasil estimasi 20 sukarelawan ini, selanjutnya dihitung rata-rata error yang didapatkan dari masing-masing PC. Hasil perhitungan error ini diberikan pada Tabel 13.5. Dari rata-rata keseluruhan error, PC4 memberikan rata-rata error paling rendah. Artinya, dengan PC4, hasil estimasi yang diberikan adalah yang paling baik.

Tabel 13.5. Rata-rata error yang dihasilkan antara hasil estimasi dan data asli

PC	Rata-rata error (cm)			Rata-rata keseluruhan error
	Lebar pinggul	Lingkar perut	Berat badan	
PC1	2.80	8.87	12.55	8.07
PC2	2.49	8.71	9.29	6.83
PC3	2.57	8.73	9.26	6.85
PC4	2.37	6.87	7.56	5.60
PC5	2.37	6.88	8.23	5.83
PC6	2.54	6.50	8.60	5.88
PC7	2.46	6.69	8.64	5.93
PC8	2.66	6.63	8.13	5.80

13.8. Kesimpulan

Bab ini menjelaskan cara mengestimasi dimensi tubuh manusia dengan menggunakan Microsoft Kinect dengan bantuan KinectSDK. Dimensi tubuh yang tidak dapat diukur langsung dengan Microsoft Kinect, yaitu lebar pinggul, lingkar perut, dan berat badan, diestimasi dengan menggunakan regresi linier. dikombinasikan dengan PCA untuk mereduksi jumlah variabel.

Sebagai eksperimen, dikumpulkan 50 sukarelawan yang diukur dimensi tubuhnya secara manual, maupun dengan menggunakan Microsoft Kinect. Selanjutnya, lebar pinggul, lingkar perut, dan berat badan, diestimasi dengan menggunakan data pengukuran manual ini. Berdasarkan hasil perhitungan, rata-rata error antara hasil estimasi dan data asli untuk masing-masing PC diberikan. Berdasarkan rata-rata error ini, PC4 memberikan hasil estimasi yang paling baik.

Berdasarkan hasil eksperimen ini, dapat disimpulkan bahwa Microsoft Kinect dapat dimanfaatkan untuk mengukur dimensi tubuh manusia, dan mengestimasi ukuran dimensi tubuh yang tidak bisa didapatkan langsung, dengan bantuan regresi linier dan PCA.

Referensi

- [Brems, 2017] <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c> (diakses pada 24 Agustus 2020)
- [Crecente, 2009] <https://kotaku.com/everything-you-need-to-know-about-project-natal-5280268> (diakses pada 24 Agustus 2020)
- [Jiao, 2017] J. Jiao, L. Yuan, W. Tang, Z. Deng, and Q. Wu, "A Post-Rectification Approach of Depth Images of Kinect v2 for 3D Reconstruction of Indoor Scenes," International Journal of Geo-Information, Vol. 6(11):349, 2017.

- [Nosowitz, 2010] <https://www.fastcompany.com/1659724/microsofts-motion-controlling-project-natal-now-named-microsoft-kinect> (diakses 24 Agustus 2020)
- [Samejima, 2012] I. Samejima, K. Makil, S. Kagamil, M. Kouchil, and H. Mizoguchi, "A Body Dimensions Estimation Method of Subject from a Few Measurement Items Using KINECT," IEEE International Conference on Systems, Man, and Cybernetics, South Korea 14-17 Oktober, 2012.
- [Yale, 1997] <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm> (diakses pada 12 Agustus 2020)

Bab 14 Segmentasi Citra Menggunakan Algoritma *Particle Swarm Optimization*

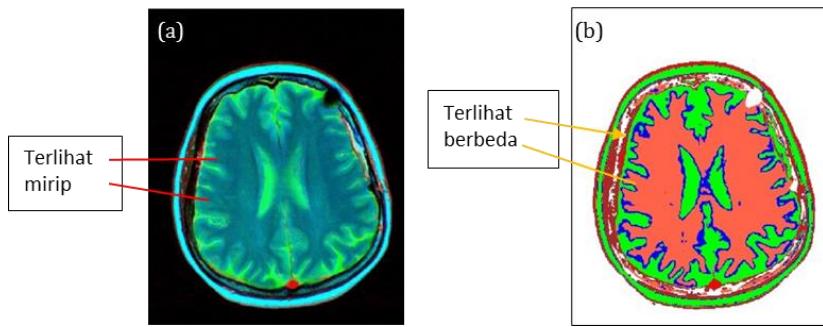
Oleh:

Alvinus Sutendy dan Natalia

14.1. Pendahuluan

Citra atau gambar merupakan salah satu media yang dapat digunakan untuk menyampaikan informasi. Informasi didapat melalui proses identifikasi terhadap objek-objek (bagian-bagian) yang berada pada gambar tersebut. Contoh informasi yang didapat misalnya ukuran seberapa besar bagian otak manusia yang abnormal. Proses identifikasi dilakukan dengan cara mengamati bagian dari gambar yang terlihat berbeda dengan bagian lainnya. Namun seringkali proses identifikasi tidak menghasilkan kesimpulan yang tepat. Hal ini salah satunya disebabkan oleh kualitas gambar yang kurang baik, misalnya objek-objek yang ada pada gambar terlihat sama padahal merupakan objek yang berbeda.

Salah satu contoh gambar yang kurang jelas adalah hasil pemeriksaan MRI (*Magnetic Resonance Imaging*) pada otak manusia (Gambar 14.1a). Dengan melakukan segmentasi terhadap gambar tersebut, dapat dihasilkan gambar yang objek-objeknya lebih mudah untuk diidentifikasi (Gambar 14.1b). Salah satu teknik yang dapat digunakan untuk melakukan segmentasi gambar adalah *clustering*. Dengan teknik *clustering*, piksel-piksel pada gambar dikelompokkan berdasarkan warnanya.



Gambar 14.1. Hasil MRI pada otak manusia: (a) sebelum segmentasi (b) sesudah segmentasi.⁵⁷

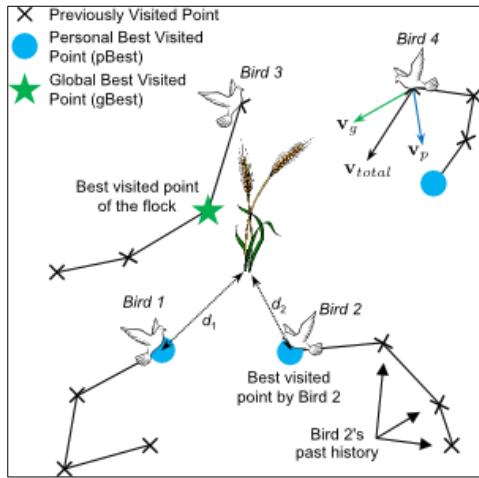
Clustering adalah teknik pengelompokan objek-objek (dalam hal ini berupa piksel) pada sebuah himpunan data (*dataset*), sehingga objek-objek yang berada dalam satu kelompok memiliki karakteristik yang sama, sedangkan objek-objek yang berada dalam kelompok yang berbeda memiliki karakteristik yang berbeda.

Pada proses *clustering*, misalnya dengan algoritma k-Means, perlu dihitung pusat *cluster* (*centroid*) untuk setiap *cluster*. Namun seringkali pencarian *centroid-centroid* tidak optimal. Untuk mengatasi hal ini telah dikembangkan algoritma yang dapat “membantu” mencari *centroid-centroid*-nya, yaitu algoritma PSO (*Particle Swarm Optimization*).

Algoritma PSO terinspirasi dari kawanan burung yang sedang terbang di langit untuk mencari makanan (Wong, 2011). Seekor burung mendekati sumber makanan dengan menggunakan kecerdasannya sendiri dan jika ada burung lain yang menemukan jalan yang lebih baik ke sumber makanan maka burung lainnya akan mengikuti. Begitu pula dengan algoritma PSO, pada algoritma ini, burung diasosiasikan sebagai partikel. Setiap partikel melakukan pencarian solusi yang optimal dengan cara melintasi *search space* (ruang pencarian). Setiap partikel melakukan penyesuaian terhadap posisi terbaik partikel itu sendiri dan posisi terbaik dari seluruh partikel dalam *swarm* (kawanan) selama melintasi ruang pencarian (Gambar 14.2). Algoritma lain selain PSO yang dapat digunakan untuk *clustering* adalah *K-means*. Algoritma ini lebih umum digunakan dibanding PSO, namun algoritma ini memiliki kekurangan yaitu solusinya dapat terjebak dalam nilai lokal optima (pencarian solusi kurang menyeluruh) sehingga hasil *clustering* tidak optimal (Wahyuni, 2016).

⁵⁷ Sumber komponen gambar:

https://www.researchgate.net/publication/313226266_Classification_of_MR_medical_images_Based_Rough-Fuzzy_K_-Means/figures?lo=1



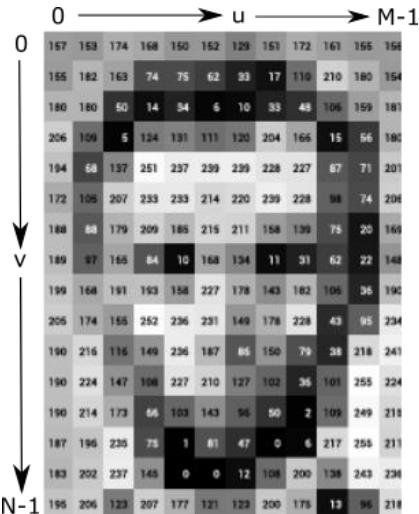
Gambar 14. 17. Ilustrasi burung sebagai partikel pada algoritma PSO.⁵⁸

14.2. Studi Literatur

14.2.1. Gambar Digital

Dalam pemrosesan gambar digital, gambar direpresentasikan sebagai fungsi dua dimensi $f(u, v)$ (Gonzales, 2007), dimana u dan v adalah koordinat piksel pada gambar. f merupakan nilai intensitas atau nilai warna pada koordinat (u, v) . Piksel merupakan elemen pada gambar. Gambar digital yaitu gambar yang nilai u , v , dan f -nya terbatas (Gambar 14.3).

⁵⁸ Sumber komponen gambar: <http://joshkovitz.com/research/projects/optimization-in-electromagnetics/>



Gambar 14.18. Ilustrasi gambar digital.⁵⁹

14.2.2. Ruang Warna

Untuk melakukan segmentasi gambar, perlu diketahui terlebih dahulu nilai-nilai piksel pada gambar. Nilai-nilai piksel tersebut digunakan sebagai fitur untuk melakukan *clustering*. Bentuk nilai piksel ada dua macam, dapat berupa nilai intensitas (dengan *range* dari 0 s/d 255) yang menghasilkan gambar berwarna abu-abu atau dapat berupa vektor nilai warna yang menghasilkan gambar berwarna. Warna ketiga komponen suatu piksel pada gambar berwarna tergantung dari ruang warna yang digunakan. Ruang warna yang umum digunakan antara lain (Burger, 2009):

- RGB (*Red*, *Green*, *Blue*): Ruang warna ini mengodekan warna sebagai kombinasi dari tiga warna primer: merah (R), hijau (G), dan biru (B). RGB bernilai positif dan terletak pada kisaran 0 s/d C_{max} . Umumnya nilai C_{max} yaitu 255. Semakin mendekati 0 maka warna komponen semakin gelap sedangkan semakin mendekati 255 maka warna komponen akan semakin cerah. Secara matematis, warna yang mungkin untuk piksel i yaitu sebagai berikut:

$$C_i = (R_i, G_i, B_i)$$

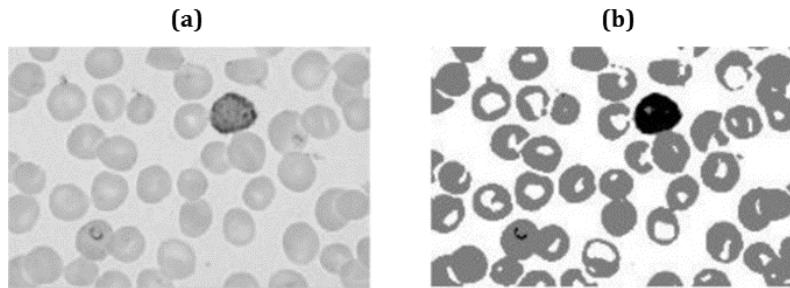
- CIE XYZ: Ruang warna ini dikembangkan setelah dilakukan pengukuran terhadap persepsi visual manusia. Ruang warna ini terdiri dari tiga warna primer imajiner X, Y, Z yang bernilai positif. Komponen Y menunjukkan luminositas (tingkat kecerahan) cahaya. Komponen Z menunjukkan warna biru. Komponen X menunjukkan campuran warna.
- CIE L*a*b: Ruang warna ini dikembangkan dengan tujuan untuk melinearisasi representasi warna sehubungan dengan persepsi warna oleh mata manusia sehingga menciptakan sistem warna yang lebih intuitif. L*a*b banyak digunakan untuk fotografi berkualitas tinggi. Dimensi dalam ruang warna ini adalah luminositas (L) dan dua buah komponen a* dan b*. Komponen a* menyatakan perubahan

⁵⁹ Sumber komponen gambar: <https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html>

warna sepanjang sumbu dari hijau ke merah, sedangkan komponen b^* menunjukkan perubahan warna sepanjang sumbu dari biru ke kuning. L^* bernilai positif dan memiliki *range* dari 0 s/d 100. *Range* untuk nilai a^* dan b^* yaitu -127 s/d 127.

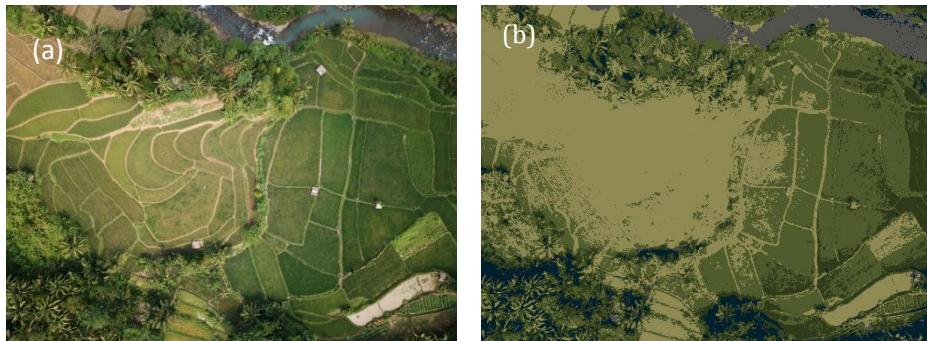
14.2.3. Segmentasi Gambar

Segmentasi gambar adalah proses membagi gambar menjadi kelompok-kelompok piksel, dimana piksel-piksel yang berada dalam satu kelompok memiliki tingkat kemiripan yang tinggi, sedangkan piksel-piksel yang berbeda kelompok memiliki tingkat kemiripan yang rendah (Dhanachandra, 2015). Segmentasi gambar dapat dilakukan pada gambar berwarna maupun gambar skala keabuan. Tujuan dari segmentasi gambar yaitu mengubah representasi dari sebuah gambar menjadi sesuatu yang berarti dan mudah untuk dianalisis. Hal ini dikarenakan objek dan batas dalam gambar lebih mudah untuk dideteksi. Ilustrasi segmentasi gambar pada gambar skala keabuan diilustrasikan seperti Gambar 14.4.



Gambar 14.19. Ilustrasi segmentasi gambar: (a) sebelum segmentasi (b) sesudah segmentasi (Dhanachandra, 2015).

Pemanfaatan dari segmentasi gambar misalnya untuk pemeriksaan kesehatan, analisis kemacetan, pengenalan pola, pengenalan wajah, pengenalan sidik jari, pemosisan objek pada satelit, dan pendekripsi ladang. Contoh pemanfaatan segmentasi gambar pada gambar berwarna untuk pendekripsi ladang ditampilkan pada Gambar 14.5. Pada Gambar 14.5a, bagian ladang yang telah siap panen kurang dapat dibedakan dengan bagian ladang yang masih muda. Dengan dilakukan segmentasi gambar, kedua bagian tersebut dapat lebih mudah dibedakan (Gambar 14.5b). Manfaatnya jika lebih mudah dibedakan yaitu dapat diketahui seberapa luas masing-masing bagian tersebut.



Gambar 14.20. Contoh pemanfaatan segmentasi gambar untuk pendekslan ladang: (a) sebelum segmentasi. (b) sesudah segmentasi.⁶⁰

14.2.4. Algoritma PSO (*Particle Swarm Optimization*)

PSO adalah teknik optimisasi stokastik berbasis populasi yang dimodelkan berdasarkan perilaku sosial kawanan burung (Wong, 2011). Dalam algoritma ini, setiap partikel mewakili solusi potensial untuk masalah optimisasi. Partikel-partikel diterbangkan dalam ruang pencarian dengan kecepatan acak. Tujuan algoritma ini yaitu untuk menemukan posisi partikel yang menghasilkan evaluasi nilai *fitness* (ukuran seberapa besar tingkat kemiripan objek-objek dalam kelompok yang sama + ukuran seberapa besar beda *centroid* antar kelompok) terbaik. Semakin kecil nilai *fitness* partikel berarti hasil *clustering* semakin baik.

Setiap partikel memiliki informasi berikut dalam ruang pencarian (Wong, 2011):

- posisi partikel saat ini
- kecepatan partikel saat ini
- posisi terbaik partikel yang telah dicapai sejauh ini (*pbest*). Posisi ini memiliki nilai *fitness* terbaik untuk partikel tersebut.

Terdapat dua buah pendekatan untuk PSO yaitu (Wong, 2011):

- Global terbaik (*gbest*) yaitu partikel terbaik ditentukan dari seluruh kawanan.
- Lokal terbaik (*lbest*) yaitu kawanan dibagi menjadi lingkungan-lingkungan partikel, kemudian partikel terbaik ditentukan untuk setiap lingkungan.

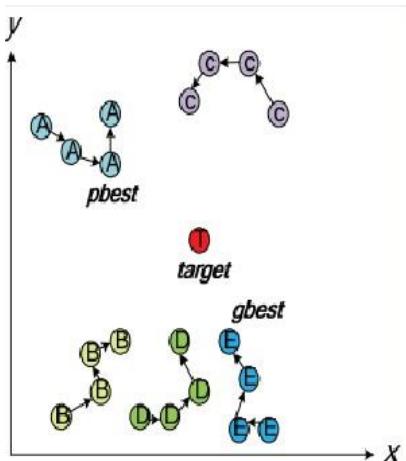
PSO mengubah kecepatan setiap partikel pada setiap waktu sehingga bergerak menuju lokasi *pbest* dan *gbest*. Algoritma PSO diimplementasikan secara umum sebagai berikut (Wong, 2011):

1. Inisialisasi populasi partikel dengan posisi dan kecepatan acak pada ruang masalah d dimensi. Jumlah partikel yang biasanya digunakan yaitu dari 20 sampai 50.
2. Untuk setiap partikel, evaluasi nilai *fitness*.

⁶⁰ Sumber komponen gambar: <https://pxhere.com/en/photo/1524167>

3. Bandingkan evaluasi *fitness* partikel dengan *pbest*. Jika nilainya lebih baik dari *pbest*, maka atur *pbest* ke nilai *fitness* saat ini dan atur posisi *pbest* ke posisi partikel saat ini.
4. Bandingkan evaluasi *fitness* partikel dengan *gbest*. Jika nilainya lebih baik dari *gbest*, maka atur *gbest* ke nilai *fitness* saat ini dan atur posisi *gbest* ke posisi partikel saat ini.
5. Ubah kecepatan dan posisi dari partikel berdasarkan posisi *pbest* dan posisi *gbest*. Secara sederhana, kecepatan partikel juga dapat dianggap sebagai besar perpindahan partikel. Kecepatan partikel pada awalnya besar, lalu perlambatan menurun sehingga daerah pencarian solusi setiap partikel menjadi semakin kecil. Penurunan besar kecepatan partikel ditentukan oleh berat inersia. Pada setiap iterasi, partikel dapat berpindah mendekati posisi *pbest* atau posisi *gbest* dengan ditentukan oleh nilai acak r_1 dan r_2 yang berubah setiap kali iterasi. Besar bobot perpindahan partikel apakah cenderung ke arah posisi *pbest* atau posisi *gbest* ditentukan oleh nilai c_1 dan c_2 yang besarnya sama untuk setiap iterasi.
6. Ulangi langkah 2 sampai 5 hingga kondisi berhenti terpenuhi. Biasanya yang menjadi kondisi yaitu *fitness* yang cukup baik atau jumlah iterasi sudah maksimum.

Ilustrasi dari proses penelusuran partikel pada algoritma PSO ditampilkan pada Gambar 14.6. Pada gambar tersebut terdapat lima partikel. Setiap partikel bertugas mencari posisi yang optimal (paling mendekati target). Pada setiap iterasi, setiap partikel bisa terbang lebih ke arah posisi terbaiknya sendiri (*pbest*) atau lebih ke arah posisi terbaik dari seluruh partikel (*gbest*). Berdasarkan gambar tersebut partikel yang memiliki posisi terbaik yaitu partikel E.



Gambar 14.21. Ilustrasi penelusuran partikel pada PSO (Dereli, 2016).

14.2.5. Algoritma *K-means*

Algoritma *K-means* merupakan algoritma *clustering* bersifat iteratif yang mempartisi dataset menjadi k buah *cluster*. Pada algoritma *K-means*, objek-objek direpresentasikan sebagai titik dalam ruang vektor d dimensi. Setiap titik dapat diberi ID untuk mengetahui titik tersebut masuk ke *cluster* yang mana. Titik dengan ID *cluster* yang sama menunjukkan berada dalam satu *cluster*, sedangkan bila ID-nya berbeda menunjukkan berada dalam *cluster* yang berbeda. Algoritma ini meminimalisir total jarak antara setiap objek dengan *centroid* terdekatnya. Bahasan lebih lengkap tentang algoritma ini dapat dilihat pada Subbab 12.3.

14.2.6. *Silhouette Coefficient*

Analisis dengan memanfaatkan *Silhouette coefficient* merupakan salah satu metode yang dapat digunakan untuk mengukur kualitas *clustering*. Setiap objek pada hasil clustering dievaluasi dengan menilai seberapa baik objek di sebuah klaster dipisahkan dengan objek-objek di klaster lain (seberapa berbeda objek di sebuah klaster dengan objek-objek klaster-klaster lain) dan seberapa berdekatan objek tersebut dengan objek-objek lain dalam klaster yang sama. Untuk keperluan ini, *Silhouette coefficient* dihitung dari tiap objek (yang sudah dilabeli dengan kelompoknya). *Silhouette coefficient* memiliki rentang nilai dari -1 s/d 1, dimana semakin mendekati 1 berarti objek terkelompok dengan semakin baik, sedangkan jika mendekati -1 berarti objek terkelompok dengan makin buruk (cenderung salah). Jika koefisien bernilai 0, objek berada di perbatasan di antara dua kelompok yang berdekatan.

Setelah nilai *Silhouette coefficient* dari seluruh objek di setiap klaster dihitung, kualitas hasil *clustering* secara keseluruhan dapat diukur melalui rata-rata nilai koefisien tersebut.

14.3. Segmentasi Gambar dengan Algoritma PSO dan *K-means*

14.3.1. Penyiapan Data Masukan

Sebelum dilakukan segmentasi gambar, gambar perlu dilakukan pemrosesan terlebih dahulu agar didapatkan hasil segmentasi yang lebih baik. Tahap-tahap yang dilakukan yaitu:

Tahap-1: Merata-ratakan nilai piksel *window* 3x3

Seringkali kemampuan mata manusia dalam melihat objek tergantung dari warna lingkungannya. Untuk mengatasi hal ini, dapat dilakukan perataan nilai piksel untuk setiap *window* 3x3. Ilustrasi dari *window* 3x3 ditampilkan pada Gambar 8. Pada gambar tersebut, yang dimaksud *window* 3x3 yaitu piksel pada indeks (0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), dan (2,2). Piksel pada indeks (1,1) akan diubah nilainya berdasarkan nilai piksel di sekelilingnya. Hal ini juga dilakukan untuk seluruh piksel dalam gambar. Dengan melakukan hal ini, kualitas gambar menjadi lebih baik yang tentunya berpengaruh pada hasil segmentasi.

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)

Gambar 14.7. Ilustrasi window 3x3.

Tahap-2: Automatic contrast adjustment

Seringkali gambar yang akan dilakukan segmentasi cukup rabun dan memiliki kontras yang rendah (Gambar 14.8a). Agar kualitas gambar menjadi lebih baik, diperlukan suatu teknik yaitu teknik *automatic contrast adjustment* (sehingga menjadi seperti Gambar 14.8b). Teknik ini dilakukan dengan cara memetakan nilai piksel terkecil dan terbesar pada gambar masing-masing menjadi bernilai 0 dan 255, lalu memetakan nilai piksel di antaranya secara linear (Burger, 2009). Kualitas gambar menjadi lebih baik karena *range* nilai piksel menjadi lebih besar.



Gambar 14.8. Ilustrasi hasil proses automatic contrast adjusment: (a) sebelum proses automatic contrast adjustment (b) setelah proses automatic contrast adjustment (Burger, 2009).

Tahap-3: Konversi ruang warna menjadi CIE L*a*b

Ruang warna ini dapat menggantikan ruang warna RGB karena pada ruang warna RGB terlalu banyak transisi antara warna biru dan warna hijau, juga antara warna hijau dan warna merah. Banyaknya transisi antara warna biru dan warna hijau mengakibatkan kurangnya warna kuning. Oleh karena itu, ruang warna CIE L*a*b lebih cocok digunakan karena memiliki variasi warna yang lebih banyak. Untuk

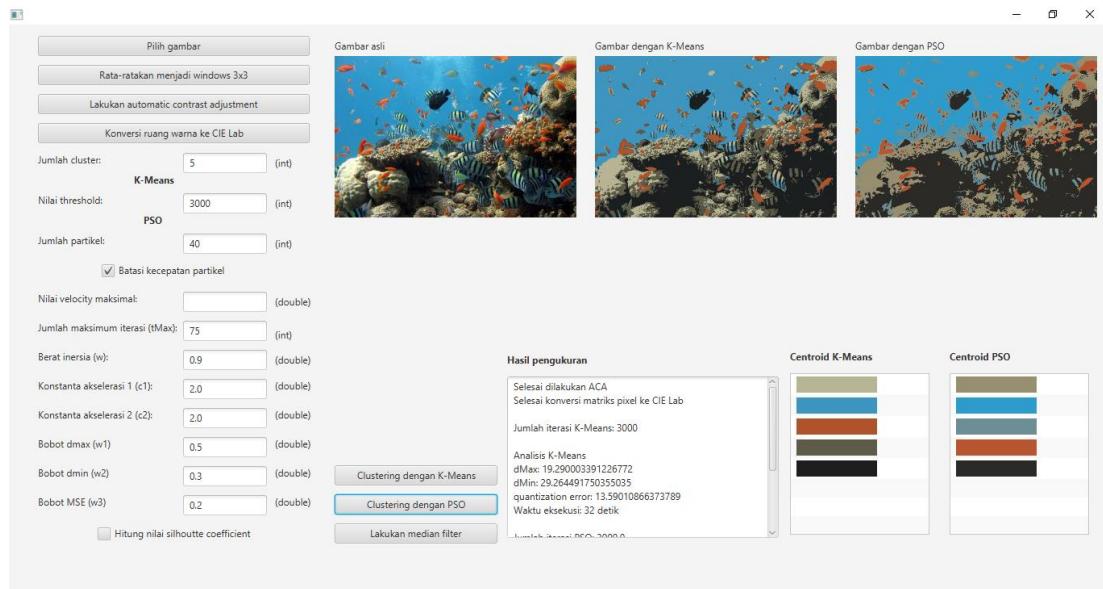
melakukan konversi ruang warna, ruang warna RGB perlu dikonversi terlebih dahulu ke ruang warna CIE XYZ, lalu dikonversi ke ruang warna CIE L*a*b (Zheng, 2018).

14.3.2. Perangkat Lunak Segmentasi Gambar

Pada penelitian ini telah dikembangkan perangkat lunak yang digunakan untuk mensegmentasi gambar. Algoritma dan PSO dan k-Means dimanfaatkan untuk keperluan tersebut.

Dalam mengimplementasikan algoritma PSO untuk segmentasi gambar, sebuah partikel didefinisikan sebagai kumpulan *centroid* dari seluruh *cluster*. Pada proses *clustering* dengan PSO, piksel-piksel gambar dikelompokkan ke *centroid* yang terdekat di dalam partikel, lalu dihitung nilai *fitness-nya*. Baik pada algoritma *K-means* maupun PSO, *centroid* inisial (awal) diisi dengan nilai acak dari 0 s/d 255 untuk gambar skala keabuan, sedangkan pada gambar berwarna berupa nilai acak dari 0 s/d 100 untuk komponen L*, -127 s/d 127 untuk komponen a*, dan -127 s/d 127 untuk komponen b*.

Perangkat lunak yang dibangun dengan menggunakan bahasa pemrograman Java dan *tools* pengembang (IDE) JavaFX. Perangkat lunak menerima masukan berupa gambar, parameter-parameter yang ditentukan oleh pengguna, dan perintah-perintah untuk memproses gambar. Format file gambar yang dapat diproses adalah JPG/JPEG, PNG, dan GIF (bukan animasi). Antarmuka dari perangkat lunak segmentasi gambar ditampilkan pada Gambar 14.9.

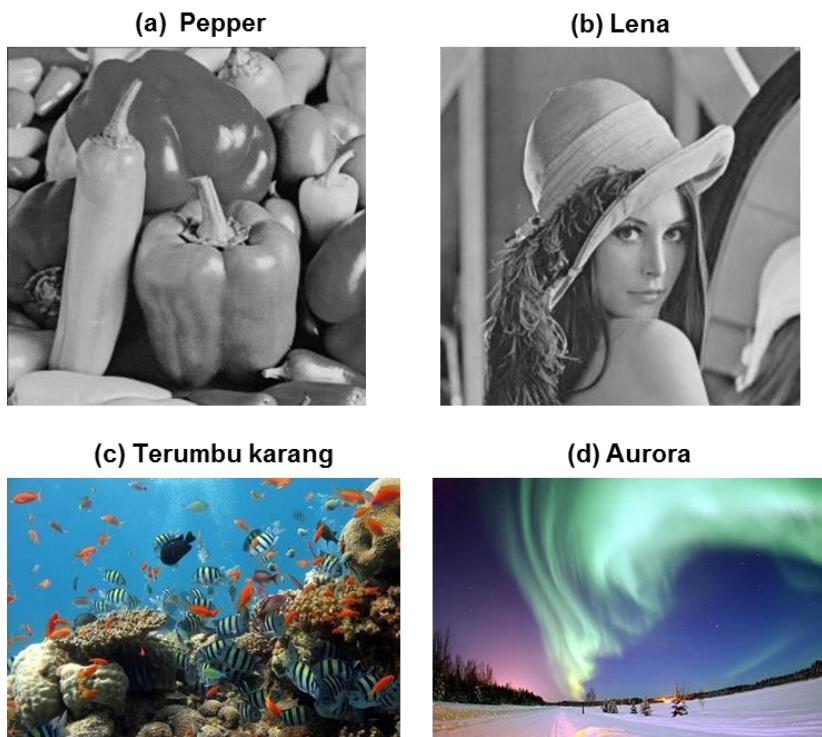


Gambar 14.9. Antarmuka perangkat lunak segmentasi gambar yang telah dikembangkan.

14.4. Eksperimen Segmentasi Gambar

Eksperimen dilakukan terhadap empat buah gambar yang dibedakan ke dalam dua tipe, yaitu skala keabuan (Gambar 11 (a) dan (b)) dan berwarna (Gambar 11 (c) dan (d)). Adapun yang menjadi tujuan eksperimen adalah:

- Untuk mengamati hasil segmentasi gambar menggunakan *clustering* dari algoritma *K-means* dan PSO.
- Membandingkan hasil segmentasi gambar menggunakan algoritma *K-means* dan algoritma PSO terhadap gambar tanpa dipraolah (menggunakan piksel-piksel asli).
- Membandingkan hasil segmentasi gambar menggunakan algoritma *K-means* dan algoritma PSO terhadap gambar dimana piksel dipraolah terlebih dahulu, yaitu nilai-nilai piksel pada *window* 3x3 dirata-rata terlebih dahulu.



Gambar 14.10. Gambar untuk eksperimen: (a) Pepper.jpg⁶¹; (b) Lena.jpg⁶²; (c) Terumbu_karang.jpg⁶³; (d) Aurora.jpg⁶⁴.

⁶¹ Sumber gambar: <https://mingyuanzhou.github.io/Results/BPFAImage/>

⁶² Sumber gambar:

https://www.researchgate.net/publication/3935609_Combined_digital_signature_and_digital_watermark_scheme_for_image_authentication

⁶³ Sumber gambar: <https://pixabay.com/da/photos/fisk-akvarium-hav-fisk-tank-288988/>

⁶⁴ Sumber gambar: <https://id.wikipedia.org/wiki/Aurora>

Dengan tujuan tersebut, eksperimen dilakukan dengan menggunakan perangkat lunak yang telah dikembangkan (Gambar 14.9) dengan langkah-langkah:

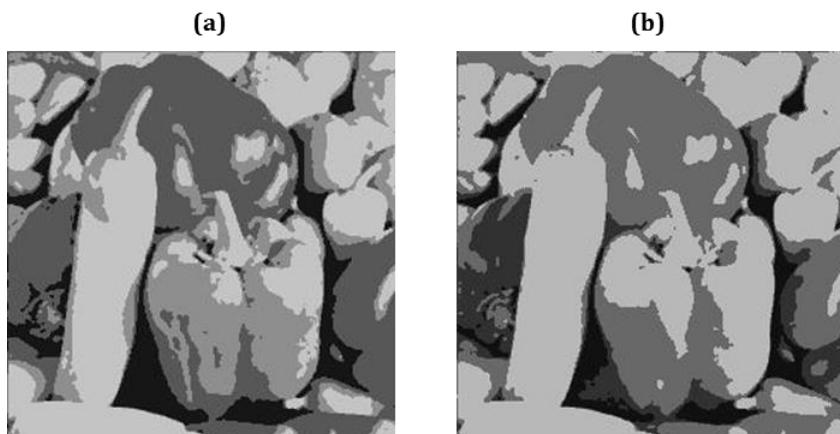
- a. Memasukkan gambar ke perangkat lunak agar piksel gambar dibaca.
- b. Melakukan perataan nilai piksel *window 3x3* (opsional).
- c. Melakukan proses *automatic contrast adjustment*.
- d. Melakukan konversi ruang warna gambar menjadi CIE L*a*b untuk gambar berwarna.
- e. Memasukkan parameter-parameter yang dibutuhkan untuk *clustering* dengan algoritma *K-means* dan PSO.
- f. Melakukan *clustering* dengan *K-means*.
- g. Melakukan *clustering* dengan PSO.
- h. Melakukan proses *median filter* untuk menghilangkan noda pada gambar hasil *clustering* (opsional).

Langkah-langkah di atas dilakukan sebanyak 25 kali untuk masing-masing gambar. Setiap kali dijalankan, nilai *Silhouette coefficient* dicatat. Setelah selesai, nilai koefisien tersebut dirata-rata.

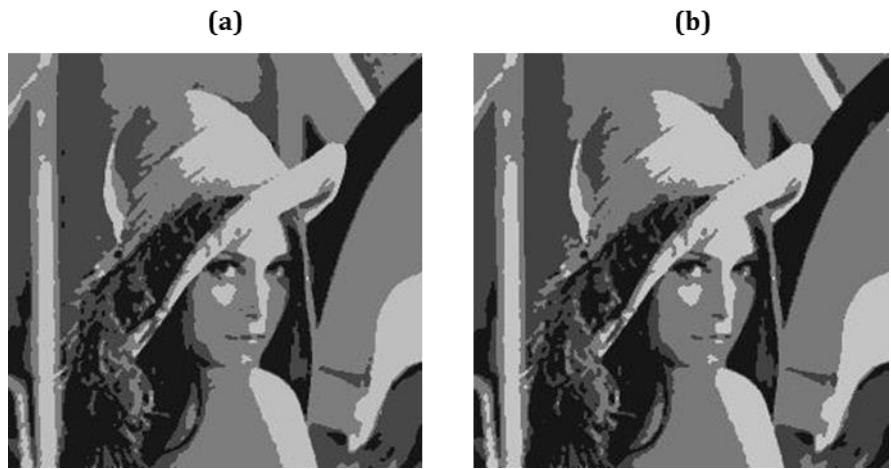
Hasil Segmentasi

Hasil eksperimen dengan piksel asli maupun dengan piksel rata-rata *window 3x3*, jika dilihat dengan mata menunjukkan hasil yang serupa (sama). Contoh hasil eksekusi perangkat lunak diberikan pada Gambar 14.11 s/d 14.14.

Pada Gambar 14.11 dan 14.12 terlihat objek tersegmentasi berdasarkan tingkat kecerahan objeknya. Pada kedua gambar tersebut, objek yang lebih cerah dan lebih gelap dapat lebih mudah teridentifikasi. Pada Gambar 14.13 dan 14.14 terlihat objek tersegmentasi berdasarkan warnanya. Dari Gambar 14.13 dapat diketahui bagian mana yang merupakan terumbu karang dan bagian mana yang merupakan ikan. Dari Gambar 14.14 dapat diketahui apa saja lapisan dari aurora dan apa saja lapisan dari langit. Demikian, gambar tersegmentasi menjadi segmen-segmen gambar yang sesuai dengan jumlah *cluster* yang diinginkan oleh pengguna.



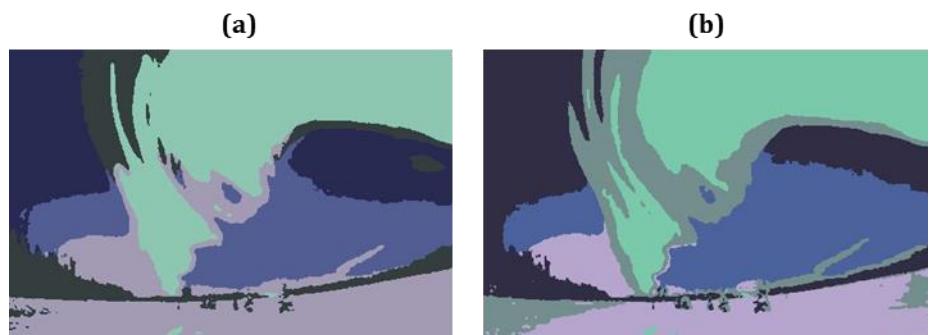
Gambar 14.11. Hasil segmentasi gambar Pepper: (a) K-means (b) PSO.



Gambar 14.12. Hasil segmentasi gambar Lena: (a) k-Means (b) PSO.



Gambar 22. Hasil segmentasi gambar Terumbu Karang: (a) k-Means (b) PSO.



Gambar 23. Hasil segmentasi gambar Aurora: (a) k-Means (b) PSO.

Perbandingan Hasil Segmentasi dengan algoritma *K-means* terhadap PSO

Pada Subbab 14.2.6. telah dipaparkan bahwa salah satu cara untuk mengukur kualitas hasil clustering (klaster) adalah dengan menghitung *Silhouette coefficient*. Untuk mengetahui algoritma mana yang lebih baik (apakah k-Means atau PSO), di sini diberikan perbandingan nilai koefisien tersebut. Hasil perhitungan koefisien dari hasil eksperimen dengan piksel asli dan dengan merata-ratakan window 3x3 diberikan pada Tabel 14.1 dan 14.2.

Tabel 14.4. Perbandingan hasil segmentasi dengan K-means dan PSO pada gambar dengan piksel asli.

Gambar	Tipe gambar	Jumlah cluster	<i>Silhouette coefficient</i>	
			<i>K-means</i>	PSO
Pepper	skala keabuan	4	0.6056 ± 0.0211	0.6124 ± 0.0036
Lena	skala keabuan	4	0.5397 ± 0.0209	0.5784 ± 0.0007
Terumbu karang	berwarna	5	0.5296 ± 0.0237	0.5437 ± 0.0386
Aurora	berwarna	5	0.3907 ± 0.0171	0.3886 ± 0.0237

Keterangan pada nilai Silhouette coefficient: angka di depan tanda “±” merupakan rata-rata, angka di belakang tanda “±” merupakan simpangan baku.

Dari Tabel 14.1 terlihat bahwa nilai *silhouette coefficient* untuk PSO pada gambar skala keabuan lebih baik dibanding *K-means*. Untuk gambar berwarna, PSO menghasilkan nilai koefiesien yang lebih baik pada satu gambar saja.

Tabel 14.5. Perbandingan hasil segmentasi dengan K-means dan PSO dengan rata-rata nilai piksel window 3x3.

Gambar	Tipe	Jumlah cluster	<i>Silhouette coefficient</i>	
			<i>K-means</i>	PSO
Pepper	skala keabuan	4	0.6034 ± 0.0204	0.6144 ± 0.0013
Lena	skala keabuan	4	0.5593 ± 0.0183	0.5773 ± 0.0005
Terumbu karang	berwarna	5	0.5020 ± 0.0374	0.5368 ± 0.0362
Aurora	berwarna	5	0.3934 ± 0.0127	0.3981 ± 0.0219

Keterangan pada nilai Silhouette coefficient: angka di depan tanda “±” merupakan rata-rata, angka di belakang tanda “±” merupakan simpangan baku.

Dari Tabel 14.2 terlihat bahwa nilai *silhouette coefficient* untuk PSO pada gambar skala keabuan maupun berwarna lebih baik dibanding *K-means*. Hal ini menunjukkan bahwa PSO berkinerja lebih baik jika digunakan untuk melakukan segmentasi pada gambar (baik skala keabuan maupun berwarna) jika dilakukan rata-rata nilai piksel pada *window* 3x3 terlebih dahulu.

14.5. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan-kesimpulan sebagai berikut:

1. Algoritma PSO dan k-Means dapat dimanfaatkan untuk segmentasi gambar. Pada gambar skala keabuan fitur dari himpunan data yang diproses algoritma berupa sebuah nilai intensitas piksel, sedangkan pada gambar berwarna fiturnya berupa vektor tiga dimensi dengan ruang warna CIE L*a*b.
2. Secara umum algoritma PSO berkinerja lebih baik dibanding k-Means yang ditunjukkan dengan rata-rata nilai koefisien Silhouette yang lebih tinggi. Namun pada pemrosesan gambar berwarna menggunakan piksel asli, pada sebuah gambar, k-Means berkinerja lebih baik.
3. Dengan melakukan segmentasi gambar, dapat diperoleh gambar baru lain dengan objek-objek yang dapat diidentifikasi dengan lebih mudah. Hasil ini dapat dimanfaatkan lebih lanjut, misalnya, untuk pengenalan bentuk-bentuk secara otomatis dari citra. Teknik yang dapat dimanfaatkan untuk keperluan ini antara lain adalah teknik klasifikasi.

Kesimpulan-kesimpulan di atas diperoleh berdasar eksperimen dengan jumlah gambar yang terbatas, hanya 4 buah gambar. Eksperimen lanjutan dengan menggunakan gambar yang lebih banyak dan lebih variatif dibutuhkan agar dapat dihasilkan kesimpulan-kesimpulan dengan justifikasi yang lebih kuat.

Referensi

- (Burger, 2009) W. Burger, M.J. Burge, *Principles of Digital Image Processing*, London: Springer-Verlag London Limited, 2009.
- (Dereli, 2016) S. Dereli dan R. Köker, In a research on how to use inverse kinematics solution of actual intelligent optimization method, *ISITES2016* (2016) 1, 506–512.
- (Dhanachandra, 2015) N. Dhanachandra, K. Manglem, Y.J. Chanu, Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm, *Procedia Computer Science* (2015) 54, 764-771.
- (Gonzalez, 2007) R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Pearson Education International, USA, 2007.
- (Wahyuni, 2016) I. Wahyuni, Y.A. Auliya, A. Rahmi, W.F. Mahmudy, Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan Hybrid Particle Swarm Optimization dengan K-Means, *Jurnal Ilmiah Teknologi dan Informasi ASIA (JITIKA)*, Vol. 10, 24-33, 2016.
- (Wong, 2011) M.T. Wong, X. He, W. C. Yeh, Image clustering using Particle Swarm Optimization, *IEEE*, 262-268, 2011.

(Zheng, 2018) X. Zheng, Q. Lei, R. Yao, Y. Gong, Q. Yin, Image segmentation based on adaptive K-means algorithm, *EURASIP Journal on Image and Video Processing*, Vol. 68, 1-10, 2018.

Biografi Editor dan Para Pengarang

Informasi tentang para editor dan pengarang buku dapat dilihat pada halaman website setiap dosen dengan URL yang diberikan di bawah ini (diurutkan menurut kemunculan bab yang ditulis setiap pengarang):

Dr. Ir. Veronica S. Moertini, MT

<http://informatika.unpar.ac.id/dosen/moertini/>

Mariskha Tri Adithia, SSi, MSc, PDEng

<http://informatika.unpar.ac.id/dosen/mariskha/>

Natalia, S.Si, M.Si

<http://informatika.unpar.ac.id/dosen/natalia/>

Vania Natali, S.Kom, M.T.

<http://informatika.unpar.ac.id/dosen/vania-natali/>

Kristopher David Harjono, M.T.

<http://informatika.unpar.ac.id/dosen/kristopher-h/>

Chandra Wijaya, S.T., M.T.

<http://informatika.unpar.ac.id/dosen/chandraw/>

Raymond Chandra Putra, S.T., M.T.

<http://informatika.unpar.ac.id/dosen/raymond-chandra/>

Husnul Hakim, S.Kom., M.T.

<http://informatika.unpar.ac.id/dosen/husnulhakim/>

Pascal Alfadian Nugroho, S.Kom, M.Comp

<http://informatika.unpar.ac.id/dosen/pascal/>

Gede Karya, S.T., M.T., CISA, IPM

<http://informatika.unpar.ac.id/dosen/gkarya/>

Muhammad Ravi

Pada saat menyiapkan bab buku ini, Ravi berstatus sebagai mahasiswa di Jurusan Teknik Informatika UNPAR.

Hereza Ardhitya

Pada saat menyiapkan bab buku ini, Hereza berstatus sebagai mahasiswa di Jurusan Teknik Informatika UNPAR.

Alvinus Sutendy

Pada saat menyiapkan bab buku ini, Alvinus berstatus sebagai mahasiswa di Jurusan Teknik Informatika UNPAR.

Program Data Science UNPAR

Sebagai jawaban atau tindak lanjut dari kebutuhan tenaga kerja dengan skill dan keahlian pada bidang Data Science (yang telah dipaparkan pada Bab 1), pada tahun 2019 Jurusan Teknik Informatika UNPAR membuka Program Data Science. Untuk tingkat S1, program Data Science tersebut merupakan salah satu yang pertama dibuka di Indonesia.

Agar lulusannya memenuhi kebutuhan nyata pada dunia kerja, kurikulum Program Data Science UNPAR dirancangan dengan tiga strategi utama, yaitu:

1. Kuliah-kuliah yang terintegrasi dengan sertifikasi dari organisasi pemberi sertifikasi yang terkemuka.
2. Kerja praktik dan tugas akhir yang mencakup 22 sks atau sekitar 15 % dari total jumlah SKS lulus (144 SKS).
3. Pada tahap akhir mahasiswa dapat memilih antara skripsi atau tugas akhir. Skripsi merupakan suatu proyek penelitian di bidang Data Science, sedangkan tugas akhir merupakan proyek aplikatif yang dikerjakan dengan cara magang di sebuah perusahaan yang membutuhkan data scientist.

Fokus dari Program Data Science UNPAR adalah untuk membekali lulusannya agar siap bekerja di industri sebagai data scientist atau data engineer pada masalah-masalah big data. Oleh karena itu topik-topik mata kuliah pilihan yang ada pada Program Data Science di Program Studi Teknik Informatika UNPAR dirancang untuk membekali lulusannya dengan pengetahuan-pengetahuan yang dibutuhkan untuk memproses, menganalisis, dan mempresentasikan hasil analisis dari Big Data.

Data Science memiliki keterkaitan yang sangat erat dengan industri, karena itu untuk membekali lulusannya dengan skill dan keahlian yang sesuai, pembelajaran pada Program Data Science UNPAR memanfaatkan bahasa pemrograman dan tools yang banyak digunakan oleh industri. Contoh bahasa pemrograman, tools dan teknologi yang digunakan dalam kuliah-kuliah Program Data Science UNPAR adalah:

- Bahasa Python dan library-nya untuk pemrosesan dan visualisasi data (Numpy, Pandas, Matplotlib, dll).
- Bahasa R untuk komputasi dengan statistika dan visualisasi data.
- Library machine learning pada Python yang populer seperti Scikit-learn.
- Framework big data Hadoop yang berfungsi untuk menyimpan, mengelola dan memproses big data.
- Ekosistem Hadoop, seperti Hive (untuk data warehouseing), HBase (basisdata untuk big data), Kafka (untuk pemrosesan data stream).
- Spark, yang merupakan mesin pemroses big data secara umum dan dimanfaatkan untuk berbagai keperluan.
- Library Spark untuk melakukan kueri SQL (Spark SQL), machine learning (Spark MLLib dan Spark ML), komputasi graf (GraphX) dan pemrosesan data stream (Spark Streaming).

Proses pembelajaran ini dilakukan secara *blended learning*, yang merupakan gabungan antara pembelajaran tatap muka (luring) dan online (daring). Para perkuliahan, mahasiswa juga dipersiapkan

untuk mengambil sertifikasi internasional di bidang Data Science. Contoh sertifikasi-sertifikasi yang dapat diambil oleh mahasiswa Program Data Science di antaranya adalah :

- IBM Professional Data Science Certificate
- Google Cloud Platform Big Data and Machine Learning Fundamentals

Dengan kurikulum dan metoda pembelajaran yang telah dirancang, lulusan dari Program Data Science UNPAR diarahkan untuk menjadi:

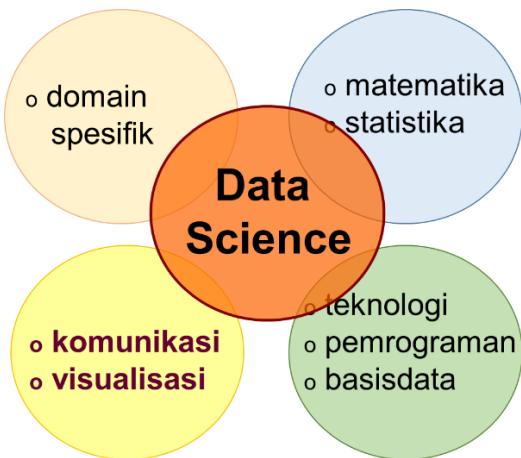
- Data Scientist
- Data Engineer
- Mahasiswa studi lanjut ke jenjang magister.

Untuk dapat menyelesaikan Program Data Science, seorang mahasiswa/i harus menyelesaikan semua mata-kuliah wajib pada kurikulum tahun 2018 dan mengambil 20 SKS dari mata kuliah pilihan program Data Science. Adapun daftar mata kuliah-mata pilihan tersebut diberikan di tabel di bawah ini:

Mata Kuliah Pilihan	Jumlah Sks
Pengantar Data Science	2
Proyek Data Science 1	3
Proyek Data Science 2	3
Data Science pada Domain Spesifik	3
Metode Numerik	3
Statistika dengan R	3
Statistika Multivariat dengan R	3
Sistem Kecerdasan Bisnis	3
Pengantar Penambangan Data dengan Python	3
Penambangan Data (Data Mining)	3
Pemrosesan Bahasa Alami (<i>Natural Language Processing</i>)	3
Pola Komputasi Big Data	3
Basis Data dan Pemrograman SQL untuk Big Data	3
Analisis Big Data	3
Teknologi Big Data dan Cloud Computing	3

Pemetaan Mata Kuliah dengan Keahlian/Skill Multi-Disiplin Data Scientist

Pada Bab 1 telah dipaparkan bahwa seorang data scientist memiliki keahlian multi-disiplin seperti ditunjukkan pada Gambar 1 di bawah ini.



Gambar 1. Bidang-bidang multi-disiplin pada data science.

Kurikulum Program Data Science UNPAR sudah dirancang agar memenuhi/mengisi semua keahlian/skill yang dibutuhkan untuk membekali lulusan menjadi seorang data scientist. Berikut ini pemetaan bidang Data Science dengan matakuliah wajib dan pilihan berdasar kurikulum Program Studi Teknik Informatika tahun 2018.

Bidang Keahlian Data Science	Contoh Mata Kuliah (Wajib dan Pilihan)
Matematika, Computational Thinking dan Algoritma	Matematika Dasar (4 sks), Matematika Diskret (3 sks), Pemodelan untuk Komputasi (3 sks), Matriks dan Ruang Vektor (3 sks), Struktur Diskret (3 sks), Metode Numerik (3 sks), Pengantar Sistem Cerdas (3 sks)
Statistika	Statistika untuk Komputasi (3 sks), Statistika dengan R (3 sks), Analisis Multivariat dengan R (3 sks), Pengantar Penambangan Data dengan Python (3 sks), Penambangan Data/Data Mining (3 sks), Pola Komputasi Big Data (3 sks), Analisis Big Data (3 sks), Proyek Data Science 1 dan 2 (6 sks)
Pemrograman, algoritma, sistem, teknologi	Dasar Pemrograman (3 sks), Algoritma dan Struktur Data (3 sks), Pemrograman Berorientasi Objek (3 sks), Desain dan Analisis Algoritma (3 sks), Pemrograman Berbasis Web (3 sks), Pola Komputasi Big Data (3 sks), Rekayasa Perangkat Lunak (3 sks), Pengolahan Bahasa Alami/Natural Language Processing (3 sks)

Bidang Keahlian Data Science	Contoh Mata Kuliah (Wajib dan Pilihan)
Basisdata	Manajemen Informasi dan Basis Data (4 sks), Teknologi Basis Data (3 sks), Sistem Kecerdasan Bisnis (3 sks), Basisdata dan Pemrograman SQL untuk Big Data (3 sks)
Teknologi, algoritma, pemrograman dan visualisasi	Teknologi Big Data (3 sks), Analisis Big Data (3 sks), Pengantar Penambangan Data dengan Python (3 sks), Data Mining (3 sks)
Visualisasi dan Algoritma	Pengantar Data Science (2 sks), Pengantar Penambangan Data dengan Python (3 sks), Penambangan Data/Data Mining (3 sks), Sistem Kecerdasan Bisnis (3 sks), Analisis Big Data (3 sks), Proyek Data Science 1 dan 2 (6 sks)
Komunikasi	Teknik Presentasi (2 sks), Bahasa Indonesia (2 sks), Penulisan Ilmiah (2 sks), Skripsi/Tugas Akhir (8 sks) dan praktik pada hampir semua mata kuliah lainnya
Domain Spesifik	Mata kuliah dari jurusan lain, Proyek Data Science 1 dan 2 (6 sks), Data Science pada Domain Spesifik (3 sks), Kerja Praktek 1 (2 sks), Kerja Praktek 2 (3 sks), Kerja Praktek 3 (4 sks), Kerja Praktek 4 (5 sks), Skripsi/Tugas Akhir (8 sks)

Fasilitas Laboratorium

Lab Perkuliahan dan Praktikum

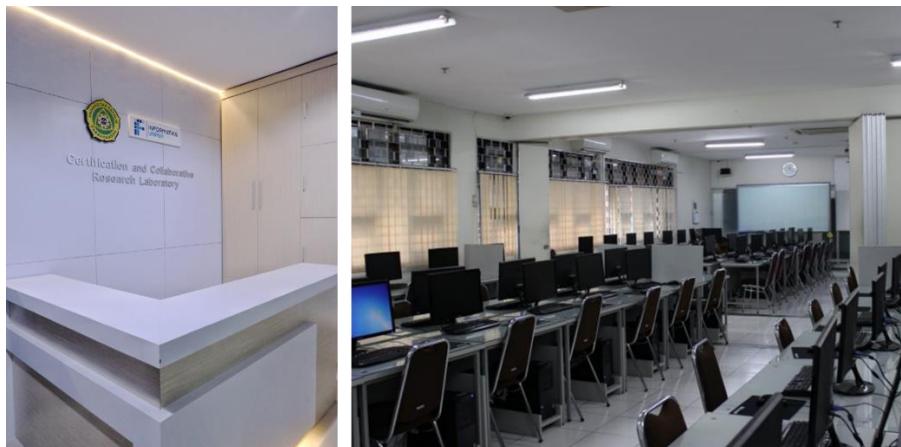
Laboratorium Komputasi FTIS telah berdiri sejak tahun 2008. Laboratorium ini terdiri dari 6 ruangan, yang masing-masing memiliki fungsi khusus. Empat ruangan digunakan untuk perkuliahan/praktikum, dengan kapasitas 40, 45, 35 dan 35 komputer per ruang (lihat Gambar 1). Dua ruangan lainnya digunakan untuk keperluan skripsi mahasiswa dan penelitian dosen yang berisi masing-masing 10 komputer.

Lab Big Data dan Data Science

Program Data Science memiliki dua klaster big data, dimana Hadoop, Spark, Scoop, Hive, Hbase, Zookeeper, Kafka, dll (bahasan teknologi ini dapat dilihat pada Bab 10) sudah beroperasi pada kedua klaster tersebut (lihat Gambar 2). Masing-masing klaster terdiri dari komputer sebuah komputer master dan 9 komputer slave. Tiap komputer memiliki CPU dengan 6 buah core dan memori (RAM) berkapasitas antara 16 Gb s/d 32 Gb.

Dua klaster big data tersebut dimanfaatkan untuk:

- Praktek mahasiswa peserta matakuliah di bidang big data (Teknologi Big Data, Pola Komputasi Big Data, Basisdata dan Pemrograman SQL untuk Big Data, dan Analisis Big Data)
- Penelitian mahasiswa dan dosen di bidang big data
- Pelatihan bagi peserta kursus (publik) di bidang big data (dengan sertifikasi)
- Penggunaan lainnya (misalnya: lomba analisis big data dan kerja-sama penelitian dengan organisasi lain di lingkungan UNPAR maupun universitas/lembaga/instansi di luar UNPAR).



Gambar 2. Contoh lab perkuliahan dan praktikum.



Gambar 3. Klaster big data di lab Program Data Science UNPAR.