DATA MINING

(Memahami Pola di Balik Angka)

Dr. Jonathan Kiwasi Wororomi, S.Si., M.Si. | Felix Reba, S.Si., M.Sc.
Samuel Aleksander Mandowen, S.Si., M.IT. | Alvian M. Sroyer, S.Si., M.Si.
Halomoan Edy Manurung, S,Si., M.Cs. | Mayko Edison Koibur, S.T., M.Eng.
Erna Hudianti Pujiarini, S.Si., M.Si. | Marwan Ramdhany Edy, S.Pd., M.Kom.
Hajiar Yuliana, S.T., M.T. | Mukarramah Yusuf, B.Sc., M.Sc.
Riadi Marta Dinata, S.Ti., M.Kom.

Editor :
Nila Rusiardi Jayanti, M.Kom.

DATA MINING

(Memahami Pola di Balik Angka)

Dalam era di mana data menjadi aset terbesar, keahlian dalam mengeksplorasi dan memahami potensi di balik angka menjadi kunci utama kesuksesan. Buku ini adalah panduan komprehensif yang membimbing pembaca melalui perjalanan menarik dari pemahaman dasar hingga teknik-teknik canggih dalam data mining.

Mulai dari konsep dasar data mining, pembaca akan diperkenalkan dengan berbagai algoritma dan metode yang digunakan untuk mengungkap pola tersembunyi dalam data. Buku ini tidak hanya membahas teknik analisis, tetapi juga fokus pada langkah-langkah penting seperti preprocessing data, pembersihan data, dan integrasi data, yang memastikan bahwa data yang digunakan untuk analisis adalah yang terbaik dari yang terbaik.Dari sini, pembaca akan dibawa ke dalam dunia yang mendalam dari clustering, classification, dan analisis deret waktu, di mana mereka akan mempelajari cara mengelompokkan data, mengklasifikasikan entitas, dan memprediksi tren masa depan. Teknik-teknik inovatif seperti neural networks, ensemble methods, dan text mining juga akan dipelajari, membuka pintu ke wawasan yang lebih dalam dan pemahaman yang lebih kaya tentang data.

Buku ini tidak hanya ditujukan bagi para ahli data, tetapi juga bagi siapa pun yang ingin memahami bagaimana mengubah data mentah menjadi informasi berharga yang dapat digunakan untuk pengambilan keputusan yang cerdas. Dengan bahasa yang jelas dan penjelasan yang mendalam, buku ini akan menjadi sumber daya yang berharga bagi siapa saja yang tertarik untuk menggali harta karun dalam data mereka sendiri. Siapkan diri Anda untuk memulai perjalanan menuju pemahaman yang mendalam tentang dunia data mining yang menarik ini!

eureka
media aksara
Anggota IKAPI
No. 225/JTE/2021

0858 5343 1992
eurekamediaaksara@gmail.com
Jl. Banjaran RT.20 RW.10
Bojongsari - Purbalingga 53362

ISBN 978-623-516-061-0
9 786235 160610

DATA MINING
(MEMAHAMI POLA DI BALIK ANGKA)

Dr. Jonathan Kiwasi Wororomi, S.Si., M.Si.
Felix Reba, S.Si., M.Sc.
Samuel Aleksander Mandowen, S.Si., M.IT.
Alvian M. Sroyer, S.Si., M.Si.
Halomoan Edy Manurung, S,Si., M.Cs.
Mayko Edison Koibur, S.T., M.Eng.
Erna Hudianti Pujiarini, S.Si., M.Si.
Marwan Ramdhany Edy, S.Pd., M.Kom.
Hajiar Yuliana, S.T., M.T.
Mukarramah Yusuf, B.Sc., M.Sc.
Riadi Marta Dinata, S.Ti., M.Kom.

PENERBIT CV.EUREKA MEDIA AKSARA

DATA MINING
(MEMAHAMI POLA DI BALIK ANGKA)

Penulis : Dr. Jonathan Kiwasi Wororomi, S.Si., M.Si.
Felix Reba, S.Si., M.Sc.
Samuel Aleksander Mandowen, S.Si., M.IT.
Alvian M. Sroyer, S.Si., M.Si.
Halomoan Edy Manurung, S,Si., M.Cs.
Mayko Edison Koibur, S.T., M.Eng.
Erna Hudianti Pujiarini, S.Si., M.Si.
Marwan Ramdhany Edy, S.Pd., M.Kom.
Hajiar Yuliana, S.T., M.T.
Mukarramah Yusuf, B.Sc., M.Sc.
Riadi Marta Dinata, S.Ti., M.Kom.
Editor : Nila Rusiardi Jayanti, M.Kom.
Penyunting : Muhammad Syaoqibihillah
Desain Sampul : Eri Setiawan
Tata Letak : Husnun Nur Afifah
ISBN : 978-623-516-061-0
Diterbitkan oleh : **EUREKA MEDIA AKSARA, JULI 2024**
ANGGOTA IKAPI JAWA TENGAH
NO. 225/JTE/2021

Redaksi:
Jalan Banjaran, Desa Banjaran RT 20 RW 10 Kecamatan Bojongsari
Kabupaten Purbalingga Telp. 0858-5343-1992
Surel : eurekamediaaksara@gmail.com
Cetakan Pertama : 2024

KATA PENGANTAR

Dalam dunia yang semakin terhubung dan penuh dengan data, kemampuan untuk menggali wawasan yang berharga dari gumpalan informasi yang tak terhitung jumlahnya menjadi semakin penting. Dengan bangga, kami mempersembahkan buku ini sebagai panduan praktis untuk memahami dan menguasai seni Data Mining. Mulai dari konsep dasar hingga teknik-teknik canggih, buku ini mengeksplorasi berbagai topik penting, termasuk pengenalan Data Mining, algoritma, preprocessing data, integrasi, transformasi, seleksi data, serta berbagai teknik analisis seperti clustering, klasifikasi, jaringan saraf tiruan, metode ensemble, dan analisis deret waktu.

Langkah awal yang esensial dalam perjalanan ini adalah memahami konsep dasar Data Mining. Dalam bab-bab awal, pembaca akan diperkenalkan dengan pengertian, sejarah, dan relevansi praktis dari Data Mining dalam berbagai bidang. Selanjutnya, buku ini membawa pembaca melalui serangkaian algoritma dan metode yang digunakan dalam proses analisis data, serta menjelaskan secara rinci langkah-langkah persiapan data yang diperlukan sebelum memulai analisis.

Dengan kombinasi pengenalan yang kuat dan aplikasi praktis, buku ini bertujuan untuk menjadi panduan yang berguna bagi siapa pun yang tertarik untuk menjelajahi dan memahami potensi tak terbatas dari Data Mining.

Tim Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1 PENGENALAN DATA MINING

Dr. Jonathan Kiwasi Wororomi, S.Si., M.Si.

Pengenalan data mining merupakan langkah awal yang penting dalam memahami konsep dan aplikasi dari analisis data yang komprehensif. Data mining adalah proses ekstraksi pola yang bermakna dari sejumlah besar data dengan menggunakan berbagai teknik statistik, matematika, dan kecerdasan buatan. Tujuan utamanya adalah untuk mengungkapkan informasi yang tersembunyi, pola tersembunyi, dan hubungan yang signifikan dalam data yang kemudian dapat digunakan untuk pengambilan keputusan yang lebih baik. Dalam era di mana data semakin melimpah, pemahaman tentang data mining menjadi semakin penting untuk berbagai bidang seperti bisnis, ilmu pengetahuan, kesehatan, dan lainnya.

A. Tipe Data

Dalam data mining, terdapat berbagai jenis data yang dapat diolah dan dianalisis untuk menemukan pola, tren, dan informasi yang berharga. Beberapa tipe data yang umum digunakan dalam data mining meliputi (Ha *et al*., 2011):

1. Data Numerik. Tipe data yang terdiri dari angka atau nilai numerik. Contoh termasuk data seperti tinggi, berat badan, suhu, dan pendapatan.
2. Data Kategorika. Tipe data ini terdiri dari kategori atau label diskrit yang mewakili karakteristik atau atribut tertentu. Contoh termasuk jenis kelamin (pria/wanita), status

pernikahan (single/menikah/cerai), atau jenis produk (elektronik/pakaian/makanan).

3. Data Teks. Data teks terdiri dari urutan karakter atau kata-kata. Contoh data teks termasuk dokumen teks, ulasan produk, tweet, dan pesan teks.
4. Data spasial atau Geografis. Tipe data ini berhubungan dengan lokasi atau koordinat geografis. Contoh termasuk data peta, koordinat GPS, atau informasi lokasi.
5. Data Gambar. Data gambar terdiri dari representasi visual dari objek atau scene. Ini bisa berupa pixel-level data atau fitur-fitur yang diekstraksi dari gambar.
6. Data Audio. Data audio terdiri dari sinyal suara atau audio. Ini bisa berupa rekaman suara, lagu, atau sinyal audio lainnya.
7. Data Multivaria. Data multivariat terdiri dari beberapa variabel atau atribut yang terkait satu sama lain. Ini bisa berupa kombinasi dari tipe data numerik, kategorikal, atau lainnya.

B. Jenis Data

Dalam data mining, terdapat beberapa jenis data yang dapat dianalisis untuk mengekstrak informasi atau pola yang berharga. Berikut adalah beberapa jenis data yang umum dalam konteks data mining (Ha *et al.*, 2011):

1. Data Terstruktur: Data terstruktur mengacu pada data yang terorganisir dalam format yang jelas dan terdefinisi sebelumnya. Contohnya adalah data dalam basis data relasional, spreadsheet, atau file CSV. Data terstruktur memiliki skema yang didefinisikan dengan baik, yang memungkinkan untuk melakukan query dan analisis dengan mudah.
2. Data Semi-Terstruktur: Data semi-terstruktur memiliki struktur yang lebih longgar daripada data terstruktur tetapi masih mengandung beberapa struktur atau metadata yang terkait.

3. Data Tidak Terstruktur: Data tidak terstruktur adalah data yang tidak memiliki struktur formal atau definisi yang jelas. Ini sering kali dalam bentuk teks bebas, gambar, audio, atau video. Contoh dari data tidak terstruktur termasuk email, tweet media sosial, dokumen teks, dan rekaman suara.
4. Data Spasial: Data spasial adalah data yang berkaitan dengan lokasi atau koordinat geografis. Ini dapat mencakup peta, citra satelit, atau data GPS.
5. Data Multidimensional: Data multidimensional mengacu pada data yang memiliki beberapa dimensi atau atribut. Contoh umumnya adalah data kubus OLAP (*Online Analytical Processing*) yang digunakan dalam analisis bisnis.
6. Data Streaming: Data streaming adalah data yang terus-menerus mengalir dalam waktu nyata. Ini dapat mencakup data sensor, data transaksi keuangan, atau data media sosial yang diperbarui secara terus-menerus.

C. Proses Data Mining

Proses data dalam data mining adalah serangkaian langkah sistematis yang digunakan untuk mengekstrak informasi berharga atau pola yang tersembunyi dari kumpulan data yang besar atau kompleks. Proses ini melibatkan beberapa tahapan yang umumnya diikuti secara berurutan (Witten *et al.*, 2016):

1. Pemahaman Masalah: Tahap awal di mana tujuan dan kebutuhan bisnis dipahami dengan baik. Ini melibatkan identifikasi tujuan analisis, pemahaman terhadap sumber data yang tersedia, dan klarifikasi asumsi yang mendasari analisis.
2. Pemahaman Data: Pada tahap ini, data yang relevan dikumpulkan dan dipahami dengan baik. Ini mencakup pemahaman terhadap struktur data, identifikasi variabel yang relevan, dan evaluasi kualitas data.
3. Persiapan Data: Langkah ini melibatkan pra-pemrosesan data untuk mempersiapkannya agar sesuai dengan proses analisis selanjutnya. Ini termasuk pembersihan data,

integrasi data dari berbagai sumber, transformasi data, dan pemilihan atribut yang relevan.

4. Pemodelan: Tahap ini melibatkan penggunaan algoritma data mining untuk mengekstrak pola atau informasi dari data yang telah dipersiapkan. Ini bisa melibatkan penggunaan berbagai teknik seperti clustering, klasifikasi, regresi, asosiasi, dan lainnya.
5. Evaluasi Model: Setelah model dibangun, langkah berikutnya adalah mengevaluasi kinerjanya. Ini dilakukan dengan menguji model pada data yang tidak terlihat sebelumnya atau dengan menggunakan metrik evaluasi yang sesuai.
6. Interpretasi dan Penggunaan: Tahap terakhir adalah menginterpretasikan hasil analisis dan menerapkannya dalam konteks bisnis. Hasil analisis dapat digunakan untuk mengambil keputusan yang lebih baik, mengidentifikasi peluang bisnis, memahami perilaku pelanggan, atau untuk tujuan lainnya sesuai dengan kebutuhan.

D. Tujuan dan Aplikasi

1. Tujuan Data Mining

Tujuan dari data mining adalah untuk mengeksplorasi dan menganalisis data besar dengan tujuan menemukan pola-pola yang bermanfaat, hubungan tersembunyi, atau pengetahuan yang berguna yang dapat digunakan untuk pengambilan keputusan yang lebih baik. Beberapa tujuan utama dari data mining meliputi (Ha *et al.*, 2011):

a. Prediksi: Memprediksi perilaku atau kejadian di masa depan berdasarkan pola atau tren yang ditemukan dalam data historis. Contohnya termasuk prediksi penjualan, prediksi risiko kredit, atau prediksi ketersediaan barang.
b. Segmentasi Pelanggan: Mengidentifikasi kelompok pelanggan yang memiliki karakteristik atau perilaku yang serupa. Ini membantu perusahaan dalam merancang strategi pemasaran yang lebih tepat sasaran dan personalisasi layanan.

c. Pengelompokan atau Clustering: Mengelompokkan data ke dalam kategori atau cluster berdasarkan kesamaan karakteristik. Ini membantu dalam pemahaman terhadap struktur data dan dapat digunakan untuk membuat segmentasi pasar atau untuk analisis kecenderungan.
d. Pengidentifikasian Pola Asosiasi: Mencari hubungan antara variabel dalam data. Contohnya termasuk menemukan pola pembelian bersamaan di toko ritel atau asosiasi antara gejala penyakit dalam data medis.
e. Deteksi Anomali: Mendeteksi anomali atau pola yang tidak biasa dalam data yang dapat menunjukkan adanya masalah atau peluang bisnis yang tidak terduga.
f. Optimisasi Proses: Meningkatkan efisiensi atau efektivitas suatu proses dengan menganalisis data historis untuk mengidentifikasi area-area di mana perbaikan dapat dilakukan.
g. Analisis Jejak: Melacak perilaku atau jejak pengguna dalam data transaksi, web, atau media sosial untuk memahami preferensi pengguna, tren, atau pola perilaku.

2. Aplikasi Data Mining

Aplikasi data mining adalah implementasi praktis dari teknik-teknik data mining dalam berbagai bidang dan industri untuk mencapai tujuan tertentu. Berikut adalah beberapa contoh aplikasi data mining yang umum:

a. Pemasaran: Data mining digunakan untuk segmentasi pelanggan, analisis perilaku pembelian, personalisasi kampanye pemasaran, dan prediksi tren pasar. Ini membantu perusahaan dalam meningkatkan retensi pelanggan, meningkatkan konversi penjualan, dan mengoptimalkan strategi pemasaran.
b. Keuangan: Di bidang keuangan, data mining digunakan untuk penilaian risiko kredit, deteksi kecurangan, manajemen portofolio investasi, dan prediksi pergerakan pasar. Hal ini membantu lembaga keuangan dalam mengelola risiko, meningkatkan keuntungan, dan membuat keputusan investasi yang lebih cerdas.

c. Kesehatan: Dalam industri kesehatan, data mining digunakan untuk prediksi penyakit, pengelompokan pasien, analisis data medis, dan penelitian klinis. Ini membantu dalam diagnosis yang lebih cepat, pengobatan yang lebih efektif, dan penemuan pola-pola baru dalam data kesehatan.
d. Ritel: Di industri ritel, data mining digunakan untuk manajemen rantai pasokan, penyesuaian harga, analisis stok, dan pengelolaan inventaris. Ini membantu dalam meningkatkan efisiensi operasional, mengurangi biaya persediaan, dan meningkatkan kepuasan pelanggan.
e. Telekomunikasi: Dalam industri telekomunikasi, data mining digunakan untuk analisis churn pelanggan, optimisasi jaringan, personalisasi layanan, dan prediksi permintaan. Ini membantu penyedia layanan telekomunikasi dalam meningkatkan retensi pelanggan, meningkatkan kualitas layanan, dan mengoptimalkan infrastruktur jaringan.
f. Transportasi: Di sektor transportasi, data mining digunakan untuk prediksi permintaan, manajemen lalu lintas, optimisasi rute, dan analisis kinerja armada. Ini membantu perusahaan transportasi dalam meningkatkan efisiensi operasional, mengurangi kemacetan lalu lintas, dan menyediakan layanan yang lebih dapat diandalkan bagi pelanggan.
g. Sosial Media: Dalam platform media sosial, data mining digunakan untuk analisis sentimen, deteksi tren, rekomendasi konten, dan identifikasi pengaruh. Ini membantu dalam memahami preferensi pengguna, meningkatkan interaksi dengan pengguna, dan meningkatkan keterlibatan dalam platform.

E. Teknik dan Analisis Data

Ada beberapa teknik analisis yang umum digunakan dalam data mining untuk mengekstrak pola atau informasi dari data. Berikut adalah beberapa teknik analisis data mining yang umum (Ha *et al*., 2011):

1. Klasifikasi: Teknik ini digunakan untuk mengklasifikasikan data ke dalam kategori atau kelas berdasarkan atribut-atribut tertentu. Contoh aplikasinya adalah klasifikasi email sebagai spam atau bukan spam, klasifikasi pelanggan sebagai berpotensi churn atau tidak, dan klasifikasi gambar sebagai kategori tertentu.
2. Clustering: Clustering adalah teknik yang digunakan untuk mengelompokkan data ke dalam kelompok-kelompok yang memiliki karakteristik atau atribut yang serupa. Ini membantu dalam mengidentifikasi struktur tersembunyi dalam data dan dapat digunakan untuk segmentasi pasar, analisis pola, atau penemuan pengetahuan baru.
3. Regresi: Teknik regresi digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen dan variabel dependen. Ini membantu dalam memprediksi nilai variabel dependen berdasarkan variabel independen yang diberikan. Contoh aplikasinya adalah prediksi harga rumah berdasarkan fitur-fitur tertentu atau prediksi penjualan berdasarkan faktor-faktor pasar.
4. Asosiasi: Teknik asosiasi digunakan untuk menemukan hubungan atau asosiasi antara item-item dalam kumpulan data. Contoh aplikasinya adalah analisis keranjang belanja di toko ritel, di mana pola pembelian bersamaan dapat diidentifikasi untuk memberikan rekomendasi produk kepada pelanggan.
5. Anomali Detection: Teknik ini digunakan untuk mendeteksi anomali atau pola yang tidak biasa dalam data. Ini membantu dalam mengidentifikasi aktivitas mencurigakan, penipuan, atau masalah operasional yang tidak terduga. Contoh aplikasinya adalah deteksi kecurangan kartu kredit, deteksi serangan cyber, atau pemantauan kinerja sistem.

6. Analisis Deret Waktu: Teknik ini digunakan untuk menganalisis data yang berkaitan dengan urutan waktu atau data deret waktu. Ini membantu dalam memprediksi tren, melakukan analisis perilaku temporal, atau mendeteksi pola dalam data berdasarkan waktu. Contoh aplikasinya adalah prediksi harga saham, prediksi cuaca, atau analisis data sensor.
7. Text Mining: Teknik text mining atau analisis teks digunakan untuk mengekstrak informasi atau pola dari data teks. Ini mencakup pengklasifikasian dokumen, analisis sentimen, ekstraksi informasi, dan pengelompokan teks berdasarkan topik atau tema tertentu. Contoh aplikasinya adalah analisis sentimen di media sosial, pengelompokan artikel berita, atau klasifikasi dokumen.

F. Algoritma Data Mining

Algoritma data mining adalah teknik komputasional yang digunakan untuk mengekstrak pola atau informasi yang bermanfaat dari data. Berikut adalah beberapa algoritma data mining yang umum digunakan (Ye, 2013):

1. Decision Trees (Pohon Keputusan): Algoritma ini menggunakan struktur pohon untuk memodelkan keputusan atau klasifikasi. Ini memecah data menjadi bagian-bagian yang lebih kecil berdasarkan atribut-atributnya dan menghasilkan pohon keputusan yang dapat digunakan untuk membuat prediksi.
2. Random Forest: Random Forest adalah ensamble dari banyak pohon keputusan yang digunakan untuk klasifikasi atau regresi. Algoritma ini menggabungkan prediksi dari beberapa pohon keputusan untuk meningkatkan akurasi dan mengurangi overfitting.
3. k-Nearest Neighbors (k-NN): Algoritma ini melakukan klasifikasi atau regresi dengan membandingkan data yang baru dengan data tetangga terdekatnya dalam ruang fitur. Prediksi dilakukan berdasarkan mayoritas label atau nilai dari tetangga terdekat.

4. Naive Bayes: Algoritma ini berdasarkan teorema Bayes dan mengasumsikan independensi antara fitur-fitur dalam data. Ini digunakan untuk klasifikasi dan memprediksi probabilitas kelas atau label berdasarkan nilai-nilai fitur.
5. Support Vector Machines (SVM): SVM mencari hyperplane terbaik yang memisahkan kelas-kelas data dalam ruang fitur. Ini digunakan untuk klasifikasi dan regresi, terutama dalam kasus data yang tidak linier terpisah.
6. K-Means Clustering: Algoritma ini digunakan untuk clustering atau pengelompokan data ke dalam kategori-kategori yang serupa. Ini membagi data menjadi k kluster berdasarkan kesamaan antara data.
7. Apriori Algorithm: Algoritma ini digunakan untuk menemukan asosiasi antara item dalam kumpulan data transaksi. Ini digunakan dalam analisis keranjang belanja untuk mengidentifikasi pola pembelian bersamaan.
8. Principal Component Analysis (PCA): Algoritma ini digunakan untuk reduksi dimensi dalam data dengan memproyeksikan data ke ruang dimensi yang lebih rendah yang mempertahankan sebagian besar variasi dalam data.
9. Linear Regression: Algoritma ini digunakan untuk memodelkan hubungan linier antara variabel independen dan variabel dependen. Ini digunakan untuk regresi dan memprediksi nilai variabel dependen berdasarkan variabel independen.

G. Tantangan dan Masalah

Ada beberapa tantangan dan masalah yang umum dihadapi dalam praktik data mining. Berikut beberapa di antaranya (Ha *et al*., 2011):

1. Kekurangan Data atau Data yang Tidak Representatif: Kualitas data yang buruk, data yang tidak lengkap, atau data yang tidak representatif dapat menghasilkan hasil yang tidak akurat atau tidak dapat diandalkan dalam proses data mining.

2. Overfitting: Overfitting terjadi ketika model data mining terlalu kompleks dan "memorize" data pelatihan, sehingga tidak dapat melakukan generalisasi dengan baik pada data baru. Ini dapat menghasilkan kinerja yang buruk pada data yang tidak terlihat sebelumnya.
3. Dimensi Tinggi (High-Dimensionality): Ketika jumlah fitur atau atribut dalam data sangat besar, analisis data menjadi lebih kompleks dan sulit untuk menemukan pola yang bermanfaat. Masalah ini sering disebut sebagai "kutukan dimensi tinggi".
4. Perilaku Dinamis: Data yang memiliki perilaku dinamis atau berubah seiring waktu memerlukan teknik analisis yang dapat menangani aspek temporal dan perubahan tersebut dengan baik.
5. Skala Besar (Big Data): Memproses data yang sangat besar, seperti big data, memerlukan infrastruktur komputasi yang kuat dan algoritma yang efisien untuk mengatasi masalah kinerja dan skala.
6. Interpretabilitas: Beberapa model data mining, seperti neural networks atau ensemble methods, mungkin sulit untuk diinterpretasikan oleh manusia. Ini dapat menjadi tantangan dalam menjelaskan dan memahami pola atau keputusan yang dihasilkan oleh model tersebut.
7. Pemilihan Atribut yang Tepat: Menemukan atribut atau fitur yang paling relevan dan informatif dalam data adalah tantangan dalam proses pra-pemrosesan data dan perancangan model.
8. Ketidakseimbangan Kelas: Ketika kelas-kelas dalam data tidak seimbang, di mana satu kelas memiliki jumlah sampel yang jauh lebih banyak daripada yang lain, model dapat menjadi bias terhadap kelas mayoritas.
9. Keamanan dan Privasi: Menggunakan data yang sensitif atau pribadi dapat menimbulkan masalah keamanan dan privasi, terutama jika data tidak dijaga dengan baik atau jika informasi sensitif dapat diidentifikasi melalui analisis.

H. Etika dan Privasi

Etika dan privasi adalah dua aspek penting yang perlu dipertimbangkan dalam praktik data mining. Berikut adalah beberapa pertimbangan terkait etika dan privasi dalam data mining (Davis, K, 2012):

1. Privasi Data: Penggunaan data yang sensitif atau pribadi dalam data mining memerlukan perlindungan privasi yang kuat. Penting untuk memastikan bahwa data yang digunakan telah disensor atau anonimisasi dengan benar agar tidak mengidentifikasi individu secara langsung.
2. Ketelitian Data: Penting untuk memastikan bahwa data yang digunakan dalam data mining akurat dan terpercaya. Kekurangan data atau data yang tidak akurat dapat menghasilkan kesimpulan yang salah atau tidak dapat diandalkan.
3. Transparansi: Penting untuk menjaga transparansi dalam penggunaan data mining. Organisasi harus menjelaskan dengan jelas bagaimana data digunakan, tujuan analisis, dan dampaknya terhadap individu atau masyarakat.
4. Keterbukaan dan Akses: Praktik data mining harus terbuka untuk pemeriksaan dan pengawasan eksternal. Ini dapat membantu memastikan kepatuhan terhadap regulasi privasi dan etika yang berlaku serta membangun kepercayaan dengan pemangku kepentingan.
5. Keterbatasan Penggunaan Data: Data mining tidak boleh digunakan untuk tujuan yang merugikan atau diskriminatif. Penting untuk memastikan bahwa data tidak digunakan untuk diskriminasi rasial, gender, agama, atau atribut lainnya yang dilindungi.
6. Konsent: Pemakaian data: Penting untuk memperoleh persetujuan yang jelas dan sesuai dari individu sebelum menggunakan data pribadi mereka dalam analisis data mining. Ini mencakup pengumpulan, pengolahan, dan penggunaan data.

7. Kesetaraan dan Keadilan: Penting untuk memastikan bahwa praktik data mining tidak menghasilkan atau memperkuat ketimpangan sosial atau ekonomi. Analisis data harus dilakukan dengan memperhatikan kesetaraan dan keadilan.
8. Penghapusan Data: Data yang tidak lagi diperlukan harus dihapus secara aman dan sesuai dengan regulasi privasi yang berlaku. Ini membantu melindungi privasi individu dan mengurangi risiko kebocoran atau penyalahgunaan data.
9. Pelaporan Dampak: Organisasi harus secara teratur melaporkan dampak penggunaan data mining, termasuk keputusan atau rekomendasi yang dihasilkan, serta langkah-langkah yang diambil untuk menjaga privasi dan etika.

I. Perangkat Lunak dan Alat

Berikut ini beberapa perangkat lunak dan alat yang umum digunakan dalam praktik data mining (Ha *et al.*, 2011):

1. Weka: Weka adalah perangkat lunak open-source yang menyediakan berbagai algoritma data mining dan analisis pola. Ini memiliki antarmuka grafis yang ramah pengguna dan mendukung berbagai tugas data mining, termasuk klasifikasi, regresi, clustering, dan visualisasi data.
2. RapidMiner: RapidMiner adalah platform analisis data open-source yang menyediakan alat untuk data mining, analisis prediktif, dan pemodelan bisnis. Ini menawarkan antarmuka drag-and-drop yang mudah digunakan dan mendukung berbagai algoritma dan teknik analisis data.
3. Python dengan library scikit-learn: Python adalah bahasa pemrograman yang populer dalam data science, dan scikit-learn adalah library yang menyediakan berbagai algoritma dan alat untuk data mining dan analisis prediktif.
4. Microsoft SQL Server Analysis Services (SSAS): SSAS adalah komponen dari Microsoft SQL Server yang menyediakan layanan analisis multidimensional dan data mining.
5. Oracle Data Mining: Oracle Data Mining adalah fitur dari Oracle Database yang menyediakan algoritma dan teknik data mining yang terintegrasi langsung ke dalam database.

Ini memungkinkan pengguna untuk melakukan data mining langsung pada data dalam database tanpa perlu mentransfer data ke perangkat lunak eksternal.

6. SAS Enterprise Miner: SAS Enterprise Miner adalah perangkat lunak data mining dan analisis prediktif yang menyediakan berbagai algoritma dan alat untuk membangun model prediktif. Ini sering digunakan oleh organisasi besar untuk melakukan analisis data yang kompleks dan membangun solusi analisis prediktif.

Cara kerja perangkat lunak dan alat dalam data mining bervariasi tergantung pada jenis perangkat lunak atau alat yang digunakan, tetapi umumnya melibatkan serangkaian langkah seperti yang berikut:

1. Persiapan Data: Ini termasuk mengumpulkan data dari sumber yang berbeda, membersihkan data dari nilai yang hilang atau tidak valid, mengubah format data, dan menggabungkan data dari berbagai sumber jika diperlukan.
2. Ekplorasi Data: Ini melibatkan pemahaman yang lebih baik tentang data, termasuk distribusi nilai, korelasi antara variabel, dan pola-pola yang mungkin ada dalam data
3. Pemilihan Model dan Algoritma: Ini tergantung pada jenis data yang Anda miliki (misalnya, apakah Anda memiliki data kategoris atau numerik), tujuan analisis (misalnya, klasifikasi atau regresi), dan karakteristik data lainnya.
4. Pemodelan: Ini melibatkan penerapan algoritma yang dipilih ke data yang ada untuk mempelajari pola dan tren dalam data.
5. Validasi Model: Memvalidasi model untuk memastikan bahwa itu memberikan hasil yang akurat dan dapat diandalkan. Ini melibatkan penggunaan data yang tidak terlihat sebelumnya untuk menguji kinerja model, dan teknik validasi seperti validasi silang atau holdout validation.
6. Evaluasi dan Interpretasi: Ini melibatkan penggunaan metrik evaluasi yang tepat (misalnya, akurasi untuk klasifikasi atau RMSE untuk regresi) untuk mengukur kinerja model, serta

interpretasi hasil analisis untuk memahami implikasi bisnisnya.

DAFTAR PUSTAKA

Davis, K. (2012). Ethics of Big Data: Balancing risk and innovation. " O'Reilly Media, Inc.".

Ha, J., Kambe, M., & Pe, J. (2011). Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*. https://doi.org/10.1016/C2009-0-61819-5

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques. In *Data Mining: Practical Machine Learning Tools and Techniques*.

Ye, N. (2013). Data Mining: Theories, Algorithms, and Examples. In *Data Mining: Theories, Algorithms, and Examples*.

BAB 2 | ALGORITMA DAN METODE DATA MINING

Felix Reba, S.Si., M.Sc.

A. Clustering

Clustering adalah sebuah teknik dalam data mining yang digunakan untuk mengelompokkan data ke dalam kelompok-kelompok (clusters) berdasarkan kesamaan karakteristik. Tujuan utama dari clustering adalah untuk menemukan struktur tersembunyi dalam data, di mana data dalam satu kelompok memiliki kesamaan yang tinggi antara satu sama lain, sedangkan data antar kelompok memiliki perbedaan yang signifikan. Beberapa metode umum dalam clustering adalah (Granell *et al.*, 2011; Ye, 2013):

1. K-Means
2. Agglomerative Hierarchical Clustering
3. DBSCAN
4. Mean Shift Clustering
5. Fuzzy C-Means (FCM)
6. Gaussian Mixture Models (GMM)
7. OPTICS
8. Self-Organizing Maps (SOM)
9. Affinity Propagation

Ada banyak lagi metode clustering lainnya, dan pilihan metode tergantung pada sifat data dan tujuan analisis. Setiap metode memiliki kelebihan dan kelemahan yang berbeda, dan pemilihan metode yang tepat sangat penting untuk mendapatkan hasil clustering yang baik.

B. Classification

Classification adalah sebuah teknik dalam data mining yang digunakan untuk memprediksi kategori atau label dari suatu data berdasarkan fitur-fitur yang diberikan. Tujuannya adalah untuk membangun model yang dapat mengklasifikasikan data baru ke dalam kelas yang telah ditentukan sebelumnya. Metode dalam data mining yang digunakan untuk melakukan klasifikasi termasuk (Granell *et al.*, 2011; Ye, 2013):

1. **Decision Trees**: Membagi data berdasarkan serangkaian keputusan berbasis fitur untuk memprediksi kelas target.
2. **Random Forest**: Kumpulan dari beberapa pohon keputusan yang digunakan untuk meningkatkan akurasi klasifikasi.
3. **Support Vector Machines (SVM)**: Membangun hyperplane atau beberapa hyperplane di ruang fitur yang memisahkan kelas-kelas dengan margin maksimum.
4. **Logistic Regression**: Model regresi yang digunakan untuk memprediksi probabilitas bahwa suatu pengamatan akan menjadi bagian dari satu kelas tertentu.
5. **k-Nearest Neighbors (k-NN)**: Mengklasifikasikan pengamatan berdasarkan mayoritas kelas dari k tetangga terdekatnya dalam ruang fitur.
6. **Naive Bayes**: Menggunakan teorema Bayes untuk memperkirakan probabilitas kelas target berdasarkan fitur-fitur yang diberikan.
7. **Neural Networks**: Jaringan saraf tiruan yang terdiri dari lapisan-lapisan neuron yang digunakan untuk memodelkan hubungan kompleks antara fitur dan kelas target.
8. **AdaBoost**: Metode boosting yang menggabungkan beberapa model lemah menjadi satu model yang kuat.
9. **Ensemble Learning**: Kombinasi dari beberapa model klasifikasi untuk meningkatkan akurasi dan ketahanan terhadap overfitting.

Setiap metode memiliki kelebihan dan kelemahan, serta berlaku untuk berbagai jenis data dan skenario. Pemilihan metode yang tepat tergantung pada sifat data, kompleksitas masalah klasifikasi, dan tujuan analisis.

C. Association Rule Mining (Pencarian Aturan Asosiasi)

Association Rule Mining, atau Pencarian Aturan Asosiasi, adalah sebuah teknik dalam data mining yang digunakan untuk menemukan pola hubungan antara item-item dalam dataset transaksional atau dataset yang berisi himpunan item. Tujuan utama dari Association Rule Mining adalah untuk menemukan asosiasi atau korelasi yang kuat antara item-item yang sering muncul bersama-sama dalam transaksi atau dataset. Metode dalam Association Rule Mining meliputi (Dunhan, 2013; Ye, 2013):

1. **Apriori Algorithm**: Algoritma yang paling populer untuk pencarian aturan asosiasi. Algoritma ini bekerja dengan menghasilkan kandidat-kandidat aturan asosiasi berdasarkan prinsip apriori, di mana setiap subset dari aturan harus sering terjadi dalam dataset.
2. **FP-Growth Algorithm**: Algoritma yang efisien untuk pencarian aturan asosiasi. FP-Growth bekerja dengan membangun struktur pohon yang disebut FP-Tree, yang kemudian digunakan untuk mengekstraksi aturan asosiasi secara efisien.
3. **Eclat Algorithm**: Algoritma alternatif untuk pencarian aturan asosiasi. Eclat menggunakan struktur tumpukan vertikal (vertical format) untuk menghitung frekuensi itemset secara efisien.
4. **Maximal Frequent Itemsets**: Selain aturan asosiasi, metode ini juga digunakan untuk menemukan himpunan item yang sering muncul bersama-sama, tetapi tidak semua itemset yang muncul secara bersamaan termasuk dalam aturan.

5. **Closed Frequent Itemsets**: Seperti maximal frequent itemsets, namun himpunan item yang dihasilkan adalah himpunan tertutup yang tidak ada subsetnya dengan dukungan yang sama.
6. **Constraint-Based Association Mining**: Pendekatan yang memungkinkan pengguna untuk menyertakan aturan atau batasan tertentu yang harus dipenuhi oleh aturan asosiasi yang dihasilkan.
7. **Sequential Pattern Mining**: Metode yang digunakan untuk menemukan pola urutan dari item-item yang sering muncul bersama-sama dalam transaksi atau urutan waktu tertentu.

Association Rule Mining memiliki banyak aplikasi dalam berbagai bidang, termasuk analisis keranjang belanja (market basket analysis), analisis log web, personalisasi rekomendasi produk, dan analisis interaksi item dalam sistem e-commerce.

D. Regresi

Regresi adalah sebuah teknik dalam data mining yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (fitur) dan variabel dependen (target) dalam dataset. Tujuan utama dari regresi adalah untuk memprediksi nilai variabel dependen berdasarkan nilai variabel independen. Beberapa metode dalam regresi yang umum digunakan meliputi (Ye, 2013):

1. **Linear Regression**: Model regresi yang paling sederhana, di mana hubungan antara variabel independen dan dependen diasumsikan linear. Regresi Linier Berganda:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

Dengan:
Y = Variabel Dependen; X = Variabel Independen; β_0 = Konstanta; $\beta_1, \ldots, \beta_n$ = Koefisien Determinasi; ε = Error Term

2. **Logistic Regression**: Digunakan ketika variabel dependen adalah biner (dua kategori). Model ini memprediksi probabilitas kejadian suatu kategori berdasarkan variabel independen. Saat data yang lebih relevan ditambahkan,

algoritma meningkatkan kemampuannya untuk memprediksi klasifikasi dalam kumpulan data:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_k X_k)}}$$

3. **Polynomial Regression**: Memodelkan hubungan non-linear antara variabel independen dan dependen dengan menggunakan fungsi polinomial.

4. **Ridge Regression**: Versi regularized dari regresi linear yang mengurangi overfitting dengan menambahkan komponen regularisasi ke fungsi biaya. Jadi dalam Regresi Ridge fungsi tujuan kita adalah:

$$Min\left(\sum \varepsilon^2 + \lambda \sum \beta^2\right)$$

$$= Min \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k))^2 + \lambda \sum \beta^2$$

Berikut ini λ adalah parameter regularisasi yang merupakan bilangan non negatif.

5. **Lasso Regression**: Regresi Lasso (Least Absolute Shrinkage and Selection Operator) adalah metode regresi yang digunakan untuk melakukan pemilihan variabel dan mengurangi dimensi dalam model regresi. Tujuannya adalah untuk meminimalkan jumlah koefisien yang signifikan secara statistik, dengan memaksa beberapa koefisien menjadi nol. Ini membantu dalam menghasilkan model yang lebih sederhana dan mudah diinterpretasikan, serta mengatasi masalah multicollinearity di antara prediktor. Berikut ini adalah Model Regresi Lasso:

$$Min\left(\sum \varepsilon^2 + \lambda \sum |\beta|\right)$$

$$= Min \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k))^2 + \lambda \sum |\beta|$$

6. **ElasticNet Regression**: Regresi ElasticNet adalah teknik regresi yang menggabungkan dua metode regresi yang populer, yaitu regresi Lasso (Least Absolute Shrinkage and

Selection Operator) dan regresi Ridge. Tujuan dari regresi ElasticNet adalah untuk mengatasi beberapa kelemahan dari kedua metode tersebut, serta memperoleh keuntungan dari keduanya. Berikut adalah Model Regresi Jaringan Elastis:

$$Min\left(\sum \varepsilon^2 + \lambda_1 \sum \beta^2 + \lambda_2 \sum |\beta|\right)$$

$$= Min \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k))^2 \lambda_1 \sum \beta^2 + \lambda_2 \sum |\beta|$$

7. **Support Vector Regression (SVR)**: Menggunakan pendekatan dari Support Vector Machines (SVM) untuk regresi, dengan tujuan untuk menemukan hyperplane yang memiliki kesalahan prediksi minimum.
8. **Decision Tree Regression**: Memodelkan hubungan antara variabel independen dan dependen dengan membagi ruang fitur ke dalam segmen-segmen berdasarkan serangkaian keputusan.
9. **Random Forest Regression**: Kumpulan dari beberapa pohon keputusan yang digunakan untuk meningkatkan akurasi dan ketahanan terhadap overfitting dalam regresi.

E. Anomaly Detection

Anomaly Detection, atau deteksi anomali, adalah sebuah teknik dalam data mining yang digunakan untuk mengidentifikasi pola atau instansi yang tidak biasa atau langka dalam dataset. Tujuan utama dari Anomaly Detection adalah untuk menemukan observasi yang berbeda secara signifikan dari pola umum dalam data. Anomali seringkali menunjukkan kejadian yang tidak biasa, kegagalan sistem, atau perilaku yang mencurigakan. Beberapa metode dalam Anomaly Detection termasuk (Ville, 2001):

1. **Statistical Methods**: Pendekatan yang menggunakan statistik deskriptif untuk menentukan apakah sebuah observasi adalah anomali berdasarkan ukuran seperti jarak, deviasi standar, atau distribusi.

2. **Density Based Methods**: Menggunakan metrik kerapatan data seperti jarak ke k-tetangga terdekat atau estimasi kerapatan kernel untuk mengidentifikasi area yang jarang atau langka dalam ruang fitur.
3. **Clustering Methods**: Memisahkan data ke dalam kelompok-kelompok dan mengidentifikasi observasi yang berada di luar kelompok utama sebagai anomali.
4. **Classification Methods**: Membangun model klasifikasi untuk membedakan antara data normal dan anomali. Contohnya adalah One-Class SVM dan algoritma klasifikasi lainnya yang dilatih dengan hanya menggunakan data normal.
5. **Proximity-Based Methods**: Mengukur jarak atau kemiripan antara setiap observasi dalam dataset, dan mengidentifikasi observasi yang memiliki jarak atau kemiripan yang rendah sebagai anomali.
6. **Deep Learning Approaches**: Menggunakan arsitektur jaringan saraf yang kompleks untuk menangkap pola yang rumit dalam data dan mengidentifikasi anomali berdasarkan perbedaan yang signifikan dari pola umum.
7. **Ensemble Methods**: Menggabungkan beberapa model deteksi anomali untuk meningkatkan keakuratan dan ketahanan terhadap noise.
8. **Time Series Analysis**: Melakukan analisis pada data deret waktu untuk mendeteksi perubahan atau kejadian yang tidak biasa dalam pola waktu.
9. **Isolation Forest**: Algoritma yang mengisolasi anomali dengan membagi ruang fitur secara acak dan mengukur jumlah partisi yang diperlukan untuk mengisolasi observasi.

F. Dimensionality Reduction

Dimensionality reduction adalah proses mengurangi jumlah fitur (variabel) dalam kumpulan data dengan mempertahankan sebanyak mungkin informasi yang relevan. Tujuan utamanya adalah untuk mengatasi masalah "kutukan dimensi" (curse of dimensionality) di mana semakin banyak fitur

yang ada dalam data, semakin sulit untuk menganalisisnya, memahaminya, dan menghasilkan model yang baik. Dimensi yang tinggi sering kali mengakibatkan kompleksitas komputasi yang tinggi, peningkatan memori yang besar, dan risiko overfitting (Ville, 2001).

1. **Principal Component Analysis (PCA):** Sebuah teknik dalam data mining yang digunakan untuk mereduksi dimensi dari dataset yang kompleks dengan memproyeksikan data ke dalam ruang dimensi yang lebih rendah, sambil mempertahankan sebanyak mungkin informasi yang relevan. PCA berusaha untuk menemukan arah (komponen) yang paling bervariasi dalam data dan mentransformasikan data ke dalam ruang yang terdefinisi oleh komponen-komponen tersebut.

2. **t-Distributed Stochastic Neighbor Embedding (t-SNE):** Sebuah teknik dalam data mining yang digunakan untuk mereduksi dimensi dari dataset yang kompleks dengan memetakan data ke dalam ruang dimensi yang lebih rendah. t-SNE terutama efektif dalam memvisualisasikan data yang kompleks dalam ruang dimensi yang lebih rendah, sehingga memungkinkan untuk menangkap struktur dan pola yang tersembunyi dalam data. Dimana, P_{ij} adalah probabilitas bersyarat antara pasangan pengamatan i dan j dalam ruang dimensi asli, dan Q_{ij} adalah probabilitas bersyarat yang dipetakan ke dalam ruang dimensi yang direduksi, maka fungsi biaya untuk t-SNE dapat dinyatakan sebagai berikut:

$$C = \sum_i KL(P_i|||Q_i) = \sum_i \sum_j P_{ij} log\left(\frac{P_{ij}}{Q_{ij}}\right)$$

Di sini, $KL(P_i||Q_i)$ adalah divergensi Kullback-Leibler antara distribusi probabilitas P_i dan Q_i untuk pengamatan ke-i. Tujuan dari t-SNE adalah untuk meminimalkan nilai dari fungsi biaya ini.

3. **Linear Discriminant Analysis (LDA)**: Sebuah teknik dalam data mining yang digunakan untuk mereduksi dimensi dari dataset dan pada saat yang sama memaksimalkan pemisahan antara kelas-kelas yang berbeda dalam dataset. LDA mencoba untuk menemukan kombinasi linear dari fitur-fitur yang memaksimalkan pemisahan antara kelas-kelas, sehingga membuatnya menjadi metode yang umum digunakan dalam klasifikasi.

G. Ensemble Learning

Ensemble learning adalah teknik dalam pembelajaran mesin di mana beberapa model (biasanya disebut sebagai "anggota ensemble") digabungkan bersama untuk meningkatkan kinerja prediksi secara keseluruhan. Konsep di balik ensemble learning adalah bahwa gabungan dari beberapa model yang secara individu mungkin tidak sempurna dapat menghasilkan prediksi yang lebih akurat dan stabil (Zhou, 2012).

1. **Bagging (Bootstrap Aggregating)**: Sebuah teknik dalam data mining yang digunakan untuk meningkatkan kinerja model dengan cara menggabungkan hasil dari beberapa model yang dilatih secara independen. Random Forest merupakan salah satu contoh populer dari algoritma yang menggunakan teknik bagging.

   Dimana, $f_i(x)$ adalah fungsi prediksi dari model ke-i untuk pengamatan x, dan terdapat M model dalam ensemble, maka prediksi dari ensemble bagging bisa dinyatakan sebagai:

$$\hat{f}_{bagging}(x) = \frac{1}{M}\sum_{i=1}^{M} f_i(x)$$

   $\hat{f}_{bagging}(x)$: prediksi akhir dari ensemble bagging untuk pengamatan x

2. **Boosting**: Sebuah teknik dalam data mining yang digunakan untuk meningkatkan kinerja model dengan cara menggabungkan hasil dari beberapa model yang lemah (weak learners) menjadi model yang kuat (strong learner).

Ada beberapa algoritma boosting yang populer, seperti AdaBoost (Adaptive Boosting) dan Gradient Boosting Machines (GBM).

Rumus umum untuk prediksi dari ensemble boosting dapat dinyatakan sebagai berikut:

$$\hat{f}_{boosting}(x) = \sum_{i=1}^{T} \alpha_t f_t(x)$$

$\hat{f}_{boosting}(x)$: prediksi akhir dari ensemble boosting untuk pengamatan x, T adalah jumlah iterasi, α_t adalah bobot yang diberikan pada model ke-t, dan $f_t(x)$ adalah prediksi dari model ke-t.

3. **Stacking**: Sebuah teknik ensemble learning yang digunakan untuk meningkatkan kinerja model dengan cara menggabungkan hasil dari beberapa model yang dilatih secara independen. Dalam stacking, prediksi dari model-model dasar (base models) digunakan sebagai fitur input untuk model meta (meta model) yang kemudian digunakan untuk membuat prediksi akhir.

   Dimana, x adalah prediksi dari model dasar ke-i pengamatan dan g(x) adalah prediksi dari model meta, maka prediksi dari stacking dapat dinyatakan sebagai:

$$\hat{y}_{stacking}(x) = g(\{f_1(x), f_2(x), \ldots, f_n(x)\})$$

$\hat{y}_{stacking}(x)$: prediksi akhir dari stacking untuk pengamatan x.

H. Deep Learning

Deep Learning adalah sebuah cabang dari machine learning yang menggunakan arsitektur jaringan saraf tiruan (neural networks) yang dalam untuk memodelkan dan mempelajari representasi hierarkis dari data. Deep Learning bertujuan untuk mengekstraksi fitur yang paling relevan atau representasi yang paling abstrak dari data input untuk melakukan tugas-tugas seperti klasifikasi, regresi, pengenalan pola, atau generasi data baru.

Beberapa metode dalam Deep Learning meliputi (Yang *et al*., 2022; Yi, 2022):

1. **Convolutional Neural Networks (CNN)**: Digunakan khusus untuk data berstruktur grid, seperti gambar. CNN memiliki lapisan konvolusi yang memungkinkan model untuk menangkap pola spasial dalam data.
2. **Recurrent Neural Networks (RNN)**: Dirancang untuk bekerja dengan data berurutan, seperti teks atau time series. RNN memiliki loop yang memungkinkan informasi untuk diproses secara berurutan.
3. **Long Short-Term Memory (LSTM)**: Jenis khusus dari RNN yang dirancang untuk mengatasi masalah vanishing gradient, yang memungkinkan model untuk mempelajari ketergantungan jangka panjang dalam data berurutan.
4. **Generative Adversarial Networks (GAN)**: Terdiri dari dua jaringan saraf yang bersaing, yaitu generator dan diskriminator, yang bekerja bersama untuk menghasilkan data baru yang terlihat seperti data yang ada.
5. **Autoencoders**: Jaringan saraf yang digunakan untuk mempelajari representasi tersembunyi dari data input dengan mengurangi dimensinya ke dalam ruang fitur yang lebih kecil.
6. **Transformer Models**: Arsitektur jaringan saraf yang berfokus pada perhatian dan dapat menangani input sekuensial dengan baik, seperti teks atau audio. Transformer terkenal karena kemampuannya dalam pemrosesan bahasa alami dan mampu menghasilkan hasil yang luar biasa dalam tugas-tugas seperti terjemahan mesin dan generasi teks.
7. **Deep Reinforcement Learning**: Menggabungkan Deep Learning dengan reinforcement learning, yang melibatkan agen yang belajar tindakan yang optimal dalam lingkungan yang dinamis.
8. **Deep Belief Networks (DBN)**: Arsitektur yang terdiri dari beberapa lapisan yang berbeda jenis, termasuk lapisan yang bertindak sebagai pembelajaran unsupervised dan lapisan yang bertindak sebagai pembelajaran supervised.

I. Text Mining

Text Mining adalah sebuah teknik dalam data mining yang digunakan untuk mengekstraksi informasi yang bermakna dan terstruktur dari teks yang tidak terstruktur, seperti dokumen, artikel, email, dan media sosial. Tujuan utama dari Text Mining adalah untuk mengidentifikasi pola, tren, hubungan, dan wawasan baru dalam teks yang dapat digunakan untuk pengambilan keputusan. Beberapa metode dalam Text Mining termasuk (Jiawei Han; Kamber Micheline, 2011) dan (Bidin A, 2017):

1. **Tokenization;** Proses memecah teks menjadi unit-unit yang lebih kecil seperti kata-kata, frasa, atau token.
2. **Stopwords Removal:** Menghapus kata-kata yang umum dan tidak memiliki makna (stopwords) dari teks untuk meningkatkan kualitas analisis.
3. **Stemming and Lemmatization:** Mengubah kata-kata ke bentuk dasar (stemming) atau kata dasar (lemmatization) untuk mengurangi variasi kata yang memiliki arti yang sama.
4. **Bag of Words (BoW):** Representasi teks sebagai himpunan kata-kata unik dalam dokumen tanpa memperhatikan urutan atau struktur kalimat.
5. **Term Frequency-Inverse Document Frequency (TF-IDF):** Menghitung bobot kata-kata dalam dokumen berdasarkan seberapa sering kata-kata tersebut muncul dalam dokumen (term frequency) dan seberapa jarang kata-kata tersebut muncul dalam seluruh korpus dokumen (inverse document frequency).
6. **Word Embeddings:** Representasi vektor yang dipelajari dari kata-kata dalam teks yang menangkap makna dan konteks kata-kata.
7. **Topic Modeling:** Mengidentifikasi topik-topik utama yang ada dalam korpus teks menggunakan algoritma seperti Latent Dirichl*et al*location (LDA).
8. **Sentiment Analysis:** Mengidentifikasi dan mengekstraksi sentimen atau opini dari teks, seperti positif, negatif, atau netral.

9. **Text Classification:** Mengklasifikasikan dokumen atau teks ke dalam kategori-kategori yang telah ditentukan sebelumnya berdasarkan konten atau topiknya, menggunakan metode seperti Naive Bayes, Support Vector Machines (SVM), atau Deep Learning.

J. Time Series Analysis

Time series analysis adalah sebuah metode statistik yang digunakan untuk menganalisis data yang dihasilkan dalam interval waktu yang teratur. Data dalam time series biasanya diambil pada titik-titik waktu yang berurutan, seperti harian, bulanan, atau tahunan, dan dianalisis untuk mengidentifikasi pola, tren, dan perilaku dalam data tersebut (Jiawei Han; Kamber Micheline, 2011).

1. *Autoregressive Integrated Moving Average* (ARIMA) adalah model statistik yang digunakan untuk menganalisis dan meramalkan data deret waktu. Model ini merupakan kombinasi dari tiga komponen utama: autoregressive (AR), integrated (I), dan moving average (MA). Rumus umum ARIMA(p, d, q) untuk nilai-nilai deret waktu y_t adalah sebagai berikut:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q} + e_t$$

dengan:
c : konstanta, $\phi_1, \phi_2, \ldots, \phi_p)$: parameter autoregressive, $(\theta_1, \theta_2, \ldots, \theta_q)$: parameter moving average, e_t : nilai gangguan pada waktu t.

2. Metode Exponential Smoothing adalah teknik yang digunakan dalam analisis deret waktu untuk meramalkan data berdasarkan pola eksponensial. Rumus dasar metode Exponential Smoothing untuk meramalkan nilai $\hat{y}_{t+1}$ berdasarkan nilai observasi sebelumnya $\hat{y}_t$ adalah sebagai berikut:

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$$

Di mana,
α : parameter smoothing yang berada di antara 0 dan 1
y_t : nilai observasi pada waktu t
$\hat{y}_t$: ramalan sebelumnya (biasanya ramalan pada waktu sebelumnya).

3. *Seasonal Decomposition of Time Series* (STL) adalah metode statistik yang digunakan untuk memisahkan sebuah deret waktu menjadi tiga komponen utama: komponen musiman, komponen tren, dan komponen sisa (residual). Secara umum, STL memisahkan deret waktu y_t menjadi tiga komponen sebagai berikut:

$$y_t = S_t + T_t + R_t$$

S_t : komponen musiman, T_t : komponen tren dan R_t : komponen sisa atau residu.

DAFTAR PUSTAKA

Bidin A. (2017). Опыт аудита обеспечения качества и безопасности медицинской деятельности в медицинской организации по разделу «Эпидемиологическая безопасностьNo Title. In *Вестник Росздравнадзора* (Vol. 4, Issue 1).

Dunhan, M. H. (2013). Data Mining: Introductory and Advanced Topics 1st Edition. *Engineering*, 1–89.

Granell, C., Gómez, S., & Arenas, A. (2011). Mesoscopic analysis of networks: Applications to exploratory analysis and data clustering. In *Chaos* (Vol. 21, Issue 1). https://doi.org/10.1063/1.3560932

Jiawei Han; Kamber Micheline. (2011). *Data Mining: Concept & Technique*.

Ville, B. de. (2001). Introduction to Data Mining. In *Microsoft Data Mining*. https://doi.org/10.1016/b978-155558242-5/50003-6

Yang, H., Zheng, Z., & Sun, C. (2022). E-Commerce Marketing Optimization of Agricultural Products Based on Deep Learning and Data Mining. *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2022/6564014

Ye, N. (2013). Data Mining: Theories, Algorithms, and Examples. In *Data Mining: Theories, Algorithms, and Examples*.

Yi, J. (2022). Deep Learning in Data Mining Management of Industrial and Commercial Enterprises. *Mobile Information Systems*. https://doi.org/10.1155/2022/6974993

Zhou, Z. H. (2012). Ensemble methods: Foundations and algorithms. In *Ensemble Methods: Foundations and Algorithms*. https://doi.org/10.1201/b12207

BAB 3 PREPROCESSING DATA

Samuel Aleksander Mandowen, S.Si., M.IT.

A. Pengantar Preprocessing Data

1. Definisi Preprocessing Data

Preprocessing data adalah serangkaian langkah yang dilakukan untuk mengubah data mentah menjadi format yang siap untuk dianalisis atau dimodelkan. Tujuan utama preprocessing adalah (Mulyawan, 2024):

a. Meningkatkan Kualitas Data: Menghilangkan atau memperbaiki data yang tidak akurat, tidak lengkap, atau tidak relevan.
b. Mengurangi Noise: Menghapus atau mengurangi data yang dapat mengganggu analisis atau pemodelan.
c. Memastikan Format yang Sesuai: Mengubah data menjadi format yang sesuai dengan persyaratan analisis atau algoritma yang akan digunakan.

Contoh: Dalam sebuah proyek prediksi harga rumah, data mentah mungkin mengandung kesalahan entri, nilai yang hilang, atau format yang tidak konsisten. Preprocessing data membantu memperbaiki dan menyusun data ini agar siap digunakan dalam model prediktif.

2. Pentingnya Preprocessing dalam Analisis Data

Data yang tidak diproses dapat mengandung kesalahan, ketidakkonsistenan, dan informasi yang tidak relevan, yang semuanya dapat mempengaruhi hasil analisis

secara negatif. Beberapa alasan pentingnya preprocessing meliputi (Nurvinda, 2021; Adam, 2022):

a. Menghapus atau Memperbaiki Data yang Salah: Data yang salah atau tidak akurat dapat mempengaruhi hasil analisis dan keputusan bisnis yang diambil berdasarkan data tersebut.
b. Menangani Data yang Hilang: Data yang hilang dapat mengurangi akurasi model prediktif atau analisis data.
c. Mengurangi Noise dan Outlier: Noise dan outlier dapat mengganggu pola yang ada dalam data dan mempengaruhi hasil analisis.
d. Mengubah Data ke Format yang Lebih Mudah Dianalisis: Data harus diubah ke format yang sesuai dengan persyaratan alat analisis atau algoritma.

Contoh: Jika dataset berisi informasi penjualan harian, data yang hilang atau entri yang salah (seperti penjualan yang tidak masuk akal) harus diperbaiki atau dihapus agar analisis penjualan yang dilakukan bisa dipercaya dan akurat.

3. Proses Umum dalam Preprocessing Data

Proses umum dalam preprocessing data meliputi beberapa langkah penting (Firdausi, 2024):

a. Pengumpulan Data (Data Collection): Mengumpulkan data dari berbagai sumber, baik internal maupun eksternal. Contoh: Mengambil data transaksi dari sistem POS dan data pelanggan dari sistem CRM.
b. Pembersihan Data (Data Cleaning): Mengidentifikasi dan memperbaiki atau menghapus data yang tidak akurat, tidak lengkap, atau tidak konsisten. Contoh: Menghapus entri duplikat, mengisi nilai yang hilang dengan mean atau median, dan memperbaiki kesalahan entri data.
c. Integrasi Data (Data Integration): Menggabungkan data dari berbagai sumber menjadi satu dataset yang koheren. Contoh: Menggabungkan data penjualan dari berbagai cabang toko menjadi satu dataset pusat.

d. Transformasi Data (Data Transformation): Mengubah data menjadi format yang sesuai dengan persyaratan analisis atau model. Contoh: Melakukan normalisasi data sehingga semua fitur berada dalam rentang yang sama.
e. Pembagian Data (Data Splitting): Membagi data menjadi set pelatihan (training) dan set pengujian (testing) untuk mengevaluasi model. Contoh: Membagi dataset menjadi 80% untuk pelatihan dan 20% untuk pengujian.

Contoh Proses:

a. Pengumpulan Data: Mengambil data penjualan dari sistem POS dan data demografi pelanggan dari sistem CRM.
b. Pembersihan Data: Menghapus entri yang duplikat dan mengisi nilai yang hilang menggunakan median.
c. Integrasi Data: Menggabungkan data penjualan dan data pelanggan menjadi satu dataset.
d. Transformasi Data: Melakukan normalisasi pada kolom pendapatan dan skala penjualan.
e. Pembagian Data: Membagi dataset menjadi 70% untuk pelatihan dan 30% untuk pengujian model prediksi.

Dengan melakukan preprocessing data, kita memastikan bahwa data yang digunakan dalam analisis atau pemodelan adalah berkualitas tinggi, akurat, dan sesuai dengan format yang diperlukan, sehingga hasil yang diperoleh dapat diandalkan dan bermakna.

B. Pengumpulan Data *(Data Collection)*

1. Sumber Data: Internal dan Eksternal (Lutfi, 2019; Haekal, 2023)

Internal: Data yang dikumpulkan dari dalam organisasi. Contohnya termasuk :

a. Database Pelanggan: Informasi tentang pelanggan seperti nama, alamat, riwayat pembelian.
b. Data Penjualan: Data transaksi penjualan yang mencakup item yang terjual, harga, jumlah, dan tanggal.

c. Log Aktivitas: Catatan aktivitas pengguna atau sistem, seperti log masuk, log keluar, dan aktivitas lainnya dalam sistem IT perusahaan.

Eksternal: Data yang diperoleh dari luar organisasi. Contohnya termasuk:

Data Pasar: Informasi tentang kondisi pasar, harga kompetitor, tren industri.

a. Data dari Media Sosial: Data yang diambil dari platform media sosial seperti Twitter, Facebook, atau Instagram.
b. Data Publik: Data yang tersedia untuk umum seperti data dari lembaga pemerintah, dataset penelitian, atau data sensus.

2. Metode Pengumpulan Data

Berikut adalah beberapa metode pengumpulan data menurut (Latifatunnisa, 2022):

a. Survei: Mengumpulkan data melalui kuesioner yang diberikan kepada responden. Contohnya adalah survei kepuasan pelanggan yang mengukur seberapa puas pelanggan terhadap produk atau layanan.
b. Sensor: Mengumpulkan data secara otomatis dari perangkat sensor. Contohnya adalah sensor IoT yang mengumpulkan data suhu, kelembaban, atau data penggunaan mesin dalam pabrik.
c. Web Scraping: Mengambil data dari situs web menggunakan skrip otomatis. Contohnya adalah mengumpulkan data harga produk dari berbagai situs e-commerce untuk analisis kompetitif.

Contoh: Menggunakan API Twitter untuk mengumpulkan data tweet tentang topik tertentu seperti sentimen publik terhadap merek atau produk.

3. **Tantangan dalam Pengumpulan Data**

Menurut (Lutfi, 2019; Haekal, 2023), tantangan dalam pengumpulan data anata lain:

a. Keterbatasan Akses: Tidak semua data dapat diakses dengan mudah. Data internal mungkin terbatas pada departemen tertentu, sementara data eksternal bisa memerlukan izin atau pembayaran untuk akses.
b. Kualitas Data yang Bervariasi: Data dari sumber yang berbeda mungkin memiliki format dan kualitas yang berbeda. Misalnya, data dari satu sumber mungkin lengkap dan akurat, sementara data dari sumber lain mungkin memiliki banyak nilai yang hilang atau tidak konsisten.
c. Masalah Privasi: Mengumpulkan data pribadi memerlukan izin dan harus mematuhi peraturan privasi seperti GDPR atau CCPA. Pelanggaran privasi dapat mengakibatkan denda yang besar dan kerugian reputasi.

4. **Validasi dan Verifikasi Data yang Dikumpulkan** (Lutfi, 2019; Haekal, 2023):

a. Cross-referencing: Membandingkan data dari berbagai sumber untuk memastikan keakuratan. Misalnya, data penjualan yang dicatat dalam sistem POS dapat diverifikasi dengan laporan keuangan bulanan untuk memastikan kesesuaian.
b. Pemeriksaan Konsistensi: Memastikan bahwa data konsisten di seluruh dataset. Misalnya, memastikan bahwa format tanggal dan mata uang konsisten di seluruh catatan transaksi.
c. Audit: Menggunakan alat dan teknik untuk memeriksa keakuratan dan keandalan data. Contohnya adalah menggunakan alat audit data seperti OpenRefine atau Talend untuk mengidentifikasi dan memperbaiki anomali dalam data.

Contoh Implementasi:

a. Pengumpulan Data: Sebuah perusahaan e-commerce mengumpulkan data penjualan dari sistem POS dan data pelanggan dari CRM. Mereka juga mengambil data harga kompetitor dari situs web e-commerce lain menggunakan web scraping.
b. Validasi Data: Data penjualan dan data pelanggan diperiksa untuk konsistensi dan keakuratan. Data yang hilang atau tidak konsisten diperbaiki atau dihapus.
c. Integrasi Data: Data penjualan dan data pelanggan digabungkan menjadi satu dataset yang siap untuk analisis lebih lanjut. Data harga kompetitor digunakan untuk analisis pasar dan strategi penetapan harga.
d. Penggunaan Data: Dataset yang telah diproses digunakan untuk analisis tren penjualan, segmentasi pelanggan, dan pengembangan strategi pemasaran.

C. Penggabungan Data *(Data Integration)*

1. Teknik Penggabungan Data dari Berbagai Sumber (IBM-United States, no date; Haider, 2024)

a. Join: Menggabungkan data dari dua tabel berdasarkan kolom yang sama. Ini umumnya dilakukan ketika ada kolom yang sama di kedua tabel yang digunakan sebagai kunci penggabungan.
b. Merge: Menggabungkan data dari beberapa sumber menjadi satu dataset. Ini berguna ketika kita memiliki beberapa dataset yang saling terkait dan ingin menggabungkannya menjadi satu dataset yang lengkap.

Contoh: Sebuah perusahaan ingin menggabungkan data pelanggan dari sistem CRM dengan data transaksi dari sistem POS. Mereka bisa menggunakan metode join atau merge berdasarkan kolom ID pelanggan yang sama di kedua dataset.

2. **Resolusi Konflik dan Duplikasi Data**
   a. Deteksi Duplikasi: Menggunakan algoritma untuk mengidentifikasi dan menghapus entri yang duplikat. Ini dilakukan untuk mencegah adanya duplikasi data yang tidak diinginkan dan menjaga kebersihan dataset.
   b. Resolusi Konflik: Ketika ada konflik dalam data (misalnya, dua entri yang berbeda memiliki nilai yang bertentangan), aturan harus ditetapkan untuk menentukan nilai mana yang akan dipilih. Ini bisa berupa memilih nilai terbaru, nilai yang paling sering muncul, atau aturan lain yang sesuai dengan konteks bisnis.

3. **Penggabungan Data dengan Alat Bantu dan Software**
   a. SQL: Bahasa kueri yang digunakan untuk menggabungkan data dari berbagai tabel dalam database. Dengan menggunakan perintah JOIN, data dapat digabungkan berdasarkan kunci yang sesuai.
   b. Pandas (Python): Library Python yang sering digunakan untuk analisis data. Fungsi merge dan join pada Pandas memungkinkan penggabungan data dengan berbagai jenis penggabungan dan strategi.
   c. ETL Tools: Alat ETL (Extract, Transform, Load) seperti Apache Nifi atau Talend memungkinkan untuk mengekstrak data dari sumber yang berbeda, melakukan transformasi pada data tersebut, dan memuatnya ke dalam penyimpanan data yang dituju. Ini berguna untuk penggabungan data yang kompleks dan skala besar.

Contoh Implementasi :

Sebuah perusahaan menggunakan SQL untuk menggabungkan data dari database pelanggan dengan data transaksi menggunakan perintah JOIN. Mereka kemudian menggunakan Pandas untuk membersihkan dan mempersiapkan data lebih lanjut sebelum analisis. Akhirnya, mereka menggunakan Apache Nifi untuk mengotomatiskan alur kerja penggabungan data dari berbagai sumber dan memuatnya ke dalam penyimpanan data pusat.

D. Pengecekan Kualitas Data *(Data Quality Assessment)*

1. **Parameter Kualitas Data** (IBM-United States, no date; Gawande, 2022)
   a. Akurasi: Seberapa tepat data merepresentasikan realitas atau keadaan sebenarnya. Data yang akurat tidak memiliki kesalahan atau distorsi yang signifikan.
   b. Konsistensi: Tingkat keseragaman atau keserupaan data di seluruh dataset. Data yang konsisten memiliki format, struktur, dan nilai yang sama di semua entri.
   c. Kelengkapan: Tingkat keberadaan nilai untuk setiap atribut atau kolom dalam dataset. Data yang lengkap tidak memiliki nilai yang hilang atau kosong.

2. **Metode Pengecekan Kualitas Data** (IBM-United States, no date; Teki, 2022)
   a. Statistik Deskriptif: Menggunakan metrik seperti mean, median, mode, dan deviasi standar untuk menganalisis distribusi dan karakteristik data. Ini membantu mengidentifikasi anomali atau nilai yang ekstrem.
   b. Visualisasi Data: Menggunakan grafik seperti histogram, scatter plot, atau box plot untuk mewakili data secara visual. Ini memungkinkan untuk mendeteksi pola, tren, atau anomali yang mungkin sulit ditemukan melalui analisis statistik.
   c. Validasi Silang: Membandingkan data dengan sumber lain yang dianggap dapat dipercaya untuk memeriksa keakuratan dan kekonsistenan. Ini membantu memvalidasi data dan menemukan kesalahan atau inkonsistensi yang mungkin terjadi.

3. **Alat untuk Audit dan Validasi Data** (IBM-United States, no date; Gawande, 2022)
   a. OpenRefine: Alat sumber terbuka yang digunakan untuk membersihkan, memperbaiki, dan mengubah format data. Ini memungkinkan untuk melakukan operasi penggantian nilai, deteksi duplikat, dan transformasi data lainnya.

b. Talend Data Quality: Alat yang dirancang khusus untuk audit dan peningkatan kualitas data. Ini menyediakan berbagai fungsi seperti profil data, deduplikasi, validasi, dan standarisasi data.
c. Trifacta: Platform untuk mengeksplorasi, mempersiapkan, dan membersihkan data. Ini menggunakan teknik machine learning untuk mengotomatisasi sebagian besar proses dan mempercepat proses pengolahan data.

Contoh Implementasi :

Sebuah tim data menggunakan Talend Data Quality untuk menganalisis data penjualan mereka. Mereka memulai dengan memprofil data untuk mengevaluasi akurasi, konsistensi, dan kelengkapan. Kemudian, mereka menggunakan visualisasi data untuk mengidentifikasi pola atau anomali yang menarik. Setelah itu, mereka menggunakan validasi silang dengan data dari sistem lain untuk memverifikasi keakuratan data mereka. Akhirnya, mereka menggunakan Talend untuk membersihkan data dan menghapus duplikat sebelum digunakan untuk analisis lebih lanjut (IBM-United States, no date; Teki, 2022).

E. Pembersihan Data *(Data Cleaning)*

1. Identifikasi dan Penanganan Data yang Hilang

a. Imputasi: Mengisi nilai yang hilang dengan menggunakan statistik seperti mean, median, atau mode dari kolom yang relevan, atau menggunakan algoritma untuk memprediksi nilai yang hilang berdasarkan pola data yang ada (Scikit-Learn, no date; Prabhakaran, 2023). Contoh: Menggunakan SimpleImputer dari Scikit-Learn untuk mengisi nilai yang hilang dalam dataset.

2. **Penghapusan atau Koreksi Data yang Tidak Konsisten** (noobtomaster, no date)
   a. Regex: Menggunakan ekspresi reguler untuk mencari pola yang tidak konsisten dalam data dan memperbaikinya.
   b. Script: Menulis skrip kustom untuk memperbaiki format atau struktur data yang tidak konsisten.

3. **Penanganan Noise dan Outlier** (Brownlee, 2020)
   a. Metode Statistik: Menggunakan teknik seperti Interquartile Range (IQR) atau Z-score untuk mendeteksi dan menghapus outlier.
   b. Algoritma: Menggunakan algoritma khusus seperti DBSCAN untuk mengidentifikasi noise dan outlier yang tersembunyi.

4. **Teknik Pembersihan Data** (Brownlee, 2020)
   a. Imputasi: Mengisi nilai yang hilang dengan pendekatan statistik atau algoritma yang sesuai.
   b. Interpolasi: Menggunakan nilai tetangga atau pola data untuk memperkirakan nilai yang hilang.
   c. Filtering: Menghapus data yang tidak relevan atau bermasalah, seperti data duplikat atau data yang tidak valid.
   d. Smoothing: Menggunakan teknik smoothing seperti moving average untuk mengurangi noise dalam data berderau.

Contoh Implementasi :

Seorang analis data menggunakan Python dan Pandas untuk membersihkan data penjualan perusahaan. Mereka mengidentifikasi nilai yang hilang dalam kolom harga dan mengisi mereka dengan mean harga dari kolom tersebut menggunakan fungsi fillna(). Selanjutnya, mereka menggunakan ekspresi reguler untuk memperbaiki format yang tidak konsisten dalam kolom alamat. Setelah itu, mereka menggunakan metode IQR untuk menghapus outlier dari kolom pendapatan. Akhirnya, mereka menerapkan

teknik smoothing dengan moving average untuk mengurangi noise dalam kolom waktu.

F. Transformasi Data (Data Transformation)

1. **Konversi Tipe Data** (Huilgol, 2020; Chip, 2023)
   a. Numerik ke Kategorikal: Menggunakan teknik binning untuk mengelompokkan nilai numerik ke dalam kategori diskrit. Ini berguna untuk memperjelas pola atau tren dalam data yang kontinu.
   b. Kategorikal ke Numerik: Menggunakan teknik encoding seperti one-hot encoding atau label encoding untuk mengubah data kategorikal menjadi representasi numerik.

   Contoh: Mengubah kolom umur menjadi kategori (remaja, dewasa, lansia).

2. **Teknik Penskalaan Data** (Huilgol, 2020; Jaiswal, 2024)
   a. Normalisasi: Mengubah data ke dalam rentang 0-1, menjaga proporsi relatif antara nilai.
   b. Standardisasi: Mengubah data ke distribusi normal dengan mean 0 dan standar deviasi 1, membuat distribusi data menjadi lebih simetris.

   Contoh: Menggunakan MinMaxScaler atau StandardScaler dari Scikit-Learn.

3. **Penerapan Transformasi Logaritmik, Eksponensial, dan Lainnya** (Huilgol, 2020; Jaiswal, 2024)
   a. Logaritmik: Menggunakan logaritma alami atau logaritma basis lain untuk mengurangi skewness dalam data yang sangat condong.
   b. Eksponensial: Menggunakan fungsi eksponensial untuk mengubah data sehingga mendekati distribusi normal.

   Contoh: Mengubah data penjualan dengan transformasi logaritmik untuk mengurangi skewness dan membuatnya lebih simetris.

4. **Agregasi Data** (Qlik, no date; Chip, 2023)

   Mengelompokkan data berdasarkan kategori tertentu dan menghitung statistik agregat seperti rata-rata, median, atau total dari setiap kelompok.

   Contoh: Menghitung rata-rata penjualan per bulan atau total pendapatan per kategori produk.

   Contoh Implementasi:

   Seorang analis data menggunakan Python dan Pandas untuk mentransformasi data penjualan. Pertama, mereka menggunakan teknik binning untuk mengelompokkan usia pelanggan menjadi kategori (remaja, dewasa, lansia). Selanjutnya, mereka menerapkan normalisasi menggunakan MinMaxScaler untuk mengubah skala fitur-fitur numerik. Setelah itu, mereka menggunakan transformasi logaritmik untuk mengurangi skewness dari data harga produk. Akhirnya, mereka mengelompokkan data berdasarkan bulan dan menghitung rata-rata penjualan per bulan untuk analisis tren penjualan (Qlik, no date; Chip, 2023; Jaiswal, 2024).

G. Pengkodean Data Kategorikal (*Encoding Categorical Data*)

1. **One-hot Encoding**

   One-hot Encoding: mengubah setiap nilai kategori menjadi vektor biner, di mana setiap elemen vektor mewakili satu kategori. Jika ada N kategori, akan ada N kolom biner, dan setiap kolom menunjukkan kehadiran atau ketidakhadiran suatu kategori.

   Contoh: Mengubah kolom warna (merah, hijau, biru) menjadi kolom biner (merah=1,0,0; hijau=0,1,0; biru=0,0,1) (Mahendra, 2023; Sethi, 2023).

2. **Label Encoding**

   Label Encoding: mengubah setiap nilai kategori menjadi bilangan bulat. Setiap kategori diberi label berdasarkan urutan pengamatan atau secara alfabetis. Ini mengubah kategori menjadi representasi numerik. Contoh:

Mengubah kolom warna (merah, hijau, biru) menjadi (1, 2, 3) (Mahendra, 2023; Sethi, 2023).

3. **Perbandingan Metode Encoding dan Aplikasinya**
   a. One-hot Encoding: Cocok untuk data kategorikal tanpa urutan tertentu. Namun, dapat menghasilkan jumlah kolom yang besar, terutama jika terdapat banyak kategori unik. Ini bisa menjadi masalah dalam model yang kompleks atau dengan dataset besar.
   b. Label Encoding: Lebih efisien untuk dataset besar dengan banyak kategori unik karena hanya menghasilkan satu kolom. Namun, dapat memperkenalkan urutan yang tidak diinginkan pada kategori yang seharusnya tidak berurutan (Mahendra, 2023; Sethi, 2023).

Kesimpulannya, pemilihan antara dua metode ini tergantung pada sifat data dan model yang akan digunakan. Jika tidak yakin, disarankan untuk mencoba keduanya dan memeriksa kinerja model.

H. Reduksi Dimensi (Dimensionality Reduction)

1. **Pentingnya Reduksi Dimensi** (Brownlee, 2020; Przyborowski, 2024)
   a. Mengurangi jumlah fitur membantu menghindari overfitting dalam model, terutama ketika jumlah fitur sangat besar dibandingkan dengan jumlah sampel.
   b. Meningkatkan efisiensi pemodelan dengan mengurangi kompleksitas model dan waktu komputasi.
   c. Meningkatkan interpretabilitas model dengan mengurangi dimensi ruang fitur menjadi ruang yang lebih mudah dipahami.

2. **Teknik-teknik Reduksi Dimensi** (Brownlee, 2020; Przyborowski, 2024)
   a. PCA (*Principal Component Analysis*): Teknik yang paling umum digunakan untuk reduksi dimensi. PCA mengidentifikasi arah (komponen utama) di mana data bervariasi secara maksimal.

b. LDA (*Linear Discriminant Analysis*): Teknik yang berfokus pada memaksimalkan pemisahan antara kelas dalam data. Ini sering digunakan dalam klasifikasi.

3. **Seleksi Fitur vs Ekstraksi Fitur** (Brownlee, 2020; Przyborowski, 2024)
   a. Seleksi Fitur: Memilih subset dari fitur asli yang paling relevan untuk memperbaiki kinerja model atau mengurangi kompleksitas.
   b. Ekstraksi Fitur: Membuat fitur baru yang menggabungkan informasi dari fitur asli, mengurangi dimensi data tetapi mempertahankan informasi penting.

4. **Implementasi dan Evaluasi Teknik Reduksi Dimensi** (Brownlee, 2020)
   a. PCA: Dapat diimplementasikan dengan menggunakan modul PCA dari Scikit-Learn dalam Python. Setelah pelatihan, kita dapat menggunakan atribut `explained_variance_ratio_` untuk mengevaluasi jumlah variansi yang dijelaskan oleh setiap komponen utama.
   b. Evaluasi: Selain memeriksa `explained_variance_ ratio_`, kita juga dapat menggunakan teknik evaluasi yang sesuai dengan tujuan analisis, seperti kinerja model atau interpretasi hasil.

Contoh Implementasi:

Seorang peneliti menggunakan PCA dari Scikit-Learn untuk mengurangi dimensi dataset genetika mereka yang memiliki ribuan fitur. Setelah melatih model PCA, mereka memeriksa explained variance ratio untuk memahami berapa banyak informasi yang dipertahankan oleh setiap komponen utama. Setelah itu, mereka menggunakan komponen utama yang paling informatif untuk analisis lebih lanjut atau pemodelan.

I. Pembagian Data (*Data Splitting*)

1. Metode Pembagian Data untuk Training dan Testing

a. Pembagian Acak (Random Splitting): Memisahkan data menjadi subset training dan testing secara acak dengan proporsi tertentu, misalnya, 70% untuk training dan 30% untuk testing (Mirko Stojiljković, 2022).

b. K-fold Cross-Validation: Memisahkan data menjadi k subset, di mana masing-masing subset digunakan sebagai data testing satu kali, sementara yang lainnya digunakan sebagai data training. Ini membantu dalam evaluasi model yang lebih stabil dan dapat digunakan untuk memilih parameter model yang optimal (KFold-scikit-learn, no date; Pandey, 2020).

2. K-fold Cross-Validation

K-fold cross-validation: membagi data menjadi k subset atau lipatan. Setiap kali, salah satu subset digunakan sebagai data testing, sementara yang lainnya digunakan sebagai data training. Proses ini diulangi k kali, sehingga setiap subset digunakan sebagai data testing tepat satu kali (Brownlee, 2020; Pandey, 2020).

3. Tools untuk Data Splitting

a. Scikit-Learn: Library Python yang populer untuk machine learning menyediakan fungsi `train_test_split` untuk membagi data secara acak menjadi subset training dan testing (KFold-scikit-learn, no date; Mirko Stojiljković, 2022).

b. Cross-validation Functions: Scikit-Learn juga menyediakan berbagai fungsi untuk melakukan k-fold cross-validation, seperti `cross_val_score` atau `KFold` (scikit-learn, no date; Pandey, 2020).

c. Paket-paket lain: Ada juga berbagai library lain dalam Python dan bahasa pemrograman lain yang menyediakan fungsi untuk membagi data, seperti TensorFlow, PyTorch, atau MATLAB (scikit-learn, no date; Pandey, 2020).

Contoh Implementasi:

Seorang peneliti menggunakan Scikit-Learn untuk membagi dataset mereka menjadi subset training dan testing dengan rasio 80:20 menggunakan fungsi `train_test_split`. Selanjutnya, mereka menggunakan k-fold cross-validation dengan k=5 untuk mengevaluasi model mereka dengan lebih baik. Ini dilakukan dengan menggunakan fungsi `cross_val_score` dari Scikit-Learn (scikit-learn, no date; Pandey, 2020; Mirko Stojiljković, 2022).

J. Kesimpulan dan Best Practices

1. Rangkuman Proses Preprocessing Data

Proses preprocessing data meliputi langkah-langkah penting seperti pengumpulan data, pembersihan data, integrasi data, transformasi data, dan pembagian data untuk pelatihan dan pengujian. Langkah-langkah ini bertujuan untuk mempersiapkan data mentah sebelum masuk ke tahap analisis atau pemodelan lebih lanjut.

2. Best Practices dalam Preprocessing Data

a. Pengumpulan Data yang Teliti: Pastikan data yang dikumpulkan bersih, konsisten, dan relevan dengan tujuan analisis.
b. Pembersihan Data yang Komprehensif: Identifikasi dan tangani nilai yang hilang, data tidak konsisten, noise, dan outlier dengan tepat.
c. Transformasi Data yang Relevan: Terapkan transformasi data seperti normalisasi, encoding kategori, atau reduksi dimensi sesuai dengan kebutuhan analisis.
d. Pembagian Data yang Tepat: Pisahkan data dengan bijaksana menjadi subset training dan testing untuk evaluasi model yang akurat.

3. Tantangan Umum dan Cara Mengatasinya

a. Data yang Tidak Lengkap: Tangani data yang hilang dengan imputasi atau penghapusan berdasarkan konteks dan kebutuhan analisis.

b. Data yang Tidak Konsisten: Identifikasi dan perbaiki data yang tidak konsisten dengan metode pembersihan data yang tepat.
c. Overfitting karena Dimensi yang Tinggi: Kurangi dimensi data menggunakan teknik reduksi dimensi seperti PCA atau seleksi fitur untuk menghindari overfitting.
d. Evaluasi yang Tidak Akurat: Gunakan teknik evaluasi yang sesuai seperti k-fold cross-validation untuk mendapatkan estimasi yang lebih stabil tentang kinerja model.

DAFTAR PUSTAKA

Adam, A. (2022) *Data Preprocessing: Pengertian, Manfaat, dan Tahapan Kerjanya - Accurate Online*. Available at: https://accurate.id/teknologi/data-preprocessing/ (Accessed: 24 May 2024).

Brownlee, J. (2020) *kNN Imputation for Missing Values in Machine Learning - MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/ (Accessed: 24 May 2024).

Chip (2023) *6 Data Transformation Techniques | QuantHub*. Available at: https://www.quanthub.com/data-wrangling-data-transformation-techniques/ (Accessed: 24 May 2024).

Firdausi, S.K.A. (2024) *dibimbing.id - Data Preprocessing Adalah: Pengertian, Manfaat, & Tahapannya*. Available at: https://dibimbing.id/blog/detail/data-preprocessing-adalah-pengertian-manfaat-tahapannya (Accessed: 24 May 2024).

Gawande, S. (2022) *A Guide for Data Quality (DQ) and 6 Data Quality Dimensions*. Available at: https://icedq.com/6-data-quality-dimensions (Accessed: 24 May 2024).

Haekal, M. (2023) *Memahami Lebih Dalam: 'Data Eksternal' dalam Dunia Digital | Tebidu Media*. Available at: https://www.tebidu.com/memahami-lebih-dalam-data-eksternal-dalam-dunia-digital/ (Accessed: 24 May 2024).

Haider, K. (2024) *What is Data Integration? Definition, Benefits, & Best Practices*. Available at: https://www.astera.com/type/blog/data-integration/ (Accessed: 24 May 2024).

Huilgol, P. (2020) *9 Feature Transformation & Scaling Techniques| Boost Model Performance*. Available at: https://www.analyticsvidhya.com/blog/2020/07/types-of-

feature-transformation-and-scaling/ (Accessed: 24 May 2024).

IBM-United States (no date) *What Is Data Integration? | IBM*. Available at: https://www.ibm.com/topics/data-integration (Accessed: 24 May 2024).

Jaiswal, S. (2024) *What is Normalization in Machine Learning? A Comprehensive Guide to Data Rescaling | DataCamp*. Available at: https://www.datacamp.com/tutorial/normalization-in-machine-learning (Accessed: 24 May 2024).

KFold-scikit-learn (no date) *KFold — scikit-learn 1.5.0 documentation*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html (Accessed: 24 May 2024).

Latifatunnisa, H. (2022) *Metode Pengumpulan Data: Jenis dan Langkah-langkah 2024 | RevoU*. Available at: https://revou.co/panduan-teknis/metode-pengumpulan-data (Accessed: 24 May 2024).

Lutfi, E. (2019) *Klasifikasi Data: Pengertian, Jenis, Hingga Metodenya – Mekari Talenta*. Available at: https://www.talenta.co/blog/klasifikasi-data-2/ (Accessed: 24 May 2024).

Mahendra, S. (2023) *One-Hot Encoding Is Great for Machine Learning - Artificial Intelligence +*. Available at: https://www.aiplusinfo.com/blog/one-hot-encoding-is-great-for-machine-learning/ (Accessed: 24 May 2024).

Mirko Stojiljković (2022) *Split Your Dataset With scikit-learn's train_test_split() – Real Python*. Available at: https://realpython.com/train-test-split-python-data/#reader-comments (Accessed: 24 May 2024).

Mulyawan, R. (2024) *Data Preprocessing: Pengertian, Arti, Fungsi, Kegunaan, Contoh, serta Penjelasannya!* Available at: https://rifqimulyawan.com/literasi/data-preprocessing/ (Accessed: 24 May 2024).

noobtomaster (no date) *Handling missing data and data cleaning techniques - Scikit Learn*. Available at: https://noobtomaster.com/scikit-learn/handling-missing-data-and-data-cleaning-techniques/ (Accessed: 24 May 2024).

Nurvinda, G. (2021) *Pentingnya Preprocessing dalam Pengolahan Data Statistik*. Available at: https://dqlab.id/pentingnya-preprocessing-dalam-pengolahan-data-statistik (Accessed: 24 May 2024).

Pandey, P. (2020) *Cross-Validation in scikit-learn – Machine Learning Geek*. Available at: https://machinelearninggeek.com/cross-validation-in-scikit-learn/ (Accessed: 24 May 2024).

Prabhakaran, S. (2023) *Missing Data Imputation Approaches | How to handle missing values in Python | MLPlus*. Available at: https://www.machinelearningplus.com/machine-learning/missing-data-imputation-how-to-handle-missing-values-in-python/ (Accessed: 24 May 2024).

Przyborowski, M. (2024) *Dimensionality Reduction - Popular Techniques and How to Use Them*. Available at: https://nexocode.com/blog/posts/dimensionality-reduction-techniques-guide/ (Accessed: 24 May 2024).

Qlik (no date) *What is Data Transformation? | Qlik*. Available at: https://www.qlik.com/us/data-management/data-transformation (Accessed: 24 May 2024).

scikit-learn (no date) *3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.5.0 documentation*. Available at: https://scikit-learn.org/stable/modules/cross_validation.html (Accessed: 24 May 2024).

Scikit-Learn (no date) *6.4. Imputation of missing values — scikit-learn 1.5.0 documentation*. Available at: https://scikit-learn.org/stable/modules/impute.html (Accessed: 24 May 2024).

Sethi, A. (2023) *Categorical Encoding | One Hot Encoding vs Label Encoding*. Available at: https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/ (Accessed: 24 May 2024).

Teki, S. (2022) *Understanding and measuring data quality | Openlayer*. Available at: https://www.openlayer.com/blog/post/understanding-and-measuring-data-quality (Accessed: 24 May 2024).

BAB 4 DATA CLEANING

Alvian M. Sroyer, S.Si., M.Si.

A. Pendahuluan

Data *cleaning* adalah langkah kritis dalam proses analisis data, di mana data yang tidak akurat, tidak lengkap, atau tidak relevan diidentifikasi dan diperbaiki atau dihapus. Proses ini memastikan bahwa data yang digunakan untuk analisis adalah akurat dan konsisten, sehingga menghasilkan hasil yang lebih baik dan dapat diandalkan. Dalam bab ini, penulis akan membahas definisi data *cleaning*, pentingnya data *cleaning*, dan berbagai teknik serta alat yang dapat digunakan dalam proses ini.

B. Definisi Data Cleaning

Data *cleaning* juga dikenal sebagai data *cleansing* atau data *scrubbing*. Data *cleaning* adalah proses mendeteksi dan memperbaiki atau menghapus data yang korup atau tidak akurat dari *dataset*. Tujuan utama dari data *cleaning* adalah untuk meningkatkan kualitas data yang mencakup keakuratan, konsistensi dan kelengkapan.

C. Pentingnya Data Cleaning

Data yang baik atau bersih sangat penting dalam berbagai konteks, terutama dalam menganalisis data dan *mechine learning*. Beberapa alasan pentingnya data *cleaning* sebagai berikut :

1. Keakuratan Hasil Analisis

Keakuratan hasil analisis merupakan salah satu faktor yang sangat penting dan dipengaruhi oleh proses pembersihan data. Analisis data yang akurat memiliki peranan yang sangat penting dalam berbagai konteks, baik itu dalam penelitian akademis, bisnis, maupun pengambilan keputusan pemerintahan. Beberapa alasan mengapa pembersihan data sangat penting untuk mencapai keakuratan hasil analisis,

a. Mengurangi Kesalahan dan Bias dalam Data

Data yang diinput secara manual sering kali mengandung kesalahan, seperti kesalahan pengetikan atau format yang tidak konsisten. Pembersihan data membantu mengidentifikasi dan memperbaiki kesalahan-kesalahan ini, sehingga hasil analisis yang dihasilkan menjadi lebih akurat. Data yang tidak bersih dapat mengandung bias sampling, yang berarti sampel data mungkin tidak mewakili populasi secara keseluruhan. Dengan melakukan pembersihan data, analisis yang dilakukan akan menjadi lebih representatif dan akurat.

b. Meningkatkan Keandalan Statistik

Adanya outlier atau nilai ekstrim dalam data dapat berdampak signifikan terhadap hasil analisis statistik. Proses pembersihan data melibatkan identifikasi dan penanganan outlier untuk memastikan bahwa analisis tidak terpengaruh oleh data yang tidak biasa atau salah. Kehilangan data dapat mengurangi keandalan hasil analisis. Pembersihan data melibatkan pengisian atau penghapusan nilai yang hilang dengan benar, sehingga analisis tetap valid dan dapat diandalkan.

c. Meningkatkan Validitas Hasil Analisis

Data dari berbagai sumber seringkali memiliki inkonsistensi dalam format atau terminologi. Pembersihan data memastikan bahwa semua data konsisten dan sesuai standar, sehingga analisis yang dilakukan berdasarkan data tersebut lebih valid. Adanya

data yang terduplikasi dapat menghasilkan analisis yang bias atau menyesatkan. Menghapus duplikasi merupakan bagian penting dari pembersihan data yang meningkatkan validitas analisis.

d. Optimisasi Algoritma Analisis dan Machine Learning

Kualitas data input sangat penting dalam algoritma analisis dan machine learning. Data yang tidak bersih dapat mengurangi efektivitas algoritma dan menghasilkan model yang tidak akurat atau bias. Preprocessing data melibatkan langkah-langkah seperti data cleaning untuk memastikan data siap untuk dianalisis dan digunakan dalam model machine learning. Hal ini meliputi normalisasi, standarisasi, dan transformasi data sesuai kebutuhan analisis.

e. Pengurangan Biaya dan Waktu dalam Proses Analisis

Dengan menggunakan data yang bersih, waktu yang dibutuhkan untuk mengelola dan memproses data selama analisis dapat dikurangi. Hal ini memungkinkan para analis untuk lebih fokus pada interpretasi hasil dan pengambilan keputusan, bukan pada pemrosesan data yang bermasalah. Dengan melakukan data cleaning yang baik, kebutuhan untuk melakukan analisis ulang setelah masalah teridentifikasi dapat dihindari. Hal ini dapat menghemat waktu dan biaya, serta meningkatkan efisiensi secara keseluruhan.

2. Keadalan Model

Keandalan model *machine learning* sangat bergantung pada kualitas data yang digunakan untuk melatih dan menguji model tersebut. Pentingnya membersihkan data sebelum digunakan dalam model *machine learning* tidak bisa diabaikan. Adapun alasan mengapa data *cleaning* sangat penting untuk menjaga keandalan model *machine learning* :

a. Menghindari Bias dan *Overfitting*

Data yang tidak bersih dapat mengandung bias yang dapat merusak model *machine learning*. Misalnya, data yang terduplikasi atau tidak representatif dapat menyebabkan model belajar pola yang tidak benar atau tidak relevan. Data yang kotor atau berisik dapat menyebabkan model *machine learning overfit,* di mana model terlalu sesuai dengan data pelatihan tetapi tidak berkinerja baik pada data baru. Dengan membersihkan data, kita dapat menghilangkan *noise* dan pola yang tidak relevan, sehingga mengurangi risiko *overfitting*.

b. Meningkatkan Generalisasi Model

Model *machine learning* harus mampu generalisasi dari data pelatihan ke data baru. Ketidaksesuaian dalam data, seperti format yang tidak standar atau nilai yang tidak konsisten, dapat menghambat kemampuan model untuk generalisasi . Pembersihan data memastikan konsistensi, sehingga model dapat bekerja lebih baik pada data baru. Pembersihan data membantu memastikan bahwa data yang digunakan untuk melatih model adalah representatif dari masalah yang ingin dipecahkan. Ini melibatkan penanganan nilai yang hilang dan penghapusan data yang tidak relevan, sehingga model dilatih dengan informasi yang tepat dan relevan.

c. Meningkatkan Kinerja Model

Nilai yang hilang dalam *dataset* dapat menyebabkan masalah dalam pelatihan model *machine learning*. Pembersihan data melibatkan pengisian atau penghapusan nilai yang hilang dengan cara yang sesuai, sehingga model dapat dilatih tanpa terganggu oleh data yang tidak lengkap. Pencilan atau nilai ekstrim dapat mempengaruhi kinerja model. Pembersihan data membantu mengidentifikasi dan mengatasi pencilan, sehingga model dapat belajar dari data yang lebih representatif dan kurang dipengaruhi oleh nilai yang tidak biasa.

d. Optimasi Proses *Feature Engineering*

Proses menciptakan fitur dari data mentah sangat penting dalam *machine learning*. Data yang bersih mempermudah proses ini, karena fitur yang relevan dan bermakna dapat diekstraksi dengan lebih efektif. Pembersihan data memastikan bahwa data dalam kondisi optimal untuk proses *feature engineerin*g. Data yang bersih membantu dalam seleksi fitur, di mana fitur yang tidak relevan dapat dihapus. Hal ini tidak hanya menyederhanakan model tetapi juga meningkatkan interpretasi dan kinerja model.

e. Mengurangi Biaya dan Waktu dalam Pengembangan Model

Data yang bersih mengurangi waktu yang diperlukan untuk membersihkan dan memproses data selama pengembangan model. Model dengan data kotor sering kali memerlukan *re-train* setelah masalah teridentifikasi. Pembersihan data yang baik di awal proses menghindari kebutuhan untuk re-train, sehingga menghemat waktu dan biaya.

3. Efisiensi Proses

Efisiensi proses adalah faktor penting yang dipengaruhi oleh kualitas data. Proses pembersihan data yang efektif dan menyeluruh dapat secara signifikan meningkatkan efisiensi dalam berbagai tahap analisis data dan pengembangan model.

Beberapa cara di mana pembersihan data berkontribusi terhadap efisiensi proses.

a. Mengurangi Beban Kerja Manual

Pembersihan data yang dilakukan dengan menggunakan alat dan skrip otomatis dapat mengurangi beban kerja manual yang biasanya diperlukan untuk memeriksa dan memperbaiki data secara manual. Hal ini dapat menghemat waktu dan tenaga para analis dan ilmuwan data. Dengan data yang bersih sejak awal, alur

kerja analisis menjadi lebih lancar dan tidak terhambat oleh masalah data yang perlu diperbaiki di tengah proses.

b. Meningkatkan Kecepatan Analisis

Data yang bersih memungkinkan proses analisis data berjalan lebih cepat karena tidak perlu lagi melakukan langkah-langkah tambahan untuk memperbaiki kesalahan atau membersihkan data saat analisis sedang berlangsung. Algoritma analisis dan machine learning dapat berjalan lebih efisien pada data yang bersih, mengurangi waktu komputasi yang diperlukan untuk menjalankan model dan analisis.

c. Mengurangi Kesalahan dan Revisi

Dengan data yang bersih, hasil analisis awal cenderung lebih akurat dan bebas dari kesalahan yang disebabkan oleh data yang tidak valid. Hal ini mengurangi kebutuhan untuk melakukan revisi dan analisis ulang, yang dapat menghemat waktu dan sumber daya. Proses validasi dan verifikasi data menjadi lebih sederhana dan cepat jika data telah dibersihkan dengan baik sebelumnya.

d. Optimalisasi Sumber Daya

Dengan data yang telah dibersihkan, tim data dapat menggunakan sumber daya mereka secara lebih efisien, fokus pada analisis lanjutan dan pengembangan model tanpa terganggu oleh masalah data. Mengurangi waktu yang dihabiskan untuk membersihkan data secara manual atau menangani masalah yang timbul akibat data yang kotor dapat mengurangi biaya operasional secara keseluruhan.

e. Meningkatkan Kolaborasi dan Konsistensi

Data yang telah dibersihkan dan distandarisasi dapat dibagikan dengan berbagai tim dalam organisasi tanpa kekhawatiran tentang inkonsistensi atau kesalahan. Hal ini memfasilitasi kolaborasi yang lebih baik dan keputusan yang lebih konsisten di seluruh organisasi.

Proses pembersihan data sering kali melibatkan dokumentasi perubahan yang dilakukan, yang membantu dalam pelacakan dan audit data di masa mendatang, memastikan bahwa semua tim bekerja dengan data yang memiliki riwayat yang jelas.

f. Mendukung Pengambilan Keputusan Cepat

Data yang bersih memungkinkan pengambilan keputusan yang cepat karena data yang digunakan sudah siap untuk dianalisis tanpa perlu pembersihan tambahan. Dengan data yang bersih, para pengambil keputusan dapat memiliki kepercayaan lebih besar pada keakuratan dan keandalan data, memungkinkan mereka membuat keputusan yang lebih cepat dan lebih tepat.

4. Keputusan

Pengambilan keputusan yang lebih baik adalah hasil dari penggunaan data yang efektif dan akurat. Data cleaning memainkan peran penting dalam memastikan bahwa keputusan yang diambil didasarkan pada informasi yang benar dan relevan. Manfaat dari data *cleaning* dalam meningkatkan kualitas pengambilan keputusan diantaranya:

a. Meningkatkan Keakuratan Informasi

Data yang tidak akurat dapat mengarah pada keputusan yang salah. Dengan melakukan proses data *cleaning*, kesalahan dalam data dapat diidentifikasi dan diperbaiki, sehingga informasi yang digunakan untuk pengambilan keputusan menjadi lebih akurat dan dapat diandalkan. Dengan membersihkan dan menstandarisasi data, konsistensi data dapat terjaga.

b. Memberikan Dasar yang Kuat untuk Analisis

Data yang telah dibersihkan memungkinkan analisis yang lebih valid dan kuat. Hasil analisis ini dapat digunakan sebagai dasar yang solid dalam pengambilan keputusan. Keputusan yang didasarkan pada analisis data yang bersih lebih dapat dipercaya karena informasinya telah diverifikasi dan disempurnakan.

c. Mengurangi Risiko Kesalahan dalam Pengambilan Keputusan

Data yang terbebas dari kesalahan membantu mengurangi bias yang mungkin ada dalam dataset awal. Dengan demikian, keputusan yang diambil menjadi lebih objektif dan didasarkan pada fakta yang ada. Mengidentifikasi Tren dan Melakukan pembersihan data memungkinkan pengenalan tren dan pola yang akurat dalam data, membantu pengambil keputusan untuk merespons dinamika pasar atau situasi dengan tepat.

d. Meningkatkan Efisiensi dalam Proses Pengambilan Keputusan

Dengan adanya data yang bersih, proses analisis dan interpretasi data menjadi lebih cepat dan efisien, memungkinkan pengambil keputusan untuk bertindak dengan lebih cepat. Data yang bersih mengurangi kebutuhan untuk melakukan revisi dan analisis ulang yang disebabkan oleh kesalahan data, sehingga pengambilan keputusan dapat dilakukan dengan lebih cepat dan lebih percaya diri.

e. Mendukung Keputusan Strategis

Data yang bersih dan akurat sangat penting untuk pengambilan keputusan strategis, seperti perencanaan jangka panjang, investasi, dan pengembangan produk. Keputusan yang didukung oleh data berkualitas memiliki peluang keberhasilan yang lebih tinggi. Dengan adanya data yang telah dibersihkan, model prediksi dan perencanaan menjadi lebih akurat, mendukung pengambilan keputusan strategis yang lebih baik.

f. Meningkatkan Kepercayaan Stakeholder

Keputusan yang diambil berdasarkan data yang bersih meningkatkan kepercayaan dari investor, mitra bisnis, dan stakeholder lainnya, karena menunjukkan bahwa organisasi memiliki kendali yang baik terhadap data dan analisisnya. Pemanfaatan data berkualitas tinggi

dalam pengambilan keputusan dapat meningkatkan reputasi organisasi sebagai entitas yang berbasis data dan dapat dipercaya.

g. Dukungan Kepatuhan dan Audit

Data yang bersih membantu organisasi mematuhi regulasi dan standar industri, yang sangat penting untuk pengambilan keputusan yang patuh terhadap hukum. Proses audit internal dan eksternal menjadi lebih mudah dengan data yang telah dibersihkan, mengurangi risiko kesalahan dan memastikan transparansi dalam pengambilan keputusan.

5. Proses Data Cleaning

Proses data *cleaning* terdiri dari beberapa langkah kunci yang meliputi identifikasi masalah data, pembersihan data, dan verifikasi hasil.

a. Identifikasi Masalah Data

Langkah awal dalam proses pembersihan data adalah mengidentifikasi masalah yang ada dalam dataset. Tahap ini sangat penting karena pemahaman yang baik tentang kondisi data awal akan menentukan efektivitas langkah-langkah pembersihan selanjutnya.

Beberapa jenis masalah data yang sering ditemui dan metode untuk mengidentifikasinya :

1) Nilai yang Hilang (*Missing Values*)

Nilai yang hilang dapat terjadi karena berbagai alasan, seperti kesalahan input, masalah dalam proses pengumpulan data, atau kegagalan teknis. Identifikasi nilai yang hilang sangat penting karena dapat mempengaruhi hasil analisis secara signifikan. Metode Identifikasi, menggunakan statistik deskriptif untuk menghitung jumlah dan persentase nilai yang hilang di setiap kolom. Serta menggunakan visualisasi heatmap untuk melihat distribusi nilai yang hilang dalam dataset. Pemeriksaan Manual:

Melakukan inspeksi manual pada dataset untuk mendeteksi pola nilai yang hilang.

2) Duplikasi Data

Duplikasi data terjadi ketika baris yang sama muncul lebih dari sekali dalam dataset, yang bisa disebabkan oleh penggabungan dataset yang tidak sempurna atau kesalahan input. Metode Identifikasi, pemeriksaan duplikasi menggunakan fungsi '*drop duplicates( )*' dalam pustaka seperti pandas di Python untuk mengidentifikasi baris yang terduplikasi. Juga menggunakan visualisasi seperti *scatter plot* untuk mengidentifikasi duplikasi data secara visual. Serta nnalisis kunci primer, dengan memeriksa kolom yang seharusnya memiliki nilai unik (misalnya, ID pengguna) untuk mengidentifikasi duplikasi.

3) Kesalahan Penulisan dan Format

Kesalahan dalam penulisan dan format yang tidak konsisten dapat terjadi pada data teks dan angka, seperti variasi dalam penulisan nama, format tanggal yang berbeda, atau ketidakkonsistenan dalam penggunaan unit pengukuran. Metode Identifikasi, menggunakan *regex* untuk mendeteksi pola yang tidak sesuai dalam data teks. - Menggunakan alat profil data seperti *Pandas Profiling* untuk mengidentifikasi format yang tidak konsisten. Dan menggunakan histogram untuk melihat distribusi data numerik dan mendeteksi anomali dalam format.

4) Inkonistensi Data

Inkonistensi dalam data terjadi ketika ada perbedaan dalam terminologi atau unit pengukuran yang digunakan dalam dataset yang sama. Metode Identifikasi, mengelompokkan data berdasarkan kolom yang seharusnya konsisten dan memeriksa adanya variasi yang tidak sesuai. Memeriksa nilai data terhadap daftar nilai yang valid atau standar

(misalnya, daftar negara atau kode pos yang valid). Serta melakukan inspeksi manual untuk mendeteksi ketidakkonsistenan yang mungkin tidak terdeteksi oleh metode otomatis.

5) Outlier

Outlier adalah nilai yang ekstrem atau tidak biasa yang mungkin tidak mewakili populasi data secara keseluruhan dan dapat mempengaruhi hasil analisis secara signifikan. Metode Identifikasi, menggunakan statistik deskriptif seperti mean, median, dan standar deviasi untuk mendeteksi nilai yang jauh di luar rentang normal. Menggunakan box plot untuk visualisasi data dan mendeteksi outlier secara grafis. Dan menghitung Z-score untuk setiap titik data untuk mengidentifikasi nilai yang berada jauh di luar distribusi normal.

b. Pembersihan Data

Setelah mengidentifikasi masalah dalam dataset, langkah selanjutnya adalah melakukan pembersihan data untuk memastikan bahwa data yang digunakan dalam analisis dan pengembangan model adalah bersih, akurat, dan konsisten. Pembersihan data mencakup berbagai teknik dan metode untuk menangani masalah yang telah diidentifikasi pada tahap sebelumnya. Langkah-langkah dalam proses data *cleaning* sebagai berikut :

1) Penanganan Nilai yang Hilang (Missing Values)

Nilai yang hilang bisa diatasi dengan berbagai cara tergantung pada jumlah dan distribusi missing values serta pentingnya kolom terkait dalam analisis.

Menghapus Baris atau Kolom

Jika nilai yang hilang relatif sedikit, baris atau kolom yang mengandung missing values dapat dihapus. Metode ini cocok ketika data yang hilang tidak signifikan terhadap keseluruhan dataset.

Contoh 4.1:

data.dropna(axis=0) untuk menghapus baris,
data.dropna(axis=1) untuk menghapus kolom dalam pustaka pandas di Python.

Mengisi dengan Nilai Tertentu

Mengganti nilai yang hilang dengan nilai rata-rata, median, modus, atau nilai tetap lainnya. Cocok untuk data numerik dan kategorikal.

Contoh 4.2:

data['column'].fillna(data['column'].mean())
untuk mengisi nilai yang hilang dengan rata-rata kolom tersebut.

Interpolasi

Menggunakan metode interpolasi untuk memperkirakan nilai yang hilang berdasarkan nilai lainnya dalam dataset. Berguna untuk data time series atau data yang memiliki hubungan linear.

Contoh 4.3:

data.interpolate() dalam pustaka pandas.

2) Eliminasi Data yang Terduplikasi

Duplikasi data dapat mengakibatkan bias dalam analisis dan harus dihapus untuk memastikan keakuratan hasil. Identifikasi dan Penghapusan. Gunakan fungsi *drop_duplicate*s( ) dalam pandas untuk mengidentifikasi dan menghapus baris yang terduplikasi.

Contoh 4.4:

data.drop_duplicates( ).

c. Koreksi Kesalahan Penulisan dan Format

Kesalahan penulisan dan format yang tidak konsisten dapat diperbaiki untuk memastikan data konsisten dan mudah dianalisis.

Standarisasi Format Data

Pastikan bahwa semua entri dalam kolom memiliki format yang konsisten, seperti format tanggal yang seragam atau konversi satuan pengukuran.

Contoh 4.5:
data['date_column']=
pd.to_datetime(data['date_column']) untuk menyatukan format tanggal. Koreksi Kesalahan.

Penulisan

Menggunakan metode seperti *regex* atau alat profil data untuk mendeteksi dan memperbaiki kesalahan penulisan.

Contoh 4.6 : data['text_column'].str.replace('misspelled', 'correct').

d. Normalisasi dan Standarisasi Data

Normalisasi dan standarisasi memastikan bahwa data berada dalam skala atau format yang konsisten, yang penting untuk analisis yang akurat.

Normalisasi
Ubah data ke dalam rentang tertentu, biasanya [0, 1].
Contoh 4.7:
data['column'] = (data['column'] - data['column'].min()) / (data['column'].max() - data['column'].min()).
Standarisasi: Ubah data sehingga memiliki mean 0 dan standar deviasi 1.

e. Penanganan Outlier

Outlier adalah nilai yang sangat ekstrem atau tidak biasa yang dapat mempengaruhi hasil analisis dan perlu ditangani dengan hati-hati.

Identifikasi Outlier

Menggunakan metode statistik seperti Z-score atau IQR untuk mengidentifikasi outlier.

Contoh 4.8 :

data[(np.abs(stats.zscore(data['kolom'])) < 3)] untuk menghapus outlier berdasarkan Z-score.

Penanganan Outlier

Menghapus outlier jika dianggap sebagai kesalahan atau tidak relevan. Mengubah nilai outlier menggunakan metode seperti winsorizing.

Contoh 4.9:

data['kolom'] = np.where(data['kolom'] > batas_atas, batas_atas, data['kolom']).

f. Konsolidasi Data

Menggabungkan data dari berbagai sumber atau mengintegrasikan berbagai dataset menjadi satu dataset yang konsisten.

Penggabungan Data

Menggunakan metode *join* atau *merge* untuk menggabungkan dataset berdasarkan kunci tertentu.

Contoh 4.10:

data = pd.merge(data1, data2, on='kolom_kunci').

Resolusi Konflik

Mengidentifikasi dan menyelesaikan konflik atau ketidakkonsistenan dalam data yang digabungkan.

Contoh 4.11:

Memilih sumber data yang lebih dipercaya atau menggunakan agregasi data.

g. Validasi Hasil

Setelah data dibersihkan, langkah selanjutnya adalah memverifikasi bahwa semua masalah telah diatasi dengan benar.

Validasi Data

Menggunakan alat dan teknik untuk memastikan bahwa data bersih dan siap untuk analisis.

Contoh 4.12:

assert data.isnull().sum().sum() == 0 untuk memastikan tidak ada nilai yang hilang.

Cross-Validation

Memeriksa konsistensi data antara subset yang berbeda untuk memastikan bahwa pembersihan telah dilakukan dengan benar.

Contoh 4.13:

Membandingkan distribusi data sebelum dan sesudah pembersihan.

h. Verifikasi Hasil

Setelah selesai proses pembersihan data, langkah berikutnya adalah memverifikasi hasil untuk memastikan bahwa data yang telah dibersihkan memenuhi standar kualitas yang diperlukan dan siap digunakan dalam analisis atau pengembangan model. Verifikasi hasil melibatkan berbagai teknik dan alat untuk memastikan bahwa semua masalah yang diidentifikasi telah diperbaiki dengan benar dan bahwa data yang dihasilkan konsisten, akurat, dan lengkap.

Langkah-langkah dalam verifikasi hasil sebagai berikut :

1) Validasi Data

Validasi data adalah langkah untuk memeriksa data yang telah dibersihkan untuk memastikan bahwa semua masalah yang teridentifikasi sebelumnya telah diatasi dan bahwa data sekarang akurat dan konsisten. Pemeriksaa.

Konsistensi

Memastikan bahwa tidak ada nilai yang hilang, duplikasi, atau kesalahan penulisan setelah proses pembersihan.

Contoh 4.14:

assert data.isnull().sum().sum() == 0 memastikan bahwa tidak ada nilai yang hilang.

Statistik Deskriptif

Menggunakan statistik deskriptif untuk memeriksa distribusi data dan memastikan bahwa tidak ada nilai yang ekstrem atau tidak wajar yang tersisa.

Contoh 4.15:

data.describe() memberikan ringkasan statistik dari dataset.

2) Cross-Validation

Cross-validation adalah teknik untuk memeriksa konsistensi data antara *subset* yang berbeda dalam *dataset*. Hal ini penting untuk memastikan bahwa pembersihan data dilakukan secara merata dan tidak ada bagian data yang terlewat.

Pembagian Data

Membagi data menjadi beberapa subset dan memeriksa apakah masing-masing subset memiliki konsistensi yang sama.

Contoh 4.16:

data.sample(frac=0.5) untuk membagi data menjadi subset dan memeriksa konsistensinya.

Perbandingan Antar Subset

Membandingkan statistik deskriptif dan distribusi data antara subset yang berbeda.

Contoh 4.17:

Menggunakan histogram atau box plot untuk membandingkan distribusi nilai antar subset.

3) Representasi Data

Representasi data adalah alat yang sangat efektif untuk memverifikasi hasil pembersihan data. Dengan menggunakan representasi visual, kita dapat dengan cepat mengidentifikasi pola, anomali, dan inkonsistensi yang mungkin tidak terlihat dari analisis statistik saja.

Plot Histogram dan Box Plot

Gunakan histogram untuk melihat distribusi data dan box plot untuk mengidentifikasi outlier.

Contoh 4.18:

data['kolom'].plot(kind='hist') untuk histogram dan data.boxplot(column='kolom') untuk box plot.

Scatter Plot dan Pair Plot

Gunakan scatter plot untuk memvisualisasikan hubungan antara dua variabel dan pair plot untuk memeriksa hubungan di antara beberapa variabel.

Contoh 4.19:

pd.plotting.scatter_matrix(data) untuk pair plot.

4) Pengecekan Integritas Data

Pengecekan integritas data memastikan bahwa data yang telah dibersihkan tidak hanya bebas dari masalah yang telah diidentifikasi tetapi juga sesuai dengan aturan dan logika bisnis yang berlaku.

Aturan Bisnis

Pastikan bahwa data mematuhi aturan dan logika bisnis yang telah ditentukan.

Contoh 4.20:

Memastikan bahwa kolom usia tidak memiliki nilai negatif atau nilai yang tidak masuk akal.

Konsistensi Antar Kolom

Periksa hubungan antar kolom untuk memastikan konsistensi logis.

Contoh 4.21:

Memastikan bahwa tanggal lahir dan umur konsisten satu sama lain.

5) **Dokumentasi dan Jejak Audit**

Dokumentasi adalah langkah penting untuk memastikan bahwa semua tindakan pembersihan dan verifikasi tercatat dengan baik. Ini membantu dalam audit dan juga berguna untuk referensi di masa mendatang.

Dokumentasi Langkah-Langkah

Catat semua langkah yang diambil selama proses pembersihan dan verifikasi data.

Contoh 4.22:

Menyimpan skrip pembersihan data dan hasil analisis dalam repository version control seperti Git.

Audit Trail

Mencatat setiap perubahan yang terjadi pada data untuk memastikan jejaknya dapat dilacak.

Contoh 4.23:

Membuat log perubahan yang mencatat setiap modifikasi pada dataset.

6) **Validasi Eksternal**

Kadang-kadang, verifikasi hasil juga memerlukan validasi eksternal untuk memastikan bahwa data yang telah dibersihkan sesuai dengan sumber data eksternal atau standar yang telah ditetapkan.

Perbandingan dengan Sumber Eksternal

Memeriksa data yang telah dibersihkan dengan sumber data eksternal untuk memastikan konsistensi dan akurasi.

Contoh 4.24:

Membandingkan data penjualan dengan laporan keuangan resmi perusahaan.

Benchmarking

Menggunakan standar atau *benchmark* eksternal untuk memvalidasi data.

Contoh 4.25:

Memastikan bahwa data kinerja karyawan sesuai dengan standar industri yang berlaku.

D. Alat dan Teknik Data Cleaning

Proses pembersihan data membutuhkan penggunaan berbagai alat dan teknik untuk mengidentifikasi serta memperbaiki masalah dalam data. Alat-alat tersebut dapat membantu dalam mengotomatisasi sejumlah tugas yang memakan waktu, memastikan konsistensi, dan meningkatkan akurasi hasil pembersihan data.

Terdapat beberapa alat dan teknik yang dapat digunakan untuk data cleaning, di antarnya :

1. Alat Data Cleaning

a. Python

Python adalah bahasa pemrograman yang populer dan kuat yang digunakan secara luas dalam analisis data dan ilmu data. Python memiliki beberapa pustaka yang sangat berguna untuk analisis data, seperti Pandas, NumPy, dan SciPy.

Pandas adalah pustaka yang digunakan untuk manipulasi dan analisis data tabular. Fungsi-fungsi seperti *dropna(), fillna(), drop_duplicates(),* dan *replace()* sangat berguna dalam penanganan nilai yang hilang, duplikasi, dan kesalahan format.

Contoh 4.26:

```
import pandas as pd

data = pd.read_csv('data.csv')

data.dropna(inplace=True)

data['column'] = data['column'].replace('old_value',
'new_value')
```

NumPy adalah pustaka yang digunakan untuk komputasi numerik dan operasi matriks.

Contoh 4.27:

```
import numpy as np
data['column']=(data['column'] - np.mean(data['column']))
/ np.std(data['column'])
```

SciPy adalah pustaka yang digunakan untuk komputasi ilmiah dan analisis statistik.

Contoh 4.28 :

```
from scipy import stats
z_scores = stats.zscore(data['column'])
data = data[(z_scores < 3)]
```

b. Database Management System

Alat yang kuat untuk data cleaning yang lebih kompleks, misalnya SQL. SQL adalah bahasa standar untuk manajemen *database relasional*.

Contoh 4.29:

Menghapus duplikasi

```
DELETE FROM table
WHERE id NOT IN (
  SELECT MIN(id)
  FROM table
  GROUP BY column1, column2, ...
);
```

Mengisi Nilai yang hilang

Contoh 4.30:

```
UPDATE table
SET column = COALESCE(column, 'default_value');
```

c. **Dedicated Data Cleaning Tools**

Alat khusus yang dirancang untuk tugas data *cleaning* yang lebih spesifik dan kompleks, misalnya *OpenRefine* dan Trifacta. *OpenRefine* adalah alat *open-source* yang berguna untuk membersihkan dan mengubah data, serta sangat efektif dalam mengidentifikasi serta memperbaiki ketidakkonsistenan. Trifacta, di sisi lain, merupakan alat komersial yang digunakan untuk pembersihan dan transformasi data dengan kemampuan pembelajaran mesin untuk merekomendasikan transformasi data.

d. **Spreadsheet Software**

Alat yang sederhana dan mudah digunakan untuk data *clearning* dasar, misalnya *Ms. Excel* dan *Google Sheets.*

2. **Teknik Data Cleaning**

a. **Cara Mengatasi Nilai yang Hilang**

Salah satu cara untuk mengatasi nilai yang hilang adalah dengan menggunakan imputasi mean, imputasi median, atau metode prediktif.

Imputasi Mean

Menggantikan nilai yang hilang dengan nilai rata-rata kolom.

Contoh 4.31:

```
data['column'].fillna(data['column'].mean(),
inplace=True)
```

K-Nearest Neighbors (KKN) Imputation

Menggunakan nilai dari k tetangga terdekat untuk mengisi nilai yang hilang.

Contoh 4.32:

```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=5)

data = imputer.fit_transform(data)
```

b. Normalisasi dan Standarisasi

Teknik untuk memastikan data berada dalam skala atau format yang konsisten.

Min-Max Normalization

Mengubah data ke dalam rentang [0,1]

Contoh 4.33:

```
data['column'] = (data['column'] -
data['column'].min()) / (data['column'].max()
- data['column'].min())
```

Score Standardization

Mengubah data sehingga memiliki mean 0 dan standar deviasi 1.

Contoh 4.34:

```
data['column'] = (data['column'] -
data['column'].mean()) / data['column'].std()
```

c. Penanganan Outlier

Teknik untuk mengidentifikasi dan mengatasi outlier dalam data.

IQR Method

Menggunakan interquartile range untuk mendeteksi dan menghapus outlier.

Contoh 4.35:

```
Q1 = data['column'].quantile(0.25)

Q3 = data['column'].quantile(0.75)

IQR = Q3 - Q1

data = data[(data['column'] >= (Q1 - 1.5 * IQR)) &
(data['column'] <= (Q3 + 1.5 * IQR))]
```

Winsorizing

Mengganti outlier dengan nilai persentil tertentu

Contoh 4.36:

```
from scipy.stats.mstats import winsorize

data['column'] = winsorize(data['column'],
limits=[0.05, 0.05])
```

d. Deteksi dan Penghapusan Duplikasi

Teknik untuk mengidentifikasi dan menghapus data yang terduplikasi.

Pemeriksaan Duplikasi

Menggunakan fungsi untuk mengidentifikasi dan menghapus barisan yang terduplikasi.

Contoh 4.37 :

```
data.drop_duplicates(inplace=True)
```

Penggabungan Data

Menggabungkan data dari berbagai sumber dengan hati-hati untuk menghindari duplikasi.

Contoh 4.38:

```
merged_data = pd.merge(data1, data2,
on='key_column')
```

DAFTAR PUSTAKA

Baker, L. 2021. *Data Cleaning: The Ultimate Practical Guide*.

Han, J. & M. Kamber. 2006. *Data mining* : *Concept and Techniques Second Edition*. San Fransisco: Morgan Kaufmann Publishers.

Kazil, J., & Jarmul, K. 2016. *Data Wrangling with Python: Tips and Tools to Make Your Life Easier*. O'Reilly Media.

Kusrini & Emha, T. L. 2009. *Algoritma Data Mining*. Yogyakarta: Andi Offset.

McKinney, W. 2017. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

Prastyadi, W.R., dkk 2024. *Buku Ajar Data Maining*: PT. Sonpedia Publishing Indonesia.

Towards Data Science. 2019. *Comprehensive Guide to Data Cleaning in Python*. Retrieved from Towards Data Science.

Wickham, H., & Grolemund, G. 2016. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media.

Walker, M. 2020. *Python Data Cleaning Cookbook*. Packt Publishing.

BAB 5 DATA INTEGRATION

Halomoan Edy Manurung, S,Si., M.Cs.

A. Pengertian Integrasi Data

Integrasi data adalah proses penggabungan data dari berbagai sumber yang berbeda untuk menyediakan pandangan terpadu dan konsisten. Dalam era digital ini, banyak organisasi bergantung pada data untuk membuat keputusan strategis yang didasarkan pada informasi yang akurat dan relevan. Oleh karena itu, integrasi data menjadi sangat penting untuk memastikan bahwa semua informasi yang relevan dapat diakses dan dianalisis dengan mudah.

Sumber data bisa berasal dari berbagai sistem seperti basis data relasional, file flat, layanan web, sensor Internet of Things (IoT), aplikasi SaaS (Software as a Service), dan lain-lain. Setiap sumber data ini sering kali memiliki format, struktur, dan skema yang berbeda, yang bisa menjadi tantangan besar dalam proses integrasi. Tanpa integrasi data yang efektif, organisasi dapat mengalami kesulitan dalam menggabungkan informasi dari berbagai departemen atau unit bisnis, yang pada akhirnya dapat menghambat kemampuan mereka untuk mendapatkan wawasan yang komprehensif dan akurat.

Proses integrasi data melibatkan tiga langkah utama: ekstraksi, transformasi, dan pemuatan (ETL). Ekstraksi adalah proses pengambilan data dari berbagai sumber. Ini mungkin melibatkan pengambilan data dari basis data operasional, file log, atau sistem lainnya. Transformasi adalah proses mengubah data yang diekstraksi ke dalam format yang sesuai untuk

analisis atau untuk digabungkan dengan data dari sumber lain. Ini mungkin melibatkan pembersihan data, penggabungan data, atau agregasi data. Pemuatan adalah proses memasukkan data yang telah diubah ke dalam sistem target seperti gudang data atau basis data terpadu.

Kualitas data adalah aspek penting dalam integrasi data. Data yang diintegrasikan harus akurat, lengkap, konsisten, dan up-to-date. Proses pembersihan data sering kali diperlukan untuk memastikan kualitas data. Data yang buruk dapat menyebabkan kesalahan dalam analisis dan pengambilan keputusan, yang pada akhirnya dapat merugikan organisasi. Oleh karena itu, memastikan bahwa data yang diintegrasikan berkualitas tinggi adalah prioritas utama dalam setiap proyek integrasi data.

Manajemen metadata adalah elemen kunci lainnya dalam integrasi data. Metadata adalah data tentang data. Ini mencakup informasi seperti definisi data, asal data, dan transformasi yang telah dilakukan pada data. Metadata membantu dalam pengelolaan, pemahaman, dan penggunaan data yang diintegrasikan. Dengan manajemen metadata yang efektif, organisasi dapat memastikan bahwa data yang diintegrasikan dapat diakses dan digunakan dengan mudah oleh semua pemangku kepentingan yang relevan.

Meskipun integrasi data memiliki banyak manfaat, proses ini juga dihadapkan dengan berbagai tantangan. Salah satu tantangan utama adalah heterogenitas sumber data. Sumber data yang berbeda mungkin memiliki format dan struktur yang berbeda, sehingga menyulitkan proses integrasi. Selain itu, volume data yang besar dalam era big data dapat menjadi tantangan dalam hal penyimpanan, pemrosesan, dan integrasi. Tantangan lainnya termasuk kualitas data yang mungkin tidak konsisten atau tidak lengkap, serta masalah keamanan dan privasi data yang harus diatasi untuk melindungi integritas dan kerahasiaan data.

Untuk mengatasi tantangan-tantangan ini, organisasi perlu mengikuti praktik terbaik dalam integrasi data. Ini termasuk perencanaan yang matang, penggunaan alat dan teknologi yang tepat, keterlibatan semua pemangku kepentingan, fokus pada kualitas data, dan monitoring serta pemeliharaan yang berkelanjutan. Dengan pendekatan yang tepat, integrasi data dapat menjadi aset yang sangat berharga bagi organisasi, membantu mereka untuk memanfaatkan sepenuhnya informasi yang mereka miliki dan membuat keputusan yang lebih baik dan lebih cepat.

B. Konsep Dasar Integrasi Data

Integrasi data merupakan proses yang esensial dalam mengoptimalkan operasional dan mendukung analisis yang lebih mendalam di berbagai sektor bisnis. Untuk memahami sepenuhnya kompleksitas dan pentingnya integrasi data, berikut adalah beberapa konsep dasar yang perlu diperhatikan:

1. **Sumber Data**: Sumber data bisa berasal dari berbagai sistem seperti database relasional, file flat, layanan web, sensor IoT, aplikasi bisnis, platform media sosial, dan layanan cloud. Setiap sumber data memiliki format, struktur, dan skema yang berbeda, mulai dari tabel dalam database SQL, file CSV, data JSON dari API, hingga aliran data real-time dari perangkat IoT. Variasi ini menuntut pendekatan yang fleksibel dan adaptif dalam proses integrasi data, agar dapat menggabungkan data dengan mulus dari berbagai sumber yang heterogen.
2. **ETL (Extract, Transform, Load)**: ETL adalah proses inti dalam integrasi data yang terdiri dari tiga tahap utama:
   a. **Extract (Ekstraksi)**: Tahap ini melibatkan pengambilan data dari berbagai sumber. Tantangan utama dalam tahap ekstraksi adalah menangani berbagai format data dan memastikan data yang diekstraksi relevan dan lengkap.
   b. **Transform (Transformasi)**: Pada tahap ini, data yang telah diekstraksi diubah ke dalam format yang sesuai untuk analisis atau penggunaan lebih lanjut. Transformasi

meliputi proses pembersihan data, normalisasi, agregasi, dan penggabungan data dari berbagai sumber. Transformasi yang efektif memastikan data konsisten, berkualitas tinggi, dan siap untuk dianalisis.

c. **Load (Memuat)**: Tahap terakhir ini melibatkan pemuatan data yang telah diubah ke dalam sistem target seperti data warehouse atau database terpadu. Proses ini harus dilakukan secara efisien untuk menghindari latensi dan memastikan data siap untuk diakses dan dianalisis dalam waktu nyata.

3. **Data Warehousing**: Data warehouse adalah solusi penyimpanan data terpadu yang memungkinkan konsolidasi data dari berbagai sumber untuk keperluan analisis dan pelaporan. Data warehouse dirancang untuk menyimpan data dalam skema yang dioptimalkan untuk query analitik, seperti skema bintang atau salju. Dengan menggunakan data warehouse, organisasi dapat menjalankan analisis yang kompleks dan menghasilkan laporan yang mendalam, sehingga mendukung pengambilan keputusan strategis yang lebih baik.

   Selain itu, arsitektur data warehouse sering kali mencakup data mart, yang merupakan subset dari data warehouse yang difokuskan pada area bisnis tertentu.

4. **Kualitas Data (Data Quality)**: Kualitas data merupakan aspek kritis dalam integrasi data. Data yang diintegrasikan harus memenuhi kriteria akurasi, kelengkapan, konsistensi, dan aktualitas. Untuk mencapai kualitas data yang tinggi, perlu dilakukan proses pembersihan data yang mencakup identifikasi dan koreksi kesalahan, penghapusan duplikasi, dan validasi data terhadap aturan bisnis yang relevan. Data berkualitas tinggi memastikan bahwa analisis dan pelaporan yang dihasilkan dapat diandalkan dan akurat, sehingga mendukung pengambilan keputusan yang lebih baik.

5. **Manajemen Metadata (Metadata Management)**: Metadata adalah informasi yang menjelaskan tentang data, termasuk definisi, asal, dan transformasi yang telah dilakukan.

Manajemen metadata yang efektif membantu dalam pengelolaan, pemahaman, dan penggunaan data yang diintegrasikan. Metadata mencakup berbagai aspek seperti struktur data, tipe data, hubungan antar data, dan aturan bisnis yang diterapkan. Dengan metadata yang lengkap dan terorganisir, pengguna dapat dengan mudah menavigasi, memahami, dan memanfaatkan data yang ada, serta memastikan bahwa data digunakan dengan cara yang konsisten dan sesuai dengan kebijakan organisasi.

C. Inovasi dalam Integrasi Data

Integrasi data terus berkembang seiring dengan kemajuan teknologi dan perubahan kebutuhan bisnis. Beberapa inovasi terkini dalam integrasi data meliputi:

1. **Data Virtualization**: Teknologi ini memungkinkan akses ke data dari berbagai sumber tanpa perlu memindahkannya ke lokasi pusat. Data tetap berada di sumber aslinya, tetapi dapat diakses dan dianalisis secara terpadu. Data virtualisasi mengurangi kebutuhan akan replikasi data dan dapat menghemat biaya serta waktu.
2. **Machine Learning for Data Transformation**: Penggunaan algoritma machine learning untuk otomatisasi proses transformasi data. Teknologi ini dapat mengenali pola dalam data, membersihkan, dan mengubah data secara otomatis dengan akurasi yang tinggi.
3. **Real-time Data Integration**: Kemampuan untuk mengintegrasikan data secara real-time memungkinkan organisasi untuk mendapatkan wawasan terkini dan merespons perubahan dengan cepat. Ini sangat penting dalam lingkungan bisnis yang dinamis dan kompetitif.
4. **Cloud-based Integration Platforms**: Platform integrasi berbasis cloud menawarkan skalabilitas, fleksibilitas, dan biaya yang lebih rendah. Platform ini dapat menangani volume data yang besar dan memungkinkan integrasi data lintas batas geografis dengan mudah.

Dengan memahami dan menerapkan konsep dasar serta inovasi terkini dalam integrasi data, organisasi dapat meningkatkan efisiensi operasional, mendukung analisis yang lebih mendalam, dan pada akhirnya, mencapai keunggulan kompetitif dalam pasar.

D. Metode Integrasi Data

Integrasi data adalah proses kritis yang memungkinkan organisasi untuk menggabungkan data dari berbagai sumber menjadi satu kesatuan yang terpadu. Metode integrasi data yang digunakan dapat bervariasi tergantung pada kebutuhan spesifik organisasi, infrastruktur teknologi yang ada, dan jenis data yang dikelola. Berikut adalah beberapa metode yang umum digunakan dalam integrasi data, disertai dengan penjelasan yang lebih mendalam dan inovatif, yaitu:

1. Integrasi Manual: Integrasi manual melibatkan individu atau tim yang secara langsung menangani penggabungan data dari berbagai sumber. Meskipun metode ini bisa digunakan dalam situasi tertentu, seperti proyek kecil atau data dengan struktur sederhana, integrasi manual sering kali memakan waktu dan rentan terhadap kesalahan manusia. Kesalahan dalam penginputan atau pengolahan data dapat menyebabkan inkonsistensi dan mengurangi keakuratan data yang diintegrasikan. Untuk mengurangi risiko ini, organisasi dapat menggunakan alat bantu seperti skrip otomatisasi sederhana atau spreadsheet dengan fungsi makro, namun ini tetap memerlukan pengawasan yang ketat.
2. Integrasi Berbasis Middleware: Middleware adalah perangkat lunak perantara yang memungkinkan berbagai sistem untuk saling berkomunikasi dan bertukar data. Middleware bertindak sebagai jembatan antara berbagai sumber data, memungkinkan pertukaran informasi yang efisien dan konsisten. Middleware modern sering kali dilengkapi dengan fitur-fitur canggih seperti orkestrasi proses bisnis, manajemen pesan, dan pemetaan data otomatis. Beberapa contoh middleware yang umum

digunakan termasuk Enterprise Service Bus (ESB) dan Message-Oriented Middleware (MOM). Dengan middleware, organisasi dapat mengurangi ketergantungan pada integrasi manual dan meningkatkan skalabilitas serta fleksibilitas sistem mereka.

3. Integrasi Berbasis Aplikasi: Aplikasi khusus untuk integrasi data dirancang untuk mengautomatisasi proses penggabungan data dari berbagai sumber. Aplikasi ini biasanya dilengkapi dengan fitur ETL (Extract, Transform, Load) yang kuat, yang memungkinkan ekstraksi data dari berbagai sistem, transformasi data ke dalam format yang diinginkan, dan pemuatan data ke dalam sistem target. Alat integrasi berbasis aplikasi sering kali memiliki antarmuka pengguna yang intuitif, alat pemantauan, dan kemampuan untuk menangani volume data yang besar. Beberapa contoh alat integrasi berbasis aplikasi termasuk Informatica, Talend, dan Microsoft SQL Server Integration Services (SSIS). Aplikasi ini membantu mengurangi kesalahan manual dan meningkatkan efisiensi proses integrasi data.
4. Integrasi Berbasis Virtualisasi: Data virtualisasi adalah metode yang memungkinkan akses ke data dari berbagai sumber tanpa perlu memindahkannya ke lokasi pusat. Dengan data virtualisasi, data tetap berada di sumber aslinya, tetapi dapat diakses dan dianalisis secara terpadu seolah-olah berada dalam satu lokasi fisik. Teknologi ini mengabstraksi detail teknis dari lokasi dan format data, memberikan pandangan tunggal yang terpadu kepada pengguna. Data virtualisasi memungkinkan organisasi untuk mengurangi biaya penyimpanan dan meningkatkan kecepatan akses data, karena tidak ada proses duplikasi data yang diperlukan. Contoh platform data virtualisasi termasuk Denodo dan IBM InfoSphere Virtualization. Teknologi ini sangat berguna dalam lingkungan big data di mana data tersebar di berbagai platform dan lokasi geografis.

5. Integrasi Berbasis Layanan Web: Menggunakan layanan web dan Application Programming Interfaces (API) adalah metode yang semakin populer untuk integrasi data. Layanan web memungkinkan sistem yang berbeda untuk berkomunikasi dan bertukar data melalui protokol standar seperti HTTP dan format data seperti XML atau JSON. API menyediakan antarmuka yang terdefinisi dengan baik untuk mengakses fungsi dan data dari aplikasi lain. Integrasi berbasis layanan web memungkinkan organisasi untuk menghubungkan sistem internal dengan aplikasi pihak ketiga, platform cloud, dan layanan SaaS dengan cara yang fleksibel dan scalable. Beberapa contoh layanan web yang umum digunakan termasuk RESTful APIs dan SOAP APIs. Penggunaan API memungkinkan integrasi yang lebih dinamis dan memungkinkan pengembangan aplikasi yang lebih cepat dan responsif terhadap perubahan kebutuhan bisnis.

E. Inovasi dalam Metode Integrasi Data

Seiring dengan kemajuan teknologi, metode integrasi data terus berkembang dan menghadirkan inovasi-inovasi baru yang dapat meningkatkan efisiensi dan efektivitas proses integrasi. Beberapa inovasi terkini dalam metode integrasi data meliputi:

1. **Integration Platform as a Service (iPaaS)**: iPaaS adalah platform cloud yang menyediakan alat dan layanan untuk mengelola integrasi data secara komprehensif. iPaaS memungkinkan organisasi untuk mengintegrasikan data dari berbagai sumber dengan mudah, tanpa perlu infrastruktur fisik yang besar. Platform ini menawarkan fitur-fitur seperti orkestrasi alur kerja, pemantauan real-time, dan manajemen API, yang semuanya dapat diakses melalui antarmuka berbasis web.
2. **AI and Machine Learning Integration**: Integrasi data berbasis kecerdasan buatan (AI) dan pembelajaran mesin (ML) memungkinkan otomatisasi proses integrasi yang lebih cerdas. Algoritma AI/ML dapat digunakan untuk mengenali

pola dalam data, membersihkan data, dan mengoptimalkan transformasi data secara otomatis. Ini tidak hanya mengurangi waktu dan usaha yang diperlukan untuk integrasi data, tetapi juga meningkatkan akurasi dan konsistensi data.

3. **Blockchain for Data Integration**: Teknologi blockchain menawarkan cara baru untuk mengelola integrasi data dengan keamanan yang lebih tinggi dan transparansi. Blockchain dapat digunakan untuk mencatat transaksi data secara aman dan tidak dapat diubah, memastikan integritas data dan memberikan jejak audit yang jelas. Ini sangat berguna dalam industri yang membutuhkan keamanan data tingkat tinggi, seperti keuangan dan kesehatan.

Dengan memahami dan mengimplementasikan metode integrasi data yang tepat, serta memanfaatkan inovasi terbaru, organisasi dapat meningkatkan kualitas data, efisiensi operasional, dan kemampuan analitik mereka. Ini pada akhirnya akan mendukung pengambilan keputusan yang lebih baik dan membantu organisasi mencapai keunggulan kompetitif di pasar.

F. Tantangan dalam Integrasi Data

Meskipun integrasi data menawarkan banyak manfaat, seperti meningkatkan efisiensi operasional dan mendukung analisis yang lebih mendalam, ada berbagai tantangan yang harus diatasi untuk memastikan keberhasilan proses ini. Tantangan-tantangan ini mencakup aspek teknis, kualitas data, keamanan, dan skalabilitas yang dapat dijelaskan sebagai berikut:

1. **Heterogenitas Sumber Data**: Sumber data yang berbeda memiliki format, struktur, dan skema yang bervariasi, yang membuat proses integrasi menjadi kompleks. Data dapat berasal dari berbagai sistem seperti database relasional, file flat, aplikasi SaaS, sensor IoT, dan platform media sosial. Setiap jenis sumber data ini menggunakan skema dan standar yang berbeda, yang sering kali tidak kompatibel satu sama lain. Untuk mengatasi tantangan ini, diperlukan pendekatan

yang fleksibel dan alat yang mampu menangani berbagai jenis data dan format secara efisien. Penggunaan teknologi seperti data virtualization dan middleware dapat membantu mengatasi heterogenitas ini dengan menyediakan lapisan abstraksi yang memungkinkan integrasi yang lebih mulus.

2. **Volume Data yang Besar**: Dalam era big data, volume data yang besar menjadi tantangan signifikan dalam hal penyimpanan, pemrosesan, dan integrasi. Data yang dihasilkan dari berbagai sumber dan dalam jumlah yang sangat besar memerlukan infrastruktur yang mampu menangani beban tersebut. Penyimpanan data yang besar memerlukan solusi skalabilitas tinggi seperti penyimpanan cloud atau sistem distribusi data. Selain itu, pemrosesan data dalam volume besar membutuhkan kemampuan komputasi yang kuat dan algoritma yang efisien. Teknologi seperti Hadoop dan Spark telah dikembangkan untuk mengatasi tantangan ini dengan menawarkan platform pemrosesan data terdistribusi yang mampu menangani volume data yang sangat besar dengan efisien.
3. **Kualitas Data**: Data dari berbagai sumber sering kali tidak konsisten, tidak lengkap, atau tidak akurat, yang dapat menghambat proses integrasi dan analisis. Kualitas data yang buruk dapat menyebabkan kesalahan dalam analisis dan pengambilan keputusan. Oleh karena itu, proses pembersihan dan validasi data yang ekstensif sangat penting. Teknologi modern menggunakan machine learning untuk otomatisasi pembersihan data, seperti mengidentifikasi dan mengoreksi anomali, menghapus duplikasi, dan mengisi data yang hilang. Alat-alat ini mampu belajar dari data dan meningkatkan kualitas data secara terus-menerus, sehingga memastikan data yang diintegrasikan berkualitas tinggi dan dapat diandalkan.
4. **Keamanan dan Privasi Data**: Integrasi data sering kali melibatkan akses dan pertukaran data yang sensitif, yang memerlukan langkah-langkah keamanan yang ketat untuk melindungi privasi dan integritas data. Ancaman keamanan

dapat berasal dari berbagai sumber, termasuk serangan siber, akses tidak sah, dan kebocoran data. Untuk mengatasi tantangan ini, organisasi harus menerapkan kebijakan keamanan data yang komprehensif, termasuk enkripsi data, autentikasi yang kuat, kontrol akses yang ketat, dan audit keamanan secara berkala. Teknologi blockchain juga dapat digunakan untuk meningkatkan keamanan dengan menyediakan catatan transaksi data yang tidak dapat diubah dan transparan.

5. **Skalabilitas**: Sistem integrasi data harus mampu menangani pertumbuhan data dan permintaan pengguna yang meningkat tanpa mengalami penurunan kinerja. Skalabilitas mencakup kemampuan untuk memperluas kapasitas penyimpanan, komputasi, dan jaringan secara efisien seiring dengan pertumbuhan volume data dan kebutuhan analisis. Solusi cloud computing, seperti Amazon Web Services (AWS), Google Cloud Platform (GCP), dan Microsoft Azure, menawarkan infrastruktur yang sangat skalabel dan fleksibel yang dapat disesuaikan dengan kebutuhan organisasi. Selain itu, arsitektur microservices dan containerization dengan alat seperti Kubernetes dapat membantu dalam menciptakan lingkungan yang skalabel dan mudah dikelola.

G. Inovasi dalam Mengatasi Tantangan Integrasi Data

Untuk menghadapi tantangan-tantangan tersebut, berbagai inovasi telah dikembangkan dalam bidang integrasi data. Berikut adalah beberapa pendekatan inovatif yang dapat membantu organisasi mengatasi tantangan dalam integrasi data:

1. **Artificial Intelligence and Machine Learning**: AI dan ML digunakan untuk meningkatkan otomatisasi dalam proses integrasi data, termasuk dalam pembersihan data, deteksi anomali, dan transformasi data. Algoritma AI/ML dapat mempelajari pola data dan memperbaiki kualitas data secara otomatis, sehingga mengurangi beban kerja manual dan meningkatkan efisiensi.

2. **Data Fabric**: Data fabric adalah arsitektur dan layanan yang menyediakan integrasi data secara terus-menerus dan real-time di seluruh lingkungan hybrid dan multicloud. Ini memungkinkan organisasi untuk mengakses, mengelola, dan mengintegrasikan data dari berbagai sumber dengan cara yang lebih terpadu dan efisien.
3. **Edge Computing**: Dengan meningkatnya penggunaan perangkat IoT, edge computing menjadi penting untuk mengurangi latensi dan meningkatkan kecepatan pemrosesan data. Edge computing memungkinkan pemrosesan data dilakukan dekat dengan sumber data, sehingga mengurangi beban pada jaringan dan pusat data utama.
4. **Serverless Computing**: Serverless computing memungkinkan organisasi untuk menjalankan fungsi dan layanan tanpa perlu mengelola infrastruktur server. Ini memberikan fleksibilitas yang tinggi dan biaya yang lebih rendah, serta mampu menangani beban kerja yang dinamis dan skala besar.

Dengan memanfaatkan inovasi-inovasi ini, organisasi dapat mengatasi tantangan integrasi data dengan lebih efektif, meningkatkan kualitas data, efisiensi operasional, dan kemampuan analitik mereka, serta mendukung pengambilan keputusan yang lebih baik dan lebih cepat.

H. Inovasi dalam Best Practices Integrasi Data

Untuk lebih meningkatkan efektivitas dan efisiensi integrasi data, beberapa inovasi terbaru dapat diterapkan dalam best practices:

1. **Automated Data Mapping and Transformation**: Penggunaan teknologi AI untuk otomatisasi pemetaan data dan transformasi dapat mengurangi waktu dan usaha yang diperlukan dalam proses integrasi. AI dapat mengenali pola dalam data, mengusulkan pemetaan yang optimal, dan melakukan transformasi data secara otomatis dengan tingkat akurasi yang tinggi.

2. **Self-Service Data Integration**: Memberikan alat integrasi data self-service kepada pengguna bisnis dapat mempercepat proses integrasi dan mengurangi ketergantungan pada tim IT. Alat ini memungkinkan pengguna untuk melakukan ekstraksi, transformasi, dan pemuatan data secara mandiri dengan antarmuka pengguna yang intuitif.
3. **Real-Time Data Integration**: Integrasi data real-time memungkinkan organisasi untuk mendapatkan wawasan terkini dan merespons perubahan dengan cepat. Menggunakan teknologi streaming data dan event-driven architecture, organisasi dapat mengintegrasikan dan menganalisis data seketika saat data tersebut dibuat atau diperbarui.
4. **Data Governance Frameworks**: Implementasi kerangka kerja tata kelola data yang kuat dapat membantu dalam mengelola kualitas, keamanan, dan kepatuhan data secara efektif. Kerangka kerja ini mencakup kebijakan, prosedur, dan alat untuk mengelola seluruh siklus hidup data, dari pengumpulan hingga penghapusan.

Dengan menerapkan best practices ini dan memanfaatkan inovasi terbaru, organisasi dapat mengatasi tantangan integrasi data dengan lebih efektif, meningkatkan kualitas data, dan memaksimalkan nilai yang dapat diperoleh dari data yang diintegrasikan. Pendekatan yang strategis dan teknologi yang tepat akan memastikan bahwa proses integrasi data berjalan lancar, efisien, dan memberikan manfaat maksimal bagi organisasi.

I. Rangkuman

Integrasi data adalah fondasi vital dalam pengelolaan informasi di era digital, memungkinkan organisasi untuk menggabungkan data dari berbagai sumber menjadi satu pandangan terpadu. Proses ini menghadapi tantangan seperti heterogenitas sumber data, volume data yang besar, kualitas data, keamanan, dan skalabilitas. Namun, dengan perencanaan yang matang, penggunaan teknologi canggih seperti platform

data integration as a service (iPaaS), big data processing frameworks, alat ETL, dan cloud computing, serta penerapan praktik terbaik, organisasi dapat mengatasi tantangan ini dan meraih manfaat maksimal dari data mereka. Inovasi seperti data virtualization, machine learning untuk transformasi data, integrasi data real-time, dan platform berbasis cloud semakin memperkuat kemampuan integrasi data.

DAFTAR PUSTAKA

B. Young, "Scalability in Data Integration: Techniques and Approaches," Scalable Computing Review, vol. 10, no. 2, pp. 150-165, 2021.

F. Thompson, "Big Data Volume Management: Strategies and Solutions," Big Data Journal, vol. 4, no. 2, pp. 90-105, 2021.

H. E. Manurung, F. Y. Wattimena, R. Koibur, A. S. Renyaan, and M. E. Koibur, "Inovasi Digital dalam Pemerintahan.pdf," M. K. Cecep Kurnia Sastradipraja, S.Kom., Ed. Bandung: Kaizen Media Publishing, 2024, p. 100.

Hartatik Mayko Edison Koibur Aris Wahyu Murdiyanto Zen Munawar Gina Purnama Insany Halomoan Edy Manurung. *et al*., "Sains data: Strategi, Teknik, dan Model Analisis Data," Bandung: Kaizen Media Publishing, 2022, p. 208.

J. Doe, "Data Integration: Concept and Approaches," Journal of Data Management, vol. 12, no. 2, pp. 45-60, 2020.

K. Johnson, "Data Virtualization: Approaches and Benefits," Journal of Modern Data Management, vol. 14, no. 1, pp. 12-27, 2021.

M. Dr. Uky Yudatama, S.Si., M.Kom., M.M. Nur Syamsiyah, ST., MTI. Dr. Irmawati, S.Kom. *et al*., "Memahami Teknologi Informasi Prinsip, Pengembangan, dan Penerapan,". Bandung: Kaizen Media Publishing, 2023, p. 305.

M. M. Dr. Uky Yudatama, S.Si., M.Kom. *et al*., Sistem Enterprise di Era Digital Inovasi, Transformasi, dan Keberlanjutan. Bandung: Kaizen Media Publishing, 2022.

R. White, "Real-time Data Integration: Challenges and Solutions," Computational Data Analysis, vol. 7, no. 3, pp. 200-215, 2019.

T. Lee, "Machine Learning for Data Transformation: Techniques and Practices," Artificial Intelligence Journal, vol. 9, no. 2, pp. 123-139, 2020.

BAB 6 | DATA SELECTION

Mayko Edison Koibur, S.T., M.Eng.

A. Definisi Data Selection

Dalam data mining, pemilihan data adalah langkah penting yang melibatkan beberapa tahap, mulai dari identifikasi sumber data, ekstraksi data, dan pemilihan data berdasarkan kriteria tertentu yang sesuai dengan tujuan analisis.

Gambar 6. 1. Data Selection

Pada tahap awal, identifikasi sumber data dilakukan untuk menentukan dari mana data diambil. Sumber data dapat berasal dari berbagai sumber, seperti data warehouse, database relasional, atau bahkan dari sumber eksternal, seperti situs web

dan media sosial. Pembersihan data menghapus atau memperbaiki data yang tidak lengkap, duplikat, atau tidak relevan. Ini memastikan bahwa data yang dibersihkan disimpan dengan aman.

Setelah data dibersihkan, tahap berikutnya adalah reduksi data. Tujuan dari reduksi data adalah untuk mengurangi jumlah data yang harus dianalisis. Dengan melakukan reduksi data, kita dapat mendapatkan subset data yang lebih kecil namun tetap representatif, yang mempermudah dan mempercepat proses analisis. Langkah terakhir dalam pemilihan data adalah integrasi data, yang menggabungkan data dari berbagai sumber untuk membuat dataset yang lebih lengkap. Ini penting untuk memastikan bahwa semua informasi yang relevan tersedia untuk analisis.

Dengan menggunakan Data Selection, analis dapat memastikan bahwa data yang digunakan dalam proses Data Mining adalah data yang paling relevan dan berkualitas tinggi. Ini sangat penting karena data yang bersih, terstruktur, dan relevan akan meningkatkan efisiensi dan akurasi proses Data Mining. Selain itu, pemilihan data juga membantu mengurangi volume dan kompleksitas data yang harus diproses, yang berarti pemrosesan lebih cepat dan penggunaan sumbe lebih sedikit (Carudin *et al.*, 2024). Secara keseluruhan, pemilihan data adalah langkah penting dalam pengolahan data karena memastikan bahwa hanya data yang paling relevan dan berkualitas tinggi yang digunakan, sehingga hasil analisis dapat diandalkan dan memberikan wawasan yang bermanfaat.

1. Pentingnya Pemilihan Data dalam Data Mining

Salah satu langkah yang paling penting dalam proses data mining adalah pemilihan data. Kualitas hasil analisis dan keputusan yang dibuat berdasarkan data sangat dipengaruhi oleh pemilihan data yang tepat dan relevan.

Pertama, pemilihan data meningkatkan kualitas analisis. Dengan memilih data yang relevan dan berkualitas tinggi, hasil analisis menjadi lebih akurat dan dapat diandalkan. Selain itu, data yang bersih dan terstruktur

memungkinkan model analisis bekerja lebih efisien dan menghasilkan wawasan yang lebih tepat dan berharga.

Kedua, pemilihan data membantu mengurangi kompleksitas data; set data yang besar dan kompleks dapat menjadi hambatan bagi proses analisis. Dengan memilih subset data yang paling relevan, jumlah data yang harus dianalisis dapat dikurangi. Hal ini tidak hanya mempercepat dan mempermudah proses analisis, tetapi juga mengurangi jumlah sumber daya komputasi yang digunakan dan waktu pemrosesan.

Selain itu, proses pemilihan data memastikan bahwa hanya data yang benar-benar relevan dan sesuai dengan tujuan analisis yang digunakan, sehingga analisis menjadi lebih fokus dan hasilnya lebih bermakna.

Selain itu, pemilihan data menangani masalah data kotor dan tidak lengkap. Proses ini melibatkan pembersihan data, yang membantu mengidentifikasi dan mengatasi masalah seperti data yang hilang, duplikasi, dan inkonsistensi. Oleh karena itu, data yang dipilih untuk dianalisis harus bersih dan siap digunakan, sehingga hasilnya lebih akurat.

Salah satu manfaat lain dari pemilihan data adalah optimalisasi sumber daya, yang membantu dalam optimalisasi penggunaan sumber daya komputasi dengan mengurangi jumlah data yang perlu dianalisis. Ini sangat penting ketika bekerja dengan data besar, atau Big Data.

Selain itu, pemilihan data meningkatkan kecepatan dan efisiensi proses analisis karena set data yang lebih kecil dan relevan memungkinkan proses analisis dilakukan lebih cepat dan efisien. Ini menghemat waktu dan memungkinkan para analis melakukan lebih banyak iterasi dan eksperimen pada saat yang sama, yang menghasilkan hasil akhir yang lebih baik.

Selain itu, pemilihan data membantu pengambilan keputusan yang lebih baik karena data yang relevan dan berkualitas tinggi menghasilkan wawasan yang lebih akurat,

yang pada gilirannya membantu pengambilan keputusan yang lebih baik. Keputusan yang dibuat berdasarkan data yang dipilih dengan baik lebih mungkin berhasil dan menghasilkan hasil yang baik.

Data Selection membantu menghadapi tantangan yang terkait dengan jumlah data yang sangat besar di era Big Data karena proses ini memungkinkan pengelolaan dan pengurangan data yang berlebihan sehingga fokus dapat diberikan pada data yang benar-benar penting untuk analisis.

Secara keseluruhan, pemilihan data adalah langkah penting dalam data mining yang memastikan bahwa data yang digunakan adalah yang paling relevan dan berkualitas tinggi. Ini tidak hanya meningkatkan kualitas dan akurasi hasil analisis, tetapi juga membantu mengoptimalkan penggunaan sumber daya dan membantu pengambilan keputusan yang lebih baik.

2. Tujuan dan Manfaat Data Selection

Langkah penting dalam proses data mining, pemilihan data memiliki tujuan dan keuntungan yang jelas. Tujuan utama pemilihan data adalah untuk meningkatkan relevansi data yang digunakan dalam analisis; dengan memilih data yang paling relevan, hasil analisis menjadi lebih bermakna dan akurat. Selain itu, pemilihan data bertujuan untuk mengurangi jumlah data yang perlu dianalisis, yang berarti proses analisis menjadi lebih efisien dan pengelolaan sumbuhnya menjadi lebih mudah.

Selain itu, pemilihan data bertujuan untuk menghilangkan data yang tidak berkualitas, seperti data yang kotor, tidak lengkap, atau duplikat, sehingga hasil analisis menjadi lebih andal. Pemilihan data juga membantu memfokuskan analisis pada subset data yang penting, yang memudahkan interpretasi dan pemahaman hasil analisis, dan proses analisis dipercepat dengan menggunakan dataset yang lebih kecil dan lebih bersih.

Data Selection memiliki banyak manfaat. Pertama dan terpenting, peningkatan kualitas hasil analisis adalah keuntungan yang sangat penting. Data yang relevan dan bersih memungkinkan analisis yang lebih akurat dan dapat diandalkan, yang menghasilkan keputusan yang lebih baik. Selain itu, ada keuntungan besar dari efisiensi waktu dan biaya. Biaya operasional juga dapat dikurangi dengan mengurangi waktu yang diperlukan untuk mengolah dan menganalisis data besar.

Salah satu manfaat lain dari pemilihan data adalah penggunaan sumber daya yang lebih efisien. Dengan menghilangkan data yang tidak relevan atau tidak berkualitas, penggunaan sumber daya komputasi dapat dikurangi. Selain itu, pemilihan data meningkatkan fokus dan kejelasan analisis, sehingga analis dapat menghasilkan wawasan yang lebih tajam dan jelas.

Selain itu, pemilihan data meminimalkan risiko kesalahan dalam analisis. Data yang bersih dan terstruktur mengurangi kemungkinan kesalahan, sehingga meningkatkan keandalan hasil. Selain itu, dataset yang lebih kecil dan relevan memungkinkan penyesuaian cepat dalam analisis tanpa membutuhkan banyak waktu dan sumber daya tambahan.

Manfaat lain yang signifikan adalah peningkatan produktivitas. Dengan dataset yang lebih mudah diatur dan memungkinkan lebih banyak iterasi dan eksperimen dalam waktu yang lebih singkat, analis dapat bekerja lebih efisien dan produktif. Terakhir, hasil analisis yang akurat dan relevan membantu pengambilan keputusan yang lebih informatif dan tepat, yang berdampak positif pada strategi bisnis secara keseluruhan.

Data selection bukan hanya tentang memilih data; itu juga tentang memastikan bahwa data yang dipilih adalah yang terbaik untuk mencapai tujuan bisnis dan analisis, sehingga organisasi dapat memaksimalkan potensi data

mereka dan mengoptimalkan proses analisis untuk mencapai hasil yang lebih baik dan keputusan yang lebih strategis.

B. Tahapan dalam Data Selection

1. Identifikasi Sumber Data

Mengidentifikasi sumber data yang akan digunakan adalah langkah pertama dalam proses pemilihan data. Sumber data yang digunakan dapat berbeda-beda tergantung pada kebutuhan analisis dan ketersediaan data, tetapi identifikasi sumber data yang tepat sangat penting untuk memastikan bahwa data yang dipilih relevan, berkualitas tinggi, dan sesuai dengan tujuan analisis.

Berikut ini adalah beberapa sumber data utama yang paling umum :

a. Database Relatif

Database relasional adalah sistem manajemen basis data yang menyimpan data dalam tabel-tabel yang saling berhubungan. Struktur data yang baik memungkinkan akses dan pengolahan data yang efektif. MySQL, PostgreSQL, dan Oracle adalah beberapa database relasional yang paling umum digunakan.

1) MySQL: MySQL adalah sistem manajemen basis data relasional sumber terbuka populer. MySQL sangat terkenal karena kecepatan, keandalan, dan kemudahan penggunaan. MySQL digunakan oleh banyak aplikasi web dan perusahaan besar untuk menyimpan data mereka.
2) PostgreSQL: PostgreSQL adalah database relasional sumber terbuka yang terkenal karena kemampuan canggihnya dan kepatuhannya terhadap standar SQL. Dia mendukung berbagai fitur, seperti ekstensibilitas, transaksi ACID, dan replikasi, yang membuatnya populer di kalangan pengembang dan lembaga yang membutuhkan solusi basis data yang kuat dan dapat diandalkan.

3) Oracle: Oracle adalah salah satu sistem manajemen basis data relasional yang paling populer dan terkenal di dunia karena berbagai fiturnya yang canggih, seperti skala besar, manajemen data yang lengkap, dan dukungan untuk berbagai jenis data dan analisis. Oracle banyak digunakan oleh perusahaan besar dan lembaga yang membutuhkan solusi basis data yang kuat dan dapat diandalkan.

Gambar 6. 2. Data Relatif

b. Data Warehouse

Data warehouse adalah sistem yang digunakan untuk mengumpulkan, menyimpan, dan mengelola data dari berbagai sumber untuk tujuan analisis. Data warehouse dirancang untuk mendukung pelaporan dan analisis bisnis dan menyediakan data yang terstruktur dengan baik dalam skala besar. Amazon Redshift, Google BigQuery, dan Microsoft Azure SQL Data Warehouse adalah beberapa contoh sistem data warehouse yang sangat populer.

1) Amazon Redshift: Layanan data warehouse yang dikelola sepenuhnya oleh AWS, Redshift dapat menangani beban kerja analitis skala besar dan mengelola petabytes data, dan sangat disukai oleh perusahaan yang membutuhkan analisis data yang cepat dan efisien.
2) Google BigQuery: BigQuery memungkinkan pengguna menjalankan query SQL terhadap data yang sangat besar dengan cepat dan merupakan data warehouse analitik tanpa server. BigQuery menarik banyak perusahaan karena kemampuan untuk menangani data besar dan integrasinya yang mudah dengan ekosistem Google Cloud.
3) Microsoft Azure SQL Data Warehouse: Layanan data gudang cloud Microsoft Azure Azure SQL Data Warehouse memiliki skalabilitas tinggi, kemampuan analisis data yang kuat, dan integrasi yang erat dengan ekosistem Azure, yang menjadikannya pilihan yang bagus untuk perusahaan yang menggunakan platform Azure.(Dr. Uky Yudatama, Syamsiyah, *et al*., 2023)

c. Data Eksternal

Data eksternal terdiri dari berbagai sumber data yang berasal dari luar organisasi, seperti layanan data pihak ketiga, media sosial, dan internet. Sumber data ini seringkali tidak terstruktur dan membutuhkan perbaikan dan transformasi sebelum dapat digunakan dalam analisis.

1) Web: Data dapat dikumpulkan melalui proses yang disebut web scraping, yang merupakan pengambilan data secara otomatis dari halaman web. Data ini dapat berupa teks, gambar, atau jenis informasi lainnya yang ditemukan di halaman web. Proses ini biasanya digunakan untuk mengumpulkan data dari situs e-commerce, portal berita, atau forum diskusi.

2) Media Sosial: Media sosial adalah sumber data yang kaya tentang interaksi dan perilaku pengguna. Data dari platform seperti Facebook, Twitter, Instagram, dan LinkedIn dapat mencakup posting, komentar, likes, shares, dan banyak lagi. API yang diberikan oleh platform ini memungkinkan pengembang mengakses data ini untuk analisis.
3) Layanan Data Pihak Ketiga: Banyak bisnis menawarkan layanan data pihak ketiga yang dapat diakses melalui API atau layanan lainnya. Layanan ini mencakup data seperti data demografis, data keuangan, data cuaca, dan banyak lagi. Menggunakan data dari layanan pihak ketiga dapat membantu Anda menganalisis dengan informasi tambahan yang mungkin tidak tersedia di dalam organisasi.

Langkah-langkah selanjutnya dalam pemilihan data mencakup pembersihan, reduksi, dan integrasi data setelah menemukan sumber data yang relevan. Proses ini memastikan bahwa data yang dipilih siap untuk dianalisis dan memberikan informasi bermanfaat. Pembersihan data berarti menghapus atau memperbaiki data yang salah, duplikat, atau tidak relevan. Tujuan dari reduksi data adalah untuk mengurangi jumlah data yang diperlukan untuk analisis melalui metode seperti sampling atau seleksi fitur. Untuk mendapatkan dataset yang lebih lengkap dan menyeluruh, integrasi data menggabungkan data dari berbagai sumber.

Organisasi dapat memastikan bahwa data yang digunakan dalam analisis adalah data yang paling relevan dan berkualitas tinggi dengan mengidentifikasi sumber data yang tepat dan menerapkan langkah-langkah selanjutnya dalam pemilihan data. Dengan demikian, hasil analisis dapat diandalkan dan memberikan wawasan yang bermanfaat.(Dr. Uky Yudatama, Ir. Aris Dianto, *et al.*, 2023).

2. Pembersihan Data

Tahap penting dalam proses pemilihan data adalah pembersihan data, yang memastikan bahwa data yang digunakan untuk analisis berkualitas tinggi bebas dari kesalahan dan siap untuk digunakan. Proses ini mencakup beberapa langkah penting, seperti deteksi dan penanganan nilai yang tidak ada, identifikasi dan penghapusan duplikasi, dan normalisasi dan standarisasi data. Masing-masing langkah tersebut dijelaskan secara cerita di bawah ini :

a. Penemuan dan Penanganan Nilai yang Terlupakan

Langkah pertama dalam pembersihan data adalah menemukan dan mengatasi nilai yang tidak ada. Tidak adanya nilai dapat terjadi karena berbagai alasan, seperti input data yang salah, kegagalan perangkat, atau kurangnya informasi. Nilai-nilai yang hilang biasanya diidentifikasi dengan mencari nilai null, NaN (bukan angka), atau simbol tertentu seperti "NA" atau "-". Setelah nilai yang tidak ada lagi ditemukan, pola dan distribusinya harus dianalisis. Misalnya, apakah kelangkaan nilai hanya terjadi pada observasi atau kolom tertentu?

Setelah identifikasi, langkah berikutnya adalah penanganan nilai yang tidak ada. Salah satu cara untuk menangani nilai yang tidak ada adalah dengan menghapus data yang tidak ada, yang dapat dilakukan pada baris atau kolom yang mengandung nilai yang tidak ada. Metode ini hanya disarankan jika jumlah nilai yang tidak ada sangat kecil dan tidak akan memengaruhi hasil analisis.

Selain penghapusan, nilai yang tidak ada juga dapat diatasi dengan metode imputasi, yang berarti mengisi nilai yang tidak ada dengan nilai lain. Ini dapat mencakup pengisian dengan median, mean, atau mode dari kolom tersebut. Untuk hasil yang lebih akurat, metode yang lebih kompleks, seperti imputasi berbasis model, seperti regresi atau algoritma pembelajaran mesin,

juga dapat digunakan. Dalam beberapa situasi, nilai yang tidak ada dapat dipenuhi dengan kategori khusus atau nilai default yang menunjukkan bahwa data tersebut hilang atau tidak tersedia. Misalnya, dalam data kategorikal, nilai yang tidak ada dapat dipenuhi dengan kategori "Tidak diketahui" atau "hilang".

b. Pengidentifikasian dan Penghapusan Duplikasi

Mengidentifikasi dan menghapus duplikat adalah langkah berikutnya dalam pembersihan data. Ketika satu atau lebih baris data yang sama muncul di dataset lebih dari satu kali, disebut duplikasi data. Baris-baris dengan nilai yang sama di semua atau sebagian besar kolom dapat ditemukan. Query SQL atau fungsi khusus dalam bahasa pemrograman data seperti Python atau R dapat digunakan untuk menjalankan proses ini.

Setelah duplikasi ditemukan, langkah selanjutnya adalah menghapus baris duplikat. Ini biasanya dilakukan dengan mempertahankan satu instance baris duplikat dan menghapus yang lain. Dengan menghilangkan duplikat, kami memastikan bahwa dataset yang digunakan untuk analisis tidak mengandung jumlah data yang berlebihan atau tidak akurat.

c. Standarisasi dan Normalisasi Data

Standarisasi dan normalisasi data adalah langkah terakhir dalam pembersihan data. Proses mengubah skala kumpulan data sehingga nilai-nilainya berada dalam rentang yang sama dikenal sebagai normalisasi data. Salah satu teknik normalisasi yang paling umum adalah skala min-max, yang mengubah nilai kumpulan data ke dalam rentang [0, 1]. Ketika menggunakan algoritma pengajaran mesin yang sensitif terhadap skala data, normalisasi sangat penting.

Standarisasi data adalah langkah penting selain normalisasi. Standarisasi adalah proses mengubah nilai-nilai dalam kumpulan sehingga memiliki distribusi dengan mean 0 dan standar deviasi 1. Proses ini dimulai

dengan menghitung standar deviasi dan mean dari setiap fitur kumpulan, kemudian mengurangi mean dari masing-masing nilai, dan kemudian membagi hasilnya dengan standar deviasi. Algoritma pengajaran mesin yang menggunakan data terdistribusi normal, seperti regresi linier atau analisis komponen utama (PCA), sering menggunakan standarisasi.

Standarisasi dan normalisasi sangat penting untuk memastikan bahwa data berada dalam rentang yang sama dan memiliki distribusi yang seragam. Pada akhirnya, ini meningkatkan kinerja algoritma pengajaran mesin dan analisis statistik.

Pembersihan data yang efektif sangat penting untuk mendapatkan hasil analisis yang akurat dan dapat diandalkan, yang pada gilirannya mendukung pengambilan keputusan yang lebih baik dan proses pengambilan keputusan yang lebih baik. Pembersihan data dilakukan melalui deteksi dan penanganan nilai yang tidak ada, identifikasi dan penghapusan duplikat, dan normalisasi dan standarisasi data.

3. Reduksi Data dalam Analisis Data: Deskripsi dan Narasi

Reduksi data adalah langkah krusial dalam analisis data yang bertujuan untuk mengurangi volume data tanpa menghilangkan informasi penting. Teknik-teknik utama dalam reduksi data termasuk seleksi fitur (feature selection), ekstraksi fitur (feature extraction), dan sampling data. Setiap teknik memiliki pendekatan dan manfaat yang berbeda, yang berkontribusi pada peningkatan efisiensi dan efektivitas proses analisis data. Mari kita jelajahi setiap teknik ini dalam bentuk narasi.

a. Seleksi Fitur (Feature Selection)

Seleksi fitur adalah proses memilih subset dari fitur asli dalam dataset yang paling relevan untuk analisis. Bayangkan Anda memiliki dataset dengan ratusan kolom, namun tidak semuanya berkontribusi terhadap hasil

analisis yang diinginkan. Beberapa fitur mungkin tidak relevan, atau bahkan redundan, yang dapat menurunkan kinerja model analisis.

Metode Seleksi Fitur :

1) Filter Method: Anda menggunakan statistik sederhana untuk menilai relevansi setiap fitur secara independen. Misalnya, dengan korelasi Pearson, Anda bisa menentukan seberapa kuat hubungan antara fitur dan target.
2) Wrapper Method:Dengan menggunakan algoritma machine learning, Anda mengevaluasi berbagai kombinasi fitur untuk menemukan set yang memberikan kinerja terbaik. Teknik seperti recursive feature elimination (RFE) menghapus fitur satu per satu dan mengevaluasi dampaknya pada kinerja model.
3) Embedded Method:Proses ini terjadi secara bersamaan dengan pelatihan model. Algoritma seperti LASSO tidak hanya memprediksi tetapi juga memilih fitur yang paling penting selama proses pelatihan.

Dengan melakukan seleksi fitur, Anda mengurangi risiko overfitting, meningkatkan interpretabilitas model, dan mempercepat waktu pelatihan model.

b. Ekstraksi Fitur (Feature Extraction)

Ekstraksi fitur adalah proses transformasi data dari ruang dimensi tinggi ke ruang dimensi lebih rendah, menciptakan fitur baru yang menggabungkan informasi dari fitur asli. Misalnya, bayangkan Anda memiliki dataset dengan fitur-fitur yang saling berkorelasi tinggi. Teknik ekstraksi fitur dapat menyederhanakan data ini.

Metode Ekstraksi Fitur :

1) Principal Component Analysis (PCA):PCA mengubah data ke dalam komponen utama yang paling bervariasi. Misalnya, jika Anda memiliki data tentang berbagai spesies bunga dengan panjang dan lebar

kelopak, PCA dapat mengurangi dimensi dengan menggabungkan panjang dan lebar menjadi satu atau dua komponen utama.

2) Linear Discriminant Analysis (LDA):LDA fokus pada memaksimalkan separasi antar kelas dalam data, berguna untuk klasifikasi. Bayangkan mengelompokkan data pasien berdasarkan hasil tes medis untuk diagnosis penyakit tertentu.
3) t-Distributed Stochastic Neighbor Embedding (t-SNE): Teknik ini sering digunakan untuk visualisasi data yang kompleks dalam dimensi rendah, seperti memetakan hubungan antara ribuan pelanggan berdasarkan kebiasaan belanja mereka.(Aggarwal, 2015)

Ekstraksi fitur membantu dalam mengurangi dimensi data tanpa kehilangan informasi penting, yang dapat meningkatkan kinerja model dan mempermudah visualisasi data yang kompleks.

c. Sampling Data

Sampling data adalah proses memilih subset dari data asli yang representatif untuk analisis. Misalkan Anda memiliki dataset dengan jutaan catatan transaksi, namun menganalisis semuanya secara langsung membutuhkan waktu dan sumber daya yang besar.

Metode Sampling Data :

1) Random Sampling: Memilih sampel secara acak dari dataset asli. Misalnya, memilih 10% dari total data transaksi secara acak.
2) Stratified Sampling:Memilih sampel dengan memastikan bahwa setiap strata atau kelompok dalam dataset terwakili dengan proporsi yang sama. Misalnya, memastikan semua kategori produk terwakili dalam sampel penjualan.

3) Systematic Sampling: Memilih setiap n-th elemen dari dataset setelah menentukan titik awal secara acak. Misalnya, memilih setiap transaksi ke-100 dari daftar transaksi.

Sampling data memungkinkan analisis yang lebih cepat dan lebih efisien dengan tetap mempertahankan representativitas data asli.(*Dunham - Data Mining*, n.d.)

Reduksi data melalui seleksi fitur, ekstraksi fitur, dan sampling data adalah langkah-langkah penting dalam analisis data yang membantu meningkatkan efisiensi dan efektivitas proses analisis. Dengan mengurangi jumlah fitur dan data yang perlu dianalisis, kita dapat mengembangkan model yang lebih cepat, lebih akurat, dan lebih mudah diinterpretasikan. Teknik-teknik ini memastikan bahwa kita hanya menggunakan data yang paling relevan, mengurangi risiko overfitting, dan memaksimalkan penggunaan sumber daya komputasi.

4. Integrasi Data dalam Analisis Data

Integrasi data adalah proses penting dalam analisis data yang bertujuan untuk menggabungkan data dari berbagai sumber menjadi satu set data yang koheren dan konsisten. Ini memungkinkan analisis data yang lebih komprehensif dan mendalam, serta membantu dalam pengambilan keputusan yang lebih baik. Proses integrasi data mencakup dua aspek utama: penggabungan data dari berbagai sumber dan resolusi konflik data.

Penggabungan Data dari Berbagai Sumber:

a. Penggabungan Data

Penggabungan data melibatkan proses pengumpulan dan penyatuan data dari berbagai sumber yang berbeda untuk membentuk satu dataset terpadu. Sumber data ini bisa mencakup sistem database relasional, data warehouse, file flat seperti CSV, serta sumber eksternal seperti situs web dan media sosial.

Tujuan utama dari penggabungan data adalah untuk menyediakan pandangan holistik terhadap data yang ada.

Langkah-langkah Penggabungan Data :

1) Identifikasi Sumber Data : Langkah pertama adalah mengidentifikasi semua sumber data yang relevan. Misalnya, mengidentifikasi data penjualan dari sistem CRM dan data keuangan dari sistem ERP.
2) Ekstraksi Data : Data diekstraksi dari setiap sumber menggunakan alat atau skrip ETL (Extract, Transform, Load).
3) Transformasi Data : Data yang diekstraksi kemudian ditransformasi ke dalam format yang konsisten. Ini mencakup pembersihan data, normalisasi, dan konversi format data.
4) Penggabungan Data : Data yang telah ditransformasi digabungkan menjadi satu dataset terpadu menggunakan metode seperti join atau merge.(Hancock, n.d.)

Contoh Penggabungan Data :

Sebuah perusahaan menggabungkan data penjualan dari sistem CRM dengan data keuangan dari sistem ERP. Data dari kedua sistem ini diekstraksi, diubah ke dalam format yang konsisten (seperti format tanggal yang sama), dan kemudian digabungkan berdasarkan ID pelanggan atau nomor faktur.

Resolusi Konflik Data :

Resolusi konflik data adalah proses mengidentifikasi dan menyelesaikan inkonsistensi atau konflik yang muncul saat data dari berbagai sumber digabungkan. Konflik data bisa terjadi karena perbedaan format data, nilai yang bertentangan, atau adanya duplikasi data.

Langkah-langkah Resolusi Konflik Data :

1) Deteksi Konflik:Mengidentifikasi konflik yang ada dalam dataset, seperti nilai yang bertentangan atau data duplikat.
2) Analisis Konflik:Menganalisis penyebab konflik untuk memahami mengapa konflik tersebut terjadi.
3) Resolusi Konflik:Menyelesaikan konflik dengan berbagai strategi:
   a) Kebijakan Prioritas Sumber:Menentukan sumber data yang lebih dipercaya dan menggunakan data dari sumber tersebut.
   b) Penggabungan Nilai:Menggabungkan nilai dari berbagai sumber untuk menciptakan nilai yang lebih representatif.
   c) Validasi Manual: Menyelesaikan konflik secara manual oleh seorang ahli yang dapat memverifikasi data yang benar.

Contoh Resolusi Konflik Data :

Jika terdapat konflik antara data alamat pelanggan dari dua sistem berbeda, perusahaan dapat menentukan bahwa data dari sistem CRM lebih dipercaya dan menggunakan data tersebut. Atau, jika terdapat perbedaan dalam data penjualan harian antara dua sistem, mereka bisa mengambil rata-rata dari kedua nilai tersebut.

Integrasi data adalah langkah penting dalam analisis data yang mencakup penggabungan data dari berbagai sumber dan resolusi konflik data. Proses ini memastikan dataset yang lengkap, konsisten, dan akurat, yang mendukung analisis yang lebih efektif dan pengambilan keputusan yang lebih baik. Dengan integrasi data yang baik, organisasi dapat mengoptimalkan penggunaan data mereka untuk mencapai wawasan yang lebih mendalam dan keputusan bisnis yang lebih informasional. (Tan *et al.*, n.d.)

DAFTAR PUSTAKA

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

Carudin, C., Marisa, M., Murnawan, M., Reba, F., Koibur, M. E., Thantawi, A. M., Halim, A., Wattimena, F. Y., Agusdi, Y., & Safitri, N. (2024). *Buku Ajar Data Mining*. PT. Sonpedia Publishing Indonesia. https://books.google.co.id/books?id=m-QGEQAAQBAJ

Dr. Uky Yudatama, S. S. M. K. M. M., Ir. Aris Dianto, S. S. S. S. T. S. K. M. A. D. S., Anggun Fergina, S. K. M. K., Rini Tisnawati, S. K. M. T. M. T. A., Remuz MB Kmurawak, S. T. M. T., Ardi Taryanto, S. S. M. M., Drs. Amna, M. I. T., Karina W Noviyanti, S. S. M. T., Cecep Kurnia Sastradipraja, S. K. M. K., & Yuda Syahidin, S. S. T. M. K. M. T. A. C. (2023). *Sistem Enterprise di Era Digital: Inovasi, Transformasi, dan Keberlanjutan*. Kaizen Media Publishing. https://books.google.co.id/books?id=m4zcEAAAQBAJ

Dr. Uky Yudatama, S. S. M. K. M. M., Syamsiyah, N., Wiranata, A. D., Rahmi Imanda S. Kom., M. K., Ma'sum, H., Murdiyanto, A. W., Widyanto, R. A., & Dewi, D. D. (2023). *Memahami Teknologi Informasi: Prinsip, Pengembangan, dan Penerapan*. Kaizen Media Publishing. https://books.google.co.id/books?id=P1HcEAAAQBAJ

Dunham - Data Mining. (n.d.).

Hancock, M. F. (n.d.). Practical Data Mining.

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

BAB 7 | CLUSTERING

Erna Hudianti Pujiarini, S.Si., M.Si.

A. Pengertian Clustering

Clustering adalah salah satu metode untuk mengelompokkan instance(sample) menjadi beberapa groupatau subsetatau clusterberdasarkan "kemiripan"dengan instance yang lain. (Bedy Purnama, 2019)

Clustering adalah proses pengelompokan data atau objek ke dalam kelompok-kelompok yang serupa berdasarkan kesamaan karakteristik tertentu.

Karakteristik setiap cluster adalah kemiripan data harus tinggi di dalam satu cluster, dan ketidakmiripan data harus tinggi pada cluster yang lain. (Irwansyah Saputra, 2022)

Tujuan dari clustering adalah untuk mengelompokkan data yang mirip ke dalam kelompok tertentu, tanpa adanya label kelas sebelumnya.

B. Implementasi Clustering

Dalam beberapa bidang analisis clustering digunakan untuk membantu pengambilan keputusan,. Beberapa contoh penggunaan analisis clustering dalam bidang-bidang sebagai berikut :

1. Bidang pemasaran menggunakan analisis clustering untuk segmentasi pasar, yaitu mengelompokkan konsumen berdasarkan preferensi, perilaku pembelian, atau demografi. Ini memungkinkan perusahaan untuk menyasar pasar yang sesuai dengan kebutuhan produk atau layanan mereka

2. Bidang e-commerce menggunakan analisis clustering untuk menganalisis perilaku konsumen, memprediksi preferensi produk, dan menyarankan produk yang relevan berdasarkan pola pembelian sebelumnya
3. Dalam ilmu geografi, urbanisme, dan lingkungan menggunakan analisis clustering untuk mengelompokkan area geografis berdasarkan karakteristik tertentu seperti polusi udara, kepadatan penduduk, atau penggunaan lahan.
4. Dalam bidang biologi molekuler dan kedokteran menggunakan analisis clustering untuk mengelompokkan gen, sel, atau pasien berdasarkan ekspresi genetik atau parameter klinis. Ini membantu dalam pemahaman tentang pola penyakit, pengobatan yang tepat, dan pengembangan obat.
5. Dalam sosiologi dan ilmu sosial menggunakan analisis clustering untuk mengelompokkan individu atau komunitas berdasarkan perilaku sosial, preferensi budaya, atau keanggotaan dalam kelompok tertentu.

C. Perbedaan Clustering dan Klasifikasi

Perbedaan antara clustering dan klasifikasi berdasarkan tujuan sebagai berikut:

1. Tujuan Clustering adalah mengelompokkan data dalam kelompok yang serupa berdasarkan kesamaan karakteristik tertentu dan menemukan pola dalam data tanpa adanya label kelas sebelumnya.
2. Tujuan Klasifikasi adalah mengklasifikasikan data ke dalam kelas yang sudah ditentukan sebelumnya dan membangun model yang dapat memprediksi kelas untuk data baru berdasarkan pembelajaran dari data yang sudah diklasifikasikan sebelumnya

Perbedaan antara clustering dan klasifikasi berdasarkan bentuk pembelajaran sebagai berikut :

1. Bentuk pembelajaran Clustering merupakan pembelajaran yang tidak terawasi (unsupervised learning), artinya tidak ada label kelas yang tersedia dalam data

2. Bentuk pembelajaran Klasifikasi merupakan pembelajaran yang terawasi (supervised learning), artinya ada label kelas yang tersedia dalam data

D. Jenis-Jenis Clustering

Jenis Clustering berdasarkan pendekatan Hierarchical Clustering dan Partitional Clustering sebagai berikut :

1. Hierarchical Clustering

Hierarchical Clustering mengelompokkan data ke dalam struktur hirarkis. Cluster yang berbentuk disajikan dalam bentuk grafik dendrogram. Jumlah cluster tidak ditentukan terlebih dahulu namun dibentuk secara bertahap dengan mempertimbangkan kesamaan antara titik data.

Terdapat dua jenis hierarchical clustering yaitu agglomerative clustering dan divisive clustering. Agglomerative clustering atau clustering bottom up dimulai dengan menganggap setiap titik data sebagai cluster tersendiri, kemudian menggabungkan cluster yang paling mirip dalam satu cluster, dan dilakukan sampai mencapai jumlah cluster yang diinginkan. Divisive clustering atau clustering top up dimulai dengan satu klaster besar yang mencakup semua titik data, kemudian memecah cluster menjadi cluster yang lebih kecil dan mencapai jumlah cluster yang diinginkan.

Beberapa algoritma agglomerative clustering diantaranya single linkage, complete linkage, average linkage, Ward's Method. Beberapa algoritma divisive clustering diantaranya Splinter Average, Automatic Interaction Detection (AID).

2. Partitional Clustering

Partitional clustering mengelompokkan data pada sejumlah cluster , di mana setiap titik data hanya bisa menjadi bagian dari satu cluster. Sebelumnya jumlah cluster ditentukan terlebih dahulu. Beberapa algoritma partitional clustering diantaranya k-means, k-medoids dan fuzzy C-means.

Jenis Clustering berdasarkan pendekatan Density based Clustering dan Centroid based Clustering sebagai berikut :

1. **Density based Clustering**

   Density based Clustering mengelompokkan data berdasarkan kepadatan data. Algoritma density based clustering dimulai dengan mencari daerah di mana terdapat kepadatan titik data yang tinggi, yang disebut sebagai cluster. Kemudian titik data yang terletak dalam daerah kepadatan yang tinggi dianggap sebagai bagian dari cluster yang sama.

   Contoh algoritma density based clustering adalah Density Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN mengidentifikasi cluster berdasarkan keterhubungan antara titik-titik data dalam jarak tertentu.

2. **Centroid based Clustering**

   Centroid based clustering mengelompokkan titik data berdasarkan posisi pusat cluster (centroid). Algoritma centroid based clustering dimulai dengan inisialisasi titik pusat cluster yang mewakili cluster secara keseluruhan kemudian cluster (centroid) diperbarui secara iteratif. Contoh algoritma centroid-based clustering termasuk K-means, K-medoids dan Fuzzy C-means.

Jenis Clustering berdasarkan pendekatan Hard Clustering dan Soft Clustering sebagai berikut :

1. **Hard Clustering**

   Pada hard clustering, setiap titik data secara tegas dikelompokkan dalam satu cluster tertentu sesuai dengan cluster yang memiliki pusat terdekat. Kelebihan dari hard clustering memudahkan interpretasi karena setiap titik data hanya memiliki satu cluster. Kekurangan dari hard clustering tidak mampu mengakomodasi data yang sebenarnya dapat dimasukkan ke dalam beberapa cluster. Contoh hard clustering adalah k-means, di mana setiap titik data diberi label cluster tunggal yang sesuai dengan cluster yang memiliki pusat terdekat.

2. Soft Clustering

Pada soft clustering, setiap titik data tidak secara tegas dikelompokkan dalam satu cluster tertentu, dimungkinkan masuk dalam beberapa cluster dengan disertai nilai keanggotaan untuk menunjukkan tingkat ketidakpastian dalam cluster tersebut.

Kelebihan dari soft clustering, kemampuannya untuk menangani ketidakpastian dalam cluster. Kekurangan dari soft clustering lembut, interpretasi yang lebih kompleks, karena setiap titik data memiliki tingkat keanggotaan yang bersifat probabilistik terhadap setiap cluster. Contoh soft clustering adalah Fuzzy C-means (FCM), di mana setiap titik data memiliki derajat keanggotaan di setiap cluster.

E. Konsep Jarak

Konsep jarak penting dalam clustering, yang nantinya digunakan untuk mengukur kesamaan antara data dalam ruang fitur. Beberapa konsep jarak yang digunakan dalam clustering :

1. Jarak Euclidean

Jarak Euclidean mengukur jarak garis lurus antara dua titik dalam ruang fitur berdimensi n. Semakin kecil jarak Euclidean antara dua titik data, maka semakin mirip atau dekat kedua data titik tersebut dalam ruang fitur. Jarak Euclidean paling sesuai ketika data berada dalam ruang Euclidean dan tidak terlalu dipengaruhi oleh outlier.

2. Jarak Manhattan

Jarak Manhattan atau jarak kota blok atau jarak L1, digunakan untuk mengukur jarak antara dua titik dalam ruang fitur dengan jumlah perbedaan absolut dalam koordinat mereka. Jarak Manhattan digunakan dalam situasi di mana perjalanan sepanjang sumbu tegak lurus lebih relevan daripada jarak garis lurus.

3. Jarak Minkowski

Jarak Minkowski generalisasi dari jarak Euclidean dan jarak Manhattan, karena jarak ini memungkinkan fleksibilitas dalam menyesuaikan dalam perbedaan dimensi dengan

menggunakan parameter p. Nilai p tergantung pada sifat, nilai p yang besar cenderung memberikan lebih banyak penekanan pada dimensi yang dominan, sedangkan nilai p yang lebih kecil cenderung memberikan penekanan yang lebih merata pada semua dimensi.

4. **Jarak Mahalanobis**

   Jarak Mahalanobis memperhitungkan kovarians antara variabel dalam ruang fitur. Jarak antara dua titik data dihitung dengan mempertimbangkan matriks kovarians dari seluruh dataset

F. Jumlah Clustering

Tahap yang tidak kalah penting dalam proses clustering adalah memilih jumlah cluster yang optimal. Penentuan jumlah cluster dapat berdampak pada interpretasi hasil clustering. Untuk memilih jumlah cluster yang optimal digunakan metode sebagai berikut :

1. **Metode Elbow**

   Metode Elbow merupakan metode untuk menentukan jumlah cluster dengan menggunakan grafik dari nilai Sum of Square Error (SSE) dari masing-masing nilai cluster. Kemudian dari grafik dipilih jumlah cluster yang nilai cluster pertama dengan nilai cluster kedua membentuk sudut dalam grafik atau nilainya mulai mengalami penurunan besar maka jumlah nilai cluster tersebut yang optimal.

2. **Metode Silhouette**

   Metode Silhouette mengukur seberapa baik setiap titik data sesuai dengan clusternya sendiri dibandingkan dengan cluster yang lain. Nilai Silhouette yang lebih tinggi menunjukkan clustering yang lebih baik.

3. **Metode Gap Statistic**

   Metode Gap statistic menghitung gap antara log variasi dalam cluster dari data asli dan rata-rata log variasi dalam cluster dari data acak. Nilai Gap statistic yang lebih tinggi menunjukkan clustering yang lebih baik.

4. Metode Davies Bouldin Index

Metode Davies Bouldin Index dapat pula digunakan untuk menentukan jumlah cluster optimal dengan melihat nilai Davies Bouldin Index yang paling rendah.

G. Algoritma Clustering

1 Algoritma K-Means

Algoritma k means merupakan algoritma clustering yang paling populer dan sederhana digunakan untuk membagi dataset menjadi *k* cluster.

Langkah-langkah Algoritma K-Means :

a. Inisialisasi jumlah cluster sebanyak *k* dan centroid awal dari cluster secara acak.
b. Tetapkan setiap titik data ke centroid terdekat berdasarkan jarak Euclidean.
c. Memperbarui Centroid dengan menghitung rata-rata dari semua titik data dalam cluster tersebut sebagai centroid baru.
d. Ulangi langkah b) dan c) sampai centroid tidak berubah secara signifikan atau mencapai jumlah iterasi maksimum yang telah ditentukan.

2 Algoritma single linkage

Langkah-langkah algoritma single linkage sebagai berikut :

a. Dimulai dengan menganggap setiap titik data sebagai satu cluster.
b. Hitung jarak antara setiap pasang cluster.
c. Gabungkan dua cluster yang memiliki jarak terdekat menjadi satu cluster.
d. Perbarui matriks jarak dengan memperhitungkan gabungan baru cluster dan jaraknya dengan semua cluster lainnya.
e. Ulangi langkah-langkah b) sampai d) sampai hanya satu cluster yang tersisa atau sejumlah cluster yang telah ditentukan sebelumnya.

H. Evaluasi Clustering

Evaluasi clustering merupakan metode untuk mengukur kualitas hasil clustering, Berikut ini merupakan metode untuk evaluasi clustering :

1. **Sum of Squared Errors**

   Sum of Squared Errors (SSE) merupakan jumlah kuadrat jarak antara setiap titik data dalam cluster dengan pusat clusternya. SSE yang lebih rendah menunjukkan kelompok yang lebih padat dan data yang lebih seragam dalam kelompoknya. Namun demikian SSE cenderung menurun saat jumlah kelompok meningkat, sehingga SSE sendiri tidak memberikan ukuran yang objektif untuk kualitas clustering

2. **Silhouette Score**

   Silhouette Score digunakan untuk mengukur seberapa baik setiap titik data dalam cluster cocok dengan clusternya sendiri dibandingkan dengan cluster lain. Silhouette Score berkisar dari -1 hingga 1, di mana nilai yang lebih tinggi menunjukkan clustering yang lebih baik. Nilai Silhouette Score yang positif menunjukkan bahwa titik data cocok dengan clusternya, sedangkan nilai yang negatif menunjukkan bahwa titik data mungkin cocok dengan cluster lain. Nilai yang mendekati nol menunjukkan bahwa titik data berada dekat dengan batas antara dua cluster.

3. **Davies Bouldin Indeks**

   Davies Bouldin Index (DBI) digunakan untuk mengukur seberapa baik sebuah cluster terpisah satu dengan yang lain. Dengan cara menghitung rata-rata ukuran rasio komponen kohesi dan separasi antara dua cluster. Nilai Davies Bouldin Index rendah menunjukkan cluster terpisah satu sama lain dengan baik, sedangkan nilai tinggi menunjukkan adanya tumpang tindih antar cluster.

I. Rangkuman

Clustering adalah proses pengelompokan data atau objek ke dalam kelompok-kelompok yang serupa berdasarkan kesamaan karakteristik tertentu. Karakteristik setiap cluster adalah kemiripan data harus tinggi di dalam satu cluster, dan ketidakmiripan data harus tinggi pada cluster yang lain.

Ada beberapa jenis pendekatan metode clustering, berdasarkan pendekatan Hierarchical Clustering dan Partitional Clustering, berdasarkan pendekatan Density based Clustering dan Centroid based Clustering, berdasarkan pendekatan Hard Clustering dan Soft Clustering.

Konsep jarak penting dalam clustering karena nantinya digunakan untuk mengukur kesamaan antara data dalam ruang fitur. Beberapa konsep jarak yang digunakan dalam clustering adalah jarak Euclidean, jarak Manhattan, jarak Minkowski, jarak Mahalanobis.

Pemilihan jumlah cluster yang optimal dapat berdampak pada interpretasi hasil clustering. Untuk memilih jumlah cluster yang optimal digunakan metode Elbow, metode Silhouette, metode Gab Statistic, metode Davies Bouldin Index.

Evaluasi clustering merupakan metode untuk mengukur kualitas hasil clustering, Metode yang digunakan untuk evaluasi clustering adalah metode Sum of Squared Errors, metode Silhouette Score, metode Davies Bouldin Indeks.

DAFTAR PUSTAKA

Bedy Purnama, S. M., 2019. *Pengantar Machine Learning "Konsep dan Praktikum dengan contoh Latihan Berbasis R da Python"*. Cetakan Pertama ed. Bandung: Informatika Bandung.

Irwansyah Saputra, D. A. K., 2022. *Machine Learning untuk Pemula.* Pertama ed. Bandung: Informatika Bandung.

BAB 8 | CLASSIFICATION

Marwan Ramdhany Edy, S.Pd., M.Kom.

A. Definisi Klasifikasi

Klasifikasi adalah salah satu tugas utama dalam pembelajaran mesin (machine learning) dan data mining. Klasifikasi melibatkan proses pengelompokan data ke dalam kategori atau kelas yang telah ditentukan sebelumnya. Tujuan utama dari klasifikasi adalah untuk memprediksi label kelas dari objek data berdasarkan atribut atau fitur yang dimilikinya.

Menurut Tan, Steinbach, dan Kumar (2005) dalam buku "Introduction to Data Mining," klasifikasi didefinisikan sebagai berikut:

"Klasifikasi adalah proses menemukan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep yang akan membantu untuk menggunakan model/fungsi tersebut dalam memprediksi kelas objek yang labelnya tidak diketahui.".

Pemrograman adalah proses menulis, menguji dan memperbaiki (debug) menggunakan kode untuk menjalankan program komputer (coding). Koding ini ditulis dalam bahasa pemrograman. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan pekerjaan sesuai keinginan seorang pemrogram atau disebut Programmer. Untuk dapat melakukan pemrograman, maka calon programmer diperlukan keterampilan khusus dan dasar seperti algoritma, logika dan penggunaan bahasa pemrograman itu sendiri.

1. Aplikasi Klasifikasi

Menurut (Han, Kamber and Pei, 2012) dalam buku "Data Mining: Concepts and Techniques", beberapa aplikasi klasifikasi yang penting antara lain :

a. Deteksi penipuan: Klasifikasi dapat digunakan untuk mendeteksi transaksi kartu kredit yang curang dengan mempelajari pola transaksi yang valid dan tidak valid.
b. Analisis risiko kredit: Klasifikasi dapat membantu lembaga keuangan dalam menganalisis risiko kredit calon peminjam berdasarkan karakteristik seperti pendapatan, pekerjaan, dan riwayat kredit.
c. Pemasaran: Klasifikasi dapat digunakan untuk memprediksi apakah seorang pelanggan akan merespons kampanye pemasaran tertentu atau tidak, sehingga memungkinkan perusahaan untuk mengarahkan upaya pemasaran mereka dengan lebih efektif.

Sementara itu, (Tan, Steinbach and Kumar, 2016) dalam buku "Introduction to Data Mining" menyebutkan beberapa aplikasi klasifikasi lainnya, seperti:

a. Diagnosis medis: Klasifikasi dapat membantu dalam mengidentifikasi jenis penyakit yang diderita pasien berdasarkan gejala-gejala dan hasil pemeriksaan medis.
b. Analisis gambar: Klasifikasi dapat digunakan untuk mengklasifikasikan objek dalam gambar atau citra digital, seperti mengenali wajah, deteksi objek, dan sebagainya.

2. Jenis-jenis Masalah Klasifikasi

Masalah klasifikasi bisa sangat beragam, mulai dari menangani data yang tidak seimbang hingga klasifikasi deret waktu multi-label. (Daly, 2003) membahas tantangan yang mendasari klasifikasi tipe rezim demokratis, sementara Papatheodoulou dkk. (2022) fokus pada klasifikasi deret waktu multi-label. Selain itu, Ma *et al*., (2017) membahas klasifikasi biner terbatas menggunakan pembelajaran ensemble, dan Pereira dkk. (2021) mengeksplorasi pengambilan sampel ulang data dalam masalah klasifikasi

hirarkis. Referensi-referensi ini menyoroti beragam pendekatan dan kompleksitas yang terlibat dalam menangani berbagai jenis masalah klasifikasi di berbagai bidang seperti politik, ilmu komputer, dan statistik.

B. Classification Process

1. Data Preparation

Persiapan data dalam klasifikasi adalah proses penting yang melibatkan pengorganisasian dan prapemrosesan data sebelum menerapkan algoritme klasifikasi. Langkah ini mencakup tugas-tugas seperti pembersihan data, penanganan nilai yang hilang, normalisasi, pemilihan fitur, dan transformasi untuk memastikan bahwa data dalam format yang sesuai untuk tugas klasifikasi (Yahya *et al*., 2021). Misalnya, dalam konteks klasifikasi digit tulisan tangan menggunakan Convolutional Neural Networks (CNN), persiapan data bertujuan untuk menghilangkan variabel yang berlebihan atau tidak relevan dari dataset untuk meningkatkan proses klasifikasi (Yahya *et al*., 2021).

Selain itu, persiapan data secara signifikan berkontribusi untuk meningkatkan akurasi klasifikasi. Misalnya, dalam pencitraan spektrometri massa, teknik prapemrosesan data digunakan untuk menghilangkan variasi teknis sambil mempertahankan informasi biologis, yang mengarah pada peningkatan substansial dalam akurasi klasifikasi (Deininger *et al*., 2022

Singkatnya, persiapan data adalah tahap awal yang penting dalam proses klasifikasi yang mencakup berbagai langkah prapemrosesan untuk memastikan bahwa data terstruktur dengan baik dan dioptimalkan untuk hasil klasifikasi yang akurat di berbagai domain.

2. Model Building

Pembuatan model dalam klasifikasi melibatkan proses pengembangan dan pelatihan model prediktif untuk mengklasifikasikan data ke dalam kategori atau kelas yang berbeda. Proses ini memainkan peran penting dalam

berbagai domain seperti ilmu komputer, geografi, dan ilmu material. Hoffmann dkk. (2019) membahas fusi model untuk klasifikasi jenis bangunan dari citra foto udara dan street view, dengan menekankan pada fusi tingkat keputusan dari beragam model ensembel yang dilatih secara independen dari setiap jenis citra

Selain itu, Taoufiq dkk. (2020) memperkenalkan HierarchyNet, sebuah model klasifikasi bangunan perkotaan berbasis CNN hierarkis yang menggabungkan hierarki kasar ke halus dalam dataset. Pendekatan ini memungkinkan model untuk mengekstrak fitur dan mengklasifikasikan berbagai tingkat hierarki secara bersamaan. Selain itu, Xiao dkk. (2022) menyelidiki klasifikasi dan anomali berdasarkan metode pembelajaran mesin yang diterapkan pada pemodelan informasi bangunan berskala besar, yang menekankan pentingnya Model Informasi Bangunan (Building Information Models/BIM) dalam mengumpulkan dan menyinkronkan data terkait konstruksi untuk tujuan klasifikasi.

3. Model Evaluation

Evaluasi model dalam klasifikasi melibatkan penilaian kinerja dan efektivitas model klasifikasi dalam mengkategorikan data secara akurat. Evaluasi ini sangat penting untuk menentukan keandalan dan efisiensi model dalam aplikasi dunia nyata. Berbagai metrik dan teknik digunakan untuk mengevaluasi model klasifikasi di berbagai domain.

Sebagai contoh, Yu dkk. (2008) menekankan pentingnya menggunakan akurasi klasifikasi dalam set pengujian sebagai kriteria evaluasi kinerja untuk model penilaian risiko kredit.

Selain itu, evaluasi model memerlukan penilaian berbagai aspek kinerja model. Salih & Abdulazeez (2021) menekankan pentingnya memilih metrik yang sesuai untuk evaluasi model, dengan mempertimbangkan berbagai metrik evaluasi yang tersedia untuk berbagai masalah dan aplikasi.

Kesimpulannya, evaluasi model dalam klasifikasi mencakup berbagai teknik dan metrik untuk mengevaluasi akurasi, keandalan, dan efisiensi model klasifikasi dalam mengkategorikan data dengan benar, memastikan penerapannya dalam skenario praktis.

C. Decision Tree

1. Cara Kerja Decision Trees

Pohon keputusan adalah alat yang mendasar dalam tugas klasifikasi, bekerja dengan cara membagi data secara rekursif berdasarkan nilai fitur untuk menciptakan struktur seperti pohon di mana node daun mewakili label kelas. Proses ini melibatkan pemilihan fitur terbaik untuk memisahkan data pada setiap node, biasanya berdasarkan kriteria seperti perolehan informasi atau ketidakmurnian Gini.

Pemisahan ini terus berlanjut hingga kriteria penghentian terpenuhi, seperti mencapai kedalaman pohon maksimum atau memiliki node dengan label kelas yang homogen.

Model pohon keputusan dibangun secara iteratif, dimulai dari node akar dan berlanjut ke node daun. Pada setiap simpul internal, keputusan dibuat berdasarkan nilai fitur, yang mengarah ke cabang-cabang yang berbeda yang sesuai dengan hasil yang berbeda. Proses ini terus berlanjut hingga pohon tersebut tumbuh sempurna, menangkap hubungan antara fitur input dan kelas target.

D. Naive Bayes

1. Teorema Bayes

Teorema Bayes adalah konsep dasar dalam klasifikasi yang melibatkan prediksi probabilitas titik data yang termasuk dalam populasi atau kelas tertentu. Teorema ini banyak digunakan dalam algoritme pembelajaran mesin, khususnya dalam konteks Naïve Bayes Classifier. Pengklasifikasi ini menggabungkan pengetahuan

sebelumnya dengan informasi baru untuk membuat prediksi berdasarkan probabilitas yang dihitung menggunakan teorema Bayes (Wibawa *et al.*, 2019).

Singkatnya, teorema Bayes memainkan peran penting dalam tugas klasifikasi, terutama dalam pengembangan dan implementasi Naïve Bayes Classifier, yang memungkinkan prediksi dan klasifikasi yang akurat berdasarkan prinsip-prinsip probabilistik.

E. Logistic Regression

1. Regresi Linear

Dalam ranah pembelajaran mesin, regresi linier menjadi landasan untuk memodelkan hubungan antar variabel. Meskipun secara tradisional dikaitkan dengan prediksi hasil yang berkelanjutan, keserbagunaan regresi linier meluas ke ranah klasifikasi, menawarkan perspektif unik tentang pengenalan pola dan analisis data.

Regresi linier, sebuah metode yang berakar pada prinsip-prinsip statistik, telah menemukan tujuan baru dalam ranah tugas klasifikasi. Dengan memanfaatkan esensi dari hubungan linier antar variabel, teknik ini mengungkap jalur untuk mengkategorikan titik data ke dalam kelas-kelas yang berbeda, melampaui batas-batas analisis regresi tradisional.

Dalam permadani besar pembelajaran mesin, klasifikasi regresi linier berdiri sebagai bukti kekuatan kesederhanaan dan keanggunan yang bertahan lama. Ketika kami mempelajari lebih dalam seluk-beluk regresi linier dalam klasifikasi, kami menemukan dunia dengan potensi yang belum tersentuh dan kemungkinan yang tak terbatas, di mana perpaduan prinsip-prinsip statistik dan teknologi modern membuka jalan bagi terobosan transformatif di bidang pengenalan pola.

2. Fungsi Logistik

Di bidang machine learning, kombinasi fungsi logistik dan regresi linier menghadirkan berbagai kemungkinan yang melampaui batas-batas tradisional. Menjelajahi kompleksitas klasifikasi dan pemodelan prediktif, integrasi kedua teknik ini muncul sebagai sumber inovasi, memberikan perspektif baru tentang pengenalan pola dan analisis data.

Inti dari perpaduan ini adalah regresi logistik, sebuah teknik yang menghubungkan regresi dan klasifikasi. Dengan memanfaatkan sifat sigmoidal dari fungsi logistik, kami memulai perjalanan untuk memperkirakan probabilitas dan membuat keputusan yang tepat berdasarkan respons kategorikal. Kekuatan regresi logistik terletak pada kemampuannya untuk tidak hanya memprediksi hasil, tetapi juga mengklasifikasikan titik-titik data secara akurat dan percaya diri (Lever dkk., 2016).

Ketika kita menavigasi hubungan yang rumit antara regresi linier dan logistik, kita menemukan banyak kemungkinan dan peluang. Mulai dari menilai risiko pada penyakit menular hingga mengoptimalkan penugasan landasan pacu dalam penerbangan, aplikasi fungsi logistik dalam regresi linier sangat bervariasi dan berdampak besar, membentuk lanskap ilmu data kontemporer dan analisis prediktif (Ren *et al*., 2014; Kanjanasurat *et al*., 2023).

3. Model Regresi Logistik

Dalam bidang pembelajaran mesin, kombinasi strategis dari fungsi logistik dan regresi linier menonjol sebagai inovasi yang sangat penting. Perpaduan ini menawarkan perspektif baru dalam klasifikasi dan pemodelan prediktif, yang membentuk kembali lanskap analisis data.

Integrasi regresi linier dengan fungsi logistik menghasilkan model regresi logistik, memperkaya bidang regresi linier dengan memberikan pendekatan baru untuk memperkirakan probabilitas dan membuat keputusan berdasarkan respons kategorikal. Kolaborasi ini melampaui

batas-batas tradisional, merevolusi praktik analisis data Landwehr dkk. (2005).

Inti dari integrasi ini adalah pohon model logistik, perluasan dari model regresi logistik linier yang mendorong melampaui batas-batas konvensional. Dengan memanfaatkan kerangka kerja pemodelan regresi kernel-mesin logistik, para peneliti dapat mempelajari studi yang kompleks seperti studi asosiasi genom dan analisis polimorfisme nukleotida tunggal, yang menjelaskan aspek genetik dari penyakit seperti kanker payudara (Wu *et al*., 2010).

Analisis komparatif model regresi linier dan logistik dalam menangani data persentase menunjukkan kemampuan adaptasi teknik-teknik ini di berbagai dataset, mulai dari Listeria monocytogenes hingga Clostridium botulinum. Studi ini menyoroti keampuhan regresi linier dan logistik dalam menangkap nuansa data persentase dengan presisi dan efisiensi (Zhao *et al*., 2001).

F. Support Vector Machine (SVM)

1. Hyperplanes dan Support Vectors

Dalam bidang pembelajaran mesin, konvergensi hyperplane dan vektor pendukung dalam Support Vector Machines (SVM) menawarkan ranah presisi dan kekuatan prediksi. Perjalanan melalui konstruksi matematis SVM ini mengungkapkan bagaimana hyperplanes dan vektor pendukung bekerja sama untuk mendefinisikan ulang batas klasifikasi dan meningkatkan pengenalan pola.

Penerapan SVM pada citra penginderaan jauh hiperspektral menunjukkan bagaimana vektor pendukung memandu klasifikasi, membantu dalam mengidentifikasi pola yang rumit dalam set data yang luas. Aplikasi ini menyoroti potensi SVM dalam mengungkap informasi tersembunyi dalam citra penginderaan jauh.

Eksplorasi lebih lanjut mengarah pada penggabungan mesin vektor pendukung kuadrat terkecil fuzzy kembar, sebuah pendekatan baru yang menggabungkan prinsip-prinsip dari SVM Kuadrat Terkecil dan SVM Kembar. Perpaduan ini menunjukkan kemampuan beradaptasi dan keserbagunaan SVM dalam menangani tugas klasifikasi yang kompleks dengan presisi dan ketangguhan.

Fleksibilitas SVM terbukti dalam berbagai domain, mulai dari deteksi cacat industri hingga klasifikasi data ekspresi gen. Keberhasilan penerapan SVM di berbagai bidang ini menggarisbawahi efektivitasnya dalam memecahkan tantangan dunia nyata dengan akurasi dan wawasan.

2. Fungsi Kernel

Dalam bidang pembelajaran mesin, Support Vector Machines (SVM) berdiri sebagai mercusuar inovasi, menawarkan alat yang ampuh untuk tugas-tugas klasifikasi dan regresi. Inti dari SVM adalah konsep fungsi kernel yang penuh teka-teki, yang memainkan peran penting dalam mentransformasi data dan membuka pola tersembunyi dalam kumpulan data yang kompleks.

Bayangkan sebuah dunia di mana titik-titik data tidak terbatas pada keterpisahan linier tetapi dengan mudah dipetakan ke dalam ruang dimensi yang lebih tinggi, di mana hubungan dan batas-batas yang rumit dapat dilihat dengan jelas. Transformasi ini dimungkinkan oleh keajaiban fungsi kernel, yang berfungsi sebagai mesin yang mendorong SVM menuju prediksi yang akurat dan efisien.

Saat kami mengungkap misteri fungsi kernel dalam SVM, kami memulai perjalanan penemuan, mengeksplorasi potensi tak terbatas dari konstruksi matematika ini dalam membentuk kembali lanskap pembelajaran mesin dan analisis prediktif. Bersama-sama, kami menyelidiki dunia fungsi kernel yang rumit, di mana inovasi bertemu dengan presisi, dan batas-batas kemungkinan didefinisikan ulang.

3. **Klasifikasi dengan SVM**

Inti dari SVM adalah sebuah konsep canggih yang dikenal sebagai fungsi kernel, yang memainkan peran penting dalam membentuk batas-batas klasifikasi dan memungkinkan prediksi yang akurat.

a. **Pengantar untuk Support Vector Machines (SVMs)** Support Vector Machines adalah sebuah kelas algoritme pembelajaran terawasi yang unggul dalam tugas klasifikasi dengan menemukan hyperplane optimal yang memisahkan kelas-kelas yang berbeda dalam ruang fitur. SVM bertujuan untuk memaksimalkan margin antar kelas, sehingga memungkinkan klasifikasi yang efektif bahkan dalam set data yang kompleks.

b. **Memahami Fungsi Kernel dalam SVM** Fungsi kernel dalam SVM berfungsi sebagai jembatan untuk mentransformasikan data ke dalam ruang dimensi yang lebih tinggi, yang memungkinkan terciptanya batas-batas keputusan yang tidak linear. Dengan memanfaatkan fungsi kernel seperti fungsi linear, polinomial, dan radial basis, SVM dapat menangkap pola dan hubungan yang rumit di dalam data.

c. **Aplikasi SVM dalam Skenario Dunia Nyata** Fleksibilitas SVM dengan fungsi kernel meluas ke berbagai domain, termasuk penginderaan jarak jauh, perawatan kesehatan, dan keuangan. Dari klasifikasi tutupan lahan pada citra satelit hingga deteksi kesalahan pada sirkuit elektronik daya, SVM dengan fungsi kernel menawarkan solusi yang akurat dan efisien untuk beragam tantangan klasifikasi.

d. **Mengoptimalkan SVM dengan Parameter Kernel** Untuk meningkatkan kinerja SVM, mengoptimalkan parameter kernel sangat penting. Dengan menyempurnakan hyperparameter dan memilih fungsi kernel yang sesuai, SVM dapat mencapai akurasi klasifikasi yang optimal dan ketangguhan dalam menangani set data yang kompleks.

G. Imbalanced Data

1. Undersampling

Undersampling adalah teknik yang digunakan dalam pembelajaran mesin untuk mengatasi masalah set data yang tidak seimbang, di mana satu kelas secara signifikan melebihi kelas lainnya. Dalam undersampling, contoh dari kelas mayoritas dihilangkan untuk mencapai distribusi yang lebih seimbang di antara kelas. Proses ini membantu mencegah model menjadi bias terhadap kelas mayoritas dan meningkatkan kemampuannya untuk memprediksi kelas minoritas secara akurat Hasanin dkk. (2019).

Undersampling dapat sangat berguna dalam skenario di mana kelas mayoritas berisi data yang berisik atau berlebihan yang dapat menghambat kemampuan model untuk melakukan generalisasi dengan baik. Dengan menghapus contoh-contoh ini, undersampling dapat meningkatkan kinerja dan efisiensi model dengan berfokus pada titik data yang paling relevan (Showalter, 2019).

Selain itu, undersampling dapat dikombinasikan dengan teknik oversampling, seperti Synthetic Minority Oversampling Technique (SMOTE), untuk lebih meningkatkan kinerja model dalam menangani dataset yang tidak seimbang (Nikiforos *et al*., 2023). Pendekatan hibrida ini memanfaatkan kekuatan dari undersampling dan oversampling untuk mencapai model prediksi yang lebih kuat dan akurat.

Singkatnya, undersampling adalah teknik yang berharga dalam pembelajaran mesin untuk mengatasi masalah ketidakseimbangan kelas dalam set data. Dengan mengurangi jumlah instance di kelas mayoritas secara strategis, undersampling membantu menciptakan training set yang lebih seimbang, yang mengarah pada peningkatan kinerja model dan prediksi yang lebih baik untuk instance kelas minoritas.

2. Oversampling

Oversampling adalah teknik yang digunakan dalam pembelajaran mesin untuk mengatasi tantangan dataset yang tidak seimbang, di mana satu kelas kurang terwakili secara signifikan dibandingkan dengan kelas lainnya. Dalam oversampling, contoh dari kelas minoritas diduplikasi atau disintesis untuk meningkatkan representasi mereka dalam dataset, sehingga menyeimbangkan distribusi kelas dan meningkatkan kemampuan model untuk belajar dari kelas minoritas Kunakorntum dkk. (2020).

Salah satu metode oversampling yang banyak digunakan adalah Synthetic Minority Oversampling Technique (SMOTE), yang menghasilkan sampel sintetis dengan melakukan interpolasi di antara contoh kelas minoritas yang ada (Albert *et al*., 2022). SMOTE telah terbukti efektif dalam berbagai aplikasi, seperti deteksi penipuan dan diagnosis medis, di mana set data yang tidak seimbang adalah hal yang umum (Nugraha *et al*., 2022; Golze *et al*., 2020). Selain itu, Adaptive Synthetic Sampling Approach (ADASYN) adalah teknik oversampling lain yang berfokus pada menghasilkan sampel berdasarkan distribusi kepadatan kelas minoritas, yang selanjutnya meningkatkan performa model (Fazry *et al*., 2022).

Oversampling memainkan peran penting dalam meningkatkan kemampuan prediksi model pembelajaran mesin, terutama dalam skenario di mana ketidakseimbangan antar kelas dapat menyebabkan prediksi yang bias. Dengan meningkatkan representasi kelas minoritas, oversampling membantu model belajar lebih efektif dari semua kelas, yang mengarah pada peningkatan akurasi dan generalisasi (Al-Shourbaji *et al*., 2021).

Kesimpulannya, oversampling adalah teknik yang berharga dalam pembelajaran mesin untuk mengatasi masalah ketidakseimbangan kelas dalam set data. Dengan meningkatkan representasi kelas minoritas secara artifisial, oversampling membantu menciptakan set pelatihan yang

lebih seimbang, sehingga memungkinkan model membuat prediksi yang lebih akurat di semua kelas.

3. Cost-Sensitive Learning

Pembelajaran yang peka terhadap biaya adalah pendekatan mendasar dalam pembelajaran mesin yang bertujuan untuk mengatasi tantangan set data yang tidak seimbang dengan mempertimbangkan berbagai biaya yang terkait dengan kesalahan klasifikasi kelas yang berbeda. Dalam skenario di mana satu kelas secara signifikan melebihi jumlah kelas lainnya, algoritme pembelajaran tradisional mungkin kesulitan untuk belajar secara efektif dari kelas minoritas karena dominasi kelas mayoritas. Pembelajaran yang peka terhadap biaya dapat mengurangi masalah ini dengan menetapkan biaya yang berbeda untuk kesalahan klasifikasi, memprioritaskan klasifikasi yang benar dari kelas minoritas, yang mungkin memiliki implikasi atau biaya yang lebih signifikan terkait dengan kesalahan klasifikasi Feng dkk. (2020).

Salah satu metode penting dalam pembelajaran yang peka terhadap biaya adalah penggunaan Support Vector Machines (SVM) dengan pengoptimalan yang peka terhadap biaya, di mana biaya misklasifikasi diintegrasikan ke dalam proses pelatihan SVM untuk mengatasi ketidakseimbangan kelas (Alhakbani & al-Rifaie, 2017). Pendekatan ini memungkinkan model untuk fokus pada meminimalkan total biaya misklasifikasi daripada hanya memaksimalkan akurasi, sehingga lebih cocok untuk dataset yang tidak seimbang (Cao *et al*., 2013).

Kesimpulannya, pembelajaran yang peka terhadap biaya adalah strategi yang berharga dalam pembelajaran mesin untuk menangani set data yang tidak seimbang dengan memasukkan konsep biaya yang bervariasi dari kesalahan klasifikasi. Integrasi ini memungkinkan model untuk mengatasi ketidakseimbangan kelas dengan lebih baik dan membuat keputusan yang lebih tepat, terutama dalam

skenario di mana konsekuensi dari misklasifikasi berbeda di seluruh kelas.

H. Feature Selection

1. Filter Methods

Metode filter dalam pemilihan fitur adalah komponen penting dari algoritma pembelajaran mesin yang bertujuan untuk meningkatkan kinerja model dengan memilih fitur yang paling relevan dari dataset. Metode-metode ini dikategorikan ke dalam tiga kelompok utama: filter, pembungkus, dan teknik tertanam (Lin dkk. (2018), Zhu dkk. (2007).

Metode filter beroperasi dengan mengevaluasi karakteristik intrinsik fitur tanpa melibatkan proses pembelajaran apa pun. Metode-metode ini menilai relevansi fitur berdasarkan ukuran statistik, seperti korelasi atau informasi timbal balik, untuk menentukan pentingnya fitur tersebut dalam memprediksi variabel target (Wang *et al*., 2022; Bommert *et al*., 2021).

Salah satu metode filter yang umum digunakan adalah Fisher score feature selection (FSFS), yang mengurutkan fitur berdasarkan kekuatan diskriminatif dan relevansinya dengan variabel target (Liu *et al*., 2023). Metode filter sangat berguna untuk dataset besar di mana efisiensi komputasi sangat penting, karena metode ini dapat dengan cepat mengidentifikasi fitur yang relevan tanpa memerlukan algoritme pembelajaran yang rumit (Alshamlan, 2021).

Singkatnya, metode filter memainkan peran penting dalam pemilihan fitur dengan secara efisien mengidentifikasi fitur yang paling informatif dalam dataset. Dengan berfokus pada sifat intrinsik fitur, metode filter membantu meningkatkan kinerja model, mengurangi kompleksitas komputasi, dan meningkatkan kemampuan interpretasi model pembelajaran mesin.

I. Memilih Algoritma yang Tepat

1. Contoh Kasus Nyata

Dalam ranah klasifikasi dalam praktik, kasus dunia nyata sering melibatkan penerapan metode feature selection untuk mengoptimalkan performa dan akurasi model. Feature selection adalah langkah krusial dalam machine learning yang melibatkan identifikasi fitur paling relevan dari sebuah dataset untuk meningkatkan proses klasifikasi (Peng *et al*., 2005).

Dalam skenario dunia nyata, penggunaan metode embedded dapat berdampak signifikan pada akurasi klasifikasi dan efisiensi model. Dengan menggabungkan feature selection dalam proses pelatihan, metode embedded menyederhanakan pipeline pengembangan model dan meningkatkan interpretabilitas fitur yang dipilih (Brankovic *et al*., 2018). Integrasi ini memastikan bahwa model fokus pada informasi yang paling relevan, menghasilkan prediksi yang lebih akurat dan peningkatan performa dalam tugas klasifikasi.

Sebagai kesimpulan, metode embedded memainkan peran penting dalam feature selection untuk tugas klasifikasi, menawarkan pendekatan yang mulus dan efektif untuk mengoptimalkan performa model dengan secara otomatis memilih fitur paling relevan selama proses pelatihan. Dengan memanfaatkan metode embedded dalam skenario klasifikasi dunia nyata, praktisi dapat meningkatkan akurasi model, mengurangi overfitting, dan menyederhanakan proses feature selection untuk kemampuan prediktif yang lebih baik.

2. Memilih Algoritma yang Tepat

Memilih algoritma klasifikasi yang tepat untuk sebuah masalah adalah langkah penting dalam proses pengembangan model machine learning. Berikut adalah beberapa panduan yang bisa membantu dalam memilih algoritma klasifikasi yang sesuai:

a. Jenis Data:

1) Data Numerik vs. Kategorikal : Beberapa algoritma bekerja lebih baik dengan data numerik (misalnya, Regresi Logistik, SVM), sementara yang lain lebih cocok untuk data kategorikal (misalnya, Naive Bayes).
2) Data Berlabel vs. Tidak Berlabel : Jika data tidak berlabel, Anda mungkin perlu menggunakan metode unsupervised seperti clustering sebelum menggunakan algoritma klasifikasi.

b. Ukuran Dataset:

1) Dataset Kecil : Algoritma seperti K-Nearest Neighbors (KNN) atau Regresi Logistik bisa menjadi pilihan karena mereka cenderung tidak overfitting pada dataset kecil.
2) Dataset Besar : Algoritma seperti Random Forest, Gradient Boosting, atau Deep Learning (jika datanya sangat besar) lebih efisien dalam memanfaatkan banyak data.

c. Dimensi Data:

1) High Dimensionality : Untuk data dengan banyak fitur, algoritma seperti SVM atau menggunakan teknik dimensionality reduction (misalnya PCA) sebelum menerapkan algoritma klasifikasi bisa membantu.
2) Low Dimensionality : Algoritma seperti Naive Bayes atau Decision Trees seringkali cukup efektif.

d. Interaksi Antar Fitur:

1) Interaksi Kompleks : Algoritma seperti Random Forest atau Neural Networks mampu menangani interaksi yang kompleks antara fitur-fitur.
2) Interaksi Sederhana : Regresi Logistik atau Naive Bayes cocok jika interaksi antar fitur tidak terlalu kompleks.

e. Waktu dan Sumber Daya Komputasi:

1) Waktu Pelatihan Cepat : Algoritma seperti Naive Bayes atau Regresi Logistik biasanya lebih cepat untuk dilatih.
2) Waktu Prediksi Cepat : KNN membutuhkan waktu prediksi yang lebih lama dibandingkan dengan model yang sudah terlatih seperti SVM atau Random Forest.

f. Akurasi vs Interpretabilitas:

1) Akurasi Tinggi : Algoritma kompleks seperti Random Forest, Gradient Boosting, atau Neural Networks biasanya menawarkan akurasi yang lebih tinggi.
2) Interpretabilitas Tinggi : Algoritma seperti Regresi Logistik, Decision Trees, atau Naive Bayes lebih mudah diinterpretasikan.

g. Handling Missing Values:

1) Algoritma yang Toleran terhadap Missing Values : Decision Trees dan Random Forest bisa menangani missing values dengan baik.
2) Algoritma yang Membutuhkan Data Lengkap : SVM dan KNN memerlukan penanganan missing values sebelum pemodelan.

3. Deployment dan monitoring

Deployment dan monitoring adalah dua aspek penting dalam siklus hidup model machine learning, terutama dalam klasifikasi.

Berikut adalah panduan tentang cara melakukannya :

Deployment Model Klasifikasi

a. Pemilihan Lingkungan Deployment:

1) **Cloud (AWS, GCP, Azure**): Platform cloud menawarkan berbagai layanan untuk deployment model, termasuk serverless functions, containerization, dan managed services.
2) **On-premises:** Jika data sensitif atau ada kebijakan perusahaan yang membatasi penggunaan cloud, deployment on-premises bisa menjadi pilihan.

3) **Edge Devices:** Untuk aplikasi yang membutuhkan inferensi real-time dan latensi rendah, seperti perangkat IoT.

b. Format Model:

1) **Serialized Model:** Model bisa disimpan dalam format seperti `pickle` atau `joblib` untuk model Scikit-Learn, atau `h5` untuk model Keras/TensorFlow.
2) **Model Export Formats:** ONNX untuk interoperabilitas antara berbagai framework.

c. API Endpoint:

1) **REST API:** Model di-deploy sebagai layanan web yang menerima input melalui HTTP request dan mengembalikan prediksi sebagai HTTP response.
2) **gRPC:** Alternatif yang lebih cepat untuk REST, terutama untuk komunikasi antar layanan mikro.

d. Containerization:

1) **Docker:** Mengemas model dalam container Docker untuk konsistensi dan portabilitas.
2) **Kubernetes**: Mengelola deployment model dalam skala besar menggunakan Kubernetes untuk orkestrasi container.

e. Serverless Deployment:

1) **AWS Lambda, Google Cloud Functions:** Men-deploy model sebagai fungsi yang hanya berjalan saat diperlukan, mengurangi biaya operasional.

DAFTAR PUSTAKA

Al-Quran Digital Kementerian Agama Republik Indonesia. 2019. : https://quran.kemenag.go.id/

Gaudah, Muhammad Gharib. 2007. 147 Ilmuwan Terkemuka dalam Sejarah Islam.Jakarta: Pustaka Al-Kautsar.

Goodrich, Michael T, dkk. 2014. Data Structures and Algorithms in Java™. Amerika: Don Fowley.

Hariyanto, Eko & Sulistianingsih, Indri. 2019. Dasar Pemrograman Java. Medan : Fakultas Ekonomi Universitas Panca Budi.

Hasan, Nur. 2019. Ulama' Pengembaraan dan Pikiran yang Jernih. Yogyakarta.

Ismah. 2017. Pemrograman Komputer Dasar-dasar Python. Jakarta : Fakultas Ilmu Pendidikan Universitas Muhammadiyah.

Kadir, Abdul. 2014. Buku Pertama Belajar Pemrograman Java untuk Pemula. Yogyakarta: Mediakom.

Stroustrup, Bjarne.Programming principle and practice using C++. Amerika: Penerbit. Pearson education, Inc.2014.

BAB 9 | NEURAL NETWORKS

Samuel Aleksander Mandowen, S.Si., M.IT.

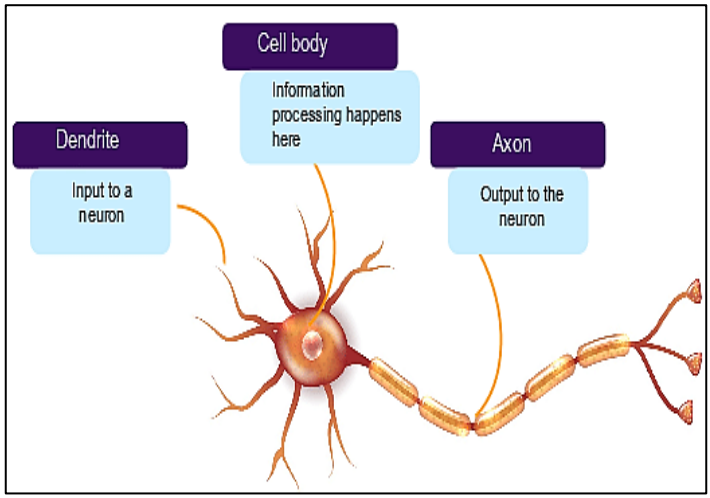
A. Pengantar Jaringan Syaraf

1. Definisi dan Ikhtisar

Jaringan syaraf *(Neural Networks)* adalah model komputasi yang terinspirasi oleh struktur dan fungsi otak manusia. Mereka terdiri dari sejumlah besar unit pemrosesan sederhana, atau "neuron," yang terhubung satu sama lain melalui "sinapsis" atau koneksi. Dalam pembelajaran mesin, jaringan syaraf digunakan untuk mempelajari dan mengenali pola dalam data. Mereka sangat efektif dalam tugas-tugas seperti klasifikasi, regresi, dan pengenalan pola karena kemampuannya untuk memodelkan hubungan yang kompleks antara input dan output (McCulloch and Pitts, 1943) .

2. Inspirasi Biologis

Jaringan syaraf terinspirasi oleh cara kerja otak manusia. Otak terdiri dari miliaran neuron yang saling terhubung melalui sinapsis. Setiap neuron menerima sinyal dari neuron lain, memprosesnya, dan mengirimkan sinyal ke neuron lain. Dalam jaringan syaraf buatan, neuron diwakili oleh unit matematika yang menerima input, mengalikan input tersebut dengan bobot, menambahkan bias, dan menerapkan fungsi aktivasi untuk menghasilkan output. Konsep ini memungkinkan jaringan syaraf untuk belajar dan membuat keputusan berdasarkan data yang diberikan.

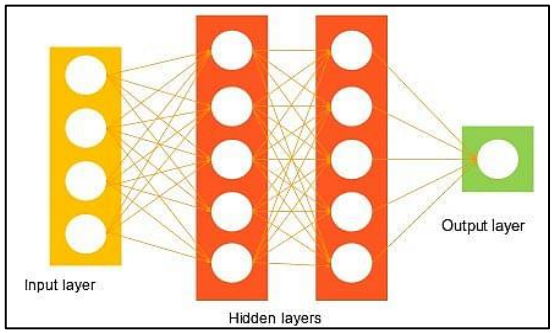


Gambar 9. 1. Cara Kerja Neuron Biologis
Sumber: (Banoula, 2023)

Dendrites: Dendrit: Ini menerima informasi atau sinyal dari neuron lain yang terhubung dengannya.

Cell Body: Badan Sel: Pemrosesan informasi terjadi dalam badan sel. Ini mengambil semua informasi yang berasal dari dendrit yang berbeda dan memproses informasi itu.

Axon: Akson: Mengirimkan sinyal keluaran ke neuron lain untuk aliran informasi. Di sini, masing-masing flensa terhubung ke dendrit atau rambut di flensa berikutnya.



Gambar 9. 2. Bagaimana ANN Mereplikasi Neuron Biologis
Sumber: (Banoula, 2023)

Jaringan dimulai dengan lapisan input yang menerima input dalam bentuk data. Garis-garis yang terhubung ke lapisan tersembunyi disebut bobot, dan garis-garis tersebut bertambah pada lapisan tersembunyi. Setiap titik di lapisan tersembunyi memproses input, dan itu menempatkan output ke lapisan tersembunyi berikutnya dan, terakhir, ke lapisan output. Melihat dua gambar di atas, Anda dapat mengamati bagaimana ANN mereplikasi neuron biologis (Biswal, 2024).

a. Input to a neuron - input layer
b. Neuron - hidden layer
c. Output to the next neuron - output layer

Jaringan saraf adalah sistem perangkat keras atau perangkat lunak yang berpola setelah pengoperasian neuron di otak manusia. Jaringan saraf, juga disebut jaringan saraf tiruan, adalah sarana untuk mencapai pembelajaran yang mendalam. Untuk mengetahui bagaimana fungsi jaringan saraf, perlu dilihat pada arsitektur jaringan saraf.

3. Latar Belakang Sejarah

Sejarah jaringan syaraf dimulai pada tahun 1943 ketika Warren McCulloch dan Walter Pitts mempublikasikan makalah mereka yang berjudul "A logical calculus of the ideas immanent in nervous activity." Dalam makalah ini, mereka mengusulkan model matematika untuk neuron dan menunjukkan bagaimana jaringan neuron dapat melakukan perhitungan logika dasar. Ini adalah tonggak awal dalam pengembangan jaringan syaraf buatan. Pada tahun 1958, Frank Rosenblatt memperkenalkan Perceptron, model jaringan syaraf yang dapat belajar melalui contoh. Kemajuan signifikan lainnya terjadi pada tahun 1986 ketika Rumelhart, Hinton, dan Williams mengembangkan algoritma backpropagation, yang memungkinkan pelatihan jaringan syaraf yang lebih dalam (McCulloch and Pitts, 1943).

Contoh:

Salah satu contoh sederhana dari aplikasi jaringan syaraf adalah pengenalan pola. Misalnya, jaringan syaraf dapat digunakan untuk mengenali tulisan tangan. Dalam kasus ini, gambar dari huruf tulisan tangan diubah menjadi matriks piksel yang digunakan sebagai input untuk jaringan syaraf. Jaringan syaraf kemudian memproses input ini melalui beberapa lapisan neuron dan menghasilkan output yang menunjukkan huruf apa yang dikenali. Proses ini melibatkan pelatihan jaringan dengan banyak contoh gambar huruf yang berbeda untuk meningkatkan akurasi pengenalan.

B. Komponen Dasar Jaringan Syaraf

1. Neuron (Node)

Neuron buatan adalah unit dasar dari jaringan syaraf buatan. Setiap neuron menerima satu atau lebih input, menggabungkan input tersebut dengan bobot yang ditetapkan, menambahkan bias, dan kemudian menerapkan fungsi aktivasi untuk menghasilkan output. Neuron buatan mencoba meniru cara kerja neuron biologis yang memproses dan mengirimkan informasi melalui sinapsis. Secara matematis, neuron dapat dinyatakan dengan persamaan (Ian Goodfellow, Yoshua Bengio, 2017):

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

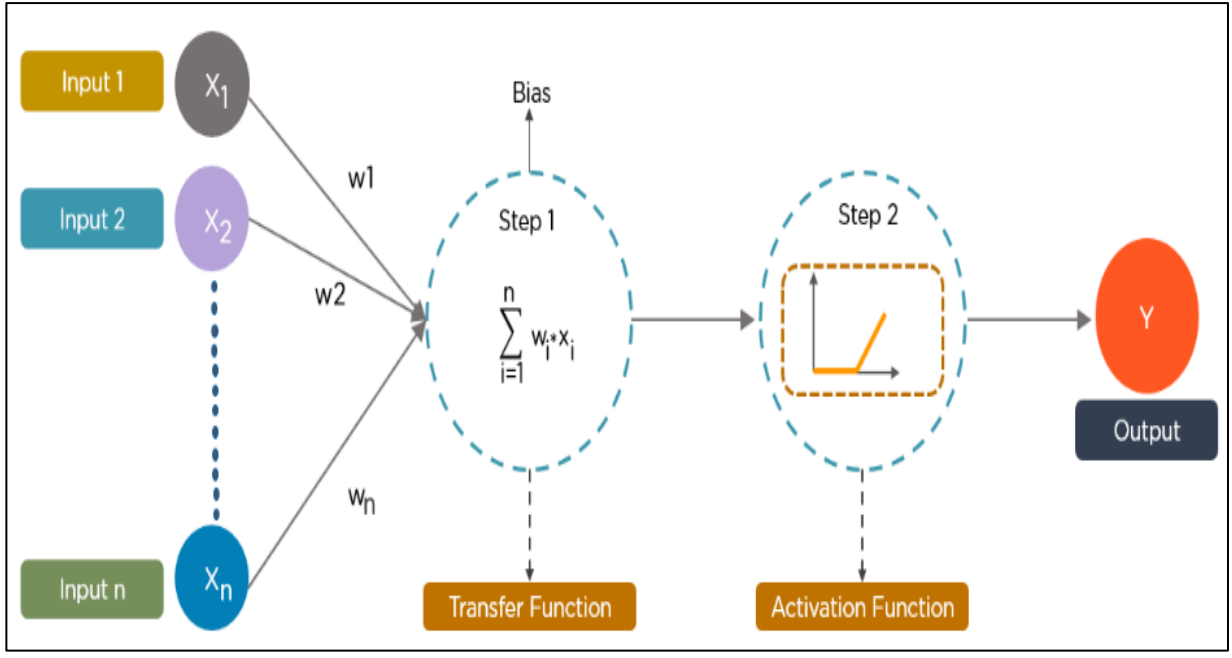
di mana (y) adalah output, (x_i) adalah input, (w_i) adalah bobot, (b) adalah bias, dan (f) adalah fungsi aktivasi.

2. Lapisan

Jaringan syaraf terdiri dari beberapa lapisan neuron yang diatur secara berurutan. Ada tiga jenis lapisan utama:

a. Lapisan Input: Lapisan pertama yang menerima input mentah dari data. Neuron pada lapisan ini hanya bertindak sebagai titik distribusi tanpa melakukan perhitungan.

b. Lapisan Tersembunyi: Satu atau lebih lapisan di antara lapisan input dan output. Lapisan ini melakukan perhitungan kompleks untuk mengenali pola dan fitur dalam data. Neuron pada lapisan ini mengaplikasikan bobot, bias, dan fungsi aktivasi.
c. Lapisan Output: Lapisan terakhir yang menghasilkan prediksi atau keputusan berdasarkan pemrosesan yang dilakukan oleh lapisan tersembunyi. Neuron pada lapisan ini memberikan hasil akhir dari jaringan syaraf.



Gambar 9. 3. Neural Networks Structure

Sumber: (Biswal, 2024)

3. Bobot dan Bias

Bobot dan bias adalah parameter yang dioptimalkan selama pelatihan jaringan syaraf. Bobot menentukan kekuatan koneksi antara neuron. Setiap input (x) dikalikan dengan bobot (w) yang sesuai, dan hasilnya dijumlahkan bersama dengan bias (b). Bias membantu menggeser fungsi aktivasi sehingga jaringan syaraf dapat lebih fleksibel dalam menyesuaikan data. Bobot dan bias diperbarui selama proses pelatihan untuk meminimalkan kesalahan prediksi jaringan.

4. Fungsi Aktivasi

Fungsi aktivasi adalah fungsi matematika yang diterapkan pada output neuron. Fungsi ini memperkenalkan non-linearitas ke dalam jaringan, memungkinkan jaringan untuk mempelajari dan mewakili hubungan yang kompleks.

Beberapa fungsi aktivasi yang umum digunakan adalah:

a. Sigmoid: Menghasilkan output antara 0 dan 1, cocok untuk masalah probabilistik.

$$\boldsymbol{\sigma(x)} = \frac{1}{1 + e^{-x}}$$

b. Tanh (Tangens Hiperbolik): Menghasilkan output antara -1 dan 1, sering digunakan dalam jaringan syaraf rekuren.

$$\mathbf{tanh}\ (\mathbf{x}) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

c. ReLU (Rectified Linear Unit): Menghasilkan output nol untuk input negatif dan identitas untuk input positif, sangat populer dalam jaringan syaraf dalam.

$$\mathbf{ReLU(x)} = \max\,(0, x)$$

Contoh:

Membangun jaringan syaraf sederhana dengan dua neuron input dan satu neuron output:

a. Lapisan Input: Dua neuron menerima input dari data (misalnya, (x_1) dan (x_2).

b. Lapisan Tersembunyi: Tidak ada dalam contoh ini untuk kesederhanaan.
c. Lapisan Output: Satu neuron menghasilkan output (y) berdasarkan kombinasi input:

$$y = \sigma(w_1 x_1 + w_2 x_2 + b)$$

dimana (w_1) dan (w_2) adalah bobot, (b) adalah bias, dan (σ) adalah fungsi aktivasi sigmoid.

C. Jenis-Jenis Jaringan Syaraf

1. Jaringan Syaraf Feedforward (FNNs)

Jaringan Syaraf Feedforward adalah jenis jaringan syaraf yang paling sederhana di mana informasi bergerak hanya dalam satu arah, dari input ke output, tanpa lingkaran balik. Setiap neuron dalam satu lapisan terhubung ke setiap neuron di lapisan berikutnya. Arsitektur ini terdiri dari lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan output. Jaringan syaraf feedforward digunakan dalam berbagai aplikasi seperti klasifikasi gambar, pengenalan suara, dan prediksi waktu ke depan. Mereka sangat baik dalam memecahkan masalah yang dapat dinyatakan dalam bentuk pemetaan statis antara input dan output (Lecun, Bengio and Hinton, 2015).

2. Jaringan Syaraf Konvolusional (CNNs)

Struktur: CNN dirancang khusus untuk pemrosesan data grid seperti gambar. Struktur utama CNN terdiri dari (Lecun, Bengio and Hinton, 2015):

a. Lapisan Konvolusional: Lapisan ini menerapkan filter (atau kernel) ke input untuk menghasilkan peta fitur. Filter ini membantu dalam mendeteksi berbagai fitur seperti tepi, sudut, dan pola.
b. Lapisan Pooling: Lapisan ini mengurangi dimensi peta fitur sambil mempertahankan informasi penting. Max pooling dan average pooling adalah teknik pooling yang umum digunakan.

c. Lapisan Sepenuhnya Terhubung (Fully Connected): Setelah beberapa lapisan konvolusional dan pooling, data diteruskan ke lapisan yang sepenuhnya terhubung untuk membuat keputusan akhir.

CNN banyak digunakan dalam pemrosesan gambar, seperti klasifikasi gambar (misalnya, mengidentifikasi objek dalam gambar), deteksi objek, dan segmentasi gambar. Mereka juga digunakan dalam pemrosesan video dan analisis data medis.

3. Jaringan Syaraf Rekuren (RNNs)

RNN adalah jenis jaringan syaraf yang dirancang untuk memproses data berurutan. Tidak seperti FNNs, RNN memiliki koneksi lingkar balik yang memungkinkan informasi sebelumnya mempengaruhi output saat ini. Ini membuat RNN cocok untuk data yang memiliki urutan temporal atau spasial. LSTM dan GRU adalah RNN tradisional memiliki masalah dengan pelatihan pada urutan panjang karena gradien yang menghilang atau meledak. Untuk mengatasi masalah ini, Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) diperkenalkan. Kedua jenis ini memiliki mekanisme gerbang yang mengontrol aliran informasi dan membantu dalam menangkap ketergantungan jangka panjang. LSTM menggunakan tiga gerbang (input, output, dan forget gate) untuk mengelola memori sel. Sedangkan GRU adalah versi yang lebih sederhana dari LSTM dengan dua gerbang (reset dan update gate). RNN dan variannya digunakan dalam pemrosesan bahasa alami (NLP) seperti penerjemahan mesin, analisis sentimen, dan pengenalan ucapan. Mereka juga digunakan dalam pemodelan deret waktu dan prediksi stok pasar (Lecun, Bengio and Hinton, 2015).

Contoh:

a. Feedforward Neural Networks (FNNs): (Rumelhart, Hinton and Williams, 1986)Klasifikasi email sebagai spam atau bukan spam berdasarkan fitur teks.
b. Convolutional Neural Networks (CNNs): Klasifikasi gambar menggunakan dataset CIFAR-10 untuk mengidentifikasi objek seperti mobil, anjing, dan kucing.
c. Recurrent Neural Networks (RNNs): Penerjemahan teks otomatis dari Bahasa Inggris ke Bahasa Spanyol menggunakan RNN dengan LSTM.

D. Melatih Jaringan Syaraf

1. Propagasi Maju

Propagasi maju adalah proses di mana informasi mengalir melalui jaringan syaraf dari lapisan input ke lapisan output. Pada setiap neuron di lapisan tersembunyi dan lapisan output, input yang diterima dikombinasikan dengan bobot dan bias, kemudian diterapkan fungsi aktivasi untuk menghasilkan output. Proses ini terus berlanjut hingga mencapai lapisan output, di mana jaringan menghasilkan prediksi akhir. Secara matematis, ini bisa diilustrasikan sebagai (Rumelhart, Hinton and Williams, 1986):

$$\textbf{Ouput} = f(W.X + B)$$

dimana (W) adalah matriks bobot, (X) adalah input, (B) adalah bias, dan (*f*) adalah fungsi aktivasi.

2. Fungsi Kerugian

Fungsi kerugian (loss function) adalah metrik yang digunakan untuk mengukur seberapa baik atau buruk jaringan syaraf melakukan tugasnya. Fungsi kerugian mengkuantifikasi perbedaan antara prediksi jaringan dan nilai target yang sebenarnya. Beberapa fungsi kerugian yang umum digunakan adalah (Rumelhart, Hinton and Williams, 1986):

a. **Mean Squared Error (MSE):** Digunakan untuk masalah regresi.

$$\textbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

dimana (y_i) adalah nilai target dan ($\hat{y}_i$) adalah prediksi.

b. **Cross-Entropy:** Digunakan untuk masalah klasifikasi, terutama klasifikasi biner dan multi-kelas.

$$\textbf{Cross Entrapy} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

dimana (y_i) adalah label sebenarnya dan ($\hat{y}_i$) adalah probabilitas yang diprediksi.

3. Propagasi Balik

Propagasi balik (backpropagation) adalah algoritma yang digunakan untuk memperbarui bobot jaringan syaraf. Proses ini melibatkan dua langkah utama (Rumelhart, Hinton and Williams, 1986) :

a. Propagasi Maju: Menghitung output jaringan dan fungsi kerugian.
b. Propagasi Balik: Menghitung gradien dari fungsi kerugian terhadap setiap bobot menggunakan aturan rantai (chain rule) dari kalkulus. Gradien ini kemudian digunakan untuk memperbarui bobot guna meminimalkan fungsi kerugian.

4. Gradient Descent

Gradient descent adalah algoritma optimisasi yang digunakan untuk meminimalkan fungsi kerugian dengan memperbarui bobot jaringan syaraf. Pada setiap iterasi, bobot diperbarui dalam arah yang berlawanan dengan gradien fungsi kerugian. Beberapa varian dari gradient descent meliputi (Rumelhart, Hinton and Williams, 1986):

a. Stochastic Gradient Descent (SGD): Memperbarui bobot menggunakan satu sampel data pada setiap iterasi. Ini membuat proses pelatihan lebih cepat dan mampu menangkap dinamika lokal dari data.

b. Adam (Adaptive Moment Estimation): Kombinasi dari SGD dengan momentum dan RMSprop, yang menjaga rata-rata eksponensial dari gradien pertama dan kedua. Adam adalah salah satu algoritma optimisasi yang paling populer karena konvergensinya yang cepat dan stabil.

$$m_t = \beta_1 m_t + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_t + (1 - \beta_2) g_t^2$$

di mana (g_t) adalah gradien, (m_t) dan (v_t) adalah estimasi pertama dan kedua momen dari gradien, dan (β_1) dan (β_2) adalah parameter eksponensial decay.

Contoh :

Melatih jaringan syaraf untuk tugas klasifikasi biner:

a. Propagasi Maju: Input data dimasukkan ke jaringan, output prediksi dihitung.
b. Menghitung Fungsi Kerugian: Cross-entropy digunakan untuk mengukur kesalahan prediksi.
c. Propagasi Balik: Gradien dari fungsi kerugian dihitung terhadap setiap bobot.
d. Memperbarui Bobot: Bobot diperbarui menggunakan algoritma gradient descent (misalnya, Adam).

E. Teknik Optimasi

Berikut adalah beberapa teknik optimasi (Srivastava *et al.*, 2014) :

1. Tingkat Pembelajaran

Tingkat pembelajaran *(learning rate)* adalah parameter yang menentukan seberapa besar langkah yang diambil dalam mengupdate bobot selama proses pelatihan jaringan syaraf. Tingkat pembelajaran yang terlalu tinggi dapat menyebabkan pelatihan yang tidak stabil dan konvergensi yang gagal karena langkah yang diambil terlalu besar, melampaui titik minimum fungsi kerugian. Sebaliknya, tingkat pembelajaran yang terlalu rendah membuat proses pelatihan menjadi sangat lambat dan mungkin terjebak pada titik minimum lokal. Oleh karena itu, memilih tingkat

pembelajaran yang tepat sangat penting untuk memastikan konvergensi yang cepat dan stabil.

2. Regularisasi

Regularisasi adalah teknik yang digunakan untuk mencegah overfitting dalam jaringan syaraf dengan menambahkan penalti pada bobot besar dalam fungsi kerugian. Beberapa teknik regularisasi yang umum digunakan meliputi:

a. L1 Regularisasi (Lasso): Menambahkan nilai absolut dari bobot ke fungsi kerugian.

$$\textbf{Loss} = \textbf{Original Loss} + \lambda \sum_i |w_i|$$

b. L2 Regularisasi (Ridge): Menambahkan kuadrat dari bobot ke fungsi kerugian.

$$\textbf{Loss} = \textbf{Original Loss} + \lambda \sum_i w_i^2$$

c. Dropout: Teknik di mana selama pelatihan, beberapa neuron dipilih secara acak untuk diabaikan atau "drop out." Ini mencegah jaringan syaraf menjadi terlalu tergantung pada neuron tertentu dan memaksa jaringan untuk menjadi lebih kuat dan lebih general.

3. Normalisasi Batch

Normalisasi batch *(Batch Normalization)* adalah teknik yang menormalkan output dari lapisan sebelumnya dengan menghitung mean dan varians mini-batch selama pelatihan. Ini membantu dalam mempercepat pelatihan jaringan syaraf dengan stabilisasi distribusi output dari lapisan tersembunyi, sehingga memungkinkan tingkat pembelajaran yang lebih tinggi dan mengurangi ketergantungan pada inisialisasi bobot. Normalisasi batch juga berfungsi sebagai regularisasi tambahan.

4. Penyetelan Hiperparameter

Penyetelan hiperparameter *(Hyperparameter Tuning)* adalah proses untuk mencari kombinasi terbaik dari hiperparameter yang memaksimalkan kinerja jaringan syaraf. Beberapa metode yang digunakan dalam penyetelan hiperparameter adalah:

a. Grid Search: Menguji semua kombinasi hiperparameter dalam ruang pencarian yang telah ditentukan.
b. Random Search: Memilih kombinasi hiperparameter secara acak dalam ruang pencarian.
c. Bayesian Optimization: Menggunakan metode probabilistik untuk membangun model dari fungsi kerugian dan mencari hiperparameter optimal berdasarkan model tersebut.

Contoh :

Menerapkan regularisasi dan menyetel hiperparameter untuk kinerja yang lebih baik:

a. Regularisasi: Misalkan kita melatih jaringan syaraf untuk klasifikasi gambar. Kita dapat menambahkan L2 regularisasi untuk mencegah overfitting:

$$\textbf{Loss} = \textbf{Original Loss} + \lambda \sum_i w_i^2$$

b. Dropout: Selama pelatihan, kita menerapkan dropout dengan rate 0.5 pada lapisan tersembunyi untuk mengurangi overfitting.
c. Penyetelan Hiperparameter: Menggunakan Grid Search, kita dapat mencoba berbagai kombinasi tingkat pembelajaran, ukuran batch, dan faktor regularisasi (lambda) untuk menemukan kombinasi yang memberikan kinerja terbaik pada data validasi.

F. Mengevaluasi Jaringan Syaraf

1. Metode Evaluasi

Metode evaluasi digunakan untuk menilai kinerja jaringan syaraf pada tugas tertentu. Beberapa metrik evaluasi yang umum digunakan adalah (Bishop, 2006):

a. **Akurasi**: Persentase prediksi yang benar dari total prediksi.

$$\textbf{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

b. **Presisi**: Proporsi prediksi positif yang benar dari semua prediksi positif.

Dimana, *TP (True Positives), TN (True Negatives), FP (False Positives) dan FN (False Negatives).*

$$\textbf{Presisi} = \frac{TP}{TP + FP}$$

c. **Recall**: Proporsi prediksi positif yang benar dari semua kejadian positif aktual.

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

d. **F1-Score**: Harmonik rata-rata dari presisi dan recall, yang memberikan keseimbangan antara keduanya.

$$\textbf{F1} - \textbf{Score} = 2 \,.\, \frac{Presisi \,.\, Recall}{Presisi + Recall}$$

e. **ROC-AUC (Receiver Operating Characteristic-Area Under Curve):** Metrik yang mengukur kinerja model klasifikasi pada berbagai ambang batas keputusan. AUC memberikan nilai probabilitas bahwa model akan memberi peringkat positif secara lebih tinggi daripada negatif yang acak.

2. Teknik Validasi

Teknik validasi digunakan untuk menilai kinerja model secara obyektif dengan membagi dataset menjadi beberapa subset untuk pelatihan dan pengujian (Bishop, 2006):

a. Cross-validation: Membagi dataset menjadi beberapa lipatan (folds) dan melatih model pada beberapa kombinasi pelatihan dan pengujian. Teknik yang umum adalah k-fold cross-validation, di mana dataset dibagi menjadi k subset, dan model dilatih dan diuji k kali, setiap kali menggunakan subset yang berbeda sebagai data uji dan sisanya sebagai data latih.
b. Train-test split: Membagi dataset menjadi dua bagian, satu untuk pelatihan dan satu lagi untuk pengujian. Biasanya, proporsi yang digunakan adalah 70-80% untuk pelatihan dan 20-30% untuk pengujian.

3. Overfitting dan Underfitting

a. Overfitting: Terjadi ketika model terlalu kompleks dan belajar terlalu detail pada data pelatihan, termasuk noise, sehingga kinerjanya menurun pada data baru. Indikasi overfitting adalah kinerja yang sangat baik pada data pelatihan tetapi buruk pada data validasi atau pengujian.
b. Underfitting: Terjadi ketika model terlalu sederhana untuk menangkap pola dalam data pelatihan. Indikasi underfitting adalah kinerja yang buruk pada data pelatihan dan validasi atau pengujian.

Cara Mengatasi Overfitting dan Underfitting:

a. Menggunakan regularisasi (L1, L2, dropout).
b. Mengumpulkan lebih banyak data pelatihan.
c. Menyederhanakan arsitektur model (mengurangi jumlah lapisan atau neuron).
d. Menggunakan teknik augmentasi data.
e. Menyempurnakan hiperparameter.

Contoh:

Mengevaluasi model jaringan syaraf pada dataset uji:

a. Akurasi: Hitung akurasi model pada dataset uji.
b. Presisi dan Recall: Evaluasi presisi dan recall untuk masing-masing kelas.
c. F1-Score: Hitung F1-Score untuk mendapatkan keseimbangan antara presisi dan recall.

d. ROC-AUC: Plot kurva ROC dan hitung nilai AUC untuk mengevaluasi kinerja keseluruhan model pada berbagai ambang batas keputusan.

G. Topik Lanjutan dalam Jaringan Syaraf

1. Pembelajaran Mendalam

Pembelajaran mendalam (*deep learning*) adalah cabang dari pembelajaran mesin yang menggunakan jaringan syaraf dalam dengan banyak lapisan tersembunyi untuk mengekstrak fitur dan pola kompleks dari data. Pembelajaran mendalam telah mengubah banyak bidang seperti pengenalan suara, visi komputer, dan pemrosesan bahasa alami. Signifikansi pembelajaran mendalam terletak pada kemampuannya untuk belajar representasi data secara otomatis dari data mentah, mengurangi kebutuhan untuk ekstraksi fitur manual (Ian Goodfellow, Yoshua Bengio, 2017).

2. Pembelajaran Transfer

Pembelajaran transfer (*transfer learning*) adalah teknik di mana model yang sudah dilatih pada satu tugas digunakan kembali untuk tugas yang berbeda tetapi terkait. Ini memungkinkan untuk memanfaatkan pengetahuan yang diperoleh dari tugas awal untuk meningkatkan kinerja pada tugas baru, terutama ketika data yang tersedia untuk tugas baru terbatas. Sebagai contoh, model jaringan syaraf yang telah dilatih pada dataset besar seperti ImageNet dapat digunakan sebagai dasar untuk klasifikasi gambar baru dengan menyesuaikan lapisan terakhir (Ian Goodfellow, Yoshua Bengio, 2017).

3. Pencarian Arsitektur Jaringan (NAS)

Pencarian Arsitektur Jaringan (Neural Architecture Search - NAS) adalah metode otomatis untuk merancang arsitektur jaringan syaraf yang optimal. NAS menggunakan teknik seperti pembelajaran penguatan (reinforcement learning) dan algoritma evolusi untuk mencari arsitektur

jaringan yang memberikan kinerja terbaik pada tugas tertentu. NAS mengotomatiskan proses desain jaringan yang biasanya memerlukan keahlian manusia, sehingga memungkinkan pengembangan jaringan syaraf yang lebih efisien dan efektif (Ian Goodfellow, Yoshua Bengio, 2017).

4. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) adalah arsitektur jaringan syaraf yang terdiri dari dua jaringan yang saling berkompetisi: generator dan discriminator. Generator berusaha menghasilkan data yang menyerupai data asli, sementara discriminator berusaha membedakan antara data asli dan data yang dihasilkan oleh generator. Proses ini berlanjut hingga generator menghasilkan data yang sangat mirip dengan data asli. GANs memiliki berbagai aplikasi, termasuk pembuatan gambar, peningkatan resolusi gambar, dan pembuatan data sintetis.

Contoh:

Menerapkan pembelajaran transfer untuk masalah klasifikasi gambar baru:

a. Model Pretrained: Ambil model yang telah dilatih pada dataset besar seperti ImageNet (misalnya, ResNet, VGG).
b. Fine-Tuning: Ganti lapisan terakhir dari model dengan lapisan yang sesuai untuk tugas baru dan latih ulang model dengan data spesifik tugas.
c. Evaluasi: Uji model pada dataset uji untuk memastikan kinerja yang optimal pada tugas baru.

H. Aplikasi Jaringan Syaraf

1. Visi Komputer

Visi komputer melibatkan berbagai tugas yang memungkinkan komputer untuk memahami dan menginterpretasi gambar digital. Beberapa aplikasi utama dari jaringan syaraf dalam visi komputer meliputi (Krizhevsky, Sutskever and Hinton, 2012):

a. Klasifikasi Gambar: Menentukan kategori atau label dari sebuah gambar. Contohnya, jaringan syaraf konvolusional (CNN) digunakan untuk mengklasifikasikan gambar dalam dataset ImageNet.
b. Deteksi Objek: Mengidentifikasi dan melokalisasi objek-objek dalam gambar atau video. Contoh metode yang umum digunakan adalah R-CNN, YOLO, dan SSD.
c. Pembuatan Gambar: Menghasilkan gambar baru yang menyerupai data asli. Contoh teknik ini adalah Generative Adversarial Networks (GANs), yang dapat membuat gambar realistis dari noise acak.

2. Pemrosesan Bahasa Alami (NLP)

Pemrosesan Bahasa Alami (NLP) menggunakan jaringan syaraf untuk memahami dan memanipulasi teks. Beberapa aplikasi utama dalam NLP meliputi (Krizhevsky, Sutskever and Hinton, 2012):

a. Klasifikasi Teks: Mengkategorikan teks ke dalam label yang telah ditentukan sebelumnya, seperti spam atau tidak spam, dan analisis sentimen.
b. Terjemahan Mesin: Menerjemahkan teks dari satu bahasa ke bahasa lain. Model seperti seq2seq dan Transformer (misalnya, BERT, GPT) sangat efektif dalam tugas ini.
c. Analisis Sentimen: Mengidentifikasi sentimen atau emosi dalam teks. Model NLP dapat digunakan untuk menganalisis ulasan produk atau media sosial untuk menentukan apakah sentimen tersebut positif, negatif, atau netral.

3. Pengenalan Suara

Pengenalan suara adalah teknologi yang mengubah ucapan menjadi teks. Jaringan syaraf rekuren (RNN) dan model yang lebih canggih seperti LSTM dan Transformer digunakan untuk menangani urutan data suara. Aplikasi ini termasuk asisten virtual (seperti Siri dan Google Assistant), transkripsi otomatis, dan sistem navigasi suara (Krizhevsky, Sutskever and Hinton, 2012).

4. Pembelajaran Penguatan

Pembelajaran penguatan (reinforcement learning) menggunakan jaringan syaraf untuk mengembangkan agen yang dapat belajar berinteraksi dengan lingkungannya untuk mencapai tujuan tertentu. Dalam pembelajaran penguatan, agen belajar melalui trial and error dengan menerima umpan balik dalam bentuk reward atau penalti. Deep Q-Network (DQN) dan Proximal Policy Optimization (PPO) adalah contoh algoritma yang menggunakan jaringan syaraf dalam pembelajaran penguatan (Krizhevsky, Sutskever and Hinton, 2012).

Contoh:

Menggunakan CNN untuk klasifikasi gambar dan RNN untuk menghasilkan teks:

1. CNN untuk Klasifikasi Gambar:
   a. Dataset: Gunakan dataset seperti CIFAR-10 atau ImageNet.
   b. Model: Bangun dan latih CNN untuk mengklasifikasikan gambar ke dalam berbagai kategori.
   c. Evaluasi: Hitung akurasi model pada dataset uji.
2. RNN untuk Menghasilkan Teks:
   a. Dataset: Gunakan korpus teks seperti karya sastra atau dataset percakapan.
   b. Model: Latih RNN atau LSTM untuk mempelajari pola teks dan menghasilkan teks baru yang koheren.
   c. Evaluasi: Uji model dengan menghasilkan teks baru berdasarkan prompt tertentu.

I. Tren Masa Depan dalam Jaringan Syaraf

1. AI yang Dapat Dijelaskan (XAI)

AI yang Dapat Dijelaskan (Explainable AI - XAI) adalah bidang yang bertujuan untuk membuat keputusan jaringan syaraf lebih dapat dipahami oleh manusia. Seiring dengan meningkatnya penggunaan AI dalam berbagai bidang, transparansi dan interpretabilitas menjadi penting

untuk mendapatkan kepercayaan dan memastikan keadilan dalam keputusan AI. XAI mencakup teknik-teknik seperti visualisasi perhatian, model surrogate, dan analisis kontribusi fitur yang membantu pengguna memahami bagaimana dan mengapa jaringan syaraf menghasilkan keputusan tertentu (Marcus, 2020).

2. **Integrasi Neuro-simbolik**

   Integrasi neuro-simbolik adalah pendekatan yang menggabungkan kekuatan jaringan syaraf dengan metode AI simbolik yang berbasis aturan. Pendekatan ini bertujuan untuk mengatasi keterbatasan jaringan syaraf dalam penalaran logis dan pengetahuan berbasis aturan. Dengan menggabungkan pembelajaran data-driven dari jaringan syaraf dengan kemampuan penalaran dari AI simbolik, sistem neuro-simbolik dapat mencapai kinerja yang lebih baik dan fleksibilitas yang lebih besar dalam memecahkan masalah kompleks (Marcus, 2020).

3. **Jaringan Syaraf Kuantum**

   Jaringan syaraf kuantum mengeksplorasi penggunaan komputasi kuantum untuk meningkatkan kinerja jaringan syaraf. Komputasi kuantum menawarkan potensi untuk mempercepat proses pelatihan dan inferensi dengan memanfaatkan prinsip-prinsip superposisi dan entanglement. Meskipun masih dalam tahap awal penelitian, jaringan syaraf kuantum diharapkan dapat menangani masalah yang sangat kompleks dan besar yang tidak dapat diatasi oleh komputer klasik (Marcus, 2020).

4. **AI Berkelanjutan**

   AI berkelanjutan (Sustainable AI) berfokus pada mengurangi dampak lingkungan dari pelatihan dan penggunaan jaringan syaraf. Pelatihan model AI yang besar memerlukan banyak energi dan sumber daya komputasi, yang berdampak pada lingkungan. Penelitian dalam AI berkelanjutan mencakup pengembangan model yang lebih efisien, optimisasi penggunaan sumber daya, dan penerapan

teknik penghematan energi untuk meminimalkan jejak karbon dari aplikasi AI (Marcus, 2020).

Contoh:

Penelitian terkini dan perkembangan potensial di masa depan:

1. **AI yang dapat Dijelaskan**

   Penelitian: Pengembangan teknik interpretabilitas seperti SHAP (SHapley Additive exPlanations) dan LIME (Local Interpretable Model-agnostic Explanations).

   Aplikasi: Menerapkan XAI dalam bidang kesehatan untuk memastikan keputusan diagnosis AI dapat dipahami oleh dokter.

2. **Integrasi Neuro-simbolik**

   Penelitian: Mengembangkan sistem hibrida yang menggabungkan jaringan syaraf dengan aturan logika untuk tugas penalaran kompleks.

   Aplikasi: Penggunaan dalam sistem pengetahuan yang memerlukan pemahaman mendalam dan penalaran berbasis konteks.

3. **Jaringan Syaraf Kuantum**

   Penelitian: Eksperimen dengan algoritma kuantum seperti Variational Quantum Eigensolver (VQE) untuk mengoptimalkan jaringan syaraf.

   Aplikasi: Menjelajahi aplikasi dalam pengenalan pola dan optimisasi yang memerlukan komputasi intensif.

4. **AI Berkelanjutan**

   Penelitian: Mengembangkan model AI yang lebih kecil dan efisien yang mempertahankan kinerja tinggi. Aplikasi: Menerapkan teknik kompresi model dan distilasi pengetahuan untuk mengurangi kebutuhan komputasi dan energi.

DAFTAR PUSTAKA

Banoula, M. (2023) *What is Neural Network: Overview, Applications, and Advantages*. Available at: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-neural-network (Accessed: 4 March 2023).

Bishop, C.M. (2006) *Pattern Recognition and Machine Learning*. Springer New York. Available at: https://doi.org/10.1007/978-0-387-45528-0.

Biswal, A. (2024) *Top 10 Deep Learning Algorithms You Should Know in 2024*. Available at: https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm (Accessed: 24 May 2024).

Ian Goodfellow, Yoshua Bengio, A.C. (2017) 'Deep Learning', *MIT Press*, 521(7553), p. 785. Available at: https://doi.org/10.1016/B978-0-12-391420-0.09987-X.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'ImageNet Classification with Deep Convolutional Neural Networks'. Available at: http://code.google.com/p/cuda-convnet/ (Accessed: 24 May 2024).

Lecun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436–444. Available at: https://doi.org/10.1038/nature14539.

Marcus, G. (2020) 'The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence', (February). Available at: http://arxiv.org/abs/2002.06177.

McCulloch, W.S. and Pitts, W. (1943) 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics*, 5(4), pp. 115–133. Available at: https://doi.org/10.1007/BF02478259.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) 'Learning representations by back-propagating errors', *Nature*, 323(6088), pp. 533–536. Available at: https://doi.org/10.1038/323533a0.

Srivastava, N. *et al.* (2014) 'Dropout: A simple way to prevent neural networks from overfitting', *Journal of Machine Learning Research*, 15, pp. 1929–1958.

BAB 10 | *ENSEMBLE METHOD*

Hajiar Yuliana, S.T., M.T.

A. Pengantar *Enesemble Method*

Dalam beberapa tahun terakhir, metode *ensemble* telah menjadi tren yang semakin populer dalam bidang data mining dan machine learning. Metode *ensemble* merupakan pendekatan yang menggabungkan beberapa model prediksi untuk meningkatkan akurasi dan kinerja keseluruhan. Teknik ini telah terbukti efektif dalam mengatasi berbagai tantangan yang dihadapi oleh model individu, seperti *overfitting* dan variabilitas hasil.

Salah satu tren yang sedang muncul adalah penggunaan *ensemble method* dalam aplikasi yang lebih kompleks dan dinamis, seperti pengolahan bahasa alami (NLP), computer vision, dan prediksi berbasis data besar (big data). Kemajuan dalam komputasi paralel dan teknologi cloud computing juga telah memungkinkan penerapan *ensemble method* pada skala yang lebih besar, membuatnya lebih aksesibel dan efisien untuk digunakan dalam berbagai industri.

Masa depan *ensemble method* tampak sangat menjanjikan, dengan potensi pengembangan lebih lanjut dalam hal otomatisasi dan integrasi dengan teknik-teknik terbaru seperti deep learning dan reinforcement learning. Selain itu, peningkatan dalam algoritma *ensemble* seperti *Random forest*, Gradient *Boosting*, dan *Stacking*, terus mendorong batas kinerja model prediksi.

Buku "Data Mining" memberikan landasan yang kuat dalam memahami konsep dan teknik dasar yang mendasari *ensemble method*. Dalam konteks ini, konten buku ini sangat relevan karena tidak hanya menjelaskan teori di balik metode *ensemble*, tetapi juga menyediakan panduan praktis tentang cara mengimplementasikan dan mengoptimalkan teknik-teknik ini. Buku ini juga mencakup studi kasus dan contoh aplikasi nyata yang menggambarkan efektivitas metode *ensemble* dalam berbagai domain, dari analisis bisnis hingga ilmu kesehatan.

Dengan memahami tren dan prospek menarik dari *ensemble method* melalui buku "Data Mining," pembaca dapat memperoleh wawasan yang mendalam tentang bagaimana memanfaatkan teknik ini untuk meningkatkan kinerja model prediksi mereka dan mempersiapkan diri menghadapi tantangan data mining di masa depan.

Metode *ensemble* dalam data mining melibatkan pembangunan sekumpulan pengklasifikasi yang secara kolektif membuat prediksi pada titik-titik data baru melalui mekanisme pemungutan suara (Dietterich, 2000). Metode-metode ini telah banyak dipelajari dan diterapkan di berbagai bidang, termasuk kecerdasan buatan, biologi, genomik, dan kedokteran (Hubbard *et al*., 2009; Rokach, 2009; Pourhoseingholi, Kheirian and Zali, 2017; Verma, Pal and Kumar, 2019). Metode ensembel telah menunjukkan keefektifan dalam mengatasi tantangan seperti *overfitting* dan dimensi yang tinggi pada dataset (Rokach, 2009; Pourhoseingholi dkk., 2017).

Para peneliti telah mengeksplorasi penggunaan metode *ensemble* dalam konteks yang berbeda, seperti prediksi kinerja akademik, klasifikasi penyakit kulit, dan prediksi tiroid, yang menunjukkan keserbagunaan dan penerapan teknik-teknik ini (Yusuf & John, 2019; Yadav & Pal, 2019; Verma *et al*., 2019). Selain itu, metode ensembel telah digunakan dalam menangani penyimpangan konsep, meningkatkan akurasi prediksi data, dan mencapai waktu pembaruan yang lebih cepat dibandingkan dengan pendekatan tradisional (Almeida *et al*., 2019; Brzeziński & Stefanowski, 2014).

Metode *ensemble* juga telah dibandingkan dengan metode data mining dasar dalam berbagai penelitian, yang menunjukkan kinerja superior mereka dalam memprediksi hasil seperti kelangsungan hidup 5 tahun pasien kanker kolorektal (Pourhoseingholi *et al*., 2017). Selain itu, metode *ensemble* telah digunakan dalam penggalian data lingkungan, analisis pengelompokan, dan klasifikasi sentimen, yang menunjukkan kemampuan beradaptasi mereka pada berbagai bidang penelitian (Tuysuzoglu *et al*., 2018; Fu *et al*., 2022; Rabiu *et al*., 2023).

Pentingnya metode ensembel terletak pada kemampuannya untuk meningkatkan akurasi prediksi, menangani set data yang kompleks, dan beradaptasi dengan pola data yang terus berkembang (Dietterich, 2000; Brzeziński & Stefanowski, 2014). Dengan memanfaatkan kecerdasan kolektif dari beberapa pengklasifikasi, metode *ensemble* menawarkan kerangka kerja yang kuat untuk meningkatkan keandalan dan efisiensi tugas-tugas penggalian data di berbagai domain.

B. Konsep Dasar *Ensemble Method*

Metode *ensemble* dalam data mining, termasuk konsep-konsep seperti pohon klasifikasi dan regresi, *bagging*, dan hutan acak, telah mendapatkan perhatian yang signifikan untuk meningkatkan kinerja prediktif dengan menggabungkan beberapa model (Strobl, Malley and Tutz, 2009; Sagi and Rokach, 2018). Metode-metode ini, seperti evolutionary undersampling *boosting*, sangat berguna dalam mengatasi tantangan klasifikasi yang tidak seimbang, seperti yang terlihat pada aplikasi seperti klasifikasi keganasan kanker payudara (Krawczyk *et al*., 2016). Selain itu, teknik seperti random subspace regression *ensemble* telah efektif dalam tugas-tugas seperti kalibrasi spektroskopi inframerah-dekat, yang menunjukkan keserbagunaan dan kegunaan metode *ensemble* di berbagai domain (Tan, Li and Xin, 2008).

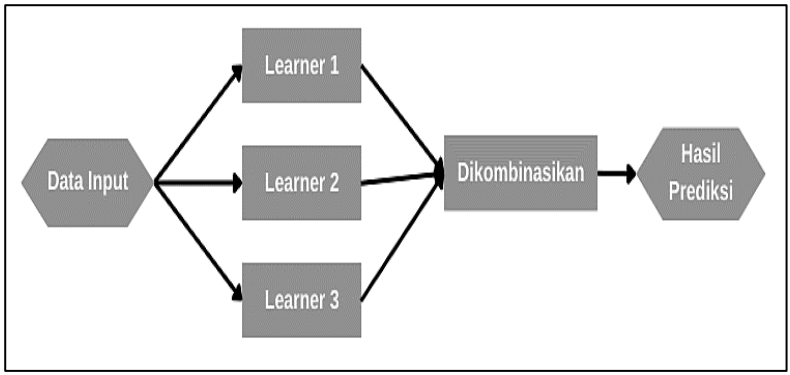
Dengan memanfaatkan kekuatan dari beberapa model dan menggabungkan teknik seperti undersampling dan regresi subruang acak, metode ensembel menawarkan kerangka kerja yang kuat untuk meningkatkan akurasi prediktif dan mengatasi tantangan penggalian data yang kompleks dalam berbagai aplikasi.

Ensemble learning adalah sebuah pendekatan dalam machine learning di mana beberapa model (disebut *ensemble*) digabungkan untuk memecahkan masalah yang sama dengan tujuan meningkatkan akurasi dan kinerja prediksi. Teknik-teknik yang digunakan dalam *ensemble* learning dikenal sebagai *ensemble method*s. Kedua konsep ini saling melengkapi dan memiliki peran penting dalam meningkatkan hasil analisis data mining.

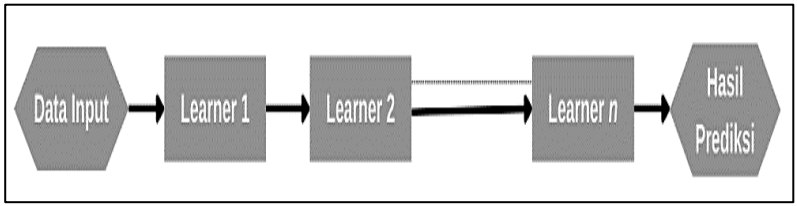
Ensemble learning didasarkan pada prinsip bahwa kombinasi beberapa model dapat menghasilkan prediksi yang lebih baik daripada model individu. Ide utama di balik *ensemble* learning adalah bahwa berbagai model akan menangkap berbagai aspek dari data dan kesalahan dari model yang satu dapat dikompensasi oleh model lainnya. Ini meningkatkan akurasi keseluruhan dan membuat prediksi lebih andal.

Ensemble learning biasanya diterapkan dalam dua cara utama:

1. ***Parallel Ensemble Learning***: Model-model dilatih secara paralel dan hasilnya digabungkan. Contohnya adalah metode *Bagging* (Bootstrap Aggregating). Prosesnya digambarkan dalam bentuk diagram di Gambar 10.1
2. ***Sequential Ensemble Learning***: Model-model dilatih secara berurutan, di mana model berikutnya mencoba untuk memperbaiki kesalahan yang dibuat oleh model sebelumnya. Contohnya adalah metode *Boosting*. Prosesnya digambarkan dalam bentuk diagram di Gambar 10.2



Gambar 10. 1. Blok Diagram Parallel *Ensemble Learning*



Gambar 10. 2. Blok Diagram Sequential *Ensemble Learning*

Terkait pembelajaran *ensemble* (*ensemble* learning) tersebut, diperlukan metode untuk dapat mengerjakan atau melakukan pembelajaran *ensemble* learning. Metode yang dimaksud tersebut adalah *ensemble method*.

*Ensemble method*s adalah teknik-teknik spesifik yang digunakan untuk mengimplementasikan *ensemble* learning. Beberapa metode *ensemble* yang paling populer meliputi ***bagging***, ***boosting***, dan ***stacking (stacked generalization).*** Secara rinci, hal ini akan dibahas lebih lanjut di sub bab selanjutnya.

Dalam konteks data mining, *ensemble* learning dan *ensemble method*s memberikan beberapa keuntungan penting, diantaranya adalah :

1. **Untuk Meningkatkan Akurasi**

   Peningkatan akurasi ini didapatkan karena dilakukan penggabungan beberapa model, sehingga dapat membantu dalam mengurangi kesalahan prediksi dan meningkatkan akurasi keseluruhan.

2. **Stabilitas dan Robustness**

   Ensemble methods mengurangi variabilitas hasil yang mungkin terjadi karena model individu rentan terhadap *overfitting* dan noise dalam data.

3. **Fleksibilitas**

   Berbagai metode *ensemble* dapat digunakan untuk menangani berbagai jenis data dan masalah prediksi yang berbeda, dari klasifikasi hingga regresi.

Ensemble learning dan *ensemble methods* adalah alat yang sangat kuat dalam data mining yang memungkinkan para peneliti dan praktisi untuk meningkatkan kinerja model prediksi mereka. Dengan memahami dan mengimplementasikan teknik-teknik ini, pembaca dapat memanfaatkan potensi penuh dari data mereka dan membuat keputusan yang lebih tepat dan andal.

C. Teknik-Teknik dalam *Ensemble Method*

Metode *ensemble* adalah teknik yang menggabungkan beberapa model pembelajaran untuk menghasilkan prediksi yang lebih akurat dan andal dibandingkan dengan model individu. Beberapa teknik utama dalam *ensemble method* meliputi *Bagging*, *Boosting*, *Stacking*, dan Voting. Mari kita bahas masing-masing teknik tersebut secara lebih mendetail.

1. ***Bagging* (*Bootstrap Aggregating*)**

   Bagging adalah salah satu teknik *ensemble* yang paling sederhana dan populer. Teknik ini melibatkan pembuatan beberapa subset data dari dataset asli melalui proses bootstrap sampling (pengambilan sampel dengan pengembalian). Setiap subset kemudian digunakan untuk melatih model yang berbeda, dan hasil prediksi dari semua model tersebut digabungkan dengan cara rata-rata (untuk regresi) atau voting mayoritas (untuk klasifikasi).

2. ***Boosting***

   Boosting adalah teknik *ensemble* yang berfokus pada meningkatkan kinerja model dengan melatih model secara berurutan, dimana setiap model baru mencoba untuk memperbaiki kesalahan yang dibuat oleh model sebelumnya. Model akhir adalah kombinasi dari semua model yang dilatih dengan memberikan bobot lebih pada model yang lebih akurat.

3. ***Stacking* (*Stacked Generalization*)**

   Stacking adalah teknik *ensemble* yang menggabungkan prediksi dari beberapa model dasar menggunakan model meta (meta-*learner*). Model dasar dilatih menggunakan dataset asli, dan prediksi mereka digunakan sebagai input untuk melatih model meta. Model meta kemudian belajar untuk membuat prediksi akhir yang lebih baik.

4. **Voting**

   Voting adalah teknik *ensemble* yang menggabungkan prediksi dari beberapa model dengan menggunakan metode voting. Ada dua jenis voting: hard voting dan soft voting. Hard voting menggabungkan prediksi berdasarkan mayoritas suara, sedangkan soft voting menggabungkan probabilitas prediksi dari setiap model dan memilih kelas dengan probabilitas tertinggi.

Metode *ensemble* telah terbukti efektif dalam meningkatkan akurasi dan keandalan prediksi model. Dengan memahami dan mengaplikasikan teknik-teknik seperti *Bagging*, *Boosting*, *Stacking*, dan Voting, praktisi data mining dapat mengatasi berbagai tantangan dan meningkatkan kinerja model mereka. Buku "Data Mining" menyediakan landasan yang kuat untuk memahami dan mengimplementasikan teknik-teknik ini, memberikan panduan praktis dan studi kasus yang relevan.

D. *Bagging*: Penguatan Model dengan Bootstrap Aggregating

Belajar membuat program tentu tidak jauh-jauh dari yang namanya struktur dasar bahasa pemrograman, logika dasar pemrograman, algoritma pemrograman dan lain sebagainnya. Setiap bahasa pemrograman memiliki struktur dasar bahasa pemrograman yang berbeda-beda, tetapi konsep dasar pemrogramannya sama dengan menggunakan bahasa pemrograman berbeda-beda.

Baging, juga dikenal sebagai *Bootstrap Aggregating* adalah salah satu teknik grup yang paling mudah dan populer dalam data mining dan pembelajaran mesin. Metode ini mengurangi variabilitas (*variance*) pada model individu untuk meningkatkan akurasi dan stabilitas model prediksi. *Bagging* melibatkan sampling dengan pengembalian untuk membuat beberapa versi dari dataset asli, melatih model untuk masing-masing subset data, dan kemudian menggabungkan hasil prediksi dari semua model tersebut. *Bagging* merupakan teknik *ensemble* paralel yang menggunakan metode *bootstrap* sampling untuk membuat beberapa subset data dari dataset asli.

Dengan *bootstrap* sampling, setiap subset data digunakan untuk melatih model yang berbeda, dan hasil prediksi dari model-model ini digabungkan untuk menghasilkan prediksi akhir yang lebih akurat dan andal.

Bootstrap aggregating (*bagging*) dikembangkan pada tahun 1994 oleh Breiman (Richman and Wüthrich, 2020) untuk meningkatkan performa klasifikasi model ML dengan menggabungkan prediksi dari training set yang dihasilkan secara acak. Pada dasarnya, metode *bagging* melibatkan pemisahan data pelatihan untuk setiap pembelajar dasar (*base learner*) menggunakan pengambilan sampel acak untuk menghasilkan sejumlah *b* himpunan bagian yang berbeda yang digunakan untuk melatih *b* pembelajar dasar. Sekian banyak base *learner* tersebut kemudian digabungkan menggunakan voting mayoritas untuk mendapatkan pengklasifikasi yang kuat (Li and Song, 2022).

Bagging akan lebih meningkatkan performa pembelajaran dasar (*base learner*) jika algoritma yang digunakan dalam mempelajari model tidak stabil. Algoritma yang tidak stabil secara signifikan mengubah kemampuan generalisasinya ketika ada sedikit modifikasi pada inputnya. *Bagging* lebih berfokus pada pengurangan varians dalam anggota *ensemble* daripada bias. Oleh karena itu, *bagging* bekerja secara optimal ketika anggota *ensemble* memiliki varians yang tinggi dan bias yang rendah. Contoh dari algoritma yang tidak stabil adalah *decision tree*; oleh karena itu, *decision tree* yang di-*bagging* biasanya berkinerja lebih baik daripada pohon keputusan tunggal. *decision tree* yang di-*bagging* ini adalah awal mula dari munculnya *Random forest*.

Sementara itu, *k-nearest Neighbor* (KNN) dan *naïve Bayes* adalah contoh algoritma yang stabil, dan *bagging* tidak bekerja dengan baik dengan algoritma ini sebagai pembelajar dasar.(Oza and Tumer, 2008)

Random forest merupakan salah satu contoh dari algoritma *bagging* yang cukup terkenal dan sering digunakan dalam berbagai prediksi. *Random forest* menggunakan pohon keputusan (*decision trees*) sebagai model dasar dan menggabungkannya untuk membuat prediksi yang lebih kuat dan andal. Teknik *bagging* yang digunakan menghasilkan sampel acak dengan penggantian dari data input dan melatih pohon keputusan (*decision trees*) dari sampel tersebut (Caie, Dimitriou and Arandjelović, 2021). *Decision trees* merupakan komponen utama dalam algoritma *random forest* (Suthaharan, 2016). Sementara itu, algoritma *random forest* ini pertama kali dikembangkan pada tahun 1995 oleh Ho [108] dengan menggunakan metode *random subspace*, dan pada tahun 2001 Breiman (Jin *et al*., 2020) mengembangkan versi yang lebih luas dari algoritma ini.

Algoritma *random forest* telah banyak diaplikasikan untuk banyak penelitian dan studi karena mudah diimplementasikan, cepat, dan mendapatkan kinerja yang sangat baik. Dua aspek penting dari algoritma *random forest* adalah pengembangan

beberapa pohon keputusan selama pelatihan dan kombinasi prediksi mereka menggunakan voting mayoritas. Karena pohon keputusan rentan terhadap *overfitting*, pendekatan voting meminimalkan peluang *random forest* untuk *overfitting* (Caie, Dimitriou and Arandjelović, 2021).

Lebih lanjut, algoritma ini menggunakan *bagging* dan fitur yang acak dalam membangun *forest* dalam *decision tree* yang tidak berkorelasi. Sementara itu, keacakan fitur dicapai dengan menggunakan metode *random subspace* yang memastikan fitur-fitur dipilih secara acak untuk melatih setiap pohon keputusan di dalam hutan (Mrabah, Bouguessa and Ksantini, 2023). Korelasi antara pohon-pohon keputusan (*decision tree*) yang membentuk *forest* berkurang karena pohon-pohon tersebut dilatih menggunakan subset fitur acak dan bukannya seluruh set fitur.

Berikut ini merupakan beberapa kelebihan dan kekurangan dari teknik *Bagging*.

Kelebihan:

1. **Mengurangi Variabilitas** : Guna meredakan variabilitas yang disebabkan oleh model yang mungkin overfit terhadap data, *bagging* digunakan untuk menggabungkan beberapa model yang dilatih pada subset data yang berbeda.
2. **Meningkatkan Akurasi** : prediksi dari hasil penjumlahan beberapa model selalu lebih akurat dari model tunggal.
3. **Sederhana dan Mudah Diimplementasikan** : Mudah diimplementasikan dan tidak memerlukan banyak parameter untuk disetel.

Kekurangan:

1. **Komputasi yang Lebih Besar** : Mengharuskan pemodelan beberapa keturunan, model yang lebih banyak memerlukan lebih banyak pekerjaan grid tersebut dimasukkan ke dalam perhitungan, memang memakan waktu dan sumber daya tervalidasi.

2. **Tidak Selalu Meningkatkan Kinerja Model** : *bagging* pergi dengan berbagai cara ketika digunakan dengan yang merupakan bentuk yang relative tinggi dari beberapa model individu. Dengan demikian, jika model dasar sudah stabil, kinerja yang lebih lebih kecil.

E. *Boosting*: Meningkatkan Kinerja Model Secara Bertahap

Boosting adalah salah satu teknik *ensemble* yang sangat efektif dalam meningkatkan kinerja model prediksi. Teknik ini bekerja dengan menggabungkan beberapa model lemah (weak *learner*s) secara berurutan untuk membentuk model yang lebih kuat. Setiap model baru berfokus pada kesalahan yang dibuat oleh model sebelumnya, sehingga secara bertahap meningkatkan akurasi prediksi keseluruhan.

Boosting didasarkan pada ide bahwa kombinasi dari beberapa model lemah dapat menghasilkan model yang lebih kuat dan akurat. Berbeda dengan *bagging* yang bekerja secara paralel, *boosting* bekerja secara berurutan. Setiap model dilatih untuk memperbaiki kesalahan dari model sebelumnya dengan memberi bobot lebih pada data yang sulit diprediksi.

Terdapat beberapa algoritma *boosting* yang populer dan banyak digunakan dalam praktek, antara lain AdaBoost dan Gradient *Boosting*.

1. AdaBoost

AdaBoost adalah salah satu algoritma *boosting* yang paling sederhana dan populer. Algoritma ini berfungsi dengan memberi bobot lebih pada kesalahan yang sulit diprediksi sehingga model berikutnya lebih fokus pada data tersebut.

AdaBoost ini merupakan algoritma *boosting* yang mudah diimplementasikan dan tidak memerlukan penyesuaian parameter yang banyak serta cukup efektif dalam meningkatkan kinerja model sederhana. Akan tetapi, AdaBoost sangat rentan terhadap noise dan outliers dalam data. Hal ini menyebabkan menurunnya performansi kinerja algoritma dan keakuratan dari prediksi yang dihasilkan.

2. Gradient *Boosting*

Gradient *Boosting* adalah algoritma yang lebih kompleks yang menggunakan pendekatan gradient descent untuk mengoptimalkan model. Setiap model baru dilatih untuk memperbaiki residu (kesalahan) dari model sebelumnya. Gradient *Boosting* sangat fleksibel dan dapat menangani berbagai jenis data dan masalah prediksi. Selain itu, algoritma ini dapat diatur dengan baik agar menghasilkan kinerja yang optimal. Akan tetapi, algoritma ini rentan terhadap *overfitting* jika tidak diatur dengan baik. Selain itu, Gradient *Boosting* masih memerlukan waktu komputasi yang lebih lama. Hal ini disebabkan karena proses iteratifnya yang cukup memakan banyak waktu.

Selain kedua algoritma *boosting* tersebut, saat ini sudah cukup banyak algoritma baru yang dihasilkan dari proses *boosting*, diantaranya adalah XGBoost, LightGBM (*Light Gradient Boosting Machine*), dan *CatBoost*.

Boosting adalah teknik *ensemble* yang sangat efektif dalam meningkatkan kinerja model prediksi secara bertahap dengan memfokuskan pada kesalahan yang dibuat oleh model sebelumnya. Algoritma seperti *AdaBoost* dan *Gradient Boosting* memberikan kerangka kerja yang kuat untuk memanfaatkan kekuatan beberapa model lemah untuk membentuk model yang lebih kuat. Dengan memahami dan mengimplementasikan teknik *boosting*, praktisi data mining dapat meningkatkan akurasi dan kinerja model mereka, membuat prediksi yang lebih tepat dan dapat diandalkan.

F. *Stacking*: Menggabungkan Prediksi Model untuk Kinerja Lebih Baik

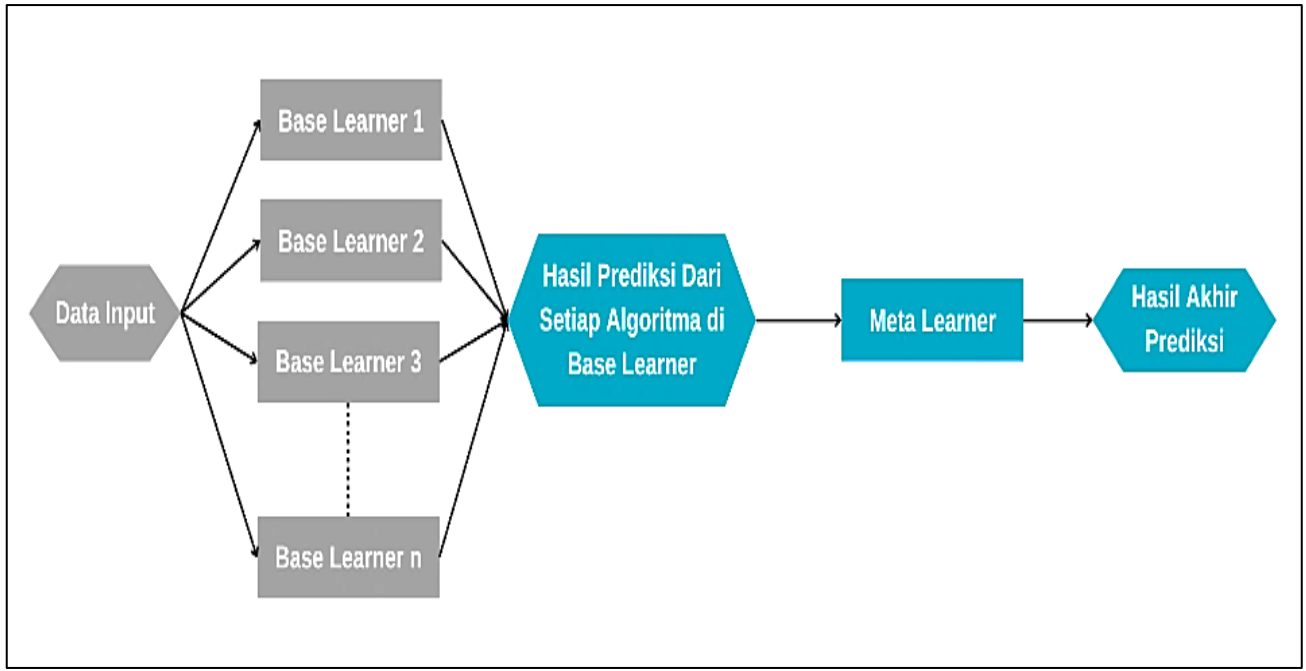
Stacking, atau *Stacked Generalization*, adalah salah satu teknik *ensemble* yang digunakan untuk meningkatkan kinerja model prediksi dengan menggabungkan beberapa model dasar (*base models*) menggunakan model meta (meta-*learner*). Berbeda dengan teknik *ensemble* lainnya seperti *bagging* dan *boosting*, *stacking* memungkinkan penggunaan model yang heterogen dan

menggabungkan kekuatan masing-masing model untuk menghasilkan prediksi yang lebih akurat dan andal.

Stacking diperkenalkan pada tahun 1992 oleh Wolpert (Wolpert, 1992) untuk mengurangi kesalahan generalisasi dalam masalah pembelajaran mesin. *Stacking* berguna dalam situasi di mana beberapa model ML memiliki keahlian yang unik dalam tugas tertentu; kemudian, pendekatan *stacking* akan menggunakan model ML yang terpisah untuk mempelajari kapan harus menggunakan prediksi dari berbagai model (Witten and Frank, 2005).

Secara khusus, *stacking* melibatkan pembuatan model menggunakan beberapa algoritma dasar, yang disebut model level-0, dan algoritma meta-learning yang melatih model lain untuk menggabungkan prediksi dari model dasar. Meta-model ini disebut sebagai model level-1 (Polikar, 2012). Ide inti dalam *stacking* adalah bahwa pembelajar dasar level-0 dilatih menggunakan dataset pelatihan dan diberikan data di luar sampel atau data yang tidak terlihat; label target yang diprediksi pada data yang tidak terlihat, bersama dengan label yang sebenarnya, membentuk pasangan input dan output dataset baru yang digunakan untuk melatih meta-*learner* (Liang *et al*., 2021).

Meta-learning adalah bagian dari pembelajaran mesin di mana algoritma dilatih menggunakan output dari algoritma ML lainnya dan membuat prediksi yang lebih akurat mengingat prediksi yang dibuat oleh pengklasifikasi dasar lainnya (Hospedales *et al*., 2022). Seperti yang ditunjukkan pada Gambar 10.3, meta-*learner* adalah bagian integral dari kerangka kerja *stacking* karena ia melatih model yang membuat prediksi akhir.



Gambar 10. 3. Blok Diagram Kerangka Kerja Konsep *Stacking*

Meta-pengklasifikasi mempelajari cara terbaik untuk menggabungkan prediksi-prediksi dari para pembelajar dasar. Metode *stacking* sangat kuat karena menggunakan kekuatan dari beberapa pengklasifikasi yang berkinerja baik untuk membuat klasifikasi yang lebih unggul daripada model individu yang membentuk *ensemble*.

Selain itu, *stacking* menggunakan algoritma dasar yang berbeda dan dataset yang sama untuk mendapatkan model yang beragam dan mendekati masalah pemodelan prediktif secara berbeda. Tidak seperti *bagging*, yang terutama menggunakan model pohon keputusan yang dilatih pada subset dari data input, model yang ditumpuk menggunakan algoritma yang berbeda dan dilatih pada set data yang sama.

Teknik *stacking* ini pun juga tidak seperti *boosting*, yang secara berurutan melatih model untuk mengoreksi prediksi dari model sebelumnya, *stacking* menggunakan model tunggal untuk mempelajari cara menggabungkan prediksi dari pembelajar dasar secara optimal. Sementara itu, meta-model biasanya sederhana karena belajar dari prediksi yang dibuat oleh model level-0. Oleh karena itu, pengklasifikasi linier, seperti regresi logistik, sering digunakan sebagai meta-*learner*. Akan tetapi dalam masalah regresi, regresi linier terutama digunakan sebagai pengklasifikasi meta.

DAFTAR PUSTAKA

Caie, P.D., Dimitriou, N. and Arandjelović, O., 2021. Chapter 8 - Precision medicine in digital pathology via image analysis and machine learning. In: S.B.T.-A.I. and D.L. in P. Cohen, ed. [online] Elsevier. pp.149–173. https://doi.org/https://doi.org/10.1016/B978-0-323-67538-3.00008-7.

Dietterich, T.G., 2000. Ensemble methods in machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1857 LNCS, pp.1–15. https://doi.org/10.1007/3-540-45014-9_1.

Hospedales, T., Antoniou, A., Micaelli, P. and Storkey, A., 2022. Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), pp.5149–5169. https://doi.org/10.1109/TPAMI.2021.3079209.

Hubbard, T., Aken, B., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., Coates, G., Fairley, S., Fitzgerald, S., Fernandez-Banet, J., Gordon, L.I., Gräf, S., Haider, S., Hammond, M., Holland, R., Howe, K., Jenkinson, A., Johnson, N., Kähäri, A., Keefe, D., Keenan, S., Kinsella, R., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Mégy, K., Meidl, P., Overduin, B., Parker, A., Pritchard, B., Ríos, D., Schuster, M., Slater, G., Smedley, D., Spooner, W., Spudich, G., Trevanion, S.J., Vilella, A.J., Vogel, J., White, S., Wilder, S.P., Zadissa, A., Birney, E., Cunningham, F., Curwen, V., Durbin, R., Fernández-Suárez, X.M., Herrero, J., Kasprzyk, A., Proctor, G., Smith, J., Searle, S.M.J. and Flicek, P., 2009. Ensembl 2009. *Nucleic Acids Research*, 37(Database), pp.D690–D697. https://doi.org/10.1093/nar/gkn828.

Jin, Z., Shang, J., Zhu, Q., Ling, C., Xie, W. and Qiang, B., 2020. RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis. *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12343 LNCS, pp.503–515. https://doi.org/10.1007/978-3-030-62008-0_35.

Krawczyk, B., Galar, M., Jeleń, Ł. and Herrera, F., 2016. Evolutionary Undersampling Boosting for Imbalanced Classification of Breast Cancer Malignancy. *Applied Soft Computing*, 38, pp.714–726. https://doi.org/10.1016/j.asoc.2015.08.060.

Li, Q.F. and Song, Z.M., 2022. High-performance concrete strength prediction based on ensemble learning. *Construction and Building Materials*, [online] 324(February), p.126694. https://doi.org/10.1016/j.conbuildmat.2022.126694.

Liang, M., Chang, T., An, B., Duan, X., Du, L., Wang, X., Miao, J., Xu, L., Gao, X., Zhang, L., Li, J. and Gao, H., 2021. A Stacking Ensemble Learning Framework for Genomic Prediction. *Frontiers in Genetics*, 12(March), pp.1–9. https://doi.org/10.3389/fgene.2021.600040.

Mrabah, N., Bouguessa, M. and Ksantini, R., 2023. Adversarial Deep Embedded Clustering: On a better trade-off between Feature Randomness and Feature Drift (Extended abstract). *Proceedings - International Conference on Data Engineering*, 2023-April, pp.3887–3888. https://doi.org/10.1109/ICDE55515.2023.00370.

Oza, N.C. and Tumer, K., 2008. Classifier ensembles: Select real-world applications. *Information Fusion*, 9(1), pp.4–20. https://doi.org/10.1016/j.inffus.2007.07.002.

Polikar, R., 2012. *Ensemble Machine Learning*. *Ensemble Machine Learning*. https://doi.org/10.1007/978-1-4419-9326-7.

Pourhoseingholi, M.A., Kheirian, S. and Zali, M.R., 2017. Comparison of Basic and Ensemble Data Mining Methods in Predicting 5-Year Survival of Colorectal Cancer Patients. *Acta Informatica Medica*, 25(4), p.254. https://doi.org/10.5455/aim.2017.25.254-258.

Richman, R. and Wüthrich, M. V., 2020. Bagging predictors. *Risks*, 8(3), pp.1–26. https://doi.org/10.3390/risks8030083.

Rokach, L., 2009. Ensemble-Based Classifiers. *Artificial Intelligence Review*, 33(1–2), pp.1–39. https://doi.org/10.1007/s10462-009-9124-7.

Sagi, O. and Rokach, L., 2018. Ensemble Learning: A Survey. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, 8(4). https://doi.org/10.1002/widm.1249.

Strobl, C., Malley, J.D. and Tutz, G., 2009. An Introduction to Recursive Partitioning: Rationale, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random Forests. *Psychological Methods*, 14(4), pp.323–348. https://doi.org/10.1037/a0016973.

Suthaharan, S., 2016. Chapter 6 - A Cognitive Random Forest: An Intra- and Intercognitive Computing for Big Data Classification Under Cune Condition. In: V.N. Gudivada, V. V Raghavan, V. Govindaraju and C.R.B.T.-H. of S. Rao, eds. *Cognitive Computing: Theory and Applications*. [online] Elsevier. pp.207–227. https://doi.org/https://doi.org/10.1016/bs.host.2016.07.006.

Tan, C., Li, M. and Xin, Q., 2008. Random Subspace Regression Ensemble for Near-Infrared Spectroscopic Calibration of Tobacco Samples. *Analytical Sciences*, 24(5), pp.647–653. https://doi.org/10.2116/analsci.24.647.

Verma, A., Pal, S. and Kumar, S., 2019. Classification of Skin Disease Using Ensemble Data Mining Techniques. *Asian Pacific Journal of Cancer Prevention*, 20(6), pp.1887–1894. https://doi.org/10.31557/apjcp.2019.20.6.1887.

Witten, I.H. and Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

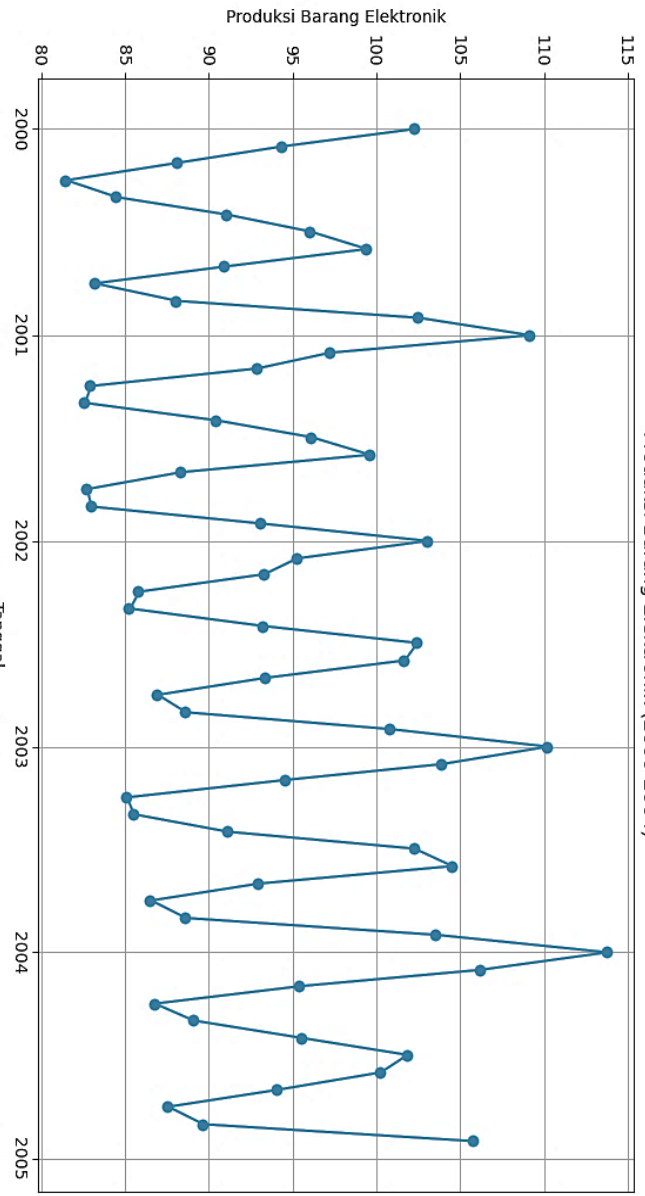
Wolpert, D.H., 1992. Stacked generalization. *Neural Networks*, 5(2), pp.241–259. https://doi.org/10.1016/S0893-6080(05)80023-1.

BAB 11 | *TIME SERIES ANALYSIS*

Mukarramah Yusuf, B.Sc., M.Sc.

A. Data Time Series

Data *time series* mengacu pada data yang dikumpulkan atau direkam mengikuti waktu sehingga dapat diurutkan berdasarkan waktu kejadiannya. Interval waktu dari data *time series* bisa detik, menit, jam, hari, bulan, atau bahkan tahun, tergantung pada konteks dan frekuensi pengumpulan data. Data *time series* dapat ditemui di berbagai bidang seperti ekonomi dan keuangan (misalnya data harga tutup saham), meteorologi (misalnya data curah hujan), dan pemrosesan sinyal (Chatfield, 2013; Pal & Prakash, 2017).



Gambar 11. 1. Contoh Data Time Series

Perkembangan sebuah variabel dari waktu ke waktu yang ditunjukkan oleh data *time series* dapat digunakan untuk analisis, peramalan atau prediksi, dan pengambilan keputusan. Analisis digunakan untuk menemukan misalnya tren dari variabel tersebut, apakah terjadi kenaikan, penurunan atau tidak ada perubahan dari waktu ke waktu. Sementara contoh peramalan adalah peramalan dan prediksi adalah prediksi penutupan harga saham di hari berikutnya. Selanjutnya hasil analisis dan hasil prediksi dapat menjadi bahan kajian atau pertimbangan untuk pengambilan keputusan tertentu, misalnya membeli atau tidak membeli saham sebuah perusahaan.

Data *time series* dapat reguler dan tidak reguler. Data time series yang reguler berarti interval waktu antar data adalah sama, disebut juga *time series* berjarak sama (*evenly spaced time series*). Contoh data ini adalah data curah hujan harian selama setahun. Gambar 11.1 adalah contoh data time series reguler yang lain, yaitu jumlah produksi barang elektronik per hari sebuah pabrik. Sedangkan data time series yang tidak reguler (irregular time series) berarti interval waktu antar data adalah tidak sama sama, disebut juga time series berjarak tidak sama (evenly spaced time series). Tetapi pada kebanyakan kasus, istilah data time series akan mengacu kepada regular time series, yaitu data time series yang memiliki jarak data yang sama.

B. Stasionaritas

Stasionaritas data time series mengacu pada sifat-sifat statistik di mana karakteristik data tetap konstan dari waktu ke waktu. Data time series dianggap stasioner jika:

1. Tidak memiliki tren
2. Nilai rata-rata datanya konstan dari waktu ke waktu
3. Variance konstan dari waktu ke waktu
4. Tidak memiliki autokorelasi

Sifat kestasioneran data time series sangat penting untuk banyak teknik analisis karena bila time series stasioner kita dapat menyederhanakan asumsi pemodelan dan memungkinkan prediksi yang lebih akurat dan kesimpulan yang lebih handal.

Mengidentifikasi stasionaritas melibatkan analisa perilaku data dari waktu ke waktu dan seringkali memerlukan transformasi atau penyesuaian untuk memastikan bahwa sifat statistiknya konstan. Memastikan stasionaritas sering kali merupakan langkah pertama dalam proses analisis ketika bekerja dengan data time series (Chatfield, 2013; Pal & Prakash, 2017).

Mengidentifikasi stasionaritas melibatkan kegiatan pemeriksaan rata-rata dan varians dari data selama interval waktu yang berbeda dan memeriksa tren atau pola. Kita dapat melakukan analisis data dengan eksplorasi (exploratory data analysis) dan tes Dickey-Fuller untuk memvalidasi sifat kestasioneran data time series.

Stasionaritas dapat dicapai melalui teknik seperti differencing untuk menghilangkan tren atau komponen musiman. Selain itu, tes statistik seperti tes Augmented Dickey-Fuller (ADF) atau tes Kwiatkowski-Phillips-Schmidt-Shin (KPSS) dapat digunakan untuk menilai stasionaritas secara formal.

1. Analisis Data dengan Eksplorasi

Analisis data explorasi merupakan metode visual untuk menemukan apakah distribusi sebuah data time series stasioner. Metode ini mencakup penggunaan rolling mean dan rolling variance untuk menjawab pertanyaan yang mendasari kestasioneran. Rolling mean menghitung rata-rata dari sebuah deret data untuk himpunan bagian sepanjang yang kita tentukan.

Misalnya untuk data 10, 50, 34, -5, 15, 19, 1, -30, 16, 37 dengan perhitungan *rolling mean* sepanjang 5, maka akan didapatkan rolling mean : NA, NA, NA, NA, NA, NA, 20,8, 22,6, 12,8, 0, 4,2, 8,6

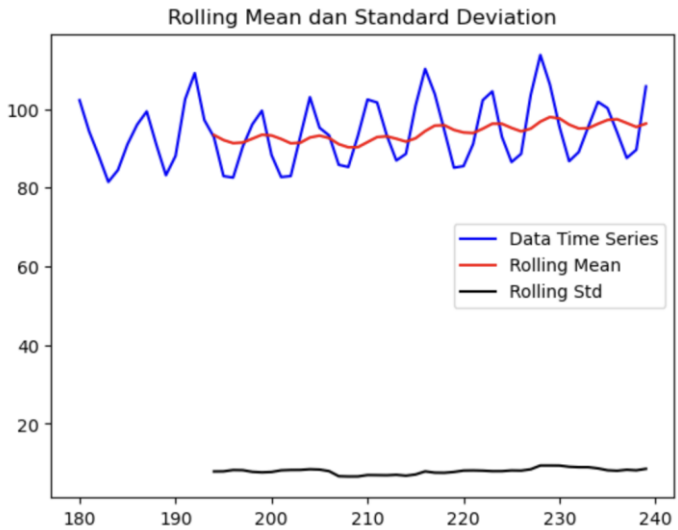
Rolling memiliki pengertian bergulir. Menghitung *rolling mean* dari 5 data berarti kita menyusuri data satu per satu dan menghitung rata-rata dari 5 data, rata-rata ini akan berubah setiap kita bergerak ke satu data yang berikutnya karena komponen dari data yang akan kita hitung rata-

ratanya berubah lagi. Hal yang sama berlaku untuk *rolling varian*, kita akan menghitung varian dari 5 data.

Listing 11.1: Rolling mean dan rolling varian

```
def plot_rolling_stats(timeseries, t=30):
#Rolling stastisik (rata-rata dan standar deviasi)
  rol_mean = timeseries.rolling(window=t).mean()
  rol_std = timeseries.rolling(window=t).std()
#Plot rolling statistik
orig = plt.plot(timeseries, color='blue',label='Data Time Series')
mean = plt.plot(rol_mean, color='red', label='Rolling Mean')
std = plt.plot(rol_std, color='black', label = 'Rolling Std Dev')
plt.legend(loc='best')
plt.title('Rolling Mean dan Standard Deviation')
plt.show(block=False)
```

Untuk contoh di atas dapat dilihat bahwa nilai rata-rata 5 data dari data *time series* sangat bervariasi, jauh dari karakateristik rata-rata konstan untuk kestasioneran. Berarti untuk contoh di atas, data time series tersebut tidak stasioner.



Gambar 11. 2. Rolling Mean dan Rolling Standar Deviasi dari Data Produksi Barang Elektronik

2. Uji Dickey-Fuller

Uji Dickey-Fuller merupakan pendekatan statistik untuk memeriksa apakah data dalam time series stasioner atau tidak. Hipotesa null dari Uji Dickey-Fuller adalah bahwa data time series yang sedang diuji tidak stasioner, dan hipotesa alternatif bahwa data time series yang sedang diuji stasioner. Saat diterapkan terhadap data time series, uji Dickey-Fuller akan mengembalikan statistik pengujian dan nilai-nilai kritis dengan tingkat kepercayaan (*confidence interval*) berbeda-beda.

Listing 11.2: Uji Dickey-Fuller

```
def ujiDF(timeseries):
#Pengujian Dickey-Fuller:
print ('Hasil uji Dickey-Fuller :')
dftest = adfuller(timeseries, autolag='AIC')
dfoutput = pd.Series(dftest[0:4], index=['Statistik Uji',
'p-value','Jumlah lag yang digunakan','Jumlah observasi yang
digunakan'])
for key,value in dftest[4].items():
dfoutput['Nilai kritis (%s)'%key] = value
print(dfoutput)
```

Hasil pemanggilan fungsi *plot_rolling_stats* di atas dengan ukuran window 15 ditunjukkan oleh gambar 10.2. Dapat dilihat bahwa nilai rolling mean (rata – rata bergulir) berfluktuasi yang berarti tidak konstan sehingga kita dapat menarik kesimpulan bahwa data time series produksi barang elektronik ini tidak stasioner. Terlebih bahwa variasi nilai juga ditunjukkan oleh rolling standar deviasi.

Kesimpulan ini juga diperkuat oleh hasil uji Dickey-Fuller dengan statistik tes yang lebih besar dari nilai kritis yang artinya hipotesa null diterima bahwa data time series ini lebih dekat kepada kondisi non-stasioner.

Hasil uji Dickey-Fuller :

Statistik Uji	0.210085
p-value	0.972822
Jumlah lag yang digunakan	11.000000
Jumlah observasi yang digunakan	48.000000
Nilai kritis (1%)	-3.574589
Nilai kritis (5%)	-2.923954
Nilai kritis (10%)	-2.600039

Berikut adalah contoh spesifik analisa data time series yang memerlukan kestasioneran:

1. **Autokorelasi dan Analisis Autokorelasi Parsial**

   Fungsi autokorelasi dan autokorelasi parsial adalah alat penting untuk memahami dependensi temporal dalam deret waktu. Data stasioner diperlukan untuk secara akurat menafsirkan autokorelasi dan pola autokorelasi parsial, karena non-stasionaritas dapat menyebabkan korelasi palsu.

2. **Pemodelan dengan ARIMA**

   Pemodelan ARIMA (*AutoRegressive Integrated Moving Average*) banyak digunakan untuk prediksi menggunakan data time series. Model ARIMA mengasumsikan stasionaritas, dan metode *differencing* diterapkan untuk membuat data stasioner sebelum melakukan *fitting* terhadap model. Stasionaritas memastikan bahwa hubungan antar variabel tetap konsisten dari waktu ke waktu, untuk mendapatkan perkiraan yang lebih akurat.

3. **Dekomposisi Musim (*Seasonality decomposition*)**

   Teknik dekomposisi musim, seperti dekomposisi tren musiman menggunakan LOESS (STL), memisahkan data time series menjadi komponen musim, tren, dan residunya. Data stasioner diperlukan untuk dekomposisi yang akurat, karena komponen non-stasioner dapat mendistorsi hasil dekomposisi dan menyebabkan interpretasi yang salah.

4. Prediksi

Stasionaritas sangat penting untuk menghasilkan perkiraan handal. Non-stasionaritas dapat menimbulkan bias dan meningkatkan kesalahan perkiraan, sehingga sulit untuk mendapatkan prediksi nilai masa depan yang akurat.

C. Membuat Data Time Series Stasioner

Terdapat beberapa metode yang diketahui dapat membuat data time series stasioner. Tetapi sesungguhnya, tidak satu pun dari metode-metode tersebut yang menjanjikan kondisi stasioner sempurna(Haroon & Clustering, 2017). Yang dapat dilakukan oleh metode ini adalah mengubah data time series agar terlihat lebih dekat ke kondisi stasioner. Hal ini dilakukan dengan menghilangkan tren dan musim dari data time series. Berikut ini adalah beberapa metode yang dapat membuat data time series lebih mendekati kondisi stasioner:

1. Menerapkan transformasi
2. Memperkirakan tren dan menghapusnya dari data time series asli
3. *Differencing* (pengukuran perbedaan)
4. Dekomposisi

1. Menerapkan Transformasi

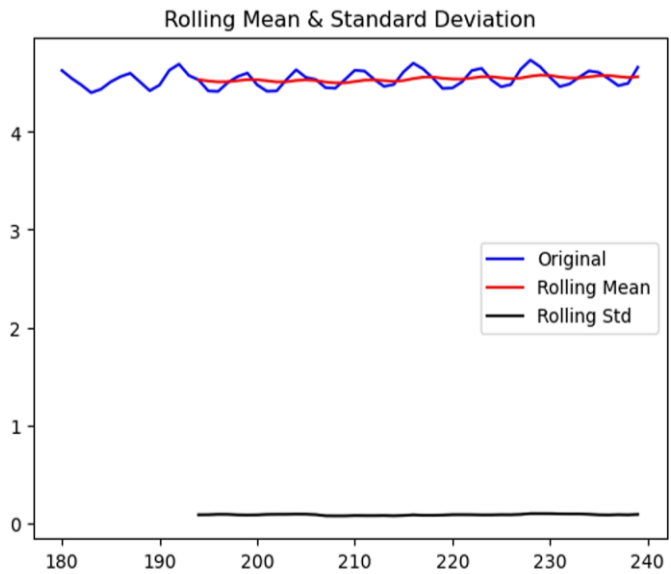
Dengan transformasi seperti log, akar pangkat tiga, dan akar kuadrat angka-angka yang lebih besar akan berkurang magnitudonya secara drastis (dibandingkan angka yang lebih kecil). Misalnya log 1 juta adalah 6 sementara log 100 adalah 2. Pengurangan magnitudo ini bisa membuat data time series mendekati stasioner.

Listing 11.2: Rolling mean dan rolling varian nilai log data

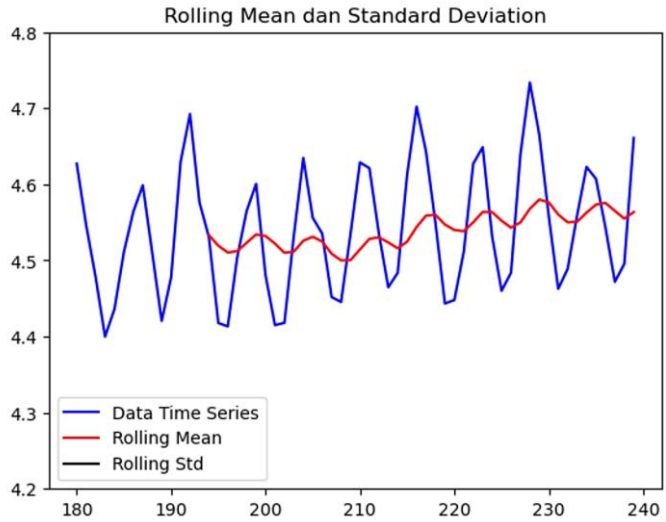
```
import numpy as np
data_log = np.log(df_filtered['IPG2211A2N'])
evaluate_stationarity(data_log, 15)
Results of Dickey-Fuller Test:
Test Statistic                  0.244861
p-value                         0.974668
```

#Lags Used	11.000000
Number of Observations Used	48.000000
Critical Value (1%)	-3.574589
Critical Value (5%)	-2.923954
Critical Value (10%)	-2.600039
dtype: float64	

Dengan melihat grafik rolling mean dari nilai yang sudah ditransformasi yang sudah menunjukkan nilai konstan (gambar 11.3) mungkin kita menyangka sudah dapat mengambil kesimpulan bahwa transformasi berhasil mengubah data time series dari non-stasioner menjadi stasioner. Akan tetapi nilai uji Dickey-Fuller menunjukkan statistik uji yang lebih besar dari nilai kritis yang artinya hipotesa null diterima yaitu bahwa data time series yang sedang diuji adalah non-stasioner. Untuk itu, kita coba zoom-in grafik rolling mean.



Gambar 11. 3. Rolling Mean dan Rolling Standar Deviasi dari Nilai Log Data Produksi Barang Elektronik



Gambar 11. 4. Zoom-In Rolling Mean dari Nilai Log Data Produksi Barang Elektronik (Rolling Standar Deviasi di Luar Batas Plot)

Hasil *zoom-in* (gambar 11.4) terhadap data menunjukkan bahwa nilai-nilai hasil transformasi log masih bervariasi sehingga rolling mean juga bervariasi sehingga hasil uji Dickey-Fuller benar bahwa data masih non-stasioner. Transformasi berupa perubahan data menjadi nilai akar kuadratnya juga dapat dilakukan, tetapi untuk contoh data produksi barang elektronik yang diberikan, transformasi akar kuadrat juga tidak dapat menghasilkan data yang stasioner.

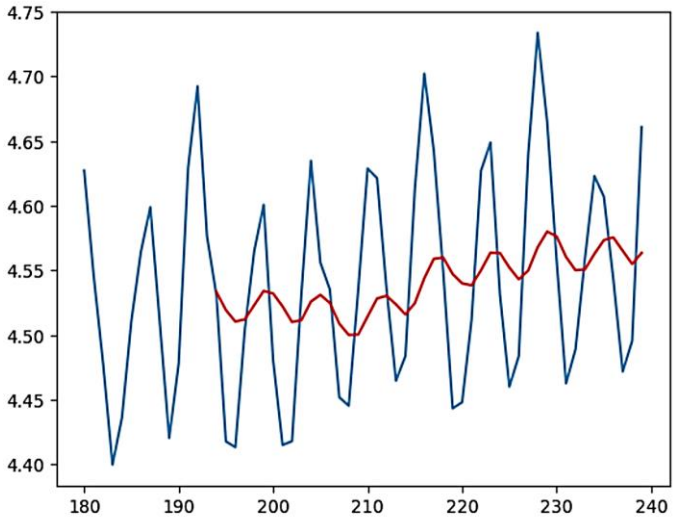
2. **Memperkirakan tren dan menghapusnya dari data time series asli**

Cara lain untuk membuat data time series yang stasioner adalah menghapus atau menghilangkan tren dari data tersebut. Untuk itu langkah pertama yang harus dilakukan adalah menemukan tren dari time series tersebut. Salah satu cara untuk menemukan tren adalah dengan metode penghalusan rata-rata bergerak (moving average smoothing).

Moving Average smoothing adalah metode penghalusan dengan menggunakan rolling mean (rata-rata bergulir), tren pada data time series dapat diperkirakan dengan nilai rolling mean.

Listing 11.3: Moving average

```
moving_avg = data_log.rolling(window=15).mean()
plt.plot(data_log)
plt.plot(moving_avg, color='red')
```



Gambar 11. 5. Moving Average dari Nilai Log Data Produksi Barang Elektronik dengan Window=5

Kurva merah pada gambar 11.5 adalah rolling mean yang didapatkan dari melakukan mo*ving average smoothing,* dan dapat dianggap mewakili tren dari data *time series* di atas (kurva biru). Dapat dilihat bahwa tidak ada nilai *rolling mean* untuk 14 data pertama, karena besarnya *window moving average* telah ditentukan 15.

Tren pada data dihilangkan dengan mengurangi nilai time series dengan nilai trennya (listing 11.4). Hasil dari pengujian Dickey-Fuller menunjukkan moving average belum dapat menghilangkan stasionaritas dari nilai log tanpa

tren data time series produksi elektronik. Sekilas data tampak sudah stationer seperti yang diperlihat gambar 11.6, namun *zoom-in* grafik akan memperlihatkan bahwa sesungguhnya data belum stationer.

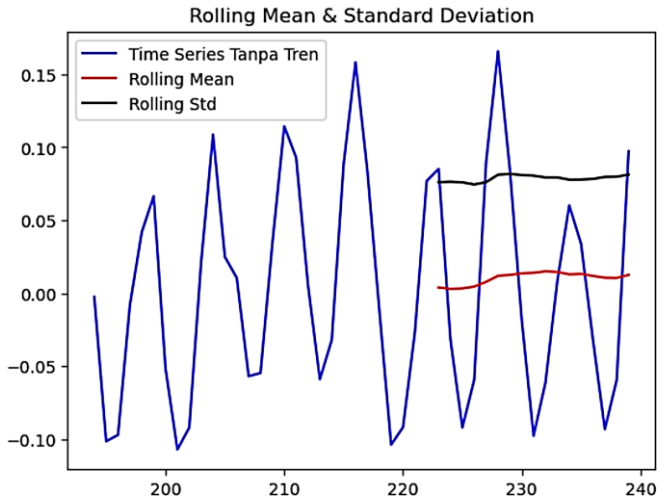
Listing 11.4: Menghilangkan tren dengan moving average

```
data_log_moving_avg_diff = data_log - moving_avg
data_log_moving_avg_diff.head(15)
data_log_moving_avg_diff.dropna(inplace=True)
evaluate_stationarity_modif(data_log_moving_avg_diff)
```

Results of Dickey-Fuller Test:

Test Statistic	-1.778322
p-value	0.391276
#Lags Used	10.000000
Number of Observations Used	35.000000
Critical Value (1%)	- 3.632743
Critical Value (5%)	-2.948510
Critical Value (10%)	-2.613017

dtype: float64



Gambar 11. 6. Rolling Mean dan Rolling Standar Deviasi dari Nilai Log Tanpa Tren Data Produksi Barang Elektronik

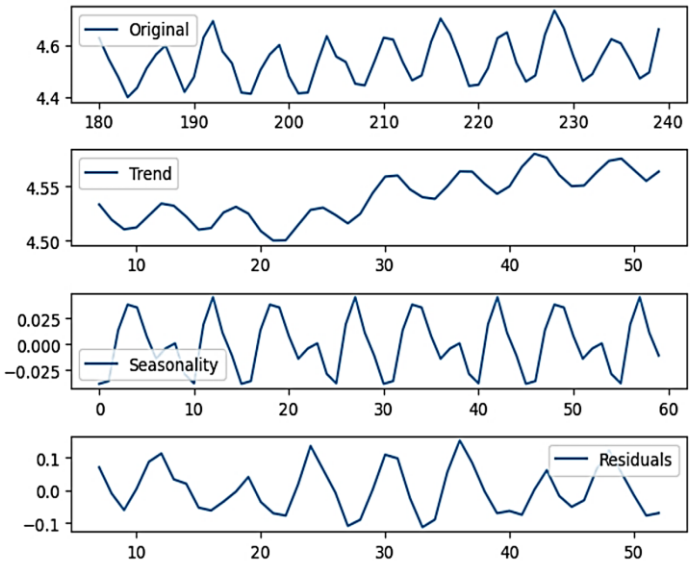
3. Dekomposisi

Dekomposisi adalah pendekatan lain untuk menghilangkan tren dan musim dari deret waktu untuk membuatnya stasioner. Dekomposisi dilakukan dengan membagi deret waktu menjadi tiga komponen: tren, musim, dan residu. Komponen yang kemudian digunakan dalam hal ini adalah residu (yaitu, deret waktu tanpa tren dan musim). Dekomposisi dilakukan seperti listing 11.5.

Listing 11.5: Dekomposi data time series

```
from statsmodels.tsa.seasonal import seasonal_decompose
decomp_res =seasonal_decompose(list(data_log), period=15)
trend = decomp_res.trend
seasonal = decomp_res.seasonal
residual = decomp_res.resid
plt.subplot(411)
plt.plot(data_log, label='Original')
plt.legend(loc='best')
plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend(loc='best')
plt.subplot(413)
plt.plot(seasonal,label='Seasonality')
plt.legend(loc='best')
plt.subplot(414)
plt.plot(residual, label='Residuals')
plt.legend(loc='best')
plt.tight_layout()
```

Tren, musim, dan residu hasil dekomposisi dapat divisualisasi seperti. Komponen yang kemudian digunakan dalam hal ini adalah residu (yaitu, deret waktu tanpa tren dan musim). Dekomposisi dilakukan seperti listing 11.5. Gambar 11.7 memperlihatkan tren, musim dan residu hasil dekomposisi terhadap data *time series* awal.



Gambar 11. 7. Data Time Series Asli, Tren, Musim dan Residu Hasil Dekomposisi

4. *Differencing* (Pembedaan)

Pembedaan adalah menilai perbedaan nilai pada satu titik waktu dengan titik waktu yang sebelumnya. Listing 11.6 melakukan proses *differencing* dan pengujian Dicky-Fuller terhadap hasil *differencing*. Proses ini dapat terus diulangi sampai didapatkan time series yang stasioner.

Listing 11.6: *Differencing* dan pengujian hasil differencing

```
df_diff = df['IPG2211A2N'].diff().dropna()
result = adfuller(df_diff)
print('ADF Statistic:', result[0])
print('p-value:', result[1])
ADF Statistic: -7.104890882267285
p-value: 4.077786565540049e-10
```

D. Membangun Model dan Melakukan Prediksi

Memodelkan time series memiliki pengertian mencari bentuk umum dari sebuah time series. Model ini dibangun untuk melakukan prediksi atau peramalan. Peramalan time

series adalah proses menggunakan data masa lalu untuk memprediksi peristiwa masa depan. Model peramalan time series adalah model statistik yang digunakan untuk membuat prediksi tentang nilai masa depan berdasarkan titik data historis yang disusun dalam urutan kronologis. Model-model ini menganalisis tren dan pola dalam data dan mengekstrapolasikannya untuk membuat prediksi tentang nilai masa depan. Model-model ini biasanya digunakan misalnya dalam bisnis dan keuangan untuk memprediksi penjualan atau harga saham, dan misalnya dalam sains untuk memprediksi pola cuaca. Model peramalan time series adalah kelompok khusus pemodelan prediktif yang digunakan untuk meramalkan peristiwa masa depan.

ARIMA adalah salah satu model prediksi time series. Parameter untuk memodelkan data time series dilambangkan dengan p, d, q, di mana:

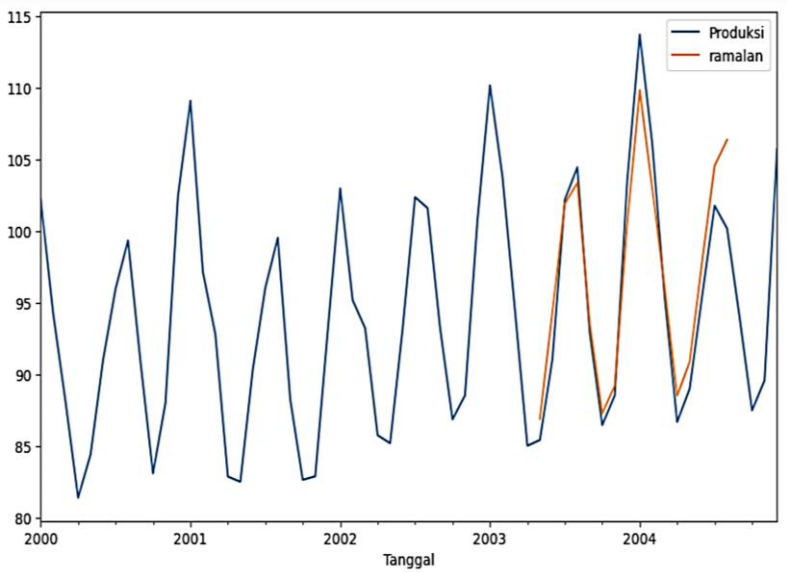
1. p adalah komponen Auto-Regressive (AR), yang menunjukkan *lag* (keterlambatan) variabel respons, atau ketergantungan nilai saat ini pada nilai-nilai sebelumnya sendiri. Misalnya, untuk nilai p = 5, untuk memprediksi variabel respons pada t5, deret waktu antara t0 dan t4 akan dianggap sebagai variabel eksplorasi.
2. d adalah tingkat perbedaan yang diperlukan untuk menghilangkan tren dan atau musim, dalam rangka mencapai stasionaritas.
3. q adalah komponen Moving Average (MA). MA menangkap hubungan antara pengamatan dan kesalahan residual dari model moving average yang diterapkan pada pengamatan sebelumnya. Variabel ini memodelkan ketergantungan nilai saat ini terhadap kesalahan perkiraan sebelumnya. Ambil, misalnya, nilai q = 5. Oleh karena itu untuk memprediksi variabel respons pada T5, deret waktu antara T0 dan T4 akan dipertimbangkan.

Dengan menyesuaikan nilai p, d, dan q, model ARIMA dapat dibentuk agar sesuai dengan berbagai jenis data deret waktu, memungkinkan pemodelan dan peramalan yang akurat

dari berbagai pola dan dinamika temporal. Untuk melakukan prediksi, ARIMA membutuhkan data stasioner yang berarti tidak memiliki tren dan musim. Sementara itu, banyak data deret waktu dunia nyata menunjukkan tren, musim, atau karakteristik non-stasioner lainnya. Faktanya, sangat umum bagi data deret waktu untuk memiliki tren, terutama dalam data ekonomi dan keuangan, di mana tren jangka panjang sering diamati. Untuk itu, dalam melakukan prediksi, ARIMA akan menganalisis tren kemudian melakukan *differencing* untuk menciptakan data yang stasioner. Secara detil, cara kerjanya adalah sebagai berikut:

1. Mendeteksi Tren: ARIMA memang dapat memodelkan data deret waktu yang menunjukkan tren. Komponen AR (autoregresif) dan MA (moving average) dari ARIMA memungkinkannya untuk menangkap berbagai jenis tren dalam data.
2. Persyaratan Stasionaritas: ARIMA mengharuskan data menjadi stasioner, tetapi ini tidak berarti deret waktu asli harus stasioner. Sebaliknya, ARIMA mencapai stasionaritas melalui *differencing*. *Differencing* melibatkan pengurangan pengamatan saat ini dari pengamatan yang sebelumnya(*lag*), secara efektif menghilangkan tren dan membuat data stasioner.

SARIMA adalah model ARIMA untuk data yang memiliki musim (seasonal), yang dapat diketahui dengan adanya pola secara periodik. SARIMA menambahkan komponen periodik data. Listing 11.6 membangun model SARIMA dan kemudian melakukan prediksi dengan model tersebut. Dengan menyesuaikan nilai p, d, dan q, model SARIMA dapat dibentuk agar sesuai dengan berbagai jenis data deret waktu, memungkinkan pemodelan dan peramalan yang akurat dari berbagai pola dan dinamika temporal.



Gambar 11. 8. Data Time Series Produksi Barang Elektronik dan Hasil Prediksinya Menggunakan Model SARIMA

Listing 11.6: Pembangunan model SARIMA dan prediksi nilai masa depan

```
import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['Produksi'],order=(5,1,0)
, seasonal_order=(5,1,0,12))
results=model.fit()
df['ramalan']=results.predict(start=40,end=55,
dynamic=True)
df[['Produksi','ramalan']].plot(figsize=(10,6))
```

DAFTAR PUSTAKA

Chatfield, C. (2013). *The analysis of time series: theory and practice*. Springer.

Haroon, D., & Clustering, I. (2017). *Python machine learning case studies*. Springer.

Pal, A., & Prakash, P. K. S. (2017). *Practical time series analysis: master time series data processing, visualization, and modeling using python*. Packt Publishing Ltd.

BAB 12 | TEXT MINING

Riadi Marta Dinata, S.Ti., M.Kom.

A. Pengertian Text Mining

Text mining atau penambangan teks, adalah proses menganalisis kumpulan data besar yang tidak terstruktur untuk menghasilkan hasil yang valid dan bermakna. Data besar ini contohnya seperti dokumen, email, artikel, data komentar media sosial, komentar di market apps, dan lain sebagainya.

Dengan hasil valid dan bermakna, artinya kita dapat menemukan pola, tren, dan hubungan antar dokumen yang mungkin tidak terlihat jika dibaca secara manual.

Beberapa contoh penggunaan text mining diantaranya adalah :

1. Menganalisis opini publik dari media sosial
2. Melakukan riset pasar dengan menganalisis ulasan pelanggan.
3. Mencari artikel penelitian yang relevan dengan topik tertentu.
4. Mendeteksi penipuan dengan menganalisis email.

Dengan demikian, text mining menjadi alat yang sangat bermanfaat di era dimana informasi berbentuk teks semakin banyak dan beragam.

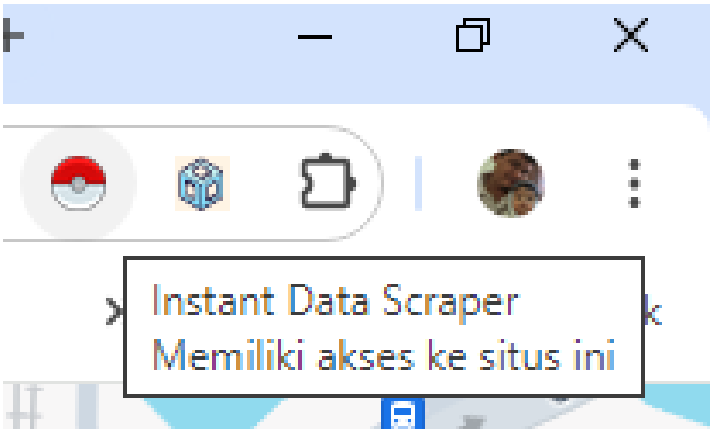
B. Cara Kerja

Pada Text Mining, sistem membutuhkan beberapa langkah untuk untuk memastikan keakuratan, keandalan, dan kegunaan maksimal dari hasil analisisnya.Adapun langkah-langkahnya adalah sebagai berikut:

1. Pengumpulan Data

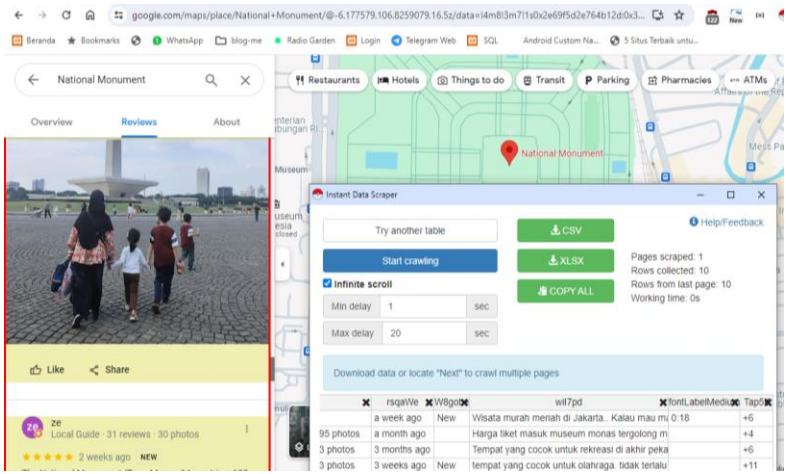
Langkah pertama adalah mengumpulkan data teks yang ingin dianalisis. Data ini dapat berasal dari berbagai sumber, seperti dokumen, email, artikel web, media sosial, dan lain sebagainya. Langkahnya bisa dengan cara crawling maupun melakukan parsing JSON API dari alamat web tersebut.

Dari pengalaman penulis, yang paling mudah adalah menggunakan tools scrapper online, misalkan Instant Data Scrapper dari https://webrobots.io/instantdata/, yaitu cukup dipasangkan pada Google Chrome maka Tools akan auto aktif.



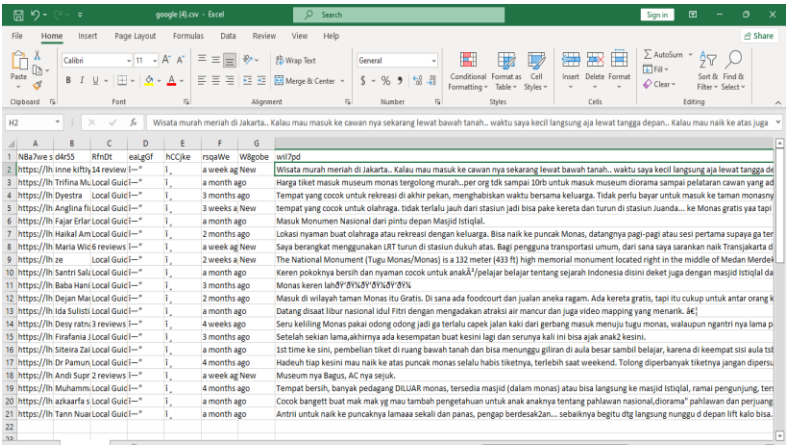
Gambar 12. 1. Tools Instant Data Scraper pada Google Chrome

Misalkan kita hendak menarik data review dari maps google dengan subyek pencarian "Monumen Nasional", maka hasilnya adalah berikut:



Gambar 12. 2. Proses Crawling Data Google Map

a. Tool akan memberi tanda blok merah, terkait data mana yang hendak di Tarik. Dan jika posisi blok merah ini belum tepat pada data area yang hendak di ambil, maka bisa menggunakan tombol "Try Another Table" untuk berpindah antar blok hingga terposisikan dengan tepat di area target.
b. Gunakan tombol "Start Crawling" jika sudah yakin data yang hendak dianalisa sudah berada posisi yang tepat.
c. Setekah data tertarik dan muncul pada table Tools, selanjutkan Langkah download ke file .csv atau file .xlsx untuk dipilah kolom mana yang hendak diproses.
d. Buka file hasil crawling, dan copas kolom komentar ke file baru misal file komentar.csv



	A	B	C	D	E	F	G
1	NBa7we s	d4r55	RfnDt	eaLgGf	hCCjke	rsqaWe	W8gobe
2	https://lh	inne kiftiy	14 review	l—ª	ì¸	a week ag	New
3	https://lh	Trifina Mu	Local Guicl	l—ª	ì¸	a month ago	
4	https://lh	Dyestra	Local Guicl	l—ª	ì¸	3 months ago	
5	https://lh	Anglina fii	Local Guicl	l—ª	ì¸	3 weeks a	New
6	https://lh	Fajar Erlar	Local Guicl	l—ª	ì¸	a month ago	
7	https://lh	Haikal Am	Local Guicl	l—ª	ì¸	2 months ago	
8	https://lh	Maria Wic	6 reviews	l—ª	ì¸	a week ag	New
9	https://lh	ze	Local Guicl	l—ª	ì¸	2 weeks a	New
10	https://lh	Santri Salu	Local Guicl	l—ª	ì¸	a month ago	
11	https://lh	Baba Hani	Local Guicl	l—ª	ì¸	3 months ago	
12	https://lh	Dejan Mar	Local Guicl	l—ª	ì¸	2 months ago	
13	https://lh	Ida Sulisti	Local Guicl	l—ª	ì¸	a month ago	
14	https://lh	Desy ratnı	3 reviews	l—ª	ì¸	4 weeks ago	
15	https://lh	Firafania J	Local Guicl	l—ª	ì¸	3 months ago	
16	https://lh	Siteira Zai	Local Guicl	l—ª	ì¸	a month ago	
17	https://lh	Dr Pamun	Local Guicl	l—ª	ì¸	4 months ago	
18	https://lh	Andi Supr	2 reviews	l—ª	ì¸	a week ag	New
19	https://lh	Muhamm	Local Guicl	l—ª	ì¸	4 months ago	
20	https://lh	azkaarfa s	Local Guicl	l—ª	ì¸	a month ago	
21	https://lh	Tann Nuai	Local Guicl	l—ª	ì¸	a month ago	
22							

Gambar 12. 3. Hasil Crawling

2. Pre-processing

Data teks yang dikumpulkan perlu dibersihkan dan dipreproses sebelum dapat diolah lebih lanjut.

Proses ini meliputi :

a. Filtering, yaitu proses menghilangkan (membersihkan) noise seperti karakter spesial, HTML tag, dan karakter non ascii standard lainnya; missal: ÿØàó:ÙïF KÃB°èIâzuåËïc Yågêyç VÞ_rO™îs¯,
,<b>,<a>,<hr> dsb

b. Casefolding, yaitu proses mengubah huruf kapital menjadi huruf kecil(lowercase); missal Selamat Pagi menjadi selamat pagi, dsb.

c. Stopword, yaitu proses menghilangkan kata-kata hubung, angka, tanda baca, dan double spasi; missal: yang, di, ketika, maka, jika, dari, dan, penuh, at, on, atas, bawah, nya, #,$,%,>,?, 1,2,3 dsb.

d. Stemming, yaitu proses mengubah kata menjadi bentuk dasarnya (akar kata) untuk mengurangi variasi bentuk kata; misal: Berlari menjadi lari, memaknai menjadi makna, belajar menjadi ajar, dsb.

e. Lemmatization, yaitu proses mereduksi kata atau istilah menjadi kata yang mudah dipahami atau standard; missal: online menjadi daring, coffee menjadi kopi, 'nyok kita cabut menjadi pergi dsb.
f. Tokenisasi, yaitu untuk memecah teks menjadi unit-unit dasar (token) seperti kata atau frasa.
g. Unifikasi, yaitu menghilangkan kata yang terduplikasi.

Contoh kalimat :

"Tulislah kisah-kisah yang penuh kebahagiaan dan makna. Happy at Morning! "

Tabel 12. 1. Contoh Pra-Proses

Filtering	Tulislah kisah-kisah yang penuh kebahagiaan dan makna. Happy at Morning!
Casefolding	tulislah kisah-kisah yang penuh kebahagiaan dan makna. Happy at morning!
Stopword	tulislah kisah kisah kebahagiaan makna happy morning
Stemming	Tulis kisah kisah Bahagia makna happy morning
Lemmatisasi	Tulis cerita-cerita Bahagia makna Bahagia pagi
Unifikasi	Tulis cerita bahagia makna pagi

3. Feature Extraction

Mengubah teks yang telah dipreproses menjadi bentuk yang terstruktur dan dapat diolah oleh algoritma. Dan konsep Entropy Term adalah yang umum diterapkan yaitu dengan cara menghitung frekunesi kemunculan setiap kata dalam dokumen. Artinya, nilai entropy yang tinggi menunjukkan bahwa kata tersebut lebih informatif dan lebih penting untuk analisis. Selanjutnya, Algoritma yang menggunakan Entropy Term kemudian akan memberikan

bobot yang lebih besar kepada kata-kata dengan nilai entropy tinggi saat melakukan tugas seperti klasifikasi, clustering, atau penemuan topik.

Misalkan kita memiliki dua dokumen:
Dokumen 1: "*Saya ingin membeli ponsel baru di toko online favorit saya. Harganya agak mahal, tapi fiturnya sangat lengkap dan ulasannya banyak positif. Apakah ada yang ingin kamu sarankan*?"

Dokumen 2: "*Saya ingin membeli komputer baru di toko elektronik favorit saya. Harganya agak mahal, tapi fiturnya sangat lengkap dan ulasannya banyak positif. Apakah ada yang ingin kamu sarankan?*"

Entropy Term dihitung untuk setiap kata dalam kedua dokumen. Kata-kata seperti "*ponsel*", "*komputer*", "*toko online*", dan "*toko elektronik*" memiliki nilai entropy yang lebih tinggi karena kata-kata tersebut lebih spesifik dan lebih membedakan kedua dokumen. Algoritma-algoritma yang menerapkan Entropy Term selanjutnya akan memberikan bobot yang lebih besar kepada kata-kata tersebut hingga diproses klasifikasi dokumen.

$$H = -\sum p(x)\log p(x)$$

Gambar 12. 4. Formula Entropy Term

H : Nilai Entropi
P : nilai probabilitas item x
Log : nilai logaritmik2 dari probabilitas x

Misal dari case di atas, kita hendak menghitung nilai Entropy Term untuk Setiap Dokumen

Dokumen 1:
Frekuensi kata "beli": 1
Total kata: 18
Probabilitas p(beli, d1) = 1 / 18 ≈ 0.056
Entropy_Term(beli, d1) = - 0.056 * log2(0.056) ≈ 0.837

Dokumen 2:
Frekuensi kata "beli": 1
Total kata: 17
Probabilitas p(beli, d2) = 1 / 17 ≈ 0.059
Entropy_Term(beli, d2) = - 0.059 * log2(0.059) ≈ 0.820

Selanjutnya kita akan hitung Entropy Term Rata-rata
Entropy term rata-rata =
= (Entropy_Term(beli, d1) + Entropy_Term(beli, d2)) / 2
= (0.837 + 0.820) / 2 ≈ 0.829

Artinya, dengan nilai H entropy term rata-rata 0.829 menunjukkan bahwa kata "beli" cukup informatif dan penting dalam kedua dokumen. Nilai ini diperoleh dengan menjumlahkan entropy term untuk kata "beli" di kedua dokumen dan kemudian membagi dengan jumlah dokumen.

Kini, beberapa algoritma yang umum digunakan dalam peningkatan akurasi dan efisiensi data tidak terstruktur agar lebih terarah lagi, antara lain:

a. TF-IDF (*Term Frequency-Inverse Document Frequency*), yaitu dengan cara menghitung frekuensi kemunculan kata dalam dokumen dan dokumen-dokumen lain dalam korpus.
b. N-gram, yaitu proses membentuk urutan kata (n-kata) dari teks untuk menangkap informasi kontekstual.

Selain kedua metode tersebut, terdapat beberapa metode pengembangan lainnya seperti:

a. *Part-of-Speech (POS) Tagging*, yaitu mengidentifikasi kategori gramatikal kata-kata dalam datasetnya, seperti kata benda, kata kerja, kata sifat, dan lain sebagainya.
b. *Dependency Parsing*,yaitu dengan cara membangun struktur dependensi antar kata dalam kalimat, dengan menunjukkan bagaimana kata-kata saling berhubungan secara sintaksis.
c. *Topic Modeling*, yaitu mengidentifikasi topik-topik yang mendasari kumpulan dokumen teks.

d. *Word Embeddings*, yaitu metode yang merepresentasikan kata-kata sebagai vektor numerik, di mana kata-kata yang memiliki makna serupa akan memiliki vektor yang berdekatan.
e. *Semantic Features*, yaitu mengekstrak fitur semantik dari teks, seperti synsets, hypernyms, dan hyponyms.

Berikut adalah contoh sederhana perhitungan TF-IDF, dengan misalakan kita memiliki 3 dokumen:

a. Dokumen 1 (D1): "kucing suka ikan"
b. Dokumen 2 (D2): "ikan suka air"
c. Dokumen 3 (D3): "kucing dan ikan suka air"

Langkah 1: Hitung Term Frequency (TF)

Term Frequency (TF) adalah jumlah kemunculan sebuah kata dalam sebuah dokumen dibagi dengan total jumlah kata dalam dokumen tersebut.

Dokumen 1 (D1): Total kata: 3
TF("kucing", D1) = 1/3
TF("suka", D1) = 1/3
TF("ikan", D1) = 1/3

Dokumen 2 (D2): Total kata: 3
TF("ikan", D2) = 1/3
TF("suka", D2) = 1/3
TF("air", D2) = 1/3

Dokumen 3 (D3): Total kata: 4
TF("kucing", D3) = 1/4
TF("dan", D3) = 1/4
TF("ikan", D3) = 1/4
TF("suka", D3) = 1/4
TF("air", D3) = ¼

Langkah 2: Hitung Inverse Document Frequency (IDF)

Merupakan item log dari total jumlah dokumen dibagi dengan jumlah dokumen yang mengandung kata tersebut.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

Gambar 12. 5. Formula TF-IDF

tf= jumlah kata x terhadap seluruh kata di dokumen y
N=jumlah dokumen (contoh total dokumen 3)
df=jumlah dokumen yang di dalamnya mengandung x

Jika diketahui total dokumen ada 3 buah, maka:
IDF per kata:
IDF("kucing") = log(3 / 2) = 0.176
IDF("suka") = log(3 / 3) = 0 (ada di semua dokumen)
IDF("ikan") = log(3 / 3) = 0 (ada di semua dokumen)
IDF("air") = log(3 / 2) = 0.176
IDF("dan") = log(3 / 1) = 0.477

Langkah 3: Hitung TF-IDF

TF-IDF adalah perkalian antara TF dan IDF untuk setiap kata dalam setiap dokumen.
Dokumen 1 (D1)
TF-IDF("kucing", D1) = (1/3) * 0.176 = 0.059
TF-IDF("suka", D1) = (1/3) * 0 = 0
TF-IDF("ikan", D1) = (1/3) * 0 = 0

Dokumen 2 (D2)
TF-IDF("ikan", D2) = (1/3) * 0 = 0
TF-IDF("suka", D2) = (1/3) * 0 = 0
TF-IDF("air", D2) = (1/3) * 0.176 = 0.059

Dokumen 3 (D3)
TF-IDF("kucing", D3) = (1/4) * 0.176 = 0.044
TF-IDF("dan", D3) = (1/4) * 0.477 = 0.119
TF-IDF("ikan", D3) = (1/4) * 0 = 0
TF-IDF("suka", D3) = (1/4) * 0 = 0
TF-IDF("air", D3) = (1/4) * 0.176 = 0.044

Tabel 12. 2. Contoh Hasil TF-IDF

Dokumen	kucing	suka	ikan	air	dan
D1	0.059	0	0	0	0
D2	0	0	0	0.059	0
D3	0.035	0	0	0.035	0.095

Artinya TF-IDF akan memberikan bobot yang lebih tinggi pada kata yang jarang muncul di semua dokumen, tetapi cukup sering muncul dalam dokumen tertentu, membantu mengidentifikasi kata-kata yang lebih relevan untuk setiap dokumen.

Sedang untuk cara kerja dari N-GRAM missal pada 3 dokumen di atas adalah berikut:

Langkah 1. Menghitung Frekuensi N-gram (untuk semua doc)
Unigram:
Frekuensi "kucing": 1 (D1) + 1 (D3) = 2
Frekuensi "suka": 1 (D1) + 1 (D2) + 1 (D3) = 3
Frekuensi "ikan": 1 (D1) + 1 (D2) + 1 (D3) = 3
Frekuensi "air": 1 (D2) + 1 (D3) = 2
Frekuensi "dan": 1 (D3) = 1

Bigram:
Frekuensi "kucing suka": 1 (D1) = 1
Frekuensi "suka ikan": 1 (D1) = 1
Frekuensi "ikan suka": 1 (D2) + 1 (D3) = 2
Frekuensi "suka air": 1 (D2) + 1 (D3) = 2
Frekuensi "kucing dan": 1 (D3) = 1
Frekuensi "dan ikan": 1 (D3) = 1

Trigram:
Frekuensi "kucing suka ikan": 1 (D1) = 1
Frekuensi "ikan suka air": 1 (D2) + 1 (D3) = 2
Frekuensi "kucing dan ikan": 1 (D3) = 1
Frekuensi "dan ikan suka": 1 (D3) = 1

Tabel 12. 3. Contoh N-GRAM

Dok	kucing	suka	ikan	air	dan	kucing suk	suka ikan	ikan suka	suka air	kucing dar	dan ikan	kucing suk	ikan suka	kucing dar	dan ikan s
D1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0
D2	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0
D3	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1

Dengan melihat distribusi N-gram (unigram, bigram, dan trigram) di dalam tiga dokumen selanjutnya siap diproses dalam analisis teks untuk memahami pola kata dan frasa yang sering muncul bersama.

4. Modeling Network

Setelah kita mendapatkan tabel normalisasi dari TF-IDF maupun N-GRAM, selanjutnya kita dapat menggunakan data tersebut sebagai fitur untuk membangun model klasifikasi Dan model statistik untuk menganalisisnya antara lain :

a. Model Klasifikasi: Mengkategorikan dokumen ke dalam kelas-kelas tertentu (misalnya, topik berita, sentimen pelanggan); Cosine Similarity, Naïve Bayes, K-Nearest Neighbour, SVM, NN, Regresi, Logistik, Decision Tree, C-45, Random Forest, dsb.

b. Model Clustering: Mengelompokkan dokumen yang memiliki kemiripan berdasarkan fitur yang diekstraksi; meliputi K-means, C-means, DBScan, Agglomerativ Clustering, dsb.

c. Model Modeling: Mengidentifikasi topik-topik yang mendasari kumpulan dokumen; meliputi Algoritma ARIMA, Support Vector Regression (SVR), Polynomial Regression, Regression Trees, Polynomial Regression, dsb.

Missal dari data normalisasi di atas kita hitung nilai kemiripan antar dokumen menggunakan Algoritma Cosine Similarity (CS) :

$$cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \sum_{i=1}^{n} B_i^2}}.$$

Gambar 12. 6. Formula Cosine Similarity

Metode CS digunakan untuk mengukur kemiripan antar dokumen berdasarkan representasi vektornya (hasil normalisasi TF-IDF maupun N-GRAM). Dan CS akan menghasilkan nilai antara 0 dan 1, di mana:

a. 0 menunjukkan tidak ada kemiripan antar dokumen.
b. 1 menunjukkan kemiripan sempurna antar dokumen.

Artinya, semakin tinggi nilai Cosine Similarity, semakin mirip isi kedua dokumen tersebut. Misal pada data TF-IDF (table 2), antara dua dokumen A dan B dihitung dengan rumus berikut :

CS(A,B) = Σ(tfidf_ai * tfidf_bi) / √Σ(tfidf_ai^2) * √Σ(tfidf_bi^2)

*tfidf_ai: Nilai TF-IDF kata i pada dokumen A
*tfidf_bi: Nilai TF-IDF kata i pada dokumen B

Cosine Similarity antara D1 dan D2:
CS(D1, D2) = (0.059 * 0) / √(0.059^2 + 0^2) * √(0^2 + 0^2) = 0

Cosine Similarity antara D1 dan D3:
CS(D1, D3) = (0.059 * 0.035 + 0 * 0 + 0 * 0 + 0 * 0.035 + 0 * 0.095) / √(0.059^2 + 0^2 + 0^2 + 0^2 + 0^2) * √(0.035^2 + 0^2 + 0^2 + 0.035^2 + 0.095^2) = 0.071

Cosine Similarity antara D2 dan D3:
CS(D2, D3) = (0 * 0.035 + 0 * 0 + 0 * 0 + 0.059 * 0.035 + 0 * 0.095) / √(0^2 + 0^2 + 0^2 + 0.059^2 + 0^2) * √(0.035^2 + 0^2 + 0^2 + 0.035^2 + 0.095^2) = 0.054

Langkah terakhir dari proses Text Mining adalah melakukan analisis dari hasil permdelannya, yaitu untuk membantu dalam pengambilan keputusan dan pemahaman konten dokumen dalam banyak aspek analisis teks dan pemrosesan bahasa alami. Dan dari perhitungan CS didapatkan kesimpulan sebagai berikut:

a. Dokumen D1 dan D2 memiliki kemiripan 0, hal ini menunjukkan tidak ada kesamaan diantaranya.

b. Dokumen D1 dan D3 memiliki kemiripan 0.071, menunjukkan sedikit kesamaan kata-kata yang sering muncul, terutama "kucing" dan "air".
c. Dokumen D2 dan D3 memiliki kemiripan 0.054, menunjukkan sedikit kesamaan kata-kata yang sering muncul, terutama "air".

Artinya Dokumen D1 dan D2 memiliki kemiripan paling rendah, sedangkan dokumen D1 dan D3 memiliki kemiripan yang lebih tinggi, meskipun masih tergolong rendah.

5. Analisa Network

Langkah-langkah yang tepat dan terarah pada Text Mining akan memberikan pengetahuan baru dari suatu data teks tidak terstruktur (raw), dan hal ini adalah penting dikarenakan didalam prosesnya kemampuan :

a. Mengubah data teks yang tidak terstruktur menjadi bentuk yang terstruktur.
b. Mengurangi noise dan inkonsistensi dalam data teks.
c. Mengidentifikasi pola dan hubungan dalam data teks
d. Mengubah data teks menjadi pengetahuan dan wawasan untuk berbagai tujuan (misalnya, memahami opini publik, meningkatkan riset pasar, mendeteksi penipuan, dsb).

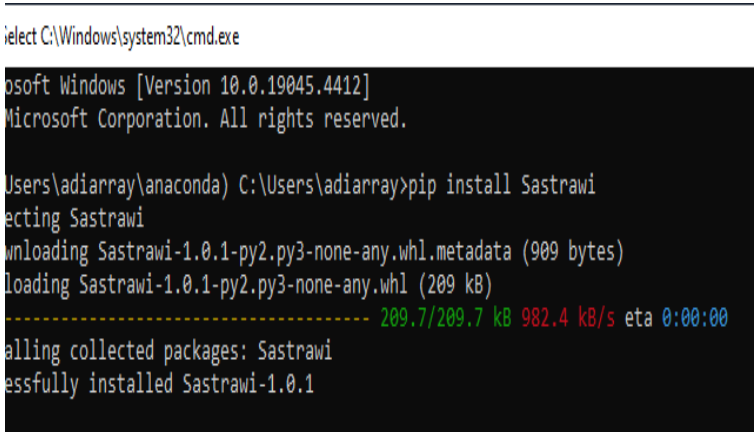
C. Implementasi

Melakukan riset tentang text mining tidak hanya memerlukan pemahaman teoritis, tetapi juga baiknya kita melakukan secara praktik langsung. Yaitu dengan mengaplikasikan konsep-konsep yang telah kita pelajari ke dalam situasi nyata, memvalidasi metode, serta mengembangkan dan menyempurnakan algoritma yang digunakan hingga diperoleh hasil maksimum, solusi praktis, dan kemampuan baik dalam menguji efektivitas suatu metode.

Dalam praktik ini kita menggunakan Bahasa Pemrograman Python. Meskipun bukan yang tercepat, namun sudah cukup efisien dalam menangani data berukuran besar melalui optimisasi dan penggunaan pustaka eksternal tanpa kehilangan performa. Dan jika belum ada library-library berikut, silakan ketik di cmd:

#pip install ntlk

#pip install Sastrawi



Gambar 12. 7. Hasil Crawling

Selanjutnya, jalankan script berikut pada editor python:

Tabel 12. 4. Contoh Script TF-IDF

```
import nltk
from nltk.tokenize import word_tokenize
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from sklearn.feature_extraction.text import TfidfVectorizer

# misalkan isi dari ke-3 dokumen
doc1 = "ikan disukai kucing"
doc2 = "air tersukai ikan"
doc3 = "kucing sebagai pemakan ikan menyukai air"
# Menggabungkan semua dokumen
all_docs = [doc1, doc2, doc3]

# Inisialisasi factory untuk stop word remover dan stemmer Indonesia
stopword_factory = StopWordRemoverFactory()
stemmer_factory = StemmerFactory()

# Membuat stop word remover dan stemmer
stopword_remover = stopword_factory.create_stop_word_remover()
stemmer = stemmer_factory.create_stemmer()

# Menghapus stop words, melakukan stemming, dan tokenisasi untuk setiap dokumen
preprocessed_docs = []
for doc in all_docs:
    # Menghapus stop words
    doc = stopword_remover.remove(doc)
    # Melakukan stemming
    doc = stemmer.stem(doc)
    # Tokenisasi
    tokens = word_tokenize(doc)
    preprocessed_docs.append(" ".join(tokens))

# Menghitung TF-IDF
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(preprocessed_docs)

# Menampilkan hasil
print("Dokumen Setelah Preprocessing:")
for idx, doc in enumerate(preprocessed_docs):
    print(f"Dokumen {idx+1}: {doc}")
```

```
print("\nTF-IDF:")
feature_names = tfidf_vectorizer.get_feature_names_out()
for idx, doc in enumerate(all_docs):
   feature_index = tfidf_matrix[idx, :].nonzero()[1]
   tfidf_scores = zip(feature_index, [tfidf_matrix[idx, x] for x in
feature_index])
   for w, s in [(feature_names[i], s) for (i, s) in tfidf_scores]:
      print(f"Dokumen {idx+1}, Kata: '{w}', TF-IDF: {s:.2f}")
```

Penjelasan script umum

Tahap 1: Import Library

Perintah mengimpor modul yang diperlukan, termasuk modul nltk untuk pemrosesan bahasa alami, Sastrawi untuk proses stopword dan stemming (Indonesia), serta TfidfVectorizer dari scikit-learn untuk TF-IDF.

Tahap 2: Initialasisasi Dokumen

Yaitu mendaftarakn list dokumen data yang lalu digabungkan dalam suatu variabel baru all_docs ([doc1, doc2, doc3]).

Tahap 3: Preprocessing

Artinya pada tahap ini data di setiap dokumen diubah menjadi huruf kecil, lalu proses penghapusan kata-dasar, angka dan tanda baca.Lanjut pada proses stemming atau pengembalian ke kata dasar dari setiap data dokumen, dan tahap ini diakhiri dengan pembentukan menjadi data array.

Tahap 4: TF-IDF

Yaitu tahap penghitungan teks secara matematis menggunakan modul TfidfVectorizer. Proses Ini menghasilkan matriks TF-IDF yang mencatat bobot dari setiap kata dalam setiap dokumen.

Tahap 5: Hasil

Hasil preprocessing selanjutnya ditampilkan, yaitu dengan menunjukkan bobot log dari kata-kata yang muncul dalam setiap dokumen beserta nilai TF-IDF-nya.

```
Dokumen Setelah Preprocessing:
Dokumen 1: ikan suka kucing
Dokumen 2: air suka ikan
Dokumen 3: kucing makan ikan suka air
```

Gambar 12. 8. Hasil Praproses Dokumen

Hasil pada gambar 12.8 menampilkan keluaran data pra-processing, yaitu data yang sudah mengalami proses filtering, casefolding, stopword stemming, lemmatization, dan tokenisasi (pembentukan matrik array / vector dengan menghilangkan kata yang terduplikasi).

```
Hasil tfidf Matrix
  (0, 2)        0.6732546652684398
  (0, 4)        0.5228423068642596
  (0, 1)        0.5228423068642596
  (1, 0)        0.6732546652684398
  (1, 4)        0.5228423068642596
  (1, 1)        0.5228423068642596
  (2, 3)        0.5918865885345992
  (2, 0)        0.45014500672563534
  (2, 2)        0.45014500672563534
  (2, 4)        0.3495777539781596
  (2, 1)        0.3495777539781596
```

Gambar 12. 9. Hasil TF-IDF

Gambar 12.9 menampilkan data hasil perhitungan tf-idf antar kata dalam suatu dokumen dan antar dokumen, yaitu dengan menggunakan formulasi gambar 12.5.

```
TF-IDF:
Dokumen 1, Kata: 'kucing', TF-IDF: 0.67
Dokumen 1, Kata: 'suka', TF-IDF: 0.52
Dokumen 1, Kata: 'ikan', TF-IDF: 0.52
Dokumen 2, Kata: 'air', TF-IDF: 0.67
Dokumen 2, Kata: 'suka', TF-IDF: 0.52
Dokumen 2, Kata: 'ikan', TF-IDF: 0.52
Dokumen 3, Kata: 'makan', TF-IDF: 0.59
Dokumen 3, Kata: 'air', TF-IDF: 0.45
Dokumen 3, Kata: 'kucing', TF-IDF: 0.45
Dokumen 3, Kata: 'suka', TF-IDF: 0.35
Dokumen 3, Kata: 'ikan', TF-IDF: 0.35
```

Gambar 12. 10. Hasil TF-IDF

Selanjutnya dihitung nilai Cosine Similarity sehingga menjadi:

Berikut adalah lanjutan dari script sebelumnya dengan penambahan perhitungan cosine similarity antara dokumen :

Tabel 12. 5. Contoh Script CS

```
from sklearn.metrics.pairwise import cosine_similarity

# Perhitungan cosine similarity antar dokumen
similarity_matrix = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Menampilkan similarity matrix
print("\nCosine Similarity Matrix:")
for i in range(len(all_docs)):
    for j in range(len(all_docs)):
        print(f"Similarity antara Dokumen {i+1} dan Dokumen {j+1}: {similarity_matrix[i][j]:.2f}")

# Contoh uji dengan kalimat lain
doc_new = "kucing dan ikan saling menyukai"
preprocessed_new_doc = stopword_remover.remove(doc_new)
preprocessed_new_doc = stemmer.stem(preprocessed_new_doc)
tokens_new_doc = word_tokenize(preprocessed_new_doc)

# Menghitung TF-IDF untuk kalimat baru
tfidf_new_doc = tfidf_vectorizer.transform([preprocessed_new_doc])
```

```
# Menampilkan hasil TF-IDF untuk kalimat baru
print("\nTF-IDF untuk Kalimat Baru:")
feature_names_new_doc =
tfidf_vectorizer.get_feature_names_out()
for i, feature in enumerate(feature_names_new_doc):
    tfidf_score = tfidf_new_doc[0, i]
    if tfidf_score > 0:
        print(f"Kata: '{feature}', TF-IDF: {tfidf_score:.2f}")

# Menampilkan similarity antara kalimat baru dan dokumen yang ada
print("\nSimilarity antara Kalimat Baru dan Dokumen yang Ada:")
for idx, doc in enumerate(all_docs):
    similarity = cosine_similarity(tfidf_new_doc, tfidf_matrix[idx])
    print(f"Similarity antara Kalimat Baru dan Dokumen {idx+1}:
{similarity[0][0]:.2f}")
```

Pada script Tabel 12.5, ditambahkan dengan sript perhitungan cosine similarity antar setiap pasangan dokumen, serta contoh pengujian atas kalimat baru untuk dianalisis kemiripannya dengan dokumen yang ada.

```
Cosine Similarity Matrix:
Similarity antara Dokumen 1 dan Dokumen 1: 1.00
Similarity antara Dokumen 1 dan Dokumen 2: 0.55
Similarity antara Dokumen 1 dan Dokumen 3: 0.67
Similarity antara Dokumen 2 dan Dokumen 1: 0.55
Similarity antara Dokumen 2 dan Dokumen 2: 1.00
Similarity antara Dokumen 2 dan Dokumen 3: 0.67
Similarity antara Dokumen 3 dan Dokumen 1: 0.67
Similarity antara Dokumen 3 dan Dokumen 2: 0.67
Similarity antara Dokumen 3 dan Dokumen 3: 1.00
```

Gambar 12. 11. Hasil CS

Model cosine_similarity yang diperoleh dapat digunakan untuk uji kalimat baru dan bisa diketahui nilai keterdekatannya dengan doc datalatih (Doc1, Doc2, Doc3), missal dibuatkan kalimat uji *doc_new* = "*kucing dan ikan saling menyukai*", maka diperoleh TF-IDF dan CS hasil sebagai berikut:

```
TF-IDF untuk Kalimat Baru:
Kata: 'ikan', TF-IDF: 0.52
Kata: 'kucing', TF-IDF: 0.67
Kata: 'suka', TF-IDF: 0.52

Similarity antara Kalimat Baru dan Dokumen yang Ada:
Similarity antara Kalimat Baru dan Dokumen 1: 1.00
Similarity antara Kalimat Baru dan Dokumen 2: 0.55
Similarity antara Kalimat Baru dan Dokumen 3: 0.67
```

Gambar 12. 12. Hasil Pengujian Kalimat Baru

Artinya, kalimat baru yang diujikan paling mirip dengan Doc1; bahkan karena memiliki kesamaan di semua kata-katanya dengan Doc1, maka bisa dikatakan kalimat uji adalah identic dengan Doc1 (CS 100%).

Dengan mengetahui script python terbuka (Tabel 4 dan Tabel 5) kita dapat memodifikasi kode / simulasi yang ada. Sehingga kita nantinya dapat melakukan penyesuaian terhadap data nyata yang seringkali tidak sempurna dan inkonsistensi (raw data). Juga untuk terus mengembangkan strategi-strategi yang terbaik dalam membersihkan dan menormalkan teks sebelum analisis lebih lanjut.

D. Rangkuman

Text Mining, bagaikan kunci ajaib yang membuka gerbang informasi tersembunyi di dalam segunung data teks tidak terstruktur. Text Mining merupakan alat yang revolusioner menuju pemahaman data di sekitar kita dengan lebih baik. Terlebih dengan pengembangan pada algoritma clustering, modelling, dan klasifikasi semakin membuka peluang besar dalam penyelesaian berbagai masalah pertanyaan kompleks.

Untuk selanjutnya, proses Text Mining bisa disederhanakan menjadi tiga tahap utama :

1. ***Praprocessing***

   Yaitu diawali dengan membersihkan data teks dari segala noise data dan ketidakteraturan, mulai dari karakter Non ASCII hingga tag HTML, serta proses menghilangkan

kata bantu, tanda baca dan angka-angka yang mungkin ada. Pada praproses juga dilakukan stemming dan lemmatization untuk mengubah kata-kata menjadi bentuk dasarnya, sehingga meminimalkan variasi dan meningkatkan akurasi analisis. Dan diakhiri dengan Proses tokenisasi, di mana kalimat-kalimat dibagi menjadi unit-unit yang lebih kecil dalam satuan kata unik.

2. ***Modelling:***

   Langkah ini adalah memilih metode pemodelan yang sesuai dengan kebutuhan. Berbagai metode tersedia, mulai dari konsep Entrioy Term, TF-IDF, hingga N-grams, yang masing-masing dengan kelebihan dan kekurangannya. Setelah memilih metode, langkah berikutnya adalah melatih model berdasarkan data teks yang telah dipreproses sebelumnya.Pemodelan bisa diperuntukkan pada proses klasifikasi, clustering maupun prediksi.

3. ***Analisis dan Interpretasi***

   Terakhir adalah tahap menganalisis hasil dari model yang telah dilatih untuk memahami pola dan tren yang teridentifikasi. Lalu hasil diinterpretasikan berupa penjelasan hubungan data dan atau makna analisisnya. Dengan menjalani ketiga tahap ini dengan hati-hati dan cermat, kita dapat menggali wawasan berharga dari data teks yang kompleks dan bermanfaat untuk pengambilan keputusan yang lebih baik.

DAFTAR PUSTAKA

Sarkar, D., Bali, R., & Sharma, T. (2019). Text Analytics with Python: A Practitioner's Guide to Natural Language Processing (2nd ed.). Apress.

Santoso, T. (2019). Text Mining: Teori dan Aplikasi. Informatika Bandung.

Mulyono, D. (2019). Analisis Teks dengan Pendekatan Text Mining. Deepublish.

Aggarwal, C. C., & Zhai, C. (2019). Mining Text Data. Springer.

Handayani, T. (2020). Text Mining untuk Analisis Data Teks. Penerbit Andi.

Wulandari, F. (2020). Text Mining dan Penerapannya dalam Penelitian. Graha Ilmu.

Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R., & Delen, D. (2020). Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications (2nd ed.). Academic Press.

Tang, J., & Liu, H. (2020). Graph Mining: Laws, Tools, and Case Studies. Cambridge University Press.

Putra, A. R. (2021). Pengantar Text Mining dan Analisis Sentimen. Pustaka Pelajar.

Findawati, Y., & Rosid, M. A. (2021). Buku Ajar Text Mining. Umsida Press.

TENTANG PENULIS

Dr. Jonathan Kiwasi Wororomi, S.Si., M.Si.
Penulis adalah dosen tetap Program Studi Statistika (S1) Fakultas Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Cenderawasih. Penulis menyelesaikan Pendidikan Program Sarjana (S1) Ilmu Matematika di Universitas Sebelas Maret, Surakarta (1998) kemudian melanjutkan Program Pasca Sarjana (S2) Ilmu Matematika di Institut Teknologi Bandung (2006) dan pada Tahun 2016 menyelesaikan program Doktor di Institut Teknologi Surabaya pada Bidang Statistik. Pernah menjabat sebagai Koordinator Pusat Studi Satistika (PSS) Uncen 2007-2011, Wakil Dekan III Fakultas MIPA Universitas Cenderawasih tahun 2007-2009, dan menjabat sebagai Wakil Rektor III Universitas Cenderawasih (2017-2024). Mengampuh beberapa Mata Kuliah pada Program S1 dan S2 setiap semester diantaranya; Quality Control Statistics, Macroeconomics, Time Series Analysis, Categorical Data Analysis and Socio-Entrepreneurship.

Felix Reba, S.Si., M.Sc.
Lulus S1 di Program Studi Matematika FMIPA Universitas Cenderawasih tahun 2008. Lulus S2 di Program Master of Science, Universitas Gadjah Mada tahun 2017. Saat ini adalah dosen tetap Program Studi Matematika, FMIPA, Universitas Cenderawasih. Mengampu mata kuliah Data Mining, Algoritma dan Pemrograman, Komputasi Statistik dan Kecerdasan Buatan. Pernah menjadi pengajar pada Program Digital Talent Scholarship (DTS) untuk Fresh Graduate Academy (FGA) Kemkominfo Jakarta Pusat. Penulis juga aktif menulis Jurnal Nasional, Internasional, Buku dan juga beberapa kali menjadi Narasumber pada Himpunan

Matematika Indonesia (IndoMS). **Informasi Kontak** Gmail : felix.reba85@gmail.com

Samuel Aleksander Mandowen, S.Si., M.IT.
Penulis lahir pada tanggal 24 Maret 1980 di Ababiadi (Supiori Selatan) Kabupaten Supiori, Provinsi Papua. Pendidikan; SD Inpres Kansai, SMPN Numfor Barat dan SMA Negeri Numfor Barat, Kab. Biak Numfor, Prov. Papua. Pendidikan S1 (Strata Satu) Matematika di Universitas Cenderawasih, Jayapura, Papua. Tahun 2011 mendapatkan Beasiswa Australian Development Scholarship (ADS)/ Australian Awards Scholarship (AAS) melanjutkan Pendidikan S2 di Bidang Information Technology *(Network Management)* di Queensland University of Technology Brisbane, Australia. Sedang melanjutkan pendidikan Doktor/S3 (PhD) di bidang Information Security di Ufa University of Science and Technology *(Уфимский Университет Науки и Технологий)* Russia, masuk tahun 2022. Sejak Tahun 2006 diangkat sebagai PNS (Dosen) pada Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Cenderawasih (UNCEN) dan sekarang homebase pada Program Studi Sistem Informasi FMIPA UNCEN

Alvian M. Sroyer, S.Si., M.Si.
Penulis. Lulus S1 di Program Studi Matematika FMIPA Universitas Cenderawasih tahun 2004. Lulus S2 di Program Magister, Institut Teknologi Bandung tahun 2009. Saat ini adalah dosen tetap Program Studi Matematika, FMIPA, Universitas Cenderawasih, dan sedang mendapatkan tugas tambahan sebagai Wakil Dekan III FMIPA Universitas Cenderawasih. Buku yang telah ditulis dan terbit berjudul di antaranya: *Machine Learning, Metode Penelitian kuantitatif & Aplikasi Pengolahan Analisa Data Statistik,Big Data Analytics :*

Halomoan Edy Manurung, S,Si., M.Cs.
Lahir di Kabupaten Biak, Provinsi Papua pada tanggal 10 januari 1984. Riwayat pendidikan penulis adalah SD dan SMP di Biak dan melanjutkan SMA di Nabire. Menyelesaikan Sarjana Sains pada Program Studi Matematika, Fakultas Matematika dan ilmu Pengetahuan Alam Universitas Cenderawasih pada tahun 2002 dan pada tahun 2010 melanjutkan Studi Magister Sistem Informasi Universitas Kristen Satya Wacana Salatiga serta pada tahun 2020 melanjutkan studi Doktoral pada ilmu komputer Universitas Dian Nuswantoro Semarang hingga saat ini. Penulis saat ini adalah Dosen di Program Studi Sistem Informasi pada Universitas Ottow Geissler Papua. Pengalaman dan keterampilan atau keahlian penulis adalah pada Research Di Bidang Trends Of Data Mining And Statistical Data, Computer Vision, Developer Web And Application, Data Scientist, Digital Visualization And 3d Animation.

Mayko Edison Koibur, S.T., M.Eng.
Penulis adalah Aparatur Sipil Negara (ASN) yang sudah menghabiskan masa pengabdian 23 Tahun sebagai ASN sejak tahun 2000 dan mengawali karier di Pemerintah Kota Jayapura, Kabupaten Sarmi dan saat ini bekerja di Pemerintah Provinsi Papua. Mengawali Pendidikan Diploma Komputer Tahun 1996, Lalu mendapatkan Beasiswa dari Yayasan Supersemar untuk Program 1- Ilmu Komputer. Kemudian melanjutkan Strata Dua (S2) di Jurusan Teknik – Elektro – Universitas Gaja Mada atas dukungan Beasiswa dari Kementerian Komunikasi & Informatika RI, dan saat ini melanjutkan Studi Doctoral Computer Science di Fukuoka – Jepang dengan Beasiswa dari Pemerintah Provinsi

Papua. Beberapa Kegiatan Training dalam Bidang Teknologi Informasi, baik dari Pemerintah Pusat & Daerah serta Lembaga Internasional dalam Program Peningkatan Kapasitas ASN dari JICA – Jepang, Selama bekerja di Pemerintahan telah terlibat secara langsung dalam beberapa proyek pengembangan data & informasi, dan juga sebagai tercatat sebagai dosen di Universitas Ottow Geissler Papua pada jurusan Sistem Informasi. Penulis lahir di Biak, 30 Juni 1977, saat ini memiliki seorang Istri dan Empat Orang Anak.

Erna Hudianti pujiarini, S.Si., M.Si.
Dosen tetap Prodi Informatika Fakultas Teknologi Informasi Universitas Teknologi Digital Indonesia. Lahir di Magetan, 28 September 1971. Penulis merupakan anak Ketiga dari empat bersaudara dari pasangan bapak Soewandi dan Ibu Sutarti. Pendidikan program Serjana (S1) Universitas Gadjah Mada Prodi Statistika dan menyelesaikan program Pasca Sarjana (S2) di Universitas Gadjah Mada prodi Matematika. *Matakuliah yang diampu Statistika, Statistika Terapan, Matematika, Data Mining, Jaringan Syarat Tiruan. Bidang konsentrasi penelitian Statistika Terapan dan Data Science.*

Marwan Ramdhany Edy, S.Pd., M.Kom.
Seorang penulis dan dosen tetap Prodi Teknik Komputer Fakultas Teknik Universitas Negeri Makassar. Lahir di Pangkep, 6 Maret 1992. Penulis merupakan anak Pertama dari empat bersaudara dari pasangan bapak Edy Sabara dan Hasmawati. Pendidikan program Serjana (S1) Universitas Negeri Makassar Prodi Pendidikan Teknik Informatika dan Komputer dan menyelesaikan program Pasca Sarjana (S2) di Institut Pertanian Bogor (IPB) prodi Ilmu Komputer konsentrasi di bidang Data Science. Buku yang telah ditulis dan terbit berjudul di

antaranya: Transformasi Pendidikan Mendorong Kemajuan Bangsa melalui Kecerdasan Buatan.

Hajiar Yuliana, S.T., M.T.

Hajiar Yuliana adalah seorang akademisi dan peneliti yang berdedikasi dalam bidang teknik telekomunikasi dan data mining. Ia lahir di Bekasi pada tanggal 13 Juli 1989 dan saat ini berprofesi sebagai dosen di Universitas Jendral Achmad Yani (UNJANI), Cimahi. Hajiar memiliki latar belakang pendidikan yang kuat dalam teknik telekomunikasi, dengan gelar Master dari Institut Teknologi Bandung (ITB) dan gelar Sarjana dari Universitas Jendral Achmad Yani (UNJANI). Saat ini, beliau sedang menempuh pendidikan doktoralnya di bidang Teknik Telekomunikasi di Institut Teknologi Bandung. Topik penelitiannya saat ini adalah pengembangan teknologi telekomunikasi khususnya dibidang jaringan seluler/wireless dengan memanfaatkan penggunaan algoritma *machine learning*. Hajiar Yuliana memiliki pengalaman luas dalam industri telekomunikasi dan bekerja di berbagai perusahaan telekomunikasi di Indonesia sejak 2010. Selama karirnya, Hajiar telah terlibat dalam berbagai proyek optimasi jaringan seluler dan perencanaan jaringan, termasuk proyek-proyek penting seperti optimasi jaringan LTE dan perencanaan jaringan 5G. Di bidang penelitian, Hajiar Yuliana telah mempublikasikan banyak karya ilmiah di jurnal dan konferensi internasional. Penelitiannya mencakup berbagai topik. Publikasinya yang terakhir menganalisis prediksi coverage pada jaringan 5G dengan menggunaakan algoritma machine learning telah dipublikasi international terindeks bereputasi Q1 (IEEE Access). Sebagai dosen dan peneliti, Hajiar terus berkontribusi dalam pengembangan ilmu pengetahuan dan teknologi melalui penelitian dan pengajaran. Ia juga aktif dalam berbagai organisasi profesional dan sering menjadi pembicara pada seminar dan workshop di bidang telekomunikasi. Dengan kombinasi keahlian praktis dan akademisnya, Hajiar Yuliana berkomitmen untuk terus

berinovasi dan memberikan kontribusi signifikan dalam bidang telekomunikasi dan data mining.

Mukarramah Yusuf, B.Sc., M.Sc.
Lahir dan besar di Sulawesi Selatan. Setelah mendapat gelar S2 dari Ochanomizu University di Jepang, mengajar di Prodi Teknik Informatika Universitas Hasanuddin. Pemegang sertifikat AWS Cloud Computing Practitioner, Red Hat Cloud Computing with Ansible. Penulis menggunakan analisa time series untuk disertasi program doktoralnya.

Riadi Marta Dinata, S.Ti., M.Kom.
Seorang penulis dan dosen tetap Prodi Teknik Informasi Fakultas Sains dan Teknologi Informasi ISTN Jakarta. Penulis sudah lama berkecimpung dalam dunia IT baik pemrograman, networking, Embedeed System hingga bidang Kecerdasan Buatan terapan. Diawali dengan jenjang Pendidikan Elektronika (D3), Teknik Informatika (S1), Ilmu Komputer (S2) dan kini tengah menjalani program doktoral di Kampus UNILA peminatan Ilmu Komputer. Selain itu penulis juga aktif dalam kegiatan workshop/ training tentang IT, webinar/seminar tentang IT dan kegiatan penulisan-penulisan / riset di kampus. Penulis juga selain sebagai pendiri, juga aktif sebagai pengajar di StartUp IT Lp2maray (From Zero to Hero) Jakarta sejak tahun 2001. Silakan menghubungi penulis di adiarray@istn.ac.id.