Editor : Dudih Gustian

DATA MINING

Rinci Kembang Hapsari | Titin Sumarni | Taufik Hidayatulloh | Farida | Herianto | Febie Elfaladonna | Nurul Aini | Nurfitria | Heri Kurniawan | Kraugusteeliana | Lestari Yusuf | Yulizar Widiatama |

DATA MINING

DATA MINING

Rinci Kembang Hapsari
Titin Sumarni
Taufik Hidayatulloh
Farida
Herianto
Febie Elfaladonna
Nurul Aini
Nurfitria
Heri Kurniawan
Kraugusteeliana
Lestari Yusuf
Yulizar Widiatama

Penerbit :

Anggota IKAPI
No. 428/JBA/2022

DATA MINING

Penulis:

Rinci Kembang Hapsari, Titin Sumarni, Taufik Hidayatulloh, Farida, Herianto, Febie Elfaladonna, Nurul Aini, Nurfitria, Heri Kurniawan, Kraugusteeliana, Lestari Yusuf, Yulizar Widiatama

ISBN: 978-623-8191-63-5, 978-623-8191-64-2 (PDF)

Editor: Dudih Gustian
Tata Letak: Handi Mardiana, A.Md
Desain Sampul: Muhammad Nafis Ridhwan, S. Sos

Penerbit: INDIE PRESS
Redaksi:
Jl. Antapani VI, No 1B, Ankid, Antapani, Bandung 40291
Telp/Faks: (022) 20526377
Website: www.indiepress.id |e-mail: indiepressbooksid@gmail.com

Cetakan Pertama:

11 Agustus 2024

Ukuran:
iii, 193, Uk: 15,5 x 23 cm

Hak Cipta 2024, Indie Press dan Penulis
Isi diluar tanggung jawab percetakan
Copyright © 2024 by Indie Press
All Right Reserved

Hak cipta dilindungi undang-undang Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku initanpa izin tertulis dari Penerbit.

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya sehingga buku kolaborasi dalam bentuk buku rampai konversi energi dapat dipublikasikan dan dapat sampai di hadapan pembaca. Buku rampai ini disusun oleh sejumlah akademisi dan praktisi sesuai dengan kepakarannya masing-masing. Buku ini diharapkan dapat hadir memberi kontribusi positif terkait dengan konversi energi.

Buku rampai konversi energi ini mengacu pada pendekatan konsep teoritis dan contoh penerapan. Buku ini terdiri atas 12 bab yang dibahas secara rinci, diantaranya: Pengantar Data Mining, Preprocessing Data (Manipulasi & Visualisasi), Preprocessing Data (Normalisasi Data), Klasifikasi & K-NN, Validasi Model dalam Klasifikasi, Decission Tree, Association Rule, Clustering, Analisa Cluster, Predictive Mining, Konsep Dasar Test Mining dan Text Mining, Mesin Pencarian.

Kami menyadari bahwa tulisan ini jauh dari kesempurnaan dan kekurangan. Oleh sebab itu, kami tentu menerima masukan dan saran dari pembaca demi penyempurnaan lebih lanjut.

Akhirnya kami mengucapkan terima kasih yang tak terhingga kepada semua pihak yang telah mendukung dalam proses penyusunan dan penerbitan buku ini, secara khusus kepada Penerbit Indie Press. Semoga buku ini dapat bermanfaat bagi pembaca sekalian.

Bandung, 11 Agustus 2024

Editor

DAFTAR ISI

Bab 1 Pendahuluan

1.1 Apa itu Data Mining?

Konsep Data Mining telah ada sejak tahun 1960-an. Pada awalnya, Data Mining lebih merupakan cabang dari statistik dan analisis data yang dikenal sebagai "pencarian pola" atau "penambangan pola" (Yahya, 2022), (Baihaqie, 2023). Pada waktu itu, teknik-teknik ini digunakan terutama dalam konteks ilmu sosial dan penelitian ilmiah. Dalam beberapa dekade berikutnya, dengan munculnya komputer yang lebih kuat dan teknologi penyimpanan data yang lebih canggih, kemampuan untuk memproses dan menganalisis data meningkat secara signifikan. Hal ini membuka pintu bagi pengembangan teknik-teknik Data Mining yang lebih kompleks dan efektif.

Pada tahun 1980-an dan 1990-an, perkembangan dalam bidang kecerdasan buatan, statistik, dan ilmu komputer memungkinkan pengembangan algoritma-algoritma baru yang lebih efisien dan kuat untuk analisis data (Putra et al., 2024), (Putra et al., 2023). Didukung pula dengan perkembangan teknologi hardware yang menyediakan kemampuan komputasi luar biasa memungkinkan pengolahan data besar, serta dengan perkembangan algoritma, perangkat lunak khusus untuk Data Mining mulai dikembangkan. Sehingga Data Mining menjadi lebih populer dan diperdebatkan sebagai aplikasi dibandingkan dengan teknologi.

Data Mining juga telah menjadi disiplin ilmu yang dibangun dalam domain kecerdasan buatan (*Artificial intelligence*) dan rekayasa pengetahuan (*Knowledge Engineering*) (Sudarsono et al., 2021). Seiring dengan kesadaran akan potensi besar yang dimiliki oleh data, Data Mining mulai diterapkan di berbagai industri, termasuk bisnis, keuangan, kedokteran, ilmu sosial, dan

lain-lain. Perusahaan-perusahaan mulai menggunakan Data Mining untuk memahami perilaku pelanggan, memprediksi tren pasar, dan meningkatkan efisiensi operasional mereka.

Dalam beberapa tahun terakhir, dengan ledakan data dari berbagai sumber seperti media sosial, sensor, dan perangkat mobile, Data Mining telah berkembang menjadi bidang yang lebih penting dari sebelumnya. Konsep Big Data Analytics, yang mencakup teknik-teknik Data Mining untuk data yang sangat besar dan kompleks, telah menjadi fokus utama dalam industri dan akademisi.

Dimana Data Mining dapat didefiniskan sebagai proses pengumpulan dan pengolahan data yang bertujuan untuk mengekstrak informasi penting yang tersembunyi pada data yang besar dan kompleks. Ini bukan sekadar tentang mengekstrak informasi yang sudah kita ketahui, tetapi juga tentang menemukan pola yang tidak terduga, hubungan yang menarik, dan pengetahuan yang berharga yang mungkin tidak dapat terlihat secara langsung oleh manusia. Data Mining memungkinkan kita untuk mengubah data mentah menjadi wawasan yang bermanfaat, yang dapat digunakan untuk mengambil keputusan yang lebih baik, memprediksi tren, mengoptimalkan proses bisnis, dan banyak lagi.

Data Mining merupakan proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan pembelajaran mesin (*machine learning*) mengindentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database yang terkait (Kharis & Zili, 2022). Definisi lainnya adalah proses-proses pembelajaran berbasis induksi (*Induction-Base Learning*), proses pembentukan defenisi-defenisi konsep umum yang dilakukan dengan cara mengobservasi contoh-contoh spesifik dari konsep-konsep yang akan dipelajari. *Knowledge Discovery in Database* (KDD) adalah penerapan suatu langkah dari proses KDD.

1.2 Kenapa Belajar Data Mining

Data Mining adalah alat yang kuat untuk mendapatkan pemahaman mendalam tentang data dan membuat keputusan yang didukung oleh bukti (Sugiana & Musty, 2023). Dalam era di mana data menjadi aset yang sangat berharga bagi perusahaan dan organisasi, kemampuan untuk menggali wawasan dari data tersebut menjadi keterampilan yang sangat dicari. Terdapat beberapa alasan lainnya mengapa belajar data mining, yaitu:

Mengubah Data Menjadi Wawasan: Data adalah aset berharga, tetapi tanpa pemahaman yang mendalam tentang bagaimana menganalisis dan menginterpretasikannya, data hanya menjadi kumpulan angka dan fakta. Data Mining memungkinkan untuk mengubah data tersebut menjadi wawasan yang berharga, yang dapat digunakan untuk mengambil keputusan yang lebih baik dan mencapai tujuan bisnis.

Keputusan Berbasis Bukti: Dalam dunia yang semakin kompleks dan berubah dengan cepat, mengandalkan intuisi atau asumsi saja tidaklah cukup. Data Mining memungkinkan untuk membuat keputusan yang didukung oleh bukti-bukti dan analisis data yang kuat. Dengan memahami pola dan tren dalam data, kita dapat mengambil keputusan yang lebih terinformasi dan lebih efektif.

Mengenal Pelanggan dengan Lebih Baik: Dalam dunia bisnis, pemahaman yang mendalam tentang perilaku pelanggan sangat penting. Data Mining memungkinkan untuk menganalisis pola pembelian, preferensi produk, dan kebutuhan pelanggan secara lebih akurat, sehingga perusahaan dapat menyediakan layanan dan produk yang lebih sesuai dengan kebutuhan mereka. Meningkatkan Efisiensi dan Produktivitas: Dengan menerapkan teknik Data Mining, dapat mengidentifikasi area di mana proses bisnis dapat ditingkatkan, sumber daya dapat dioptimalkan, dan waktu serta biaya dapat dihemat. Ini dapat membantu organisasi

meningkatkan efisiensi operasional dan produktivitas secara keseluruhan.

Meningkatkan Prediksi dan Perencanaan: Data Mining membantu dalam membuat prediksi yang lebih akurat tentang tren pasar, permintaan produk, dan hasil bisnis di masa depan. Ini memungkinkan perencanaan yang lebih baik dan pengambilan keputusan yang lebih strategis. Dengan memahami tren yang mungkin terjadi di masa depan, perusahaan dapat merencanakan langkah-langkah yang diperlukan untuk menghadapinya.

Inovasi dan Penemuan Baru: Data Mining dapat menjadi sumber inspirasi untuk inovasi dan penemuan baru. Dengan menganalisis data secara mendalam, dapat mengidentifikasi tren baru, pola yang tidak terduga, dan peluang-peluang yang belum terpikirkan sebelumnya. Ini memungkinkan untuk mengembangkan produk dan layanan baru yang dapat menghasilkan nilai tambah bagi perusahaan atau organisasi.

Dengan memahami konsep dan teknik Data Mining, Anda akan dapat menjadi seorang praktisi yang mahir dan berkontribusi secara signifikan dalam berbagai bidang, mulai dari bisnis dan keuangan hingga ilmu pengetahuan dan teknologi. Dengan demikian, belajar Data Mining merupakan kunci untuk memahami dan mengambil manfaat dari potensi besar yang dimiliki oleh data dalam dunia modern ini.

1.3 Tujuan Data Mining

Tujuan utama dari Data Mining adalah untuk menemukan pola atau informasi yang berguna yang tidak secara langsung terlihat atau dapat diprediksi oleh manusia (Kharis & Zili, 2022). Ini melibatkan pencarian untuk pola yang bermakna, pengelompokan data menjadi kategori yang berbeda, identifikasi hubungan antara variabel, dan membuat prediksi tentang hasil di masa depan.

Dengan melakukan data mining, perusahaan dapat mendapatkan banyak manfaat. Beberapa manfaat dari data mining adalah:

1. Memudahkan pengambilan keputusan. Dimana perusahaan tanpa adanya penundaan karena penilaian manusia bisa terus melakukan analisa dan mengotomatisasi keputusan rutin.
2. Membuat prediksi akurat untuk perencanaan. Berdasarkan tren masa lalu dan kondisi saat ini, maka data mining mampu membantu tahapan perencanaan dan memberikan informasi yang tepat untuk membuat prediksi
3. Pengurangan biaya. Data mining memungkinkan perusahaan menggunakan alokasi dana lebih efisien karena otomatisasi pengambilan keputusan dapat mengurangi biaya.
4. Mendapat wawasan tentang pelanggan. Dengan mengetahui karakteristik pelanggan sehingga dapat merancang strategi yang dapat meningkatkan pengalaman pelanggan dengan tepat.

1.4 Tipe Data Mining

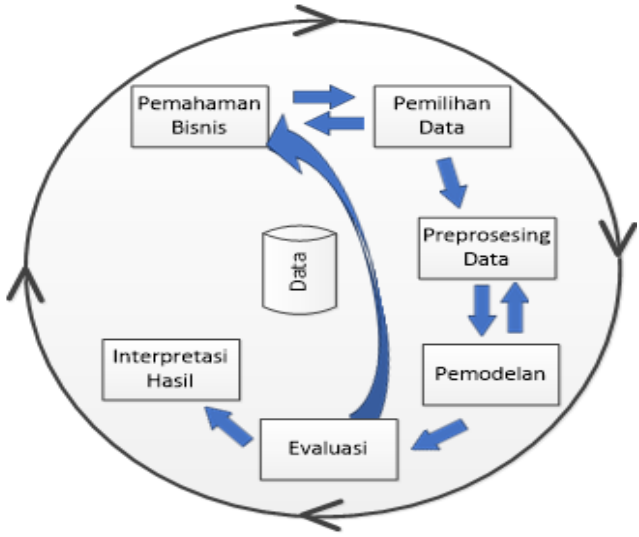
Data Mining dapat dibagi menjadi beberapa kategori berdasarkan jenis informasi yang dicari dan teknik yang digunakan. Kategori utama meliputi:

1. Klasifikasi: Mengklasifikasikan data ke dalam kategori yang telah ditentukan berdasarkan atribut yang ada.
2. Regresi: Memprediksi nilai dari sebuah variabel berdasarkan variabel lainnya.
3. Klastering: Mengelompokkan data ke dalam kelompok-kelompok yang serupa berdasarkan pola yang ada.
4. Asosiasi: Menemukan hubungan antara variabel dalam data.
5. Analisis Anomali: Mengidentifikasi pola atau kejadian yang tidak biasa atau tidak diharapkan dalam data.

1.5 Proses Data Mining

Proses Data Mining adalah serangkaian langkah yang sistematis yang dilakukan untuk mengekstraksi informasi yang bermanfaat dari data (Munthe & Juledi, 2021). Meskipun proses ini dapat bervariasi tergantung pada konteks dan tujuan spesifik, gambaran proses Data Mining ditunjukkan pada Gambar 1. Dimana umumnya terdiri dari langkah-langkah berikut:

1. Pemahaman Terhadap Masalah: Memahami tujuan bisnis dan masalah yang akan dipecahkan dengan menggunakan Data Mining. Proses ini melibatkan komunikasi dengan pemangku kepentingan dan mengidentifikasi pertanyaan atau hipotesis yang ingin dijawab dengan menggunakan analisis data.
2. Pemilihan Data: Setelah memahami masalah yang ada, langkah selanjutnya adalah memilih data yang relevan untuk dianalisis. Ini mungkin melibatkan pengumpulan data dari berbagai sumber atau menggunakan data yang sudah tersedia dalam basis data perusahaan.
3. Preprocessing Data: Data yang dikumpulkan seringkali tidak siap untuk analisis. Oleh karena itu, langkah ini mempersiapkan data awal yang akan digunakan untuk keseluruhan fase berikutnya. Pada tahap ini melibatkan pembersihan data untuk menghilangkan nilai yang hilang atau tidak valid, mengatasi masalah duplikasi atau kesalahan, dan melakukan transformasi data yang diperlukan, seperti normalisasi atau pengkodean variabel kategorikal dan mengintegrasikan data untuk persiapan analisis.



Gambar 1. Tahapan proses Data Mining (Widaningsih, 2022)

4. Pemodelan: Langkah ini melibatkan penerapan teknik Data Mining ke data yang telah diproses. Teknik yang digunakan tergantung pada tujuan analisis, tetapi bisa mencakup klasifikasi, regresi, klastering, asosiasi, atau analisis anomali. Proses ini sering melibatkan eksperimen dengan berbagai algoritma untuk menemukan model yang paling sesuai dengan data. Jika diperlukan, proses dapat kembali ke fase persiapan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik data mining tertentu.
5. Evaluasi: Setelah model dibangun, langkah selanjutnya adalah mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum digunakan atau disebarkan. Evaluasi dapat dilakukan dengan menggunakan k-fold cross validation dan melihat kinerja dari teknik terssebut. Untuk melihat kinerjanya dapat dilakukan dengan menggunakan metrik evaluasi yang sesuai dengan jenis analisis yang dilakukan, seperti akurasi untuk klasifikasi, Mean Squared Error (MSE) untuk regresi, atau

Silhouette Score untuk klastering. Mengevaluasi hasil model untuk memastikan keakuratannya dan relevansinya dengan tujuan bisnis.

6. Interpretasi Hasil: Langkah terakhir dalam proses Data Mining adalah menginterpretasikan hasil analisis untuk mendapatkan wawasan yang bermanfaat. Ini melibatkan memahami pola atau hubungan yang ditemukan dalam data, mengidentifikasi implikasi bisnisnya, dan membuat rekomendasi atau keputusan berdasarkan temuan tersebut.

Meskipun proses Data Mining dapat dilihat sebagai serangkaian langkah yang linear, dalam praktiknya seringkali bersifat iteratif dan berulang. Ini berarti bahwa langkah-langkah tersebut mungkin perlu diulang beberapa kali, dengan penyesuaian yang dibuat berdasarkan hasil evaluasi dan interpretasi, untuk mencapai hasil yang diinginkan.

1.6 Aplikasi Data Mining

Data Mining memiliki aplikasi yang luas di berbagai industri, termasuk:

1. Bisnis: Di dunia bisnis, Data Mining dapat digunakan untuk mengidentifikasi segmen pelanggan, menganalisis perilaku pelanggan, memprediksi preferensi konsumen, mengoptimalkan rantai pasokan, meningkatkan efisiensi operasional, dan mengembangkan strategi pemasaran yang ditargetkan. Contohnya, a) perusahaan ritel menggunakan Data Mining untuk membuat rekomendasi produk yang disesuaikan dengan preferensi pelanggan berdasarkan pola pembelian mereka. b) Data Mining dapat membantu perusahaan mengidentifikasi pelanggan bernilai tinggi dan mengembangkan program loyalitas untuk mempertahankan mereka.
2. Kedokteran: Dalam bidang kedokteran, pengumpulan data digunakan untuk menganalisis rekam medis dan

mengidentifikasi pola yang dapat memberikan hasil yang lebih baik bagi pasien, Data Mining dapat membantu dalam diagnosis penyakit, prediksi risiko kesehatan, identifikasi pola epidemiologi, dan pengembangan obat baru. Misalnya, Data Mining digunakan dalam analisis genomik untuk memahami faktor genetik yang berkontribusi terhadap penyakit tertentu dan merancang terapi yang disesuaikan secara individual.

3. Pertanian: Di sektor pertanian, Data Mining dapat digunakan dalam menganalisis data hasil panen untuk meningkatkan hasil panen, mengoptimalkan penggunaan sumber daya, memprediksi cuaca dan hasil panen, mengelola risiko pertanian, serta pengelolaan hama untuk mengoptimalkan praktek pertanian. Contoh penerapannya termasuk penggunaan sensor untuk memantau kondisi tanah dan tanaman, serta penggunaan analisis data untuk membuat keputusan tentang pengairan dan pemupukan.

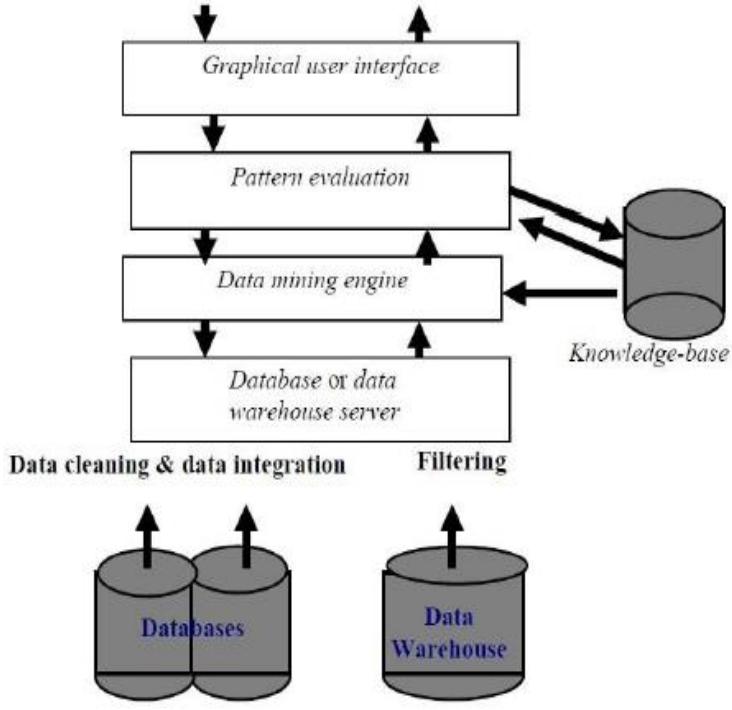
Data Mining melibatkan berbagai teknik, termasuk pohon keputusan, pengelompokan, dan jaringan saraf. Hal ini juga melibatkan penggunaan alat visualisasi data untuk menyajikan hasil dengan cara yang bermakna. Dimana aplikasi data mining beragam dan terus berkembang seiring dengan bertambahnya jumlah data yang dihasilkan

1.7 Arsitektur Data Mining

Arsitektur Data Mining adalah struktur atau kerangka kerja yang digunakan untuk mengorganisasi dan mengintegrasikan berbagai komponen yang terlibat dalam proses Data Mining. Arsitektur ini bertujuan untuk menyediakan kerangka kerja yang jelas dan terstruktur untuk menjalankan proses Data Mining dengan efisien. Arsitektur data mining ditunjukkan pada Gambar 2.

Berikut adalah beberapa komponen utama dalam arsitektur Data Mining:

1. Database: Komponen ini berfungsi sebagai tempat penyimpanan data yang akan digunakan dalam proses Data Mining. Database dapat berupa database relasional, database non-relasional, atau data warehouse. Dimana sumber data dapat berupa basis data perusahaan, data warehousing, data dari sensor atau perangkat Internet of Things (IoT), data dari media sosial, atau sumber data lainnya.



Gambar 2. Arsitektur Sistem Data Mining

2. Data Cleaning: Komponen ini berfungsi untuk membersihkan data yang akan digunakan dalam proses Data Mining. Data cleaning melibatkan penghapusan data yang tidak lengkap, tidak akurat, atau tidak relevan.

3. Data Integration: Komponen ini berfungsi untuk menggabungkan data dari berbagai sumber menjadi satu. Data integration melibatkan penggabungan data yang berbeda-beda dalam struktur dan format.
4. Data Mining Engine: Komponen ini berfungsi sebagai mesin yang melakukan proses Data Mining. Data Mining Engine melibatkan penggunaan algoritma yang sesuai untuk menemukan pola dan informasi yang berguna dari data.
5. Pattern Evaluation: Komponen ini berfungsi untuk mengevaluasi hasil dari proses Data Mining. Pattern Evaluation melibatkan pengujian hasil Data Mining untuk memastikan apakah hasil tersebut sesuai dengan tujuan yang diinginkan.
6. Graphical User Interface (GUI): Komponen ini berfungsi sebagai antarmuka pengguna yang memungkinkan pengguna untuk mengakses dan mengontrol proses Data Mining. GUI melibatkan penggunaan antarmuka yang mudah digunakan dan intuitif

Arsitektur Data Mining terdiri dari beberapa komponen yang berfungsi untuk mengumpulkan, mengolah, dan menganalisis data. Komponen-komponen ini bekerja sama untuk mencapai tujuan Data Mining, yaitu menemukan pola dan informasi yang berguna dari data. Dimana arsitektur Data Mining ini membantu dalam mengatur dan mengintegrasikan berbagai komponen yang terlibat dalam proses Data Mining, mulai dari pengumpulan data hingga interpretasi hasil. Dengan menggunakan arsitektur yang baik, organisasi dapat menjalankan proses Data Mining dengan lebih efisien dan efektif, serta memaksimalkan nilai yang dapat diperoleh dari analisis data.

Bab 2. Processing Data (Manipulasi dan Visualisasi)

2.1 Pendahuluan

Pemrosesan data adalah serangkaian langkah yang memanipulasi, mengubah, mengatur, dan menganalisis data mentah untuk mengekstrak informasi yang bermakna yang melibatkan informasi unik dan metode untuk mengubah data menjadi informasi yang berguna dan berharga untuk membuat keputusan yang lebih baik (Jannah, 2024). Pemrosesan data dapat mencakup berbagai kegiatan, mulai dari tugas dasar seperti entri data dan validasi hingga langkah yang lebih maju seperti analisis dan pemodelan data (Admin, 2023).

Siklus pemrosesan data adalah tahapan yang dilalui data mulai dari pengumpulan awal hingga penggunaan dan pembuangan akhir (Ranti, 2023). Siklus ini terdiri dari berbagai tahap pemrosesan data untuk mengubah data yang tidak terorganisir menjadi informasi yang bermakna. Berbagai langkah yang terlibat dalam pemrosesan data adalah seperti: Pengumpulan data, Entri data, Pemrosesan data, Menyimpan data, Analisis data, Visualisasi data, Interpretasi data, Pengambilan Keputusan, Pelaporan, Umpan balik.

1. Pengumpulan data
   Pengumpulan data adalah tahap awal, di mana data yang tidak terstruktur berasal dari berbagai sumber seperti transaksi, survei, interaksi media sosial, dan banyak sumber lainnya. Data yang dikumpulkan harus memiliki akurasi dan kualitas yang dapat menyebar ke seluruh siklus pemrosesan.
2. Entri data
   Setelah data terkumpul, informasi dimasukkan ke dalam sistem untuk diproses lebih lanjut yang dapat dimasukkan

secara manual atau metode alternatif lainnya, yang tentunya bergantung pada sumber dan volume data.

3. Pemrosesan data

   Data mentah dibersihkan dan diubah menjadi informasi yang berguna, yang meliputi identifikasi dan perbaikan kesalahan, data duplikat, dan nilai yang hilang. Transformasi data meliputi pengubahan data ke dalam format terstruktur, penggabungan atau peringkasan data, dan penghitungan untuk mempersiapkan evaluasi lebih lanjut.

4. Menyimpan data

   Data yang diproses disimpan dalam sistem, gudang data atau sistem penyimpanan lainnya, untuk memastikan pengambilan dan integritas yang mudah. Pilihan teknologi dan arsitektur penyimpanan tergantung pada elemen-elemen seperti volume, frekuensi akses, dan persyaratan khusus.

5. Analisis data

   Analisis data mencakup berbagai teknologi statistik atau komputerisasi untuk mengeksplorasi data dan mengungkap pola dan tren. Analisis data dapat mencakup rangkuman data dan perkiraan analisis bisnis di masa depan.

6. Visualisasi data

   Setelah analisis, wawasan data sering kali disajikan secara visual menggunakan bagan, grafik, dan dasbor, yang membuat data yang rumit menjadi lebih mudah dipahami dan memungkinkan para pemangku kepentingan untuk dengan mudah memahami hal-hal penting.

7. Interpretasi data

   Pakar data atau manajer menginterpretasikan data yang dianalisis untuk mendapatkan kesimpulan dan wawasan yang berarti dan mencapai tujuan yang memandu pengambilan keputusan yang tepat.

8. Pengambilan keputusan

   Dengan menggunakan pengetahuan yang diperoleh dari interpretasi data, para pemangku kepentingan membuat keputusan yang tepat yang membantu dalam operasi organisasi, memproses dan membuat strategi yang tepat yang berkisar dari penyesuaian taktis hingga inisiatif strategis.
9. Pelaporan

   Hasil yang diperoleh dari interpretasi data, para pemangku kepentingan membuat keputusan berharga yang didokumentasikan dan dikomunikasikan melalui presentasi, laporan, atau dasbor interaktif.
10. Umpan balik

    Organisasi menggunakan hasil keputusan dan tindakan yang diambil berdasarkan data yang diproses untuk menilai keefektifannya, yang membantu menyempurnakan strategi, mengoptimalkan proses, dan meningkatkan tindakan lebih lanjut.

2.2 Manipulasi Data

Data Manipulation atau manipulasi data adalah proses memanipulasi atau mengubah informasi agar lebih terorganisir dan mudah dibaca (Manullang, 2020). Proses *Data Manipulation* membutuhkan DML atau Data Manipulation Language. DML adalah bahasa pemrograman yang mampu menambah, menghapus, dan mengubah *database* (Kristanto, 1994). Artinya, dengan DML, kamu bisa mengubah informasi atau data mentah menjadi sesuatu yang bisa kita baca.

Proses manipulasi data menggunakan DML adalah tahap yang sangat penting di bidang data. Data mentah yang baru saja dikumpulkan tidak akan bisa dibaca ataupun dipahami dengan mudah. Dengan DML, data mentah tersebut bisa dibersihkan dan dipetakan sehingga ada kesimpulan atau informasi tertentu yang bisa diambil dari data yang sudah dikumpulkan.

Data Manipulation adalah salah satu langkah penting dalam pengolahan data dan dengan data, kamu bisa mengelola bisnismu dengan lebih baik. Tentunya peran proses manipulasi data tidak bisa dianggap ringan dalam keseluruhan pengolahan data. Dengan *Data Manipulation*, kamu bisa memprediksi tren, memahami perilaku pelanggan, meningkatkan produktivitas, mengurangi biaya, dan masih banyak lagi.

Ada beberapa manfaat lain dari *Data Manipulation*, yaitu:

1. Data yang Konsisten
   Data yang sudah melalui proses manipulasi data pastinya lebih konsisten dan rapi, sehingga bisa dibaca dan bisa disimpan secara konsisten.
2. Proyeksi Data
   Proyeksi data adalah suatu hal yang sangat penting bagi perusahaan atau bisnis. Dengan menganalisis data yang didapat dari proses manipulasi data perusahaan atau bisnis bisa mendapatkan proyeksi data yang bisa digunakan untuk mengambil langkah bisnis di kemudian hari.
3. Data yang Rapi dan Efisien
   Data Manipulation membantu kamu mendapatkan data terbaru yang tidak hanya akurat, tapi juga rapi. Manipulasi data membantumu merapikan data ketika kamu mengubah, memperbarui, menghapus, atau memasukkan data ke dalam *database.* Tak hanya itu, proses manipulasi data juga membantumu mendapatkan data yang rapi ketika kamu mendapatkan data yang tidak lengkap atau ganda.

Ada beragam bentuk manipulasi data. Masing-masing memiliki tujuan dan fungsi masing-masing. Berikut adalah beberapa contoh proses *Data Manipulation*:

a. Menyusun data menurut abjad atau tanggal.
b. Mengelola *log* server dan melihat lalu lintas website.
c. Membuat prakiraan tren pasar saham.

d. Menilai biaya produk, pola penetapan harga, atau kewajiban pajak di masa mendatang.
e. Melihat informasi *online*.

Langkah-langkah dalam Manipulasi data diantaranya:

1) Buat database dari data yang dimiliki. Manipulasi data hanya bisa dilakukan jika kamu memiliki data. Oleh karena itu, kamu harus membuat *database* yang dihasilkan dari sumber data. Dengan adanya *database,* kamu memiliki data yang bisa dimanipulasi.
2) Bersihkan, *restrukrisasi*, dan reorganisasi *database*. *Database* mentah biasanya mengandung data lain yang tidak lengkap, ganda, rusak, atau tidak rapi. Maka dari itu, ketiga proses ini dibutuhkan agar kamu tidak menghitung atau memasukkan data-data yang ganda atau rusak. Untuk menyempurnakan *database* yang kamu miliki, kamu bisa mengatur atau menyusun ulang isi *database* yang kamu miliki.
3) *Bangun database yang dapat dibaca*. Langkah ini bisa kamu lakukan dengan cara membangun *database* dari awal atau mengimpor *database* yang sudah ada sebelumnya.
4) Cek Kembali data yang dimiliki. Jika ada data yang kurang, kamu bisa menggabungkan data yang ada. Sebaliknya, jika kamu menemukan data ganda atau yang berlebihan, kamu bisa melakukan penghapusan.
5) Lakukan Analisi data. Lakukan analisis data untuk mendapatkan *insight* yang bisa membantu kamu mengambil keputusan atau langkah selanjutnya. Semua langkah-langkah tersebut perlu dilakukan agar kamu bisa memperoleh informasi yang mudah dibaca dan dipahami dari *database* yang kamu miliki. Apabila setelah kelima langkah tersebut informasi yang kamu dapatkan malah makin membingungkan, kemungkinan besar ada kesalahan dalam proses manipulasi data yang kamu lakukan.

2.3 Visualisasi Data

Visualisasi data adalah proses penyajian data dalam bentuk grafik yang membuat informasi mudah dimengerti, hal ini membantu menjelaskan tentang fakta dan menentukan arah tindakan (Pandensolang, 2022). Definisi visualisasi data menjelaskan tentang pentingnya data dengan menempatkan data dalam konteks visual. Hal ini melibatkan penciptaan dan studi representasi visual dari data yang dikenal sebagai informasi. Visualisasi data memungkinkan pengguna untuk memperoleh pengetahuan yang lebih banyak mengenai data mentah yang didapatkan dari berbagai sumber. Visualisasi dapat dilakukan dengan menggunakan dashboard, di mana teks, pola, dan korelasi yang tidak terdeteksi dapat dengan mudah divisualisasikan dengan menggunakan perangkat lunak visualisasi.

Visualisasi data tidak hanya mengubah data menjadi grafik visual, akan tetapi visualisasi data juga memerlukan perencanaan. Setiap jenis data memerlukan teknik visualisasi yang sesuai berdasarkan kebutuhannya. Berdasarkan tingkat kompleksitas data, untuk menghasilkan solusi yang berharga perlu melibatkan berbagai disiplin ilmu, seperti statistika, data mining, desain grafis, dan information visualization (Pandensolang, 2022).

1. Proses Visualisasi Data

Proses memahami data dimulai dari beberapa pertanyaan, tidak semerta-merta dijawab begitu saja, akan tetapi terdapat langkah-langkah dalam menjawab pertanyaan berdasarkan data. langkah-langkah tersebut adalah sebagai berikut:

a. *Acquire*. Tahap ini adalah proses pengumpulan data dari berbagai sumber, baik dari file penyimpanan atau sumber melalui jaringan. Tahap Acquire hanya berfokus pada bagaimana data didapatkan, jika produk akhir akan didistribusikan melalui internet maka, data yang ada harus memiliki struktur dan dapat disimpan pada sebuah server.

b. *Parse*. Tahap ini adalah proses penyesuian data kedalam sebuah format yang telah ditentukan yang kemudian akan dikategorikan kedalam beberapa kategori, agar data bapat dibaca dan bisa dibedakan satu dengan yang lainnya.
c. *Filter*. Pada tahap ini adalah proses seleksi data dengan menghapus data yang tidak diperlukan. Beberapa data yang terdapat pada berkas, mungkin perlu diterjemahkan ke dalam model matematika atau dilakukan normalisasi terlebih dahulu.
d. *Mine*. Pada tahap ini adalah proses penerapan metode disiplin ilmu statistika dan data mining sebagai jalan untuk mencari pola atau dijabarkan pada konteks matematis
e. *Represent*. Pada tahap ini adalah proses pengubahan data dalam bentuk visual seperti bar *graph*, *tree*, atau *tree*. Tahap Represent menunjukkan bentuk dasar data yang akan diambil. Tahap ini merupakan tahap yang sangat penting dalam visualisasi data. Pemilihan model visualisasi yang tepat akan mempengaruhi kualitas dari produk yang dihasilkan.
f. *Refine*. Pada tahap ini adalah proses meningkatkan hasil representasi agar terlihat lebih menarik. Graphic design lebih banyak terlibat pada tahap ini. Poin-poin yang cukup penting pada visual grafik dibandingkan denga poin lainnya diberikan pembeda agar data mudah dibaca.
g. *Interact*. Pada tahap ini adalah proses menambahkan metode untuk manipulasi data atau mengendalikan fitur yang terlihat dengan kata lain data bisa ditampilkan sesuai kehendak pengguna. Contoh interaksi antara pengguna dan data seperti zoom-in, zoom-out, merubah rentang data, melakukan filtering dan sebagainya.

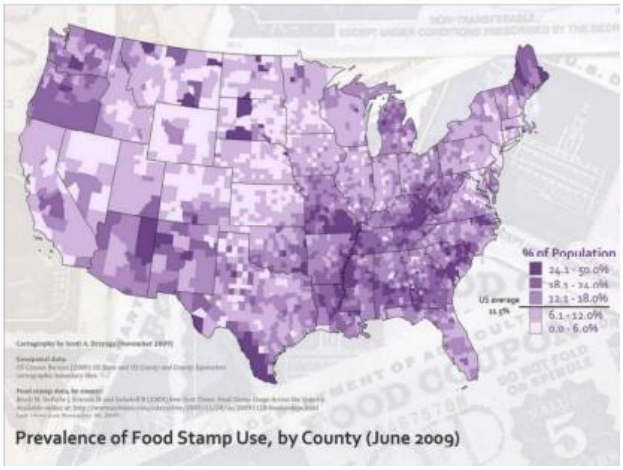
2. Tipe-tipe Visualisasi Data

Tujuan visualisasi adalah untuk membantu pemahaman manusia terhadap data dengan memaksimalkan sistem penglihatan manusia yang bisa membedakan pattren, spot the trends, dan identifikasi outlier (Rizki, 2020).

Tantangan dari visualisasi data adalah bagaimana membuat visualisasi yang effektif, menarik, dan tepat terhadap data yang dipakai. Terdapat tujuh hal yang harus dipenuhi dalam melakukan abstraksi tingkat tinggi (*high-level abstraction*), semakin banyak hal yang disembunyikan, semakin banyak juga langkah-langkah yang harus dipenuhi, langkah tersebut yaitu:

a. *Overview*: Melihat gambaran dari keseluruhan data.
b. *Zoom*: Memperbesar item yang terlihat menarik.
c. *Filter*: Melakukan penyaringan terhadap item yang dirasa kurang menarik.
d. *Details-on-demand*: Pilih satu item dari grup tertentu dan dapat melihat detail kapan saja.
e. *Relate*: Lihat relasi dari setiap item.
f. *History*: Dapat mengulang kembali atau kembali ke aksi sebelumnya.
g. *Extract*: Dapat melakukan ekstraksi dari parameter yang diberikan

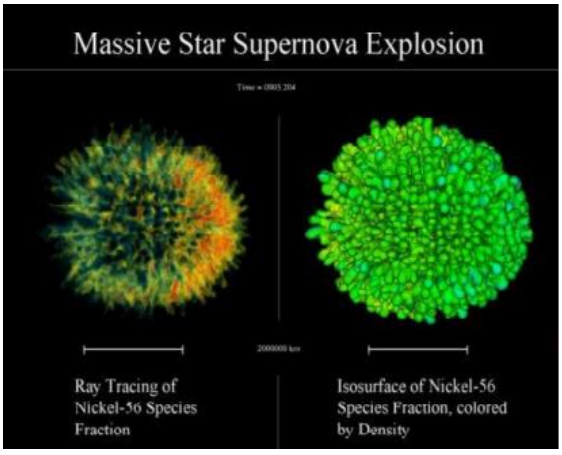
Berdasarkan taksonominya, grafik visual dibedakan menjadi beberapa bagian yaitu: 1D/Linear, 2D/Planar, 3D/Volumetric, Temporal, Multidimensional, Tree/ Hierarchical, dan Network. 1D/Linear. Grafik 1-dimensi termasuk di dalamnya adalah tipe data tekstual, kode sumberprogram, dan huruf alfabet. Setiap item yang digambarkan memiliki elemen garis. Contoh dari grafik 1D seperti kode-kode DNA, perbedaan kode sumber, urutan alfabat dan lain-lain. (tidak divisualisasikan secara umum). 2D/Planar. Grafik 2-dimensi termasuk di dalamnya peta geografis, denah rancangan, atau layout koran. Setiap item pada grafik 2-dimensi memiliki total area dan atribut (warna, ukuran, dll). Contoh grafik 2D/Planar dapat dilihat pada gambar 3.



Gambar 3. Contoh Grafik 2D/Planart

1) 3D/Volumetric

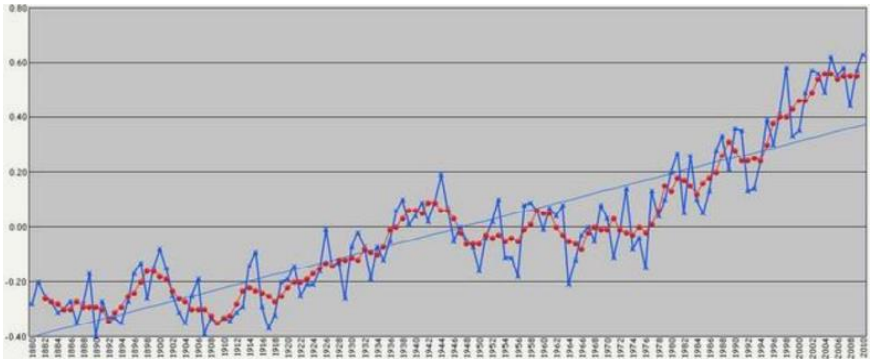
Grafik 3-dimensi adalah visual yang menggambarkan objek nyata, seperti tubuh manusia, bentuk bangunan, dll. Setiap item pada grafik 3-dimensi memiliki volume. Contoh grafik 3D/Volumetric dapat dilihat pada gambar 4.



Gambar 4. Contoh Grafik 3D/Volumetric

2) Temporal

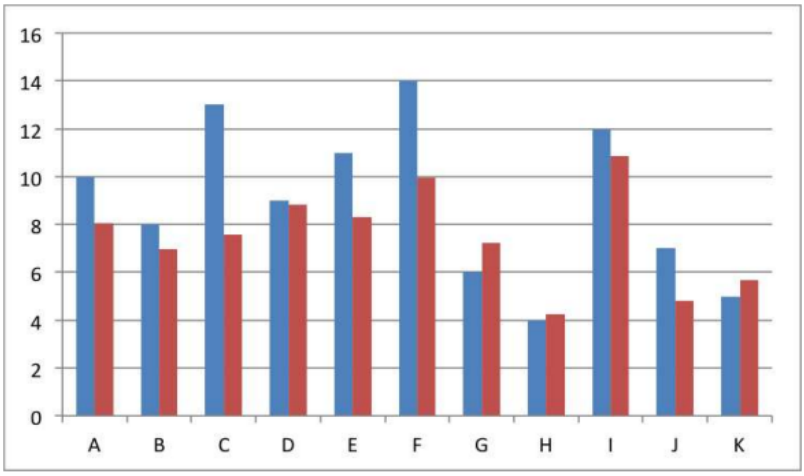
Grafik temporal adalah grafik yang berhubungan dengan waktu (time lines). Grafik ini menggambarkan persentasi historikal dari data 1-dimensi. Yang membedakan, grafik temporal memiliki item dengan waktu awal dan waktu akhir, atau periode tertentu. Contoh grafik temporal dapat dilihat pada gambar 5.



Gambar 5. Contoh Grafik Temporal

3) Grafik Multidimensional

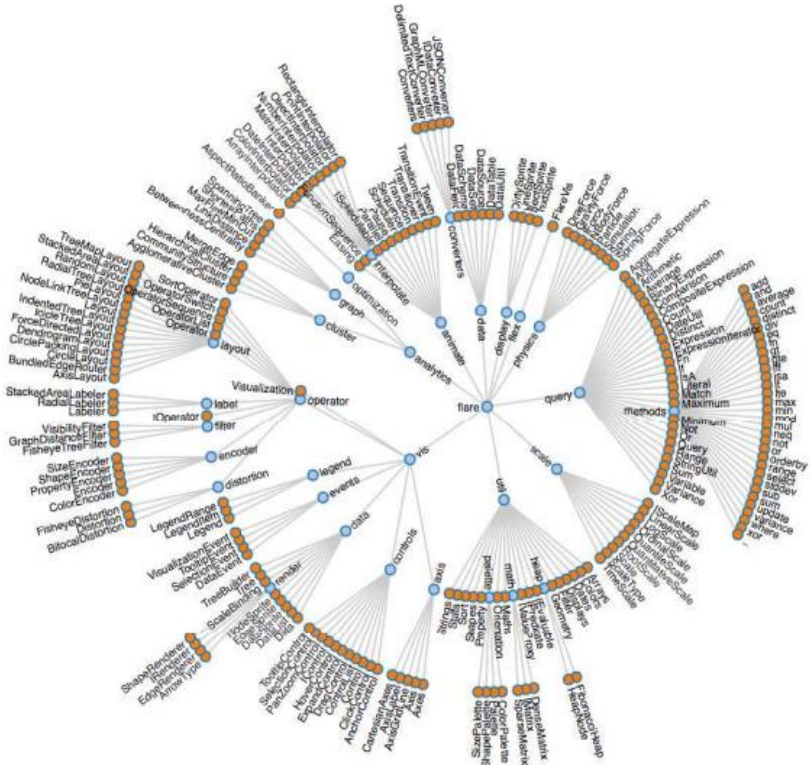
Grafik temporal didalamnya termasuk grafik-grafik yang dihasilkan dari manipulasi data dari disiplin ilmu statistika. Antarmuka representasi multidimensional adalah grafik 2-dimensi. Grafik multi-dimensi termasuk didalamnya grafik pie, histogram, tag cloud, bubble cloud, bar, tree-map, scatter plot, bubble chart, line chart, step chart, heat-map, parallel sets, spider chart, boxplot, mosaic display, waterfall, dan tabular. Contoh grafik multidimensional dapat dilihat pada gambar 6.



Gambar 6. Contoh Grafik Multidimensional

4) Tree/Hierarchial

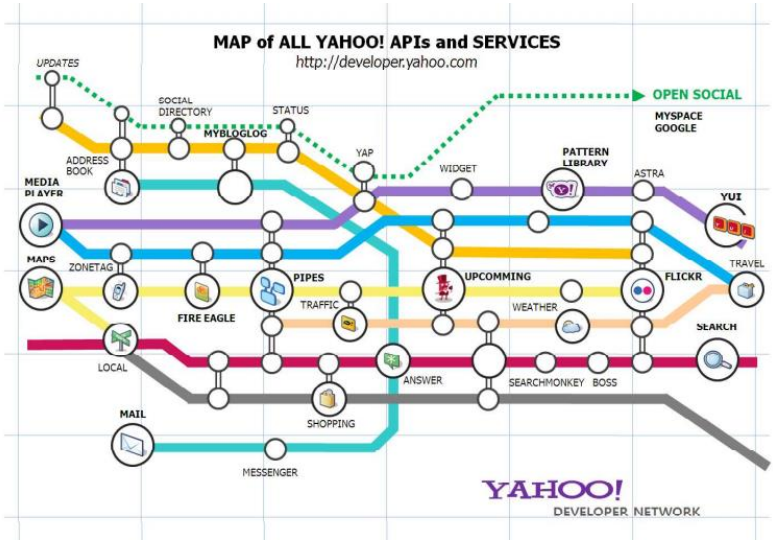
Tree adalah grafik herarkikal dari item-item yang memiliki hubungan satu dengan lainnya, atau yang memiliki induk (kecuali root). Setiap item antara induk dan anak bisa memiliki banyak atribut. Grafik tree termasuk didalamnya grafik tree, dendorogram, radial-tree, hyperbolic-tree, tree-map, dan sunburst. Contoh grafik Tree/Hierarchial dapat dilihat pada gambar 7.



Gambar 7. Contoh Grafik Tree/Hierarchial

5) Network

Network adalah grafik relasi yang tidak bisa digambarkan oleh grafik tree, network memvisualisasikan relasi-relasi item yang banyak (semena-mena). Grafik network termasuk di dalamnya grafik matrix, node-diagram, dependency, hive-plot, alluvial diagram, dan subway. Contoh grafik network dapat dilihat pada gambar 8.



Gambar 8. Contoh Grafik Network

Bab 3. Preprocessing Data (Normalisasi Data)

3.1 Pengertian, Pentingnya Preprocessing Data Mining

Preprocessing data juga mencakup normalisasi dan diskritisasi sebagai langkah penting dalam mempersiapkan data. Normalisasi merupakan proses penyesuaian skala nilai data ke dalam rentang tertentu untuk menghindari bias yang disebabkan oleh perbedaan skala. Prosedur-prosedur ini, sebagaimana diuraikan oleh Tan, Steinbach, dan Kumar, memungkinkan data menjadi lebih homogen dan mudah diklasifikasikan, membawa dampak signifikan terhadap kemudahan interpretasi dan akurasi dalam Data Mining (Tan et al., 2014).

Sejalan dengan apa yang diungkapkan oleh Tan, Steinbach, dan Kumar, dengan memastikan bahwa data bersih, relevan, dan konsisten sejak awal, maka risiko terhadap kesalahan interpretasi akakn lebih kecil. Proses ini memberi jalan bagi insight dan keputusan yang lebih informatif dan bertanggung jawab. Normalisasi membuktikan dirinya sebagai fondasi dasar yang mendukung keseluruhan struktur dan keberhasilan aplikasi Data Mining.

1. Definisi Normalisasi Data dan Alasan Mengapa Diperlukan

Normalisasi adalah proses dalam data preprocessing yang bertujuan untuk mengubah nilai atribut numerik dalam dataset ke dalam skala yang sama tanpa mengubah perbedaan dalam rentang nilai. Normalisasi sangat penting terutama jika algoritma mining yang digunakan melibatkan perhitungan jarak antar data, seperti pada teknik K-Means clustering atau algoritma berbasis jaringan saraf (Han et al., 2011).

Normalisasi data merujuk pada proses penyesuaian nilai data agar berada dalam skala yang sama. Langkah ini sering kali

melibatkan transformasi data mentah menjadi format yang lebih selaras agar bisa dianalisis dengan lebih akurat dan efisien. Normalisasi biasanya diperlukan saat dataset terdiri dari nilai dengan rentang yang berbeda, seperti skala pendapatan dan usia, sehingga model data tidak bias terhadap nilai yang lebih besar.

Pentingnya normalisasi data dalam data mining terletak pada kemampuannya untuk memaksimalkan kualitas hasil analisis. Ketika data tidak ternormalisasi, algoritma dapat memberikan bobot lebih kepada atribut dengan rentang nilai yang lebih besar. Sebagai contoh, dalam sebuah model yang mencakup atribut seperti berat badan dan tinggi badan, jika tidak ternormalisasi, atribut dengan satuan kilogram bisa mendominasi hasil analisis dibanding atribut dalam satuan sentimeter.

Normalisasi data juga membantu dalam meningkatkan keefektifan beberapa algoritma data mining. Algoritma seperti K-Nearest Neighbors (K-NN) dan jaringan saraf tiruan sering kali bekerja lebih baik ketika data mereka berada dalam rentang yang serupa. Hal tersebut dikarenakan algoritma ini sensitif terhadap jarak antar nilai data, membuat normalisasi menjadi langkah esensial untuk menghindari distorsi kalkulasi (Han et al., 2011). Selain itu, normalisasi data memungkinkan pengolahan data yang lebih efisien dalam jumlah besar. Proses normalisasi dapat meminimalisir ketidakseragaman dan meningkatkan kecepatan serta akurasi analisis data. Hasil yang lebih akurat ini menjadi dasar yang kuat untuk membuat keputusan bisnis yang lebih baik, berdasarkan temuan-temuan dari proses data mining.

Witten et al. (2016) menekankan bahwa normalisasi data juga berperan penting dalam mempercepat konvergensi algoritma pembelajaran mesin. Algoritma seperti jaringan saraf tiruan dan gradient descent sering kali memerlukan data yang dinormalisasi untuk mencapai hasil optimal dalam waktu yang lebih singkat. Normalisasi mengurangi variasi antar fitur,

sehingga memungkinkan algoritma untuk menemukan solusi optimal lebih cepat dan mengurangi risiko terjebak dalam solusi suboptimal. Selain itu, normalisasi dapat meningkatkan stabilitas numerik dalam perhitungan, mencegah masalah yang disebabkan oleh nilai ekstrem, dan memastikan bahwa algoritma bekerja secara konsisten pada berbagai jenis data (Witten, et al, 2016).

Secara keseluruhan, normalisasi data merupakan langkah penting yang tidak boleh diabaikan dalam setiap proses data mining. Secara umum, hal ini memastikan bahwa semua nilai data diperlakukan secara adil dan analisis bisa dilakukan dengan optimal tanpa dipengaruhi oleh perbedaan skala nilai antar atribut. Pada akhirnya, normalisasi berkontribusi besar dalam menyediakan hasil yang lebih jujur dan representatif dari analisis data yang dilakukan.

3.2 Prinsip Dasar Normalisasi Data

Dalam persiapan data untuk data mining, memahami berbagai teknik normalisasi sangat penting untuk mengoptimalkan kinerja model. Terdapat beberapa metode yang digunakan untuk normalisasi. Beberapa di antaranya adalah sebagai berikut:

1. Min-Max Scaling. Min-Max Scaling merupakan teknik normalisasi data yang mentransformasikan fitur dengan skala ulang range nilai menjadi skala tertentu, biasanya antara 0 dan 1. Metode ini efektif untuk menyederhanakan berbagai jenis data ke dalam sebuah format yang lebih umum, memudahkan perbandingan dan pemrosesan lebih lanjut.
2. Z-Score Normalization (Standardization). Z-Score Normalization, atau standardisasi, merupakan metode normalisasi lain yang sering dipakai. Teknik ini mengubah data sehingga memiliki rata-rata (mean) 0 dan standar deviasi 1. Dengan melakukan ini, data diubah ke dalam distribusi yang

memiliki karakteristik standar, memfasilitasi analisis dan pembelajaran mesin yang lebih akurat dengan data tersebut.

3. Decimal Scaling. Decimal Scaling adalah teknik normalisasi yang mengubah nilai data dengan membagi nilai oleh pangkat sepuluh. Tujuannya adalah untuk memindahkan titik desimal dari angka tersebut, sehingga nilai maksimumnya berada antara -1 dan 1, namun tetap mempertahankan proporsi relatif nilai-nilai data. Ini merupakan metode yang sederhana namun efektif, terutama ketika kecepatan komputasi dan kapasitas memori menjadi pertimbangan.
4. Ketiga metode normalisasi ini menawarkan pendekatan yang berbeda-beda namun komplementer untuk mengolah data, memastikan bahwa mereka yang bekerja dalam ruang data mining dapat memilih teknik yang paling sesuai dengan jenis data dan kebutuhan analitik yang dihadapi. Mendesign sistem dengan pemahaman yang baik tentang masing-masing teknik ini dapat membantu dalam memaksimalkan kinerja model analitik dan pembelajaran mesin.

Teknik normalisasi merupakan salah satu metode kunci dalam pengolahan data untuk keperluan data mining dan pembelajaran mesin. Esensi dari teknik ini adalah mengubah nilai data ke dalam skala yang seragam atau standar tanpa mengubah distribusi data asli. Sehingga, memungkinkan perbandingan atau pengolahan yang lebih efektif antara variabel-variabel dengan rentang nilai yang berbeda. Konsep matematika dasar di balik normalisasi meliputi berbagai macam formula, namun pada intinya, teknik ini berfokus pada penyederhanaan data agar lebih mudah diinterpretasikan oleh algoritma pembelajaran mesin.

Dalam praktiknya, ada beberapa metode dalam normalisasi, masing-masing dengan pendekatan matematikanya sendiri. Salah satu metode yang sering digunakan adalah Min-Max scaling, di mana nilai dari setiap elemen dalam dataset dikurangi dengan nilai minimum dan dibagi dengan rentang (maksimum dikurangi

minimum) dari dataset tersebut. Proses ini menghasilkan nilai-nilai baru yang berada dalam rentang tertentu, biasanya antara 0 dan 1, sehingga memudahkan pemrosesan data lebih lanjut tanpa kehilangan informasi esensial dari data asli.

Selain Min-Max scaling, Z-score normalisasi adalah metode lain yang sering diterapkan. Metode ini melibatkan pengurangan setiap nilai data dengan rata-rata (mean) dataset dan pembagian hasilnya dengan standar deviasi dari dataset tersebut. Hasil akhir dari proses ini adalah dataset dengan rata-rata sama dengan nol dan standar deviasi sama dengan satu. Pendekatan ini sangat berguna untuk data yang mengikuti distribusi normal, dan memungkinkan perbandingan antar variabel data yang memiliki satuan pengukuran atau skala yang berbeda.

Penerapan teknik normalisasi ini sangat penting dalam berbagai kasus penggunaan data mining dan pembelajaran mesin, karena memungkinkan model untuk bekerja dengan lebih efektif. Tanpa normalisasi, model pembelajaran mesin dapat menjadi bias atau kurang akurat karena dominasi variabel dengan skala yang lebih besar. Oleh karena itu, pengertian yang kuat tentang matematika dasar di balik teknik normalisasi adalah kunci untuk memaksimalkan efektivitas teknik data mining dan pembelajaran mesin.

3.3 Metode Normalisasi Data

1. Min-Max Scaling

Min-Max Scaling adalah sebuah teknik normalisasi skala yang cukup populer dalam pemrosesan data, terutama digunakan dalam machine learning dan data science untuk mengoptimalkan performa algoritma. Teknik ini bertujuan untuk mengubah skala nilai dalam dataset sehingga berada dalam rentang baru, umumnya antara 0 dan 1. Hal ini sangat berguna dalam kasus ketika data dalam variabel fitur memiliki skala yang sangat

berbeda. Min-Max Scaling membantu dalam mengurangi disparitas skala tanpa menghilangkan pola dalam data.

Implementasi Min-Max Scaling dapat dengan mudah dilakukan menggunakan berbagai bahasa pemrograman yang mendukung machine learning, termasuk Python. Dalam konteks Python, salah satu cara untuk melaksanakan operasi ini adalah dengan menggunakan pustaka *sklearn.preprocessing* yang menyediakan fungsi MinMaxScaler. Fungsi ini dapat disesuaikan untuk menentukan rentang skala baru yang diinginkan, dengan nilai default adalah 0 untuk minimum dan 1 untuk maksimum.

Sebagai contoh implementasi, misalkan kita memiliki dataset yang terdiri dari beberapa fitur, dan kita ingin menerapkan Min-Max Scaling ke salah satu fitur tersebut. Pertama-tama kita perlu mengimpor MinMaxScaler dari 'sklearn.preprocessing'. Setelah itu, kita dapat membuat instance dari MinMaxScaler dan menerapkannya ke fitur yang diinginkan dengan cara memasukkannya sebagai parameter ke fungsi 'fit_transform()' dari instance tersebut. Hasilnya adalah array fitur yang telah dinormalisasi, yang kemudian dapat digunakan dalam pembelajaran mesin untuk meningkatkan kinerja algoritma.

Teknik normalisasi seperti Min-Max Scaling sangat penting dalam preprocessing data, karena menyamakan skala fitur sehingga memungkinkan algoritma machine learning bekerja lebih efisien dan efektif (Han et al., 2011). Dengan mengaplikasikan teknik ini, dapat dihindari pemberian bobot yang tidak proporsional kepada fitur-fitur tertentu hanya karena perbedaan skala nilai.

Preprocessing dengan metode min max scalling terbilang merupakan cara yang paling mudah. Langkahnya diawali dengan menentukan nilai paling kecil dari sebuah data dan paling besar dari sebuah data. Formula untuk min max scalling adalah:

$$Xscaled = \frac{X - Xmin}{Xmax - Xmin}$$

X= nilai persatuan data

Contoh:

Data tekanan darah dari tiga orang dewasa: [110,90,120]

Menghitung Z-score :

Min = 90

Max= 120

$$Xscaled = \frac{110 - 90}{120 - 90} = \frac{20}{30} = 0.666$$

$$Xscaled = \frac{90 - 90}{120 - 90} = \frac{0}{30} = 0$$

$$Xscaled = \frac{120 - 90}{120 - 90} = \frac{30}{30} = 1$$

Maka hasil untuk normalisasi data tekanan darah adalah:
[0.666, 0, 1].

2. Z-Score Normalization (*Standardization*)

Normalisasi Z-score, juga dikenal sebagai standardisasi, merupakan metode untuk mengubah data sedemikian rupa sehingga memiliki rata-rata 0 dan deviasi standar 1. Pada metode ini penting sekali mengetahui perhitungan mean (rata-rata) dari data yang dimiliki (Bruce et al., n.d.). Menurut Aggarwal (2019), metode ini sangat efektif dalam menangani fitur-fitur yang memiliki unit pengukuran berbeda atau rentang nilai yang sangat bervariasi. Dengan menerapkan Z-Score, setiap nilai dalam dataset dikurangi dengan nilai rata-rata dan kemudian dibagi dengan standar deviasi. Proses ini menghasilkan data yang terdistribusi secara normal, memungkinkan algoritma pembelajaran mesin untuk bekerja lebih efisien, terutama dalam situasi di mana asumsi normalitas penting (Anggarwal, 2019).

Z-Score memiliki langkah-langkah seperti berikut:

a. Menghitung Mean. Perhitungan Mean untuk mendapatkan nilai rata-rata dari sebuah kumpulan data bisa dihitung dengan formula:

$$\mu = \frac{1}{N} \sum_{i-1}^{N} Xi$$

Dengan keterangan :
N= Jumlah data
$X_{i=}$ Nilai ke-i

Contoh:
Jika anda memiliki data berat tubuh dari 3 anak : [21,28,23]

$$\mu = \frac{21 + 28 + 23}{3} = 24$$

Menghitung standar deviasi, dan Standar deviasi merupakan perhitungan untuk mengetahui seberapa jauh penyebaran data dari mean. Formula standar deviasi bisa didapat menggunakan fungsi:

$$\sigma = \sqrt{\frac{1}{N} + \sum_{i-1}^{N} (Xi - \mu)^2}$$

Contoh:
Setelah mendapatkan nilai rata-rata (mean) hitung standar deviasi dari data diatas:

$$\sigma = \sqrt{\frac{(21 - 24)^2 + (28 - 24)^2 + (23 - 24)^2}{3}} = 8{,}6$$

b. Menghitung Z-Score. Setelah mengetahui standar deviasi, barulah kita dapat menghitung Z-score. Z-score dapat dihitung dengan formula:

$$Z = \frac{X - \mu}{\sigma}$$

Contoh:
Perhitungan Z-Score dilakukan untuk setiap data perjenis data. Berikut contoh dari perhitungan Z-score dari berat tubuh 3 orang anak sebagai berikut:
Z-score untuk data berat badan 21:

$$Z = \frac{21 - 24}{8.6} = -0.348$$

Z-score untuk data berat badan 28:

$$Z = \frac{28 - 24}{8.6} = -0.465$$

Z-score untuk data berat badan 23:

$$Z = \frac{23 - 24}{8.6} = -0.116$$

Setelah proses normalisasi menggunkan metode Z-score data akan terbaca seperti ini:
Berat tubuh:[-0.348, -0.465, -0.116]

c. *Decimal Scaling*
Diantara kedua teknik yang telah dibahas, metode yang terakhir dalam teknik normalisasi pada proses preprocessing adalah metode decimal scalling (Murphy, n.d.). Normalisasi Decimal Scaling adalah metode yang digunakan untuk menyesuaikan skala nilai-nilai dalam dataset dengan cara membagi setiap nilai dengan pangkat sepuluh dari bilangan bulat terbesar dalam dataset. Singh & Singh (2020) menjelaskan bahwa teknik ini sangat berguna untuk

memastikan bahwa semua nilai berada dalam rentang yang lebih kecil dan seragam, memudahkan analisis lebih lanjut. Dengan membagi nilai-nilai tersebut, decimal scaling mengurangi perbedaan skala yang ekstrem dan mencegah fitur-fitur dengan skala besar mendominasi analisis, yang bisa mengganggu hasil akhir. Teknik ini sangat berguna dalam situasi di mana data memiliki rentang nilai yang sangat bervariasi dan memerlukan keseragaman skala untuk analisis yang lebih akurat (Singh & Singh, 2020). Decimal Scaling adalah cara untuk mengubah ukuran data sehingga nilainya lebih kecil, tetapi tetap memiliki arti yang sama. Ini berguna ketika kita ingin membandingkan data yang memiliki skala yang sangat berbeda, seperti mengukur berat badan dan harga rumah. Dengan menggunakan Decimal Scaling, kita dapat mengurangi pengaruh nilai-nilai besar tanpa mengubah bentuk data secara keseluruhan. Langkah-langkah dalam menghitung decimal scalling secara manual bisa menggunakan cara ini:

- Tentukan nilai j. j merupakan bilangan bulat terkecil yang membuat nilai absolut terbesar dalam data menjadi nilai yang kurang dari 1. Perhitungan j menggunakan formula sebagai berikut:

$$j = \lceil \log_{10}(\max(|X|)) \rceil$$

Keterangan :
Max ($|X|$) = nilai terbesar dalam data

Yang perlu diketahui dan merupakan hal penting adalah symbol ⌈⌉ merupakan fung ceil yang berarti nilai akan dibulatkan ke atas ke bilangan bulat terdekat.
Hitung normalisasi setiap data. Nilai dalam dataset dibagi dengan 10^j.
Contoh:

Data 3 orang gaji karyawan :[6500000, 7000000, 4800000]
Hitung normalisasi:
Diketahui :
Max (|X|) = 7000000

$$j = \lceil \log_{10}(7000000) \rceil = 7$$

Mencari nilai j bisa dilakukan dengan cara $\frac{X_i}{10^j} \leq 1$
Contoh:

$$j = \frac{7000000}{10^j} \leq 1$$

Bisa dilihat, 7000000 bisa dibagi 10 pangkat berapa agar hasilnya kurang dari sama dengan 1?
Jawabannya adalah 10^7. Maka jawaban dari j=7

$$j = \frac{7000000}{10000000} = 0{,}7$$

$$j = 0{,}7 \leq 1 = true$$

Selanjutnya lakukan perhitungan normalisasi.

$$Normalisasi = \frac{x}{10^j}$$

Maka untuk ketiga data diatas hasilnya adalah:

$$X^1 = \frac{6500000}{10^7} = 0.65$$

$$X^2 = \frac{7000000}{10^7} = 0.7$$

$$X^3 = \frac{6800000}{10^7} = 0.68$$

Maka data gaji hasil normalisasi adalah [0.65, 0.7, 0.68]. Lalu bagaimana memilih preprocessing yang tepat untuk permasalahan penelitian kita?

d. Memilih Teknik Normalisasi
Faktor-Faktor yang Mempengaruhi Pemilihan Metode Normalisasi. Pemilihan metode normalisasi yang tepat

merupakan langkah krusial dalam preprocessing data, karena dapat mempengaruhi kinerja algoritma data mining secara signifikan. Kotsiantis et al. menyoroti bahwa karakteristik dataset, seperti rentang nilai dan distribusi data, adalah faktor utama yang mempengaruhi keputusan ini (Kotsiantis et al., 2006). Misalnya, jika dataset memiliki rentang nilai yang sangat bervariasi, metode seperti normalisasi Min-Max dapat membantu menyamakan skala fitur secara proporsional. Di sisi lain, jika data memiliki distribusi yang tidak normal, metode seperti Z-Score dapat membantu dengan mengubah data sehingga memiliki distribusi normal, yang seringkali lebih cocok untuk banyak algoritma pembelajaran mesin. Selain karakteristik dataset, tujuan analisis juga memainkan peran penting dalam memilih metode normalisasi. García et al. menambahkan bahwa tujuan analisis juga memainkan peran penting dalam pemilihan metode normalisasi. Untuk algoritma yang sensitif terhadap skala data, seperti K-Means atau K-Nearest Neighbors, normalisasi Min-Max lebih disarankan García et al. (2014). Sebaliknya, untuk algoritma yang memerlukan data berdistribusi normal, seperti regresi logistik atau analisis komponen utama (PCA), standarisasi Z-Score lebih tepat digunakan. Selain itu, ketersediaan alat-alat preprocessing yang mendukung metode tertentu dan kebutuhan spesifik analisis juga harus dipertimbangkan dalam memilih metode normalisasi yang paling sesuai. Pemilihan metode normalisasi harus mempertimbangkan juga potensi dampak terhadap hasil akhir analisis. Kotsiantis et al. (2006) menyarankan agar peneliti melakukan eksperimen dengan beberapa metode normalisasi dan mengevaluasi kinerja masing-masing metode terhadap algoritma yang digunakan. Melalui pendekatan eksperimental ini, peneliti dapat memilih metode yang memberikan hasil paling konsisten dan akurat sesuai dengan karakteristik dan tujuan data. Kesimpulannya,

memilih metode normalisasi yang tepat memerlukan pemahaman mendalam tentang data, tujuan analisis, serta alat dan teknik yang tersedia, untuk memastikan hasil analisis yang optimal dan andal.

3.4 Studi Kasus Penggunaan Metode Normalisasi

Studi Kasus 1: Penggunaan Normalisasi Min-Max pada Data Sensor IoT. Sebuah perusahaan teknologi mengumpulkan data dari berbagai sensor IoT yang dipasang di berbagai lokasi untuk memantau kondisi lingkungan. Data yang dikumpulkan termasuk suhu, kelembaban, dan tekanan udara, yang memiliki rentang nilai yang sangat berbeda. Misalnya, suhu berkisar antara -10°C hingga 50°C, sementara kelembaban berada dalam persentase (0% hingga 100%) dan tekanan udara dalam satuan Pascal (Pa).
Metode Normalisasi yang Digunakan: Normalisasi Min-Max. Alasan Pemilihan: Normalisasi Min-Max dipilih karena tujuan utama adalah untuk menyamakan skala semua fitur sehingga masing-masing fitur memiliki skala yang sama, yaitu antara 0 dan 1. Dengan demikian, algoritma machine learning seperti K-Nearest Neighbors (K-NN) atau K-Means Clustering dapat bekerja lebih efektif karena perbedaan skala yang besar antar fitur dapat menyebabkan bias dalam perhitungan jarak. Hasil: Setelah normalisasi, data sensor menjadi lebih seragam dalam hal skala, yang mengarah pada peningkatan akurasi dalam deteksi anomali dan clustering. Algoritma K-Means Clustering dapat mengelompokkan data sensor dengan lebih akurat, mengidentifikasi kondisi lingkungan yang serupa dengan lebih efisien.

Studi Kasus 2: Penggunaan Standarisasi Z-Score pada Data Keuangan. Sebuah lembaga keuangan melakukan analisis data keuangan untuk memprediksi default kredit. Data yang digunakan mencakup pendapatan tahunan, jumlah hutang, dan jumlah pembayaran bulanan. Nilai-nilai ini memiliki distribusi

yang tidak normal dan rentang yang sangat bervariasi, dengan beberapa atribut memiliki nilai ekstrem yang dapat mempengaruhi hasil analisis. Metode Normalisasi yang Digunakan: Standarisasi Z-Score. Alasan Pemilihan: Z-Score dipilih karena metode ini menyesuaikan data sehingga memiliki rata-rata nol dan standar deviasi satu. Hal ini membantu dalam menangani data yang memiliki distribusi tidak normal dan outliers. Standarisasi memastikan bahwa semua fitur berkontribusi secara proporsional dalam model, terutama untuk algoritma yang sensitif terhadap distribusi data seperti regresi logistik dan analisis komponen utama (PCA). Hasil: Dengan menggunakan standarisasi Z-Score, model prediksi default kredit menjadi lebih stabil dan akurat. Algoritma regresi logistik menunjukkan peningkatan dalam kinerja prediksi, dengan akurasi yang lebih tinggi dan pengurangan overfitting. PCA juga memberikan hasil yang lebih baik dalam mengurangi dimensi data tanpa kehilangan informasi penting.

Studi Kasus 3: Penggunaan Decimal Scaling pada Data Pengolahan Citra. Sebuah proyek penelitian di bidang pengolahan citra bertujuan untuk mengenali objek dalam gambar. Dataset yang digunakan terdiri dari piksel gambar dengan nilai intensitas yang bervariasi dari 0 hingga 255. Karena rentang nilai yang lebar ini, algoritma machine learning mengalami kesulitan dalam konvergensi dan akurasi. Metode Normalisasi yang Digunakan: Decimal Scaling. Alasan Pemilihan: Decimal Scaling dipilih karena metode ini mengubah skala data dengan membagi setiap nilai dengan pangkat sepuluh dari bilangan bulat terbesar dalam dataset. Ini membantu dalam menangani nilai piksel yang besar dan mengkonversinya ke skala yang lebih kecil dan seragam, misalnya dari 0 hingga 1, tanpa kehilangan informasi penting. Hasil: Setelah normalisasi menggunakan Decimal Scaling, performa algoritma seperti Convolutional Neural Networks (CNN) dalam mengenali objek dalam gambar meningkat secara

signifikan. Algoritma dapat belajar lebih cepat dan dengan akurasi yang lebih tinggi, karena nilai-nilai piksel yang telah dinormalisasi membuat proses pelatihan lebih efisien.

Studi Kasus 4: Penggunaan Normalisasi Log pada Data Penjualan E-commerce. Sebuah perusahaan e-commerce menganalisis data penjualan yang mencakup transaksi harian, jumlah pelanggan, dan pendapatan harian. Data ini menunjukkan skewness yang tinggi dengan beberapa nilai outlier yang sangat besar. Metode Normalisasi yang Digunakan: Normalisasi Log. Alasan Pemilihan: Normalisasi log dipilih karena metode ini efektif dalam mengurangi skewness data dengan mengaplikasikan transformasi logaritmik. Ini membantu dalam menangani distribusi data yang tidak normal dan mengurangi dampak outliers, sehingga fitur-fitur yang dihasilkan lebih seimbang untuk analisis selanjutnya. Hasil: Setelah penerapan normalisasi log, distribusi data menjadi lebih normal, dan outliers lebih terkontrol. Algoritma regresi linier dan analisis time-series menunjukkan peningkatan dalam performa prediksi penjualan dan analisis tren, dengan hasil yang lebih akurat dan stabil.

Studi Kasus 5: Penggunaan Normalisasi Robust Scaler pada Data Medis. Sebuah studi kesehatan menggunakan data medis yang mencakup berbagai pengukuran klinis seperti tekanan darah, kadar gula darah, dan indeks massa tubuh (BMI). Data ini seringkali memiliki outliers yang dapat mempengaruhi analisis. Metode Normalisasi yang Digunakan: Normalisasi Robust Scaler. Alasan Pemilihan: Robust Scaler dipilih karena metode ini menggunakan median dan interquartile range (IQR) untuk menskalakan fitur, yang membuatnya lebih tahan terhadap outliers. Ini memastikan bahwa normalisasi tidak terpengaruh oleh nilai ekstrem yang sering muncul dalam data medis. Hasil: Dengan normalisasi menggunakan Robust Scaler, analisis data medis menjadi lebih reliabel. Algoritma seperti Random Forest dan Support Vector Machine (SVM) menunjukkan peningkatan

akurasi dalam klasifikasi kondisi medis, karena data yang lebih stabil dan representatif tanpa terpengaruh oleh outliers.

Studi kasus ini menunjukkan bagaimana pemilihan metode normalisasi yang tepat dapat meningkatkan kinerja algoritma machine learning sesuai dengan karakteristik dan kebutuhan data yang dianalisis.

3.5 Tantangan dan Solusi dalam Normalisasi Data

1. Masalah Skala Data Berbeda

Penanganan masalah skala data yang berbeda adalah tantangan utama dalam normalisasi data. Aggarwal menggarisbawahi bahwa ketika fitur-fitur dalam dataset memiliki rentang nilai yang sangat bervariasi, algoritma pembelajaran mesin dapat menjadi bias terhadap fitur-fitur dengan skala yang lebih besar. Hal ini dapat menyebabkan interpretasi yang salah dan kinerja yang buruk dari model. Oleh karena itu, normalisasi menjadi kritis untuk menyamakan skala data sehingga setiap fitur memiliki pengaruh yang seimbang dalam analisis, Aggarwal (2015).

Batista & Monard menunjukkan bahwa untuk menangani masalah skala data yang berbeda, metode normalisasi seperti Min-Max Scaling atau Z-Score Standarization dapat diterapkan (Batista & Monard, 2003). Min-Max Scaling mengubah skala nilai setiap fitur ke dalam rentang tertentu, sering kali antara 0 dan 1, sementara Z-Score Standarization mentransformasi nilai-nilai menjadi z-score, dengan rata-rata nol dan standar deviasi satu. Kedua metode ini efektif dalam menyeimbangkan pengaruh fitur-fitur dalam analisis dan meningkatkan konsistensi hasil.

2. Data Skewed dan Outliers

Pendekatan untuk menangani data skewed dan outliers dalam proses normalisasi merupakan aspek penting dalam preprocessing data. Aggarwal (2015) menggambarkan bahwa data skewed, atau data yang tidak terdistribusi normal, dapat

memengaruhi kinerja algoritma pembelajaran mesin dengan cara yang tidak diinginkan. Selain itu, keberadaan outliers, atau titik data yang jauh dari mayoritas data lainnya, dapat menyebabkan bias dan kesalahan dalam analisis.

Untuk mengatasi tantangan ini, Batista & Monard (2003) merekomendasikan pendekatan yang berbeda dalam normalisasi data. Salah satu pendekatan yang umum digunakan adalah mengaplikasikan teknik robust scaler atau normalisasi logaritmik. Teknik ini bertujuan untuk mengurangi dampak outliers dan mengubah distribusi data menjadi lebih normal. Dengan menerapkan pendekatan ini, data menjadi lebih stabil dan konsisten, yang pada gilirannya meningkatkan kinerja algoritma pembelajaran mesin dan analisis data secara keseluruhan.

3.6 Kesimpulan

Pentingnya Normalisasi: Seperti yang dijelaskan oleh Han, Pei, dan Kamber (2011) serta Witten et al. (2016), normalisasi data adalah proses penting yang membantu mengurangi bias dan meningkatkan kinerja model dalam data mining. Normalisasi menyamakan skala fitur, sehingga memungkinkan algoritma pembelajaran mesin untuk beroperasi lebih efektif.

Metode Normalisasi Populer: Beberapa metode normalisasi yang umum digunakan termasuk Min-Max scaling, Z-score (standardisasi), dan normalisasi desimal. Aggarwal (2015) dan García et al. (2015) menyatakan bahwa pemilihan metode tergantung pada jenis data dan distribusi asli data, serta sensitivitas algoritma terhadap skala data.

Dampak pada Algoritma Pembelajaran Mesin: Singh dan Singh (2010) menunjukkan bahwa normalisasi dapat secara signifikan mempengaruhi performa klasifikasi. Algoritma seperti k-Nearest Neighbors (kNN) dan clustering sangat bergantung

pada normalisasi karena mereka sensitif terhadap jarak antar instance dalam data.

Penanganan Data Skewed dan Outliers: Batista dan Monard (2002) menekankan pentingnya menggunakan metode robust terhadap outliers saat melakukan normalisasi, seperti menggunakan teknik transformasi logaritmik atau robust scaler untuk mengurangi pengaruh nilai ekstrem.

Rekomendasi untuk Praktek: Kotsiantis et al. (2006) serta Müller dan Guido (2016) menyarankan bahwa selain memilih metode normalisasi yang sesuai, penting untuk secara konsisten menerapkannya di seluruh dataset untuk memastikan bahwa transformasi adalah tepat dan efektif. Keberlanjutan dalam aplikasi normalisasi penting untuk integritas model analisis data. Kompleksitas dan Tantangan: Rahm dan Do (2000) mengakui bahwa meskipun normalisasi adalah langkah penting, proses ini bisa menjadi kompleks dan memerlukan pertimbangan cermat mengenai karakteristik data yang spesifik. Memahami distribusi data dan bagaimana metode tertentu mempengaruhi analisis merupakan kunci dalam mengoptimalkan proses normalisasi.

Dengan mengikuti panduan dan prinsip tersebut, proses normalisasi dapat dilakukan dengan lebih efektif, menghasilkan dataset yang lebih homogen yang mendukung analisis data yang lebih akurat dan pengambilan keputusan yang lebih informasi.

Preprocessing data merupakan langkah fundamental dalam data mining yang tidak boleh diabaikan, karena memiliki dampak signifikan terhadap efektivitas dan efisiensi algoritma yang digunakan. Proses ini mencakup berbagai teknik seperti pembersihan data, transformasi, normalisasi, dan pengurangan dimensi yang bertujuan untuk mengoptimalkan dataset sehingga siap untuk analisis lebih lanjut. Kualitas data yang dimasukkan ke dalam model pembelajaran mesin atau algoritma analitik secara langsung menentukan kualitas output dan insight yang dihasilkan. Preprocessing yang efektif membantu dalam

mengeliminasi noise dan ketidaksesuaian dalam data, mengurangi kerumitan komputasi, dan meningkatkan akurasi serta keandalan hasil prediktif atau analitis. Oleh karena itu, menghabiskan waktu dan sumber daya untuk preprocessing yang matang adalah investasi yang penting untuk mendapatkan keuntungan maksimal dari proses data mining.

Bab 4. Klasisifikasi dengan K-Nearest Neighbors

4.1 Klasifikasi

Klasifikasi merupakan proses penemuan model (fungsi) yang membedakan dan menggambarkan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui (Leidiana, 2013). Klasifikasi merupakan bagian dari data mining, dimana Data mining merupakan suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam database. Data mining merupakan proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstrasi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar (Mardi, 2017). Klasifikasi memiliki peran krusial dalam membantu kita memahami dan mengorganisir dunia di sekitar kita. Algoritma klasifikasi memungkinkan kita untuk mengelompokkan data berdasarkan karakteristik yang serupa, sehingga dapat kita manfaatkan untuk:

1. Membuat prediksi: Mengidentifikasi kemungkinan kelas atau kategori suatu data baru.
2. Memahami pola: Mengungkap tren dan hubungan tersembunyi dalam dataset.
3. Meningkatkan pengambilan keputusan: Mendukung keputusan yang lebih tepat dengan informasi yang lebih terstruktur.

4.2 Jenis-jenis Metode Klasifikasi

Berbagai metode klasifikasi tersedia untuk menyelesaikan berbagai tugas, seperti:

1. Klasifikasi Biner: Memisahkan data menjadi dua kelas (misalnya, spam/bukan spam, positif/negatif).
2. Klasifikasi Multi-Kelas: Mengelompokkan data menjadi lebih dari dua kelas (misalnya, jenis bunga, kategori produk).
3. Klasifikasi Multi-Label: Menugaskan beberapa label ke setiap titik data (misalnya, topik dokumen, genre musik).
4. Contoh Penerapan Klasifikasi:
5. Klasifikasi Email: Menyaring email spam dari email penting.
6. Klasifikasi Gambar: Mengidentifikasi objek dalam gambar, seperti orang, hewan, atau pemandangan.
7. Klasifikasi Pelanggan: mengelompokkan pelanggan berdasarkan kebiasaan berbelanjanya.
8. Klasifikasi Penipuan: Mendeteksi aktivitas penipuan dalam transaksi keuangan.
9. Klasifikasi Medis: Mendiagnosis penyakit berdasarkan gejala dan hasil tes.

Berikut ini beberapa Algoritma Klasifikasi popular diantaranya:

a. K-Nearest Neighbors (KNN): Algoritma sederhana dan mudah diimplementasikan yang mengklasifikasikan data baru berdasarkan tetangga terdekatnya dalam dataset pelatihan.
b. Naïve Bayes: Algoritma probabilistik yang mengklasifikasikan data berdasarkan probabilitas kemunculan fitur-fiturnya di setiap kelas.
c. Support Vector Machines (SVM): Algoritma yang mencari hyperplane untuk memisahkan data dengan margin maksimum.
d. Decision Trees: Algoritma yang membangun pohon keputusan untuk mengklasifikasikan data berdasarkan aturan yang mudah dipahami.
e. Random Forest: Algoritma ensemble yang menggabungkan banyak pohon keputusan untuk meningkatkan akurasi klasifikasi.

Pemilihan algoritma klasifikasi yang tepat tergantung pada beberapa faktor:

- Jenis tugas klasifikasi: Biner, multi-kelas, atau multi-label.
- Ukuran dan kompleksitas dataset: Algoritma yang lebih kompleks mungkin membutuhkan dataset yang lebih besar.
- Kecepatan dan efisiensi komputasi: Beberapa algoritma lebih cepat dan hemat komputasi daripada yang lain.
- Interpretasi hasil: Algoritma tertentu seperti pohon keputusan lebih mudah diinterpretasikan daripada yang lain.

4.3 Klasifikasi K-Nearest Neighbor

Algoritma kNN adalah salah satu algoritma klasifikasi paling terkenal yang yang digunakan untuk memprediksi kelas dari catatan atau (Sampel) dengan kelas yang tidak ditetukan berdasarkan kelas dari catatan tetangganya (Panjaitan et al., 2018). Metode kNN menentukan nilai jarak pada pengujian data testing dengan data training berdasarkan nilai terkecil dari nilai ketentanggaan terdekat (Alghifari & Wibowo, 2019).

Algoritma membuat prediksi dengan menghitung kemiripan antara sampel input dari setiap training. Algoritma ini tidak membuat asumsi yang kuat tentang bentuk fungsi pemetaan maka itu adalah Nonparametrik. Dengan kata sederhana, dengan tidak membuat asumsi, algoritma bebas untuk mempelajari bentuk fungsional apa pun dari data training. Cara Algoritma KNN bekerja, yaitu:

1. Memilih Nilai K: Nilai K mewakili jumlah "tetangga terdekat" yang akan dipertimbangkan dalam proses klasifikasi. Nilai K harus dipilih dengan tepat, karena dapat mempengaruhi performa algoritma.
2. Mengukur Jarak: Hitung jarak antara titik data baru dengan semua titik data dalam dataset pelatihan. Jarak dapat dihitung menggunakan berbagai metrik, seperti jarak Euclidean atau Manhattan.

3. Menentukan Tetangga Terdekat: Identifikasi K tetangga terdekat dari titik data baru berdasarkan jarak yang dihitung pada langkah 2.
4. Menetapkan Kelas: Klasifikasikan titik data baru berdasarkan kelas mayoritas dari K tetangga terdekatnya. Untuk Perhitungan Jarak menggunakan Euclidean Distance. Jarak Euclidean adalah akar kuadrat dari jumlah jarak kuadrat antara dua titik. Ini juga dikenal sebagai norma L2. Persamaan Euclidean Distance ditunjukkan pada Gambar 1.

$$\text{Euclidean Distance between } (x_1, y_1) \text{ and} (x_2, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Gambar 1. Persamaan Euclidean Distance

Contoh Perhitungan K-Nearest Neighbors (KNN)
Berikut adalah contoh perhitungan KNN untuk mengklasifikasikan bunga berdasarkan warnanya:

Tabel 1. Dataset

Bunga	Warna (R, G, B)	Kelas
Mawar 1	(255, 100, 50)	Mawar
Mawar 2	(200, 50, 25)	Mawar
Melati 1	(255, 255, 255)	Melati
Melati 2	(200, 200, 200)	Melati
Anggrek 1	(100, 150, 255)	Anggrek
Anggrek 2	(50, 100, 200)	Anggrek

Titik data baru:
Bunga baru memiliki warna (150, 100, 150).
Nilai K:
Misalkan kita memilih nilai K = 3, yaitu akan mempertimbangkan 3 tetangga terdekat.

Langkah-langkah:
Hitung jarak: yaitu menghitung jarak antara bunga baru dengan semua bunga yang ada dalam dataset. Gunakan metrik jarak Euclidean, rumusnya sesuai dengan Gambar 1:
jarak = √ ((x1 - x2) ^2 + (y1 - y2) ^2 + (z1 - z2) ^2)
Dimana:
(x1, y1, z1) adalah koordinat warna bunga baru
(x2, y2, z2) adalah koordinat warna bunga dalam dataset
Contoh perhitungan jarak antara bunga baru dan Mawar 1:
jarak = √ ((150 - 255) ^2 + (100 - 100) ^2 + (150 - 50) ^2) = √ (1025 + 0 + 1025) = √2050 ≈ 45.24
Hitung jarak untuk semua bunga dalam dataset dan catat hasilnya.
Urutkan tetangga: Urutkan semua bunga berdasarkan jaraknya dengan bunga baru, dari yang terdekat hingga terjauh.
Bunga	Jarak
Mawar 1	45.24
Mawar 2	36.06
Anggrek 1	53.85
Melati 2	60.83
Melati 1	68.33
Anggrek 2	82.07
3 tetangga terdekat adalah Mawar 1, Mawar 2, dan Anggrek 1.
Tentukan kelas: Kelas yang diprediksi untuk bunga baru adalah kelas mayoritas dari 3 tetangga terdekat.
Mawar 1 dan Mawar 2 memiliki kelas Mawar.
Anggrek 1 memiliki kelas Anggrek.
Karena kelas mayoritas adalah Mawar, maka bunga baru diklasifikasikan sebagai Mawar.
2.1 Implementasi K-NN dengan Python:
Python
import numpy as np

```
# Data pelatihan
X_train = np.array([
   [1, 2],
   [3, 4],
   [5, 6],
   [7, 8],
   [9, 10]
])
y_train = np.array([0, 0, 1, 1, 2])

# Titik data baru
x_new = np.array([11, 12])

# Hitung jarak
distances = np.sqrt(np.sum((X_train - x_new)**2, axis=1))

# Urutkan tetangga
sorted_distances = np.sort(distances)

# Pilih K tetangga terdekat
k = 3
nearest_neighbors = X_train[distances.argsort()[:k]]
# Tentukan kelas
class_counts = np.bincount(y_train[distances.argsort()[:k]])
predicted_class = np.argmax(class_counts)
print ("Kelas yang diprediksi:", predicted_class)
```

Penjelasan kode:

Import numpy: Digunakan untuk perhitungan matematis.

Data pelatihan: Menyimpan data pelatihan dalam array NumPy.

Titik data baru: Menyimpan titik data baru yang ingin diklasifikasikan.

Hitung jarak: Menghitung jarak antara titik data baru dengan semua titik data dalam dataset pelatihan.

Urutkan tetangga: Mengurutkan tetangga terdekat berdasarkan jaraknya dengan titik data baru.

Pilih K tetangga terdekat: Memilih K tetangga terdekat berdasarkan nilai K yang ditentukan.

Tentukan kelas: Menghitung kelas mayoritas dari K tetangga terdekat dan menetapkan kelas tersebut sebagai kelas yang diprediksi untuk titik data baru.

a. Kelebihan Algoritma KNN:

   Mudah Dipahami dan Diimplementasikan: Konsep KNN sederhana dan mudah diimplementasikan, bahkan bagi pemula yang baru mengenal machine learning. Efektif untuk Berbagai Jenis Data: KNN dapat digunakan untuk mengklasifikasikan berbagai jenis data, seperti data numerik, kategorikal, dan bahkan gambar. Tidak Memerlukan Pra-pemrosesan Data yang Rumit: KNN tidak memerlukan pra-pemrosesan data yang rumit, sehingga dapat digunakan dengan mudah pada dataset yang baru.

b. Kekurangan Algoritma KNN:

   Sensitif terhadap Outlier: KNN dapat dipengaruhi oleh outlier, yaitu titik data yang jauh berbeda dari titik data lain dalam dataset.

   Perhitungan Jarak yang Kompleks: Pada dataset yang besar, perhitungan jarak antara titik data baru dengan semua titik data dalam dataset pelatihan dapat memakan waktu lama. Pemilihan Nilai K yang Krusial: Pemilihan nilai K yang tepat sangat penting untuk performa KNN. Nilai K yang terlalu kecil dapat menyebabkan overfitting, sedangkan nilai K yang terlalu besar dapat menyebabkan underfitting.

Bab 5. Validasi Model Dalam Klasifikasi

5.1 Pengertian dan Pentingnya Validasi Model Klasfikasi

Validasi model adalah proses mengevaluasi dan menilai apakah model yang telah dibuat mampu memenuhi tujuan dan kebutuhan yang diharapkan (Imania & Bariah, 2019). Tujuan utama dari validasi model adalah untuk memastikan bahwa model tersebut akurat, handal, berhasil melakukan generalisasi, dan dapat diinterpretasikan.

Manfaat Validasi Model :

- Meningkatkan kepercayaan terhadap hasil yang diberikan oleh model.
- Mengidentifikasi potensi kesalahan dan kelemahan pada model.
- Membantu dalam memilih model yang paling tepat untuk suatu masalah.
- Menyediakan informasi untuk memperbaiki model.

Metode Validasi Model

- Validasi Holdout (Holdout Validation)
- Holdout validation adalah salah satu metode validasi yang umum digunakan dalam data mining.

Metode validasi holdout ada 2 jenis, yaitu :

- Bertujuan untuk Evaluasi Model (*Model Evaluation*)
- Bertujuan untuk Memilih Model (*Model Selection*)

Berikut penjelasan kedua jenis metode validasi holdout tersebut :

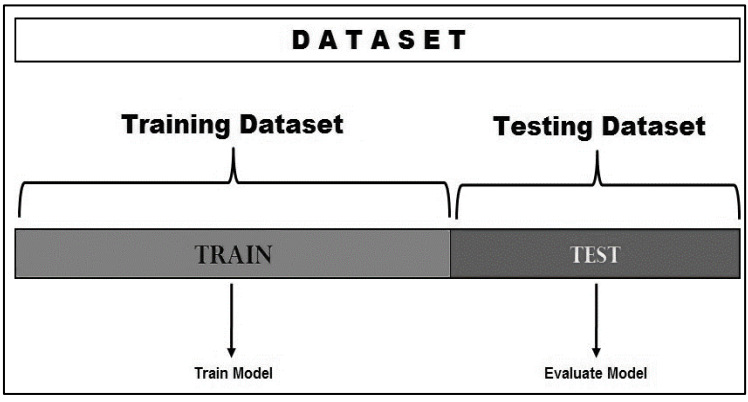
1. Jenis *Model Evaluation*

Validasi holdout Jenis *Model Evaluation* adalah untuk mengevaluasi kinerja model yang telah dilatih. Metode holdout jenis ini membagi dataset menjadi 2 (dua) bagian yaitu :

a. Dataset pelatihan (*training dataset*)

Dataset pelatihan digunakan saat proses training berlangsung atau pada tahap membangun model (*modelling*) (Asroni et al., 2023). Sementara dataset pengujian digunakan saat model yang sudah terbangun akan diuji untuk mengukur kinerja model. Proporsi banyaknya dataset pada masing-masing bagian biasanya diatur agar dataset pelatihan lebih besar daripada dataset pengujian. Ini untuk memastikan bahwa model memiliki data yang cukup untuk belajar dengan baik. Yang umum biasanya 80% untuk dataset pelatihan dan 20% dataset pengujian, atau pilihan lain yang juga populer: 70% untuk dataset pelatihan dan 30% dataset pengujian. Proporsi lain bisa saja dicoba selama eksperimen berlangsung.

Secara ringkas pembagian dataset pada metode validasi holdout jenis model evaluation disajikan seperti gambar 9 berikut :



Gambar 9. Pembagian Dataset Metode Holdout Model Evaluation

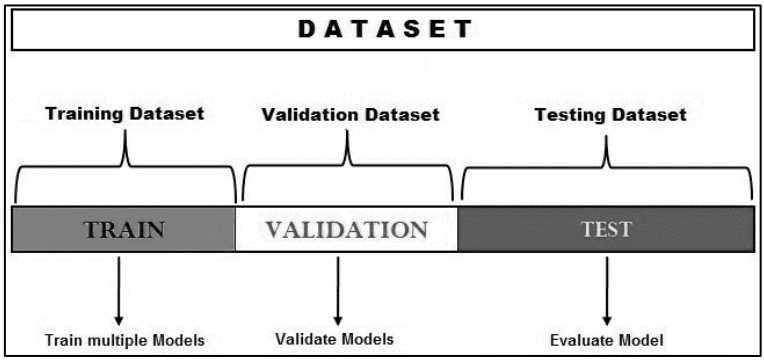
b. Jenis *Model Selection*

Validasi holdout jenis *Model Selection* digunakan untuk memilih model terbaik dari beberapa model yang dilatih. Dengan membandingkan kinerja model dapat dipilih model yang lebih baik, bahkan pada model ini dengan melakukan perubahan

parameter saat pelatihan dapat dihasilkan arsitektur model yang memiliki kinerja yang paling baik. Hal ini membantu dalam memastikan bahwa model yang digunakan adalah yang paling sesuai untuk kasus penggunaan tertentu.

Ada pun metode holdout jenis *Model Selection* ini membagi dataset menjadi 3 (tiga) bagian yaitu :

- Dataset pelatihan (training dataset)
- Dataset pengujian (testing dataset)
- Dataset validasi (validation dataset)

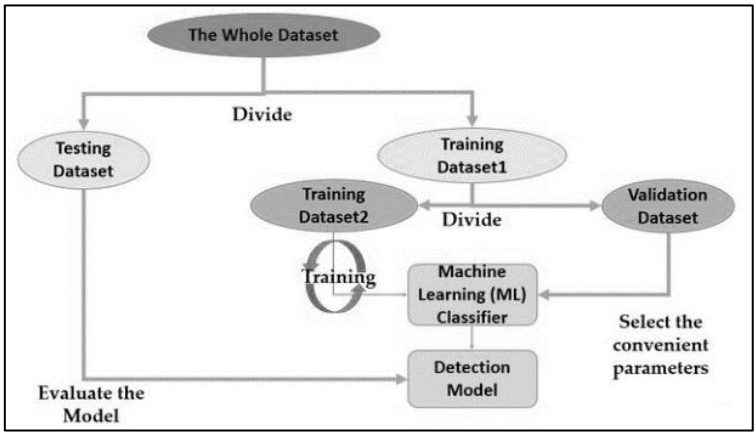


Gambar 10. Pembagian Dataset Metode Holdout Model Selection (Maulid, n.d.)

Seperti pada gambar 10, dataset tambahan adalah dataset validasi, diambil dari sebagian dataset pelatihan. Proporsi untuk dataset pelatihan dan dataset validasi dapat bervariasi tergantung pada ukuran dataset dan kompleksitas model yang diuji. Misalnya, 60% data untuk pelatihan, 20% untuk validasi, dan 20% untuk pengujian. Dengan membagi dataset menjadi tiga bagian (pelatihan, validasi, dan pengujian), kita dapat menggunakan subset validasi untuk mengevaluasi kinerja model dan melakukan penyetelan parameter selama pelatihan berlangsung, sementara dataset pengujian tetap digunakan untuk menguji kinerja akhir model yang telah dipilih. Ini telah terbukti membantu dalam memastikan bahwa model yang dipilih tidak

hanya memiliki kinerja yang baik pada data pelatihan, tetapi juga dapat menggeneralisasi dengan baik pada data baru yang belum pernah dilihat sebelumnya. Jenis ini termasuk salah satu metode mencegah *overfitting*.

Lebih jelas bagaimana metode validasi holdout membagi dataset saat pelatihan dan pengujian dilakukan dapat dilihat pada gambar 11 berikut :



Gambar 11. Alur Pembagian Dataset

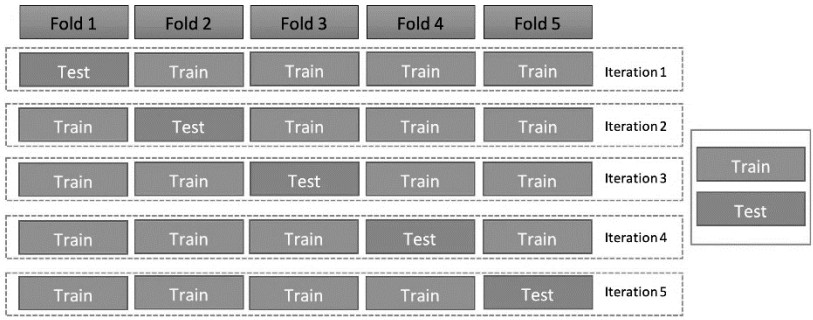
Dari gambar di atas terlihat bahwa keseluruhan dataset (*the whole dataset*) pada saat awal dibagi 2 (yaitu testing dan training). Lalu pada dataset training dapat dibagi lagi untuk mendapatkan porsi dataset lain yang disebut dataset validasi (validation dataset). Dari gambar di atas tampak bahwa validation dataset digunakan untuk mengontrol parameter model selama training berlangsung agar dihasilkan kinerja yang lebih baik.

c. Cross-Validation

Cross-validation, dalam konteks data mining, merupakan teknik evaluasi model yang lebih maju dibandingkan dengan holdout validation (Milenia, 2022). Sementara holdout validation hanya membagi data menjadi dua atau tiga bagian, yakni

pelatihan, pengujian dan validasi, maka cross-validation mengadopsi pendekatan yang lebih mendalam dengan menggunakan dataset secara berulang untuk evaluasi yang lebih menyeluruh.

Proses *cross-validation* melibatkan pembagian dataset menjadi k subset yang sama ukurannya, yang disebut fold. Selama proses ini, satu fold digunakan sebagai data pengujian untuk menguji kinerja model, sedangkan k-1 fold lainnya digunakan sebagai data pelatihan. Proses ini diulang k kali, dengan setiap fold digunakan sebagai data pengujian secara bergantian. Hasil kinerja model pada setiap iterasi kemudian diambil rata-ratanya untuk mendapatkan estimasi kinerja model secara keseluruhan. Contoh pembagian dataset Metode *Cross-validation* dengan k =5 seperti berikut :



Gambar 12. Pembagian dataset Pada Metode Cross-validation (Handayanto, 2020)

Dari gambar 12 diatas tampak bahwa pada *cross-validation* terdapat dua konsep penting berkaitan dengan iterasi dan pembagian dataset, yaitu : iterasi (iterations) dan per fold (k-fold).

d. Iterasi (Iterations)

Iterasi mengacu pada jumlah kali proses *cross-validation* dilakukan. Dalam setiap iterasi, dataset dibagi ulang menjadi subset pelatihan dan pengujian, model dilatih pada subset

pelatihan, dan kemudian diuji pada subset pengujian. Jumlah iterasi ini dapat ditentukan sebelumnya dan bergantung pada ukuran dataset serta kompleksitas model yang sedang diuji. Misalnya, jika kita memilih untuk melakukan 5-fold *cross-validation* dan memiliki 1000 sampel data, maka akan ada 5 iterasi di mana dataset akan dibagi menjadi 5 bagian yang sama besar.

e. Fold (k-fold)

Fold, atau k-fold, mengacu pada jumlah bagian atau lipatan dataset dibagi selama setiap iterasi *cross-validation*. Misalnya, dalam 5-fold *cross-validation*, dataset akan dibagi menjadi 5 bagian yang sama besar (Nugroho & Amrullah, 2023). Pada setiap iterasi, satu bagian akan digunakan sebagai subset pengujian, sementara bagian lainnya digunakan sebagai subset pelatihan. Proses ini diulang sebanyak k kali, di mana setiap bagian dataset digunakan sekali sebagai subset pengujian dan k-1 bagian lainnya digunakan sebagai subset pelatihan.

Dengan menggunakan konsep iterasi dan per fold, *cross-validation* memastikan bahwa seluruh dataset digunakan baik untuk pelatihan maupun pengujian. Ini membantu dalam menghasilkan estimasi yang lebih stabil dari kinerja model dan mengurangi kemungkinan bias yang mungkin muncul karena karakteristik khusus dari satu set data tertentu. Selain itu, dengan melakukan iterasi dan pembagian dataset secara berulang-ulang, *cross-validation* memberikan perkiraan yang lebih akurat tentang seberapa baik model akan berkinerja pada data baru yang belum pernah dilihat sebelumnya.

Sebagai tambahan, ada satu jenis metode *cross-validation* yang lebih tajam/rinci yang disebut dengan *Leave-One-Out Cross-Validation* (LOOCV). Pada metode ini yang menjadi fold adalah 1 sampel data. Misalnya kita memiliki 1000 baris dataset, maka fold nya adalah 1000, jadi k (jumlah fold) sama dengan jumlah data yang dimiliki. Metode LOOCV ini memang memberikan estimasi

kinerja model yang lebih akurat karena menggunakan seluruh dataset untuk evaluasi. Namun, kelemahannya adalah LOOCV membutuhkan waktu komputasi yang lebih besar, terutama untuk dataset yang besar, karena harus melakukan pelatihan dan evaluasi model sebanyak jumlah sampel dalam dataset.

f. Evaluasi Model Klasifikasi

Evaluasi model adalah bagian dari validasi model. Validasi tujuannya untuk memastikan kualitas model secara umum, sedangkan evaluasi lebih fokus pada penilaian kinerja model (Kulsum et al., 2024).

Evaluasi pada model datamining klasifikasi adalah proses untuk mengevaluasi seberapa baik model klasifikasi dapat memprediksi label kelas dari data baru yang belum pernah dilihat sebelumnya, dalam hal ini yang dimaksud data baru adalah daset pengujian.

Ada beberapa metode yang dapat digunakan untuk mengevaluasi model klasifikasi, yaitu :

- Matriks Konfusion (Confusion Matrix):
- Matriks konfusion menyajikan jumlah prediksi yang benar dan salah yang dibuat oleh model klasifikasi dalam bentuk tabel. Ini mencakup True Positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN).

Tabel 2. Tabel Matriks Konfusion

		Aktual	
		Position	Negatip
Prediksi	Positip	TP	FP
	Negatip	FN	TN

- True Positives (TP) : jumlah prediksi positif yang benar oleh model, artinya model dengan benar mengidentifikasi instans positif.

- True Negatives (TN) : jumlah prediksi negatif yang benar oleh model, artinya model dengan benar mengidentifikasi instans negatif.
- False Positives (FP) : jumlah prediksi positif yang salah oleh model, artinya model dengan keliru mengidentifikasi instans negatif sebagai positif.
- False Negatives (FN) : jumlah prediksi negatif yang salah oleh model, artinya model dengan keliru mengidentifikasi instans positif sebagai negatif.

g. Akurasi (Accuracy):

Akurasi mengukur proporsi dari total prediksi yang benar dari model terhadap jumlah total data.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

- Presisi (Precision)
  Presisi mengukur proporsi dari prediksi positif yang benar terhadap total prediksi positif yang dilakukan oleh model.
- Precision = (TP)/(TP+FP)
- Recall (Sensitivity atau True Positive Rate)
  Recall mengukur proporsi dari kelas positif yang berhasil diidentifikasi oleh model.

$$\text{Recall} = (TP)/(TP+FN)$$

h. F1-Score

F1-score adalah rata-rata harmonik dari presisi dan recall, memberikan gambaran tentang keseimbangan antara keduanya.

$$\text{F1-Score} = 2\times(\text{Precision}\times\text{Recall})/(\text{Precision}+\text{Recall})$$

i. Kurva ROC (Receiver Operating Characteristic Curve) dan AUC-ROC (Area Under the ROC Curve):

Kurva ROC menggambarkan perbandingan antara True Positive Rate (TPR) dan False Positive Rate (FPR) pada berbagai nilai threshold. AUC-ROC mengukur area di bawah kurva ROC, memberikan gambaran tentang seberapa baik model membedakan antara kelas positif dan negatif.

j. Kurva Precision-Recall:

Kurva Precision-Recall menggambarkan hubungan antara presisi dan recall pada berbagai nilai threshold. Ini dapat memberikan wawasan tambahan terutama pada dataset yang tidak seimbang (imbalance).

k. F-Measure:

F-Measure merupakan ukuran gabungan dari presisi dan recall, dengan menggunakan parameter beta untuk menekankan pentingnya salah satu ukuran lebih dari yang lain.

5.2 Contoh Penerapan Evaluasi Model

Diketahui model klasifikasi email yang dilatih untuk membedakan antara email yang mengandung spam dan email yang tidak mengandung spam (normal). Tahap pengujian dengan 200 dataset testing diperoleh hasil seperti berikut :

Tabel 3. Matriks Konfusion

		Aktual	
		Spam (Aktual Positip)	Not Spam (Aktual Negatip)
Prediksi	Spam (Prediksi Positip)	90	10
	Not Spam (Prediksi Negatip)	5	95

Informasi dari tabel di atas :

TP (True Positive) : 90.

Ini adalah jumlah email spam yang dengan benar diklasifikasikan sebagai spam.
FN (False Negative) : 5.
Ini adalah jumlah email normal yang salah diklasifikasikan sebagai spam.
FP (False Positive) : 10.
Ini adalah jumlah email spam yang salah diklasifikasikan sebagai normal.
TN (True Negative) : 95.
Ini adalah jumlah email normal yang dengan benar diklasifikasikan sebagai normal.
Dari Tabel Matriks Konfusion dan informasi di atas selanjutnya dapat dilakukan perhitungan untuk paramater lain :
Akurasi : (TP + TN) / Total = (90 + 95) / 200 = 92.5%
Presisi : TP / (TP + FP) = 90 / (90 + 10) = 90%
Recall : TP / (TP + FN) = 90 / (90 + 5) = 94.7%
F1-Score : 2 * (Precision * Recall) / (Precision + Recall)
= 2 * (0.9 * 0.947) / (0.9 + 0.947) = 0.923

Hasil di atas dapat diinterpretasi bahwa :

Akurasi 92.5% menunjukkan bahwa model cukup baik dalam mengklasifikasikan email secara keseluruhan.
Presisi 90% menunjukkan bahwa 90% dari email yang diklasifikasikan sebagai spam oleh model benar-benar spam.
Recall 94.7% menunjukkan bahwa model mengidentifikasi sebagian besar email spam (hanya 5% yang terlewatkan).
F1-Score 0.923 memberikan keseimbangan antara presisi dan recall.

Bab 6. Decission Tree

Decission tree bekerja dengan merepresentasikan berbagai pilihan yang tersedia dan akibat dari setiap pilihan tersebut melalui struktur berbentuk pohon. Dengan memecah permasalahan menjadi serangkaian keputusan yang lebih kecil, decission tree memungkinkan kita untuk mengamati pola dan relasi dalam data yang mungkin sulit dipahami secara langsung. Dengan kata lain decission tree atau pohon keputusan merupakan representasi sederhana yang berasal dari teknik pemilahan data dalam data mining (Novita, Elfaladonna, & Ganiardi, 2024).

Dalam proses analisis, kita akan mempelajari bagaimana decission tree digunakan untuk:

- Klasifikasi Data: Meramalkan kategori atau label dari data berdasarkan fitur-fitur yang ada.
- Identifikasi Pola: Mendeteksi pola atau hubungan yang terdapat dalam dataset yang mungkin tidak terlihat secara langsung.
- Pengambilan Keputusan: Menyediakan landasan untuk membuat keputusan berdasarkan pemahaman yang diperoleh dari analisis decission tree.

6.1 Konsep Dasar Decission Tree

Decission tree adalah metode prediksi yang dapat diterapkan untuk klasifikasi dan prediksi tugas (Bahri & Lubis, 2020). Teknik "membagi dan menaklukkan" digunakan oleh decission tree untuk memecah masalah menjadi subset-subset yang lebih kecil. Ini berarti bahwa decission tree membagi ruang masalah menjadi himpunan masalah yang lebih kecil. Proses

yang terjadi dalam decission tree adalah mengubah data tabel menjadi struktur pohon yang disebut model tree. Model tree ini akan menghasilkan aturan dan kemudian disederhanakan (Bahri & Lubis, 2020).

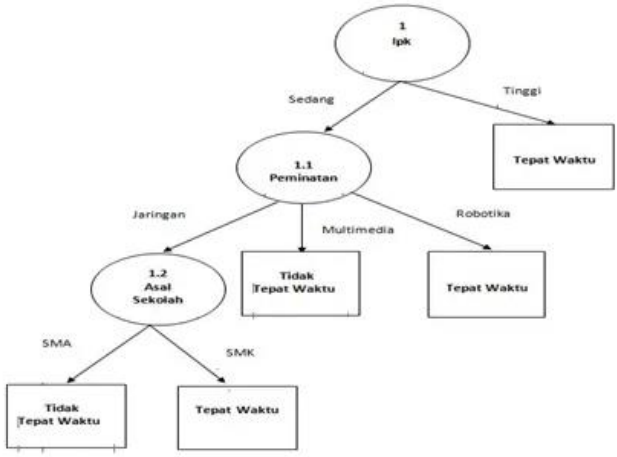
Dalam pohon keputusan, setiap simpul mewakili pengujian terhadap atribut tertentu (dilukiskan sebagai kotak), cabang menunjukkan hasil dari pengujian tersebut (dilukiskan sebagai panah dengan label dan arah), sedangkan daun mewakili kelas yang diprediksi (dilukiskan sebagai lingkaran) (Kuntum Khairina).

Saat membuat pohon keputusan, metode pemilihan atribut digunakan untuk menentukan kriteria terbaik dalam membagi data ke dalam kelas-kelas yang berbeda. Kriteria ini mencakup pemilihan atribut pemisah, titik pemisah, dan subset pemisah.

Attribute selection measure adalah pendekatan heuristik untuk memilih kriteria terbaik dalam membagi data latih ke dalam kelas-kelas (Prehanto et al., 2020). Idealnya, setiap partisi yang dihasilkan harus murni, yang berarti semua tupel dalam suatu partisi harus memiliki kelas yang sama. Oleh karena itu, kriteria terbaik adalah kriteria yang dapat membagi data mendekati murni. Attribute selection measure membuat peringkat atribut-atribut pada data latih. Atribut yang menduduki peringkat teratas dipilih sebagai atribut pemisah. Jika atribut pemisah bersifat kontinu, titik pemisah juga ditentukan. Jika atribut pemisah bersifat diskrit tetapi pohon keputusan yang dibentuk harus biner, subset pemisah akan ditentukan. Ada tiga jenis attribute selection measures yang umum digunakan, yaitu information gain, gain ratio, dan gain index.

Pada sisi lain, jumlah cabang yang berlebihan pada pohon keputusan mungkin menandakan adanya gangguan seperti noise (kesalahan dalam mencatat nilai) atau outlier (nilai yang keluar dari rentang yang seharusnya) dalam data latih. Pemotongan pohon dapat dilakukan untuk mengidentifikasi dan

menghilangkan cabang-cabang tersebut, dengan harapan dapat meningkatkan akurasi model klasifikasi.



Gambar 13. Strutur Pohon Keputusan (Hin, 2019)

Kelebihan dan Kekurangan Penggunaan Pohon Keputusan jika dibandingkan dengan metode lain dalam mengembangkan algoritma data mining, yaitu:

1. Pohon keputusan sangat mudah dipahami, diinterpretasikan, dan sangat cocok untuk visualisasi data. Meski tanpa data yang sangat kuat, pohon keputusan tetap mampu menghasilkan informasi yang berguna. Persiapan data untuk prosesnya juga hanya memerlukan persiapan minimal, bahkan Anda tidak perlu melalui proses one-hot encoding.
2. Kemampuan Penambahan Opsi Baru. Opsi baru selalu dapat ditambahkan dengan mudah ke dalam struktur yang sudah ada.
3. Kemampuan Memilih Opsi Terbaik. Pohon keputusan dapat memilih opsi terbaik dari semua opsi yang tersedia. Penggunaan pohon keputusan dapat digabungkan dengan alat pengambil keputusan lainnya.

4. Pohon keputusan dapat bekerja dengan baik dengan variabel numerik maupun kategoris.
5. Pemilihan variabel dilakukan secara otomatis, sehingga variabel yang tidak penting tidak akan memengaruhi hasil akhir, meskipun ada variabel yang saling berkorelasi (multikolinearitas).

Namun, meskipun memiliki keunggulan, pohon keputusan juga memiliki kelemahan. Kelemahan utamanya diantaranya :

a. Strukturnya yang terbuka. Sifat struktur pohon keputusan yang terbuka dapat membuatnya menjadi sangat rumit.
b. Meskipun struktur yang rumit mungkin menghasilkan hasil yang akurat, namun juga bisa mengurangi fokus hanya pada keputusan dan masukan (PT. Algoritma Data Indonesia, 2022).

6.2 Algoritma Decission Tree

ID3 (Iterative Dichotomiser 3), C4.5 (penerus ID3), dan CART (Classification and Regression Trees) adalah beberapa contoh algoritma pembangunan pohon keputusan. Ketiga algoritma ini pada dasarnya memiliki karakteristik yang serupa dalam proses pembentukan pohon keputusan, yaitu top-down dan divide-and-conquer. Top-down berarti pohon keputusan dibangun dari simpul akar ke daun, sedangkan divide-and-conquer berarti data pelatihan dipartisi secara rekursif ke dalam bagian yang lebih kecil saat membangun pohon. Apakah pohon keputusan yang dihasilkan biner atau tidak ditentukan oleh pengukuran pemilihan atribut atau algoritma yang digunakan.

Algoritma ID3

Algoritma ID3, atau Iterative Dichotomiser 3, merupakan sebuah metode yang diciptakan oleh J. Ross Quinlan sejak tahun 1986 untuk membangun pohon keputusan. Algoritma ini melakukan pencarian menyeluruh pada semua kemungkinan pohon keputusan dan berupaya membangun pohon keputusan secara top-down (dari atas ke bawah) (Dellila, 2023).

Langkah-langkah algoritma ID3 dibagi menjadi 6 tahapan, yaitu:

1. Menyiapkan Dataset. Dataset yang digunakan yaitu dataset yang berlabel nominal. Untuk algoritma ID3 menggunakan data jenis klasifikasi.
2. Menghitung Nilai Entropy. Konsep Entropy digunakan untuk mengukur "seberapa informatifnya" sebuah node (yang biasanya disebut seberapa baiknya).
   Rumus Entropy:

$$\text{entropy}(s) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

   Keterangan :
   S : Himpunan (dataset) Kasus
   p_+ : Jumlah label yang bersolusi positif (mendukung) dibagi total kasus.
   p_- : Jumlah label yang bersolusi negative (tidak mendukung) dibagi total kasus.
3. Menghitung Nilai Information Gain Information gain adalah metrik pemisahan yang memanfaatkan perhitungan entropy. Information gain digunakan untuk menilai seberapa baik suatu atribut dalam memisahkan data. Rumus Information Gain:

$$\text{Gain}(A) = \text{entropy}(s) - \sum_{i=1}^{k} \frac{|Si|}{|S|} x \, \text{entropy}(Si)$$

   Keterangan :
   S : Ruang data (sampel)
   A : Atribut
   $|\ Si\ |$: Jumlah sampel data
   $|\ S\ |$: Jumlah seluruh sampel data
   Entropy (Si): Untuk sampel-sampel yang memiliki nilai i
4. Menentukan Node Akar Node akar atau root node dipilih berdasarkan nilai information gain yang telah dihitung.

Atribut yang memiliki information gain tertinggi akan menjadi root node. Atribut yang sudah dipilih tidak akan dipertimbangkan lagi dalam perhitungan entropy dan information gain selanjutnya.

5. Membuat Node Cabang Node cabang dibentuk berdasarkan perhitungan entropy dan information gain yang berikutnya.
6. Mengulangi Langkah 2 dan 4. Proses langkah 2 dan 4 diulangi sampai terbentuk pohon keputusan.

a. Algoritma C.45

Algoritma C4.5 merupakan hasil pengembangan dari algoritma ID3 (Iterative Dichotomiser 3) yang digunakan dalam pembentukan pohon keputusan (Setio et al., 2020). Algoritma ini dirancang oleh Ross Quinlan pada tahun 1993. C4.5 memiliki keunggulan dalam menangani data yang memiliki nilai yang hilang dan data dengan sifat kontinu. Selain itu, algoritma ini mampu menghasilkan pohon keputusan yang lebih singkat dan lebih mudah dimengerti dibandingkan dengan ID3. Salah satu keunggulan utama dari C4.5 adalah kemampuannya dalam menangani atribut yang memiliki nilai kontinu dan kategorikal secara bersamaan.

Ada beberapa tahapan dalam membuat sebuah pohon keputusan dalam algoritma C4.5 yaitu (Larose, 2006):

1) Menyiapkan data pelatihan. Data pelatihan sering kali diambil dari rekaman historis yang mencakup peristiwa-peristiwa masa lampau dan telah dikelompokkan ke dalam kelas-kelas tertentu.
2) Menghitung root node dari pohon. Root node akan dipilih dari atribut yang akan menjadi pilihan pertama, dengan cara menghitung nilai gain dari setiap atribut. Atribut dengan nilai gain tertinggi akan menjadi root node. Sebelum menghitung nilai gain dari atribut, perlu dihitung terlebih dahulu nilai entropi. Rumus yang digunakan untuk menghitung nilai entropi adalah:

$$Entropy\ (S) = \sum_{i=1}^{n} -p_i \log_2(p_i)$$

Keterangan :
S = Himpunan kasus
n = Jumlah partisi S
pi = Proporsi Si terhadap S

3) Dalam proses pencarian nilai *Entropy*(S) jika semua nilai kriteria terhadap *attribute* keputusan sama, maka nilai *Entropy* =1, dan jika hanya satu nilai kriteria yang tidak sama dengan 0, maka nilai *Entropy* = 0 (Puspita, Aminah, & Arif, 2021).
4) Menghitung nilai Gain menggunakan Persamaan 2.

$$Gain(S, A) = Entropy(S) - \sum_{I=1}^{n} \frac{|Si|}{|S|} * Entropy(Si)$$

Keterangan :
S : Jumlah kasus (*Sampling*)
A : *attribute*
n : Jumlah partisi S
|Si| : Jumlah kasus pada partisi ke-i
|S| : Jumlah kasus dalam S

5) Lakukan iterasi langkah 2 dan langkah 3 sampai semua rekaman terbagi. Proses partisi pohon keputusan akan berhenti ketika: a. Semua rekaman dalam simpul N memiliki kelas yang sama. b. Tidak ada atribut dalam rekaman yang dapat dipartisi lagi. c. Tidak ada rekaman kosong di dalam cabang.

b. Algoritma CART

CART merupakan salah satu metode atau algoritma yang digunakan untuk menganalisis data dengan menggunakan pohon keputusan. Metode ini dikembangkan oleh Leo Breiman, Jerome H. Friedman, Richard A. Olshen, dan Charles J. Stone sekitar tahun

1980-an. Tujuan utama dari CART adalah untuk menghasilkan kelompok data yang akurat sebagai penjelasan lebih lanjut dari suatu klasifikasi (Sadat et al., 2023). Hasil dari algoritma ini berupa pohon klasifikasi jika variabel targetnya adalah data kategoris. Sedangkan jika variabel targetnya adalah data numerik atau kontinu, maka hasilnya berupa pohon regresi (Ilham, Harun, & Arwansyah, 2013).

Langkah-langkah algoritma CART adalah sebagai berikut (Astuti, 2018):

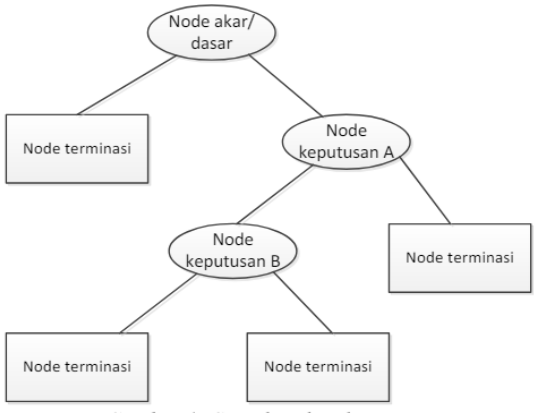
1) Mempersiapkan data yang akan diklasifikasikan
2) Menentukan variabel prediktor sebagai variabel sebagai dasar pengelompokan berdasarkan variabel tujuan (target)
3) Menentukan calon cabang (candidate split) kiri dan kanan
4) Mengukur goodness (kesesuaian) dari calon masing-masing cabang s pada simpul keputusan t yang dihitung dengan rumus :

$$\Phi(s \mid t) = 2\, P_L P_R \sum_{j=1}^{jlh\ kategori} \left| P(j \mid t_L) - P(j \mid t_R) \right|$$

Keterangan :
tL = calon cabang kiri dari node keputusan t
tR = calon cabang kanan dari node keputusan t
PL = jumlah data catatan pada calon cabang kiri tL / jumlah data catatan seluruhnya
PR = jumlah data catatan pada calon cabang kanan tR / jumlah data catatan seluruhnya P(j | tL) = jumlah data catatan berkategori j pada calon caang kiri tL / jumlah data catatan pada node keputusan t
P(j | tR) = jumlah data catatan berkategori j pada calon caang kiri tR / jumlah data catatan pada node keputusan t

5) Tentukan calon cabang untuk node keputusan dengan memilih nilai terbesar, cabang ini tidak dihitung lagi selanjutnya.

6) Gambarkan cabang node keputusan dan node kejadian terminasi.
7) Ulangi langkah 4 sampai tidak terdapat lagi cabang node keputusan.
8) Berikut adalah ilustrasi dari hasil penerapan algoritma CART dalam bentuk pohon keputusan.



Gambar 14. Ilustrasi pohon Keputusan

a) Pemilihan Atribut

Pemilihan atribut dalam pohon keputusan adalah tahap penting dalam pembentukan model. Atribut dipilih berdasarkan kriteria tertentu, seperti information gain atau gini impurity, yang mengevaluasi kemampuan atribut dalam memisahkan data ke dalam kelas yang berbeda. Atribut dengan nilai kriteria tertinggi dipilih sebagai pemisah pada setiap tingkat pohon keputusan. Proses ini berlanjut sampai tidak ada lagi atribut yang dapat memisahkan data secara lebih baik atau sampai mencapai kriteria penghentian tertentu, seperti mencapai batas kedalaman maksimum pohon.

Langkah-langkah dalam pemilihan atribut dalam pembentukan pohon keputusan adalah sebagai berikut:

1. Penilaian Kriteria Pemilihan: Tentukan kriteria yang digunakan untuk seleksi atribut, seperti information gain, gain ratio, atau gini impurity.
2. Perhitungan Kriteria: Lakukan perhitungan kriteria yang dipilih untuk setiap atribut dalam kumpulan data, seperti menghitung information gain untuk setiap atribut.
3. Seleksi Atribut: Pilih atribut yang memiliki nilai kriteria tertinggi sebagai pemisah pada tingkat pohon keputusan saat ini.
4. Pembagian Data: Pisahkan data ke dalam cabang-cabang berdasarkan nilai atribut yang terpilih.
5. Penilaian Kondisi Berhenti: Evaluasi apakah kondisi berhenti telah terpenuhi, seperti ketika semua catatan dalam simpul memiliki kelas yang sama atau tidak ada lagi atribut yang bisa dipilih.
6. Iterasi: Ulangi langkah-langkah di atas untuk setiap cabang pohon keputusan sampai kondisi berhenti terpenuhi.
7. Proses-proses ini sangat penting dalam membentuk pohon keputusan yang optimal untuk dataset tertentu.
8. Pengukuran Impurity dan Informasi:
9. Membangun model Decission Tree menantang karena mengharuskan pemilihan atribut untuk root node dan node di cabang selanjutnya, yang dilakukan dengan mengukur tingkat impuritas atribut (variabel independen) terhadap target (variabel dependen). Atribut yang ideal untuk menjadi root node atau node adalah yang memiliki tingkat keberagaman rendah atau homogenitas tinggi, sehingga impuritasnya rendah. Impuritas adalah ukuran dari seberapa homogen label pada node atau atribut tersebut. Dalam perhitungan impuritas, Decission Tree biasanya melakukan iterasi untuk membuat pohon sederhana untuk setiap atribut. Pada pohon klasifikasi, pohon sederhana tersebut akan mengevaluasi seberapa baik

atribut tersebut dalam memprediksi label kelas pada target (Nurfaizi, 2022).

Secara umum, Decission Tree memproses satu per satu atribut terhadap kelas targetnya. Ada beberapa metode untuk menghitung tingkat impuritas pada setiap atribut, termasuk:

Gini Impurity Gini Impurity adalah metode pengukuran yang digunakan dalam pembentukan Decission Tree untuk menentukan cara atribut dari kumpulan data harus membagi node untuk membentuk pohon.

Entropy & Information Gain Entropy dan Information Gain adalah metode lain untuk mengukur impuritas dan keuntungan informasi dari atribut dalam pembentukan pohon keputusan (Dwi, 2023).

Rumus Gini Impurity :

$$Gini(t) = 1 - \sum_{i=1}^{j} P(i|t)^2$$

Penjelasan : Rumus Gini Impurity adalah dengan mengurangkan jumlah probabilitas kuadrat dari setiap kelas dari satu

b) Pruning

Mengidentifikasi dan menghapus cabang yang tidak diperlukan pada pohon yang sudah terbentuk adalah penting karena pohon keputusan yang kompleks dapat disederhanakan dengan melakukan pemangkasan berdasarkan tingkat kepercayaan (confident level). Tujuan dari pemangkasan pohon adalah untuk mengurangi ukuran pohon dan juga untuk mengurangi tingkat kesalahan prediksi pada data baru yang diproses melalui pendekatan divide and conquer. Terdapat dua pendekatan pruning (Lestari, 2020):

a. Pre-pruning, yaitu menghentikan pembangunan subtree lebih awal (dengan memutuskan untuk tidak membagi data pelatihan lebih lanjut). Ketika dihentikan, node berubah

menjadi leaf (simpul akhir), yang mewakili kelas yang paling umum di antara subset sampel.

b. Post-pruning, yaitu menyederhanakan pohon setelah pembangunan selesai dengan menghapus beberapa cabang subtree. Node yang jarang digunakan akan diubah menjadi leaf (simpul akhir) dengan kelas yang paling umum.

Setelah pohon keputusan dibangun, langkah selanjutnya adalah membentuk aturan keputusan. Aturan ini dapat dihasilkan dari pohon keputusan dengan melakukan penelusuran dari akar hingga daun. Setiap simpul dan percabangannya akan menjadi bagian dari pernyataan "if", sementara nilai di daun akan menjadi bagian dari pernyataan "then". Setelah semua aturan dibuat, mereka dapat disederhanakan atau digabungkan.

Penerapan Decission Tree dalam Berbagai Bisang. Berikut adalah beberapa contoh penerapan pohon keputusan dalam berbagai bidang:

- Keuangan: Dalam industri keuangan, pohon keputusan digunakan untuk mengidentifikasi profil risiko pelanggan, menilai kelayakan peminjam untuk pinjaman, atau memprediksi perilaku pasar keuangan.
- Kesehatan: Dalam bidang medis, pohon keputusan digunakan untuk mendiagnosis penyakit, memprediksi risiko kesehatan pasien, atau menentukan jenis perawatan yang paling cocok.
- Pemasaran: Dalam pemasaran, pohon keputusan membantu dalam segmentasi pelanggan, menentukan strategi harga, atau memprediksi perilaku pembelian pelanggan.
- Sumber Daya Manusia: Dalam manajemen SDM, pohon keputusan digunakan dalam proses rekrutmen karyawan, evaluasi kinerja, atau mengidentifikasi faktor-faktor yang memengaruhi kepuasan karyawan.
- E-Commerce: Dalam industri e-commerce, pohon keputusan digunakan untuk memberikan rekomendasi produk kepada

pelanggan, mendeteksi penipuan transaksi, atau memprediksi tingkat konversi penjualan.

- Teknologi: Dalam bidang teknologi, pohon keputusan digunakan untuk pengenalan pola, deteksi intrusi keamanan, atau pengambilan keputusan dalam sistem otomatisasi dan jaringan cerdas.
- Pendidikan: Dalam dunia pendidikan, pohon keputusan digunakan untuk memprediksi kelulusan siswa, menilai kinerja sekolah, atau mengidentifikasi faktor-faktor yang memengaruhi keberhasilan belajar.

Penerapan pohon keputusan dapat bervariasi tergantung pada kebutuhan dan lingkungan industri. Ini menunjukkan fleksibilitas dan kegunaan yang luas dari teknik ini dalam berbagai konteks.

6.2 Studi Kasus

Tabel 4. Contoh Data Keputusan Bermain Tenis

NO	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Cloudy	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	Yes
7	Cloudy	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Cloudy	Mild	High	TRUE	Yes
13	Cloudy	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai akar.
2. Buat cabang untuk tiap-tiap nilai.

3. Bagi kasus dalam cabang.
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

a. Tahapan pengerjaan:

- Langkah pertama adalah menghitung total kasus, jumlah kasus dengan keputusan "Yes", jumlah kasus dengan keputusan "No", dan entropi dari seluruh kasus serta kasus yang dibagi berdasarkan atribut OUTLOOK, TEMPERATURE, HUMIDITY, dan WINDY. Setelah itu, lakukan perhitungan Gain untuk setiap atribut. Hasil perhitungan tersebut akan ditampilkan di bawah ini.
- Perhitungan Node 1

Tabel 5. Perhitungan node

		jml kasus(S)	Tidak (S1)	Ya (S2)	Entropy	Gain
total		14	4	10	0.86312	
outlook						0.258521
	cloudy	4	0	4	0	
	rainy	5	1	4	0.72193	
	sunny	5	3	2	0.97095	
temp						0.1838509
	col	4	0	4	0	
	hot	4	2	2	1	
	mild	6	2	4	0.9183	
humidity						0.3705065
	high	7	4	3	0.98523	
	normal	7	0	7	0	
windy						0.0059777
	FALSE	8	2	6	0.81128	
	TRUE	6	4	2	0.9183	

Cara Perhitungan Node 1 : Baris total kolom entropy dihitung dengan persamaan:

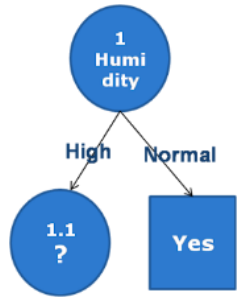
$$Entropy(Total) = \left(-\frac{4}{14} * log_2\left(\frac{4}{14}\right)\right) + \left(-\frac{10}{14} * log_2\left(\frac{10}{14}\right)\right)$$

$$Entropy(Total) = 0{,}863120569$$

Cara Perhitungan Node 2 : Nilai gain pada baris outlook dihitung.

- $Gain(Total, Outlook) = Entropy(Total) - \sum_{i=1}^{n} \frac{|Outlook_i|}{|Total|} * Entropy\ (Outlook_i)$
- $Gain(Total, Outlook) = 0.863120569 - \left(\left(\frac{4}{14} * 0\right) + \left(\frac{5}{14} * 0.723\right) + \left(\frac{5}{14} * 0.97\right)\right)$
- $Gain(Total, Outlook) = 0.23$

Dari hasil tersebut, dapat dilihat bahwa atribut dengan gain tertinggi adalah HUMIDITY, yakni sebesar 0.37. Oleh karena itu, HUMIDITY dapat dipilih sebagai node akar. Terdapat dua nilai atribut dari HUMIDITY, yaitu HIGH dan NORMAL. Nilai atribut NORMAL langsung mengklasifikasikan satu kasus menjadi "Yes", sehingga tidak perlu dilakukan perhitungan lebih lanjut. Namun, untuk nilai HIGH, perlu dilakukan perhitungan tambahan.



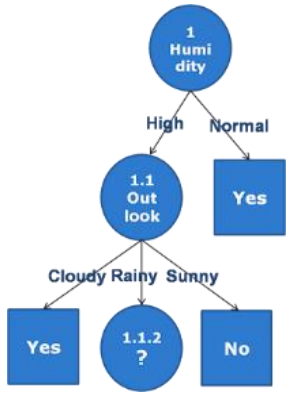
Gambar 15. Langkah pertama pohon Keputusan

- Langkah kedua adalah menghitung total kasus, jumlah kasus dengan keputusan "Yes", dan jumlah kasus dengan keputusan "No". Kemudian, menghitung entropi dari seluruh kasus serta kasus yang dibagi berdasarkan atribut OUTLOOK, TEMPERATURE, dan WINDY, yang dapat menjadi node akar dari nilai atribut HIGH. Setelah itu, lakukan perhitungan Gain untuk setiap atribut.

Tabel 6. Perhitungan Node 1.1

		jml kasus(S)	Tidak (S1)	Ya (S2)	Entropy	Gain
Humidity High		7	4	3	0.98522814	
outlook						0.69951385
	cloudy	2	0	2	0	
	rainy	2	1	1	1	
	sunny	3	3	0	0	
temp						0.02024421
	col	0	0	0	0	
	hot	3	2	1	0.91829583	
	mild	4	2	2	1	
windy						0.02024421
	FALSE	4	2	2	1	
	TRUE	3	2	1	0.91829583	

Atribut OUTLOOK, dengan Gain tertinggi sebesar 0.6995, menjadi node cabang untuk nilai atribut HIGH, dengan nilai-nilai CLOUDY langsung mengklasifikasikan satu kasus sebagai "Yes", SUNNY langsung mengklasifikasikan satu kasus sebagai "No", dan untuk nilai RAINY, perlu dilakukan perhitungan lanjutan.



Gambar 16. Langkah kedua pohon Keputusan

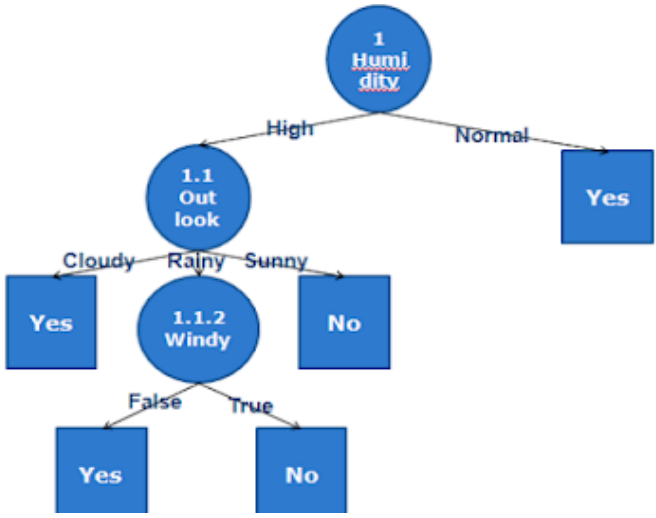
- Langkah ketiga melibatkan menghitung total kasus, jumlah kasus dengan keputusan "Yes", dan jumlah kasus dengan keputusan "No". Selanjutnya, menghitung entropi dari

seluruh kasus serta kasus yang dibagi berdasarkan atribut TEMPERATURE dan WINDY, yang dapat menjadi node cabang dari nilai atribut RAINY. Kemudian, lakukan perhitungan Gain untuk setiap atribut.

Tabel 7. Perhitungan Node 1.1.2

Node 1.1.2		jml kasus(S)	Tidak (S1)	Ya (S2)	Entropy	Gain
Humidity High and Outlook Rainy		2	1	1	1	
temp						0
	cool	0	0	0	0	
	hot	0	0	0	0	
	mild	2	1	1	1	
windy						1
	FALSE	1	0	1	0	
	TRUE	1	1	0	0	

Atribut WINDY memiliki Gain tertinggi sebesar 1, sehingga menjadi node cabang untuk nilai atribut RAINY. Atribut WINDY memiliki dua nilai, yaitu FALSE dan TRUE. Nilai atribut FALSE dan TRUE langsung mengklasifikasikan kasus masing-masing sebagai "Yes" dan "No", sehingga tidak perlu dilakukan perhitungan lebih lanjut.



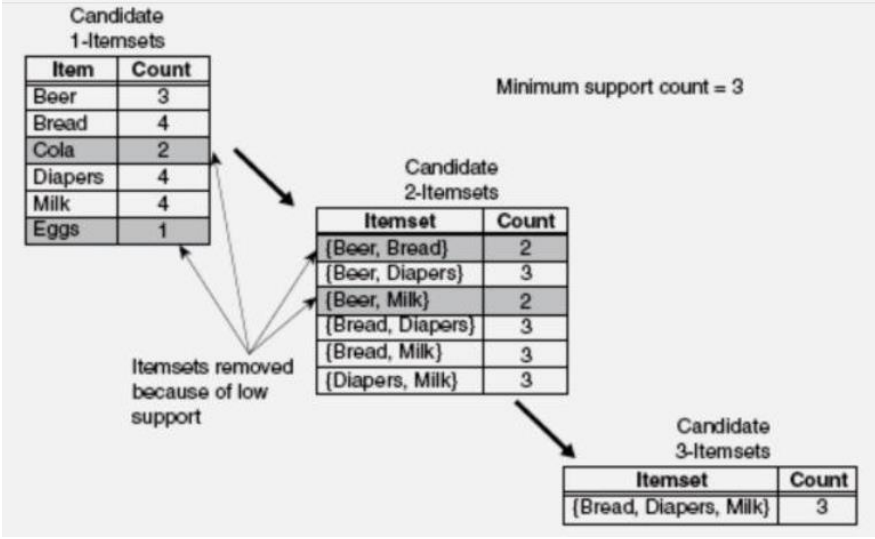
Gambar 17. Hasil penuh contoh pohon Keputusan

- Kesimpulan. Jika tingkat kelembaban (humidity) berada pada tingkat normal, maka aktivitas bermain tenis dapat dilakukan. Namun, jika tingkat kelembaban tinggi, langkah selanjutnya adalah melihat pada outlook. Jika outlook cloudy, maka bermain tenis dapat dilakukan, tetapi jika outlook sunny, aktivitas bermain tenis tidak dapat dilakukan. Ketika outlook windy, komponen lain yang perlu dipertimbangkan. Jika outlook windy adalah false, maka dapat bermain tenis, tetapi jika outlook windy adalah true, maka aktivitas bermain tenis tidak dapat dilakukan.

Bab 7. Association Rule

7.1 Pengenalan Association Rule dengan Apriori

Metode Association Rule atau sering disebut metode Apriori merupakan teknik yang digunakan dalam data mining untuk menemukan pola asosiasi antara item dalam dataset transaksi atau data yang dihasilkan oleh suatu proses bisnis (Gumilang, 2020). Tujuan utama dari metode ini adalah untuk mengidentifikasi hubungan antara item atau itemset yang sering muncul bersama-sama dalam suatu transaksi atau kejadian.



Gambar 18. Ilustrasi penerapan Association Rule
Sumber : (Tank, 2014)

Proses utama dalam metode Apriori dimulai dengan mengidentifikasi semua item yang terdapat dalam dataset dan menghitung frekuensi kemunculan masing-masing item. Kemudian, dilakukan pencarian terhadap itemset yang memiliki frekuensi kemunculan di atas suatu nilai ambang batas yang telah

ditentukan sebelumnya, yang disebut sebagai support threshold. Itemset-itemset ini kemudian digunakan untuk membangun aturan asosiasi (Ardilla et al., 2021). Secara sederhana metode association rule ini dapat dilihat dari keranjang belanja seorang costumer, kita dapat menganalisis keterkaitan satu barang dengan barang lainnya didalam keranjang, dengan tujuan dalam proses bisnis apakah hubungan satu barang dengan barang yang lainnya, dimana peluang untuk dibeli secara bersamaan dapat meningkatkan nilai penjulan pada barang tersebut.

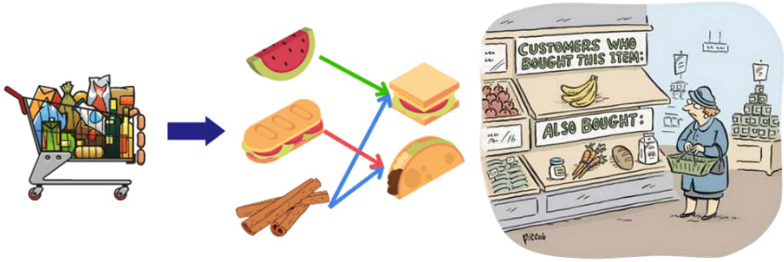
Aturan asosiasi adalah pernyataan yang menyatakan hubungan antara dua atau lebih item atau itemset (Qoniah & Priandika, 2020). Salah satu parameter penting dalam pembentukan aturan asosiasi adalah tingkat kepercayaan (confidence) yang menunjukkan seberapa sering aturan tersebut terbukti benar. Tingkat kepercayaan ini mencerminkan seberapa kuat hubungan antara item atau itemset dalam aturan tersebut.

Metode Apriori sangat berguna dalam berbagai aplikasi seperti analisis pembelian pelanggan di bidang ritel, rekomendasi produk dalam e-commerce, dan pemahaman pola perilaku konsumen secara umum. Dengan memahami pola asosiasi antara item, perusahaan dapat mengoptimalkan strategi pemasaran, meningkatkan kepuasan pelanggan, dan meningkatkan efisiensi operasional.

7.2 Penggunaan Association Rule

Penggunaan Association Rule dalam berbagai bidang telah menjadi sangat signifikan dalam analisis data modern. Salah satu contoh paling umum adalah dalam industri ritel, di mana perusahaan menggunakan Association Rule untuk menganalisis pola pembelian pelanggan. Dengan memahami asosiasi antara produk yang dibeli bersama-sama, toko dapat mengatur penempatan produk dirak, menawarkan rekomendasi produk

yang relevan kepada pelanggan, dan merancang strategi promosi yang lebih efektif.



Gambar 19. Tujuan dari Association Rule Learning
Source: (Admin, 2024)

Dalam e-commerce, Association Rule juga digunakan untuk memberikan rekomendasi produk kepada pelanggan berdasarkan riwayat pembelian mereka atau pola perilaku pengguna lainnya. Misalnya, jika seorang pelanggan sering membeli laptop bersama dengan tas laptop, sistem rekomendasi dapat mengidentifikasi asosiasi ini dan menyarankan tas laptop kepada pelanggan yang sedang mencari laptop.

Di bidang kesehatan, Association Rule juga dapat digunakan untuk menganalisis data medis dan pola penyakit. Misalnya, metode ini dapat digunakan untuk menemukan hubungan antara gejala dan diagnosis penyakit tertentu, sehingga membantu dalam diagnosis dini dan perawatan yang lebih efisien.

Dalam manajemen rantai pasokan, Association Rule dapat membantu perusahaan dalam perencanaan persediaan, penjadwalan produksi, dan pengelolaan persediaan. Dengan menganalisis pola pembelian atau permintaan pelanggan, perusahaan dapat membuat keputusan yang lebih baik tentang kapan dan berapa banyak barang yang harus diproduksi atau dipesan.

Selain itu, Association Rule juga digunakan dalam bidang lain seperti keamanan informasi untuk mendeteksi pola ancaman dan serangan, dalam penelitian biologi untuk menganalisis

interaksi gen, dan dalam analisis teks untuk menemukan pola hubungan antara kata-kata dalam dokumen. Dengan demikian, penggunaan Association Rule sangat luas dan beragam, memberikan kontribusi besar dalam pemahaman pola dan pengambilan keputusan berbasis data di berbagai industri dan bidang.

1. Jenis Jenis Association Rule

a. Algoritma Apriori : Algoritma ini merupakan algoritma yang pertama hadir dalam pembelajaran Asosiasi dan juga merupakan teknik dasar yang paling banyak digunakan. Algoritma Apriori menggunakan teknik pendekatan "bawah ke atas ", dimana model aturan asosiasi dilaksanakan secara bertahap dengan membangun kandidat yang semakin besar nilai frekuensi setiap item.
b. Eclat Algoritma : Teknik yang digunakan dalam algoritma ini fokus pada pencarian aturan asosiasi dengan pendekatan " atas ke bawah ", dimana model aturan ini dilakukan secara iteratif dengan mengekstraksi item yang memilki nilai yang lebih kecil. Algoritma FP-Growth : FP Growth merupakan algoritma pengembangan dari Apriori dan juga merupakan algoritma mining yang paling sering digunakan. Teknik pendekatan yang dijalankan pada FP Growth adalah "Top-down" untuk dapat menemukan pola keterkaian item dalam sejumlah dataset. Dari hasil penggabungan item-item yang secara bersamaan menjadi suatu itemset yang bary dengan nilai yang lebih besar.

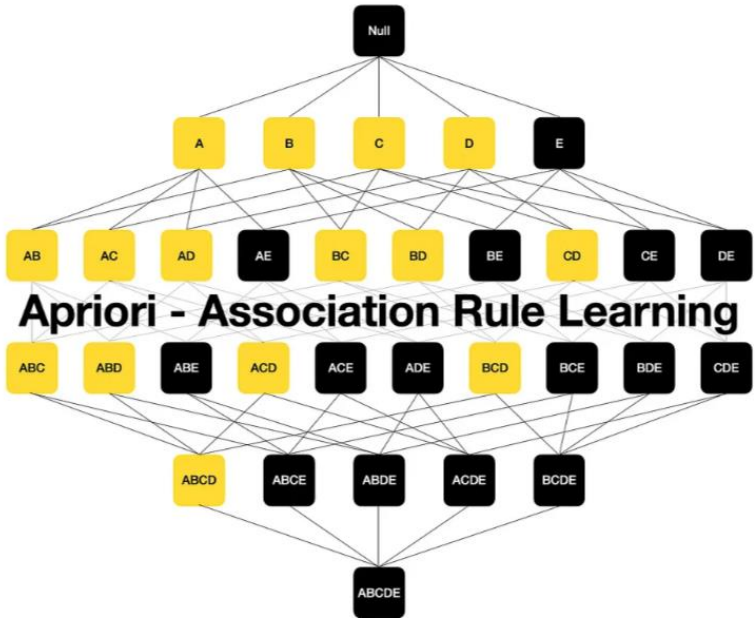
2. Parameter

Parameter dalam Association Rule sangat penting untuk mengontrol kualitas aturan yang dihasilkan dan relevansi dari informasi yang ditemukan. Salah satu parameter utama adalah tingkat support, yang merupakan proporsi transaksi dalam dataset yang mengandung semua item dalam sebuah aturan(luthfi & Amikom, 2009). Tingkat support yang tinggi mengindikasikan bahwa aturan tersebut relevan dalam dataset,

sementara tingkat support yang rendah dapat menunjukkan bahwa aturan tersebut jarang terjadi dan mungkin tidak signifikan.

Selain itu, tingkat kepercayaan (confidence) juga merupakan parameter kunci dalam Association Rule. Tingkat kepercayaan mengukur seberapa sering aturan tersebut terbukti benar dalam dataset. Tingkat kepercayaan yang tinggi menunjukkan bahwa hubungan antara item atau itemset dalam aturan tersebut kuat, sementara tingkat kepercayaan yang rendah menunjukkan bahwa hubungan tersebut lebih lemah atau tidak signifikan.

Selain support dan kepercayaan, terdapat juga parameter tambahan seperti lift, yang mengukur seberapa banyak kemunculan item dalam aturan tersebut dibandingkan dengan kemunculan item secara independen. Lift yang lebih besar dari 1 menunjukkan bahwa kemunculan item-item tersebut bersama-sama lebih sering daripada yang diharapkan secara acak, sehingga aturan tersebut lebih relevan. Dengan mengatur parameter-parameter ini dengan bijak, analisis Association Rule dapat menghasilkan aturan yang berguna dan informatif untuk mendukung pengambilan keputusan.



Gambar 20. Pola keterkaitan setiap item pada aturan Asosiasi

3. Metodelogi

Metodologi Association Rule, terutama yang diimplementasikan melalui algoritma Apriori, memiliki beberapa langkah utama dalam prosesnya. Pertama, langkah persiapan data dilakukan dengan memastikan bahwa dataset telah diproses secara benar, seperti menghilangkan data yang tidak relevan atau duplikat, dan mengubah data transaksional ke dalam format yang sesuai untuk analisis. Kemudian, langkah kedua adalah mengidentifikasi item-item yang ada dalam dataset dan menghitung frekuensi kemunculan masing-masing item.

Setelah langkah persiapan data, langkah ketiga adalah menentukan nilai ambang batas (threshold) untuk tingkat support. Threshold ini mengontrol seberapa sering sebuah itemset harus muncul dalam transaksi untuk dianggap signifikan. Setelah threshold ditentukan, langkah keempat adalah menemukan itemset yang memenuhi threshold support yang

telah ditetapkan. Langkah ini melibatkan pencarian itemset-itemset yang sering muncul bersama dalam transaksi.

Langkah terakhir dalam metodologi Association Rule adalah membangun aturan asosiasi dari itemset-itemset yang ditemukan sebelumnya. Aturan-aturan ini dinyatakan dalam bentuk "jika A maka B", di mana A adalah item atau itemset pada sisi kiri aturan (antecedent) dan B adalah item atau itemset pada sisi kanan aturan (consequent). Aturan asosiasi kemudian dievaluasi berdasarkan parameter seperti tingkat kepercayaan (confidence) dan lift untuk menilai kualitas dan relevansinya dalam dataset. Dengan mengikuti metodologi ini, analisis Association Rule dapat memberikan wawasan berharga tentang pola hubungan antara item dalam dataset transaksional.

Rumus analisa pola frekuensi:

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Total Transaksi}} \qquad \text{Support (A} \cap \text{B)} = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}}$$

Rumus pembentukan aturan asosiatif :

Nilai Confidence dari aturan A → B dapat digunakan rumus sebagai berikut:

$$\text{Confidence} = P(B \mid A) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}}$$

4. Contoh Kasus:

Untuk dapat memahami lebih jauh dari pembelajaran Association Rule berikut contoh data yang dapat diolah, dataset berikut dapat kita selesaikan menggunakan Algoritma Apriori.

Tabel 8. Data Set dari transaksi pelanggan

Transaksi	Item yang dibeli
1	Susu,teh,gula
2	Teh,gula,roti
3	Teh,gula
4	Susu,roti
5	Susu,gula,roti
6	Teh,gula
7	Gula,kopi,susu
8	Gula,kopi,susu
9	Susu,roti,kopi
10	Gula,teh,kopi

Hal yang perlu diketahui:
I adalah himpunan yang dibahas
Contoh: {Susu,kopi,gula,teh,roti)
D adalah himpunan seluruh transaksi yang dibahas
Contoh: {transaksi1 …. Transaksi 10}
K-item set adalah item set yang terdiri dari K buah item yang ada pada I
Item Set Frekuensi adalah jumlah transaksi di I yang mengandung jumlah item set tertentu.
Frekuensi Item set adalah item yang muncul sekurang-kurangnya "sekian" kali di D. "Sekian disimbolkan dengan ϕ yang merupakan batas minimum.

- Langkah 1: Pisahkan masing-masing Item yang dibeli

Tabel 9. Data item dari sejumlah transaksi

Item yang dibeli
Teh
Gula
Kopi
Susu
Roti

- Langkah 2: Menyusun Tabel Tabular

Tabel 10. Data Tabular dari transaksi

Transaksi	Teh	Gula	Kopi	Susu	Roti1
1	1	1	0	1	0
2	1	1	0	0	1
3	1	1	0	0	0
4	0	0	0	1	1
5	0	1	0	1	1
6	1	1	0	0	0
7	0	1	1	1	0
8	0	1	1	1	0
9	0	0	1	1	1
10	1	1	1	0	0

- Langkah 3: Tentukan ϕ, misalnya $\phi = 3$, untuk menentukan frekuensi item set, dilihat untuk transaksi k.
  Langkah 4: Hitung jumlah setiap set
  Untuk k = 1

Tabel 11. Frekuensi dari setiap item pada transaksi

Transaksi	Teh	Gula	Kopi	Susu	Roti
1	1	1	0	1	0
2	1	1	0	0	1
3	1	1	0	0	0
4	0	0	0	1	1
5	0	1	0	1	1
6	1	1	0	0	0
7	0	1	1	1	0
8	0	1	1	1	0
9	0	0	1	1	1
10	1	1	1	0	0
Jumlah	**5**	**8**	**4**	**6**	**4**

Karena semua item lebih besar ϕ dari m maka dibentuk F1 = ={{Teh}, {Gula}, {Kopi}, {Susu}, {Roti}}

Untuk k = 2

Tabel 12. Frekuensi 2 item yang dibeli bersamaan

Transaksi	Teh	Gula		Frek
1	1	1		T
2	1	1		T
3	1	1		T
4	0	0		F
5	0	1		F
6	1	1		T
7	0	1		F
8	0	1		F
9	0	0		F
10	1	1		T
Jumlah	**5**	**8**		**5**

Karena semua item lebih besar ϕ dari m maka dibentuk F2= {{The, Gula}, {dst…}}
Lakukan kombinansi pada setiap item yang telah dihitung pada k =1. sehingga dapat disimpulkan untuk proses k=2, pada tabel berikut.

Tabel 13. Rekapan data kombinasi 2 item yang dibeli bersamaan

Kombinasi	Jumlah
Teh, Gula	5
Teh,Kopi	1
Teh, Susu	1
Teh,Roti	1
Gula, Kopi	3
Gula,Susu	4
Gula,Roti	2
Kopi,Susu	3
Kopi,Roti	1
Susu,Roti	3

Dari perhitungan frekuensi pada tabel 7.6, terdapat beberapa kombinasi item yang memenuhi nilai dari ϕ, sehingga untuk k = 2, diperoleh F2 = {{Teh, Gula}, {Gula, Kopi}, {Gula, Susu}, {Kopi, Susu}, {Susu, Roti}}
Untuk k = 3. Perhitungan frekuensi dilanjutkan untuk mencari kemunculan 3 item yang sering dibeli bersamaan.

Tabel 14. Kombinasi 3 item pada transaksi

Transaksi	Teh	Gula	Kopi		Frek
1	1	1	0		F
2	1	1	0		F
3	1	1	0		F
4	0	0	0		F
5	0	1	0		F
6	1	1	0		F
7	0	1	1		F
8	0	1	1		F
9	0	0	1		F
10	1	1	1		T
Jumlah	**5**	**8**	**4**		**1**

Untuk k=3, semuanya kurang dari ϕ, F3={ }

Tabel 15. Rekapan data kombinasi 3 item dari transaksi

Kombinasi	Jumlah
Teh, Gula,Kopi	1
Teh,Gula,Susu	1
Gula,Kopi, Susu	2
Gula, Susu, Roti	1
Kopi,Susu, Roti	1

Dari tabel tersebut didapat F3= {} sehingga F4, F5 dan seterusnya juga himpunan kosong.

- Langkah 5: Menentukan Rule dengan IF x then Y.
  Dikarenakan perhitungan frekuensi terakhir terbentuk pada F2, maka dapat dibuat Rule sebagai berikut.
  F2={{Teh,Gula},{Gula,Kopi},{Gula,Susu},{Kopi, Susu},{Susu,Roti}}
  Jika membeli Teh maka membeli Gula (The →Gula)
  Jika membeli Gula maka membeli Teh (Gula →Teh)
  ...
  Jika membeli Roti maka membeli Susu (Roti → Susu)
- Langkah 6: Hitung nilai Support dan Confidence

Tabel 16. Hasil penentuan nilai Support dan Confidence

Rule (Aturan)	Support	Confidence
Teh→ Gula	(5/10) x 100% = 50%	(5/5) x 100% = 100 %
Gula → Teh	(5/10) x 100% = 50%	(5/8) x 100% = 62,5%
Gula→ Kopi	(3/10) x 100% = 30%	(3/8) x 100% = 37,5%
Kopi→ Gula	(3/10) x 100% = 30%	(3/4) x 100% = 75%
Gula→Susu	(4/10) x 100% = 40%	(4/8) x 100% = 50%
Susu→Gula	(4/10) x 100% = 40%	(4/6) x 100% = 67 %
Kopi→Susu	(3/10) x 100% = 30%	(3/4) x 100% = 75 %
Susu→Kopi	(3/10) x 100% = 30%	(3/6) x 100% = 50%
Susu→Roti	(3/10) x 100% = 30%	(3/6) x 100% = 50%
Roti→Susu	(3/10) x 100% = 30%	(3/4) x 100% = 75 %

- Langkah 7: Tentukan confidence minimal, misalkan 60%, lalu kalikan support dan confidencenya.

Tabel 17. Nilai kali dari Support dan Confidence

Rule (Aturan)	Support	Confidence	Support x Confidence
Teh→ Gula	(5/10) x 100% = 50%	(5/5) x 100% = 100%	0,5
Gula → Teh	(5/10) x 100% = 50%	(5/8) x 100% = 62,5%	0,3125
Kopi→ Gula	(3/10) x 100% = 30%	(3/4) x 100% = 75%	0,2250
Susu→Gula	(4/10) x 100% = 40%	(4/6) x 100% = 67 %	0,2680
Kopi→Susu	(3/10) x 100% = 30%	(3/4) x 100% = 75 %	0,2250
Roti→Susu	(3/10) x 100% = 30%	(3/4) x 100% = 75 %	0,2250

Dari hasil table 17 diperoleh nilai yang paling tinggi adalah kombinasi Teh dan Gula sebanyak 100%, maka Rule yang dapat digunakan adalah Jika membeli Teh maka akan membeli Gula.

Bab 8. *Clustering*

8.1 Pendahuluan

Clustering adalah metode dalam data mining yang mengelompokkan data ke dalam beberapa kelompok berdasarkan kesamaan karakteristik antar objeknya (Munandar, 2022). Pengelompokan ini dilakukan sedemikian rupa sehingga setiap kelompok memiliki persamaan yang mendasar. Metode data mining ini bersifat tanpa arahan (*unsupervised*), artinya metode ini diterapkan tanpa memerlukan data *training*, *teacher*, atau target *output*. *Clustering* digunakan untuk menganalisis data dengan tujuan menyelesaikan masalah pengelompokan data, atau lebih tepatnya untuk mempartisi dataset menjadi subset. Tujuan dari teknik *clustering* adalah untuk mendistribusikan objek, orang, peristiwa, dan lainnya ke dalam kelompok tertentu, sehingga tingkat keterhubungan antar anggota dalam satu *cluster* sangat kuat, sedangkan keterhubungan antar anggota di *cluster* yang berbeda lebih lemah.

Clustering merupakan proses untuk membagi data yang tidak memiliki label menjadi beberapa kelompok berdasarkan kemiripan data tersebut (Pratama et al., 2022). Misalkan K adalah jumlah kelompok yang ingin dibentuk, C adalah label yang menunjukkan kelompok, dan P adalah *dataset* yang akan dikelompokkan. *Clastering* harus memenuhi kriteria berikut:

$$C_i \neq \Phi, \forall i \ \in \{1,2, \dots ., .K\} \quad (1)$$

$$C_i \cap C_j = \Phi, \forall i \ \neq jandi, j \in \{1,2, \dots ., .K\} \quad (2)$$

$$\cup_{i=1}^{K} Ci = P \quad (3)$$

8.2 Penerapan Metode *Clustering*

Metode *clustering* telah diterapkan dalam berbagai bidang kehidupan manusia, sebagai berikut ini:

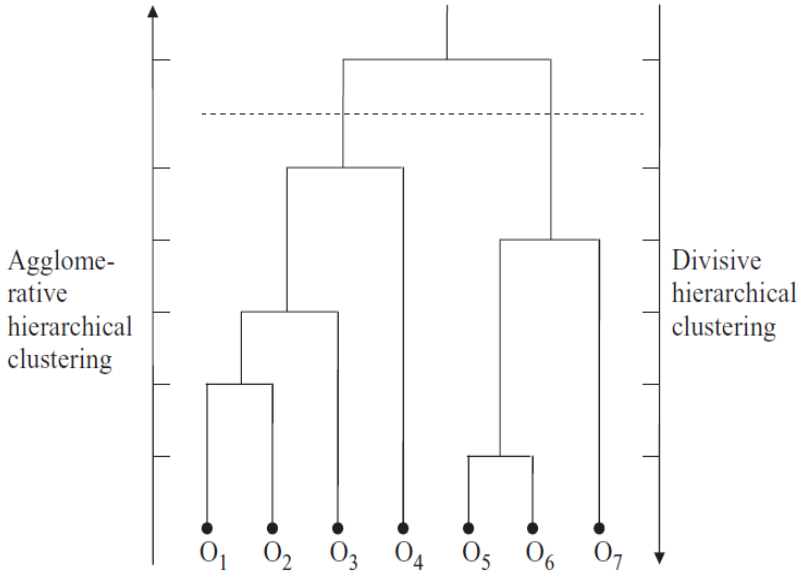
- Medis
  Clustering digunakan dalam mendefinisikan taksonomi dalam biologi, mengidentifikasi fungsi protein dan gen, serta mendiagnosis penyakit dan penanganannya.
- Sosial
  Clustering digunakan untuk analisis pola perilaku, identifikasi hubungan antara budaya yang berbeda pembentukan sejarah evolusi bahasa, dan studi psikologi kriminal.
- Ekonomi
  Clustering diterapkan dalam pengenalan pola pembelian dan karakteristik konsumen, pengelompokan perusahaan, serta analisis tren saham
- Astronomi
  Clustering bermanfaat untuk mengelompokkan bintang dan planet, menyelidiki formasi tanah, mengelompokkan wilayah atau kota, serta digunakan dalam studi sistem Sungai dan gunung.
- Teknik
  Clustering diterapkan dalam pengenalan suara, analisis sinyal radar, kompresi informasi, dan penhilangan *noise*.
- Ilmu Komputer
  Clustring digunakan dalam pengembangan web, analisis basis data spasial, pencarian informasi, koleksi dokumen tekstual, dan segmentasi gambar.

8.3 Jenis-Jenis Pengelompokan

Clustering dapat dibedakan berdasarkan struktur kelompok, keanggotaan data dalam kelompok, dan kekompakan data dalam kelompok (Al Islami, 2024). Berdasarkan struktur kelompok, *clustering* dibagi ke dalam dua kelompok utama, yaitu: *clustering* hierarkis dan partisi. *Clustering* partisi melibatkan pembagian objek-objek data ke dalam kelompok-kelompok yang tidak saling tumpang tindih, sehingga setiap data hanya berada

dalam satu *cluster*. *Clustering* partisi membagi titik data menjadi k *cluster* tanpa mengetahui struktur hierarki. Metode ini membagi set data ke dalam sejumlah kelompok yang tidak saling tumpeng tindih, artinya setiap data hanya masuk ke dalam satu kelompok saja. Contoh metode ini yaitu: K-Means, Fuzzy K-Means, *Mixture Modelling*, dan DBSCAN.

Berdasarkan keanggotaan data dalam kelompok, *clustering* dapat dibagi menjadi dua jenis: eksklusif dan tumpang tindih (Ghaniy & Indriyaningsih, 2020). Dalam kategori eksklusif, sebuah data hanya bisa menjadi anggota satu kelompok dan tidak dapat menjadi anggota kelompok lain. Metode yang termasuk dalam kategori ini adalah K-Means dan DBSCAN. Sebaliknya, dalam kategori tumpang tindih, sebuah data dapat menjadi anggota lebih dari satu kelompok, seperti pada metode Fuzzy K-Means dan *Hierarchical Clustering*. Menurut kekompakan data dalam kelompok, *clustering* terbagi menjadi dua kategori: lengkap dan parsial. Jika semua data dapat bergabung menjadi satu kelompok (dalam konsep partisi), maka semua data dianggap kompak menjadi satu kelompok. Namun, jika ada satu atau beberapa data yang tidak bergabung dalam kelompok mayoritas, data tersebut dianggap sebagai *outlier* atau *noise*, atau dikenal sebagai "*uninterested background*". Beberapa metode yang dapat mendeteksi *outlier* ini termasuk DBSCAN dan K-Means (dengan sejumlah komputasi tambahan).



Gambar 21. *Hierarchical Clustering*
Sumber: (Xu et al., 2010)

Sebaliknya, *clustering* hierarkis adalah metode yang mengelompokan data dengan membentuk sekumpulan *cluster* yang bersarang dalam struktur seperti pohon bertingkat (hierarki), metode ini dibagi menjadi dua pendekatan: agglomeratif dan divisif. Metode agglomerative dimulai dari objek-objek individual, di mana jumlah *cluster* awal sama dengan jumlah objek. Pertama-tama, objek-objek yang paling mirip dikelompokkan, dan kelompok awal ini digabungkan berdasarkan kemiripannya. Akhirnya, ketika kemiripan berkurang, semua subkelompok digabungkan menjadi satu *cluster* tunggal. Sebaliknya, metode hierarkis divisif adalah proses kebalikan dari agglomeratif. Kedua metode ini mengorganisasikan data ke dalam struktur hierarki yang didasarkan pada matriks kedekatan. Hasil dari *clustering* hierarkis biasanya digambarkan dalam bentuk pohon biner atau

dendogram. Akar pohon mewakili keseluruhan dataset dan setiap cabang mewakili titik data. *Clustering* akhir dapat diperoleh dengan memotong dendogram pada level-level tertentu yang sesuai.

8.4 Konsep Dasar *Clustering*

Hasil *clustering* yang baik akan menunjukkan tingkat kesamaan yang tinggi di dalam satu kelompok dan tingkat kesamaan yang rendah antara kelompok-kelompok yang berbeda. Kesamaan ini diukur secara numerik antara dua objek. Nilai kesamaan antar objek akan lebih tinggi jika kedua objek yang dibandingkan memiliki kemiripan yang besar, dan sebaliknya. Kualitas hasil *Clustering* sangat tergantung pada metode yang digunakan. Dalam *clustering*, ada empat jenis data yang dikenal:

- Variabel berskala interval
- Variabel biner
- Variabel nominal, ordinal, dan rasio
- Variabel dengan tipe lainnya

1. Metode *clustering* juga harus mampu mengevaluasi kemampuannya sendiri dalam menemukan pola tersembunyi pada data yang dianalisis. Ada berbagai metode yang dapat digunakan untuk mengukur kesamaan antara objek-objek yang dibandingkan. Salah satunya adalah *weighted Euclidean Distance. Euclidean Distance* menghitung jarak antara dua titik dengan mengetahui nilai dari masing-masing atribut pada kedua titik tersebut. Berikut adalah formula yang digunakan untuk menghitung jarak dengan *Euclidean Distance.*
2. Syarat *Clustering*

   Beberapa persyaratan dan tantangan yang harus dipenuhi oleh suatu algoritma klasterisasi (Mining, 2006):

   a. Skalabilitas metode *clustering* harus mampu menangani data dalam jumlah besar. Saat ini, data dalam jumlah besar sangat umum digunakan dalam berbagai bidang, misalnya

pada basis data. Basis data dengan ukuran besar bisa berisi lebih dari jutaan objek, bukan hanya ratusan.

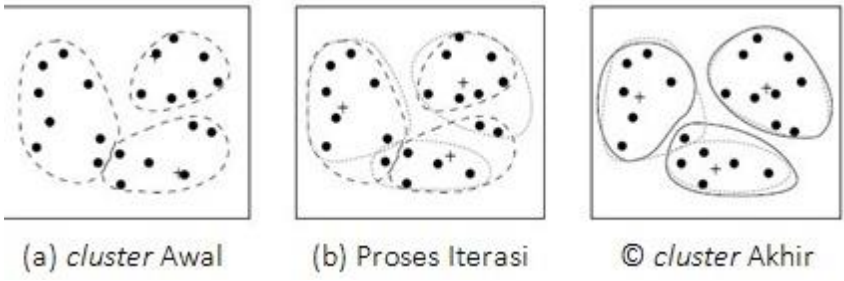
b. Kemampuan analisis beragam bentuk data algoritma *clustering* harus dapat diterapkan pada berbagai jenis data, seperti data nominal, ordinal, atau kombinasi keduanya.
c. Menemukan *cluster* dengan bentuk yang tidak terduga, banyak algoritma *clustering* menggunakan metode Euclidean atau Manhattan yang cenderung menghasilkan *cluster* berbentuk bulat. Namun, hasil klasterisasi bisa memiliki bentuk yang tidak teratur dan berbeda satu sama lain. Oleh karena itu, algoritma *clustering* harus mampu menganalisis *cluster* dengan berbagai bentuk.
d. Kemampuan untuk menangani noise data tidak selalu dalam kondisi sempurna. Terkadang terdapat data yang rusak, tidak jelas, atau hilang. Algoritma *clustering* harus mampu menangani data yang rusak ini.
e. Sensitivitas terhadap perubahan *input***,** perubahan atau penambahan data *input* dapat menyebabkan perubahan pada *cluster* yang sudah ada. Algoritma *clustering* harus memiliki sensitivitas yang cukup untuk mengakomodasi perubahan tersebut tanpa menyebabkan perubahan drastis pada hasil klasterisasi.
f. Kemampuan *clustering* untuk data berdimensi tinggi sekelompok data dapat memiliki banyak dimensi atau atribut. Oleh karena itu, algoritma *clustering* harus mampu menangani data dengan jumlah dimensi yang tinggi.
g. Interpretasi dan kegunaan hasil dari *clustering* harus dapat diinterpretasikan dengan jelas dan berguna dalam konteks yang relevan.

8.5 Metode K-Means

Algoritma K-Means adalah salah satu metode *clustering* yang paling sederhana dan umum digunakan. Hal ini karena K-Means mampu mengelompokkan data dalam jumlah besar dengan waktu komputasi yang cepat dan efisien. Algoritma ini pertama kali diusulkan oleh MacQueen pada tahun 1967 dan kemudian dikembangkan oleh Hartigan dan Wong pada tahun 1975. Tujuan K-Means adalah untuk membagi M titik data dalam N dimensi menjadi sejumlah k *cluster*, di mana proses *clustering* dilakukan dengan meminimalkan jumlah kuadrat jarak antara data dan pusat *cluster* masing-masing (berbasis *centroid*).

Dalam penerapannya, algoritma K-Means membutuhkan tiga parameter yang ditentukan oleh pengguna, yaitu jumlah *cluster* k, inisialisasi *cluster*, dan sistem pengukuran jarak. Biasanya, K-Means dijalankan secara independen dengan inisialisasi yang berbeda-beda, menghasilkan *cluster* akhir yang berbeda karena algoritma ini cenderung mengelompokkan data menuju lokal minima. Salah satu cara untuk mengatasi masalah lokal minima adalah dengan menjalankan algoritma K-Means untuk nilai K yang diberikan dengan beberapa partisi awal yang berbeda dan kemudian memilih partisi dengan kesalahan kuadrat terkecil (Jain, 2010). K-Means adalah salah satu algoritma klasterisasi dengan metode partisi (*partitioning method*) yang berbasis titik pusat (*centroid*), berbeda dengan algoritma k-Medoids yang berbasis objek.

Algoritma K-Means secara iteratif meningkatkan variasi nilai dalam setiap *cluster*, di mana setiap objek ditempatkan ke dalam kelompok terdekat yang dihitung dari titik tengah *cluster*. Setelah semua data ditempatkan dalam *cluster* terdekat, titik tengah baru ditentukan. Proses penentuan titik tengah dan penempatan data dalam klaster diulangi sampai nilai titik tengah dari semua *cluster* yang terbentuk tidak lagi berubah (Mining, 2006).



Gambar 22. Proses *Clustering* Objek Menggunakan Metode K-Means (EDY IRWANSYAH, S.T., n.d.)

Pada dasarnya, algoritma k-means melakukan dua proses utama: pendeteksian lokasi pusat *cluster* (*centroid*) dan pencarian anggota untuk setiap *cluster*. Proses klasterisasi dimulai dengan mengidentifikasi data yang akan dikelompokka, X_{ij} (i=1 ,..., m;j=1,...,m), di mana n adalah jumlah data yang akan dikelompokkan dan m adalah jumlah variabel. Pada iterasi awal, pusat setiap klaster ditentukan secara acak, C_{kj} (k=1,...,k; j=1,...,m). Kemudian, jarak antara setiap data dengan setiap pusat *cluster* dihitung menggunakan formula Euclidean. Data akan menjadi anggota dari *cluster* ke-k jika jaraknya ke *centroid* ke-k adalah yang paling kecil dibandingkan dengan jarak ke pusat *cluster* lainnya. Proses dasar algoritma k-means dapat dilihat di bawah ini:

1. Tentukan jumlah *cluster* yang ingin dibentuk dan tetapkan pusat *cluster* k. Menggunakan jarak euclidean kemudian hitung setiap data ke pusat *cluster*.

$$d_{ij} = \sqrt{\sum_{k=1}^{n}\{x_{ik} - x_{jk}\}^2} \qquad (4)$$

Keterangan:
d_{ij} = Jarak antara data ke – i dan data ke – j
n = dimensi data
x_{ik} = Koordinat data ke – i pada dimensi k

x_{jk} = koordinat data ke – j pada dimensi k

2. Kelompokkan data ke dalam *cluster* berdasarkan jarak terpendek menggunakan persamaan berikut:

$$Min\ \sum_k^k = d_{ik} = \sqrt{\sum_j^m \left(C_{ij} - C_{kj}\right)^2} \qquad (5)$$

3. Hitung pusat *cluster* baru menggunakan persamaan berikut:

$$C_{kj} = \frac{\sum_{i=1}^{p} x_{ij}}{p} \qquad (6)$$

Dengan:

X_{ij} *€ Cluster* ke – k

p = banyaknya anggota *cluster* ke – k

4. Ulangi langkah dua sampai empat hingga tidak ada lagi data yang berpindah ke *cluster* lain.

8.6 Implementasi Algoritma K-Means

Algoritma k-means membagi data ke dalam sejumlah k-buah cluster yang telah ditentukan sebelumnya. Ada beberapa metode yang dapat digunakan untuk menghitung jarak, termasuk *euclidean distance*, *manhattan distance*, dan *chebyshev distance*. Adapun penjelasan terkait masing-masing metode perhitungan jarak sebagai berikut:

1. Euclidean Distance

Formula untuk menghitung jarak antara dua titik menggunakan *euclidean distance* dalam satu, dua, dan tiga dimensi secara berturut-turut ditunjukan pada persamaan (7), (8), dan (9) sebagai berikut:

$$\sqrt{(x-y)^2} = |x-y| \qquad (7)$$

$$d(p,q) = \sqrt{(p_1-q_2)^2 + (p_2-q_2)^2} \qquad (8)$$

$$d(p,q) = \sqrt{(p_1-q_2)^2 + (p_2-q_2)^2 + (p_3-q_3)^2} \qquad (9)$$

Mantahan Distance

$$d1(p,q) = \|p-q\|1 = \sum_{i=1}^{n} |p_i - q_i| \qquad (10)$$

2. Chebisev

Untuk menghitung jarak menggunakan *chebyshev distance*, caranya adalah dengan mengambil nilai maksimum dari perbedaan setiap koordinat dimensinya. Jika dinyatakan dalam persamaan matematika, *chebyshev distance* dapat dilihat pada persamaan (11).

$$D_{cheb}(p,q) = max(|p_i - q_i|) \quad (11)$$

Pada algoritma k-means atau algoritma *clustering* yang mengelompokkan data berdasarkan pada titik pusat *cluster* (*centroid*) terdekat dengan data tersebut. Tujuan dari k-mens adalah untuk mengelompokkan data dengan cara memaksimalkan kemiripan data dalam satu *cluster* dan meminimalkan kemiripan data antara *cluster* yang berbeda. Ukuran kemiripan yang digunakan dalam *clustering* ini adalah fungsi jarak. Oleh karena itu, pemaksimalan kemiripan data dicapai dengan menemukan jarak terpendek antara data dan titik *centroid*.

Tahap awal dalam proses *clustering* data menggunakan algoritma k-means adalah pembentukan titik awal *centroid* c_j. Biasanya, titik awal *centroid* ini dihasilkan secara acak. Jumlah *centroid* c_j yang dihasilkan sesuai dengan jumlah *cluster* yang telah ditentukan di awal. Setelah *k centroid* terbentuk, jarak antara setiap data x_i dengan setiap *centroid* c_j dihitung, dinotasikan sebagai d (x_i, c_j). Terdapat beberapa ukuran jarak yang digunakan sebagai ukuran kemiripan data, salah satunya adalah jarak euclidean.

8.7 Metode K-Medoid

Algoritma K-Medoid juga dikenal sebagai Algoritma Partitioning Around Medoid (PAM). Metode k-medoid dikembangkan oleh Leonard Kaufman dan Peter J. Rousseeuw

pada tahun 1987 (Wira et al., 2019). Metode K-Medoid mirip dengan Metode K-Means karena keduanya termasuk dalam kategori Metode *Partitioning*. Metode *Partitioning* adalah teknik pengelompokan data ke dalam sejumlah *cluster* tanpa adanya struktur hierarki di antara *cluster-cluster* tersebut (Andriana et al., 2022). Metode K-Medoid memiliki keunggulan dibandingkan dengan Metode K-Means, terutama dalam kinerja yang lebih optimal ketika jumlah data yang digunakan relatif kecil. Algoritma ini menggunakan objek dari kumpulan data untuk mewakili sebuah *cluster*. Objek yang dipilih untuk mewakili *cluster* disebut medoid.

K-Medoid adalah teknik berbasis objek yang menggunakan objek nyata sebagai representatif *cluster*. Dalam metode ini, kita memilih objek sebenarnya untuk mewakili *cluster*, bukan menggunakan nilai rata-rata dari objek dalam *cluster* sebagai titik referensi. *Partitioning Around Medoids* (PAM) adalah salah satu algoritma K-Medoid pertama. Strategi dasar dari algoritma ini adalah:

1. Menemukan objek representatif untuk setiap klaster.
2. Mengelompokkan setiap objek yang tersisa dengan objek representatif yang paling mirip.

Secara iteratif menggantikan salah satu medoid dengan non-medoid selama "kualitas" pengelompokan tetap terpenuhi (Tiwari & Singh, 2012).

Metode K-Medoids merupakan bagian dari *clustering partitioning*. Algoritma ini efisien pada dataset kecil. Langkah awalnya adalah mencari titik yang paling representatif (medoids) dalam dataset dengan menghitung jarak antara kelompok dalam semua kemungkinan kombinasi dari medoids, sehingga jarak antara titik dalam suatu *cluster* kecil sedangkan jarak antara titik antar *cluster* besar (Damanik et al., 2019).

Perbedaan mendasar antara kedua metode ini adalah Metode K-Medoids menggunakan objek sebagai perwakilan

(medoid) sebagai pusat *cluster* untuk setiap *cluster*, sementara Metode K-Means menggunakan nilai rata-rata (*mean*) sebagai pusat *cluster*. Metode K-Medoids memiliki kelebihan dalam menangani kelemahan Metode K-Means yang sensitif terhadap *noise* dan *outlier*, di mana objek dengan nilai yang besar dapat menyimpang dari distribusi data. Kelebihan lainnya adalah hasil klasterisasi tidak tergantung pada urutan masukan data.

8.8 Implementasi Algoritma K-Means

Langkah-langkah Metode K-Medoids *clustering* adalah sebagai berikut (Sundari et al., 2019):

1. Inisialisasi pusat *cluster* sebanyak k (jumlah *cluster*).
2. Alokasikan setiap data (objek) ke *cluster* terdekat menggunakan persamaan ukuran jarak *Euclidean Distance* dengan persamaan:

$$d_{ij} = \sqrt{\sum_{a=1}^{p}\left(x_{ia} - x_{ja}\right)^2} = \sqrt{\left(x_i - x_j\right)'\left(x_i - x_j\right)} \quad (12)$$

   di mana i = 1,...,n; j = 1,...,n, dan p adalah jumlah variabel. V adalah matriks varian kovarian.
3. Pilih secara acak objek dalam masing-masing klaster sebagai kandidat medoid baru.
4. Hitung jarak setiap objek dalam masing-masing klaster dengan kandidat medoid baru.
5. Hitung total deviasi (S) dengan menghitung nilai total jarak baru - total jarak sebelumnya. Jika S < 0, maka tukar objek dengan data klaster untuk membentuk kumpulan k objek baru sebagai medoid.
6. Ulangi langkah 3 sampai 5 hingga tidak ada perubahan medoid, sehingga diperoleh klaster beserta anggota klaster masing-masing.

8.9 Hierarchical Clustering

Hierarchical Clustering adalah metode analisis yang berupaya membentuk hirarki kelompok data. Terdapat 2 (dua) strategi utama dalam pengelompokan, yaitu: Agglomerative (*Bottom-Up*) dan Devisive (*Top-Down*). *Agglomerative* adalah strategi yang dimulai dengan menempatkan setiap objek sebagai sebuah *cluster* tersendiri (*atomic cluster*) dan selanjutnya menggabungkan *atomic cluster* – *atomic cluster* tersebut menjadi *cluster* yang lebih besar secara bertahap. Proses ini terus berlangsung sampai akhirnya semua objek bergabung menjadi satu *cluster* besar atau hingga mencapai batasan kondisi tertentu. Metode Pengelompokan Hierarki Aglomeratif:

1. Hitung matrik jarak antar data
2. Ulangi langkah 3 dan 4 hingga hanya satu kelompok yang tersisa
3. Gabungkan dua kelompok terdekat berdasarkan parameter kedekatan yang ditentukan (*single linkage*, *complete linkage*, *average linkage*)
4. Perbarui matrik jarak antar data untuk merepresentasikan kedekatan diantara kelompok baru dan kelompok yang masih tersisa
5. Selesai

Membentuk matrik jarak, contoh dengan Manhattan *Distance*:

$$D_{man}(x, y) = \sum_{j=1}^{d} |x_j - y_j| \qquad (13)$$

atau menggunakan Euclidian *Distance*:

$$D_{man}(x_2, x_1) = \sum_{j=1}^{d} |x_{2j} - y_{1j}|^2 \qquad (14)$$

Beberapa metode pengelompokan secara hierarki aglomeratif:

Single Linkage (Jarak Terdekat)

$d_{uv} = min\{d_{uv}\}, d_{uv} \in D$ (15)

Complete Lingkage (Jarak Terjauh)

$d_{uv} = max\{d_{uv}\}, d_{uv} \in D$ (16)

Average Lingkage (Jarak rata-rata)

$d_{uv} = average\{d_{uv}\}, d_{uv} \in D$ (17)

8.10 Kesimpulan

Clustering dalah teknik analisis data yang digunakan untuk mengelompokkan objek-objek ke dalam kelompok atau klaster berdasarkan kesamaan tertentu. Metode ini sangat bermanfaat dalam berbagai bidang seperti bisnis, medis, astronomi, ilmu komputer, dan sosial, karena mampu mengungkapkan struktur tersembunyi dalam data yang tidak berlabel. Algoritma *clustering*, seperti K-Means, memungkinkan pemrosesan data dalam jumlah besar dengan efisiensi waktu komputasi yang tinggi.

Algoritma K-Means mengelompokkan data berdasarkan titik pusat *cluster* (*centroid*) dan terus memperbarui posisi *centroid* hingga tidak ada perubahan signifikan dalam *cluster*. Proses dimulai dengan memilih *centroid* secara acak, menghitung jarak setiap data ke centroid, dan mengelompokkan data berdasarkan *centroid* terdekat. Meskipun sederhana dan populer, algoritma ini memiliki tantangan seperti ketergantungan pada inisialisasi awal dan kecenderungan menuju lokal minima. Masalah ini dapat diatasi dengan menjalankan algoritma beberapa kali dengan inisialisasi berbeda atau menggunakan metode inisialisasi yang lebih canggih.

Di sisi lain, algoritma K-Medoids, atau *Partitioning Around Medoid* (PAM), menggunakan objek nyata dari *dataset* sebagai pusat *cluster*. Metode ini lebih robust terhadap *outlier* dan *noise* dibandingkan K-Means karena menggunakan objek nyata sebagai medoid, bukan rata-rata data. Algoritma ini iteratif dan terus mengganti medoid dengan objek lain untuk meningkatkan

kualitas clustering. K-Medoids lebih cocok untuk dataset yang lebih kecil karena intensitas komputasinya lebih tinggi dibandingkan K-Means.

Selain K-Means dan K-Medoids, *Hierarchical Clustering* adalah metode penting lainnya yang membentuk hirarki kelompok data. Metode ini tidak membagi data ke dalam jumlah *cluster* yang telah ditentukan sebelumnya tetapi membentuk hirarki dari cluster kecil hingga satu *cluster* besar, menggunakan teknik seperti *single linkage*, *complete linkage*, dan *average linkage*. *Hierarchical Clustering* sangat berguna untuk analisis eksploratif karena memberikan pandangan yang mendalam tentang struktur data.

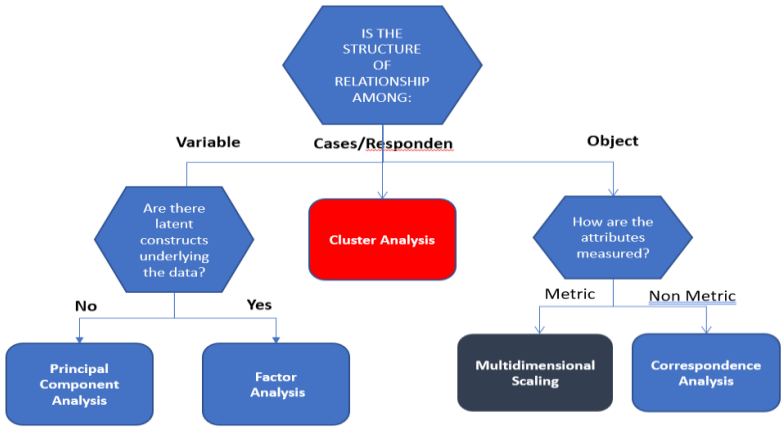
Keberhasilan klasterisasi sangat tergantung pada pemilihan metode yang tepat dan kemampuan algoritma untuk menangani berbagai jenis data, termasuk data dengan bentuk yang tidak terduga, data yang mengandung *noise*, dan data berdimensi tinggi. Kualitas hasil *clustering* diukur berdasarkan tingkat kesamaan dalam satu *cluster* dan perbedaan antar *cluster*, dengan berbagai metrik jarak yang digunakan untuk mengevaluasi kesamaan, seperti *Euclidean Distance* dan *Chebyshev Distance*.

Dengan kemampuan untuk mengidentifikasi pola tersembunyi dan segmentasi yang efektif, klasterisasi menjadi alat yang sangat berguna dalam analisis data, memberikan wawasan yang mendalam dan mendukung pengambilan keputusan yang lebih baik di berbagai aplikasi praktis.

Bab 9. Analisis Klaster

9.1. Pengertian

Sebelum membahas tentang pengertian secara keseluruhan, berikut adalah tampilan analisis klaster dalam struktur hubungan dalam Analisis multivariat lainnya



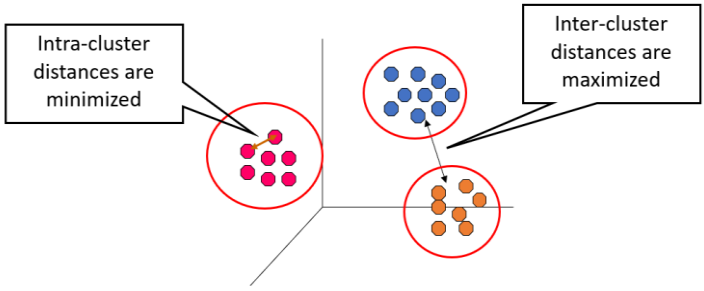
Gambar 23. Posisi Analisis Klaster dalam Struktur Hubungan antar Variabel Penelitian

Dari gambar di atas dapat terlihat bahwa analisis klaster adalah bagian dari analisis multivariat dengan struktur hubungan yang akan dibentuk adalah antar cases/responden, atau dengan kata lain jika kita ingin menganalisis case/responden dalam data penelitian kita (apakah dengan cara mengelompokkan data, membuat profiling data berdasarkan case/responden dalam data) maka analsis klaster adalah pilihannya.

Cluster analysis adalah analisis yang bertujuan untuk menempatkan sekumpulan *cases*/responden ke dalam dua atau lebih grup (klaster) berdasarkan *cases*/responden atas dasar

berbagai karakteristik. Tujuan *cluster analysis* adalah agar cases/responden di dalam grup mirip (atau berhubungan) satu dengan lainnya, dan berbeda dengan *cases* dalam grup lainnya. Sehingga tingkat kemiripan menjadi kata kunci dalam analisis klister

Semakin besar tingkat kemiripan di dalam grup (homogenitas) dan semakin besar tingkat perbedaan di antara grup (heterogenitas), maka semakin baik pengelompokan yang dihasilkan.



Gambar 24. Tingkat Kemiripan dalam Analisis Klaster

Banyak sekali implementasi dari Analisis Klaster, misalnya, Bidang pemasaran: segmentasi pasar. Dalam Database: clustering data log web untuk mendapatkan group dengan pola akses yang sama, atau bahkan dalam Data analytic, misal spatial data mining. Berikut adalah beberapa contoh pertanyaan penelitian yang dapat diselesaikan menggunakan analisis klaster

Berapa banyak kelompok mahasiswa yang terbentuk berdasarkan pertimbangan dalam memilih perguruan tinggi?
Berapa banyak kelompok mahasiswa berdasarkan prestasi mereka dalam berbagai mata kuliah yang diikuti?.

9.2 *Data, Assumptions, and Sampling*

Berikut adalah jenis data dan asumsi data yang akan dilibatkan dalam analisis klister. Umumnya data berskala

numerik (interval/rasio), namun dalam beberapa kasus data kategori (nominal/ordinal) juga dapat dianalisis.

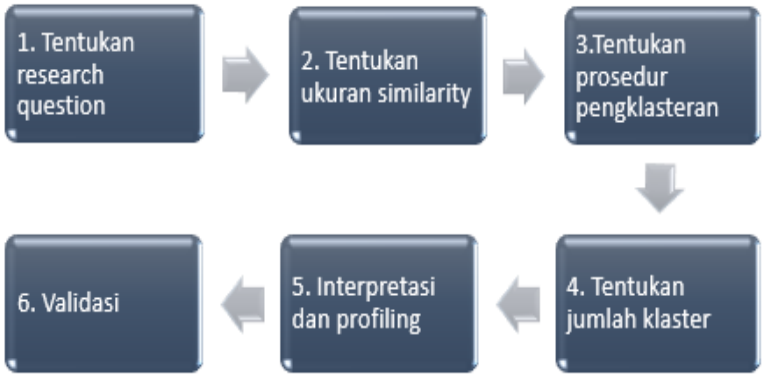
- Data numerik yang memiliki skala yang berbeda, sebaiknya dilakukan standardisasi lebih dahulu.
- Data variabel tidak saling berkorelasi
- Data harus terbebas dari *outlier* (data pencilan)

Ukuran sampel harus cukup untuk mewakili seluruh kelompok di dalam populasi.

- Jenis data, asumsi dan ukuran sampel tidak akan dijelaskan lebih rinci dalam bahasan kali ini karena memerlukan ruang atau bab tersendiri. Silakan membuka sumber sumber referensi terkait dengan skala pengukuran data, analisis data eksplorasi dan ukuran dan teknik sampling.

9.3 Prosedur Analisisi Klaster dan Interpretasi

Dalam analisis Klaster terdapat prosedur/tahapannya, berikut adalah tahapan atau prosedur dalam melakukan analisis klaster:



Gambar 25. Tahapan Analisis Klaster

Berikut adalah penjelasan atas setiap tahapan tersebut pertanyaan penelitian, diantaranya:

- Keperluan Pengelompokan (*taxonomy description*), yaitu analisis cluster dipakai untuk keperluan pengelompokan sekumpulan cases/responden/objek secara empiris
- Profiling Group, yaitu mengetahui karakteristik khusus kelompok-kelompok data dari data set yang diteliti

Misal:

- Menentukan berapa jenis kelompok mahasiswa berdasarkan pertimbangan dalam memilih sebuah Perguruan Tinggi.
- Konsumen yang bagaimana yang akan dibidik sebagai target produk yang akan ditawarkan

1. Ukuran Similarity

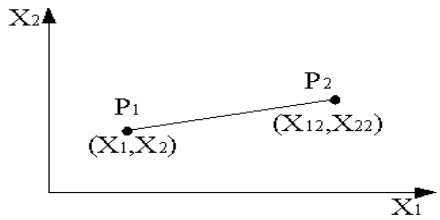
Similarity adalah ukuran kemiripan dari sebuah objek atau entitas atau dalam hal ini adalah kemiripan dari setiap responden atau cases. Terdapat dua jenis ukuran similarity

a. Asosiasi atau korelasi antar objek/case/responden
b. Kedekatan atau jarak antar objek/case/responden.

Terdapat beberapa ukuran jarak yang digunakan yaitu:

1) Euclidean Distance. Adalah jarak berupa akar dari jumlah perbedaan antar objek yang dikuadratkan.

/ $$d(P_1, P_2) = \sqrt{(X_{12} - X_{11})^2 + (X_{22} - X_{21})^2}$$



Gambar 26. Euclidean Distance

2) Cityblock/Manhattan Distance. Adalah jumlah dari nilai mutlak selisih pasangan titik.

3) Chebychev Distance. Adalah harga maksimum dari city block pemilihan ukuran similarity yang berbeda akan menghasilkan output yang berbeda pula, sehingga disarankan menggunakan ukuran yang lebih dari satu kemudian tentukan dengan menggunakan jarak mana yang terbaik

Contoh:

a) Mengukur Similarity

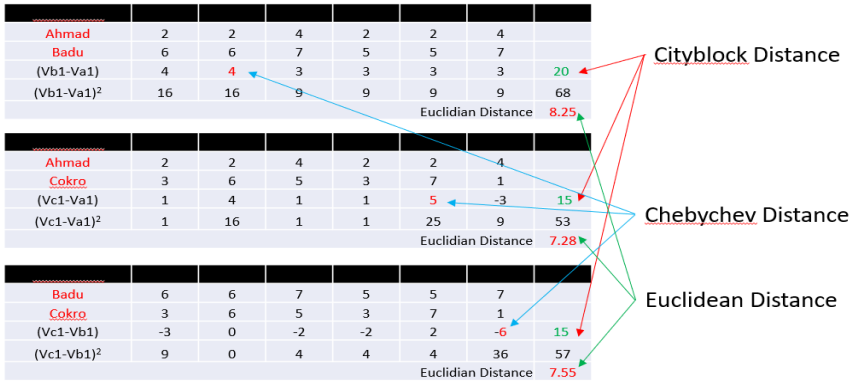
Terdapat 3 orang yang diberikan pertanyaan atas 6 variabel, dengan data sebagai berikut:

Tabel 18. Penguukuran Similarity

Responden	Var1	Var2	Var3	Var4	Var5	Var6
Ahmad	2	2	4	2	2	4
Badu	6	6	7	5	5	7
Cokro	3	6	5	3	7	1

Tabel 19. Contoh Asosiasi/korelasi

Korelasi	Ahmad	Badu	Cokro
Ahmad	1		
Badu	0.866025	1	
Cokro	-0.4055	-0.40134	1



Gambar 27. Ilsutrasi Perhitungan Distance

Dapat terlihat bahwa berdasarkan Euclidean Distance, Ahmad dan Cokro mempunyai karakteristik yang paling mirip dibandingkan dengan yang lain, sementara berdasarkan Cityblock: Ahmad-Cokro dan Badu-Cokro, dan berdasarkan Chebychev: Ahmad-Badu yang paling dekat

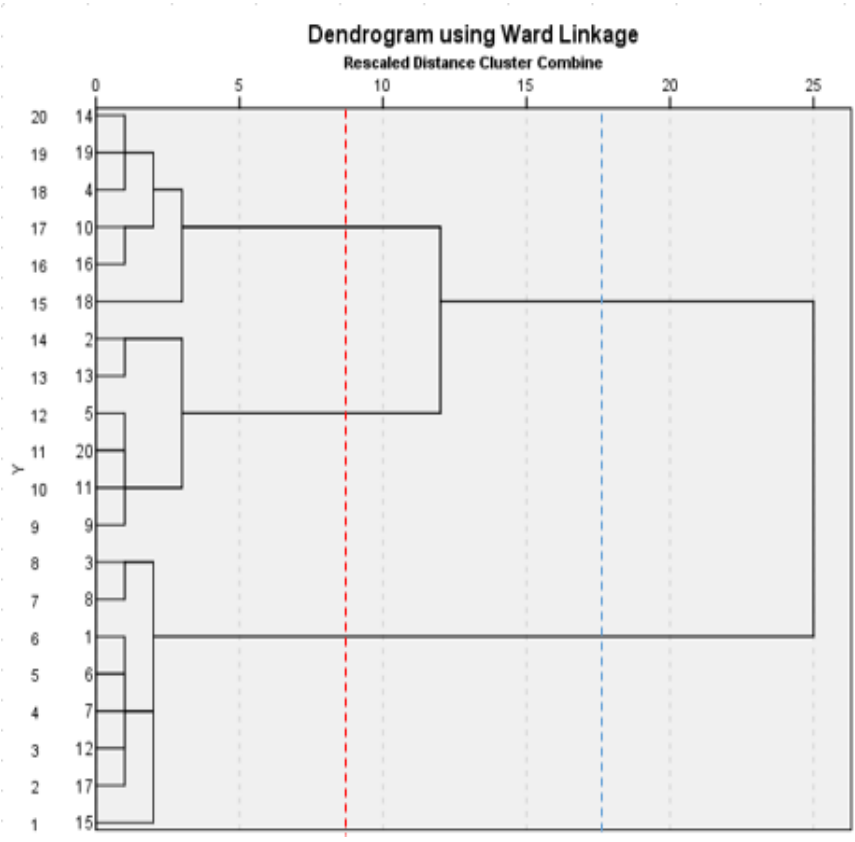
2. Prosedur Pengklasteran

Sebelum dilakukan pengklasteran dengan metode tertentu, maka diwajibkan untuk mendeteksi outlier, Jika terdapat data outlier maka case/objek/responden yang bersangkutan (dimana data oulier tersebut berada) harus dikeluarkan dari data yang akan dianalisis.

Cara mengidentifikasi outlier dapat dilakukan dengan menggunakan profil diagram atau menggunakan analisis Box whisker Plot (dibahas dalam modul stastistika deskriptif). Jika terdapat variabel yang berbeda skala data maka diperlukan standardisasi.

Cara menentukan klaster:

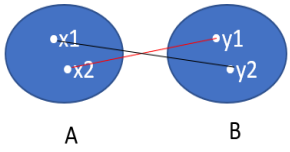
- Hirarkis
- Banyak kelompok belum diketahui
- Output berupa dendogram



Gambar 28. Dendogram

Langkah pengklasteran hirarkis:

a. Menentukan Ukuran similarity yang akan digunakan (dijelaskan pada poin 2 ukuran similarity)
b. Pilih metode hirarki (*Linkage, Ward Method*, atau *Centroid*)
c. Linkage: jarak antar dua klaster yang terbentuk berdasarkan jarak objek (terjauh atau terpendek)

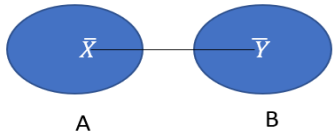


$d_{AB} = \min\{s_{xy}\}$ → single lingkage

$d_{AB} = \max\{s_{xy}\}$ →complete linkage

dimana Sx,y adalah jarak dua data x dan y masing-masing dari *cluster* A dan B.

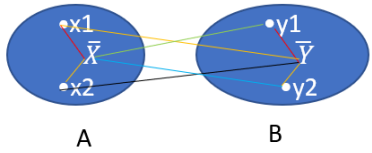
d. Centroid: jarak antara dua pusat klaster. dengan cara menghitung rata-rata jarak semua anggota dalam tiap klaster kemudian dihitung jarak kedua rata-rata tersebut



$$d_{AB} = s_{\bar{x},\bar{y}}$$

dimana $s_{\bar{x},\bar{y}}$ adalah jarak dua pusat *cluster* A dan B.

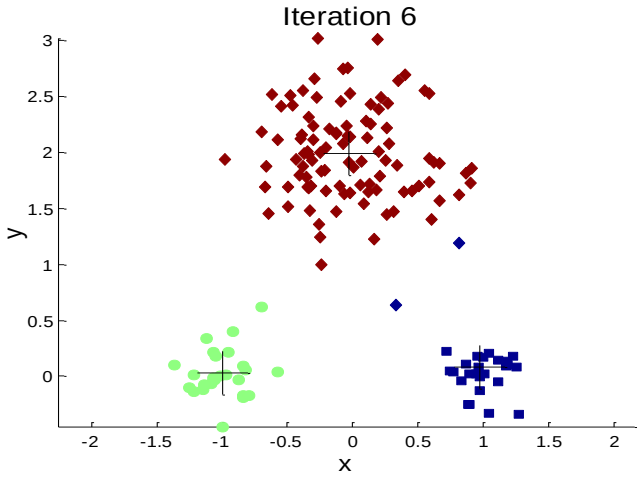
e. Ward Method: Variansi jarak antar objek dalam klaster seminimal mungkin



$$d_{AB} = \frac{n_A . n_B \, s_{AB}^2}{n_A + n_B}$$

dimana s_{AB}^2 adalah jarak *cluster* A dan B dengan menggunakan formula Centroid.

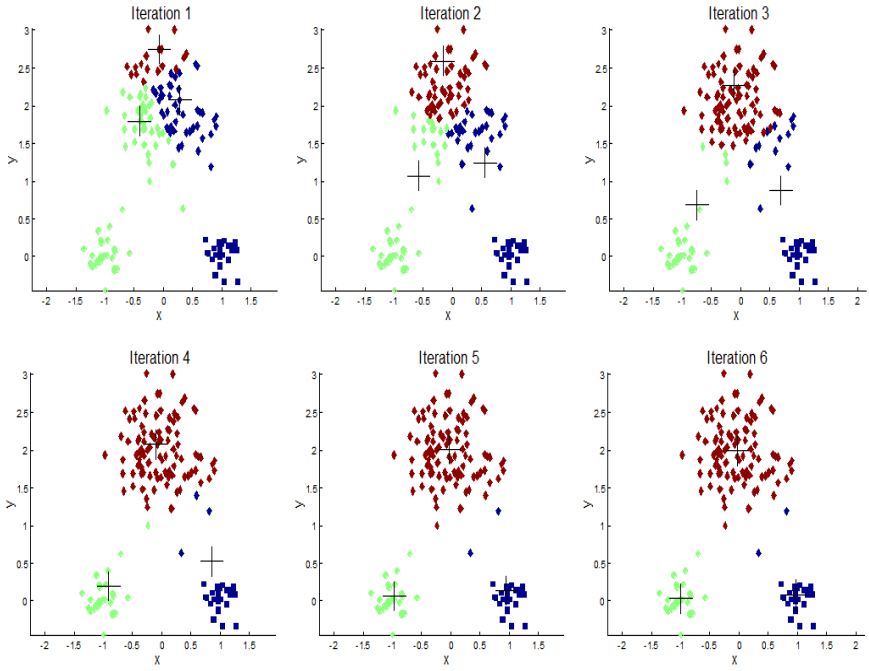
3. Buat Dendogram
   - Non Hirarki, misal K-Means Klastering
   - Banyak kelompok ditentukan terlebih dahulu (diketahui)
   - Dapat menangani sampel besar
   - Output: anggota kelompok dan centroid



Gambar 29. Hasil K-Means Klastering

Langkah pengklasteran K-Means (non-hirarki):

a. Tentukan banyaknya K Centroid (titik pusat klaster awal) sebagai langkah awal (inisiasi)
b. Bentuk K klaster dengan cara memasukan setiap objek kedalam klaster terdekat
c. Hitung kembali centroid dari klaster yang terbentuk pada langkah ke-2
d. Ulangi langkah 2 dan 3 sampai dengan centroid yang terbentuk konvergen (tidak berubah)

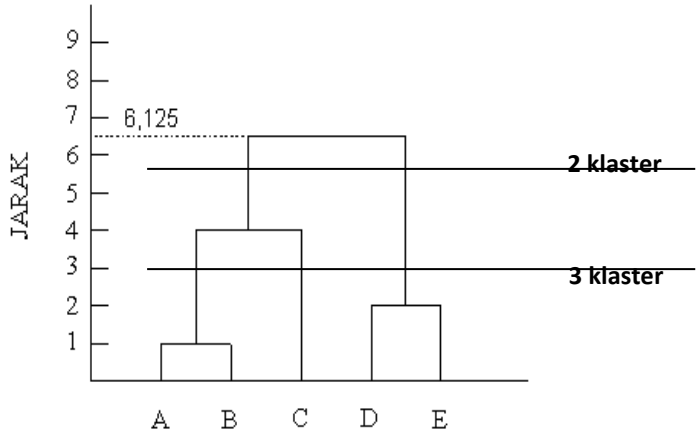


Gambar 30. Ilsutrasi K-Means Klastering

4. Penentuan Jumlah Klaster

Pada prinsipnya penentuan jumlah klaster bersifat subjektif tidak eksak. Penentuan jumlah klaster yang terbentuk ditentukan sepenuhnya oleh peneliti berdasarkan dendogram atau hasil K-means klaster (non hirarki). Terdapat beberapa panduan dalam memilih banyaknya klaster.

- Terdapat kosep, teori atau model dan pertimbangan praktis dari pakar (metode non hierarki)
- Dapat menggunakan Grafik koefesien dan penggunaan garis potong di dendogram (dalam metode hirarki)



Gambar 31. Penentuan Jumlah Klaster

5. Interpretasi dan profiling

Berikut adalah sedikit panduan bagaimana melakukan interpretasi dan profiling dalam analsiis klaster.

a. Hitung rata-rata tiap setiap klaster pada setiap variabel yang diteliti. Rata-rata ini disebut dengan centroid.
b. Identifikasi nilai centroid yang tinggi disetiap klaster untuk tiap variabel.
c. Beri nama klaster yang terbentuk berdasarkan karakteristik dengan centroid yang tinggi pada tahap sebelumnya.

6. Validasi

Tidak ada alat statistik yang dapat melakukan validasi hasil analisis klaster. Namun demikian berikut adalah beberapa panduan dalam melakukan validasi analisis klaster

a. Buat data sampel menjadi dua (split sampel), lakukan analisis klaster untuk dua sampel tersebut, jika hasilnya relatif sama, maka analisis klaster yang dilakukan sudah baik.
b. Jika dalam melakukan profiling hasil klaster didapatkan hasil yang spesifik, mudah diinterpretasikan maka analisis klaster sudah baik

c. Lakukan analisis klaster dengan metode yang berbeda, atau metode sama namun menggunakan similarity yang berbeda, jika hasil klaster relatif sama maka dikatakan analisis klaster yang dilakukan sudah baik

9.4 Latihan/Praktik Analisis Klaster

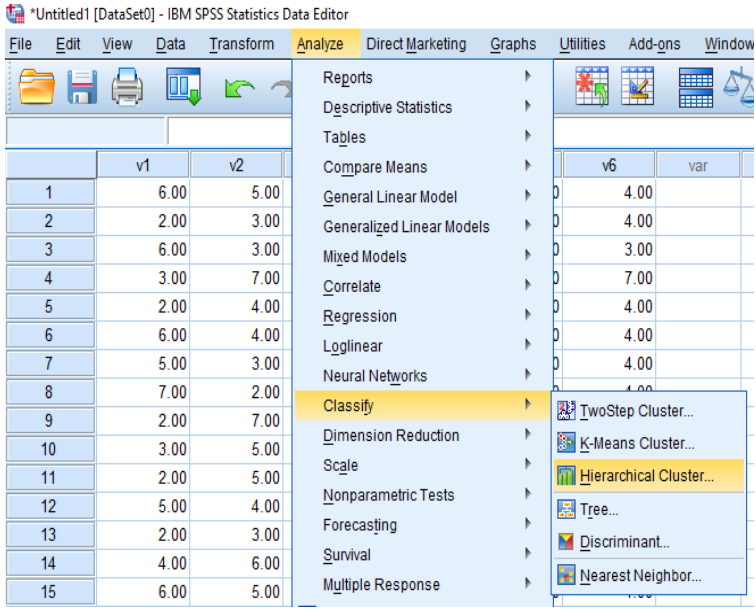
Bagian ini akan menjelaskan bagaimana melakukan Klaster Analisis menggunakan SPSS.

Menentukan berapa jenis kelompok mahasiswa berdasarkan pertimbangan dalam memilih sebuah Perguruan Tinggi. Responden diminta memberikan terhadap 6 pertanyaan
Saya lebih memilih PT kecil berkualitas daripada PT besar tapi kualitas diragukan. Saya banyak mendengarkan nasihat teman dalam memilih perguruan tinggi. Saya mempelajari seluruh informasi tentang perguruan tinggi sebelum menentukan pilihan Bagi saya kampus adalah tempat terbaik untuk memperluas pergaulan. Pendidikan yang saya terima di PT sudah sesuai dengan biaya yang dikeluarkan. Saya lebihbanyak menghabiskan waktu dikampus bersama teman-teman. Seluruh jawaban dibuat dalam bentuk skala likert dengan skala 1-7, dan dibagikan kepada 20 orang responden.

Tabel 20. Data sampel yang akan dianalisis

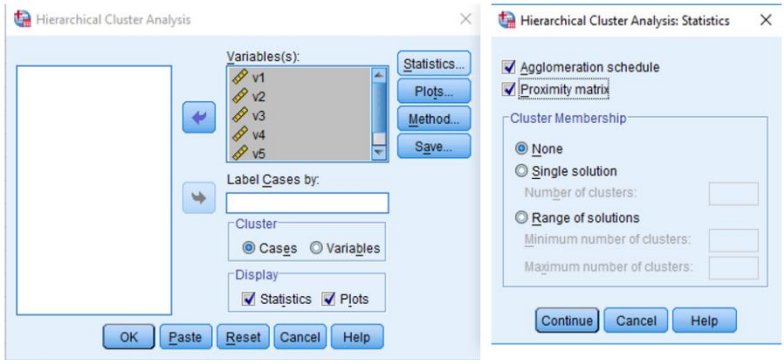
Resp	Var1	Var2	Var3	Var4	Var5	Var6
1	6	5	6	3	3	4
2	2	3	2	4	5	4
3	6	3	6	4	2	3
4	3	7	4	5	2	7
5	2	4	2	2	7	4
6	6	4	6	3	3	4
7	5	3	6	3	3	4
8	7	2	7	4	2	4
9	2	7	2	3	7	3
10	3	5	3	6	4	6
11	2	5	2	3	5	3
12	5	4	5	4	2	4
13	2	3	2	5	4	4
14	4	6	4	6	3	6
15	6	5	4	2	1	4
16	3	5	4	6	5	7
17	4	4	7	2	2	5
18	3	7	2	6	4	3
19	4	6	3	6	3	6
20	3	4	3	4	7	3

1. Input data ke dalam SPSS
   Klik analyze>classify> hierarchical cluster



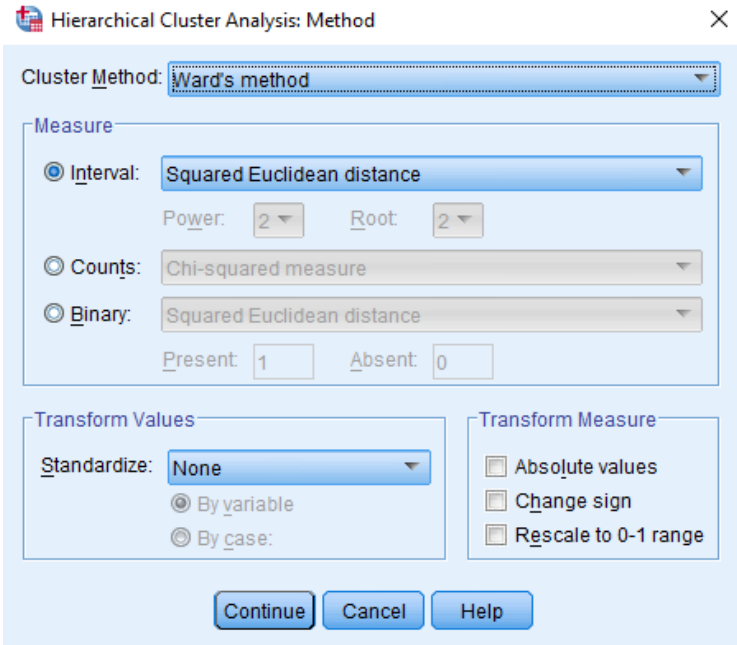
Gambar 32. Input ke SPSS

2. Pindahkan seluruh variable ke dalam kolom variables
   Centang “agglomeration schedule” dan “proximity matrix” klik kontinu, lalu OK



Gambar 33. Pengisian variable

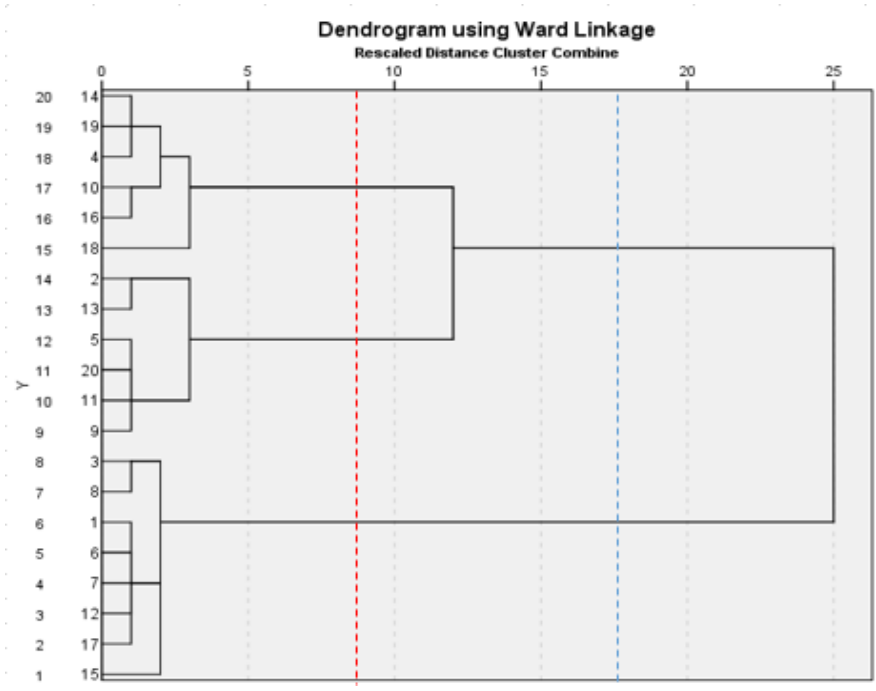
3. Klik plots, aktifkan “dendogram”,
   Klik kontinu, lalu klik button “Method”, pilih “ward methods” klik kontinu. Pada kotak dialog utama, Klik OK.



Gambar 33. Seting dendogram

Dari hasil eksekusi dengan SPSS didapat output dendogram sebagai berikut:

- Penentuan jumlah klaster ditentukan berdasarkan pertimbangan peneliti (2 klaster dengan garis bru; 3 klaster dengan garis merah, dst)

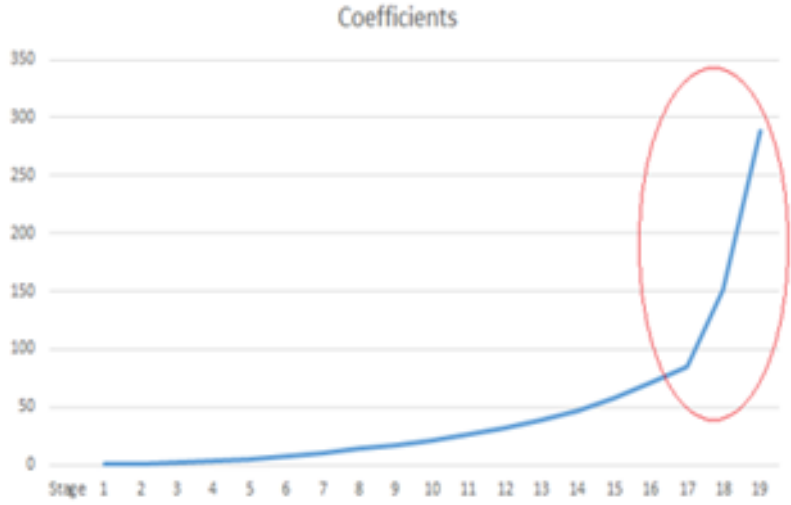


Gambar 34. Penentuan kluster dengan dendogram

Dari hasil eksekusi dengan SPSS didapat output koefesien sebagai berikut:

Agglomeration Schedule

Stage	Cluster Combined		Coefficients	Stage Cluster First Appears		Next Stage
	Cluster 1	Cluster 2		Cluster 1	Cluster 2	
1	14	19	.500	0	0	9
2	1	6	1.000	0	0	7
3	2	13	2.000	0	0	17
4	10	16	3.500	0	0	13
5	7	12	5.500	0	0	7
6	3	8	7.500	0	0	15
7	1	7	10.250	2	5	11
8	5	20	13.750	0	0	10
9	4	14	17.250	0	1	13
10	5	11	21.083	8	0	12
11	1	17	26.633	7	0	14
12	5	9	32.550	10	0	17
13	4	10	39.450	9	4	16
14	1	15	47.150	11	0	15
15	1	3	57.275	14	6	19
16	4	18	70.708	13	0	18
17	2	5	85.292	3	12	18
18	2	4	150.792	17	16	19
19	1	2	288.550	15	18	0



Gambar 35. Hasil eksekusi angglomeration schedule

Berdasarkan grafik koefesien agglomeration, dapat dilihat bahwa grafik naik tajam setelah stage ke 16, atau pada saat stage 17, 18 dan 19 (tiga stage) sehingga dapat diambil ketetapan bahwa banyaknya klaster adalah 3. Setelah penentuan banyaknya klaster selesai dilakukan maka selanjutnya adalah menentukan profiling. dari hasil sebelumnya dapat ditetapkan bahwa terdapat 3 klaster. Dengan anggota masing masing klaster dapat dilihat berdasarkan dendogram.

Tabel 21. Hasil klasterisasi

Klaster I	Klaster II	Klaster III
14	2	3
19	13	8
4	5	1
10	20	6
16	11	7
18	9	12
		17
		15

Kemudian setiap klaster diberi nama sesuai dengan karakteristik yang dominan pada klaster tersebut. Hal ini dapat dianalisis dengan menghitung centroid maksimum tiap klaster.

Tabel 22. Menghitung centroid maksimum tiap klister

Resp	Var1	Var2	Var3	Var4	Var5	Var6
14	4	6	4	6	3	6
19	4	6	3	6	3	6
4	3	7	4	5	2	7
10	3	5	3	6	4	6
16	3	5	4	6	5	7
18	3	7	2	6	4	3
Rata-rata	3.33	6.00	3.33	5.83	3.50	5.83

Resp	Var1	Var2	Var3	Var4	Var5	Var6
2	2	3	2	4	5	4
13	2	3	2	5	4	4
5	2	4	2	2	7	4
20	3	4	3	4	7	3
11	2	5	2	3	5	3
9	2	7	2	3	7	3
Rata-rata	2.17	4.33	2.17	3.50	5.83	3.50

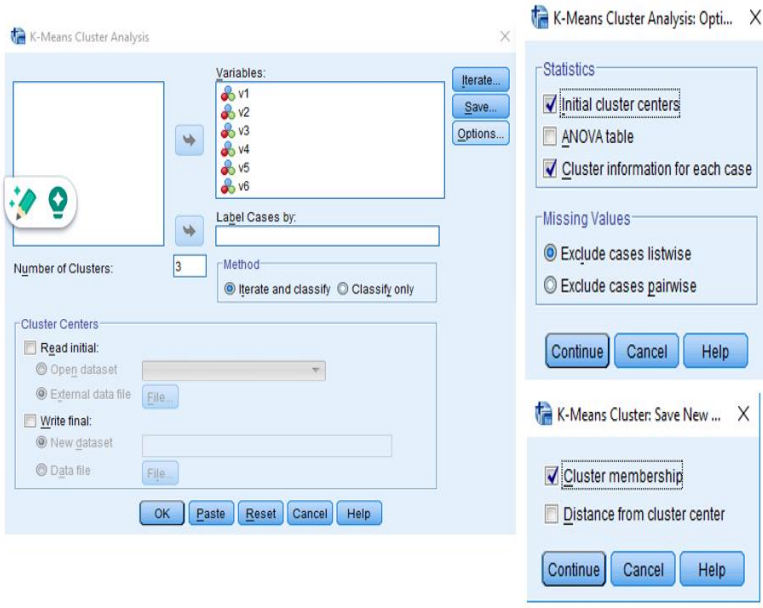
Resp	Var1	Var2	Var3	Var4	Var5	Var6
3	6	3	6	4	2	3
8	7	2	7	4	2	4
1	6	5	6	3	3	4
6	6	4	6	3	3	4
7	5	3	6	3	3	4
12	5	4	5	4	2	4
17	4	4	7	2	2	5
15	6	5	4	2	1	4
Rata-rata	5.63	3.75	5.88	3.13	2.25	4.00

Dapat terlihat dari hasil pengerjaan di atas Klaster I didominasi oleh V2, V4, dan V6, klaster ini bisa saja diberi nama sebagai "klaster Sosial", Klaster II didominasi oleh V5 bisa saja klaster ini diberi nama "klaster Entrepreneur" dan Klaster III didominasi oleh V1 dan V3 bisa saja diberi nama "Klaster Rasional".

Berikut akan dijelaskan hasil SPSS dengan analis Klaster K-Means

- Dari menu Analyze > classify> K-mean Cluster, pindahkan semua variabel ke dalam kolom "variables", isi dengan "3" klaster. Klik option "initial cluster centre", dan "cluster information for each case", klik "continu", klik "OK"

Pada button “Save” aktifkan “Cluster membership”, Klik “continu”, klik “OK”



Gambar 36. Aktifasi Cluster membership

Dari sekian banyak output, berikut akan ditampilkan output intinya yaitu sebagai berikut:

Cluster Membership

Case Number	Cluster	Distance
1	3	1.516
2	2	1.740
3	3	1.596
4	1	2.421
5	2	2.007
6	3	.893
7	3	1.244
8	3	2.655
9	2	3.005
10	1	1.236
11	2	1.302
12	3	1.431
13	2	2.774
14	1	1.093
15	3	2.837
16	1	2.279
17	3	2.509
18	1	3.346
19	1	.928
20	2	1.833

Final Cluster Centers

	Cluster		
	1	2	3
v1	3.33	2.17	5.63
v2	6.00	4.33	3.75
v3	3.33	2.17	5.88
v4	5.83	3.50	3.13
v5	3.50	5.83	2.25
v6	5.83	3.50	4.00

Distances between Final Cluster Centers

Cluster	1	2	3
1		4.673	5.388
2	4.673		6.268
3	5.388	6.268	

Number of Cases in each Cluster

Cluster	1	6.000
	2	6.000
	3	8.000
Valid		20.000
Missing		.000

Gambar 37. Hasil final klusterisasi

Dari output disamping dapat terlihat bahwa dengan menentapkan K=3, maka secara otomatis proses pengklasteran dilakukan oleh SPSS sehingga didapatkan member setiap klaster. Selain member setiap klaster, terlihat juga centroid setiap klaster dan jarak antar centroid dari ke tiga klaster yang terbentuk yaitu Jumlah member klaster I=6, klaster II=6 dan Klaster III=8 mahasiswa. Profiling ketiga klaster yang terbentuk berdasarkan cara pada hierarchical klaster sebelumnya.

Langkah selanjutnya adalah melakukan validasi. Dari berbagai teknik validasi, dapat digunakan pendekatan subjektif yaitu tiap klaster sangat mudah untuk dikelompokan dan hasil pemberian nama berdasarkan karakter variabel yang diajukan serta member setiap klaster sangat jelas, maka hasil analisis klaster sudah dianggap baik.

Bab 10. Predictive Mining

10.1 Pendahuluan

Perusahaan atau organisisi dipenuhi atau dibanjiri dengan data Sebagian besar di sebagian besar bidang (*Most Field)*. Namun disayangkan sebagian besar data berharga ini yang telah terkumpul dengan menghabiskan tenaga, waktu, biaya dolar bagi perusahaan dalam mengumpulkan dan menyusunnya tersebut tidak dipergunakan secara optimal. Hal ini salah satunya dikarenakan belum cukup tersedia Sumber daya manusia atau seornag analis yang terlatih, terampil dalam menerjemahkan semua data ini menjadi pengetahuan, dan memprediksikan kebermanfaatan data tersebut. Untuk itulah saat ini perlu pemanfaatan Teknologi memanfaakan *Machine Learning* dan pola penambangan data (*Data Mining*) yang tepat sehingga dapat melakukan analisis kebutuhan Perusahaan atau organissai mengunakan Predictic Analytics.

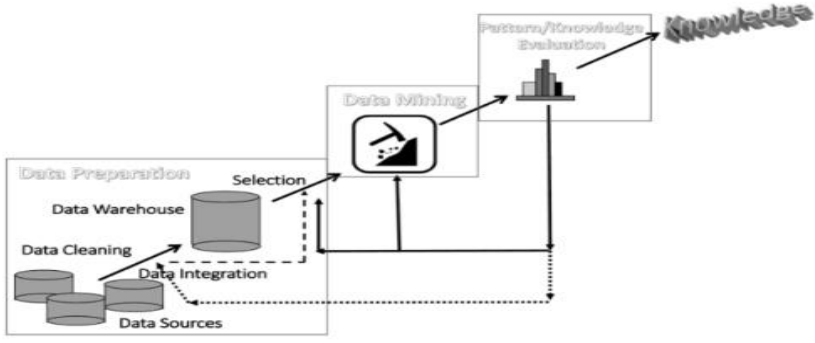
10.2 *Data Mining* dan Prediksi Peramalan

menemukan pola-pola menarik, model, dan pemanfaatan jenis-jenis pengetahuan lainnya berdasarkan suatu dataset bervolume besar (Han et al., 2022). Sumber himpunan data bervolume besar dapat berbentuk dalam berbagai jenis yang terdiri dari data terstruktur dan data tidak terstruktur. Data terstruktur dapat berbentuk basis data relasional, data multidimensional (*data cubes*), dan data matrices. Sedangkan, data tidak terstruktur dapat berbentuk data teks dan data multimedia, seperti data audio, data gambar, dan data video (Han et al., 2022). Dengan bentuk data yang berbeda-beda, dataset diolah dengan berbagai teknik dengan tujuan mengekstraksi

informasi yang berharga. Proses ini melibatkan identifikasi pola yang mencerminkan hubungan atau tren yang mungkin tidak terlihat secara langsung. Pola ini dapat menciptakan wawasan baru dan berperan krusial dalam penentuan Keputusan (Rahayu et al., 2024).

Prediksi peramalan atau prediksi adalah proses memperkirakan keadaan di masa depan dengan menggunakan data masa lalu. Setelah mengetahui proyeksi keadaan di masa depan, prediksi dapat digunakan untuk mengambil tindakan yang tepat untuk menghadapi. Oleh karena itu, prediksi merupakan alat yang efektif untuk perencanaan dan pengambilan keputusan yang lebih baik dalam berbagai konteks, baik di bidang ekonomi, ilmu pengetahuan, maupun penjualan.

Penambangan data (*Data Mining*) adalah proses menemukan pola dan tren yang berguna dalam kumpulan data yang besar. Analisis prediktif (*Predictive Analytics*) adalah proses mengekstraksi informasi dari kumpulan data besar untuk membuat prediksi dan perkiraan tentang hasil di masa depan (Larose & Larose, 2014). Tahapan Data Mining Menurut (Mining, 2006), terdapat empat langkah utama dalam melakukan data mining yaitu:



Gambar 38. Tahapan Data Mining (Mining, 2006)

Data preparation, proses persiapan dan pengolahan data mentah sebelum data tersebut dapat digunakan untuk analisis lebih lanjut. Sebelum memulai tahapan ini, data yang ingin dieksplorasi perlu dikumpulkan terlebih dahulu, dan selanjutnya melalui proses data preparation. Terdapat 4 proses dalam data preparation yang meliputi data cleaning, data integration, data transformation, dan data selection.

Data cleaning adalah tahap untuk membersihkan data dan harus bebas dari hal inkonsistensi data dan noisy data seperti duplikasi data, data yang tidak relevan, dan data anomali. Data integration merupakan tahap penggabungan data antara satu tempat dan tempat yang lain. Selanjutnya, transformasi data merupakan tahap penggabungan suatu data ke dalam bentuk yang sudah sesuai untuk dieksplorasi dengan melakukan serangkaian operasi. Salah satu operasi yang dilakukan yaitu penggabungan data atau kolom. Penggabungan data atau kolom bertujuan untuk menggabungkan data atau kolom dari beberapa sumber atau tabel yang berbeda ke dalam satu sumber data. Terakhir, data selection merupakan tahap penyeleksian data yang relevan untuk dianalisis lebih lanjut.

Data mining, tahap penerapan suatu metode untuk mencari pola atau model informasi yang menarik. Pattern/model evaluation, pola atau model informasi yang direpresentasikan menjadi wawasan, perlu diukur untuk menentukan apakah pola yang ditemukan berguna atau tidak.

Knowledge presentation, pihak – pihak terkait dapat memahami wawasan yang dihasilkan berdasarkan visualisasi yang mudah dipahami. Penambangan data (*DataMining*) menjadi lebih luas (*Widesread*) Dimana setiap hari perusahaan akan memberdayakan data yang ada untuk melakukan, mengungkap pola dan tren yang menguntungkan Perusahaan berdasarkan database yang ada diperusahaan.

10.3 Hubungan Data Mining dan Machine Learning

Metode Data Mining, atau umumnya disebut sebagai salah satu kerangka teknik *Machine Learning*, adalah tahapan untuk melakukan prediksi terhadap suatu keadaan dengan mengembangkan model dari data (Anwar & Permana, 2021). Dalam proses ini, data yang dianalisis dipergunakan untuk melatih model yang dapat mengidentifikasi pola dan tren yang kemudian digunakan untuk membuat prediksi atau pengambilan keputusan di masa depan. Data Mining digunakan untuk menggali informasi dari data, yang kemudian digunakan untuk melatih model Machine Learning. Model Machine Learning kemudian digunakan untuk membuat prediksi atau pengambilan keputusan di masa depan. Dengan demikian, Data Mining dan Machine Learning saling melengkapi dalam memproses data dan menciptakan nilai tambah melalui pemahaman dan prediksi berdasarkan pengetahuan yang ditemukan.

Menurut (Kotu & Deshpande, 2014), Analisis prediktif (*Predictive*) dan teknik Penambangan Data (*Data Mining*) meliputi: Analisis Data Eksplorasi, Visualisasi, Pohon Keputusan, Induksi Aturan (rule induction), k-Nearest Neighbors, Naïve Bayesian, Jaringan Syaraf Tiruan (*Artificial Neural Networks*), Mesin Vektor Dukungan (*Support Vector machines*), Model Ensemble, *Bagging*, *Boosting*, *Random Forests*, Regresi linier, Logistik regresi, Analisis Asosiasi menggunakan Apriori (*Association analysis using Apriori*) dan FP Growth, Pengelompokan K-Means (*K-Means clustering*), Pengelompokan Berbasis Kepadatan (*Density based clustering*), Peta Pengorganisasian Mandiri (*Self Organizing Maps*), Penambangan Teks, Peramalan Rangkaian Waktu (*Text Mining, Time series forecasting*), Deteksi Anomali (*Anomaly detection*), dan Pemilihan Fitur (*feature selection*).

Data Mining dan Analysis Predictive telah menjadi alat penting bagi perusahaan atau organisasi melalui proses

mengumpulkan, menyimpan, dan memproses data sebagai bagian dari operasinya, terutama bagi pengguna bisnis, analis data, analis bisnis, profesional intelijen bisnis (*business intellegence*) dan pergudangan data (*data warehousing*) perlu menerapkan *data mining* (Kotu & Deshpande, 2014).

10.4 Kelompok Data Mining

Menurut (Han et al., 2022), kelompok Data Mining terbagi menjadi tujuh bagian, yaitu:

1. Deskriptif, mengidentifikasi ciri-ciri kumpulan data yang ingin dianalisis untuk mengenal perilaku dari data.
2. Prediktif, menggunakan data untuk menghasilkan prediksi atau perkiraan suatu nilai tertentu.
3. Multidimensional Data Summarization, meringkas data multidimensional atau data cubes untuk mengurangi kompleksitas data multidimensional yang masif.
4. The Mining of Frequent Patterns, Associations, dan Correlations, mencari dan menemukan pola serta hubungan antar variabel.
5. Classification, proses untuk menemukan model yang memiliki karakteristik yang serupa dalam suatu kelompok atau kelas. Klasifikasi memprediksi label kategorik yang terdiri dari data diskret dan data yang tidak berurutan.
6. Regression, serupa dengan klasifikasi tetapi regresi memprediksi nilai data numerik yang tidak ada, bukan untuk memprediksi ke dalam label kategorik.
7. Cluster Analysis, mengelompokkan suatu kelas berdasarkan kesamaan atribut yang ditentukan. Outlier Analysis, mengetahui data anomali dalam suatu kejadian.

Streamlit library atau pustaka yang digunakan untuk membangun situs web berbasis Python yang dirancang untuk memudahkan penyebaran hasil analisis, menciptakan interaksi

yang kompleks, serta memvisualisasikan model-model machine learning terkini (Richards, 2021).

Data preparation merupakan proses untuk mengumpulkan data, membersihkan data dari hal yang tidak relevan, mengubah bentuk data, serta melakukan seleksi fitur yang bertujuan untuk analisis data. Tahap penelitian ini difokuskan pada solusi untuk mengatasi masalah persediaan dan penjualan produk makanan beku, sehingga memungkinkan pelaku bisnis untuk menganalisis dan membuat keputusan secara efektif. Data preparation memiliki empat subtahap, antara lain pengumpulan data, data cleaning, data transformation, dan data selection. Namun untuk penambangan gambar menurut (Diyasa & Saputra, 2022), lebih tepat mengunakan Algoritma CNN karena algoritma ini sangat cocok untuk proses pendeteksian objek pada gambar.

10.5 Contoh penerapan *Predictive Data Mining*

Berikut contoh penerapan dalam predictive dan data mining secara bertahap pada contoh kasus penjualan, sebagai berikut:

1. Pengumpulan Data

Data yang digunakan pada penelitian ini berasal dari data penjualan dari Perusahaan. Diperkirakan data mulai Januari – September 2023 yang disusun dalam satu file Google Sheet. Data penjualan tersebut dapat digunakan untuk memberikan estimasi jumlah penjualan produk dalam kurun waktu tertentu. Data yang berhasil dikumpulkan dalam satu file adalah sebanyak 24 sheet berisi penjualan produk di bulan tertentu dan 5 sheet berisi informasi mengenai harga produk makanan beku dan harga promo produk pada acara atau kegiatan promosI seerti contoh kegiatan dibawah ini.

Tabel 23. Data Sheet dan kegiatan

Nama Sheet	Kegiatan
EID DELIVERY 2023	Detail mengenai pengiriman produk pada Idul Fitri tahun 2021
DELIVERY 2023	mengenai pengiriman produk pada Natal

2. Data Cleaning

Data cleaning merupakan tahap berikutnya setelah menyeleksi sheet. Pembersihan data bertujuan untuk beberapa hal, di antaranya adalah menangani nilai-nilai yang kosong dan mengoreksi ketidakkonsistenan dalam data (Han et al., 2022).

Berikut ini merupakan tahapan yang dilakukan:

a. Mengecek dan menghapus baris yang berisi nilai kosong untuk setiap sheet data penjualan per bulan. Setiap sheet data penjualan memiliki baris yang bernilai kosong. Sebagai ilustrasi, sheet data penjualan bulan Februari 2021 mencatat 4 (empat) baris berisi nilai kosong atau null. Tabel 24 menunjukkan jumlah baris yang bernilai kosong atau null.

Tabel 24. Data Bulan pada proses data cleasing

Nama Sheet	Kegaitan
Nama Sheet	Jumlah baris bernilai Null
JANUARI 2023	4
FEBRUARI 2023	4
MARET 2023	4
…	
DESEMBER 2023	12

Berdasarkan tabel 24 terdapat kecenderungan peningkatan jumlah baris dengan nilai kosong seiring dengan semakin barunya data penjualan yang dianalisis. Oleh karena itu, peneliti menghapus baris data yang bernilai null untuk memastikan keakuratan dan kualitas analisis data. Mengecek dan menghapus nilai suatu kolom yang tidak diinginkan. Hal ini merupakan langkah penting dalam pembersihan data untuk

memastikan kualitas dan relevansi informasi yang akan dianalisis. Nilai suatu kolom yang tidak relevan dapat mengarahkan analisis ke kesimpulan yang salah. pengganti penamaan pada kolom data. Beberapa kolom data yang digunakan untuk analisis data lebih lanjut memerlukan perubahan nama, karena nama kolom tersebut jika tidak cukup representatif.

3. Data Selection

Seleksi data atau data selection merupakan proses untuk memilih dan mengekstraksi data yang relevan dari keseluruhan data untuk analisis lebih lanjut atau penggunaan spesifik. Pada contoh daya selection mengunakan kolom tanggal penjualan (DATE) serta nama produk penjualan tertentu, agar dapat menampilkan jumlah produk yang berhasil dijual pada bulan tersebut. Proses ini akan dilakukan untuk setiapsheet data penjualan per bulan.

Berikut contoh menunjukkan informasi setiap kolom setelah dilakukan seleksi data.

Tabel 25. Proses data selection

Kolom	Jumlah Kolom Terisis	Type Data
Date	87	Date Time
Produk A	30	Integer
Produk B	20	Integer
Produk C	10	Integer
Produk D	10	Integer

4. Data Transformation

Data transformation merupakan langkah untuk mengubah suatu pola struktur, data menjadi suatu bentuk yang diinginkan. Berikut ini merupakan tahapan yang dilakukan:

a. Pada penelitian ini, peneliti mengubah bentuk struktur data yang semula diurutkan data penjualan per hari menjadi data

penjualan setiap bulan kolom DATE akan dilakukan pengelompokkan atau grouping serta dilakukan perhitungan untuk mendapatkan jumlah keseluruhan hasil penjualan produk Produk A. Produk B, Produk C, Produk D selama bulan yang akan di presiksiakan tersebut. Proses ini akan dilakukan untuk setiap sheet data penjualan per bulan dengan melakukan gruping.

Tabel 26. Proses data Transformation

Kolom	Jumlah Kolom Terisi sebelem Grouping	Jumlah Kolom Terisi setelah Grouping
Date	87	1
Produk A	30	1
Produk B	20	1
Produk C	10	1
Produk D	10	1

b. Berikutnya, data penjualan per bulan yang masih terurai dalam berbagai sheet akan digabungkan ke dalam satu data terpusat. Hal ini bertujuan untuk memudahkan dan mempercepat analisis data serta meningkatkan efisiensi dalam pengolahan data penjualan untuk mendapatkan wawasan yang lebih akurat dan komprehensif. Selanjutnya, data penjualan yang sebelumnya tidak terurut berdasarkan tanggal penjualan, kemudian diatur berdasarkan urutan tanggal. Data penjualan per bulan yang masih terurai dalam berbagai sheet akan digabungkan ke dalam satu data terpusat. Hal ini bertujuan untuk memudahkan dan mempercepat analisis data serta meningkatkan efisiensi dalam pengolahan data penjualan untuk mendapatkan wawasan yang lebih akurat dan komprehensif.

5. Pembagian Data

Setelah melalui tahap data preparation, dataset kini siap untuk diterapkan dalam pemodelan data dengan algoritma *Extra Trees*. Namun, sebelum melakukan pemodelan, data perlu dibagi menjadi dua bagian utama, yaitu data latih (*train data*) dan data tes (*test data*). Data latih berguna untuk melatih model Extra Trees, sedangkan data uji digunakan untuk menguji model Extra Trees yang telah dikembangkan. Peneliti memiliki tiga variasi rasio perbandingan data latih dan data tes yaitu 80% data latih dengan 20% data tes, 70% data latih dengan 30% data tes, serta 60% data latih dengan 40% data tes.

10.6 Contoh Pemodelan Dengan Algoritma *Extra Trees*

Setelah menampilkan perbandingan rasio pembagian data latih dan data tes, pemodelan data dengan algoritma *Extra Trees*. Penelitian ini enggunakan pendekatan regresi, sehingga diperlukan *class Extra Trees* Regressor. Peneliti memiliki tiga (3) skenario percobaan dengan pembagian data latih dan data tes yang bervariasi. setiap skenario percobaan pembuatan model menggunakan metode algoritma *Extra Trees* dengan class Ext*ra Trees Regresso*r. Parameter yang digunakan pada kelas tersebut adalah default dengan parameter random_state sebesar 64. Parameter default tersebut memiliki nilai jumlah pohon keputusan yang terbentuk (n_estimators) sebesar seratus (100) pohon, memiliki metode untuk mengukur kualitas split atau titik kriteria pemecahan (criterion) menggunakan metode Mean Squared Error (squared_error), serta memiliki metode penentuan jumlah fitur terbaik (max_features) menggunakan metode akar kuadrat (sqrt(n_features)). Selanjutnya, jumlah minimum sampel atau baris data yang diambil ditentukan secara default, sesuai dengan konfigurasi setelannya, yaitu sebesar 2 (min_samples_split = 2).

1. Persiapan Data

Peneliti membuat subset data atau isi sebagian dataset secara acak sebanyak sepuluh baris data dengan kolom fitur (X) dan kolom target (y) berturut-turut yang digunakan:

a. Pemilihan Fitur Secara Acak

   Pemilihan fitur secara acak dilakukan dengan menggunakan metode akar kuadrat (*square root*). Total fitur pada penelitian ini sebesar 1, dikarenakan jika dikalkulasikan 1 menjadi 1. Sehingga, algoritma hanya mengambil satu fitur.

b. Penentuan Nilai Threshold & Penghitungan Kriteria Pengukuran Split

   Setelah pemilihan fitur didapatkan, algoritma menghitung nilai ambang batas atau threshold pada setiap kolom secara acak, untuk dapat membagi data (split) sesuai minimum jumlah sampel yang akan dipecah sebesar 2 sampel (min_samples_split = 2). Pemisahan data baru hanya dimasukkan ke dalam pohon keputusan apabila jumlah sampel setara atau lebih besar dari nilai tersebut. Dalam menentukan nilai ambang batas algoritma menghitung nilai rata-rata isi data pertama dan kedua dari subset data. Berikutnya, data dibagi menjadi dua bagian, yaitu bagian kiri dan bagian kanan. Bagian kiri merupakan representasi data yang kurang dari atau sama dengan nilai threshold. Sebaliknya, bagian kanan merupakan representasi data yang lebih dari nilai *threshold*. Dua bagian tersebut memiliki persamaan, yaitu untuk menghitung nilai rata-rata dari setiap bagian. Proses penghitungan nilai rata-rata dilakukan secara iterative hingga tidak ada lagi baris yang dapat dihitung.

Tabel 27. Pengurutan Subset Data Berdasarkan Fitur Month XX & Pembagian Nilai Threshold

index	Month (X)	Produk A (y^1)	Jumlah 2 Baris Data (X n + X n+1)	Threshold
0	3	25	5	2,5
1	5	17	8	4
2	11	3	16	18
…	…	..	…	..
9	18	7	29	14,5

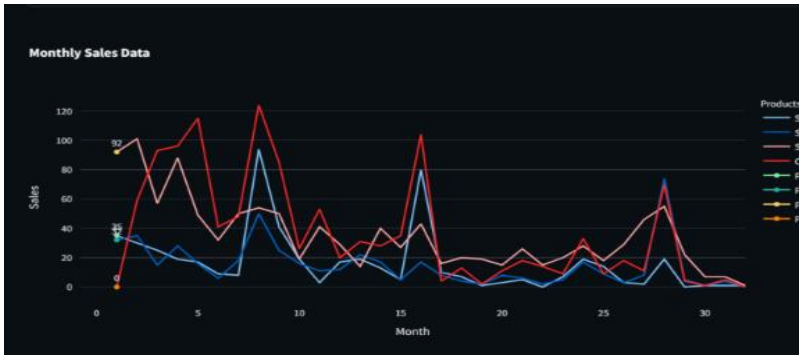
c. Perhitungan rata-rata kolom fitur dan kolom target
Menampilkan perhitungan nilai rata-rata berdasarkan isi dari subset data dannilai threshold. Sesuai perhitungan nilai rata-rata berdasarkan isi dari subset data dan nilai ambang batas Dimana nilai rata-rata kolom Month dianggap sebagai nilai prediksi. setiap nilai dari kolom fitur Produk A Pcs dipastikan masuk ke dalam setiap kelompok pembagian. Setiap nilai yang masuk ke dalam Bagian Kiri dan Bagian Kanan akan dihitung nilai rata-rata. Hasil nilai rata-rata setiap Bagian Kiri dan BagianKanan pada masing-masing kelompok dapat diasosiasikan sebagai nilai prediksi. Selanjutnya, nilai Mean Squared Error pada setiap kelompok dihitung agar mendapatkan titik penentuan kriteria terbaik dalam pembentukan node pertama. Perhitungan nilai Mean Squared Error untuk setiap bagian perlu dilakukan perhitungan total, atau nilai Mean Squared Error total. Persamaan (4.2) bertujuan untuk menghitung MSE total serta perhitungan nilai Mean Squared Error. Nilai Mean Squared Error pada suatu kelompok pada Kelompok 1 adalah 57.28dibandingkan dengan kelompok lainnya. Jika nilai Mean Squared Error pada suatu kelompok memiliki nilai terkecil, maka nilai tersebut dapat dijadikan simpul atau node.

d. Penentuan error

Evaluasi model diukur dengan beberapa metode nilai eror seperti *Mean absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE). Pengukuran kualitas suatu nilai eror didasarkan pada prinsip bahwa nilai eror yang kecil lebih baik dibandingkan dengan skenario percobaan yang menghasilkan nilai eror yang besar.

e. Prediksi (*Predictive Analysis*)

Berdasarkan evaluasi model, didapatkan hasil model terbaik dengan rasio data latih dandata tes sebesar 70% banding 30%. Oleh karena itu, peneliti menggunakan model tersebut.untuk melakukan prediksi. Prediksi dibuat ketika pengguna memasukkan input data, dan model Extra Trees sesuai dengan data latih tersebut untuk menghasilkan prediksi penjualan produk. Setelah model selesai diproses, hasil prediksi ditampilkan berdasarkan hasil input yang dimasukkan pengguna. tabel 12.4 yang memuat bulan penjualan berdasarkan data yang dimiliki dengan nilai aktual (nilai penjualan yang sebenarnya) serta bulan penjualan berdasarkan keinginan pengguna. Berikut contoh gambar 39 grafik visualisasi prediksi dari data penjualan tersebut



Gambar 39. Grafik visualisasi prediksi data penjualan

Pada sisi kiri layar, terdapat satu panel interaktif, yaitu panel yang meminta pengguna untuk memasukkan angka penjualan yang ingin diprediksi penjualannya, dengan tombol "Predict Sales" di bawahnya untuk memulai prediksi. Selanjutnya, Di sisi kanan layar, terdapat tabel "View Sales Data" yang menampilkan data penjualan historis dengan kolom DATE, Month, dan jumlah penjualan untuk produk A, PRODUK B, PRODUK C, dan produk d.

10.7 Kesimpulan

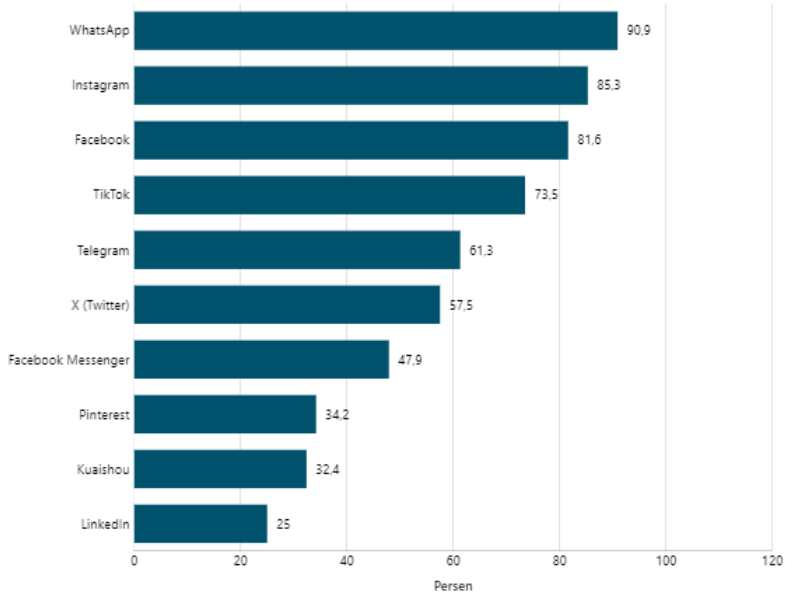
Monitoring terhadap pemanfaat Teknologi TI berkembang pesat, sebagian besar berkat pembelajaran mesin (*machine learning*) dan analisis prediktif. (*predictive analytics*). Jika masih ada Perusahaan yang masih bertahan dengan solusi TI yang lama makan kemungkinan besar akan kehilangan kempatan tidak hanya sekadar benda cemerlang (*Shiny Object*).

Analisis Prediktif dapat mengetahui bagaimana pendekatan prediktif berbasis data (*data-driven*) terhadap monitoring TI akan membawa Perusahaan ke dunia TI Baru. Perusahaan atau organisasi dapat mengetahui posisi dalam Kerangka Kematangan TI (*Maturity Framework*).

Bab 11. Konsep Text Mining

11.1 Pengantar

Kondisi dunia media social saat ini terpantau cukup giat diminati oleh masyarakat Indonesia. Kehidupan media social kini terasa lebih mudah diakses dan sangat terbuka untuk siapa saja.



Gambar 40. Penggunaan media social per Januari 2024
Sumber: (Annur, 2024)

Dikarenakan keaktifkan social media yang sangat besar setiap menitnya, itu diartikan banyak sekali data tidak terstruktur yang bisa digunakan untuk penelitian dari social media. Yang termasuk ke dalam data tidak terstruktur atau *unstructed data* ini diantaranya adalah: data berupa *text*, data berupa gambar, data berupa audio dan juga data berupa video (Mount & Zumel, 2019).

Jenis *unstructed data* ini belum memiliki nilai, data akan memiliki nilai setelah terjadi analisa dan pengolahan untuk data tersebut. Dengan adanya kegiatan sehari-hari yang berhubungan dengan komunikasi baik secara individu ataupun komunikasi di media social, maka data yang berkaitan erat dan selalu digunakan adalah data berupa *text*. Muncul pertanyaan seperti ini, "Apakah data text tersebut bermanfaat?". Jawabannya adalah data-data tersebut sangat bermanfaat untuk mendapatkan *knowledge* baru. Solusi dari bagaimana data-data tersebut bisa digunakan adalah

11.2 Text mining

Penambangan teks atau yang lebih sering kita kenal dengan *text mining* merupakan cabang ilmu dari data mining Firdaus & (Firdaus, 2021). *Text Mining* adalah proses ekstraksi berharga dari data berupa teks dan mengidentifikasi pola, proses penerapannya menggunakan bantuan metode-metode penelitian. Tujuan text mining adalah mengubah data teks mentah menjadi bentuk yang terstruktur sehingga menghasilkan *knowledge* (Ilmu pengetahuan) (Chakraborty et al., 2014). Proses ekstraksinya nya mencakup segala hal mulai dari pencarian informasi (misalnya, pencarian dokumen atau situs web) hingga klasifikasi dan pengelompokan teks, hingga (yang lebih baru) yaitu melakukan ekstraksi entitas, relasi, dan peristiwa (Kao & Poteet, 2018).

Selain pengertian yang sudah dibahas, text mining juga memiliki arti dari sudut pandang bisnis. Karena pada kenyataannya text mining sangat bermanfaat dalam menentukan kebutuhan produk konsumen. Dari sisi bisnis pengertian *text mining* merupakan tindaklanjut dari sebuah teks melewati proses penyaringan pengetahuan atau wawasan (Kwartler, 2017).

Dari pengertian diatas kita dapat mengetahui bahwa penambangan teks atau *text mining* merupakan sebuah pembuatan informasi baru dari data mentah berupa teks yang

diperoleh dari ekspresi manusia menggunakan metode-metode penelitian yang melewati proses ekstraksi.

1. Seberapa Penting Text Mining dalam Analisis Data?

Text mining memiliki peran penting dalam analisis data, karena prosesnya yang mengubah data teks tidak terstruktur menjadi sebuah informasi bernilai. Berikut alasan mengapa text mining memiliki peran penting dalam analisis data:

a. Mengolah data teks yang tidak terstruktur
   Sebagian besar data di dunia tidak memiliki struktur yang terdefinisi dengan baik, terutama dalam bentuk teks seperti email, media sosial, dokumen bisnis, catatan medis, dan laporan berita. Teknik text mining mempermudah manusia untuk mengekstrak informasi dari sumber-sumber tersebut dan mengubahnya menjadi bentuk yang dapat dianalisis secara sistematis.
b. Mengidentifikasi Pola dan Tren
   Teknik text mining membantu mengenali pola, tren, dan keterkaitan dalam data teks yang luas. Dengan kemampuan ini, kita dapat menemukan informasi yang tidak terlihat hanya dengan membaca teks. Sebagai contoh, dalam analisis media sosial, text mining dapat mengungkap tren yang sedang populer atau sentimen pengguna terkait suatu topik.
c. Analisis Sentimen
   Analisis sentimen merupakan salah satu penerapan yang populer dalam text mining, dalam hal ini kita dapat mengidentifikasi sikap, emosi, atau opini dalam teks. Hal ini sangat bermanfaat dalam bisnis untuk memahami umpan balik dari pelanggan, di media sosial untuk mengukur sentimen publik, atau dalam analisis politik untuk memahami pandangan pemilih.
d. Ekstraksi Informasi dan Entitas
   Teknik text mining memungkinkan kita mengidentifikasi entitas dan informasi krusial dari teks, seperti nama individu,

lokasi, organisasi, tanggal, dan lainnya. Teknik ini sangat bermanfaat dalam ekstraksi informasi medis, di mana kita dapat mengekstraksi nama obat atau gejala penyakit dari catatan medis.

e. Pengambilan Keputusan yang Lebih Baik
   Dengan kemampuannya dalam mengekstrak wawasan dan informasi yang mendalam, text mining berperan dalam mendukung pengambilan keputusan yang lebih baik. Data yang diperoleh melalui text mining dapat memberikan informasi yang relevan untuk menginformasikan strategi bisnis, keputusan pemasaran, atau bahkan kebijakan pemerintah.
f. Efisiensi dan Automasi
   Teknik text mining memungkinkan otomatisasi analisis data teks dalam skala besar, yang jika dilakukan secara manual akan memakan waktu yang lama. Dengan otomatisasi ini, perusahaan dan organisasi dapat menganalisis data dalam jumlah besar dengan cepat dan efisien.

Secara keseluruhan, text mining adalah alat yang efektif untuk mengonversi data teks yang tidak terstruktur menjadi informasi yang dapat digunakan untuk tindakan lebih lanjut. Teknologi ini memiliki peran signifikan dalam analisis data, karena memungkinkan kita mengenali pola, tren, dan informasi yang relevan, yang pada akhirnya mendukung keputusan yang lebih baik di berbagai bidang.

11.3 Aplikasi Text Mining dalam Berbagai Bidang

Seletah mengetahui beberapa peran text mining, sekarang pertanyaan berikutnya muncul lebih detail. Bagaimana aplikasi text mining digunakan?

1. Aplikasi text mining dalam bidang keuangan.

Text mining telah menjadi alat yang sangat berharga dalam bisnis dan keuangan, memungkinkan perusahaan dan institusi

keuangan untuk mengolah dan menganalisis data teks dalam jumlah besar. Berikut adalah beberapa manfaat utama aplikasi text mining dalam bisnis dan keuangan:

a. Analisis umpan balik pelanggan
   Sebagai sebuah perusahan adanya umpan balik dari pelanggan itu merupakan hal yang penting, hal ini dijadikan bahan evaluasi untuk kebijakan yang dibuat oleh perusahaan. Menurut (Fawcett, 2013) text mining dapat membantu bisnis memahami umpan balik pelanggan dari berbagai sumber seperti ulasan produk, survei, dan media social.
b. Analisis sentimen dan tren pasar
   Text mining digunakan untuk menganalisis sentimen pasar dari berita, laporan keuangan, dan media sosial. Analisis ini dapat membantu perusahaan memahami tren pasar, sentimen investor, dan persepsi publik, sehingga mereka dapat mengambil keputusan investasi yang lebih baik.
c. Ekstraksi Informasi dari dokumen keuangan
   Dalam buku "*The Data Warehouse Toolkit*" (Kimball & Ross, 2013), Kimbal menyebutkan bahwa text mining dapat digunakan untuk mengekstrak informasi penting dari dokumen keuangan seperti laporan tahunan, laporan audit, dan berita perusahaan. Ini membantu analis keuangan mendapatkan wawasan yang lebih mendalam tentang kinerja perusahaan dan risiko keuangan.
d. Prediksi penipuan dan kemanan
   Teknik text mining selanjutnya pernah dituliskan dapat memprediksi penipuan dan juga keamanan. Seperti yang pernah ditulis oleh Delena pada bukunya (Spann, 2014), text mining mendeteksi penipuan dengan menganalisis pola teks dalam transaksi keuangan dan laporan. Hal ini membantu perusahaan mendeteksi aktivitas mencurigakan dan mencegah kerugian akibat penipuan.

2. Aplikasi text mining dalam bidan kesehatan dan kedokteran.

Aplikasi text mining didunia kesehatan dan kedokteran juga sudah banyak digunakan dan dikembangkan. Para peneliti menggunakan teknik text mining untuk beberapa aplikasi dibidang kesehatan dan kedokteran. Diantaranya:

a. Ekstraksi Informasi dari Catatan Medis Elektronik (Electronic Health Records, EHR).

   Text mining membantu mengidentifikasi informasi penting dari catatan medis yang tidak terstruktur, seperti diagnosis, obat-obatan, gejala, dan riwayat pasien. Ini memungkinkan para profesional medis untuk mengekstrak data klinis dengan cepat dan akurat untuk analisis lebih lanjut. Beberapa referensi, seperti studi oleh (Cowie et al., 2017) menunjukkan bahwa text mining dapat meningkatkan efisiensi dalam menangani data medis.

b. Penelitian dan Pengembangan Obat

   Penelitian obat untuk mengidentifikasi senyawa potensial, interaksi obat, atau informasi penting dari jurnal medis dan basis data penelitian dapat menggunakan teknik text mining. Seperti pada penelitian sebelumnya yang dibuat oleh (Gonzalez et al., 2013) memberikan contoh aplikasi text mining dalam penelitian obat dan bagaimana ini membantu mempercepat proses penemuan obat.

Bermanfaat sekali ya text mining?! . Selain dua bidang yang telah disebutkan, masih ada aplikasi text mining yang berhubungan dengan aktifitas kita sehari-hari, yaitu:

3. Aplikasi text mining pada penggunaan social media.

Analisis sentimen, pemantauan tren dan isu hangat, membuat target pasar dan segmentasi pelanggan dan pendeteksian berita palsu atau yang biasa kita sebut *hoax* merupakan aplikasi dari text mining yang digunakan sampai saat ini. Selain itu yang tak kalah penting berikutnya aplikasi text mining untuk menganalisis competitor dan *brenchmarking*.

Seperti penelitian yang pernah dibuat oleh Nahda (Ramadhanti & Mariyah, 2019).

11.4 Langkah Dasar Text Mining

Proses text mining melibatkan serangkaian langkah untuk mengubah data teks yang tidak memiliki struktur yang terdefinisi menjadi bentuk yang dapat dianalisis. Tujuan utamanya adalah mengekstrak informasi dan pengetahuan dari teks tersebut. Bagaimana langkah-langkah memulai text mining?.

Berikut langkah-langkah dasar text mining sebelum data teks diekstrak menggunakan metode penelitian.

1. Proses Pengumpulan Data Teks

Langkah awal dalam text mining adalah menghimpun data teks yang relevan. Data teks dapat diperoleh dari berbagai sumber, termasuk dokumen, email, laporan, situs web, media sosial, atau catatan medis. Darimana data teks didapatkan?. Sumber data teks bisa didapatkan dari sumber-sumber berikut:

- Dokumen Teknis: Laporan, penelitian, atau dokumentasi teknis.
- Media Sosial: Postingan dan komentar dari platform seperti Twitter, Facebook, Instagram bahkan juga Tiktok.
- Situs Web: Konten dari situs berita atau blog.
- Catatan Medis: Rekam medis pasien dalam bentuk elektronik.
- Contoh pengambilan data dari twitter bisa menggunakan bantuan API Twitter. Untuk dapat terhubung dengan API Twitter kita dapat menggunakan python agar mendapatkan akses mengambil komentar dari twitter. Terlebih dahulu install extension Tweepy. Kita akan dapat mengambil teeks dengan berdasarkan hastag ataupun kata kunci.
- Install extention tweepy:
- !pip install tweepy

Berikutnya kita dapat membuat *script* python sebagai berikut:

```
import tweepy
import csv
# Kredensial API
API_KEY = 'your_api_key'
API_SECRET_KEY = 'your_api_secret_key'
ACCESS_TOKEN = 'your_access_token'
ACCESS_TOKEN_SECRET = 'your_access_token_secret'
# Autentikasi dengan Twitter API
auth = tweepy.OAuthHandler(API_KEY, API_SECRET_KEY)
auth.set_access_token(ACCESS_TOKEN,
ACCESS_TOKEN_SECRET)
# Membuat objek API
api = tweepy.API(auth)
# Mencari tweet berdasarkan kata kunci
tweets = api.search_tweets(q="artificial intelligence",
count=100, lang="en")
# Simpan hasil ke dalam file CSV
with open('tweets.csv', 'w', newline='') as csvfile:
csvwriter = csv.writer(csvfile)
 # Header
 csvwriter.writerow(["username", "tweet", "created_at"])
 # Menyimpan data tweet
 for tweet in tweets:
csvwriter.writerow([tweet.user.screen_name, tweet.text,
tweet.created_at])
```

script diatas akan menyimpan data kedalam bentuk csv. Selanjutnya data dapat diimport kedalam aplikasi rapidminer untuk langkah berikutnya, yaitu perhitungan menggunakan metode penelitian.

2. Prakondisi dan Pembersihan Data

Setelah mengumpulkan data teks, langkah selanjutnya adalah membersihkan dan melakukan prakondisi pada data tersebut. Proses ini sangat penting untuk memastikan bahwa data

siap untuk dianalisis. Beberapa langkah umum dalam pembersihan data meliputi:

a. Penghapusan Karakter Tidak Perlu: Menghilangkan karakter yang tidak relevan, seperti tanda baca, simbol, atau angka yang tidak diperlukan.
b. Normalisasi Teks: Mengubah teks ke format yang konsisten, misalnya dengan mengonversi semua huruf menjadi huruf kecil.
c. Penghapusan Spasi Berlebih: Menghilangkan spasi ekstra antara kata atau di akhir kalimat.

   Contoh:

   Teks asli: "Hai..! apa arti dari text mining? 123."

   Setelah pembersihan: "hai apa arti dari text mining"

3. Tokenisasi dan Stemming

Tokenisasi merupakan proses membagi teks menjadi unit-unit kecil yang disebut token, biasanya berdasarkan kata atau frasa (Martin, 2019). Dan stemming adalah proses menghilangkan akhiran dari kata untuk mendapatkan akar kata. Contoh:

Tokenisasi: Memecah teks menjadi kata-kata atau frasa.

Contoh: "Hai apa arti dari text mining" menjadi ["hai", "apa", "arti", "dari", "text", "mining"]

Stemming: Menghilangkan akhiran untuk mendapatkan akar kata.

Contoh: ["menghilankan", "dihilangkan"] menjadi ["hilang", "hilang"]

4. Proses Stopword dan Kata Kunci

Stopword merujuk pada kata-kata umum yang sering diabaikan dalam analisis karena kurang memberikan informasi yang signifikan, seperti “the”, “is”, “and”, “a”, dan sejenisnya. Dalam tahap ini, stopword dihilangkan dari token yang dihasilkan. Sementara itu, kata kunci adalah kata atau frasa penting yang

memiliki nilai informatif dan dapat digunakan untuk analisis lebih lanjut.

Contoh menggunakan bahasa inggris:

Penghapusan Stopword: Menghilangkan kata-kata umum sehingga [“hello”, “world”, “this”, “is”, “a”, “test”, “message”] tanpa stopword menjadi [“hello”, “world”, “test”, “message”].

Identifikasi Kata Kunci: Menemukan kata-kata penting yang mewakili konsep atau topik tertentu. Misalnya, dalam teks berita tentang teknologi, kata kunci bisa berupa [“Sentimen analisis”, “klasifikasi”, “presiden”].

Setelah dilakukan langkah-langkah awal, maka teks yang sudah dibersihkan siap diolah menggunakan bantuan software rapidminer ataupun menggunakan python untuk menyelesaikan permasalahan klasifikaksi, prediksi maupun yang lain.

Bab 12. Text Mining, Mesin Pencarian

12.1 Pengenalan *Text Mining*

Text mining, juga dikenal sebagai analitik teks, merupakan proses menganalisis teks dalam format tidak terstruktur untuk menggali informasi bermakna dan memungkinkan organisasi dan individu untuk mengidentifikasi tren, pola, dan hubungan dalam data teks yang mungkin tidak terlihat pada pandangan pertama (Rahayu et al., 2024). Teknik ini melibatkan berbagai aspek seperti: linguistik, statistik, dan pembelajaran mesin untuk mengubah teks menjadi data yang dapat dianalisis secara kuantitatif.

Tujuan utama *text mining* adalah untuk mengubah data teks tidak terstruktur menjadi data terstruktur yang dapat digunakan untuk analisis lebih lanjut (Mauliza & Sipayung, 2024). *Text mining* dapat diimplementasikan menggunakan analisis sentimen, pengelompokan dokumen, penemuan topik, dan pengambilan keputusan bisnis. Beberapa langkah dasar dalam *text mining* meliputi pra-pemrosesan teks, tokenisasi, penghapusan *stop words*, *stemming*, dan *lemmatization* (Diantoni et al., 2024).

Contoh penggunaan *text mining* sederhana

Ilustrasi:

1. Perusahaan XYZ ingin melakukan ekstraksi data dan ekloprasi data dari hasil *review* produk yang mereka tawarkan yakni:

"Produk ini sangat bagus dan berkualitas tinggi.",
"Saya tidak puas dengan produk ini.",
"Layanan pelanggan sangat membantu dan ramah.",
"Produk ini tidak sesuai dengan harapan saya."

Lakukan eksplorasi teks dari *review* yang diberikan sehingga dapat memprediksikan *review* yang baru untuk produk mereka serta lihat apakah *review* tersebut mengindikasikan perspektif baik dari konsumen.

- Python

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score, StratifiedKFold
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
from sklearn import metrics
import numpy as np

# Unduh stopwords dan punktuasi
nltk.download('stopwords')
nltk.download('punkt')

# Kumpulan ulasan
reviews = [
    "Produk ini sangat bagus dan berkualitas tinggi.",
    "Saya tidak puas dengan produk ini.",
    "Layanan pelanggan sangat membantu dan ramah.",
    "Produk ini tidak sesuai dengan harapan saya."
]

# Label ulasan (1: positif, 0: negatif)
labels = [1, 0, 1, 0]
```

TAMPILAN OUTPUT

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
```

- Python

```
# Preprocessing function
def preprocess(text):
  # Tokenisasi
  tokens = word_tokenize(text.lower())
  # Menghapus stop words
tokens = [word for word in tokens if word not in stopwords.words('indonesian')]
  # Stemming
  stemmer = PorterStemmer()
  tokens = [stemmer.stem(word) for word in tokens]
  print(tokens)
  return tokens

# Preprocess all reviews
# processed_reviews = [preprocess(review) for review in reviews]
processed_reviews = [' '.join (preprocess(review)) for review in reviews]
print (processed_reviews)
print (len(processed_reviews))
```

TAMPILAN OUTPUT

```
['produk', 'bagu', 'berkualita', '.']
['pua', 'produk', '.']
['layanan', 'pelanggan', 'membantu', 'ramah', '.']
['produk', 'sesuai', 'harapan', '.']
```

```
['produk bagu berkualita .', 'pua produk .', 'layanan pelanggan membantu ramah .', 'produk sesuai harapan .']
4
```

Tahapan diatas dilakukan untuk menentukan akhir kalimat dan kata sambung, lalu dijadikan dalam bentuk kata sederhana serta mengubah menjadi suatu *array text* baru pada fungsi preprocess. Untuk memahami lebih lanjut, lakukan print(tokens) pada fungsi ini. Sementara pada *variable* processed_reviews bertugas untuk menggabungkan beberapa *array* tersebut menjadi *array of strings* dengan panjang 4.

- Python

```
# Membuat Representasi Fitur
# Vectorizer
vectorizer = TfidfVectorizer()

# Mengubah teks menjadi vektor
tfidf_matrix = vectorizer.fit_transform(processed_reviews)
print (tfidf_matrix.toarray())
```

TAMPILAN OUTPUT

```
[[0.64450299 0.64450299 0.         0.         0.         0.
  0.41137791 0.         0.         0.                    ]
 [0.         0.         0.         0.         0.         0.
  0.53802897 0.84292635 0.         0.                    ]
 [0.         0.         0.         0.5        0.5        0.5
  0.         0.         0.5        0.                    ]
 [0.         0.         0.64450299 0.         0.         0.
  0.41137791 0.         0.         0.64450299            ]]
```

Pada tahapan ini, mengubah processed_reviews dari *array of strings* menjadi bentuk *vector* sehingga dapat melakukan *train test split* menggunakan *vector* tersebut yang dianggap sebagai fitur.

- Python

```
# Membangun model Naive Bayes
model = MultinomialNB()
# Menggunakan StratifiedKFold untuk cross-validation
skf = StratifiedKFold(n_splits=len(processed_reviews)-2, shuffle=True, random_state=42)
# Evaluasi dengan cross-validation
for train_index, test_index in skf.split(tfidf_matrix, labels):
X_train, X_test = tfidf_matrix[train_index], tfidf_matrix[test_index]
y_train, y_test = [labels[i] for i in train_index], [labels[i] for i in test_index]
  model.fit(X_train, y_train)
  predictions = model.predict(X_test)
  report = metrics.classification_report(y_test, predictions, output_dict=True)
  f1_scores.append(report['macro avg']['f1-score'])
  supports.append(report['macro avg']['support'])
print(f"F1-scores dari setiap fold: {f1_scores}")
print(f"F1-score rata-rata: {np.mean(f1_scores)}")
print(f"Supports dari setiap fold: {supports}")
```

```
TAMPILAN OUTPUT
StratifiedKFold(n_splits=2, random_state=42, shuffle=True)
F1-scores dari setiap fold: [0.33333333 0.33333333]
F1-score rata-rata: 0.3333333333333333
Supports dari setiap fold: [2, 2]
```

Dalam tahap ini berfungsi untuk membangun model *naïve bayes* dari fitur yang berbentuk vektor. Secara sederhana, dapat secara langsung melakukan *model fit* dengan kode dibawah ini

- Python

```
# Membagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix, labels, test_size=0.25, random_state=42)
# Membangun model Naive Bayes
model = MultinomialNB()
model.fit(X_train, y_train)
# Prediksi
predictions = model.predict(X_test)
# Evaluasi
print(metrics.classification_report(y_test, predictions))
```

TAMPILAN OUTPUT

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1.0
1	0.00	0.00	0.00	0.0
accuracy			0.00	1.0
macro avg	0.00	0.00	0.00	1.0
weighted avg	0.00	0.00	0.00	1.0

Dapat terlihat dari *output* yang didapat f1-score menunjukkan nilai 0. Hal ini menandakan bahwa model gagal untuk memprediksi salah satu kelas dengan benar. Ada beberapa kemungkinan penyebab masalah ini, terutama dengan dataset kecil seperti yang digunakan dalam contoh. Berikut beberapa langkah yang dapat dilakukan untuk mengevaluasi dan memperbaiki masalah tersebut.

a. Periksa Distribusi Data:
   Pastikan bahwa data dalam set pelatihan dan pengujian memiliki distribusi kelas yang seimbang.

- Cross-Validation:
  Gunakan teknik *cross-validation* untuk memastikan bahwa model tidak *overfitting* atau *underfitting*.
- Tambah Data:
  Menggunakan lebih banyak data untuk pelatihan akan membantu model untuk mempelajari pola yang lebih baik.
- Periksa Preprocessing:
  Pastikan bahwa preprocessing dilakukan dengan benar dan tidak menghapus terlalu banyak informasi dari teks.

Untuk mengatasi permasalahan tersebut, maka dalam bab ini memutuskan untuk menggunakan metode *cross validation* sebagai contoh Solusi seperti kode yang tertera sebelumnya. Algoritma yang digunakan menggunakan *Stratified K Fold* dengan konfigurasi

StratifiedKFold(n_splits=2, random_state=42, shuffle=True

n_splits yang digunakan bernilai 2 hal ini dikarenakan panjang *array processed_reviews* bernilai 4. Jika menggunakan keseluruhan panjang *array* akan menghasilkan error.

- ValueError: n_splits=4 cannot be greater than the number of members in each class.
  Sehingga dalam contoh ini, penulis menggunakan panjang *array* – 2 untuk membagi kedalam kedua kelompok saja (0,1) dimana 0 menunjukkan *negative* respon dari teks dan 1 menunjukkan *positive* respon dari teks. Akan lebih tepat jika langsung menggunakan n_splits = 2 jika panjang *array* > 4. Maka, hasil yang didapatkan seperti dibawah ini:
  F1-scores dari setiap fold: [0.33333333 0.33333333]
  F1-score rata-rata: 0.3333333333333333
  Supports dari setiap fold: [2, 2]
- Dapat disimpulkan bahwa: *StratifiedKFold*: Memastikan setiap *fold* memiliki distribusi kelas yang sama, ini penting untuk *dataset* kecil.

- *Cross-Validation*: Memberikan evaluasi yang lebih baik tentang bagaimana model akan berkinerja pada data yang tidak terlihat.
- Latih ulang model: Setelah evaluasi *cross-validation*, model dilatih ulang dengan seluruh dataset untuk digunakan dalam prediksi baru.

Dengan teknik *cross-validation*, dapat diperoleh gambaran yang lebih baik tentang performa model dan mengurangi risiko *overfitting*, terutama dengan dataset kecil seperti yang dilakukan pada contoh ini.

Adapun penjelasan terkait *terms* hasil akurasi sebagai berikut:

1) *Precision* (Presisi): Proporsi prediksi positif yang benar dari semua prediksi positif yang dibuat
2) Recall: Proporsi prediksi positif yang benar dari semua kasus positif yang sebenarnya
3) *F1-Score*: Rata-rata harmonis dari presisi dan *recall*.
4) *Support*: Jumlah kejadian sebenarnya dari kelas tertentu dalam data pengujian.

Selanjutnya, diperlukan prediksi untuk ulasan baru atau teks baru dengan Langkah berikut:

- Python

```
# Latih ulang model dengan seluruh dataset
model.fit(tfidf_matrix, labels)

# Ulasan baru
new_reviews = [
   "Produk ini sangat mengagumkan dan memenuhi harapan saya.",
   "Saya sangat kecewa dengan produk ini, tidak sesuai deskripsi."
]

# Preprocess new reviews
```

```
processed_new_reviews = [' '.join(preprocess(review)) for review in new_reviews]

# Transform ulasan baru dengan vectorizer yang sudah dilatih
new_tfidf_matrix = vectorizer.transform(processed_new_reviews)

# Prediksi ulasan baru
new_predictions = model.predict(new_tfidf_matrix)
print(new_predictions)
# Menampilkan hasil prediksi
for review, pred in zip(new_reviews, new_predictions):
  sentiment = "Positif" if pred == 1 else "Negatif"
  print(f"Review: {review}\nPrediksi Sentimen: {sentiment}\nNilai Prediksi: {pred}\n")
```

TAMPILAN OUTPUT

```
[0 0]
Review: Produk ini sangat mengagumkan dan memenuhi harapan saya.
Prediksi Sentimen: Negatif
Nilai Prediksi: 0

Review: Saya sangat kecewa dengan produk ini, tidak sesuai deskripsi.
Prediksi Sentimen: Negatif
Nilai Prediksi: 0
```

Dalam contoh ini, model tidak mampu untuk memprediksi dari ulasan baru. Namun, bila kode diganti menjadi :

- Python

```
# Ulasan baru
new_reviews = [
   "Produk ini sangat bagus dan berkualitas tinggi.",
   "Saya sangat kecewa dengan produk ini, tidak sesuai deskripsi."
]

# Preprocess new reviews
processed_new_reviews = [' '.join(preprocess(review)) for review in new_reviews]

# Transform ulasan baru dengan vectorizer yang sudah dilatih
new_tfidf_matrix = vectorizer.transform(processed_new_reviews)

# Prediksi ulasan baru
new_predictions = model.predict(new_tfidf_matrix)
print(new_predictions)
# Menampilkan hasil prediksi
for review, pred in zip(new_reviews, new_predictions):
   sentiment = "Positif" if pred == 1 else "Negatif"
   print(f"Review: {review}\nPrediksi Sentimen: {sentiment}\nNilai Prediksi: {pred}\n")

#  Latih ulang model dengan seluruh dataset
model.fit(tfidf_matrix, labels)
print(report_all)
```

TAMPILAN OUTPUT

```
[1 0]
Review: Produk ini sangat bagus dan berkualitas tinggi.
Prediksi Sentimen: Positif
Nilai Prediksi: 1
```

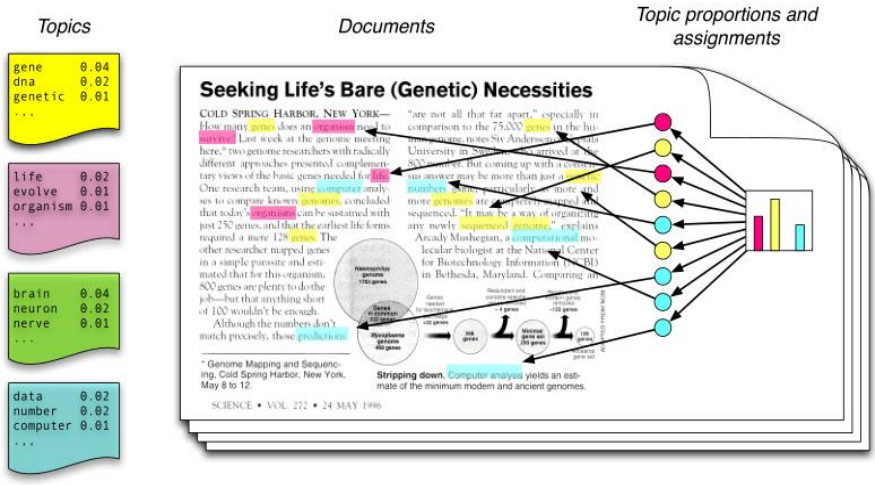
```
Review: Saya sangat kecewa dengan produk ini, tidak sesuai deskripsi.
Prediksi Sentimen: Negatif
Nilai Prediksi: 0

              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         2

    accuracy                           1.00         4
   macro avg       1.00      1.00      1.00         4
weighted avg       1.00      1.00      1.00         4
```

Hasil prediksi yang didapatkan sesuai dengan pembagian kelasnya, serta laporan presisi dan akurasi secara *overall* mampu melakukan prediksi. Namun, dikarenakan *dataset* yang digunakan terlampau sedikit untuk dipelajari oleh model maka model mengalami *overfit* dimana model hanya mampu memprediksi dari *input* baru yang hampir atau sama dengan *dataset* yang diberikan. Perlu diperhatikan bahwa, *text mining* mengharuskan untuk menggunakan *dataset* yang sangat besar untuk mendapatkan hasil yang akurat dan andal. Selanjutnya, pada pembahasan lanjutan akan memaparkan berbagai teknik dasar dalam *text mining* yang digunakan untuk memilih algoritma model yang optimal ataupun metode optimasi model yang sudah dibangun.

12.2 Teknik-Teknik Dasar dalam Text Mining

Text mining akan berusaha untuk mengelompokkan kalimat dalam beberapa klaster (topik/tema) lalu menggunakan topik tersebut menemukan informasi yang tersirat (Xiong et al., 2024). Dalam kasus contoh sebelumnya dilakukan 2 pengelompokkan, yakni 0 (untuk pesan negatif) dan 1 (untuk pesan positif). Konsep

ini disebut dengan *latent Dirichlet allocation* (LDA), dimana memungkinkan untuk menemukan factor lainnya yang tersirat secara otomatis dan meningkatkan *data utilization efficiency*.



Gambar 41. Ilustrasi konsep LDA
Sumber: (STELLA et al., n.d.)

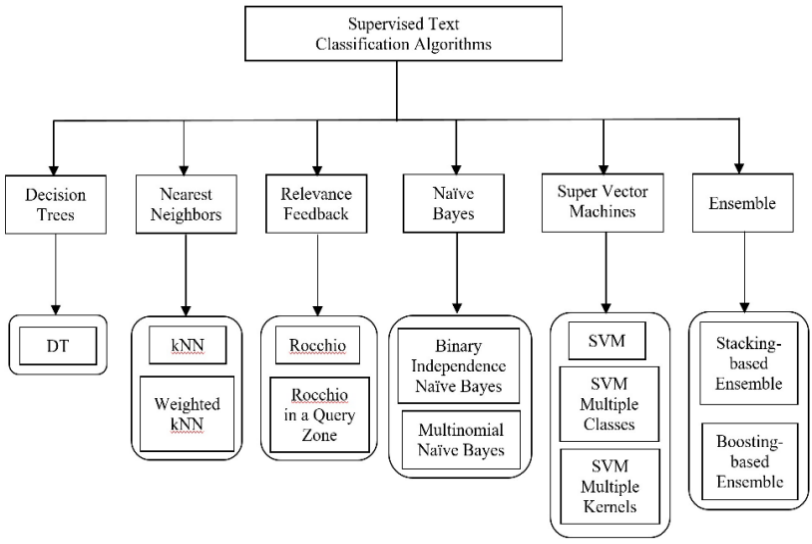
Teknik dasar yang digunakan dalam text mining mencakup: Tokenisasi: Proses memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau frasa.

- *Stemming* dan *Lemmatization*: Teknik untuk mengurangi kata-kata ke bentuk dasarnya.
- Penghapusan *Stop Words*: Menghilangkan kata-kata umum yang tidak memiliki banyak makna seperti "dan", "atau", "di".
- *Bag of Words* (BoW): Model representasi teks di mana teks diubah menjadi vektor kata tanpa memperhatikan urutan kata.
- TF-IDF (*Term Frequency-Inverse Document Frequency*): Teknik untuk menilai kepentingan sebuah kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya.

Adapun dalam contoh pembahasan pada awal bab menerapkan tokenisasi, *stemming, stop words, bag of words* dan

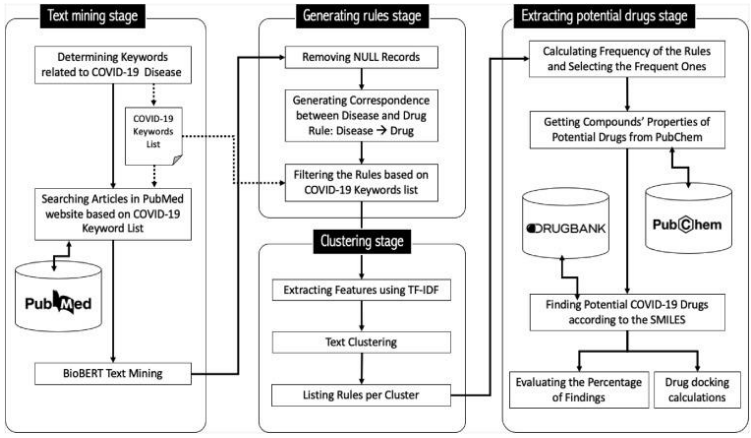
TF-IDF. Secara konsep, *text mining* melibatkan beberapa langkah utama, yakni:

1. Pengumpulan data
   Mengumpulkan data teks dari berbagai sumber seperti artikel berita, posting media sosial, ulasan pelanggan, email, dan dokumen bisnis.
2. Pra-pemrosesan teks
   Membersihkan dan menyiapkan data teks untuk analisis lebih lanjut. Langkah ini mencakup tokenisasi, penghapusan *stop words, Stemming dan Lemmatization,* normalisasi teks (*to lower case / to upper case*)
3. Transformasi teks
   Mengubah data teks menjadi representasi yang dapat digunakan oleh algoritma pembelajaran mesin. Teknik transformasi termasuk: *Bag of Words,* TF-IDF
4. Pemodelan dan Analisis
   Menggunakan teknik statistik dan pembelajaran mesin untuk menganalisis data teks dan mengekstrak informasi yang berguna. Teknik yang sering digunakan termasuk:
5. Klasifikasi: Mengelompokkan teks ke dalam kategori yang telah ditentukan sebelumnya. Menurut Rasjid & Setiawan (2017) terdapat 6 model klasifikasi yang dapat digunakan dalam *text mining* dalam konsep *information retrieval (IR)*. Model yang dikatakan optimal, merupakan model yang dapat mengumpulkan informasi, penarikan kesimpulan otomatis dan mengklasifikan informasi tersebut. Oleh karena itu, 6 model ini dapat digunakan dalam *text mining*.



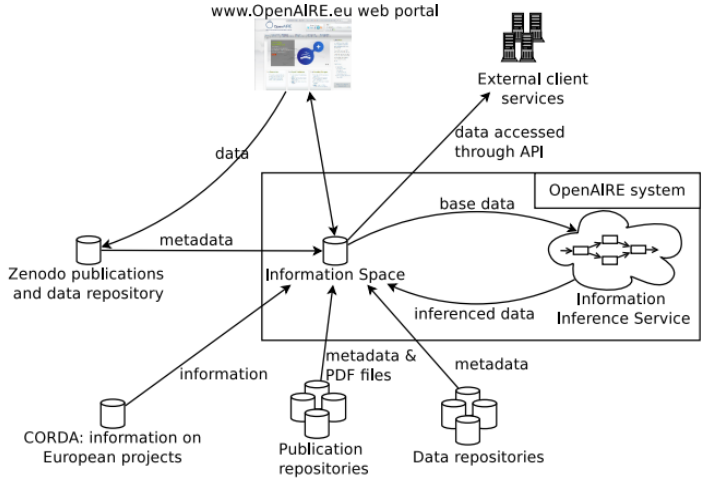
Gambar 42. Model yang dapat digunakan dalam klasifikasi *text*
Sumber: (Rasjid & Setiawan, 2017)

6. Klasterisasi: Mengelompokkan dokumen berdasarkan kesamaan mereka tanpa kategori yang telah ditentukan (Supianto et al., 2023). Melakukan klasterisasi untuk memilih obat yang sesuai dengan penyakit. Klasterisasi yang dilakukan dengan cara memilih dan memilah kesamaan penyakit menggunakan TF-IDF dan membuat korelasi antara asosiasi dari penyakit-obat. Maka akan terbentuk klaster obat – penyakit, tidak menutup kemungkinan juga satu obat berada dalam klaster penyakit yang berbeda.



Gambar 43. Ilustrasi metode klasterisasi teks dalam asosiasi obat – penyakit
Sumber: (Supianto et al., 2023)

7. Ekstraksi Entitas Bernama (*Named Entity Recognition*, NER): Mengidentifikasi dan mengkategorikan entitas dalam teks seperti nama orang, organisasi, dan lokasi.
8. Analisis Sentimen: Menilai emosi atau sentimen yang diekspresikan dalam teks, apakah positif, negatif, atau netral



Gambar 44. Ilustrasi alur data teks yang disimpan dan juga tipe data serta klasifikasinya
Sumber: (Kobos et al., 2014)

Untuk memahami lebih dalam bagaimana text mining diterapkan dalam praktik. Berikut beberapa contoh implementasinya:

a. Ekstraksi informasi dari artikel berita

   Sebuah organisasi penelitian ingin mengekstrak informasi penting dari ribuan artikel berita tentang perubahan iklim. Langkah yang dapat diterapkan untuk menyelesaikan kasus ini sebagai berikut:

   1) Mengumpulkan artikel berita dari berbagai sumber
   2) Melakukan pra-pemrosesan teks untuk membersihkan data
   3) Menggunakan teknik ekstraksi entitas bernama (NER) untuk mengidentifikasi entitas seperti nama negara, organisasi lingkungan, dan tokoh penting
   4) Menggunakan klasterisasi untuk mengelompokkan artikel berdasarkan topik yang berkaitan dengan perubahan iklim

   Hasil yang didapat dengan cepat untuk mengidentifikasi tren dan pola dalam peliputan media mengenai perubahan iklim dan menyusun laporan yang memberikan wawasan mendalam.

8. Klasifikasi *e-mail spam*

   Penyedia layanan email ingin mengklasifikasikan email masuk sebagai *spam* atau bukan *spam.*

   Adapun Langkah yang dapat diterapkan sebagai berikut:

a. Mengumpulkan dataset email yang telah diberi label sebagai spam atau bukan spam
b. Melakukan pra-pemrosesan teks untuk membersihkan dan menyiapkan data
c. Menggunakan teknik pembelajaran mesin seperti *Naive Bayes* atau *Support Vector Machine* (SVM) untuk membangun model klasifikasi.
d. Menerapkan model pada email masuk untuk mengklasifikasikan mereka sebagai spam atau bukan spam

Diharapkan setelah melakukan Langkah tersebut, layanan email dapat secara efektif mem*filter email spam* dan

meningkatkan pengalaman pengguna dengan mengurangi jumlah *email* tidak diinginkan yang mencapai kotak masuk pengguna

Seperti yang telah dibahas pada kasus awal dalam bab ini, setiap model yang dibangun ataupun *dataset* yang tersedia memiliki kemungkinan memiliki kekurangan dataset, *natural outlier*, distribusi data yang tidak sesuai dengan model dan masih banyak lagi. Dalam kasus awal, *error* yang terjadi dikarenakan kekurangan dataset, namun terdapat beberapa Langkah untuk peningkatan model.

Selain *cross validation,* terdapat beberapa metode lain untuk optimalisasi model, yakni:

1) Mengumpulkan Lebih Banyak Data: *Dataset* yang lebih besar dan lebih beragam akan membantu model mempelajari pola yang lebih baik dan meningkatkan generalisasi.
2) Mencoba Berbagai Algoritma: Selain *Naive Bayes*, algoritma lain seperti *Logistic Regression, Support Vector Machine* (SVM), dan *Random Forest* dapat dicoba untuk melihat apakah mereka memberikan performa yang lebih baik.
9. Penggunaan Preprocessing yang lebih canggih:
    - *Lemmatization*: Menggunakan *lemmatization* alih-alih *stemming* untuk memetakan kata ke bentuk dasar mereka.
    - *N-grams*: Menggunakan n-grams (misalnya, bigrams) dalam TF-IDF untuk menangkap lebih banyak konteks.
    - *Hyperparameter Tuning*: Menggunakan teknik seperti *Grid Search* atau *Random Search* untuk menemukan kombinasi *hyperparameter* yang optimal. Teoritis *hyperparameter tuning* menempatkan variabel β (*symmetric Dirichlet)* lainnya, variabel β bertugas untuk *smooth* distribusi kalimat pada klaster topik yang tersedia sebelum kumpulan kalimat tersebut diobservasi melalui model yang dipilih (STELLA et al., n.d.). Secara umum, *hyperparametric tuning* dapat menggunakan konfigurasi dimana α = jumlah topik dan β = ukuran kosa kata dengan nilai awal α = 50 / T dan β = 0.01

- *Text mining* secara tidak langsung digunakan untuk mencari informasi dari data teks tidak terstruktur, untuk meningkatkan hasil pencarian dengan mengekstrak konteks dan makna dari kueri pencarian. Dengan demikian, metode *text mining* dapat digunakan pada mesin pencarian untuk memberikan informasi yang dikehendaki. Mesin pencarian adalah sistem perangkat lunak yang dirancang untuk mencari informasi di *internet*. Cara kerja mesin pencarian dengan mengindeks konten dari berbagai sumber dan menggunakan algoritma untuk mengembalikan hasil yang paling relevan berdasarkan kueri pengguna.

12.3 Mesin Pencarian

Mesin pencarian adalah alat yang sangat penting dalam dunia digital yang digunakan untuk mencari informasi dari sejumlah besar data yang tersedia secara online. *Text mining* adalah proses mengekstraksi informasi berguna dari teks. Gabungan kedua teknologi ini memungkinkan pencarian informasi lebih efektif dan efisien. Berikut adalah beberapa komponen utama dan teknologi yang digunakan dalam mesin pencarian.

a. *Crawler (Web Spider)*

Crawler adalah program yang mengunjungi situs *web* secara otomatis dan mengumpulkan data dari halaman-halaman web. Data ini kemudian disimpan dalam indeks yang akan digunakan untuk pencarian.

a. Indeksing

   Indeksing adalah proses mengorganisir data yang dikumpulkan oleh *crawler* sehingga dapat dengan cepat diakses oleh mesin pencarian. Proses ini melibatkan pra pemrosesan teks.

b. Peringkat Dokumen (*Document Ranking*)

Mesin pencarian menggunakan algoritma untuk menentukan urutan hasil pencarian berdasarkan relevansi. Algoritma ini biasanya mencakup:

- TF-IDF (*Term Frequency-Inverse Document Frequency*): Mengukur seberapa sering sebuah kata muncul dalam dokumen dan seberapa unik kata tersebut dalam kumpulan dokumen.
- *PageRank*: Algoritma yang digunakan oleh Google untuk menentukan pentingnya halaman web berdasarkan jumlah dan kualitas tautan yang mengarah ke halaman tersebut.
- *Machine Learning Models*: Model pembelajaran mesin yang dilatih untuk memprediksi relevansi dokumen berdasarkan berbagai fitur (misalnya, frekuensi kata, klik pengguna, waktu tinggal).

12.4 Pemrosesan Bahasa Alami (*Natural Language Processing - NLP*)

NLP adalah teknologi yang memungkinkan mesin memahami dan memproses bahasa manusia. Beberapa teknik NLP yang sering digunakan dalam mesin pencarian termasuk:

1. *Named Entity Recognition* (NER): Mengidentifikasi dan mengkategorikan entitas yang disebutkan dalam teks (seperti nama orang, tempat, organisasi).
2. *Sentiment Analysis*: Menentukan sentimen (positif, negatif, netral) dari teks.
3. *Topic Modeling*: Mengidentifikasi topik utama yang dibahas dalam kumpulan dokumen.
    - Pencarian Semantik (*Semantic Search*). Pencarian semantik berusaha memahami konteks dan makna dari *query* pengguna untuk memberikan hasil yang lebih relevan. Teknologi yang digunakan dalam pencarian semantik meliputi:

- *Knowledge Graph*: Basis data yang menghubungkan entitas dan konsep untuk memahami hubungan antar mereka.
- *Latent Semantic Analysis* (LSA): Teknik statistik untuk menganalisis hubungan antara kata-kata dalam teks untuk memahami makna kontekstual.
- *Word Embeddings*: Representasi kata dalam bentuk vektor yang menangkap makna kata berdasarkan konteks penggunaannya (misalnya, Word2Vec, GloVe).
- Pencarian Berbasis Pembelajaran Mesin (*Machine Learning-Based Search*). Mesin pencarian modern sering menggunakan model pembelajaran mesin untuk meningkatkan hasil pencarian. Model ini dilatih dengan data yang mencakup *query* pengguna dan interaksi mereka dengan hasil pencarian sebelumnya untuk memprediksi relevansi dokumen di masa depan.

a. Evaluasi dan Pengoptimalan Mesin Pencarian
   Untuk memastikan mesin pencarian memberikan hasil yang relevan dan berguna, evaluasi terus menerus dan pengoptimalan diperlukan. Teknik yang digunakan termasuk:
   - A/B Testing: Menguji dua versi algoritma atau antarmuka untuk melihat mana yang memberikan hasil optimal.
   - *Relevance Feedback*: Menggunakan umpan balik pengguna untuk meningkatkan algoritma pencarian.
   - *Click-Through Rate* (CTR): Mengukur berapa banyak pengguna yang mengklik hasil pencarian dibandingkan dengan jumlah total yang melihat hasil tersebut.

b. Algoritma Peringkat Mesin Pencarian
   Algoritma peringkat mesin pencarian menggunakan berbagai faktor untuk menentukan relevansi dan kualitas halaman web, termasuk:
   - Kata Kunci (*Keywords*): Keberadaan dan kepadatan kata kunci yang relevan dengan kueri pencarian.

- Kualitas Konten: Konten yang informatif, unik, dan bernilai tinggi.
- Tautan Balik (*Backlinks*): Jumlah dan kualitas tautan dari situs *web* lain yang mengarah ke halaman tersebut.
- Pengalaman Pengguna (*User Experience*): Faktor seperti kecepatan halaman, kemudahan navigasi, dan desain responsif.

c. SEO (Search Engine Optimization) adalah praktik mengoptimalkan halaman *web* untuk meningkatkan peringkat mereka di hasil pencarian. Beberapa teknik SEO meliputi:
   - Optimalisasi *On-Page*: Meningkatkan elemen-elemen pada halaman seperti meta tag, header, dan struktur URL.
   - Optimalisasi *Off-Page*: Meningkatkan visibilitas halaman melalui link *building* dan aktivitas media sosial.
   - Teknikal SEO: Meningkatkan aspek teknis dari sebuah situs web, seperti kecepatan loading dan keamanan.

Disamping dari manfaat yang diberikan oleh *teks mining* dan mesin pencarian, berikut beberapa tantangan yang dihadapi dalam *text mining* dan mesin pencarian meliputi:

- Pemrosesan Bahasa Alami (NLP): Memahami konteks dan makna dari bahasa manusia yang kompleks dan bervariasi.
- Big Data: Mengelola dan menganalisis volume data teks yang sangat besar.
- Keamanan dan Privasi: tantangan dalam melindungi data pengguna saat melakukan analisis teks dan pencarian.

Tren masa depan mencakup peningkatan kemampuan AI dalam memahami dan memproses bahasa alami, serta penggunaan *text mining* untuk analitik prediktif dan pengambilan keputusan yang lebih cerdas. Dengan memahami dasar-dasar *text mining* dan mesin pencarian, kita dapat memanfaatkan kekuatan data teks untuk berbagai aplikasi praktis, serta terus mengikuti perkembangan teknologi untuk meningkatkan efisiensi dan efektivitas dalam pengolahan informasi.

Daftar Pustaka

Admin. (2023). Apa Itu Pemodelan Data? Aws. https://aws.amazon.com/id/what-is/data-modeling/

Admin. (2024). Introduction to Machine Learning. AI ML Analytics. https://ai-ml-analytics.com/introduction-to-machine-learning-blog-1/

Aggarwal, C. C. (2015). Data Mining. Springer International Publishing. https://doi.org/10.1007/978-3-319-14142-8

Al Islami, H. (2024). Implementasi Data Mining Untuk Pengelompokan Aset Perusahaan Pada Regional Office PT. NAK Dengan Metode K-Means Clustering. OKTAL: Jurnal Ilmu Komputer Dan Sains, 3(04), 937–945.

Alghifari, M. R., & Wibowo, A. P. (2019). Penerapan metode k-Nearest Neighbor untuk klasifikasi kinerja satpam berbasis web. Jurnal Teknologi Dan Manajemen Informatika, 5(1).

Andriana, D., Irawan, E., & Sormin, R. K. (2022). Implementasi Algoritma K-Medoids pada Pengelompokkan Keragaman Kelompok Tani. FATIMAH: Penerapan Teknologi Dan Sistem Komputer, 1(1), 1–10.

Anggarwal, C. C. (2019). Data Mining text book. In Statistical Field Theor (Vol. 53, Issue 9).

Annur, C. M. (2024). Ini Media Sosial Paling Banyak Digunakan di Indonesia Awal 2024. Databoks. https://databoks.katadata.co.id/datapublish/2024/03/01/ini-media-sosial-paling-banyak-digunakan-di-indonesia-awal-2024

Anwar, M. T., & Permana, D. R. A. (2021). Perbandingan Performa Model Data Mining untuk Prediksi Dropout Mahasiswa. Jurnal Teknologi Dan Manajemen, 19(2), 87–94.

Ardilla, Y. et al. (2021) DATA MINING DAN APLIKASINYA. Penerbit Widina. Available at: https://books.google.co.id/books?id=53FXEAAAQBAJ.

Ardilla, Y., Manuhutu, A., Ahmad, N., Hasbi, I., Manuhutu, M. A., Ridwan, M., & Wardhani, A. K. (2021). DATA MINING DAN APLIKASINYA. Penerbit Widina.

Asroni, A., Indrawan, G., & Dewi, L. J. E. (2023). Implementasi hirarki dataset dalam membangun model language aksara Bali menggunakan framework Tesseract OCR. Jurnal RESISTOR (Rekayasa Sistem Komputer), 6(1), 20–28.

Bahri, S., & Lubis, A. (2020). Metode Klasifikasi Decision Tree Untuk Memprediksi Juara English Premier League. Jurnal Sintaksis, 2(1), 63–70.

Baihaqie, M. R. Q. (2023). Analisis Perilaku Konsumen pada Usaha Ritel dengan menggunakan Metode Association Rule-Market Basket Analysis dan Clustering sebagai Usulan Strategi Peningkatan Penjualan (Studi Kasus: Intimart Gedongan). Universitas Islam Indonesia.

Batista, G. E. A. P. A., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence, 17(5–6), 519–533. https://doi.org/10.1080/713827181

Bruce, P., Bruce, A., Gedeck, P., & Safari, an O. M. Company. (n.d.). Practical Statistics for Data Scientists, 2nd Edition.

Chakraborty, Dr. G., Pagolu, M., & Garla, S. (2014). Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS. SAS Institute.

Cowie, M. R., Blomster, J. I., Curtis, L. H., Duclaux, S., Ford, I., Fritz, F., Goldman, S., Janmohamed, S., Kreuzer, J., Leenay, M., Michel, A., Ong, S., Pell, J. P., Southworth, M. R., Stough, W. G., Thoenes, M., Zannad, F., & Zalewski, A. (2017). Electronic health records to facilitate clinical research. In Clinical Research in Cardiology (Vol. 106, Issue 1). Dr. Dietrich Steinkopff Verlag GmbH and Co. KG. https://doi.org/10.1007/s00392-016-1025-6

Damanik, I. I. P., Solikhun, S., Saragih, I. S., Parlina, I., Suhendro, D., & Wanto, A. (2019). Algoritma K-Medoids untuk Mengelompokkan Desa yang Memiliki Fasilitas Sekolah di Indonesia. Prosiding Seminar Nasional Riset Information Science (SENARIS), 1, 520–527.

Diantoni, C., Mufidah, R., & Triana, H. (2024). MEMBANGUN CHATBOT UNTUK INFORMASI MAGANG DAN STUDI INDEPENDEN KAMPUS MERDEKA DENGAN ALGORITMA NAIVE BAYES. JATI (Jurnal Mahasiswa Teknik Informatika), 8(2), 1389–1397.

Diyasa, I. G. S. M., & Saputra, W. S. J. (2022). Sperm Cell Classification System Carrying X or Y Chromosome In Human With CNN Algorithm. 2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), 360–364.

Dwi, L. (2023). Perbandingan Performa Model Prediksi Customer Churn Berbasis Machine Learning Pada Fashion E-Commerce.

EDY IRWANSYAH, S.T., M. S. (n.d.). CLUSTERING. Binus Unersity. https://socs.binus.ac.id/2017/03/09/clustering/

Fawcett, T. (2013). Data Science for Business. https://www.researchgate.net/publication/256438799

Firdaus, A., & Firdaus, W. I. (2021). Text Mining Dan Pola Algoritma Dalam Penyelesaian Masalah Informasi : (Sebuah Ulasan). In Jurnal JUPITER (Vol. 13, Issue 1).

García, S., Luengo, J., & Herrera, F. (2014). Intelligent Systems Reference Library 72 Data Preprocessing in Data Mining. http://www.springer.com/series/8578

Ghaniy, R., & Indriyaningsih, F. (2020). Penerapan Metode Fuzzy C-Means dalam Pemilihan Program Studi Mahasiswa Baru di Perguruan Tinggi. Teknois: Jurnal Ilmiah Teknologi Informasi Dan Sains, 10(2), 19–30.

Gonzalez, G., Cohen, K. B., Greene, C. S., Kann, M. G., Leaman, R., Shah, N., & Ye, J. (2013). TEXT AND DATA MINING FOR BIOMEDICAL DISCOVERY HHS Public Access. In Pac Symp Biocomput.

Gumilang, J. R. (2020). Implementasi Algoritma Apriori Untuk Analisis Penjualan Konter Berbasis Web. Jurnal Informatika Dan Rekayasa Perangkat Lunak, 1(2), 226–233.

Han, J., Kamber, M., & Pei, J. (2011). Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems). https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

Han, J., Pei, J., & Tong, H. (2022). Data mining: concepts and techniques. Morgan kaufmann.

Handayanto, R. T. (2020). Cross Validation dengan Scikit-Learning Python. https://rahmadya.com/2020/04/17/cross-validation-dengan-scikit-learning-python/

Hin, R. (2019). Decison Tree / Pohon Keputusan. https://medium.com/@revahindami/decison-tree-pohon-keputusan-66f7dd2f05e4

Imania, K. A. N., & Bariah, S. K. (2019). Rancangan pengembangan instrumen penilaian pembelajaran berbasis daring. Petik: Jurnal Pendidikan Teknologi Informasi Dan Komunikasi, 5(1), 31–47.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651–666.

Jannah, E. R. (2024). KLATERISASI DATA PENDUDUK BERDASARKAN PEKERJAAN MENGGUNAKAN METODE K-MEANS PADA WILAYAH JAWA BARAT. JATI (Jurnal Mahasiswa Teknik Informatika), 8(2), 1709–1716.

Kao, A., & Poteet, S. R. (2018). Natural Language Processing and Text Mining.

Kharis, S. A. A., & Zili, A. H. A. (2022). Learning Analytics dan Educational Data Mining pada Data Pendidikan. Jurnal Riset Pembelajaran Matematika Sekolah, 6(1), 12–20.

Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit Second Edition The Complete Guide to Dimensional Modeling.

Kobos, M., Bolikowski, Ł., Horst, M., Manghi, P., Manola, N., & Schirrwagen, J. (2014). Information inference in scholarly communication infrastructures: the OpenAIREplus project experience. Procedia Computer Science, 38, 92–99.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Learning. https://www.researchgate.net/publication/228084519

Kotu, V., & Deshpande, B. (2014). Predictive analytics and data mining: concepts and practice with rapidminer. Morgan Kaufmann.

Kristanto, I. H. (1994). Konsep & Perancangan Database. Penerbit Andi.

Kulsum, U., Nurfitriani, N., Sunarto, S., & Heriyanto, H. (2024). PENGEMBANGAN MODEL EVALUASI KINERJA SDM BERBASIS BALANCED SCORECARD: STUDI PADA KANTOR AKUNTAN PUBLIK DI KOTA SAMARINDA. JURNAL ILMIAH M-PROGRESS, 14(2), 361–372.

Kwartler, T. (2017). Text Mining in Practice with R.

Larose, D. T., & Larose, C. D. (2014). Discovering knowledge in data: an introduction to data mining (Vol. 4). John Wiley & Sons.

Leidiana, H. (2013). Penerapan algoritma k-nearest neighbor untuk penentuan resiko kredit kepemilikan kendaraan bemotor. PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic, 1(1), 65–76.

luthfi, E. and Amikom, U. (2009) Algoritma Data Mining. Penerbit Andi. Available at: https://books.google.co.id/books?id=-Ojclag73O8C.

luthfi, E., & Amikom, U. (2009). Algoritma Data Mining. Penerbit Andi.

Manullang, M. B. (2020). Sistem Informasi Pemesanan Wisata Berbasis Web Pada Cantigi Camp. Universitas Komputer Indonesia.

Mardi, Y. (2017). Data mining: Klasifikasi menggunakan algoritma c4. 5. Jurnal Edik Informatika Penelitian Bidang Komputer Sains Dan Pendidikan Informatika, 2(2), 213–219.

Martin, J. (2019). Speech and Language Processing_ An Introduction to Natural Language Processing.

Maulid, R. (n.d.). Kenali Cross Validation dalam Model Machine Learning. DqLab. https://dqlab.id/kenali-cross-validation-dalam-model-machine-learning

Mauliza, R. N., & Sipayung, Y. R. (2024). Penerapan Text Mining Dalam Menganalisis Pendapat Masyarakat Terhadap Pemilu 2024 Pada Media Sosial X Menggunakan Metode Naive Bayes: Application of Text Mining in Analyzing Public Opinions on the 2024 Election on Social Media X Using the Naive Bayes Method. Technomedia Journal, 9(1), 1–16.

Milenia, E. (2022). SENTIMENT ANALYSIS SATU TAHUN KEPEMIMPINAN WALIKOTA BANDARLAMPUNG PERIODE 2021-2026 MENGGUNAKAN ALGORITME SUPPORT VECTOR MACHINE (SVM) DAN NAÏVE BAYES.

Mining, W. I. D. (2006). Data mining: Concepts and techniques. Morgan Kaufinann, 10(559–569), 4.

Mount, J., & Zumel, N. (2019). Practical Data Science with R. https://books.google.co.id/books?hl=en&lr=&id=ozkzEAAAQBAJ&oi=fnd&pg=PT16&dq=Practical+Data+Science+with+R+oleh+Nina+Zumel+dan+John+Moun&ots=KVUQWYPJrm&sig=PMsbeujIaZgqtRYKJwnzK2Yxoak&redir_esc=y#v=onepage&q=Practical%20Data%20Science%20with%20R%20oleh%20Nina%20Zumel%20dan%20John%20Moun&f=false

Munandar, T. A. (2022). Penerapan Algoritma Clustering Untuk Pengelompokan Tingkat Kemiskinan Provinsi Banten. JSiI (Jurnal Sistem Informasi), 9(2), 109–114.

Munthe, I. R., & Juledi, A. P. (2021). Implementasi Data Mining Algoritma Apriori untuk Meningkatkan Penjualan. Jurnal Teknik Informatika UNIKA Santo Thomas, 6(1), 188–197.

Murphy, K. P. (n.d.). Machine learning : a probabilistic perspective.

Nugroho, A., & Amrullah, A. (2023). Evaluasi Kinerja Algoritma K-Nn Menggunakan K-Fold Cross Validation Pada Data Debitur Ksp Galih Manunggal. Jurnal Informatika Teknologi Dan Sains (JINTEKS), 5(2), 294–300.

Pandensolang, F. (2022). Implementasi Business Intelligence Untuk Analisa dan Visualisasi Perbandingan Perencanaan dan Realisasi Anggaran pada BNNP Sulawesi Utara.

Panjaitan, W. T. B., Utami, E., & Al Fatta, H. (2018). Prediksi Panen Padi Menggunakan Algoritma K-Nearest Neigbour. SNATIF, 5(1).

Pratama, E. F. A., Khairil, K., & Jumadi, J. (2022). Implementasi Metode K-Means Clustering Pada Segmentasi Citra Digital. Jurnal Media Infotama, 18(2), 291–301.

Prehanto, D. R., Kom, S., & Kom, M. (2020). Buku Ajar Model Sistem Pendukung Keputusan Dengan AHP dan IPMS. Scopindo Media Pustaka.

Putra, R. F., Mukhlis, I. R., Datya, A. I., Pipin, S. J., Reba, F., Al-Husaini, M., Mandowen, S. A., Zain, N. N. L. E., & Judijanto, L. (2024). Algoritma Pembelajaran Mesin: Dasar, Teknik, dan Aplikasi. PT. Sonpedia Publishing Indonesia.

Putra, R. F., Zebua, R. S. Y., Budiman, B., Rahayu, P. W., Bangsa, M. T. A., Zulfadhilah, M., Choirina, P., Wahyudi, F., & Andiyan, A. (2023). Data Mining: Algoritma dan Penerapannya. PT. Sonpedia Publishing Indonesia.

Qoniah, I., & Priandika, A. T. (2020). Analisis Market Basket Untuk Menentukan Asossiasi Rule Dengan Algoritma Apriori (Studi Kasus: Tb. Menara). Jurnal Teknologi Dan Sistem Informasi, 1(2), 26–33.

Rahayu, P. W., Sudipa, I. G. I., Suryani, S., Surachman, A., Ridwan, A., Darmawiguna, I. G. M., Sutoyo, M. N., Slamet, I., Harlina, S., &

Maysanjaya, I. M. D. (2024). Buku Ajar Data Mining. PT. Sonpedia Publishing Indonesia.

Ramadhanti, N. R., & Mariyah, S. (2019). Document Similarity Detection Using Indonesian Language Word2vec Model.

Ranti, S. (2023). Pengertian dan Siklus Pemrosesan Data Pada Komputer. Kompas.Com. https://tekno.kompas.com/read/2023/02/19/03000017/pengertian-dan-siklus-pemrosesan-data-pada-komputer-?page=all

Rasjid, Z. E., & Setiawan, R. (2017). Performance comparison and optimization of text document classification using k-NN and naïve bayes classification techniques. Procedia Computer Science, 116, 107–112.

Richards, T. (2021). Getting Started with Streamlit for Data Science: Create and deploy Streamlit web applications from scratch in Python. Packt Publishing Ltd.

Rizki, D. (2020). Visualisasi Data Sentimen Terhadap Organisasi Perangkat Daerah Pemerintah Provinsi Jawa Barat Di Jabar Digital Service. (Doctoral Dissertation, Universitas Komputer Indonesia), 7–30.

Sadat, M. A., Pujiono, P., Pambudi, A., & Ibad, S. (2023). COMPARISON OF ALGORITHM BETWEEN CLASSIFICATION & REGRESSION TREES AND SUPPORT VECTOR MACHINE IN DETERMINING STUDENT ACCEPTANCE IN STATE UNIVERSITIES. Jurnal Teknik Informatika (Jutif), 4(6), 1589–1604.

Setio, P. B. N., Saputro, D. R. S., & Winarno, B. (2020). Klasifikasi Dengan Pohon Keputusan Berbasis Algoritme C4. 5. PRISMA, Prosiding Seminar Nasional Matematika, 3, 64–71.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft

Computing, 97(xxxx), 105524. https://doi.org/10.1016/j.asoc.2019.105524

Spann, D. D. (2014). Fraud Analytics: Strategies and Methods for Detection and Prevention. John Wiley & Sons.

STELLA, F., ZANKER, M., & SOTTOCORNOLA, G. (n.d.). Experts Recommendation with Probabilistic Topic Models.

Sudarsono, B. G., Leo, M. I., Santoso, A., & Hendrawan, F. (2021). Analisis Data Mining Data Netflix Menggunakan Aplikasi Rapid Miner. JBASE-Journal of Business and Audit Information Systems, 4(1).

Sugiana, N. S. S., & Musty, B. (2023). Analisis Data Sistem Informasi Monitoring Marketing; Tools Pengambilan Keputusan Strategic. Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi, 12(2), 696–708.

Sundari, S., Damanik, I. S., Windarto, A. P., Tambunan, H. S., Jalaluddin, J., & Wanto, A. (2019). Analisis K-Medoids Clustering Dalam Pengelompokkan Data Imunisasi Campak Balita di Indonesia. Prosiding Seminar Nasional Riset Information Science (SENARIS), 1, 687–696.

Supianto, A. A., Nurdiansyah, R., Weng, C.-W., Zilvan, V., Yuwana, R. S., Arisal, A., Pardede, H. F., Lee, M.-M., Huang, C.-H., & Ng, K.-L. (2023). Cluster-based text mining for extracting drug candidates for the prevention of COVID-19 from the biomedical literature. Journal of Taibah University Medical Sciences, 18(4), 787–801.

Tan, P.-N., Steinbach, M., & Kumar, V. (2014). Introduction to data mining.

Tank, D. (2014) 'Improved Apriori Algorithm for Mining Association Rules', International Journal of Information Technology and Computer Science, 6, pp. 15–23. doi: 10.5815/ijitcs.2014.07.03.

Tank, D. (2014). Improved Apriori Algorithm for Mining Association Rules. International Journal of Information Technology and Computer Science, 6, 15–23. https://doi.org/10.5815/ijitcs.2014.07.03

Tiwari, M., & Singh, R. (2012). Comparative investigation of k-means and k-medoid algorithm on iris data. International Journal of Engineering Research and Development, 4(8), 69–72.

Widaningsih, S. (2022). Penerapan Data Mining untuk Memprediksi Siswa Berprestasi dengan Menggunakan Algoritma K Nearest Neighbor. JATISI (Jurnal Teknik Informatika Dan Sistem Informasi), 9(3), 2598–2611.

Wira, B., Budianto, A. E., & Wiguna, A. S. (2019). Implementasi Metode K-Medoids Clustering Untuk Mengetahui Pola Pemilihan Program Studi Mahasiswa Baru Tahun 2018 Di Universitas Kanjuruhan Malang. Rainstek: Jurnal Terapan Sains & Teknologi, 1(3), 53–68.

Witten, Ian H; Frank, Eibe; Hall, Mark A; Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.

Xiong, M., Wang, H., Che, C., & Sun, M. (2024). Application of text mining and coupling theory to depth cognition of aviation safety risk. Reliability Engineering & System Safety, 245, 110032.

Xu, R., Xu, J., & Wunsch, D. C. (2010). Clustering with differential evolution particle swarm optimization. IEEE Congress on Evolutionary Computation, 1–8.

Yahya, S. T. (2022). Data Mining. CV Jejak (Jejak Publisher).

Biodata Penulis

Dr. Rinci Kembang Hapsari, S.Si., M.Kom

Lahir di Tulungagung pada tahun 1977, alumni dari SMADA Tulungagung angkatan 1996. Lulus S1 dari Jurusan Matematika bidang minat Informatika, Institut Teknologi Sepuluh November Surabaya pada tahun 2001. Lulus S2 dari Jurusan Teknik Informatika, Institut Teknologi Sepuluh November, pada tahun 2008. dan menyelesaikan Doktoral di Universitas Airlangga pada tahun 2022. Aktif sebagai dosen sejak tahun 2001 hingga saat ini.

Penulis memiliki kepakaran dibidang Data Mining, *Artificial Intelligence, Image Processing* dan Optimasi. Dalam rangka mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Penulis telah mempublikasikan banyak karya ilmiah dalam jurnal tingkat nasional dan internasional, serta seminar ilmiah baik tingkat nasional maupun internasional. Juga aktif sebagai reviewer jurnal bereputasi tingkat nasional dan internasional. Penulis mulai aktif membuat dan mengunggah video pembelajaran di youtube, dengan channel RKH-Tebar Ilmu. Video pembelajaran diharapkan dapat membantu rekan-rekan mahasiswa dalam memahami dan memperdalam materi perkuliahan secara mandiri. Penulis memiliki komitmen untuk memberikan pendidikan yang berkualiatas untuk kemajuan negeri.

e-mail Penulis: rincikembang@itats.ac.id

Titin Sumarni

Lahir tanggal 25 Mei 1981di Kabupaten Manna, Bengkulu Selatan. Pendidikan Formal jenjang S1 yaitu pada Fukaltas Ilmu Komputer Jurusan Sistem Informasi Universitas Putra Indonesia (UPI) Padang Sumatra Barat. S2 Pasca sarjana di Universitas Putra Indonesia (UPI) Padang dengan Fakultas Ilmu Komputer Jurusan Sitem Informasi.

Pengalaman kerja: sebagai Karyawan pada PT Simano Batamindo Batam, Tenaga Pengajar pada SMKN 1 Bengkalis, Dosen di Politeknik Negeri Bengkalis, Dosen di STAIN Bengkalis. Untuk meningkatkan karir seabagai Dosen Profesional penulis aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi Bangsa Indonesia, buku yang di tulis adalah buku-buku referensi pembelajaran dan buku antologi. Penulis juga aktif dalam bidang penelitian baik penelitian mandiri maupun penelitian yang mendapatkan bantuan dana pemerintah.

e-mail penulis: titinijal2005@gmail.com

Blog Peribadi penulis: https://goresan-inspirasi-1.blogspot.com/

Google Scholar:https://scholar.google.co.id/citations?user=IxkQkYEAAAAJ&hl=id

Taufik Hidayatulloh

Anak pertama dari tiga bersaudara. Menyelesaikan Pendidikan formal Pasca Sarjana Ilmu Komputer di STMIK Nusa Mandiri Jakarta, pada tahun 2013. Mengawali karir sebagai asisten dosen pada tahun 2007 di AMIK BSI Sukabumi. Hingga melanjutkan karir saat ini aktif menjadi salah satu dosen di Universitas Bina Sarana Informatika.

Bidang kajian peminatan yang penulis minati dibidang data mining, sistem pakar, kecerdasan buatan. Selain aktif melakukan tri dharma perguruan tinggi, penulis juga aktif sebagai editor dan reviewer di beberapa jurnal nasional terakreditasi.
e-mail Penulis: taufik.tho@bsi.ac.id

Farida

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2003 silam. Hal tersebut membuat penulis memilih untuk kuliah S1 Teknik Informatika kampus ITATS dengan memilih Jurusan Kecerdasan Buatan dan berhasil lulus pada tahun 2009. Penulis pada Tahun 2013 melanjutkan pendidikan S2 di STTS di Surabaya dan berhasil menyelesaikan studi nya di prodi Teknologi Informasi pada tahun 2016.

Selain itu penulis juga aktif di komunitas Data Science Indonesia sebagai pengurus pusat dan Sebagai Founder Data Science Indonesia Regional Jawa Timur. Penulis juga sebagai narasumber terkait Data Science di goverment Indonesia serta Tenaga Ahli DS di provinsi Jawa Timur tahun 2021- 2022. Penulis memiliki kepakaran dibidang Image Processing dan Data Science. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut. Saat ini penulis merupakan Co-Founder dari Tech Company Narasio Data. Saat menjadi Dosen beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Saat ini penulis juga aktif menulis buku,artikel dan sharing knowledge dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini.
e-mail Penulis: farida@narasiodata.com

Herianto

Mulai mengajar sejak tahun 1999. Pendidikan master (S2) ditempuh di Universitas Gadjah Mada (UGM) bidang Sistem Komputer dan Informatika -Teknik Elektro. Mengampu beberapa matakuliah seperti: Pengantar data Science, Artificial Intelligence Analisis Big Data, Data Mining, Perancangan Basis Data, Jaringan Komputer, Pemrograman Internet dan sejumlah mata kuliah dasar Teknologi Informasi lainnya.
Secara Khusus penulis memiliki ketertarikan di bidang Artificial Inteligence dan Data Science. Karir sebagai dosen profesional penulis di bidang data science telah dinyatakan Kompeten melalui Sertikasi BNSP pada tahun 2021. Penulis juga aktif sebagai peneliti di bidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI.
e-mail Penulis: heri.unsada@gmail.com

Febie Elfaladonna

Penulis mulai tertarik dengan ilmu komputer sejak awal masuk perkuliahan di tahun 2011 silam. Saat itu penulis memilih program studi teknik infomatika dan lulus sarjana komputer pada tahun 2015. Tak berhenti disitu penulis kembali melanjutkan study hingga jenjang S2 Teknologi Informasi dan kembali menamatkannya di tahun 2017.
Seluruh jenjang pendidikan S1 dan S2 Penulis di tempuh di Universitas Putra Indonesia “YPTK” Padang. Penulis pernah bekerja pada salah satu Politeknik Swasta yang berada di Cikarang sebelum akhirnya pindah ke Palembang menjadi dosen

di Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya. Selama menjadi dosen penulis melakukan kegiatan pengajaran, penelitian serta pengabdian. Menulis buku adalah sesuatu yang baru dikerjakan oleh penulis. Saat ini penulis sedang aktif menulis dan sudah menerbitkan beberapa book chapter dan buku ajar. Ada beberapa sertifikasi yang sudah penulis ikuti yaitu sertifikasi Animasi, Bahasa Pemrograman Phyton, dan Junior Web Developer, serta Data Science. Penulis tertarik belajar data science dengan mengikuti pelatihan – pelatihan data science. Kepakaran penulis dibidang Data Science, Sistem Penndukung Keputusan, dan Junior Web. Semoga kontribusi penulis dalam membuat buku ini dapat menjadi jalan untuk mengabdi kepada bangsa dan negara, bermanfaat bagi masyarakat luas khususnya siswa atau mahasiswa yang sedang menempuh pendidikan di bidang ilmu Komputer

e-mail Penulis: febie.elfakhrul@gmail.com

Nurul Aini

Ketertarikan penulis terhadap Ilmu Komputer dimulai pada tahun 2005 silam. Hal tersebut membuat penulis lanjut ke perguruan tinggi di UNIVERSITAS DIPA MAKASSAR dengan memilih Jurusan SISTEM INFORMASI dan berhasil lulus pada tahun 2009. Penulis kemudian melanjutkan pendidikan ke studi S2 di prodi Teknik Informatika Program Pasca Sarjana Universitas Hasanuddin dan menyelesaikan studi di tahun 2013.

Penulis memiliki kepakaran dibidang Decision Support System (DSS) dan Data Science. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan

didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Atas dedikasi dan kerja keras dalam menulis buku, Perpustakaan Nasional RI memberikan penghargaan sebagai salah satu Pemenang Buku Terbaik Tahun 2022.
e-mail Penulis: nurulaini.m11@undipa.ac.id

Nurfitria Khoirunnisa

Lahir di Sumedang pada tanggal 11 Maret 1996, penulis lebih dikenal dengan panggilan Icha. Setelah menyelesaikan D4 Teknik Informatika di Politeknik Pos Indonesia, penulis meraih Magister Sistem Informasi dari STMIK Likmi Bandung pada tahun 2019. Dalam perjalanan karirnya, penulis berkesempatan membuktikan diri sebagai System Analyst dan Project Manager di salah satu BUMN Karya.
Setelah berhasil meraih gelar Magister, penulis melanjutkan langkahnya membawa semangat pengabdian ke dunia pendidikan sebagai seorang dosen di Politeknik Negeri Subang.
Penulis memiliki spesialisasi keahlian riset di bidang sistem informasi, yang berfokus pada Analisis Sistem dan Desain, Manajemen Proyek TI, dan Pengembangan Perangkat Lunak. Dalam setiap tindakan dan tulisannya, penulis menunjukkan komitmen tinggi terhadap tridharma perguruan tinggi, membawa semangat pendidikan, penelitian, dan pengabdian pada masyarakat menjadi bagian tak terpisahkan dari perjalanannya menuju kesuksesan.

Heri Kurniawan

Penulis adalah alumni Statistika Universitas Padjadjaran tahun 2004, dan memperolah magister Ilmu Aktuaria ITB pada tahun 2013. Aktif menjadi dosen PNS DPK di ITHB (2005 - 2019), Universitas 'Aisyiyah (Unisa) Bandung (2019-2022) dan Dosen Tamu di Telkom University (2016-2021), mengajar Bidang Manajemen Risiko, Operasi Riset, Statistika Bisnis, Biostatistika, Probabilitas dan Statistika, Statistika Industri dan Metode Penelitian, aktif juga menjadi pembicara untuk pelatihan yang berkaitan dengan Metode Penelitian dan Statistika.

Sejak 2023 sampai dengan sekarang menjadi Dosen Prodi Matematika di Fakultas Sains dan Teknologi, Universitas Terbuka. Di sela kesibukan sebagai dosen, pernah menjadi Pembicara diberbagai workshop yang berkaitan dengan Metode Penelitian (Kualitatif, Kuantitatif dan Mixed Approach), Bimbingan Teknis Bantuan TIK Direktorat SMP dan SMA 2021-2022, Tim teknis PPDB online dibeberapa Dinas Pendidikan provinsi maupun Kota/Kab di Indonesia. Tim Penyusun Telaah Soal Ujian Nasional, Direktorat Pembinaan SMA, Kemdikbud (2016), Kegiatan Penyusunan Kisi-kisi Naskah Soal Kejar Paket A, B, dan C dilingkungan Dinas Pendidikan Kabupaten Bogor (2018), Tim Penyusun Modul Integrasi Pendidikan Anti Korupsi Komisi Pemberantasan Korupsi (KPK) – IndonesiaBermutu (IB) (2017). Karya buku yang sudah dibuat diantaranya adalah: SPSS Complete (2009), Structural Equation Modelling (Belajar lebih mudah mengolah data dengan Lisrel dan PLS (2009), Regresi dan Korelasi dalam genggaman anda (2010), SEM dan PLSM dengan XLStat (2011).

Kraugusteeliana, M.Kom, MM

Menghabiskan masa sekolah SD – SMAN 2 Cilegon (dh/ SMAKS) melanjutkan kuliah Strata 1 (S1) di Univ Budi Luhur, 2000 Melanjutkan S2 bidang Software Engeenering (Informatika) di STTBI Benarif dan mengambil S2 Magister Manajemen kosentrasi Human Capital (SDM) di Universitas Budi Luhur. Aktif mengajar sejak 1999 diberbagai PT di Jakarta namun sejak 2014 bergabung dengan UPN Veteran Jakata sebagai dosen tetap sistem informasi hingga saat ini.

Pengalaman mengajar mengampu matakuliah diantaranya ANSI-Analisa Sistem Informasi Management Sains, Knowledge Management (KM), Sistem Informasi Manajemen, Sistem Penunjang Keputusan, APB-Analisa Proses Bisnis, Rekayasa SI, RPL, SBD (Sistem basis data), TKTI (Tata Kelola Teknologi Informasi), Manajemen Resiko TI, Audit Sistem Informasi, Sistem Enterprise dan juga Manajemen layanan TI. Beberapa buku yang telah diterbitkan antara lain PTI, SIM, Analisa perancangan sistem dan pengembang SI, Manajemen layanan TI, Arsitektur Enterprise, Tata Kelola TI, Kemanan sistem Informasi, E-Govermen, Digitalisasi Goverment, Tata Kelola TI, Literasi Digital dan Digital Tourism. Dalam hal penelitian telah dipublikasikan terkait egovernance, LMS, SPK, SIM, egoverment evaluasi atau Audit system Cobit, digital marketing , VR dan Etourism.

e-mail Penulis: kraugusteeliana@upnvj.ac.id

Lestari Yusuf

Lahir pada bulan Mei 1989 sebagai anak ketiga dari empat bersaudara. Besar di kota Serang, Banten dan melanjutkan pendidikan Formal Pasca Sarjana di Universitas Nusa Mandiri pada akhir tahun 2015. Mulai Berkarir menjadi staff pengajar di AMIK BSI Jakarta dan saat ini menjadi staf juga dosen di Universitas Nusa Mandiri.

Scope bidang penelitian yang dikuasai diantaranya data mining, text mining dan Fuzzy logic. Selain mengajar penulis melakukan aktif melakukan tri dharma perguruan tinggi dan juga aktif sebagai editor dan reviewer di beberapa jurnal nasional terakreditasi.

e-mail Penulis: lestari.lyf@nusamandiri.ac.id

Yulizar Widiatama

Dosen di Departemen Teknik Industri di Departemen Teknik Industri di Universitas Pembangunan Nasional Veteran Jakarta, Jakarta, Indonesia. Beliau memiliki latar belakang akademis yang menggabungkan manajemen industri dan teknik industri, masing-masing diperoleh dari Universiti Utara Malaysia dan Universiti Teknologi Malaysia.

Penelitian utamanya berfokus pada optimalisasi industri dan pembelajaran mesin yang dapat ditafsirkan untuk meningkatkan ketahanan rantai pasokan. Berpengalaman menerbitkan makalah di berbagai konferensi dan jurnal baik publikasi lokal maupun internasional. Beliau juga memiliki pengalaman selama 8 tahun sebagai Dosen untuk Produksi dan Perencanaan dan Pengendalian Persediaan, Manajemen Rantai Pasokan, dan

berbagai mata pelajaran berbasis pemrograman. Selain itu, ia memiliki minat untuk melakukan penelitian di bidang terkait energi terbarukan. Saat ini, beliau memiliki tanggung jawab sebagai Koordinator Laboratorium untuk Perencanaan Produksi dan Pengendalian Persediaan (PPIC). Terutama difokuskan untuk memberi siswa pengalaman real-time menggunakan sistem openERP untuk memecahkan masalah berbasis industri di bidang terkait Produksi Industri. Selanjutnya, PPIC Lab membuka layanannya kepada masyarakat yang berminat untuk mengakui penerapan sistem OpenERP untuk usaha kecil dan menengah mereka.

e-mail: yulizar.w@upnvj.ac.id

Buku ini hasil kolaborasi penulis dalam bentuk buku rampai ini dan memberikan langkah dalam memahami Data Mining serta penerapannya. Pembaca akan memperoleh pengetahuan, analisis dan kemampuan dalam Data Mining. Buku ini menyajikan konsep, analisis dan kasus dari Data Mining. Topik pada buku ini diantaranya:

1. Pengantar Data Mining
2. Preprocessing Data (Manipulasi & Visualisasi)
3. Preprocessing Data (Normalisasi Data)
4. Klasifikasi & K-NN
5. Validasi Model dalam Klasifikasi
6. Decission Tree
7. Association Rule
8. Clustering
9. Analisa Cluster
10. Predictive Mining
11. Konsep Dasar Text Mining
12. Text Mining & Mesin Pencarian

Oleh karena itu, diharapkan dapat membantu para pembaca dalam memahami serta dapat mengimplementasikannya dalam permasalahan dilapangan. Selain itu dapat menjadi rujukan bagi para Dosen, Praktisi, Mahasiswa maupun pihak yang ingin memperdalam Data Mining.

Indie Press
Bersama Mencerdaskan Bangsa
IKAPI
DAPATKAN DI Google Play
ISBN 978-623-8191-63-5
9 786238 191635