MATEMATIKA PADA KECERDASAN BUATAN

Bab 1

Peran Penting Matematika dalam Kecerdasan Buatan

Matematika merupakan fondasi fundamental yang menopang perkembangan dan aplikasi kecerdasan buatan (AI) di berbagai bidang. Dengan memahami konsep-konsep matematika, seperti aljabar linear, kalkulus, teori probabilitas, dan statistik, dapat dikembangkan algoritma yang memungkinkan mesin untuk belajar dari data, membuat prediksi, dan mengambil keputusan yang optimal (Chatterjee, 2020; Jia, Mingyang and Yueen, 2023). Aljabar linear, misalnya, memainkan peran penting dalam representasi data dan operasi matriks yang digunakan dalam *neural networks*, Kalkulus digunakan untuk mengoptimalkan fungsi biaya melalui metode seperti *gradient descent*, yang esensial dalam pelatihan model AI (Anwarudin *et al.*, 2022; Fauzi *et al.*, 2022; Iriyanta, Putranto and Andriyani, 2023). Teori probabilitas dan statistik memungkinkan untuk menangani ketidakpastian dan membuat model prediktif yang akurat. Teknik pengoptimalan membantu menemukan solusi terbaik dalam berbagai masalah kompleks yang dihadapi dalam AI. Dengan integrasi matematika, AI dapat diterapkan dalam berbagai sektor, termasuk kesehatan, transportasi, keuangan, dan teknologi informasi, membawa inovasi dan efisiensi yang signifikan (Berman, Chubb and Williams, 2023; Kutyniok, 2023; Čajić, 2024).

1.1 Pemodelan dan Representasi Data

Matematika menyediakan kerangka kerja untuk memodelkan dan merepresentasikan data dalam bentuk yang dapat diproses oleh komputer, Kerangka kerja matematika dalam AI tak hanya memungkinkan komputer memahami pola, tren, dan hubungan dalam data, tetapi juga membuka gerbang bagi berbagai aplikasi canggih. Algoritma klasifikasi, prediksi, clustering, dan deteksi anomali, yang mendasari banyak aplikasi AI, dimungkinkan dengan adanya kerangka kerja ini (Berman, Chubb and Williams, 2023). Contohnya, algoritma klasifikasi, yang dilatih dengan data berlabel, mampu mengidentifikasi gambar kucing dan anjing dengan presisi tinggi. Algoritma prediksi, dengan menganalisis data historis, dapat memprediksi harga saham atau mendeteksi risiko penipuan. Clustering membantu mengelompokkan pelanggan berdasarkan kebiasaan berbelanja, sehingga memungkinkan strategi pemasaran yang lebih sesuai target. Deteksi anomali, berperan penting dalam memerangi penipuan kartu kredit dan mencegah kegagalan mesin. Kinerja model AI sangat bergantung pada kualitas pemodelan dan representasi data (Ali *et al.*, 2023).

Semakin baik data dimodelkan, semakin akurat dan andal pula hasil yang dihasilkan. Aljabar linier, kalkulus, dan statistika menjadi alat utama untuk memodelkan data. Aljabar linier merepresentasikan data sebagai vektor dan matriks, memungkinkan operasi matematika kompleks. Kalkulus membantu menganalisis perubahan data dari waktu ke waktu, seperti memprediksi tren harga saham. Statistika berperan dalam meringkas data, mengidentifikasi pola, dan mengukur ketidakpastian (Anwarudin *et al.*, 2022; Fauzi *et al.*, 2022; Iriyanta, Putranto and Andriyani, 2023). Pemilihan alat dan teknik yang tepat untuk memodelkan data bergantung pada jenis data, tugas yang ingin dicapai, dan keterbatasan komputasi. Para peneliti terus mengembangkan metode matematika baru untuk meningkatkan kemampuan AI dalam memodelkan dan merepresentasikan data, membuka jalan bagi aplikasi AI yang lebih canggih dan bermanfaat di masa depan (Friedrich *et al.*, 2022).

a. Aljabar linier, kalkulus, dan statistik tak hanya berperan penting dalam pemodelan dan representasi data, tetapi juga dalam proses transformasi data, ekstraksi fitur, dan reduksi dimensi (Anwarudin *et al.*, 2022; Fauzi *et al.*, 2022; Iriyanta, Putranto and Andriyani, 2023). Ketiga cabang matematika ini menjadi landasan bagi berbagai teknik yang digunakan untuk mempersiapkan data agar dapat diolah dengan optimal oleh model AI. Transformasi data bertujuan untuk mengubah format data agar kompatibel dengan algoritma AI. Aljabar linier, dengan operasi matriksnya, memungkinkan transformasi data seperti rotasi, perubahan skala dan translasi. Kalkulus, di sisi lain, membantu dalam transformasi data yang melibatkan perubahan nilai secara kontinu, seperti normalisasi data. Ekstraksi fitur bertujuan untuk mengidentifikasi karakteristik penting dari data yang relevan dengan tugas yang ingin dicapai. Teknik-teknik seperti *Principal Component Analysis* (PCA) dan *Singular Value Decomposition* (SVD) yang berdasarkan aljabar linier dan kalkulus, membantu dalam mengekstraksi fitur yang paling informatif dari data, sehingga mengurangi dimensi data dan meningkatkan efisiensi komputasi (Soori, Arezoo and Dastres, 2023). Reduksi dimensi bertujuan untuk mengurangi jumlah variabel dalam data tanpa kehilangan informasi penting. Teknik-teknik seperti *Linear Discriminant Analysis* (LDA) dan t-distributed Stochastic Neighbor Embedding (t-SNE) yang didasarkan pada statistik, membantu dalam mereduksi dimensi data sambil mempertahankan struktur dan pola data yang relevan. Proses transformasi data, ekstraksi fitur, dan reduksi dimensi yang efektif sangat penting untuk meningkatkan performa model AI. Dengan memanipulasi data dengan tepat, model AI dapat belajar dari data lebih mudah, menghasilkan prediksi yang lebih akurat, dan meningkatkan *generalizability* model. Penelitian dan pengembangan dalam bidang matematika terus dilakukan untuk mengembangkan teknik transformasi data, ekstraksi fitur, dan reduksi dimensi yang lebih canggih dan efisien. Hal ini membuka peluang baru untuk meningkatkan performa model AI dan memperluas jangkauan aplikasinya di berbagai bidang (Broekhuizen *et al.*, 2023).

b. Teori probabilitas dan statistika memungkinkan AI untuk memproses data yang tidak pasti dan membuat prediksi dengan tingkat kepercayaan yang terukur, Ketidakpastian merupakan hal yang inheren dalam data di dunia nyata. Hal ini menghadirkan tantangan bagi AI dalam memproses data dan membuat prediksi yang akurat. Di sinilah teori probabilitas dan statistika memainkan peran penting (Friedrich *et al.*, 2022; Seoni *et al.*, 2023). Teori probabilitas memberikan kerangka kerja untuk mengukur kemungkinan terjadinya suatu peristiwa. Dengan menggunakan probabilitas, AI dapat memproses data yang tidak pasti dan membuat prediksi dengan tingkat kepercayaan yang terukur. Contohnya, dalam klasifikasi gambar, AI dapat memperkirakan kemungkinan suatu gambar adalah kucing dengan nilai probabilitas tertentu, bukan hanya memberikan jawaban ya atau tidak (Pratomo and Andriyani, 2023). Statistika, di sisi lain, menyediakan alat untuk menganalisis dan menginterpretasikan data yang tidak pasti. Alat-alat seperti distribusi probabilitas dan interval kepercayaan memungkinkan AI untuk mengukur variabilitas data dan membuat prediksi yang lebih andal (Broekhuizen *et al.*, 2023; Soori, Arezoo and Dastres, 2023; Mehdiyev, Majlatow and Fettke, 2024).

Pengembangan teknik probabilistik dan statistikal dalam AI terus dilakukan. Hal ini memungkinkan AI untuk menangani ketidakpastian dengan lebih baik, membuat prediksi yang lebih canggih, dan beroperasi di dunia nyata yang penuh dengan ketidakpastian.

1.2 Pembelajaran Mesin dan Algoritma AI

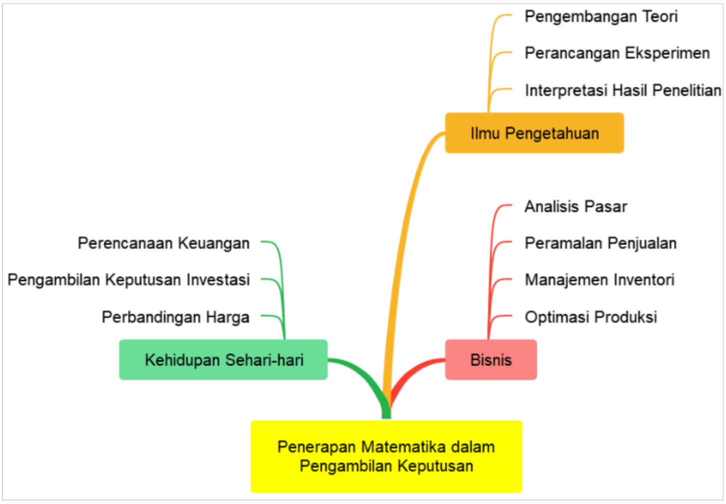
Algoritma pembelajaran mesin, seperti regresi linier, pohon keputusan, dan jaringan saraf tiruan, berdasarkan pada konsep matematika yang kuat. Kalkulus dan optimasi digunakan untuk melatih dan menyempurnakan model AI, meminimalkan kesalahan, dan meningkatkan akurasi prediksi (Sarker, 2021). Teori informasi dan komputasi memungkinkan AI untuk memproses informasi secara efisien dan menangani tugas-tugas kompleks. Dengan pemahaman mendalam tentang matematika, model AI dapat dikembangkan dan dioptimalkan untuk memberikan hasil yang lebih akurat dan andal, sehingga mempermudah pengambilan keputusan berbasis data di berbagai bidang (Weinan *et al.*, 2020; (Argisitawan *et al.*, 2022; Muhrial *et al.*, 2022; Kufel *et al.*, 202; Semadi, Kristomo and Purnomosidi, 2023).

Gambar 1.1 Pembelajaran Mesin dan Algoritma Kecerdasan Buatan

1.3 Pengambilan Keputusan dan Penalaran

1. Matematika sebagai Landasan Pengambilan Keputusan

   Matematika, bahasa universal yang mendasari pengambilan keputusan, memungkinkan kita untuk menggali makna tersembunyi dalam data yang kompleks melalui statistik dan probabilitas. Dengan demikian, kita dapat mengidentifikasi pola, tren, dan hubungan sebab-akibat yang mungkin tidak terlihat sekilas. Matematika membekali dengan kemampuan untuk menyederhanakan situasi rumit menjadi model-model matematika yang dapat diprediksi (Kufel *et al.*, 2023). Model-model ini menjadi alat yang ampuh untuk mengoptimalkan sumber daya, meminimalkan risiko, dan memaksimalkan hasil dalam berbagai aspek kehidupan, mulai dari bisnis hingga penelitian ilmiah. Dengan kemampuan menganalisis data dan membuat model, matematika memungkinkan untuk mengevaluasi berbagai alternatif secara rasional dan memilih keputusan terbaik berdasarkan bukti-bukti yang ada (Taherdoost and Mitra Madanchian, 2023).



Gambar 1.2 Matematika pada Pengambilan Keputusan

Matematika sering digunakan sebagai dasar dalam pengambilan keputusan, terutama dalam situasi yang melibatkan analisis data, perhitungan probabilitas, dan optimasi (Mazur, 2006; Andi, Andriyani and Putranto, 2023; Roh Bintang Jaya *et al.*, 2024). Berikut adalah contoh penggunaan matematika pada Analisis Risiko dalam Investasi. Masalah yang terjadi adalah: seorang investor ingin memutuskan portofolio investasi terbaik yang meminimalkan risiko dan memaksimalkan pengembalian. Matematika yang digunakan adalah: teori portofolio modern (*Modern Portfolio Theory*) menggunakan statistik dan aljabar linear untuk menentukan komposisi optimal dari berbagai aset dalam portofolio. Langkah-langkah:

1. Menghitung Pengembalian Harian:

$$R_i = \frac{p_{t+1} - p_t}{p_t} \tag{1.1}$$

Dengan R_i adalah pengembalian aset i, p_{t+1}adalah harga pada waktu matematika sebagai alat untuk melatih penalaran $t + 1$, dan p_t adalah harga pada waktu t.

2. Menghitung Rata-rata Pengembalian:

$$\bar{R}_i = \frac{1}{n}\sum_{t=1}^{n} R_{i,t} \tag{1.2}$$

Dengan $\bar{R}_i$ adalah rata-rata pengembalian aset i dan n adalah jumlah periode waktu.

3. Menghitung Variansi dan Kovariansi:

$$\sigma_i^2 = \frac{1}{n-1}\sum_{t=1}^{n}\left(R_{i,t} - \bar{R}_i\right)^2 \tag{1.3}$$

$$Cov\left(R_i R_j\right) = \frac{1}{n-1}\sum_{t=1}^{n}\left(R_{i,t} - \bar{R}_i\right)\left(R_{j,t} - \bar{R}_j\right) \tag{1.4}$$

4. Menghitung Matriks Kovariansi:

$$\sum = \begin{bmatrix} \sigma_1^2 & Cov(R_1, R_2) & Cov(R_1, R \\ Cov(R_2, R_1) & \sigma_2^2 & Cov(R_2, R \\ Cov\ (R_n, R_1) & Cov\ (R_n, R_2) & \sigma_n^2 \end{bmatrix} \quad (1.5)$$

5. Optimasi:

   Menggunakan optimasi kuadratik untuk menentukan bobot w yang meminimalkan risiko

$$min_w = w^T \sum w \qquad (1.6)$$

   dengan syarat,

$$\sum_{i=1}^{n} w_i = 1 \qquad (1.7)$$

2. Matematika sebagai alat untuk melatih penalaran

   Matematika, lebih dari sekadar angka dan rumus, matematika adalah sebuah alat yang ampuh untuk mengasah kemampuan penalaran. Dalam setiap langkah penyelesaian masalah matematika, kita dilatih untuk berpikir secara logis, menyusun argumen yang kuat, dan menarik kesimpulan yang valid berdasarkan bukti-bukti yang ada. Melalui latihan soal yang beragam, kita diajak untuk tidak hanya mencari jawaban yang benar, tetapi juga untuk mengembangkan strategi pemecahan masalah yang efektif dan efisien (Kumar Yadav, 2021; Bulut and Kuzu, 2023). Lebih dari itu, matematika mendorong untuk berpikir kreatif, mencari cara-cara baru dan inovatif untuk mengatasi tantangan yang kompleks. Dengan kata lain, matematika tidak hanya mengajarkan kita untuk berpikir, tetapi juga untuk berpikir lebih baik (Andriyani *et al.*, 2024). Salah satu implementasi pada penerapan matematika dalam penalaran pemrograman yaitu dengan membuat algoritma dan program komputer memerlukan pemahaman yang kuat tentang logika matematika pada Algoritma Dijkstra untuk Pencarian Jalur Terpendek.

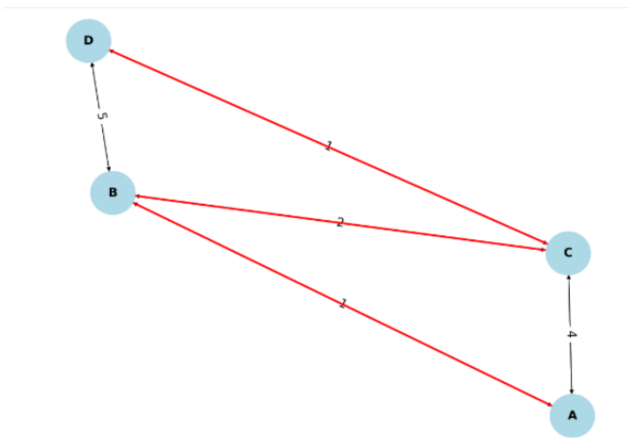
Masalah yang akan dicari solusinya yaitu: menemukan jalur terpendek dari satu node ke node lainnya dalam graf berbobot.

Logika Matematika yang digunakan adalah: Algoritma Dijkstra dengan menggunakan konsep graf dan bobot untuk menentukan jalur terpendek. Ini melibatkan pemahaman tentang teori graf dan algoritma Greedy.

Implementasi dengan pemrograman:

```
def dijkstra(graph, start):
  heap = [(0, start)]
  distances = {node: float('infinity') for node in graph}
  distances[start] = 0
      while heap:
      current_distance, current_node = heapq.heappop(heap)
      if current_distance > distances[current_node]:
        continue
      for neighbor, weight in graph[current_node].items():
        distance = current_distance + weight
        if distance < distances[neighbor]:
          distances[neighbor] = distance
          heapq.heappush(heap, (distance, neighbor))
```

```
    return distances
# Contoh penggunaan
graph = {
        'A': {'B': 1, 'C': 4},
        'B': {'A': 1, 'C': 2, 'D': 5},
        'C': {'A': 4, 'B': 2, 'D': 1},
        'D': {'B': 5, 'C': 1}
}
start_node = 'A'
print(dijkstra(graph, start_node))  # Output: {'A': 0, 'B': 1, 'C': 3, 'D': 4}
```



Gambar 1.3 Jalur Terpendek Algoritma Dijkstra

Graf yang menunjukkan jalur terpendek menggunakan algoritma Dijkstra. Node yang disorot dengan warna merah menunjukkan jalur terpendek dari node 'A' ke semua node lainnya dalam graf. Jalur ini dihasilkan oleh algoritma Dijkstra, yang memastikan bahwa total bobot jalur dari node awal ke node tujuan adalah yang terkecil.

Matematika bukan hanya sekumpulan rumus dan angka, tetapi juga merupakan alat yang sangat berharga untuk mengembangkan kemampuan berpikir kritis, analitis, dan kreatif. Dengan menguasai matematika, kita dapat membuat keputusan yang lebih baik, memecahkan masalah yang lebih kompleks, dan menghadapi tantangan hidup dengan lebih percaya diri

1.4 Pemrosesan Bahasa Alami (NLP)

Matematika memainkan peran yang sangat penting dalam Pemrosesan Bahasa Alami (NLP), salah satu kontribusi utamanya adalah dalam representasi teks dan dokumen, yaitu konsep seperti vektor kata (*word vectors*) dan *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk menangkap makna dan hubungan antar kata dalam bentuk numerik (Triaji *et al.*, 2022). Model probabilistik seperti model N-gram dan *Latent Dirichlet Allocation* (LDA) membantu dalam memprediksi kata berikutnya dalam kalimat dan menemukan topik dalam kumpulan dokumen dengan menggunakan distribusi probabilitas (Ferreira, 2022; Mohana, 2024).

Term Frequency-Inverse Document Frequency (TF-IDF):

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{1.8}$$

Dengan:

$$TF(t,d) = \frac{Jumlah\ kemunculan\ t\ dalam\ dokumen\ d}{Jumlah\ kemunculan\ t\ dalam\ dokumen\ d} \tag{1.9}$$

$$IDF(t) = log\left(\frac{Jumlah\ total\ dokumen}{Jumlah\ dokumen\ yang\ mengandung\ t}\right) \tag{1.10}$$

Statistik dan teori informasi juga sangat berperan dalam NLP, dengan konsep entropi dan informasi mutual yang digunakan untuk mengukur ketidakpastian dan korelasi antara variabel dalam teks. Model Markov Tersembunyi (*Hidden Markov Models*, HMMs) sering digunakan untuk tugas-tugas seperti penandaan bagian kata (*part-of-speech tagging*) dan pengenalan entitas bernama (*named entity recognition*). Dalam aljabar linear, penggunaan matriks dan vektor sangat penting dalam teknik seperti *singular value decomposition* (SVD) dan *principal component analysis* (PCA) untuk mengurangi dimensi data teks dan menemukan struktur penting.

Statistik dan Teori Informasi:

$$Entropi: H(X) = -\sum_{i} p\,(x_i) \log\ p\,(x_i) \tag{1.11}$$

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P\,(x,y) \log \left(\frac{P(x,y)}{P(x)P(y)} \right) \tag{1.12}$$

Optimasi dan pembelajaran mesin juga merupakan bidang matematika yang krusial dalam NLP. Algoritma optimasi seperti *gradient descent* digunakan dalam pelatihan model pembelajaran mesin, termasuk jaringan saraf dalam (*deep learning*). Teknik klasifikasi membantu memprediksi label atau kelas dari teks berdasarkan fitur yang diekstraksi. Teori graf dan jaringan saraf tiruan (*neural networks*) memberikan fondasi untuk pengembangan model NLP yang canggih. Graf semantik digunakan untuk menghubungkan konsep-konsep dalam teks melalui hubungan graf, sementara jaringan saraf tiruan seperti RNN (*Recurrent Neural Networks*) dan Transformer digunakan dalam tugas-tugas seperti pemahaman bahasa, terjemahan mesin, dan analisis sentimen.

Algoritma A*:

$$f(n) = g(n) + h\ (n) \tag{1.13}$$

Neural Networks (Perceptron Sederhana):

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1.14}$$

Backpropagation:

$$\Delta w_{ij}(t+1) = \eta \delta_j O_i \tag{1.15}$$

Dengan menggabungkan berbagai konsep matematis ini, NLP dapat mengembangkan model yang efektif untuk memahami, menganalisis, dan menghasilkan bahasa manusia. Matematika memungkinkan pemrosesan bahasa alami menjadi lebih akurat, efisien, dan dapat diandalkan.

1.5 Visi Komputer dan Robotika

Matematika memainkan peran yang sangat penting dalam Visi Komputer dan Robotika. Dalam visi komputer, matematika digunakan untuk berbagai tugas seperti pemrosesan gambar, pengenalan pola, dan analisis adegan. Aljabar linear, misalnya, digunakan untuk manipulasi gambar dan transformasi geometris, seperti rotasi, penskalaan, dan translasi, yang memungkinkan transformasi gambar. Fourier Transform membantu dalam analisis frekuensi gambar, yang penting untuk pengenalan pola dan penghilangan noise, sedangkan teknik konvolusi digunakan dalam *Convolutional Neural Networks* (CNNs) untuk ekstraksi fitur dari gambar (Shapiro and Stockman, 2000; Lámer, Cymbalak and Jakab, 2013; Che *et al.*, 2024).

Aljabar Linear:

$$Rotasi\text{: } \frac{x'}{y'} = \begin{bmatrix} Cos\ \theta & -Sin\ \theta \\ Sin\ \theta & Cos\ \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.16}$$

$$Penskalaan\text{: } \frac{x'}{y'} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.17}$$

$$Translasi\text{: } \frac{x'}{y'} = \begin{bmatrix} x & + & t_x \\ y & + & t_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.18}$$

Fourier Transform:

$$F\ (u, v) = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) e^{-2\mu i\ (\frac{ux}{m} + \frac{vy}{n})} \tag{1.19}$$

Konvolusi dalam CNNs:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau) d\tau \tag{1.20}$$

Model probabilistik dan statistik juga sangat penting dalam visi komputer dan robotika. Model Markov dan Filter Kalman digunakan untuk pelacakan objek dan estimasi posisi, membantu dalam memperkirakan posisi objek yang bergerak dengan memperhitungkan ketidakpastian. Bayesian Networks digunakan untuk pengenalan pola dan keputusan dalam lingkungan yang tidak pasti. Geometri proyektif membantu memahami perspektif dan transformasi pandangan dalam gambar, yang penting untuk rekonstruksi 3D dari gambar 2D, sementara trigonometri digunakan untuk perhitungan sudut dan jarak, terutama dalam pengaturan kamera dan sensor.

Model Markov Tersembunyi (*Hidden Markov Models*, HMMs):

$$Prediksi: (x_t|x_{t-1}) = \sum_{x_{t-1}} p \ \ (x_t|x_{t-1}) \, p \, (x_{t-1}|y_{1:t-1}) \qquad (1.21)$$

Transformasi Geometris (Aljabar Linear):

$$Rotasi: \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Cos\ \theta & -Sin\ \theta \\ Sin\ \theta & Cos\ \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (1.22)$$

$$Penskalaan: \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (1.23)$$

$$Translasi: \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} \qquad (1.24)$$

Optimasi dan pembelajaran mesin adalah bidang matematika yang krusial dalam visi komputer dan robotika. Algoritma optimasi seperti *gradient descent* digunakan dalam pelatihan model pembelajaran mesin, termasuk jaringan saraf dalam. Teknik klasifikasi membantu dalam pengenalan objek dan klasifikasi gambar, dan algoritma seperti *Support Vector Machines* (SVM) digunakan untuk klasifikasi dan regresi dalam visi komputer. Dalam robotika, kinematika dan dinamika memainkan peran penting dalam perencanaan gerakan dan pengendalian robot. Kinematika langsung dan terbalik digunakan untuk menentukan posisi dan orientasi *end-effector* berdasarkan parameter *joint*, sementara dinamika menggunakan hukum Newton dan Lagrange untuk memahami gerakan dan gaya yang bekerja pada robot.

Support Vector Machines (SVM):

$$Fungsi\ Keputusan: f(x) = sgn\left(\sum_{i=1}^{n} \alpha_i y_i\, K(x_i, x) + b\right) \tag{1.25}$$

Teori kontrol, termasuk pengendali PID (*Proportional-Integral-Derivative*), digunakan untuk mengontrol gerakan robot dan menjaga kestabilan, sedangkan pengendali adaptif dan optimal digunakan untuk mengoptimalkan kinerja robot dalam berbagai kondisi lingkungan. Algoritma graf dan teori jaringan digunakan untuk pemetaan dan navigasi dalam robotika. Algoritma seperti Dijkstra dan A* membantu robot menemukan jalur terpendek dalam lingkungan, dan SLAM (*Simultaneous Localization and Mapping*) menggunakan kombinasi teori probabilitas dan graf untuk membuat peta lingkungan yang tidak diketahui sambil melacak posisi robot. Dengan integrasi berbagai konsep matematika ini, visi komputer dan robotika dapat mengembangkan sistem yang lebih canggih dan efisien untuk pengenalan, analisis, dan interaksi dengan dunia nyata, memungkinkan sistem ini bekerja dengan presisi tinggi dan ketahanan terhadap variasi lingkungan.

SLAM (*Simultaneous Localization and Mapping*):

$$x_t = f(x_{t-1}, u_t) + w_t \tag{1.26}$$

Matematika memainkan peran yang sangat penting dalam kecerdasan buatan, mulai dari pemodelan data dan pengembangan algoritma hingga pengambilan keputusan dan penalaran. Kemajuan dalam matematika terus mendorong batas-batas AI dan membuka peluang baru untuk aplikasi AI di berbagai bidang. Matematika adalah bahasa universal yang memungkinkan kita untuk memahami dan mendeskripsikan dunia di sekitar kita. Dalam konteks AI, matematika menyediakan alat yang kuat untuk membangun sistem yang cerdas, belajar dari data, dan membuat keputusan yang optimal. Seiring dengan perkembangan AI, peran matematika akan semakin penting dalam mendorong inovasi dan memajukan teknologi ini.

1.6 Masa Depan Matematika untuk Kecerdasan Buatan

Matematika memainkan peran yang sangat penting dalam kecerdasan buatan (AI), mulai dari pemodelan data dan pengembangan algoritma hingga pengambilan keputusan dan penalaran. Konsep-konsep matematis seperti aljabar linear, kalkulus, teori probabilitas, dan statistik memberikan fondasi kuat yang memungkinkan pengembangan algoritma pembelajaran mesin yang canggih . Algoritma seperti regresi linier, pohon keputusan, dan jaringan saraf tiruan semuanya bergantung pada prinsip-prinsip matematika yang kuat untuk melatih model, meminimalkan kesalahan, dan meningkatkan akurasi prediksi. Kalkulus dan teknik optimasi digunakan untuk menyempurnakan model AI, sementara teori informasi dan komputasi memungkinkan AI untuk memproses informasi secara efisien dan menangani tugas-tugas kompleks. Masa depan bidang matematika untuk AI sangat menjanjikan. Dengan kemajuan teknologi dan peningkatan kemampuan komputasi, matematika akan terus mendorong batas-batas AI dan membuka peluang baru untuk aplikasi di berbagai sektor. Penelitian dan pengembangan terus dilakukan untuk menemukan metode matematika baru yang lebih canggih dan efisien. Misalnya, teknik transformasi data, ekstraksi fitur, dan reduksi dimensi yang lebih baik akan meningkatkan performa model AI dan memperluas jangkauan aplikasinya. Pengembangan teknik probabilistik dan statistikal yang lebih canggih akan memungkinkan AI untuk menangani ketidakpastian dengan lebih baik, membuat prediksi yang lebih akurat, dan beroperasi di dunia nyata yang penuh dengan ketidakpastian. Matematika adalah bahasa universal yang memungkinkan kita untuk memahami dan mendeskripsikan dunia di sekitar kita. Dalam konteks AI, matematika menyediakan alat yang kuat untuk membangun sistem yang cerdas, belajar dari data, dan membuat keputusan yang optimal. Seiring dengan perkembangan AI, peran matematika akan semakin penting dalam mendorong inovasi dan memajukan teknologi ini. Dengan integrasi berbagai konsep matematika, AI dapat diterapkan dalam berbagai bidang, termasuk kesehatan, transportasi, keuangan, dan teknologi informasi, membawa inovasi dan efisiensi yang signifikan.

Bab 2

Matematika dalam Pengolahan Data

Matematika adalah fondasi kokoh yang menopang kelancaran dan ketepatan proses pengolahan data. Beragam konsep dan formula matematis menjadi alat penting dalam mengekstrak informasi berharga dari sekumpulan data yang kompleks. Fokus utama pada bab ini adalah bertujuan mengolah data menjadi matang sehingga siap dibangunkan model prediktif untuk memprediksi tren dan pola di masa yang akan datang.

2.1 Data dan *Encoding*

Data adalah bagian terkecil dari suatu informasi. Menurut Kamus Besar Bahasa Indonesia (KBBI), data adalah keterangan yang benar dan nyata. Dalam kata lain, data dapat direpresentasikan sebagai fakta / kenyataan. Data dapat diperoleh melalui berbagai macam cara, yaitu pengamatan, pengukuran maupun penelitian. Bentuk dari data bisa berupa teks seperti huruf dan angka, gambar, audio dan juga video. Pada bidang ilmu komputer / informatika, data dalam bentuk digital khususnya tabular seperti pada Tabel 2.1 berikut.

Tabel 2.1 adalah contoh dataset dari diabetes dengan fitur / variabel yaitu jenis kelamin, usia, BMI (*body mass index*) dan diabetes. Fitur jenis kelamin dan diabetes memiliki tipe data kategorikal, fitur usia dan BMI memiliki tipe data numerik. Perbedaan antara kategorikal dan numerik adalah jika data dengan tipe kategorikal memiliki nilai dalam bentuk kategori seperti contoh di atas, contoh lainnya adalah tinggi / rendah, besar / kecil, lebar / sempit, ya / tidak, dan lain sebagainya. Sedangkan tipe data numerikal contohnya adalah data dengan nilai dalam bentuk angka / numerik.

Tabel 2.1 Contoh Dataset

Jenis kelamin	**usia**	**BMI**	**Diabetes**
Perempuan	30	22	Tidak
Laki-laki	30	19	Tidak
Laki-laki	41	22	Tidak
Laki-laki	33	22	Tidak
Perempuan	61	35	Ya
Laki-laki	54	29	Ya
Laki-laki	60	33	Ya
Laki-laki	53	27	Ya

Komputer adalah sebuah mesin yang hanya dapat memproses data dalam bentuk angka, sehingga tipe data kategorikal biasanya akan diubah menjadi tipe numerik. Jika dilihat fitur jenis kelamin pada Tabel 2.1, bagaimana cara untuk mengubahnya menjadi numerik? Cara ini dinamakan dengan *label encoding*, diantaranya adalah sebagai berikut.

1. *One Hot Encoding*

   Teknik ini akan memproduksi data baru dalam bentuk biner, yaitu 0 dan 1 (Wu *et al.*, 2022). Hasil *encoding* pada fitur jenis kelamin disajikan pada Tabel 2.2 berikut.

Mengacu pada Tabel 2.2, dapat dilihat bahwa terjadi penambahan fitur sesuai dengan banyaknya kategori pada fitur jenis kelamin sebelumnya. Pada data yang bernilai laki-laki akan memiliki nilai 1 pada kolom laki-laki, sedangkan pada kolom perempuan akan bernilai 1 jika data adalah perempuan. Kelemahan dari teknik ini adalah meningkatnya data dengan nilai 0 dan bertambahnya fitur dari dataset. Dasar dari kecerdasan buatan adalah model matematis. Sehingga jika terdapat data dengan sebaran nilai 0 di dalamnya, akan sangat mempengaruhi model.

Tabel 2.2 Contoh Dataset Dengan *One Hot Encoding*

Laki-laki	**Perempuan**	**usia**	**BMI**	**Diabetes**
0	1	30	22	Tidak
1	0	30	19	Tidak
1	0	41	22	Tidak
`1	0	33	22	Tidak
0	1	61	35	Ya
1	0	54	29	Ya
1	0	60	33	Ya
1	0	53	27	Ya

2. *Ordinal Encoding*

Sedikit berbeda dengan *encoding* sebelumnya, teknik ini cenderung lebih mudah diterapkan dan tidak memproduksi fitur tambahan yang berlebih. Jika menilik pada Tabel 2.1, maka hasil *ordinal encoding*nya adalah sebagai berikut.

Berdasarkan Tabel 2.3, maka jenis kelamin perempuan akan bernilai 0 dan laki-laki bernilai 1. Kelemahan dari teknik ini adalah tidak bisa menganggap sebuah data memiliki nilai lebih tinggi / lebih rendah. Misalkan saja untuk perempuan = 0 dan laki-laki = 1. Jika pada matematika angka 1 memiliki nilai lebih besar dari 0. Sedangkan pada kenyataannya jenis kelamin itu setara. Apakah data hasil *encoding* akan bernilai biner? Jawabannya adalah tidak. Sebagai contoh jika terdapat data warna {tosca, pink, vermilion, terracota}. Hasil *encoding*nya adalah {tosca: 0; pink:1; vermilion:2; terracota:3}.

Tabel 2.3 Contoh dataset dengan *ordinal encoding*

Jenis kelamin	usia	BMI	Diabetes
0	30	22	Tidak
1	30	19	Tidak
1	41	22	Tidak
1	33	22	Tidak
0	61	35	Ya
1	54	29	Ya
1	60	33	Ya
1	53	27	Ya

2.2 Word Embedding

Data tidak selalu dalam bentuk angka, pada beberapa kasus data dalam bentuk teks kalimat / paragraf. Karena komputer dalam komputasinya hanya mengenal angka, sehingga data dalam bentuk teks perlu dilakukan ekstraksi untuk mendapatkan esensi data dalam bentuk angka. Data angka tersebut merupakan representasi dari data berupa teks tersebut. Sebagai contoh, misalkan terdapat dataset seperti pada Tabel 2.4 berikut:

Tabel 2.4 Contoh Dataset Berisi Kata

Dokumen 1	Saya makan nasi
Dokumen 2	Saya minum kopi
Dokumen 3	Saya suka makan nasi dan suka minum kopi

Beberapa cara untuk word embedding dapat dilakukan dengan 2 metode, yaitu *count vectorizer* dan TF-IDF. Berikut penjelasannya

1. *Count Vectorizer*

   Proses perhitungannya didasarkan pada kemunculan kata pada dataset. (Tripathy, Agrawal and Rath, 2015). Dengan menggunakan *count vectorizer* akan memproduksi *sparse matrix*. Dengan berdasarkan pada dataset Tabel 2.4, didapatkan *corpus* seperti pada Tabel 2.5 beserta nilai *embedding*nya. *Corpus* adalah sekumpulan kata yang membentuk kalimat.

Tabel 2.5 Dataset Hasil *Embedding* Dengan *Count Vectorizer*

	saya	makan	nasi	minum	kopi	dan	suka
Dokumen 1	1	1	1	0	0	0	0
Dokumen 2	1	0	0	1	1	0	0
Dokumen 3	1	1	1	1	1	1	2

Pada dokumen 1 terdapat kata "saya" yang kemunculannya hanya satu kali, sehingga pada Tabel 2.5 bernilai 1. Namun kata "minum" pada tidak muncul pada dokumen 1 sehingga bernilai 0. Hal ini berlaku hingga kata terakhir pada tiap dokumennya. Oleh karena itu kata "suka" pada dokumen 3 bernilai 2 karena kemunculannya terjadi sebanyak dua kali. Sehingga dapat dikatakan bahwa *count vectorizer* yang ditunjukkan oleh Tabel 2.5 adalah representasikan kemunculan kata dari seluruh *corpus* pada tiap dokumennya.

2. TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*). Metode ini banyak digunakan karena dalam mekanismenya dengan melakukan pembobotan dari suatu *term*. Kelebihan dari metode ini pada bidang kecerdasan buatan adalah dapat meningkatkan performa pada *recall* dan *precision* (Deolika, Kusrini and Luthfi, 2019). TF berarti rekuensi dari kemunculan kata pada suatu dokumen, sedangkan IDF adalah untuk melakukan pembobotan pada kata yang berbeda dari keseluruhan dokumen. Perhitungan dari TF-IDF mengacu pada persamaan (2.1) berikut:

$$tfidf(t_j, d_i) = tf(t_j, d_i) \times (\log \frac{N}{df(t_j)}) + 1 \qquad (2.1)$$

dengan $tf(t_j, d_i)$ adalah frekuensi kemunculan kata t_j dalam satu dokumen d_i, $df(t_j)$ adalah banyaknya dokumen terhadap kemunculan t_j dan N adalah jumlah banyaknya dokumen. Berdasarkan formula di atas, rumus IDF = $(\log \frac{N}{df(t_j)}) + 1$. Pada formula terdapat penambahan 1 yang bertujuan untuk menghindari produksi nilai 0. Apabila tanpa penambahan 1, maka hasil perkalian TF dengan IDF akan banyak menghasilkan nilai 0. Mengacu pada dataset Tabel 2.4 di atas, maka nilai N= 3 dan nilai $tf(t_j, d_i)$ didapatkan dari tabel *count vecctorizer* di atas. Tabel akhir perhitungan TF-IDF ditunjukkan pada Tabel 2.6 berikut:

Tabel 2.6 Dataset Hasil *Embedding* dengan TF-IDF

	saya	makan	nasi	minum	kopi	dan	suka
Dokumen 1	1	1,176091	1,176091	1	1	1	1
Dokumen 2	1	1	1	1,176091	1,176091	1	1
Dokumen 3	1	1,176091	1,176091	1,176091	1,176091	1,477121	1,954243

$$tfidf(saya, dokumen_1) = 1 \times (\log \frac{3}{3}) + 1 = 1$$
$$tfidf(makan, dokumen_1) = 1 \times (\log \frac{3}{2}) + 1 = 1{,}176091$$
$$tfidf(minum, dokumen_1) = 0 \times (\log \frac{3}{2}) + 1 = 1$$
$$tfidf(saya, dokumen_2) = 1 \times (\log \frac{3}{3}) + 1 = 1$$
$$tfidf(dan, dokumen_3) = 1 \times (\log \frac{3}{1}) + 1 = 1{,}477121$$
$$tfidf(suka, dokumen_3) = 2 \times (\log \frac{3}{1}) + 1 = 1{,}954243$$

Berdasarkan Tabel 2.6, dapat diamati bahwa semakin unik kata pada *corpus* dalam sekumpulan dokumen, akan menghasilkan nilai TF-IDF yang tinggi. Berbeda dengan kata yang sering muncul atau yang jarang tidak muncul, maka nilainya akan bernilai 1. TF-IDF sangat sering digunakan sebagai dalam pembobotan karena dapat menyeimbangkan bobot antara kata yang jarang muncul dan kata yang sering muncul (Wibowo, Hasbi and Anshori, 2024).

2.3 Transformasi Data

Data yang telah dikumpulkan untuk dimasukkan ke dalam proses kecerdasan buatan tidak selalu siap untuk diolah. Pada beberapa kasus, data harus ditransformasi terlebih dahulu ke dalam bentuk standard atau normal. Kedua metode itu adalah teknik yang umum digunakan.

Sebagai contoh pada Tabel 2.7 terdapat sebuah dataset tentang harga kamar kos yang terdiri dari jumlah fasilitas dan luas kamar dalam bentuk meter persegi sebagai berikut. Data pada Tabel 2.7 memiliki rentang dan skala yang berbeda.

Tabel 2.7 Dataset Untuk Transformasi Data

n fasilitas	Luas Kamar (m2)	Harga (ribu)
4	4	450
5	6,25	500
6	9	900
7	6,25	600

Bagaimana jika standardisasi dan nomalisasi diterapkan pada data Tabel 2.7 di atas? Berikut adalah penjelasannya.

1. **Standardisasi**

Standardisasi adalah teknik yang dapat diaplikasikan untuk mentransformasi data ke dalam bentuk yang lebih standar. Cara kerja dari metode ini adalah melakukan skala ulang pada data ke dalam bentuk distribusi normal sehingga nilai rerata 0 dan standar deviasi = 1 (Ghosh, 2022). Pada beberapa literatur lain, standardisasi juga disebut juga dengan normalisasi *Z*-score. Perhatikan formula yang disajikan berikut:

$$x' = \frac{x - \bar{x}}{\sigma} \qquad (2.2)$$

dengan x′ adalah data baru hasil standardisasi, x adalah data yang akan di standardisasi, $\bar{x}$ adalah rerata data pada satu kolom yang sama, dan σ adalah standar deviasi. Jika standardisasi diterapkan pada data Tabel 2.7, maka diperoleh data baru seperti pada Tabel 2.8.

Tabel 2.8 adalah data baru hasil dari produksi menggunakan standardisasi. Jika pada tiap baris data per fitur dilakukan perhitungan rerata, maka akan didapatkan nilai 0 dan standar deviasinya akan bernilai 1.

Tabel 2.8 Dataset Hasil Standardisasi Data

n fasilitas	Luas Kamar (m2)	Harga (ribu)
-1,34164	-1,34015667	-0,93094934
-0,44721	-0,070534562	-0,64450339
0,447214	1,481225793	1,64706421
1,341641	-0,070534562	-0,07161149

2. **Normalisasi**

Sebuah metode yang dapat digunakan untuk menyeragamkan rentang data yang sangat berbeda adalah normalisasi. Dengan normalisasi, data yang memiliki rentang beragam dapat di transformasi ke dalam skala dengan rentang yang sama (Anshori, Haris and Teja Kusuma, 2023). Kelebihan dari normalisasi ini adalah dapat meningkatkan performa dari model kecerdasan buatan dan berpengaruh terhadap waktu komputasinya. Salah satu metode normalisasi yang umum digunakan adalah *min-max*.

$$x' = (max - \min)\frac{x - minX}{maxX - minX} + min \qquad (2.3)$$

dengan x′ adalah data hasil normalisasi, x adalah data yang akan dinormalisasi, maxX adalah data dengan nilai terbesar dan minX adalah data dengan nilai terkecil. Sedangkan max adalah rentang terbesar dan min adalah rentang terkecil yang diinginkan. Jika ingin transformasi data dengan rentang 0 – 1, ikuti formulasi yang ditunjukkan pada persamaan (2.4).

$$x' = \frac{x - minX}{maxX - minX} \qquad (2.4)$$

Sebagai catatan, normalisasi bekerja berdasarkan kolom yang sama karena kolom pada sebuah dataset berupa satu buah variabel independen (berdiri sendiri). Contoh implementasi dari normalisasi *min-max*, kita gunakan data dari Tabel 2.7 di atas. Tabel 2.9 berikut adalah data hasil dari implementasi normalisasi *min-max*.

Tabel 2.9 Dataset Hasil Normalisasi Data

n fasilitas	Luas Kamar (m2)	Harga (ribu)
0	0	0
0,333333	0,45	0,11111111
0,666667	1	1
1	0,45	0,33333333

Tabel 2.9 di atas menunjukkan hasil normalisasi *min-max*. Data yang diproduksi menghasilkan rentang yang seragam, yaitu 0 hingga 1. Walaupun data terlihat jauh berubah, tetapi normalisasi ini masih menjaga relasi dengan data aslinya (Patro and Kumar, no date) sehingga informasi yang terkandung tidak hilang.

2.4 Seleksi Fitur

Beberapa kasus pada dataset adalah terdapat fitur yang tidak ada hubungannya dalam penggalian pola tersembunyi dari data. Oleh karena itu perlu dilakukan seleksi fitur untuk menghilangkan fitur yang tidak penting tersebut. Keberadaan fitur yang bias dapat mempengaruhi kinerja dari kecerdasan buatan saat diterapkan nantinya di masa mendatang. Selain itu dengan seleksi fitur maka fitur dataset akan berkurang dan dapat mempercepat waktu komputasi sehingga tidak membutuhkan *resource* yang lebih banyak. Berikut terdapat beberapa teknik sederhana yang dapat diterapkan dalam melakukan seleksi fitur.

1. *Correlation*

Koefisien korelasi atau *correlation coefficient* adalah derajat nilai untuk mengetahui seberapa erat hubungan antar variabel / fitur (Zhou, Wang and Zhu, 2022). Pada beberapa literatur, metode ini disebut juga dengan *Pearson Correlation Coefficient* dan formulanya ditunjukkan pada persamaan berikut:

$$PCC = \frac{Cov(X,Y)}{\sigma X \cdot \sigma Y} \tag{2.5}$$

dengan $Cov(X,Y)$ adalah *covariance* dari dua buah matrik fitur dan adalah σ standar deviasi. Kecenderungan nilai korelasi yang digunakan adalah yang mendekati nilai 1 atau -1. Hal ini dikarenakan hubungan kuat antar variabel adalah nilai PCC yang tinggi. Hubungan kuat positif adalah PCC mendekati nilai 1 dan sebaliknya jika hubungan kuat negatif adalah PCC dengan nilai mendekati -1. Berbeda jika nilai PCC mendekati nilai 0 yang menandakan bahwa fitur tersebut memiliki hubungan lemah antar variabel. Bahkan dapat dikatakan tidak memiliki hubungan jika bernilai sama dengan nol. Pada beberapa kasus, PCC yang bernilai rendah menunjukkan variabel tersebut akan dihapuskan dari dataset. Literatur lain menyebutkan bahwa *threshold* untuk PCC antara -0.5 hingga 0.5 yang dapat dihapupskan dari dataset. Sebagai contoh dengan mengacu Tabel 2.7 di atas, maka didapatkan Tabel 2.10 PCC seperti berikut:

Tabel 2.10 Korelasi Koefisien Data Per Fitur

	Fasilitas	*luas kamar*	*harga*
fasilitas	1		
Luas kamar	0,599336	1	
harga	0,544436	0,934451	1

Tabel 2.10 di atas menunjukkan nilai korelasi antar variabelnya dan menghasilkan nilai > 0.5. Maka dari itu fitur tidak ada yang di seleksi untuk dihapus dan masih dipertahankan.

2. *Information Gain*

Information gain (IG) adalah metode yang secara luas digunakan untuk seleksi dan menyaring fitur. Kelebihan metode ini adalah handal diterapkan untuk mengurangi *noise* pada dataset terhadap fitur yang tidak relevan (Syafitri Hidayatul AA, Yuita Arum S, 2018). Umumnya IG diaplikasikan pada kecerdasan buatan khususnya dalam algoritma *decision tree* dalam pembuatan cabang terhadap pohon keputusannya. Berikut formula dari IG.

$$InformationGain(S,A) = Entropy\ (S) - \sum_{v \epsilon Values(A)} \frac{|S_v|}{|S|} Entropy\ (S_v) \tag{2.6}$$

$$Entropy\ (S) = \sum_{i}^{c} -p_i \log_2 p_i \tag{2.7}$$

dengan S adalah banyaknya fitur, A adalah kelas target, S_v adalah banyaknya sampel pada tiap v, dan v adalah nilai yang menunjukkan probabilitas untuk kelas A. Dalam komputasinya, IG memerlukan *entropy* yang memiliki arti informasi mengenai proporsi pembagian kelas. Nilai dari *entropy* memiliki rentang antara 0 – 1. Nilai ini memiliki makna banyaknya sampel pada kelas dataset. Jika bernilai 0 maka sampel hanya berada di satu kelas. Jika bernilai 1 maka sampel berada di masing-masing kelas (Lishania, Goejantoro and Nasution, 2019).

Dapat dikatakan *entropy* adalah nilai untuk mengukur sebaran acak data.

Dalam proses seleksi fitur dengan IG dengan melakukan perankingan nilai IG dari tiap fiturnya. Nilai IG yang tinggi menandakan fitur tersebut mengandung informasi yang terikat dengan kelasnya. Berkebalikan jika nilai IG rendah mendekati 0 menandakan tidak memiliki informasi yang berhubungan dengan kelasnya. Fitur yang dipertahankan jika memiliki nilai tinggi, dan yang rendah mendekati nilai 0 akan dikecualikan dari dataset.

2.5 Reduksi Dimensi

Pada dunia nyata, data tidak selalu memiliki jumlah fitur yang sedikit. Semakin ke sini data memiliki ukuran yang sangat besar sehingga disebut dengan *big data*. Data yang berukuran besar bisa jadi memiliki baris data yang sangat banyak dan juga memiliki fitur yang teramat banyak. Semakin banyak fitur pada dataset, memiliki pengaruh terhadap dimensi data. Data dengan dimensi yang tinggi susah untuk divisualisasikan dan pada komputasinya membutuhkan sumber daya yang besar. Cara yang paling tepat untuk mengatasinya adalah dengan reduksi dimensi. Terdapat dua pendekatan yang terkenal handal untuk mengatasinya, yaitu PCA dan LDA.

1. *Principal Component Analysis* (PCA)

   Metode ini sering diaplikasikan pada data dengan dimensi yang tinggi untuk mereduksi dimensinya. Dimensi data yang tinggi adalah data dengan jumlah fitur atau variabel yang sangat banyak. Selain itu kelebihan dari PCA adalah dapat mempertahankan informasi yang dikandung oleh dataset walau telah mengalami reduksi dimensi, meningkatkan visualisasi data karena dimensi data menjadi rendah, dapat mendeteksi adanya informasi penting dan pada kasus klasifikasi pada kecerdasan buatan dapat meningkatkan performanya (Zhu, Idemudia and Feng, 2019). PCA dapat dikatakan sebagai metode reduksi dimensi yang bersifat *unsupervised* karena tidak memperhatikan kelas target dari dataset.

Langkah-langkah kalkulasi dari PCA cukup sederhana dan mudah. Dimulai dengan standardisasi seluruh fitur, selanjutnya menghitung *eigenvalues* dan *eigenvectors* dengan metode *eigen value decomposition* (EVD) atau *single value decomposition* (SVD). Secara general, formula dari PCA adalah sebagai berikut:

$$Ax = \lambda x \tag{2.8}$$

dengan λ adalah skalar. λ di sini disebut juga dengan *eigenvalue* dari matriks A dan x menjadi *eigenvector* yang berkorespondensi dengan λ. Jika x adalah vektor yang bernilai bukan nol, maka akan memenuhi formula (2.9) berikut:

$$(\lambda I - A)x = 0 \tag{2.9}$$

dengan I adalah matrik identitas. Sehingga *eigenspace* tercipta mengikuti formula (2.10):

$$E = \{x: (A - \lambda I)x\} = 0 \tag{2.10}$$

Saat *eigen space* telah didapatkan, tahap reduksi dimensinya adalah dengan mengurutkan *eigenvector* oleh *eigenvalue* dari tertinggi hingga terendah. Ukuran dimensi luarannya adalah $m \times k$ dengan k adalah banyaknya *principal component* yang digunakan dan m adalah banyaknya baris data. Pada beberapa literatur dikatakan bahwa pemilihan *principal component* yang optimal adalah mengandung 85% varian dari data asli (Zhongwen and Huanghuang, 2017).

2. *Linear Discriminant Analysis* (LDA)

Selain PCA, terdapat metode yang pembelajarannya bersifat *supervised* yaitu *linear discriminant analysis* (LDA). Metode ini sangat memperhatikan kelas target pada proses reduksi dimensinya dan cocok untuk data dengan kasus klasifikasi / regresi karena ada target yang di prediksi. Cara kerjanya cukup sederhana yaitu dengan memproyeksikan data ke dalam ruang vektor serta meminimalkan jarak data pada kelas yang sama dan memaksimalkan jarak antar kelas yang berbeda (Xie, 2015). Sehingga data akan saling mendekat dengan kelas yang sama.

Langkah kalkulasi dari LDA ada 3, yaitu menghitung jarak dalam kelas yang sama, menghitung jarak antar kelas yang berbeda dan menciptakan data dengan ruang dimensi yang lebih rendah(Ilias *et al.*, 2016). Berbeda dengan PCA, LDA akan otomatis melakukan transformasi dataset ke dalam dimensi yang lebih rendah tanpa perlu melakukan *tuning* persentasenya.

Bab 3

Pemodelan Matematika untuk Algoritma *Machine Learning*

3.1 *Machine Learning*

Machine Learning adalah cabang dari kecerdasan buatan (AI) yang berfokus pada pengembangan algoritma yang memungkinkan komputer untuk belajar dari dan membuat prediksi atau keputusan berdasarkan data. *Machine Learning* digunakan di berbagai bidang, termasuk pengenalan gambar, pemrosesan bahasa alami, prediksi pasar dan lainnya.

Pemodelan matematika adalah fondasi dari algoritma *Machine Learning*. Pemodelan ini membantu dalam memahami bagaimana algoritma bekerja, mengoptimalkan kinerjanya dan mengevaluasi hasilnya. Pemahaman matematika yang mendalam memungkinkan pengembangan algoritma yang lebih efisien dan akurat.

Algoritma *Machine Learning* dapat dikelompokkan menjadi beberapa kategori, antara lain : *Supervised Learning* yaitu algoritma yang belajar dari data yang diberi label, *Unsupervised Learning* yaitu algoritma yang belajar dari data yang tidak diberi label, dan *Reinforcement Learning* yaitu algoritma yang belajar melalui interaksi dengan lingkungan.

3.2 Regresi

Regresi merupakan Algoritma dalam *Machine Learning* yang digunakan untuk memodelkan hubungan antara *variable dependent* (biasanya berupa nilai kontinu) dengan satu atau lebih *variable independent* (fitur atau *predictor*). Tujuannya untuk memprediksi atau memperkirakan nilai dari *variable dependent* berdasarkan nilai *variable independent*.

Konsep dasar regresi mencoba menemukan hubungan matematis antara *variable dependent* (Y) dan satu atau lebih *variable independent* (X). Model Regresi biasanya berbentuk persamaan: $Y = f(X) + \epsilon$, dengan $f(X)$ adalah fungsi dari *variable independent*, dan ϵ adalah error atau noise.

Regresi memiliki beberapa jenis berdasarkan hubungan antara X dan Y, antara lain : Regresi Linier, Regresi Logistik dan Regresi Polinomial.

3.2.1 Regresi Linier

Regresi Linier adalah regresi dengan hubungan linier antara X dan Y.

1. Model Regresi Linier Sederhana:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{3.1}$$

   dengan Y adalah *variable dependent*, X adalah *variable independent*, β_0 adalah intercept (titik potong dengan sumbu Y), β_1 adalah slope (kemiringan garis regresi), dan ϵ adalah *error* atau residu.

2. Model Regresi Linier Berganda:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{3.2}$$

   Regresi Linier Berganda bersifat Multikolinearitas ketika kondisi terjadi *variable independent* saling berkorelasi tinggi. Asumsi dalam Regresi Linier yaitu untuk validitas model Regresi Linier, ada beberapa asumsi yang harus dipenuhi yaitu hubungan antara *variable independent* dan *dependent* harus linier (Linearitas), variasi *error* harus konstan diseluruh rentang *variable independent* (Homoskedastisitas), *error* harus terdistribusi normal (*Normalitas Error*),

3.2.2 Regresi Polinomial

Regresi Polinomial adalah bentuk regresi linier ketika hubungan antara variable independent x dan variable dependen y dimodelkan sebagai polynomial derajat n. Persamaan umum regresi polynomial derajat n dituliskan sebagai berikut:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon \tag{3.3}$$

dengan y adalah *variable dependent*, x adalah *variable independent*, β_0, $\beta_1, \ldots, \beta_n$adalah koefisien yang diperkirakan, dan ϵ adalah kesalahan (*error*) atau *noise*.

Untuk memperkirakan koefisien metode umum yang digunakan adalah Metode Kuadrat Terkecil (*Least Square Method*). Tujuanmya untuk meminimalkan jumlah kuadrat dari residual (selisih antara nilai yang diamati dan nilai yang diprediksi). Untuk Matriks desain X sistem persamaan regresi polinomial adalah sebagai berikut :

$$X = \begin{bmatrix} 1 & \mathrm{x}_1 & x_1^2 & \ldots & x_1^n \\ 1 & \mathrm{x}_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathrm{x}_\mathrm{m} & x_m^2 & \ldots & x_m^n \end{bmatrix} \tag{3.4}$$

dengan $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{m}$ adalah nilai-nilai dari *variable independent*.

Koefisien β dapat diperoleh dengan menyelesaikan persamaan normal:

$$\beta = (X^T X)^{-1} X^T y \tag{3.5}$$

dengan X^T adalah transpose dari matriks desain X, dan y adalah vektor dari nilai-nilai *variabel dependent*.

3.2.3 Regresi Logistik

Regresi Logistik adalah teknik analisis statistik yang digunakan untuk memodelkan data ketika *variable dependent* (tergantung) adalah kategori atau biner (misalnya, 0 atau 1, ya atau tidak, benar atau salah). Jenis regresi logistik ini digunakan Ketika hasil dari model berupa probabilitas suatu kejadian tertentu.

Persamaan dasar regresi logistik adalah fungsi logistik atau sigmoid seperti berikut:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.6}$$

dengan z adalah kombinasi linier dari variabel *independent*. Model regresi logistik memprediksi probabilitas kejadian kelas positif (misalnya, y=1). Model ini dapat dinyatakan sebagai berikut:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{3.7}$$

dengan $\beta_0$adalah intercept(bias), β_i adalah koefisien untuk variabel prediktor x_i, sedangkan x_i adalah variabel *predictor*. Parameter β dalam regresi logistik diestimasi menggunakan metode *Maksimum Likelihood Estimation* (MLE) untuk mencari parameter yang memaksimalkan kemungkinan (*likelihood*) pengematan data. Persamaan *Log-likelihood*(LL) untuk regresi logistik sebagai berikut:

$$LL(\beta) = \sum_{i=1}^{N} [y_i \, log(\sigma(z_i)) + (1 - y_i) log(1 - \sigma(z_i))] \tag{3.8}$$

dengan y_i adalah label kelas actual (0 atau 1) untuk pengamatan ke-i. Sedangkan $\sigma(z_i)$ adalah probabilitas yang diprediksi oleh model untuk pengamatan ke-i.

3.3 *K-Nearest Neighbors* (K-NN)

K-Nearest Neighbors (K-NN) merupakan algoritma dalam *Machine Learning* non-parametrik untuk klasifikasi dan regresi. K-NN bekerja dengan mengidentifikasi nilai K tetangga terdekatdari sebuah titik data baru dan melakukan prediksi berdasarkan nilai rata-rata dari tetangga-tetangga tersebut. Langkah-langkah Algoritma K-NN sebagai berikut:

1. Pemilihan Parameter K yaitu menentukan jumlah tetangga terdekat K yang akan digunakan dalam klasifikasi atau prediksi.
2. Menghitung Jarak yaitu menghitung jarak antara sampel baru dengan titik data dalam dataset pelatihan. Rumus perhitungan jarak yang umum digunakan adalah :
   a. Jarak Euclidean adalah konsep dasar dalam geometri yang digunakan untuk mengukut jarak antara dua titik dalam ruang Euclidean. Jarak Euclidean sering digunakan dalam berbagai bidang termasuk bidang matematika, ilmu komputer dan statistik.

      Rumus Jarak Euclidean :

      Jarak Euclidean antara dua titik P = $(p_1, p_2, \ldots, p_n)$ dan Q = $(q_1, q, \ldots, q_n)$ dalam ruang Euclidean n-dimensi yaitu panjang dari segmen garis yang menghubungkan kedua titik tersebut. Secara matematika, jarak Euclidean didefinisikan sebagai berikut:

$$d(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \tag{3.9}$$

b. Jarak Manhattan adalah ukuran jarak antar dua titik dalam ruang n-dimensi yang diukur sepanjang sumbu-sumbu koordinat pada grid persegi. Nama "Manhattan" berasal dari cara menghitung jarak di grid jalanan kota Manhattan, dimana jalan-jalan berbentuk persegi Panjang.

   Rumus Jarak Manhattan :

   Jarak Manhattan antara dua titik P = $(p_1, p_2, \ldots, p_n)$ dan Q = $(q_1,\ q,\ \ldots,\ q_n)$ dalam ruang Euclidean n-dimensi adalah jumlah nilai mutlak dari perbedaan koordinat-koordinat mereka. Secara matematis, jarak Manhattan didefinisikan sebagai:

$$d_{Manhattan}(P,Q) = \sum_{i=1}^{n} |p_i - q_i| \qquad (3.10)$$

3. Mengidentifikasi K tetangga terdekat yaitu dengan mengurutkan semua jarak yang telah dihitung dari yang terkecil hingga terbesar. Kemudian memilih K titik data dengan jarak terkecil sebagai tetangga terdekat.
4. Untuk kasus klasifikasi dengan cara menghitung kategori dengan menentukan kelas dengan menghitung jumlah kemunculan setiap kelas di antara K tetangga terdekat. Kemudian Kelas dengan jumlah terbanyak adalah kelas yang diprediksi.
5. Untuk kasus regresi yaitu dengan mengambil rata-rata nilai K tetangga terdekat.

3.3.1 Contoh Perhitungan

Diketahui dataset berikut dengan 2 fitur dan 2 kelas, terlihat pada Tabel 3.1. Dari data akan memprediksi kelas baru dengan data test Fitur1 = 3, Fitur2 = 3, dengan nilai K=3.

Tabel 3.1 Dataset untuk KNN

No.	Fitur1	Fitur2	Kelas
1	1	2	A
2	2	3	A
3	3	1	B
4	6	5	B

Langkah-langkah perhitungan sebagai berikut :

1. Hitung jarak Euclidean, jarak antar dua titik (x_1, y_1) dan (x_2, y_2) yaitu :

$$Jarak = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

   - Jarak ke Data No.1 :

     $\sqrt{(3-1)^2 + (3-2)^2} = \sqrt{4+1} = \sqrt{5} \approx 2{,}24$

   - Jarak ke Data No.2 :

     $\sqrt{(3-2)^2 + (3-3)^2} = \sqrt{1+0} = 1$

   - Jarak ke Data No.3 :

     $\sqrt{(3-3)^2 + (3-1)^2} = \sqrt{0+4} = 2$

   - Jarak ke Data No.4 :

     $\sqrt{(3-6)^2 + (3-5)^2} = \sqrt{9+4} = \sqrt{13} \approx 3{,}61$

2. Tentukan 3 tetangga terdekat yaitu dengan mengurutkan berdasarkan jarak pendek : Data No. 2, Data No. 3, dan Data No. 1.

3. Prediksi kelas yaitu dengan melihat kelas-kelas Data tetangga terdekat :

- Kelas tetangga terdekat : Data No.2 = A, Data No.3 = B, dan Data No.1 = A
- Kelas yang paling banyak adalah kelas A

  Oleh karena itu, data baru dengan Fitur1 = 3, Fitur2 =3, diprediksi sebagai Kelas A.

Jadi dapat ditarik kesimpulan, K-NN adalah algoritma yang mudah diimplementasikan dan dipahami, tetapi memerlukan optimalisasi pada nilai K dan pengukuran jarak yang tepat untuk kinerja terbaik.

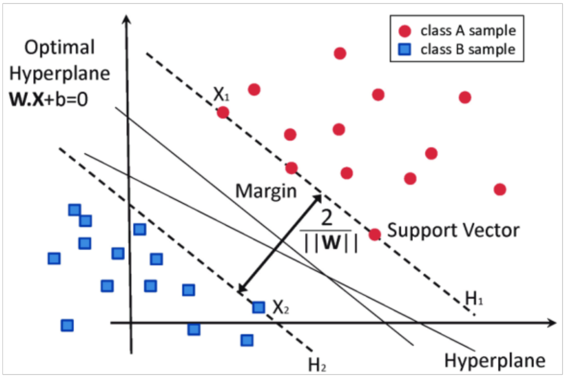
3.4 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan algoritma machine learning yang digunakan untuk klasifikasi dan regresi. SVM dikenal sebagai algoritma yang mampu menangani data berdimensi tinggi dan bekerja dengan baik pada kasus dengan jumlah sample lebih kecil dibandingkan dengan dimensi fitur. Berikut adalah teori yang mendasari SVM dan langkah-langkah utama dalam algoritma SVM.

3.4.1 Konsep Dasar

1. **Hyperplane Pemisah**: SVM mencari hyperplane yang memisahkan data ke dalam dua kelas berbeda. Hyperplane ini adalah batas keputusan yang memisahkan satu kelas dari kelas lainnya dalam ruang fitur.
2. **Margin Maksimum**: SVM berusaha menemukan hyperplane yang memaksimalkan margin, yaitu jarak antara hyperplane dan titik data terdekat dari setiap kelas. Titik data yang paling dekat ke hyperplane disebut sebagai *support vectors*.
3. **Linear dan Non-linear SVM**:
   a. **Linear SVM** digunakan ketika data dapat dipisahkan dengan hyperplane linear.

b. **Non-linear SVM** digunakan ketika data tidak dapat dipisahkan secara linear. Dalam kasus ini, SVM menggunakan teknik yang disebut *kernel trick* untuk memetakan data ke dimensi yang lebih tinggi di mana data tersebut dapat dipisahkan secara linear.



Gambar 3.1 Konsep Hyperplane SVM

3.4.2 Langkah-langkah Algoritma SVM :

1. Mendefiniskan Fungsi Tujuan

Untuk menemukan hyperplane, SVM memecahkan masalah optimisasi yang bertujuan untuk memaksimalkan margin:

$$Minimize\ \frac{1}{2}\|w\|^2 \tag{3.11}$$

dengan kendala : $y_i(w \cdot x_i + b) \geq 1$ dengan w adalah vektor bobot, x_i adalah vector fitur, y_i adalah label kelas, dan b adalah bias.

2. Menggunakan Kernel Trick (Jika diperlukan)

Ketika data tidak dapat dipisahkan secara linier, fungsi kernel $K(x_i, x_j)$ digunakan untuk memetakan data ke ruang fitur berdimensi lebih tinggi. Beberapa kernel umum meliputi kernel linier, polinomial dan Radial Basis Function (RBF). Beberapa rumus kernel umum tersebut adalah sebagai berikut :

a. Linier:

$$K(x_i, x_j) = x_i \,.\, x_j \tag{3.12}$$

b. Polinomial:

$$K(x_i, x_j) = (x_i \,.\, x_j + c)^d \tag{3.13}$$

c. Gaussian (RBF):

$$K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2 \tag{3.14}$$

3. Menentukan Support Vector

Solusi dari masalah optimisasi di atas memberikan vektor bobot w dan bias b. Support vektor adalah data yang memenuhi rumus berikut:

$$y_i(w \,.\, x_i + b) = 1 \tag{3.15}$$

4. Klasifikasi Data Baru

Setelah model SVM dilatih, data baru dapat diklasifikasikan dengan menggunakan fungsi keputusan:

$$f(x) = sign(w \,.\, x_i + b) \tag{3.16}$$

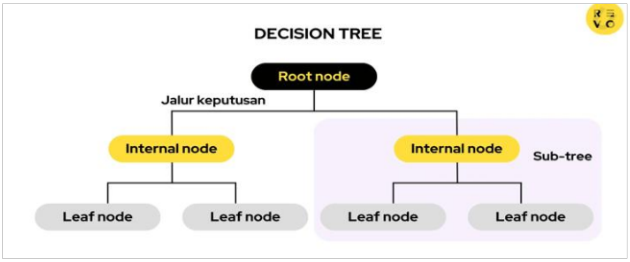
Data baru x di klasifikasikan berdasarkan $f(x)$ apakah hasil dari positif atau negatif. Jika sign $f(x)$ positif, maka sampe data x tersebut diklasifikasikan ke dalam satu kelas (misalnya, +1) dan jika sign $f(x)$ negatif, maka sample tersebut diklasifikasikan ke kelas yang lain (misalnya, -1). Algoritma SVM adalah algoritma yang kuat dalam analisis data, khususnya Ketika menghadapi data yang sulit dipisahkan secara linier, dan memberikan hasil yang dapat diandalkan jika parameter model dipilih dengan baik.

3.5 Decision Tree

Decision Tree adalah salah satu algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. Algoritma ini bekerja dengan membangun model pohon keputusan berdasarkan fitur dari data untuk memprediksi nilai target. Berikut adalah teori dasar dan langkah-langkah utama dari algoritma *decision tree*, beserta contoh perhitungannya.

3.5.1 Konsep Dasar

Struktur *decision tree* dimulai dari node, cabang dan keputusan.



Gambar 3.2 Struktur Decision Tree

1. Node

   Node terdiri dari 3 level yaitu Root Node, Internal Node, dan Leaf Node. Root Node adalah node paling atas dari pohon yang mewakili seliuruh dataset.

2. Cabang

   Cabang adalah penghubung node dan menunjukan hasil dari pengujian pada atribut.

3. Keputusan

   Keputusan diambil dengan mengikuti jalur dari rood node ke leaf node.

3.5.2 Langkah-langkah Algoritma *Decision Tree*

1. Pilih Atribut Terbaik

   Untuk memilih atribut terbaik sebagai node, maka perlu menggunakan kriteria tertentu seperti Gini Impurity, Entropy, dan Information Gain.

2. Pembagian Dataset

   Dataset dibangun menjadi subset berdasarkan nilai atribut yang dipilih.

3. Pembangunan Pohon

   Proses ini dilakukan secara berulang dengan cara rekrusif untuk setiap subset sampai salah satu dari kondisi berikut ini terpenuhi :

   a. Semua sample dalam subset memiliki label yang sama

   b. Tidak ada atribut yang tersisa untuk dibagi

   c. Tidak ada data dalam subset.

4. Pohon Keputusan

   Pohon keputusan dibentuk dengan node dan cabang dimana setiap node internal menunjukkan uji pada atribut, dan setiap leaf node menunjukkan hasil klasifikasi.

3.5.3 Contoh Perhitungan Decision Tree

Studi kasus sederhana, terdapat dataset Tabel 3.2 yang akan digunakan untuk mengklasifikasi apakah seseorang akan bermain tenis berdasarkan cuaca dengan menggunakan Algoritma Decision Tree. Dataset :

Tabel 3.2 Dataset untuk Decision Tree

Cuaca	Suhu	Kelembaban	Berangin	Bermain Tenis
Cerah	Tinggi	Tinggi	Tidak	Tidak
Cerah	Tinggi	Tinggi	Ya	Tidak
Mendung	Tinggi	Tinggi	Tidak	Ya
Hujan	Sedang	Tinggi	Tidak	Ya
Hujan	Rendah	Normal	Tidak	Ya
Hujan	Rendah	Normal	Ya	Tidak
Mendung	Rendah	Normal	Ya	Ya
Cerah	Sedang	Tinggi	Tidak	Tidak
Cerah	Rendah	Normal	Tidak	Ya
Hujan	Sedang	Normal	Tidak	Ya
Cerah	Sedang	Normal	Ya	Ya
Mendung	Sedang	Tinggi	Ya	Ya
Mendung	Tinggi	Normal	Tidak	Ya
Hujan	Sedang	Tinggi	Ya	Tidak

Langkah-langkahnya :

1. Menghitung Nilai Entropy Awal

   Menghitung entropy untuk seluruh dataset dengan rumus:

$$\text{Entropy}\ (S) = \sum_{i=1}^{n} - p_i * \log_2 p_i \tag{3.17}$$

Berdasarkan Tabel 3.2, terdapat 14 record data, 9 data bermain tenis (Kelas Positif) dan 5 tidak bermain tenis (Kelas Negatif) :

$$\text{Entropy (S)} = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) + -\frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0{,}940$$

2. Menghitung Nilai *Information Gain* untuk Atribut "Cuaca":

Membagi dataset berdasarkan nilai Cuaca (Cerah, Mendung, Hujan), kemudian menghitung entropy setiap subset.

- Mendung

$$Entropy\ (S_{Mendung}) = -\frac{4}{4}log_2\left(\frac{4}{4}\right)$$
$$= 0\ (Semua\ Positif)$$

- Cerah

$$Entropy(S_{Cerah}) = -\frac{2}{5}log_2\left(\frac{2}{5}\right) - \frac{3}{5}log_2\left(\frac{3}{5}\right) = 0{,}971$$

- Hujan

$$Entropy\left(S_{Hujan}\right) = -\frac{3}{5}log_2\left(\frac{3}{5}\right) - \frac{2}{5}log_2\left(\frac{2}{5}\right) = 0{,}971$$

- Menghitung Nilai Information Gain :

$$Gain(S, Cuaca)$$
$$= Entropy(S)$$
$$-\left(\frac{5}{14}\ x\ 0{,}971 + \frac{4}{14}\ x\ 0 + \frac{5}{14}\right)$$

$$Gain\ (S, Cuaca) = 0.940 - 0{,}693 = 0{,}247$$

3. Pilih atribut dengan Gain Tertinggi

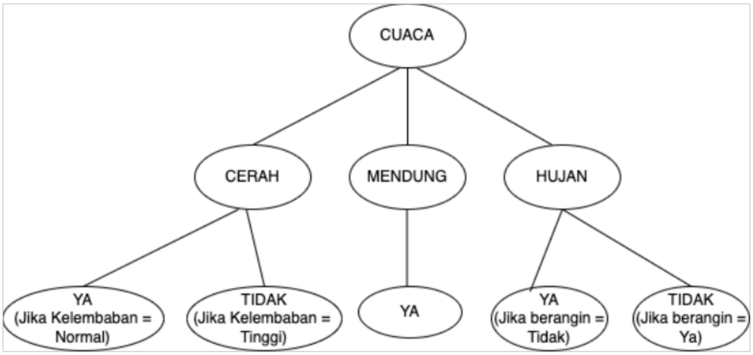
Bandingkan nilai Information Gain unutk atribut lainnya yaitu Suhu, Kelembaban, Berangin), dan pilih atribut denan nilai Gain tertinggi untuk node berikutnya.

4. Rekursif membagi dataset

Ulangi proses diatas untuk setiap cabang menggunakan subset yang dihasilkan dari pembagian sebelumnya

3.5.4 Pohon Keputusan (*Decision Tree*)

Berikut merupakan Pohon Keputusan yang terbentuk dari dataset Tabel 3.2.

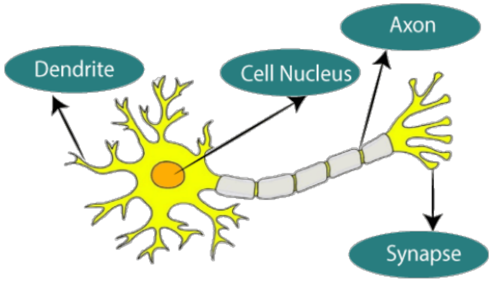


Gambar 3.3 *Decision Tree* berdasarkan Tabel 3.2

Pada Gambar 3.3 merupakan pohon keputusan yang diambil dengan mengikuti jalur dari root ke leaf berdasarkan nilai atribut-atribut dari dataset Tabel 3.2.

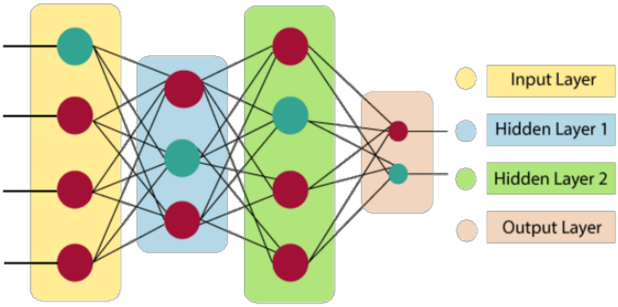
3.6 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) atau Artifivial Neural Network (ANN) adalah model komputasi yang terinspirasi oleh cara kerja otak manusia dalam memproses informasi. Gambar 3.4 merupakan ilustrasi dari jaringan syaraf biologis manusia.



Gambar 3.4 Ilustrasi Jaringan Syaraf Biologis Manusia
(Sumber : javapoint.com)

JST terdiri dari sejumlah besar unit pemrosesan sederhana yang disebut neuron, yang dihubungkan oleh bobot yang dapat disesuaikan. JST dapat belajar dari data dengan menyesuaikan bobot-bobot ini. Arsitektur Jaringan Syaraf Tiruan menggunakan beragam lapisan pemrosesan matematis untuk menginterpretasikan informasi yang diberikan. Umumnya, JST terdiri dari puluhan hingga jutaan neuron buatan yang disebut unit, yang disusun dalam beberapa lapisan atau layer. Untuk gambar arsitektur JST terlihat pada Gambar 3.5.



Gambar 3.5 Arsitektur Jaringan Syaraf Tiruan

(Sumber : javapoint.com)

JST memiliki komponen utama yaitu Neuron, Lapisan (*Layers*), Bobot (*Weights*), dan Fungsi Aktivasi. Neuron adalah Unit dasar dari JST yang memproses informasi, Lapisan (*Layers*) terdiri dari Lapisan Masukan (*Input Layer*) : tempat ata mentah dimasukkan ke dalam jaringan; Lapisan Tersembunyi (*Hidden Layer*) : lapisan yang memproses data setelah input, bisa lebih dari satu; dan Lapisan Keluaran (*Output Layer*) : hasil akhir dari pemrosesan jaringan, Bobot (*Weights*) yaitu nilai yang menghubungkan neuron satu dengan neuron yang lain ketika bobot ini akan diperbarui selama proses pembelajaran, dan Fungsi Aktivasi yaitu fungsi matematis yang menentukan keluaran dari suatu neuron berdasarkan masukan yang diterima, contohnya : sigmoid, relu, tanh.

Adapun proses pembelajaran JST sebagai berikut :

1. **Inisialisasi Bobot:** Bobot-bobot diinisialisasi dengan nilai acak kecil.
2. **Feedforward:** Data masukan diproses dari lapisan masukan menuju lapisan keluaran melalui lapisan tersembunyi.
3. **Perhitungan Kesalahan (Loss):** Menghitung seberapa jauh hasil jaringan dari nilai yang diharapkan menggunakan fungsi loss.
4. **Backpropagation:** Menyesuaikan bobot berdasarkan kesalahan yang dihitung untuk meminimalkan loss.
5. **Pembaharuan Bobot:** Bobot diperbarui menggunakan algoritma optimasi seperti Stochastic Gradient Descent (SGD).

3.6.1 Contoh Perhitungan JST

Misalkan kita memiliki JST sederhana dengan satu neuron, satu masukan (x), dan satu keluaran (y). Fungsi aktivasi yang digunakan adalah fungsi linear.

1. Inisialisasi

   Bobot awal (w) diinisialisasi dengan nilai acak, misalkan $w = 0{,}5$. Kemudian nilai Bias (b) juga diinisialisasi dengan nilai, misalkan $b = 0{,}1$.

2. Feedforward

Fungsi aktivasi linier $y = wx + b$. Misalkan, masukan $(x = 2)$, maka keluaran (y) adalah

$$y = (0{,}5\ x\ 2)\ +\ 0{,}1\ =\ 1{,}0\ +\ 0{,}1\ =\ 1{,}1$$

3. Perhitungan Kesalahan

Misalkan target keluaran $(y_{terget} = 1{,}5)$, maka nilai kesalahan (E) adalah :

$$E = \frac{1}{2}(y_{target} - y)^2 = \frac{1}{2}(1{,}5\ -\ 1{,}1)^2\ =\ 0{,}08$$

4. Backpropagation dan Pembaruan Bobot

Gradien dari kesalahan terhadap bobot $(\frac{\partial E}{\partial w})$ dihitung dan digunakan untuk memperbarui bobot :

$$\frac{\partial E}{\partial w} =\ (y\ - y_{target})\,.\,x\ =\ (1{,}1 - 1{,}5)\,.\,2\ = 10{,}8$$

Menggunakan laju pembelajaran $(\alpha = 0{,}01)$, bobot baru (w') diperoleh :

$$\mathrm{w}' = w\ - \alpha\,.\frac{\partial E}{\partial w}\ =\ 0{,}5\ -\ 0{,}01\,.\,(-0{,}8)\ =\ 0{,}508$$

Iterasi ini akan terus berulang sampai kesalahan (E) mencapai nilai yang diinginkan atau konvergensi tercapai. Ini adalah gambaran dasar bagaiaman JST bekerja dalam proses pembelajaran.

3.7 Evaluasi Model

Evaluasi model algoritma machine learning adalah langkah penting untuk memastikan bahwa model yang telah dibangun mampu melakukan prediksi dengan baik pada data yang belum pernah dilihat sebelumnya. Berikut adalah beberapa konsep teori dasar dan contoh perhitungan sederhana dalam evaluasi model machine learning:

3.7.1 Teori Evaluasi Model

1. Pembagian data

   Pada proses machine learning menggunakan dataset, dimana dataser di bagi menjadi : Data Training, Data Validation dan Data Test.

   - **Data Training** adalah data yang digunakan untuk melatih model.
   - **Data Validation** adalah data yang digunakan untuk mengoptimalkan model dan memilih parameter terbaik.
   - **Data Test** adalah data yang digunakan untuk mengevaluasi kinerja akhir model.

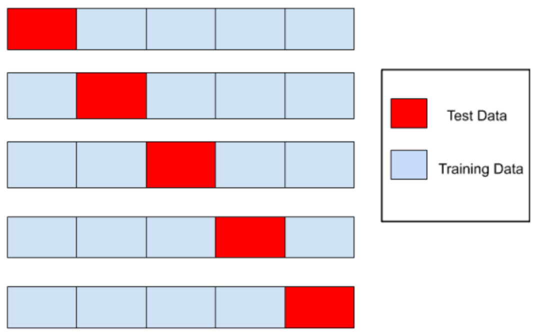
2. Metode Evaluasi

   - **Akurasi:** Proporsi prediksi yang benar dari keseluruhan prediksi. Cocok untuk dataset yang seimbang.
   - **Precision, Recall, dan F1-Score:** Digunakan untuk mengevaluasi kinerja model pada dataset yang tidak seimbang.
   - **Precision:** Jumlah prediksi positif benar dibagi dengan jumlah total prediksi positif.
   - **Recall (Sensitivity):** Jumlah prediksi positif benar dibagi dengan jumlah total data positif yang sebenarnya.
   - **F1-Score:** *Harmonic mean* dari *precision* dan *recall*.
   - ***Confusion Matrix*:** Tabel yang digunakan untuk mendeskripsikan kinerja model classification, terdiri dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN).

- **ROC-AUC Curve:** Grafik yang menunjukkan kemampuan model dalam membedakan antara kelas positif dan negatif.

3. Cross Validation

Metode pembagian data yang lebih robust dengan membagi dataset menjadi beberapa subset (folds) dan melatih serta menguji model pada setiap kombinasi subset. Ilustrasi pembagian data terlihat pada Gambar 3.5 di bawah ini.



Gambar 3.6 Ilustasi Pembagian Dataset Cross Validation

3.7.2 Contoh Perhitungan

Diketahui data hasil prediksi yaitu TP=50, FP=10, TN=30, FN=10. Hitung nilai akurasi, precision, recall dan F1-score.

1. Hitung nilai akurasi

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} = \frac{50 + 30}{50 + 10 + 30 + 10} = \frac{80}{100} = 80$$

Akurasi dari model di~~a~~ atas adalah 80%.

2. Hitung nilai precision

$$Precision \ = \ \frac{TP}{TP + FP} = \frac{50}{50 + 10} = \ \frac{50}{60} \ \approx \ 0{,}833$$

Precisiaon dari kedua model adalahj 83%

3. Hitung nilai Recall

$$Precision \ = \ \frac{TP}{TP + FN} = \frac{50}{50 + 10} = \ \frac{50}{60} \ \approx \ 0{,}833$$

Recall dari model adalah 83,33%

4. Hitung nilai F1-Score

$$F1 - Score \ = 2x \ \frac{Precision \ X \ Recall}{Precision \ + \ Recall} = 2x \frac{0{,}833 \ x \ 0{,}833}{0.833 + \ 0{,}833}$$
$$= \ \frac{50}{60} \ \approx \ 0{,}833$$

F1-Score dari model adalah 83,33%

Dengan menggunakan metrik-metrik di atas, kita dapat mengevaluasi model machine learning dan memahami di mana kekuatan dan kelemahan model tersebut. Dalam kasus dataset yang tidak seimbang, metrik seperti precision, recall, dan F1-score sering kali lebih informatif daripada hanya mengandalkan akurasi.

Pemodelan matematika memberikan dasar yang kuat untuk memahami dan mengimplementasikan algoritma *Machine Learning*. Setiap algoritma memiliki pendekatan unik untuk permodelan dan optimisasi, yang memungkinkan aplikasi yang luas dalam berbagai bidang. Dengan memahami matermatika dibalik ~~A~~algoritma-algoritma *Machine Learning* dapat dikembangkan model yang lebih akurat dan efisien untuk berbagai tugas *Machine Learning*.

Bab 4

Matematika dalam Pengembangan Model Prediktif

4.1 Model Prediktif

Setiap hari, kita sering kali dihadapkan pada pertanyaan-pertanyaan seperti:

- ″Apakah besok pagi cuaca akan hujan?″
- ″Apakah saya harus membeli rumah sekarang atau perlu menunggu?″
- ″Apakah produk baru ini akan sukses di pasaran?″

Pertanyaan-pertanyaan ini mencerminkan keinginan kita untuk memprediksi kejadian di masa depan, agar kita dapat membuat keputusan yang paling bijak untuk menghadapinya. Kita biasanya membuat keputusan berdasarkan informasi yang kita peroleh. Dalam beberapa kasus, kita memiliki data yang aktual dan obyektif, seperti laporan lalu lintas pagi hari atau prakiraan cuaca (Adam, 2011). Di lain waktu, kita mengandalkan intuisi dan pengalaman, seperti "Saya harus menghindari jembatan pagi ini karena biasanya jembatan tersebut macet pada saat jam berangkat kerja", atau ″Saya akan berbelanja pada saat *payday*, karena biasanya pada saat itu banyak barang yang diberi diskon″. Apa pun kasusnya, kita memprediksi kejadian di masa depan berdasarkan informasi dan pengalaman yang kita miliki saat ini, dan kita mengambil keputusan berdasarkan prediksi tersebut (Allgöwer & Zheng, 2012).

Ketika informasi semakin mudah diakses melalui internet, keinginan untuk memanfaatkannya dalam pengambilan keputusan semakin meningkat. Meskipun otak manusia, baik secara sadar maupun tidak sadar, mampu mengumpulkan data dalam jumlah besar, kemampuan otak untuk memproses informasi yang relevan dan mudah didapatkan dalam jumlah besar tetap terbatas dalam menyelesaikan masalah. Untuk membantu proses pengambilan keputusan, sekarang mengandalkan *platform* seperti Google untuk menemukan informasi yang paling tepat untuk pertanyaan dan Google Maps untuk merencanakan rute perjalanan terbaik, melihat kondisi lalu lintas secara *real-time*, dan menemukan tempat-tempat menarik yang ada di sekitar.

Platform yang disebutkan di atas mengumpulkan informasi, menyaring data untuk mengidentifikasi pola yang relevan dengan kebutuhan kita dan memberikan jawaban. *Platform* ini tidak hanya memanfaatkan informasi yang tersedia secara luas, tetapi juga menggunakan algoritma kompleks untuk menganalisis dan memproses data secara cepat dan efisien. *Platform* ini dapat menggabungkan data historis, tren saat ini, dan preferensi individu untuk menyajikan prediksi yang lebih personal dan akurat. Dengan demikian, *platform* ini tidak hanya berfungsi sebagai alat pencarian atau navigasi semata, tetapi juga sebagai solusi yang membantu untuk mengambil keputusan di kehidupan sehari-hari.

Di dalam bidang ilmu komputer, proses pengembangan *platform* semacam ini disebut sebagai "pembelajaran mesin", "kecerdasan buatan", "pengenalan pola", "penambangan data", dan "analisis prediktif" (Sarker, 2021). Tujuannya adalah untuk membuat prediksi yang akurat berdasarkan studi kasus yang digunakan dan disebut sebagai model prediktif. Definisi model prediktif adalah proses pengembangan alat atau model matematika yang bertujuan untuk menghasilkan prediksi yang akurat (Kuhn & Johnson, 2016).

Sebagai contoh aplikasi model prediktif dalam kehidupan sehari-hari adalah Netflix dan Spotify. Netflix yang merupakan perusahaan media perfilmnan menggunakan pemodelan prediktif untuk memberikan rekomendasi film yang disukai pemirsanya berdasarkan pilihan film yang telah diputar atau disukai sebelumnya (Amatriain, 2013). Apabila pemirsa sering menonton film dengan genre misteri, horor, dan *thriller*, maka Netflix akan merekomendasikan film dengan genre serupa sebagai prediksi bahwa pemirsa akan menyukainya. Selain itu, terdapat pula Spotify yang dapat memberikan rekomendasi untuk membantu pendengarnya menemukan musik baru yang sesuai dengan seleranya berdasarkan riwayat mendengarkan sebelumnya (Pareek et al., 2022).

Meskipun model prediktif dapat membantu untuk mendapatkan pelayanan yang lebih baik, terkadang model tersebut dapat menghasilkan prediksi yang kurang akurat sehingga memberikan jawaban yang salah (Lantz, 2019). Misalkan saja, dalam bidang kesehatan, sebuah model prediktif digunakan untuk memprediksi kemungkinan seorang pasien menderita penyakit tertentu berdasarkan data medisnya. Jika model tersebut tidak akurat, ada beberapa kemungkinan skenario yang bisa terjadi. Misalnya, model memprediksi bahwa pasien menderita penyakit padahal sebenarnya tidak. Dampaknya pasien mungkin akan menjalani tes dan perawatan yang tidak diperlukan. Bisa juga model memprediksi bahwa pasien tidak menderita penyakit padahal sebenarnya iya. Dampaknya pasien tidak mendapatkan perawatan yang dibutuhkan tepat waktu sehingga dapat memperburuk kondisi kesehatannya.

Pada bab ini akan dibahas mengenai matematika untuk proses pengembangan model prediktif dengan menggunakan studi kasus secara langsung. Model prediktif merupakan alat penting dalam analisis data, pembelajaran mesin, dan berbagai aplikasi untuk membuat prediksi atau estimasi mengenai data masa depan berdasarkan data historis (Brink et al., 2016). Proses pengembangan model prediktif melibatkan beberapa langkah matematis yang kompleks (Kuhn & Johnson, 2016). Adapun pembahasan mengenai matematika yang terlibat dalam pengembangan model prediktif akan dijelaskan langsung dengan menggunakan contoh studi kasus menggunakan algoritma *Decision Tree* dan K-*Nearest Neighbor* (K-NN).

4.2 Studi Kasus: Prediksi Perekrutan Karyawan

Sebuah perusahaan ingin meningkatkan efisiensi dan akurasi dalam proses perekrutan karyawannya. Setiap tahun, perusahaan menerima ribuan aplikasi untuk berbagai posisi, dan proses penyaringan serta penilaian kandidat memakan banyak waktu dan sumber daya. Oleh karena itu, perusahaan memutuskan untuk mengimplementasikan model prediktif untuk membantu dalam proses seleksi awal kandidat. Tujuan utama dari studi kasus ini adalah untuk membangun model prediktif yang dapat mengidentifikasi kandidat pelamar kerja yang paling berpotensi untuk dipekerjakan.

4.3 *Dataset*

Berdasarkan studi kasus di atas, dibutuhkan kumpulan data (*dataset*) untuk bisa membuat sistem prediksi perekrutan karyawan. Pada contoh kasus nyata di lapangan, perlu dikumpulkan data terkait studi kasus di suatu perusahaan. Misalkan, diminta data pelamar kerja dari bagian HRD selama 3 tahun terakhir agar data yang dikumpulkan valid dan *up to date*. Kemudian, data-data tersebut perlu dianalisis untuk menentukan atribut yang mempengaruhi keputusan bagian HRD untuk menerima kandidat pelamar kerja. Akan tetapi pada sub bab ini, akan digunakan sampel *dataset* untuk perekrutan karyawan.

Dataset ini memberikan wawasan tentang faktor - faktor yang mempengaruhi keputusan perekrutan. Setiap *record* data merepresentasikan seorang kandidat pelamar kerja dengan berbagai atribut yang dipertimbangkan selama proses perekrutan. Tujuannya adalah untuk memprediksi apakah seorang kandidat akan dipekerjakan berdasarkan atribut-atribut tersebut. Pada Tabel 4.1 dijelaskan detail atribut yang digunakan. Nomor 1 sampai 4 adalah atribut, sedangkan nomor 5 adalah kelas prediksi untuk menentukan hasil dari keputusan perekrutan (diterima atau tidak diterima).

Tabel 4.1 Detail atribut untuk prediksi perekrutan karyawan

no	atribut	rentang data	tipe data
1.	usia	20 – 50 tahun	integer
2.	skor wawancara	0 - 100	integer
3.	skor keterampilan	0 - 100	integer
4.	skor kepribadian	0 - 100	integer
5.	keputusan perekrutan	Tidak diterima (0) / diterima (1)	biner

Sebagai contoh untuk pengerjaan menggunakan algoritma prediksi di sub bab berikutnya, maka akan digunakan sampel *dataset* pada Tabel 4.2.

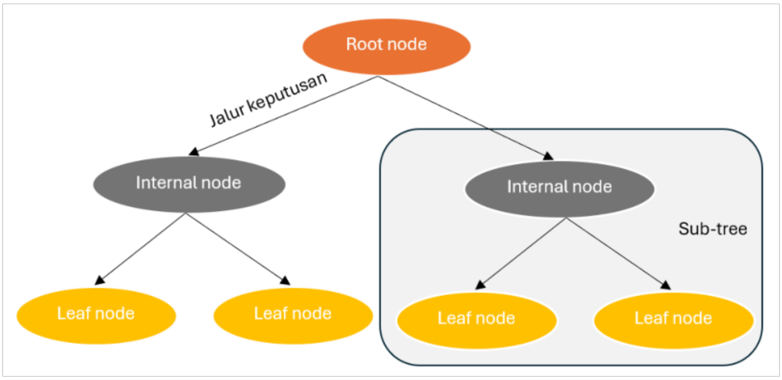
Tabel 4.2 Sampel *dataset* untuk prediksi perekrutan karyawan

no	usia	skor wawancara	skor keterampilan	skor kepribadian	keputusan
1.	25	70	78	91	1
2.	35	75	80	80	1
3.	25	40	40	60	0
4.	27	60	80	80	1
5.	35	75	80	80	1
6.	40	40	55	60	0
7.	40	75	60	50	1
8.	35	75	80	80	1
9.	27	40	55	60	0
10.	30	60	40	50	0

4.4 *Decision Tree*

Decision Tree adalah model *supervised learning*, yang belajar dari data historis untuk membuat prediksi pada data baru yang belum diketahui hasilnya. *Decision Tree* adalah model pembelajaran mesin yang dapat digunakan untuk klasifikasi dan regresi (Suthaharan, 2016). Apabila untuk klasifikasi, *Decision Tree* digunakan untuk memprediksi kategori atau kelas dari sebuah data input. Contohnya adalah memprediksi apakah seorang kandidat akan diterima atau tidak berdasarkan atribut tertentu. Apabila untuk regresi, *Decision Tree* juga bisa digunakan untuk memprediksi nilai numerik. Misalnya, memprediksi harga rumah berdasarkan atribut seperti luas bangunan, lokasi, dan jumlah kamar.

Decision Tree membuat prediksi dengan membuat serangkaian keputusan berdasarkan atribut data input. Pada setiap *node* di dalam pohon keputusan, atribut tertentu digunakan untuk memisahkan data menjadi *subset* yang lebih homogen (Tsang et al., 2009). Proses ini berlanjut sampai model mencapai keputusan akhir di *leaf node*. Struktur model *Decision Tree* dapat dilihat pada Gambar 4.1 (Fatimah & Rahmawati, 2021).



Gambar 4.1 Struktur model *Decision Tree*

Root node adalah *node* pertama dan paling atas. *Root node* merupakan titik awal ketika seluruh proses pemisahan data dimulai. Internal *node* adalah *node* yang berada di antara *root node* dan *leaf node*. Internal *node* digunakan untuk memisahkan data berdasarkan nilai dari atribut tertentu. *Leaf node* adalah *node* terakhir yang tidak memiliki cabang lebih lanjut. *Leaf node* memberikan keputusan akhir atau hasil dari jalur keputusan. Jalur keputusan adalah lintasan dari *root node* ke *leaf node*. Jalur ini terdiri dari serangkaian *node* dan cabang yang dilalui untuk mencapai keputusan akhir. Dalam Gambar 4.1, jalur keputusan ditunjukkan dengan garis yang menghubungkan *root node* ke *leaf node* melalui internal *node*. *Sub-tree* adalah bagian dari *Decision Tree* yang dimulai dari suatu internal *node* dan mencakup semua *node* di bawahnya. Untuk membuat model prediksi *Decision* Tree, digunakan beberapa rumus dan konsep penting dari teori informasi, khususnya konsep *Entropy* dan *Information Gain*. Berikut adalah rumus dan langkah-langkahnya secara detail:

1. *Entropy* (Pengukuran Ketidakpastian)

   Entropy adalah ukuran ketidakpastian atau *impurity* dalam sebuah *dataset*. *Entropy* untuk sebuah *dataset* S dengan c kelas yang berbeda dihitung sebagai (Sandag & Manuke, 2020):

$$Entropy\ (S) = \sum_{\mathrm{i}=1}^{\mathrm{c}} \mathrm{p_i} \log_2 (\mathrm{pi}) \qquad (4.1)$$

   dengan p_i adalah proporsi elemen dalam kelas i di *dataset* S.

2. *Information Gain* (Pengurangan Ketidakpastian)

   Information Gain adalah pengurangan ketidakpastian yang diperoleh dengan membagi *dataset* S berdasarkan atribut A. *Informasi Gain* $\mathrm{IG(S, A)}$ dihitung sebagai (Sandag & Manuke, 2020):

$$\mathrm{IG(S, A)} = Entropy\ (\mathrm{S}) - \sum \mathrm{v} \in Values(\mathrm{A}) \frac{|\mathrm{S_V}|}{|\mathrm{S}|} \times Entropy\ (\mathrm{S_V}) \qquad (4.2)$$

   dengan $Values(\mathrm{A})$ adalah set nilai-nilai unik dari atribut A, S_V adalah *subset* dari S di mana atribut A memiliki nilai v, |S| adalah jumlah elemen dalam *dataset* S, dan S_v adalah jumlah elemen dalam *subset* S_v.

Sedangkan langkah-langkah yang harus diikuti pada saat menggunakan *Decision Tree* adalah sebagai berikut:

1. Hitunglah *Entropy* dari *dataset* awal S
2. Hitunglah *Information Gain* untuk setiap atribut. Pada langkah ini, gunakan rumus *Information Gain* untuk menghitung pengurangan ketidakpastian.
3. Pilihlah atribut yang memiliki *Information Gain* tertinggi dan jadikan *root node* atau *node* cabang berikutnya.
4. Bagilah *dataset* berdasarkan atribut yang dipilih. Tiap *subset* akan menjadi cabang dari *node* tersebut.
5. Ulangi langkah 1-4 untuk setiap *subset* data sampai:
6. semua data dalam *subset* memiliki kelas yang sama,
7. tidak ada atribut lagi yang bisa dipecah,
8. tercapai kedalaman maksimum *tree* yang diinginkan

Untuk lebih memahami rumus dan langkah-langkah di atas, maka disediakan contoh manual model prediksi menggunakan *Decision Tree* yang disajikan pada sub bab 4.5 di bawah ini.

4.5 Contoh Manual Model Prediksi Menggunakan *Decision Tree*

Sebagai contoh untuk perhitungan manual, akan digunakan *dataset* pada Tabel 4.2. Terdapat 10 kandidat pelamar kerja dengan 4 atribut penilaian yaitu: usia, skor wawancara, skor keterampilan, skor kepribadian. Keempat atribut ini akan menentukan kelas prediksi (keputusan perekrutan). Pada atribut skor wawancara, skor keterampilan dan skor kepribadian terdapat *threshold* untuk menentukan nilai ambang batas. Berikut ini adalah langkah-langkahnya:

1. Hitung *Entropy* Awal (S)

   Total kelas Keputusan Perekrutan adalah 10 baris dengan 6 sampel positif (+) yang dinyatakan diterima dan 4 sampel negatif (-) yang dinyatakan tidak diterima, sehingga:

$$p_+ = \frac{6}{10} = 0.6$$
$$p_- = \frac{4}{10} = 0.4$$

$$\mathit{Entropy}\ (\mathrm{S}) = -0.6\log_2(0.6) - 0.4\log_2(0.4) = \mathbf{0.971}$$

2. Hitung *Entopy* dan *Information Gain* untuk tiap atributnya

   a. Atribut Usia

      Usia = 25, terdapat 2 kandidat yang berusia 25 tahun, 1 kandidat dinyatakan diterima sedangkan 1 kandidat lainnya tidak diterima (Total: 2; Positif: 1; Negatif: 1), sehingga:

$$p_+ = \frac{1}{2} = 0.5$$
$$p_- = \frac{1}{2} = 0.5$$

$$\mathit{Entropy}\ (\mathrm{Usia} = 25) = -0.5\log_2(0.5) - 0.5\log_2(0.5) = 1$$

      Usia = 27, terdapat 2 kandidat yang berusia 27 tahun, 1 kandidat dinyatakan diterima sedangkan 1 kandidat lainnya tidak diterima (Total: 2; Positif: 1; Negatif: 1), sehingga:

$$p_+ = \frac{1}{2} = 0.5$$
$$p_- = \frac{1}{2} = 0.5$$

$$\mathit{Entropy}\ (\mathrm{Usia} = 27) = -0.5\log_2(0.5) - 0.5\log_2(0.5) = 1$$

      Usia = 30, terdapat 1 kandidat yang berusia 30 tahun, 0 kandidat dinyatakan diterima sedangkan 1 kandidat tersebut dinyatakan tidak diterima (Total: 1; Positif: 0; Negatif: 1), sehingga:

$$p_{+} = \frac{0}{1} = 0$$

$$p_{-} = \frac{1}{1} = 1$$

$$Entropy\ (\text{Usia} = 30) = -0 \log_2(0) - 1 \log_2(1) = 0$$

Usia = 35, terdapat 3 kandidat yang berusia 35 tahun, 3 kandidat dinyatakan diterima sedangkan 0 kandidat lainnya tidak diterima (Total: 3; Positif: 3; Negatif: 0), sehingga:

$$p_{+} = \frac{3}{3} = 1$$

$$p_{-} = \frac{0}{3} = 0$$

$$Entropy\ (\text{Usia} = 35) = -1 \log_2(1) - 0 \log_2(0) = 0$$

Usia = 40, terdapat 2 kandidat yang berusia 40 tahun, 1 kandidat dinyatakan diterima sedangkan 1 kandidat lainnya tidak diterima (Total: 2; Positif: 1; Negatif: 1), sehingga:

$$p_{+} = \frac{1}{2} = 0.5$$

$$p_{-} = \frac{1}{2} = 0.5$$

$$Entropy\ (\text{Usia} = 40) = -0.5 \log_2(0.5) - 0.5 \log_2(0.5) = 1$$

Setelah diketahui semua nilai *Entropy* untuk atribut usia, maka langkah selanjutnya adalah mencari *Information Gain* yang ditunjukkan sebagai berikut ini:

$$\text{Gain}\ (\text{S}, \text{Usia}) = 0.971 - (2/10\ \text{x}\ 1 + 2/10\ \text{x}\ 1 + 1/10\ \text{x}\ 0 + 3/10\ \text{x}\ 0 + 2/10\ \text{x}\ 1)$$

$$= 0.971 - (0.2 \times 1 + 0.2 \times 1 + 0.1 \times 0 + 0.3 \times 0 + 0.2 \times 1)$$

$$= 0.971 - 0.6$$

$$= \mathbf{0.371}$$

b. Atribut Skor Wawancara

Skor Wawancara ≤ 60, terdapat 5 kandidat yang memiliki skor wawancara kurang dari sama dengan 60, 1 kandidat dinyatakan diterima sedangkan 4 kandidat lainnya tidak diterima (Total: 5; Positif: 1; Negatif: 4), sehingga:

$$p_+ = \frac{1}{5} = 0.2$$

$$p_- = \frac{4}{5} = 0.8$$

$$Entropy\ (\text{Skor Wawancara} \leq\ 60)$$
$$= -0.2 \log_2(0.2) - 0.8 \log_2(0.8)$$
$$= 0.7219$$

Skor Wawancara > 60, terdapat 5 kandidat yang memiliki skor wawancara lebih dari 60, 5 kandidat dinyatakan diterima sedangkan 0 kandidat lainnya tidak diterima (Total: 5; Positif: 5; Negatif: 0), sehingga:

$$p_+ = \frac{5}{5} = 1$$

$$p_- = \frac{0}{5} = 0$$

$$Entropy\ (\text{Skor Wawancara} >\ 60)$$
$$= -1 \log_2(1) - 0 \log_2(0) = 0$$

Setelah diketahui semua nilai *Entropy* untuk atribut wawancara, maka langkah selanjutnya adalah mencari *Information Gain* yang ditunjukkan sebagai berikut ini:

$$Gain\ (S, \text{Skor Wawancara})$$
$$= 0.971 - \left(\frac{5}{10} \text{ x } 0.7219 + \frac{5}{10} \text{ x } 0\right)$$
$$= 0.971 - 0.361 = \mathbf{0.61}$$

c. Atribut Skor Keterampilan

Skor Keterampilan $\leq$ 55, terdapat 4 kandidat yang memiliki skor keterampilan kurang dari sama dengan 55, 0 kandidat dinyatakan diterima sedangkan 4 kandidat lainnya tidak diterima (Total: 4; Positif: 0; Negatif: 4), sehingga:

$$\text{p}_+ = \frac{0}{4} = 0$$
$$\text{p}_- = \frac{4}{4} = 1$$

$$Entropy\ (\text{Skor Keterampilan} \leq\ 55)$$
$$= -0 \log_2(0) - 1 \log_2(1) = 0$$

Skor Keterampilan $>$ 55, terdapat 6 kandidat yang memiliki skor keterampilan lebih dari 55, 6 kandidat dinyatakan diterima sedangkan 0 kandidat lainnya tidak diterima (Total: 6; Positif: 6; Negatif: 0), sehingga:

$$\text{p}_+ = \frac{6}{6} = 1$$
$$\text{p}_- = \frac{0}{6} = 0$$

$$Entropy\ (\text{Skor Keterampilan} > 55)$$
$$= -1 \log_2(1) - 0 \log_2(0) = 0$$

Setelah diketahui semua nilai *Entropy* untuk atribut keterampilan, maka langkah selanjutnya adalah mencari *Information Gain* yang ditunjukkan sebagai berikut ini:

$Gain$ (S, Skor Keterampilan)

$$= 0.971 - \left(\frac{4}{10} \text{ x } 0 + \frac{6}{10} \text{ x } 0\right)$$

$$= 0.971 - 0 = \mathbf{0.971}$$

d. Atribut Skor Kepribadian

Skor Kepribadian ≤ 60, terdapat 5 kandidat yang memiliki skor kepribadian kurang dari sama dengan 60, 1 kandidat dinyatakan diterima sedangkan 4 kandidat lainnya tidak diterima (Total: 5; Positif: 1; Negatif: 4), sehingga:

$$p_{+} = \frac{1}{5} = 0.2$$

$$p_{-} = \frac{4}{5} = 0.8$$

$Entropy$ (Skor Keterampilan $\leq$ 60)

$$= -0.2 \, log_2(0.2) - 0.8 \, log_2(0.8)$$

$$= 0.722$$

Skor Kepribadian > 60, terdapat 5 kandidat yang memiliki skor kepribadian lebih dari 60, 5 kandidat dinyatakan diterima sedangkan 0 kandidat lainnya tidak diterima (Total: 5; Positif: 5; Negatif: 0), sehingga:

$$\mathrm{p}_{+} = \frac{5}{5} = 1$$

$$\mathrm{p}_{-} = \frac{0}{5} = 0$$

$Entropy$ (Skor Kepribadian > 60)

$$= -1 \log_2(1) - 0 \log_2(0) = 0$$

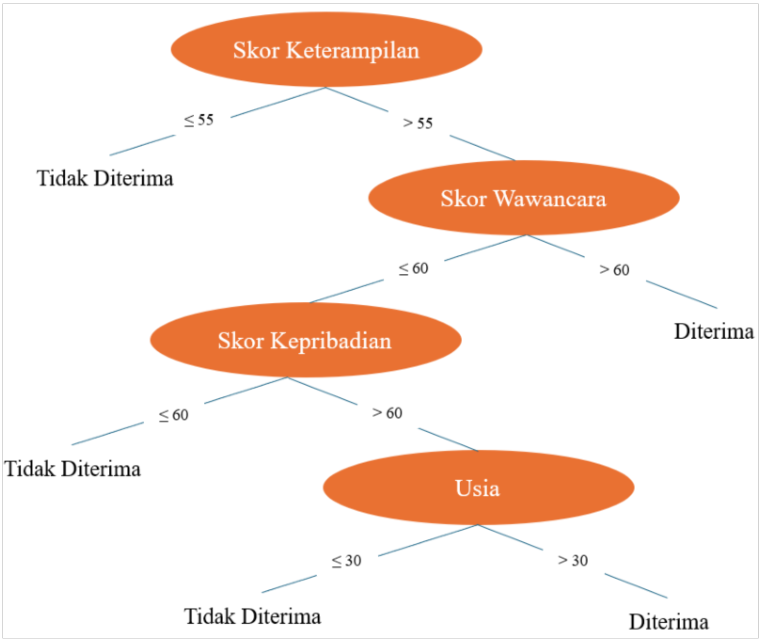
Setelah diketahui semua nilai *Entropy* untuk atribut kepribadian, maka langkah selanjutnya adalah mencari *Information Gain* yang ditunjukkan sebagai berikut:

$$Gain\ (\text{S, Skor Kepribadian})$$
$$= 0.971 - \left(\frac{5}{10}\ x\ 0.722 + \frac{5}{10}\ x\ 0\right)$$
$$= 0.971 - 0.361 = \mathbf{0.61}$$

Berdasarkan perhitungan *Information Gain* yang telah dilakukan, berikut ini adalah peringkat nilai *Information Gain* dari yang terbesar ke terkecil:

- Skor keterampilan memiliki *information gain* = **0.97**
- Skor wawancara memiliki *information gain* = **0.61**
- Skor kepribadian memiliki *information gain* = **0.61**
- Usia memiliki *information gain* = **0.371**

3. Menggambarkan struktur pohon keputusan



Gambar 4.2 Struktur pohon keputusan prediksi perekrutan

Atribut yang memiliki nilai *Information Gain* tertinggi akan menjadi *root node*, dan atribut dengan nilai *Information Gain* berikutnya akan menjadi internal *node*, sampai kita mencapai *leaf node* yang memberikan keputusan akhir. Gambar 4.2 menggambarkan struktur pohon keputusan yang dihasilkan. Berdasarkan Gambar 4.2, Skor Keterampilan adalah atribut yang paling penting dalam menentukan keputusan penerimaan berdasarkan *dataset* yang diberikan. Skor Wawancara dan Skor Kepribadian juga berpengaruh signifikan setelah Skor Keterampilan. Dan yang terakhir adalah usia juga turut berpengaruh pada keputusan penerimaan. Berikut adalah penjelasan dari Gambar 4.2:

a. *Root Node* adalah atribut Skor Keterampilan karena memiliki nilai *Information Gain* tertinggi.

b. Cabang kiri dari *root node* adalah untuk nilai Skor Keterampilan ≤ 55 dan langsung mengarah pada keputusan Tidak Diterima karena semua data pada cabang ini memiliki keputusan 0.

c. Cabang kanan dari *root node* adalah untuk nilai Skor Keterampilan > 55 dan ini dibagi lagi berdasarkan atribut Skor Wawancara.

d. Jika Skor Wawancara ≤ 60, maka kita mengecek Skor Kepribadian:

   - Jika Skor Kepribadian ≤ 60, keputusan adalah Tidak Diterima.
   - Jika Skor Kepribadian > 60, keputusan adalah Diterima.

e. Jika Skor Wawancara > 60, keputusan langsung adalah Diterima.

Dengan pohon keputusan ini, bisa diprediksi keputusan penerimaan kandidat pelamar kerja berdasarkan nilai dari atribut-atribut yang ada. Pohon keputusan memberikan cara yang mudah dipahami untuk menginterpretasikan data dan membuat keputusan berdasarkan informasi yang paling relevan. Pohon keputusan ini dapat digunakan sebagai model prediktif untuk mempermudah proses seleksi calon yang sesuai dengan kriteria yang telah ditentukan. Hal ini juga menunjukkan bahwa metode manual untuk menghitung *Entropy* dan *Information Gain* dapat membantu dalam memahami proses pembentukan model prediktif.

4.6 K-*Nearest Neighbor*

K-*Nearest Neighbors* (KNN) adalah salah satu metode klasifikasi yang digunakan dalam pengembangan model prediktif (Zhang, 2016). KNN termasuk dalam kategori *supervised learning* yang berarti algoritma ini memerlukan data latih yang telah diberi label untuk mempelajari pola yang ada (Ying, et al., 2021). Prinsip dasar dari KNN adalah menentukan kelas suatu data uji berdasarkan kelas mayoritas dari K tetangga terdekatnya (Shokrzade, et al., 2021). Untuk mengukur kedekatan atau kemiripan antara data uji dan data latih, digunakan jarak *Euclidean*, yang menghitung jarak linear antara dua titik dalam ruang atribut multidimensi. Berikut ini merupakan rumus *Euclidean* untuk mengukur jarak terdekat (Hu, et al., 2016):

$$d\ (x, y) = \sum_{i=1}^{n} (x_i - \ y_i)^2 \tag{4.3}$$

Langkah pertama dalam penerapan KNN adalah menentukan nilai K, yaitu jumlah tetangga terdekat yang akan dipertimbangkan. Setelah itu, dihitung jarak *Euclidean* antara data uji dan setiap data latih. Kemudian, data latih diurutkan berdasarkan jarak tersebut, dan K data latih dengan jarak terdekat dipilih. Kelas yang paling sering muncul di antara K tetangga tersebut menjadi prediksi untuk data uji (Boateng, et al., 2020). Kelebihan utama KNN adalah kesederhanaannya dan kemampuannya untuk menangani data non-linear. Sedangkan kelemahannya termasuk sensitif terhadap nilai K yang dipilih dan dapat menjadi tidak efisien ketika menghadapi *dataset* besar (Suyanto, 2017).

4.7 Contoh Manual Model Prediksi Menggunakan K-NN

Dalam studi kasus pengembangan model prediktif, seperti penilaian kelayakan calon karyawan berdasarkan usia, skor wawancara, skor keterampilan, dan skor kepribadian, KNN dapat membantu memprediksi keputusan penerimaan dengan mempertimbangkan pola dari data karyawan sebelumnya. Proses ini melibatkan perhitungan jarak untuk menentukan kedekatan antara kandidat pelamar kerja dan karyawan yang sudah ada, sehingga memberikan keputusan yang didasarkan pada informasi empiris yang ada.

Penggunaan *dataset* pada Tabel 4.2 sebagai data latih untuk memberikan hasil klasifikasi pada data uji yang terdapat di Tabel 4.3 di bawah ini. Pada Tabel 4.3, nilai pada kolom keputusan masih tanda tanya (?) dikarenakan belum bisa ditentukan keputusannya (Diterima 1/Tidak Diterima 0). Untuk mengisi kolom keputusan ini maka dapat digunakan algoritma K-NN.

Tabel 4.3 Data uji untuk prediksi perekrutan karyawan

usia	skor wawancara	skor keterampilan	skor kepribadian	keputusan
25	70	78	91	?

Berikut ini langkah-langkah menggunakan K-NN:

1. Tentukan nilai parameter K, misalnya kita pilih $K = 3$
2. Hitunglah jarak *Euclidean* untuk tiap atribut terhadap data uji

Rumus *Euclidean* sesuai *dataset*:

$$d = \sqrt{(Usia_u - Usia_i)^2 + (SkorW_u - SkorW_i)^2 + (SkorKet_u + (SkorKep_u - SkorKep_i)^2} \quad (4.4)$$

dengan $Usia_u, SkorKet_u, SkorKep_u$ adalah nilai atribut dari data uji sedangkan $Usia_i$, $SkorW_i$, $SkorKet_i$, $SkorKep_i$ adalah nilai atribut data latih, sehingga:

- Data latih 1

$$d_1 = \sqrt{(31 - 25)^2 + (65 - 70)^2 + (75 - 78)^2 + (80 - 91)^2}$$
$$= \sqrt{6^2 + (-5)^2 + (-3)^2 + (-11)^2}$$
$$= \sqrt{36 + 25 + 9 + 121} = \sqrt{191} = \mathbf{13.82}$$

- Data latih 2

$$d_2 = \sqrt{(31 - 35)^2 + (65 - 75)^2 + (75 - 80)^2 + (80 - 80)^2}$$
$$= \sqrt{(-4)^2 + (-10)^2 + (-5)^2 + 0^2}$$
$$= \sqrt{16 + 100 + 25 + 0} = \sqrt{141} = \mathbf{11.87}$$

- Data latih 3

$$d_3 = \sqrt{(31 - 25)^2 + (65 - 40)^2 + (75 - 40)^2 + (80 - 60)^2}$$
$$= \sqrt{6^2 + 25^2 + 35^2 + 20^2}$$

$$= \sqrt{36 + 625 + 1225 + 400} = \sqrt{2286} = \mathbf{47.81}$$

- Data latih 4

$$d_4 = \sqrt{(31 - 27)^2 + (65 - 60)^2 + (75 - 80)^2 + (80 - 80)^2}$$
$$= \sqrt{4^2 + 5^2 + (-5)^2 + 0^2}$$
$$= \sqrt{16 + 25 + 25 + 0} = \sqrt{66} = \mathbf{8.12}$$

- Data latih 5

$$d_5 = \sqrt{(31 - 35)^2 + (65 - 75)^2 + (75 - 80)^2 + (80 - 80)^2}$$
$$= \sqrt{(-4)^2 + (-10)^2 + (-5)^2 + 0^2}$$
$$= \sqrt{16 + 100 + 25 + 0} = \sqrt{141} = \mathbf{11.87}$$

- Data latih 6

$$d_6 = \sqrt{(31 - 40)^2 + (65 - 40)^2 + (75 - 55)^2 + (80 - 60)^2}$$
$$= \sqrt{(-9)^2 + 25^2 + 20^2 + 20^2}$$
$$= \sqrt{81 + 625 + 400 + 0} = \sqrt{1506} = \mathbf{38.81}$$

- Data latih 7

$$d_7 = \sqrt{(31 - 40)^2 + (65 - 75)^2 + (75 - 60)^2 + (80 - 50)^2}$$
$$= \sqrt{(-9)^2 + (-10)^2 + 15^2 + 30^2}$$
$$= \sqrt{81 + 100 + 225 + 900} = \sqrt{1306} = \mathbf{36.14}$$

- Data latih 8

$$d_8 = \sqrt{(31 - 35)^2 + (65 - 75)^2 + (75 - 80)^2 + (80 - 80)^2}$$
$$= \sqrt{(-4)^2 + (-10)^2 + 5^2 + 0^2}$$
$$= \sqrt{16 + 100 + 25 + 0} = \sqrt{141} = \mathbf{11.87}$$

- Data latih 9

$$d_9 = \sqrt{(31 - 27)^2 + (65 - 40)^2 + (75 - 55)^2 + (80 - 60)^2}$$

$$= \sqrt{4^2 + 25^2 + 20^2 + 20^2}$$

$$= \sqrt{16 + 625 + 400 + 400} = \sqrt{1441} = \mathbf{37.93}$$

- Data latih 10

$$d_{10} = \sqrt{(31 - 30)^2 + (65 - 60)^2 + (75 - 40)^2 + (80 - 50)^2}$$

$$= \sqrt{1^2 + 5^2 + 35^2 + 30^2}$$

$$= \sqrt{1 + 25 + 1225 + 900} = \sqrt{2151} = \mathbf{46.39}$$

3. Urutkan objek terdekat berdasarkan jarak *Euclidean* terkecil ke terbesar. Tabel 4.4 menunjukkan urutan rangking berdasarkan jarak *Euclidean* yang telah dihitung pada poin 2 sebelumnya.

Tabel 4.4 Urutan rangking jarak *Euclidean*

data latih ke-i	*distance*	rangking	keputusan
4	8.12	1	1
2	11.87	2	1
5	11.87	3	1
8	11.87	4	1
1	13.82	5	1
6	38.81	6	0
7	36.14	7	1
9	37.93	8	0
10	46.39	9	0
3	47.81	10	0

4. Pilihlah objek terdekat dengan jumlah K yang sudah ditentukan. Pada poin 1 kita sudah menentukan $K = 3$, sehingga data yang ditampilkan adalah rangking 1 sampai 3 saja sesuai yang ditunjukkan Tabel 4.5.

Tabel 4.5 Rangking sesuai jumlah K

data latih ke-i	*distance*	rangking	keputusan
4	8.12	1	1
2	11.87	2	1
5	11.87	3	1

5. Gunakan kategori kelas mayoritas untuk menentukan hasil klasifikasi. Dari tiga tetangga terdekat, semua memiliki keputusan 1 (Diterima). Oleh karena itu, keputusan untuk data uji adalah: 1 (Diterima) sesuai yang ditunjukkan Tabel 4.6.

Tabel 4.6 Hasil klasifikasi menggunakan KNN

Usia	Skor Wawancara	Skor Keterampilan	Skor Kepribadian	Keputusan
31	65	75	80	1

Dengan menggunakan metode K-*Nearest Neighbors* (KNN) pada *dataset* yang diberikan, dihitung jarak *Euclidean* antara data uji dan setiap data latih. Setelah mengurutkan jarak tersebut, dipilih K tetangga terdekat dan menentukan keputusan berdasarkan mayoritas dari tetangga terdekat tersebut. Keputusan untuk data uji dalam contoh ini adalah diterima (1).

Studi kasus di atas menggunakan dua algoritma pemodelan prediktif, yaitu *Decision Tree* dan K-*Nearest Neighbors* (KNN). Kedua algoritma ini digunakan untuk menganalisis dan memprediksi keputusan perekrutan karyawan berdasarkan beberapa atribut seperti usia, skor wawancara, skor keterampilan, dan skor kepribadian. Melalui perhitungan manual, telah dilihat bagaimana masing-masing algoritma berfungsi dan bagaimana matematika digunakan dalam proses tersebut. Melalui studi kasus ini, dapat dilihat bagaimana matematika memainkan peran kunci dalam algoritma pemodelan prediktif. *Decision Tree* menggunakan konsep *Entropy* dan *Information Gain* untuk membuat keputusan yang optimal di setiap *node*, sementara KNN menggunakan jarak *Euclidean* untuk mengukur kemiripan dan menentukan kelas berdasarkan tetangga terdekat. Pemahaman dan penerapan matematika dalam algoritma ini memungkinkan untuk dibangun model prediktif yang akurat dan dapat diandalkan, yang penting untuk pengambilan keputusan dalam berbagai domain, termasuk perekrutan karyawan.

Bab **5**

Konsep Matematika Pada Statistik dan Probabilitas

5.1 Statistik dan Probabilitas

Penggunaan statistik dan probabilitas dalam kecerdasan buatan terus berkembang dengan sangat pesat, terutama didorong oleh kemajuan dalam teori statistik dan probabilitas, peningkatan dalam teknologi komputasi, dan ketersediaan data dalam jumlah besar. Dalam bab ini, akan dibahas mengenai berbagai konsep dan teknik statistik dan probabilitas yang secara umum banyak digunakan dalam kecerdasan buatan.

Statistik dan probabilitas merupakan salah satu landasan utama dalam kecerdasan buatan (*Artificial Inteligence*, AI) dan pembelajaran mesin (*machine learning*) (LeCun *et al.*, 2015). Metode statistik memungkinkan analisis data yang efektif dan validasi model, sedangkan konsep probabilitas mendasari banyak algoritma yang digunakan dalam kecerdasan buatan (Dangeti, 2017). Pemahaman yang baik tentang statistik dan probabilitas dinilai sangat penting untuk mengembangkan model kecerdasan buatan yang akurat, dapat diandalkan, dan dapat diinterpretasikan. Dalam konteks kecerdasan buatan, statistik digunakan untuk memahami data melalui analisis deskriptif dan inferensial, sementara probabilitas digunakan untuk membuat prediksi dan mengambil keputusan dalam tingkat ketidakpastian tertentu. Saat ini, banyak algoritma pembelajaran mesin, seperti regresi, klasifikasi, dan klasterisasi, mengandalkan prinsip-prinsip statistik dan probabilistik untuk mengidentifikasi pola dan membuat prediksi berdasarkan data historis (Sarker, 2021).

Berbagai metode statistik dan probabilitas banyak bergantung pada penggunaan peralatan-peralatan berbasis matematika, ketika persamaan-persamaan sering kali dibuat secara lebih umum untuk diterapkan dalam berbagai keperluan. Penggunaan konsep matematika pada statistik dan probabilitas diterapkan untuk tujuan optimasi dan menghindari penilaian secara bias. Mengingat luasnya bidang matematika yang dapat digunakan untuk keperluan statistik dan probabilitas, kecenderungan untuk terus mengembangkan algoritma yang semakin efisien dan akurat terus dilakukan. Selain sebagai "jangkar" daripada statistik dan probabilitas, matematika melalui serangkaian penerapan aljabar linier mampu menyusun algoritma yang dinilai sangat *robust* untuk mengukur, memprediksi, dan mengambil keputusan terhadap data dalam jumlah yang sangat banyak dan telah membantu manusia dalam berbagai keperluan, baik pada skala yang sangat teknis maupun kehidupan sehari-hari.

5.2 Sejarah Singkat

Penggunaan statistik dan probabilitas dalam kecerdasan buatan memiliki sejarah yang cukup panjang yang dapat ditelusuri kembali ke beberapa dekade yang lalu (Stigler, 2002). Beberapa tonggak penting dalam sejarah ini yaitu meliputi:

1. Tahun 1950-an dan 1960-an: Pada masa awal perkembangan kecerdasan buatan, peralatan statistika banyak digunakan, terutama dalam pengolahan data (*data processing*) dan pengenalan pola (*pattern recognition*). Model probabilistik seperti distribusi Gaussian mulai digunakan untuk memodelkan data dan membuat prediksi.

2. Tahun 1970-an dan 1980-an: Metode statistik dan probabilistik mulai diterapkan secara lebih luas dalam pembelajaran mesin. Algoritma regresi linear dan logistik, yang didasarkan pada prinsip-prinsip statistik, menjadi populer. Pada saat yang sama, konsep probabilistik seperti Teorema Bayes mulai diterapkan dalam algoritma pengenalan pola dan klasifikasi.
3. Tahun 1990-an: Pengenalan jaringan saraf (*neural network*) Bayesian dan model Markov tersembunyi (*Hidden Markov Models* - HMM) menandai kemajuan besar dalam penggunaan probabilitas dalam kecerdasan buatan dan pembelajaran mesin. Algoritma ini memungkinkan pemodelan yang lebih canggih dan akurat dari data yang kompleks, terutama dalam pengenalan suara dan teks.
4. Tahun 2000-an dan seterusnya: Dengan munculnya konsep *big data* serta peningkatan daya komputasi yang signifikan dengan berkembang pesatnya teknologi prosesor komputer, penggunaan statistik dan probabilitas dalam kecerdasan buatan semakin meluas. Metode seperti pengambilan keputusan berbasis probabilitas, pohon keputusan (*decission tree*), dan algoritma pembelajaran *ensemble* seperti *Random Forest* dan *Boosting*, menjadi bagian penting dari peralatan pembelajaran mesin.

5.3 Dasar-dasar Probabilitas dan Distribusi

Probabilitas mengukur sejauh mana suatu peristiwa kemungkinan terjadi dan menyediakan kerangka untuk memahami dan mengelola ketidakpastian dalam data (Ross, 2014). Konsep dasar probabilitas mencakup aturan-aturan dasar seperti probabilitas kondisional, hukum probabilitas total, dan teorema Bayes, yang memungkinkan kita untuk menghitung dan memprediksi kemungkinan kejadian berdasarkan informasi yang tersedia. Selain itu, distribusi probabilitas menggambarkan bagaimana probabilitas didistribusikan di seluruh kemungkinan nilai dari variabel acak, dan meliputi berbagai jenis distribusi seperti distribusi normal, binomial, dan Poisson.

Memahami dasar-dasar probabilitas dan distribusi adalah kunci untuk membangun model yang akurat dan efektif dalam berbagai aplikasi, mulai dari analisis data hingga pengembangan algoritma pembelajaran mesin. Sub-bab ini bertujuan untuk memberikan landasan yang solid dalam konsep-konsep ini, sehingga mempersiapkan pembaca untuk memahami teknik-teknik statistik yang lebih kompleks dan aplikasinya dalam kecerdasan buatan.

5.3.1 Definisi dan konsep probabilitas dan distribusi

Dalam suatu probabilitas, terlebih dahulu diperlukan pemahaman mengenai analisis kombinatorial, yang mana dapat membantu banyak hal dalam probabilitas, termasuk bilangan faktorial, permutasi, dan kombinasi (Sumarlin, 2023). Bilangan faktorial adalah bilangan hasil kali dari sebuah bilangan bulat positif dengan semua bilangan bulat positif yang kurang dari bilangan tersebut. Umumnya bilangan ini dinotasikan dengan tanda seru (!), misalnya 5! (dibaca "lima faktorial"), yang sama dengan $5 \times 4 \times 3 \times 2 \times 1 = 120$. Bilangan faktorial sering digunakan dalam permutasi dan kombinatorika.

Secara umum, permutasi adalah penyusunan jumlah (n) objek yang diambil sebanyak (r) objek tertentu dengan memperhatikan penyusunannya. Permutasi mengacu pada pengaturan dan susunan objek-objek dalam suatu himpunan, di mana urutan objek-objek tersebut menjadi penting. Misalnya, 3 buah bola berwarna merah, hijau, dan biru, maka berapa banyak cara yang dimungkinkan untuk mengatur bola-bola tersebut secara berbeda dalam sebuah garis? Jawabannya adalah $3 \times 2 \times 1 = 6$, karena setiap bola memiliki tiga tempat yang mungkin untuk ditempatkan pada urutan pertama, kemudian dua tempat untuk urutan kedua, dan satu tempat untuk urutan ketiga.

Sedangkan kombinasi mengacu pada pemilihan objek-objek dari suatu himpunan, ketika urutan objek-objek tersebut tidak menjadi penting. Dalam kombinasi, objek-objek dapat dipilih dalam urutan apa pun. Sebagai contoh, jika ada 4 buah bola berwarna merah, hijau, biru, dan kuning, dan ingin dipilih 2 bola dari keempat bola tersebut, berapa banyak cara yang mungkin? Jawabannya adalah 6, karena 6 adalah kemungkinan kombinasi yang berbeda: merah dan hijau, merah dan biru, merah dan kuning, hijau dan biru, hijau dan kuning, serta biru dan kuning. Rumus untuk menghitung permutasi dan kombinasi adalah sebagai berikut:

$$nPr = \frac{n!}{(n-r)!} \tag{5.1}$$

$$nCr = \frac{n!}{(r!\,(n-r)!)} \tag{5.2}$$

Dengan n adalah jumlah objek dalam himpunan, dan r adalah jumlah objek yang diatur (dalam kasus permutasi) atau dipilih (dalam kasus kombinasi).

Selanjutnya, probabilitas adalah ukuran dari kemungkinan terjadinya suatu peristiwa. Probabilitas sebuah kejadian A dilambangkan dengan P(A), dan nilai ini berada di antara 0 dan 1, yaitu:

- P(A) = 0 menunjukkan bahwa kejadian tersebut tidak mungkin terjadi.
- P(A) = 1 menunjukkan bahwa kejadian tersebut pasti terjadi.

Ruang Sampel (S) adalah himpunan semua hasil yang mungkin terjadi dari suatu percobaan, sering juga disebut dengan himpunan dari titik sampel. Sedangkan, kejadian (A) adalah himpunan bagian dari ruang sampel.

1. Hukum Penjumlahan: yaitu jika A dan B adalah dua kejadian yang saling eksklusif/unik, maka:

$$P(A \cup B) = P(A) + P(B) \tag{5.3}$$

Misalnya, jika sebuah dadu dilemparkan, maka probabilitas munculnya mata dadu 4 (A) dan munculnya mata dadu lebih kecil dari 3 adalah:

$$P(A) = \frac{1}{6}, \qquad P(B) = \frac{2}{6}$$

$$P(A \cup B) = P(A) + P(B)$$
$$= \frac{1}{6} + \frac{2}{6} = \frac{3}{6} = \frac{1}{2} = 0{,}5$$

2. Hukum Perkalian: yaitu jika A dan B adalah dua kejadian yang independen, maka hukum perkaliannya yaitu $P(A \cap B) = P(A) \times P(B)$. Misalnya sebuah mata uang logam dan sebuah dadu dilemparkan satu kali secara bersamaan. Maka probabilitas munculnya sisi muka pada uang logam dan mada dadu 4 adalah:

$$P(A \cap B) = P(A) \times P(B) \tag{5.4}$$

$$P(A \cap B) = \frac{1}{2} \times \frac{1}{6} = \frac{1}{12}$$

5.3.2 Probabilitas bersyarat dan Teorema Bayes

Probabilitas bersyarat merupakan suatu prinsip dalam teori probabilitas, prinsip ini mengacu pada probabilitas terjadinya suatu peristiwa tertentu berdasarkan fakta bahwa peristiwa sebelumnya telah terjadi . Misalnya, probabilitas dari kejadian A terjadi dengan syarat bahwa B telah terjadi, dilambangkan sebagai P(A|B) :

$$P(A|B) = \frac{A \cap B}{P(B)} \tag{5.5}$$

Sementara itu, Teorema Bayes memberikan cara untuk memperbarui probabilitas suatu hipotesis berdasarkan bukti baru, yang dilambangkan dengan persamaan:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{5.6}$$

Teorema ini sangat penting dalam banyak aplikasi kecerdasan buatan, terutama dalam algoritma pengklasifikasian seperti Naive Bayes.

5.3.3 Distribusi diskrit

1. Distribusi binomial

   Distribusi ini menggambarkan jumlah sukses dalam sejumlah tetap percobaan independen yang masing-masing memiliki dua kemungkinan hasil (sukses atau gagal). Jika X adalah variabel acak binomial dengan n percobaan dan probabilitas sukses p, maka:

   $$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \tag{5.7}$$

   dengan $k = 0, 1, 2, \dots, n$.

2. Distribusi Poisson

   Distribusi ini digunakan untuk menggambarkan jumlah kejadian dalam interval waktu atau ruang tertentu yang terjadi secara independen dengan laju rata-rata λ. Jika X adalah variabel acak Poisson dengan parameter λ, maka:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{!} \tag{5.8}$$

dengan $k = 0, 1, 2, \ldots, n$.

5.3.4 Distribusi kontinu

1. Distribusi normal

   Distribusi ini sangat penting dalam statistik dan banyak digunakan karena banyak fenomena alam mengikuti distribusi ini. Distribusi normal dengan rerata μ dan standar deviasi σ memiliki fungsi kepadatan probabilitas (*probability density function*, PDF):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.9}$$

2. Distribusi eksponensial

   Distribusi ini sering digunakan untuk memodelkan waktu antara kejadian dalam proses Poisson. Jika X adalah variabel acak eksponensial dengan parameter laju λ, maka PDF-nya adalah:

$$f(x) = \lambda\, e^{-\lambda x}\{-\lambda\, x\} \tag{5.10}$$

   untuk $x \geq 0$.

5.3.5 Fungsi Distribusi Kumulatif (CDF) dan Fungsi Kepadatan Probabilitas (PDF)

Cumulative Distribution Function (CDF) dan *Probability Density Function* (PDF) adalah komponen fundamental dalam analisis probabilistik. Keduanya berperan penting dalam memodelkan dan memahami distribusi probabilitas dari suatu data. CDF memberikan informasi tentang probabilitas kumulatif hingga titik tertentu dalam data, sedangkan PDF menguraikan bagaimana probabilitas terdistribusi di sekitar nilai-nilai yang berbeda dalam suatu rentang. Dalam konteks kecerdasan buatan, CDF dan PDF digunakan dalam berbagai algoritma untuk membuat prediksi yang lebih akurat dan membantu dalam pengambilan keputusan yang didasarkan pada analisis data yang mendalam. Dengan memanfaatkan alat ini, algoritma dapat menangani ketidakpastian dan variabilitas dalam data dengan lebih efektif, yang pada akhirnya meningkatkan kinerja model kecerdasan buatan atau pembelajaran mesin.

1. Fungsi Distribusi Kumulatif (CDF)

   CDF $F(x)$ dari variabel acak X memberikan probabilitas bahwa X akan lebih kecil atau sama dengan x:

$$F(x) = P(X \leq x) \tag{5.11}$$

2. Fungsi Kepadatan Probabilitas (PDF)

   Untuk variabel acak kontinu, PDF $f(x)$ menggambarkan kepadatan probabilitas di sekitar nilai x. Integral dari PDF memberikan CDF:

$$F(x) = \int_{-\infty}^{x} f(t)\, dt \tag{5.12}$$

5.4 Statistik Deskriptif dan Inferensi Statistik

Statistik deskriptif dan inferensi statistik merupakan komponen kunci dalam analisis data. Statistik deskriptif memberikan gambaran umum mengenai data, sementara inferensi statistik memungkinkan dibuatnya generalisasi dan simpulan tentang populasi berdasarkan sampel data. Kedua metode ini esensial dalam kecerdasan buatan untuk memahami data dan membuat prediksi yang akurat.

5.4.1 Ukuran pemusatan dan penyebaran

Ukuran Pemusatan merujuk pada serangkaian nilai yang menggambarkan posisi tengah atau pusat dari distribusi data. Nilai-nilai ini memberikan gambaran tentang lokasi sentral ketika data cenderung berkumpul atau berfokus. Dengan memahami ukuran pemusatan, bisa diperoleh wawasan tentang bagaimana data terdistribusi di sekitar titik-titik kunci ini, yang sangat penting dalam berbagai analisis statistik dan aplikasinya. Ukuran-ukuran ini mencakup beberapa konsep penting, di antaranya adalah:

1. *Mean* (Rerata), yaitu umlah dari semua nilai dibagi dengan jumlah data. Rerata dapat dirumuskan dengan

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (5.13)$$

2. Median, yaitu nilai tengah dari data yang diurutkan. Jika jumlah data ganjil, median adalah nilai tengah. Jika jumlah data genap, median adalah rata-rata dari dua nilai tengah.
3. Mode: Nilai yang paling sering muncul dalam data.

Ukuran penyebaran mengukur sejauh mana data tersebar atau bervariasi dari nilai pusatnya. Ini memberikan gambaran tentang tingkat keragaman atau dispersi dalam suatu dataset, yang membantu kita memahami distribusi data lebih dalam. Ukuran penyebaran ini penting untuk memahami variasi dalam data dan untuk melakukan analisis lebih lanjut, seperti mendeteksi *outlier* atau menilai konsistensi data di sekitar nilai pusatnya. Ukuran penyebaran mencakup beberapa metrik penting, seperti:

1. *Range*: Selisih antara nilai maksimum dan minimum.

$$Range = x_{max} - x_{min} \tag{5.14}$$

2. *Variance* (Varians): Rata-rata dari kuadrat selisih setiap nilai dengan mean.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{5.15}$$

3. *Standard Deviation* (Simpangan Baku): Akar kuadrat dari varians.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{5.16}$$

5.4.2 Estimasi titik dan interval

Estimasi titik dan interval adalah dua pendekatan utama dalam statistik inferensial yang digunakan untuk menyimpulkan karakteristik populasi berdasarkan data sampel. Estimasi titik memberikan perkiraan tunggal dari parameter populasi, seperti rata-rata atau proporsi, yang dianggap sebagai representasi terbaik dari nilai sebenarnya. Namun, karena pengambilan sampel melibatkan ketidakpastian, estimasi titik tidak dapat memberikan gambaran lengkap tentang akurasi estimasi tersebut. Oleh karena itu, estimasi interval digunakan untuk melengkapi estimasi titik dengan menyediakan rentang nilai yang lebih realistis, yang mencerminkan ketidakpastian inheren dalam pengukuran tersebut. Interval ini, yang dikenal sebagai interval kepercayaan, mencakup kemungkinan lokasi parameter populasi dan dilengkapi dengan tingkat kepercayaan tertentu, seperti 95% atau 99%. Kombinasi dari kedua jenis estimasi ini memungkinkan kita untuk membuat inferensi yang lebih akurat dan terpercaya tentang populasi, sekaligus mengakomodasi variabilitas yang mungkin terjadi dalam proses sampling.

1. Estimasi Titik adalah nilai tunggal yang digunakan untuk memperkirakan parameter populasi. Estimasi Mean Populasi: Menggunakan mean sampel ($\bar{x}$) untuk mengestimasi mean populasi (μ).

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x^{i} \tag{5.17}$$

2. Estimasi Interval memberikan rentang nilai yang kemungkinan besar mengandung parameter populasi. Interval Kepercayaan: Rentang nilai yang berasosiasi dengan tingkat kepercayaan tertentu, biasanya 95%.

$$CI = \overline{x \pm z \frac{\sigma}{\sqrt{n}}} \tag{5.18}$$

dengan z adalah nilai z dari distribusi normal standar untuk tingkat kepercayaan yang diinginkan.

5.4.3 Uji hipotesis

Uji hipotesis adalah metode statistik yang digunakan untuk membuat keputusan atau menarik kesimpulan tentang suatu populasi berdasarkan data sampel (Raftery et al., 1995). Proses ini melibatkan pengujian dugaan awal, yang dikenal sebagai hipotesis nol (H_0), terhadap alternatif yang mungkin, yaitu hipotesis alternatif (H_a). Uji hipotesis digunakan untuk menentukan apakah ada cukup bukti dalam data sampel untuk menolak hipotesis nol demi hipotesis alternatif. Melalui pendekatan ini, dapat diukur kemungkinan bahwa hasil yang diamati terjadi secara kebetulan, dengan mempertimbangkan tingkat signifikansi tertentu. Uji hipotesis memiliki aplikasi luas dalam berbagai bidang, mulai dari penelitian ilmiah hingga pengambilan keputusan bisnis, dan merupakan alat penting dalam memastikan validitas klaim atau pernyataan yang didasarkan pada data.

1. Hipotesis Nol (H_0): Pernyataan awal yang diasumsikan benar.
2. Hipotesis Alternatif (H_a): Pernyataan yang akan diterima jika bukti menunjukkan H_0 salah.

Langkah-langkah dalam uji hipotesis:

1. Formulasi hipotesis: tentukan H_0 dan H_a.
2. Pilih tingkat signifikansi (α): Biasanya 0,05 atau 0,01.
3. Hitung statistik uji: misalnya, uji t untuk mean.

$$t = \frac{\bar{x}\mu_0}{s/\sqrt{n}} \tag{5.19}$$

   dengan $\bar{x}$ adalah rerata yang dihipotesiskan, s adalah simpangan baku sampel, dan n adalah ukuran sampel.

4. Tentukan nilai kritis atau p-value:
   a. p-value: probabilitas mendapatkan hasil yang sama atau lebih ekstrem dari yang diamati, jika H_0 benar.
   b. Nilai kritis: nilai batas dari distribusi uji statistik untuk tingkat signifikansi yang dipilih.
5. Buat Keputusan:
   a. Jika p-value $\leq \alpha$, tolak H_0.
   b. Jika p-value $> \alpha$, gagal menolak H_0.

Contoh uji hipotesis:

1. Uji t satu sampel: menguji apakah mean sampel berbeda dari mean populasi yang diketahui.
2. Uji t dua sampel: menguji apakah mean dari dua sampel berbeda secara signifikan.

3. Uji Chi-square: menguji independensi antara dua variabel kategori atau kesesuaian distribusi sampel dengan distribusi yang diharapkan.

5.5 Regresi dan Korelasi

Regresi dan korelasi adalah teknik statistik penting dalam analisis data dan kecerdasan buatan. Regresi memungkinkan prediksi nilai variabel dependen berdasarkan variabel independen, sementara korelasi mengukur kekuatan dan arah hubungan antara dua variabel (Miles dan Shevlin, 2000). Evaluasi model regresi dengan metrik yang tepat membantu memastikan model yang dikembangkan akurat dan andal.

5.5.1. Regresi linier sederhana

Regresi linear sederhana digunakan untuk memodelkan hubungan antara dua variabel, satu variabel independen (X) dan satu variabel dependen (Y). Model ini mengasumsikan bahwa hubungan antara X dan Y dapat dijelaskan dengan garis lurus:

$$Y = \beta_0 + \beta_1 X + \epsilon \qquad (5.20)$$

dengan:

- Y adalah variabel dependen.
- X adalah variabel independen.
- β_0 adalah intercept (titik potong sumbu-Y).
- β_1 adalah slope (kemiringan garis).
- ϵ adalah error term.

Tujuan regresi linear adalah untuk menemukan estimasi β_0 dan β_1 yang meminimalkan jumlah kuadrat *error* (*residual sum of squares*, RSS):

$$RSS = \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_1)\right)^2 \qquad (5.21)$$

Metode kuadrat terkecil (*Least Squares*) biasanya digunakan untuk menemukan estimasi parameter ini.

5.5.2 Regresi linier berganda

Regresi linear berganda memperluas konsep regresi linear sederhana untuk mencakup lebih dari satu variabel independen:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \qquad (5.22)$$

dengan:

- Y adalah variabel dependen.
- $X1, X2, \dots, Xp$ adalah variabel independen.
- β_0 adalah intercept.
- $\beta_1, \beta_2, \dots, \beta_p$ adalah koefisien regresi untuk masing-masing variabel independen.
- ϵ adalah *error term*.

Estimasi parameter dalam regresi linear berganda juga dilakukan menggunakan metode Least Squares, dengan tujuan meminimalkan RSS.

5.5.3 Analisis korelasi

Korelasi mengukur kekuatan dan arah hubungan linier antara dua variabel. Korelasi tidak menyiratkan sebab akibat tetapi menunjukkan adanya hubungan antara variabel.

1. Koefisien Korelasi Pearson (r) mengukur kekuatan dan arah hubungan linier antara dua variabel kontinu:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})\sum_{i=1}^{n}(y_i - \bar{y})}} \qquad (5.23)$$

   Nilai r berkisar antara -1 dan 1, dengan ketentuan:

   - $r = 1$ menunjukkan hubungan linier positif sempurna.

- $r = -1$ menunjukkan hubungan linier negatif sempurna.
- $r = 0$ menunjukkan tidak ada hubungan linier.

2. Koefisien Korelasi Spearman (ρ) mengukur kekuatan dan arah hubungan monotoonik antara dua variabel ordinal atau tidak berdistribusi normal. Korelasi Spearman adalah korelasi Pearson yang dihitung pada peringkat data.

5.5.4 Pengujian model statistik

Evaluasi model statistik penting untuk menentukan seberapa baik model memprediksi atau menjelaskan data. Beberapa metrik evaluasi untuk model regresi meliputi:

1. Koefisien Determinasi (R^2): Mengukur proporsi variabilitas dalam variabel dependen yang dapat dijelaskan oleh variabel independen dalam model. Nilai R^2 berkisar antara 0 dan 1.

$$R^2 = 1 - \frac{RSS}{TSS} \tag{5.24}$$

2. Adjusted R^2: Mengoreksi R^2 untuk jumlah variabel dalam model, memberikan ukuran yang lebih adil untuk model dengan sejumlah besar variabel independen.
3. Mean Absolute Error (MAE): Rata-rata dari absolut nilai error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}| \tag{5.25}$$

4. Mean Squared Error (MSE): Rata-rata dari kuadrat nilai error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{5.26}$$

5. Root Mean Squared Error (RMSE): Akar kuadrat dari MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (5.27)$$

5.6 Teori Keputusan dan Metode Nonparametrik

Teori keputusan dan metode nonparametrik merupakan alat penting dalam analisis statistik dan kecerdasan buatan. Pohon keputusan dan algoritma Naive Bayes menyediakan cara untuk membuat keputusan berdasarkan data dan probabilitas, sementara uji nonparametrik memberikan fleksibilitas dalam analisis data yang tidak memenuhi asumsi distribusi tertentu (Berger, 2013). Analisis keputusan memungkinkan evaluasi tindakan dalam situasi ketidakpastian, membantu pengambilan keputusan yang lebih informasional.

5.6.1 Pohon Keputusan

Pohon keputusan adalah model prediktif yang digunakan untuk membuat keputusan berdasarkan data. Model ini membagi data ke dalam subset yang semakin kecil berdasarkan aturan keputusan yang berulang hingga mencapai keputusan akhir.

1. Komponen Pohon Keputusan:
   - Akar: Node awal yang mewakili seluruh dataset.
   - Cabang: Jalur dari satu node ke node berikutnya.
   - Node Internal: Node yang merepresentasikan tes atau keputusan berdasarkan satu atribut.
   - Daun (*Leaf*): Node terminal yang memberikan hasil akhir atau keputusan.

2. Algoritma Pembentukan Pohon Keputusan:
   - ID3 (*Iterative Dichotomiser* 3): Menggunakan entropi dan informasi gain untuk memilih atribut yang membagi data dengan optimal.
   - C4.5: Perbaikan dari ID3 yang menangani atribut kontinu dan missing values.
   - CART (*Classification and Regression Trees*): Menghasilkan pohon keputusan biner untuk klasifikasi dan regresi.
3. Kriteria Pemilihan Atribut:
   - Informasi *Gain*: Mengukur pengurangan ketidakpastian atau entropi setelah membagi dataset.
   - *Gini Index*: Mengukur *impurity* dari node.
   - *Chi-Square*: Mengukur asosiasi antara atribut dan kelas.

5.6.2 Algoritma Naive Bayes

Algoritma Naive Bayes adalah metode klasifikasi yang sederhana namun sangat efektif, yang didasarkan pada teorema Bayes dengan asumsi independensi yang kuat antara fitur-fitur dalam data. Meskipun asumsi independensi ini jarang terpenuhi sepenuhnya dalam praktik, algoritma ini tetap menunjukkan kinerja yang baik dalam berbagai aplikasi, seperti pengenalan teks, klasifikasi email spam, dan analisis sentimen. Naive Bayes bekerja dengan menghitung probabilitas dari setiap kelas berdasarkan fitur-fitur yang diberikan, kemudian memilih kelas dengan probabilitas tertinggi sebagai prediksi. Kesederhanaan dan efisiensi dari Naive Bayes membuatnya menjadi pilihan populer dalam situasi ketika kecepatan dan interpretabilitas lebih diutamakan daripada kompleksitas model. Meskipun sederhana, Naive Bayes sering memberikan hasil yang kompetitif dibandingkan dengan algoritma klasifikasi yang lebih kompleks, terutama pada dataset yang besar dan berdimensi tinggi.

1. Teorema Bayes:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{5.28}$$

   dengan:

   - $P(C|X)$ adalah probabilitas kelas C pada atribut X.
   - $P(X|C)$ adalah probabilitas atribut X pada kelas C.
   - $P(C)$ adalah probabilitas awal kelas C.
   - $P(X)$ adalah probabilitas atribut X.

2. Dalam algoritma Naive Bayes terdapat asumsi idependensi, yang dapat dihitung menggunakan persamaan:

$$P(X|C) = \prod_{i=1}^{n} P(x_i|C) \tag{5.29}$$

   dengan x_i adalah fitur individual.

3. Algoritma Naive Bayes memiliki beberapa keunggulan, yaitu sebagai berikut:

   - Cepat dan efisien.
   - Cocok untuk dataset besar.
   - Menghasilkan performa yang baik meskipun asumsi independensi tidak sepenuhnya terpenuhi.

5.6.3 Uji Nonparametrik

Uji nonparametrik adalah metode statistik yang digunakan ketika asumsi-asumsi dasar uji parametrik, seperti distribusi normal atau homoskedastisitas, tidak terpenuhi oleh data. Uji nonparametrik tidak memerlukan asumsi tentang distribusi populasi dari mana sampel diambil, sehingga lebih fleksibel dalam penerapannya, terutama ketika berhadapan dengan data yang berskala ordinal, memiliki *outlier*, atau sampel berukuran kecil. Metode ini meliputi berbagai uji seperti uji Wilcoxon, uji Mann-Whitney, dan uji Kruskal-Wallis, yang semuanya dirancang untuk menguji hipotesis tentang median, distribusi, atau peringkat data tanpa mengandalkan parameter populasi tertentu. Karena sifatnya yang lebih umum, uji nonparametrik sering digunakan dalam analisis data yang tidak memenuhi syarat untuk uji parametrik, memberikan alternatif yang *robust* dan reliabel untuk menguji hipotesis dan membuat simpulan statistik dalam berbagai konteks penelitian.

1. Uji Mann-Whitney:
    - Digunakan untuk membandingkan dua sampel independen.
    - Hipotesis nol: Tidak ada perbedaan antara dua distribusi populasi.
    - Uji statistik:

    $$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad (5.30)$$

    dengan n_1 dan n_2 adalah ukuran sampel, dan R_1 adalah jumlah peringkat dalam sampel pertama.

2. Uji Kruskal-Wallis:
    - Perluasan dari uji Mann-Whitney untuk lebih dari dua grup.
    - Hipotesis nol: Semua grup berasal dari distribusi yang sama.
    - Uji statistik:

$$H = \frac{12}{N(N+1)}\sum\nolimits_{i=1}^{k}\frac{R_i^2}{n_1} - 3(N+1) \qquad (5.31)$$

dengan N adalah total ukuran sampel, k adalah jumlah grup, R_i adalah jumlah peringkat dalam grup ke-i, dan n_i adalah ukuran grup ke-i.

5.6.4 Analisis Keputusan

Analisis keputusan adalah proses sistematis yang digunakan untuk membuat pilihan yang optimal di antara berbagai alternatif berdasarkan data dan informasi yang tersedia. Proses ini melibatkan penilaian risiko, manfaat, dan kemungkinan hasil dari setiap opsi, serta mempertimbangkan faktor-faktor ketidakpastian yang dapat mempengaruhi keputusan. Dalam analisis keputusan, berbagai alat dan teknik, seperti pohon keputusan, analisis biaya-manfaat, dan analisis sensitivitas, digunakan untuk menyusun dan mengevaluasi pilihan secara kuantitatif maupun kualitatif.

Tujuan dari analisis ini adalah untuk membantu pengambil keputusan memilih tindakan yang paling sesuai dengan tujuan yang ingin dicapai, baik itu dalam konteks bisnis, manajemen, kebijakan publik, atau bidang lainnya. Dengan pendekatan yang terstruktur, analisis keputusan memungkinkan para pengambil keputusan untuk mengevaluasi trade-off yang ada, mengidentifikasi potensi risiko, dan memilih strategi yang memaksimalkan nilai atau utilitas dalam situasi yang kompleks dan sering kali penuh ketidakpastian.

1. Beberapa aspek terkait matriks keputusan:
   - Merupakan tabel yang menampilkan hasil dari berbagai tindakan dalam situasi yang berbeda.
   - Setiap entri dalam matriks mewakili nilai hasil (*payoff*) untuk kombinasi tindakan dan keadaan.

2. Beberapa aspek terkait pohon keputusan:
   - Digunakan untuk memvisualisasikan keputusan dan hasil yang mungkin dalam bentuk pohon.
   - Setiap cabang mewakili tindakan atau keputusan, dan setiap daun mewakili hasil atau *payoff*.
3. Beberapa aspek terkait kriteria keputusan:
   - Maximin: memilih tindakan dengan nilai minimum terbesar.
   - Maximax: memilih tindakan dengan nilai maksimum terbesar.
   - Hurwicz: kombinasi dari maximin dan maximax, dengan parameter optimisme (konsep bias optimisme, yang terkait dengan evaluasi model).
   - Laplace: mengasumsikan semua kejadian sama kemungkinan, memilih tindakan dengan nilai ekspektasi tertinggi.
   - Minimax Regret: meminimalkan penyesalan maksimum dari tidak memilih tindakan terbaik.

5.7 Analisis Komponen Utama dan Reduksi Dimensi

Principal Component Analysis (PCA) dan *Linear Discriminant Analysis* (LDA) adalah teknik utama dalam reduksi dimensi, masing-masing dengan aplikasi dan keunggulan yang spesifik. PCA lebih umum digunakan untuk mereduksi dimensi data kontinu dengan mempertahankan variabilitas, sedangkan LDA lebih cocok untuk tugas klasifikasi dengan memaksimalkan separasi antar kelas (Wu, 2014). Keduanya membantu dalam menyederhanakan data dan model, meningkatkan interpretabilitas, dan memperbaiki performa analisis data dan kecerdasan buatan.

5.7.1 Konsep Dasar Reduksi Dimensi

Reduksi dimensi adalah proses mengurangi jumlah variabel acak yang dipertimbangkan dengan mendapatkan sekumpulan variabel utama yang lebih kecil. Ini bertujuan untuk menyederhanakan model tanpa kehilangan informasi signifikan. Beberapa manfaat dari reduksi dimensi meliputi:

- Mengurangi kompleksitas model, membuatnya lebih mudah dipahami.
- Mengurangi risiko overfitting.
- Mempercepat waktu pemrosesan dan analisis data.
- Menghilangkan noise dari data.

5.7.2 Analisis Komponen Utama (PCA)

Analisis Komponen Utama (PCA) adalah teknik statistik yang digunakan untuk mereduksi dimensi data sambil mempertahankan sebanyak mungkin variabilitas dalam dataset. PCA mengubah data ke dalam set baru dari variabel yang tidak berkorelasi yang disebut komponen utama.

Langkah-langkah PCA:

1. Standarisasi Data: Skala variabel sehingga masing-masing memiliki mean 0 dan varians 1.
2. Kalkulasi Matriks Kovarians: Menghitung kovarians antar variabel dalam data.

$$Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) \tag{5.32}$$

3. *Eigen decomposition*: Menghitung eigen vektor dan eigen *value* dari matriks kovarians.
   - Eigen vektor: Arah dari komponen utama.
   - Eigen *value*: Besaran variabilitas sepanjang masing-masing eigenvektor.

4. Pilih Komponen Utama: Urutkan eigen vektor berdasarkan eigen *value* dan pilih komponen utama dengan eigen *value* terbesar.

5. Transformasi Data: Proyeksikan data asli ke ruang baru yang dibentuk oleh komponen utama yang dipilih.

   Rumus Transformasi PCA: $Z = XW$

   dengan X adalah data asli, W adalah matriks eigen vektor, dan Z adalah data yang ditransformasikan.

5.7.3 Pemilihan Jumlah Komponen Utama

Menentukan jumlah komponen utama yang optimal adalah langkah kritis dalam PCA. Beberapa metode yang digunakan meliputi:

1. *Screen Plot*: *plotting eigenvalue* dari setiap komponen utama dan mencari "*elbow point*" di mana penurunan eigenvalue mulai melambat.

2. *Kaiser's Criterion*: memilih komponen utama dengan eigenvalue lebih besar dari 1.

3. *Cumulative Explained Variance*: memilih jumlah komponen yang menjelaskan persentase tertentu dari variabilitas total (misalnya, 90-95%).

5.7.4 *Linear Discriminant Analysis* (LDA)

Linear Discriminant Analysis (LDA) adalah teknik reduksi dimensi yang mencari kombinasi linier dari fitur yang memaksimalkan separasi antara dua atau lebih kelas. LDA sering digunakan untuk klasifikasi dan memperbaiki performa model dengan mengurangi noise.

Langkah-langkah LDA:

1. Standarisasi Data: seperti PCA, data harus distandarisasi.

2. Hitung Mean dan Scatter Matrix: hitung mean dari masing-masing kelas dan scatter matrix antar kelas.

- Within-class scatter matrix (S_W): menunjukkan penyebaran dalam kelas.

$$S_W = \sum_{i=1}^{c} S_i \tag{5.33}$$

dengan S_i adalah scatter matrix untuk kelas ke-i.

- Between-class scatter matrix (S_B): menunjukkan penyebaran antar kelas.

$$S_B = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{5.34}$$

dengan n_i adalah jumlah sampel dalam kelas ke-i, μ_i adalah mean kelas ke-i, dan μ adalah mean keseluruhan.

3. *Eigen decomposition*: Menghitung eigen vektor dan eigen *value* dari matriks $S_W^{-1}S_B$.
4. Pilih Komponen Diskriminan: pilih eigen vektor dengan eigen *value* terbesar yang memaksimalkan separasi antar kelas.
5. Transformasi Data: proyeksikan data asli ke ruang baru yang dibentuk oleh komponen discriminant yang dipilih.

5.7.5 Penggunaan Reduksi Dimensi dalam Kecerdasan Buatan

Reduksi dimensi memainkan peran penting dalam kecerdasan buatan dan machine learning. Beberapa aplikasi utamanya meliputi:

1. Pra-pemrosesan Data: mengurangi dimensi data sebelum melatih model kecerdasan untuk meningkatkan efisiensi dan performa.
2. Visualisasi Data: mengurangi data ke dalam 2 atau 3 dimensi untuk memudahkan visualisasi dan interpretasi.

3. *Noise Filtering*: menghilangkan variabel yang tidak relevan atau *noise* dari data, yang dapat meningkatkan akurasi model.

5.8 Metode Monte Carlo dan Simulasi

Metode Monte Carlo adalah alat yang sangat kuat dalam analisis data dan kecerdasan buatan, memberikan cara fleksibel dan efisien untuk menangani masalah kompleks yang melibatkan ketidakpastian dan variabilitas. Monte Carlo merupakan sekelompok teknik komputasi yang menggunakan *sampling* acak untuk memperoleh solusi numerik dari masalah matematika atau fisika yang kompleks (Rubinstein dan Kroese, 2016). Metode ini dinamai berdasarkan kasino Monte Carlo di Monako, yang dikenal karena permainan peluangnya, karena elemen acak dalam teknik ini mirip dengan permainan di kasino. Simulasi Monte Carlo merupakan penerapan dari metode Monte Carlo ketika teknik ini digunakan untuk memodelkan probabilitas dari berbagai hasil dalam proses yang tidak dapat ditentukan secara deterministik, terutama ketika menghadapi ketidakpastian. Metode ini sering digunakan dalam masalah yang sulit atau tidak mungkin dipecahkan secara analitis.

5.8.1 Sejarah dan Aplikasi

Metode ini pertama kali dikembangkan selama Perang Dunia II oleh matematikawan seperti Stanislaw Ulam dan John von Neumann sebagai alat untuk memecahkan masalah kompleks dalam fisika nuklir. Nama "Monte Carlo" diambil dari kasino Monte Carlo di Monako, karena metode ini bergantung pada prinsip acak yang mirip dengan permainan peluang. Sejak saat itu, teknik ini telah berkembang pesat dan digunakan dalam berbagai bidang, termasuk keuangan, teknik, ilmu komputer, dan ilmu data. Dalam keuangan, Monte Carlo digunakan untuk menilai risiko portofolio dan harga opsi; dalam teknik, untuk memodelkan sistem yang kompleks dan mengoptimalkan desain; dan dalam ilmu komputer, untuk algoritma seperti simulasi *annealing* dan pencarian pohon keputusan. Aplikasi metode Monte Carlo yang luas dan kemampuannya untuk menangani ketidakpastian dan kompleksitas menjadikannya alat yang sangat berharga dalam analisis dan pengambilan keputusan berbasis data.

Aplikasi Metode Monte Carlo:

1. Fisika dan Kimia: memodelkan perilaku partikel, dinamika molekul.
2. Keuangan: menilai risiko dan harga opsi, simulasi portofolio.
3. Teknik: analisis keandalan, optimasi desain.
4. Ilmu Komputer: algoritma optimasi, pemrosesan citra.
5. Statistika: estimasi integral, pengujian hipotesis.

5.8.2 Prinsip Dasar Metode Monte Carlo

Prinsip dasar metode Monte Carlo berfokus pada penggunaan sampling acak untuk menyelesaikan masalah matematis dan statistik yang kompleks, terutama ketika metode analitik konvensional tidak memadai. Metode ini didasarkan pada konsep sederhana bahwa, dengan melakukan simulasi acak dalam jumlah besar, dapat mendekati solusi dari suatu masalah atau menghitung estimasi probabilitas dengan akurasi yang memadai. Proses ini melibatkan tiga langkah utama: (1) pembuatan sampel acak dari distribusi yang relevan, (2) penerapan fungsi atau model yang ingin dievaluasi pada sampel tersebut, dan (3) analisis hasil untuk memperoleh estimasi atau prediksi. Dengan memanfaatkan hukum bilangan besar, metode Monte Carlo memungkinkan untuk memperoleh hasil yang mendekati solusi yang tepat melalui repetisi simulasi, meskipun data yang digunakan bersifat acak dan tidak deterministik. Prinsip ini membuat Monte Carlo sangat berguna dalam berbagai aplikasi, dari perhitungan risiko finansial hingga simulasi sistem fisik yang kompleks.

Langkah-langkah umum dalam metode Monte Carlo:

1. Definisikan Model: Tentukan fungsi atau proses yang akan disimulasikan.
2. Generate Sampel Acak: Buat sampel acak dari distribusi yang relevan.
3. Hitung Hasil: Gunakan sampel untuk menghitung hasil.
4. Analisis Hasil: Evaluasi hasil simulasi untuk mendapatkan estimasi.

Contoh sederhana adalah memperkirakan nilai π menggunakan titik acak dalam lingkaran yang terinskripsi dalam bujur sangkar:

1. Generate titik acak dalam bujur sangkar.
2. Hitung proporsi titik yang jatuh dalam lingkaran.
3. Gunakan proporsi ini untuk memperkirakan π.

5.8.3 Simulasi Monte Carlo dalam Statistik

Dalam konteks statistik, simulasi Monte Carlo memungkinkan analisis yang mendalam terhadap model statistik yang kompleks, ketika solusi analitik mungkin sulit atau tidak mungkin dicapai. Proses ini melibatkan pembuatan banyak sampel acak dari distribusi yang diharapkan, penerapan model atau fungsi statistik pada sampel tersebut, dan kemudian menganalisis hasilnya untuk memahami distribusi kemungkinan hasil dan estimasi parameter. Dengan cara ini, dapat dievaluasi ketidakpastian, memodelkan berbagai skenario, dan memperoleh interval kepercayaan yang lebih akurat. Simulasi Monte Carlo juga berguna dalam memperkirakan p-value, menganalisis distribusi hasil dari uji hipotesis, dan mengatasi masalah kompleks seperti estimasi variansi dalam model non-linier. Teknik ini sangat berharga dalam statistik karena kemampuannya untuk menangani masalah yang melibatkan ketidakpastian dan variabilitas dengan cara yang fleksibel dan dapat diterapkan pada berbagai jenis data dan model. Beberapa contoh simulasi Monte Carlo dalam statistik:

1. Estimasi Integral: menghitung integral yang kompleks dengan sampling acak.

$$I = \int_a^b f(x)dx \approx \frac{b-a}{n}\sum_{i=1}^{n} f(x_i) \tag{5.35}$$

   dengan x_i adalah sampel acak dalam interval $[a, b]$

2. Pengujian Hipotesis: menguji hipotesis dengan simulasi distribusi dari statistik uji.
3. Estimasi Interval Kepercayaan: menyediakan estimasi interval kepercayaan dengan bootstrap atau resampling.

5.8.4 Metode Monte Carlo Markov Chain (MCMC)

Metode Monte Carlo Markov Chain (MCMC) adalah teknik lanjutan dalam statistika dan pemodelan probabilistik yang digunakan untuk menghasilkan sampel dari distribusi probabilitas kompleks dan tidak terdistribusi dengan cara yang sulit dihitung secara langsung. MCMC menggabungkan prinsip Monte Carlo dengan proses Markov untuk memungkinkan eksplorasi sistem yang memiliki dimensi tinggi dan kompleksitas tinggi (Carlo, 2004). Teknik ini beroperasi dengan membangun rantai Markov, ketika setiap langkah bergantung pada langkah sebelumnya, dan menghasilkan sampel yang mendekati distribusi target setelah periode iterasi yang memadai. MCMC sangat berguna dalam situasi ketika distribusi posterior sulit untuk diperoleh secara analitik atau ketika data memiliki struktur yang kompleks. Dengan menerapkan algoritma seperti Metropolis-Hastings atau Gibbs Sampling, MCMC memungkinkan para peneliti untuk memodelkan dan menganalisis data dalam berbagai aplikasi, mulai dari inferensi bayesian hingga estimasi parameter dalam model statistik yang rumit. Teknik ini menjadi alat yang sangat penting dalam statistika modern karena kemampuannya untuk menangani berbagai bentuk distribusi dan memberikan solusi yang akurat dalam masalah inferensi statistik yang kompleks.

Langkah-langkah umum dalam MCMC:

1. Inisialisasi: Pilih titik awal.
2. Proposal: Generate titik proposal baru dari distribusi proposal.
3. *Acceptance/Rejection*: Tentukan apakah titik proposal diterima atau ditolak berdasarkan kriteria probabilitas.

Algoritma MCMC yang populer:

1. Metropolis-Hastings: Menggunakan distribusi proposal untuk menghasilkan titik baru dan memutuskan penerimaan berdasarkan rasio probabilitas.
2. Gibbs Sampling: Menyampel setiap variabel secara bergantian dari distribusi kondisionalnya.

5.8.5 Implementasi Metode Monte Carlo dalam Kecerdasan Buatan

Metode Monte Carlo telah menjadi alat yang sangat berharga dalam kecerdasan buatan (AI) karena kemampuannya untuk menangani ketidakpastian dan kompleksitas dalam model dan algoritma. Dalam konteks AI, metode ini digunakan untuk memodelkan dan mengevaluasi berbagai skenario dengan menghasilkan sampel acak dari distribusi yang relevan dan menganalisis hasilnya. Metode Monte Carlo membantu dalam berbagai aplikasi AI, seperti pengoptimalan algoritma, estimasi nilai fungsi objektif dalam algoritma pembelajaran mesin, dan pengembangan strategi dalam pengambilan keputusan otomatis. Teknik ini memungkinkan para praktisi AI untuk mengeksplorasi ruang solusi yang luas dan menangani masalah yang melibatkan variabilitas dan ketidakpastian data dengan lebih efektif. Dengan menggunakan simulasi Monte Carlo, AI dapat memprediksi hasil yang lebih akurat, mengidentifikasi pola tersembunyi dalam data, dan mengembangkan model yang lebih robust dalam menghadapi situasi yang kompleks dan dinamis. Metode Monte Carlo digunakan dalam berbagai aplikasi kecerdasan buatan:

1. Optimasi dan Pencarian:
   - Simulated Annealing: algoritma optimasi yang menggunakan Metode Monte Carlo untuk menemukan minimum global dari fungsi objektif.
   - Randomized Search: pencarian hiperparameter model dengan sampling acak.
2. Pembelajaran Penguatan (*reinforcement learning*):
   - Monte Carlo *Tree Search* (MCTS): digunakan dalam game AI untuk memodelkan kemungkinan langkah dan memilih langkah optimal.
   - Estimasi Nilai: menggunakan sampel acak untuk memperkirakan nilai fungsi dalam pembelajaran penguatan.
3. Model Generatif:
   - Bayesian *Networks*: estimasi parameter dan inferensi dalam jaringan Bayesian dengan sampling Monte Carlo.

5.8.6 Contoh Penerapan: Simulasi Portofolio Keuangan

Simulasi Monte Carlo sering digunakan dalam keuangan untuk memodelkan risiko dan nilai masa depan portofolio investasi.

Langkah-langkah simulasi portofolio:

1. *Model Return*: tentukan model *return* dari aset, misalnya, distribusi normal.

2. *Generate Sampel*: *generate return* harian atau bulanan dari distribusi yang dipilih.
3. Kalkulasi Nilai Portofolio: gunakan *return* sampel untuk menghitung nilai portofolio di masa depan.
4. Analisis Hasil: Evaluasi hasil simulasi untuk mengukur risiko (VaR, CVaR) dan performa portofolio.

5.9 Penerapan Statistik dan Probabilitas dalam Kecerdasan Buatan

5.9.1 Pengantar

Penerapan statistik dan probabilitas dalam kecerdasan buatan adalah aspek penting dalam membangun model yang dapat belajar dari data dan membuat keputusan berdasarkan informasi yang tersedia. Statistik menyediakan alat untuk menganalisis data dan menarik kesimpulan, sementara probabilitas memberikan kerangka kerja untuk menangani ketidakpastian dan membuat prediksi. Penerapan statistik dan probabilitas dalam kecerdasan buatan adalah fundamental untuk mengembangkan model yang andal dan akurat. Teknik-teknik yang sudah dibahas sebelumnya membantu dalam memodelkan ketidakpastian, membuat prediksi, dan mengevaluasi kinerja model, sehingga memungkinkan pengambilan keputusan yang lebih baik berdasarkan data.

5.9.2 Pembelajaran Mesin (*Machine Learning*)

Pembelajaran mesin adalah cabang AI yang berfokus pada pengembangan algoritma yang memungkinkan komputer untuk belajar dari dan membuat prediksi atau keputusan berdasarkan data. Statistik dan probabilitas memainkan peran kunci dalam banyak aspek pembelajaran mesin, termasuk:

1. Regresi: menggunakan metode statistik untuk memodelkan hubungan antara variabel.

2. Klasifikasi: menggunakan algoritma probabilistik untuk mengklasifikasikan data ke dalam kategori.
3. *Clustering*: mengelompokkan data ke dalam grup berdasarkan karakteristik yang mirip.

Beberapa metode populer dalam pembelajaran mesin yang menggunakan statistik dan probabilitas meliputi:

1. Regresi Linear dan Logistik: model statistik yang digunakan untuk prediksi dan klasifikasi.
2. *k-Nearest Neighbors* (KNN): algoritma non-parametrik yang menggunakan probabilitas berdasarkan kedekatan.
3. *Support Vector Machines* (SVM): menggunakan konsep statistik untuk klasifikasi dan regresi.
4. *Decision Trees* dan *Random Forests*: menggunakan probabilitas untuk membuat keputusan berdasarkan data.

5.9.3 Model Probabilistik

Model probabilistik digunakan untuk menangani ketidakpastian dalam data dan membuat prediksi yang lebih akurat. Beberapa model probabilistik yang umum digunakan dalam kecerdasan buatan meliputi:

1. Naive Bayes: algoritma klasifikasi yang didasarkan pada Teorema Bayes dengan asumsi independensi antara fitur.

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (5.36)$$

2. *Hidden Markov Models* (HMMs): model probabilistik yang digunakan untuk memodelkan proses yang menghasilkan urutan data, seperti pengenalan ucapan dan teks.

$$P(X,Z) = P(Z_1)\prod_{t=2}^{T} P(Z_t|Z_{t-1})\prod_{t=1}^{T} P(X_t|Z_t) \quad (5.37)$$

3. *Gaussian Mixture Models* (GMMs): model yang mengasumsikan data berasal dari beberapa distribusi Gaussian.

$$P(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X|\mu_k, \Sigma_k) \quad (5.38)$$

dengan μ_k adalah berat/bobot dari komponen ke-k, dan $\mathcal{N}(X|\mu_k, \Sigma_k)$ adalah distribusi Gaussian dengan rerata μ_k dan kovarians Σ_k.

5.9.4 *Deep Learning*

Deep learning adalah subset dari pembelajaran mesin yang berfokus pada jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*). Statistik dan probabilitas digunakan dalam beberapa aspek *deep learning,* termasuk:

1. Regularisasi: teknik statistik untuk menghindari overfitting dengan menambahkan penalti pada kompleksitas model.

$$Loss_{regularized} = Loss + \lambda \cdot Regularization\ Term$$

2. Optimisasi: algoritma probabilistik seperti Stochastic Gradient Descent (SGD) digunakan untuk menemukan parameter model yang optimal.

$$\theta = \theta - \eta \nabla_\theta J(\theta) \quad (5.39)$$

dengan θ adalah parameter model, η adalah *learning rate,* dan $J(\theta)$ adalah fungsi loss.

3. Bayesian Neural Networks: memperkenalkan ketidakpastian ke dalam model jaringan saraf dengan memodelkan bobot sebagai distribusi probabilistik.

$$P(W|D) = \frac{P(D|W)P(W)}{P(D)} \quad (5.40)$$

5.9.5 Inferensi Bayes

Inferensi Bayes adalah metode untuk memperbarui probabilitas hipotesis seiring dengan tersedianya bukti baru. Dalam AI, inferensi Bayes digunakan untuk:

1. Pembelajaran Parameter: memperkirakan distribusi parameter model dari data yang diamati.
2. *Decision Making*: membuat keputusan optimal berdasarkan distribusi probabilitas yang diperbarui.
3. *Bayesian Networks*: struktur grafis yang merepresentasikan hubungan probabilistik antara variabel.

5.9.6 Evaluasi Model

Evaluasi model adalah langkah penting dalam pembelajaran mesin dan kecerdasan buatan untuk memastikan model yang dibangun memiliki performa yang baik. Statistik dan probabilitas digunakan dalam berbagai metrik evaluasi, seperti:

1. Akurasi: proporsi prediksi benar dari total prediksi.

$$Akurasi = \frac{True\ Positives\ +\ True\ Negatives}{Total\ Predictions} \quad (5.41)$$

2. *Precision dan Recall*: mengukur akurasi dari prediksi positif.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (5.42)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (5.43)$$

3. F1 *Score*: *harmonic mean* dari *precision* dan *recall*.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precission + Recall} \quad (5.44)$$

4. *Area Under Curve* (AUC-ROC): Mengukur kemampuan model untuk membedakan antara kelas.

$$AUC = \int_0^1 TPR(FPR^{-1}(x))dx \quad (5.45)$$

dengan TPR adalah *True Positive Rate* dan FPR adalah *False Positive Rate*.

5.9.7 Studi Kasus: Penerapan dalam Analisis Sentimen

Analisis sentimen adalah aplikasi kecerdasan buatan yang menggunakan statistik dan probabilitas untuk menentukan sentimen (positif, negatif, netral) dalam teks. Langkah-langkah umumnya meliputi:

1. *Preprocessing* Data: membersihkan teks dari *noise* seperti *stop words* dan tokenisasi.
2. Ekstraksi Fitur: menggunakan metode statistik seperti *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk representasi teks.

$$TF - IDF(t,d) = tf(t,d) \cdot \log \frac{N}{df(t)} \quad (5.46)$$

dengan $tf(t,d)$ adalah *frekuency term* t dalam dokumen d, N adalah total dokumen, dan $df(t)$ adalah jumlah dokumen yang mengandung term tt.

3. Modeling: menggunakan algoritma seperti Naive Bayes atau *Support Vector Machines* (SVM) untuk klasifikasi sentimen.

4. Evaluasi Model: menggunakan metrik seperti akurasi, *precision*, *recall*, dan F1 *score* untuk mengevaluasi performa model.

Pada bab ini telah dibahas konsep dasar statistik dan probabilitas serta penerapannya dalam kecerdasan buatan (AI). Dari pendahuluan konsep dasar probabilitas hingga metode canggih seperti simulasi Monte Carlo dan Metode Monte Carlo Markov Chain (MCMC), telah dieksplorasi bagaimana alat-alat ini digunakan untuk menangani ketidakpastian dan memodelkan berbagai fenomena kompleks dalam dunia kecerdasan buatan, lebih lanjut lagi memberikan prediksi. Dalam banyak hal, statistik menyediakan fondasi untuk pengambilan keputusan berbasis data, sementara probabilitas memungkinkan untuk mengukur dan memodelkan ketidakpastian dalam prediksi dan analisis data. Teknik-teknik seperti regresi, model probabilistik, dan metode Monte Carlo menunjukkan pentingnya pendekatan statistik dalam merancang dan mengevaluasi model kecerdasan buatan atau pembelajaran mesin yang efektif. Penerapan simulasi Monte Carlo, misalnya, menyoroti bagaimana kita dapat menggunakan sampling acak untuk memahami risiko dan hasil dalam situasi yang penuh ketidakpastian. Hal ini sangat penting dalam berbagai bidang, mulai dari keuangan hingga pengembangan model pembelajaran mesin yang *robust*.

Secara keseluruhan, penggunaan statistik dan probabilitas dalam kecerdasan buatan bukan hanya tentang mengolah angka, tetapi juga tentang memahami dan mengelola kompleksitas dunia nyata. Pendekatan ini memungkinkan para praktisi kecerdasan buatan untuk membangun sistem yang lebih akurat, andal, dan responsif terhadap perubahan data serta ketidakpastian. Bab ini memberikan dasar yang kuat bagi para pembaca untuk memahami peran kritis statistik dan probabilitas dalam kemajuan teknologi kecerdasan buatan, serta mendorong aplikasi lebih lanjut dalam berbagai domain yang memanfaatkan kecerdasan buatan. Lebih lanjut lagi, pembaca disarankan untuk mempelajari terkait topik-topik tertentu pada bab ini untuk dipahami secara lebih mendalam, misalnya terkait dengan algoritma pembelajaran mesin. Beberapa pembahasan secara lebih detail dan menyeluruh tidak dapat disampaikan pada bab ini dikarenakan keterbatasan penulis dan halaman pada bab ini.

Bab 6

Konsep Matematika pada Aljabar Linear

6.1 Aljabar Linear

Aljabar merupakan cabang matematika Aljabar linear merupakan topik yang sangat penting dari matematika Aljabar yang banyak digunakan dalam berbagai dasar ilmu keteknikan, dan juga diperdalam bahkan diperluas lagi dalam berbagai mata kuliah; komputasi numerik, fenomena perpindahan, aliran fluida, perancangan struktur, rekayasa reaksi kimia, pemodelan, program linear, teknik informatika dan masih banyak yang lainnya. Selain itu, Penerapan aljabar linear di dalam ilmu komputer diantaranya; penggunaan aplikasi matlab, *image processing* dalam proses pengenalan wajah, *game* komputer dan kecerdasan buatan (AI). Dalam buku ini akan dibahas secara singkat beberapa topik Aljabar linear diantaranya mengenai; pengenalan matriks, operasi matriks; persamaan linear, sistem linear; determinan, kofaktor, hubungan bilangan skalar, determinan matriks; pengenalan vektor, vektor dalam sistem koordinat, penjumlahan dan pengurangan vektor, perkalian skalar dengan vektor, kombinasi linear, panjang vektor, *dot product*, *cross product*; eigen *value* dan eigen *vector*; transformasi linear.

6.2 Matriks

6.2.1 Pengenalan Matriks

Bentuk matriks pertama kali diperkenalkan oleh ahli matematika dari Inggris bernama James Sylvester tahun 1850 dengan istilah *Oblong Arrangement of Terms*. Banyak para ahli yang menyebut matrik berupa susunan berupa data sains dalam bentuk baris dan kolom membentuk persegi panjang diataranya menurut Anton & Rorres (2014). *Information in science, business, and mathematics is often organized into rows and columns to form rectangular arrays called "matrices"* (plural of "matrix"). Matriks merupakan bilangan-bilangan yang berbentuk *array* persegi panjang. Matriks yang merupakan susunan informasi numerik, sangat diperlukan untuk mengembangkan program komputer dalam memecahkan sistem persamaan, karena komputer sangat cocok untuk memanipulasi susunan informasi numerik. Matriks dalam penerapannya bukan hanya sebatas notasi untuk menyelesaikan sistem persamaan, melainkan dapat dipandang sebagai objek matematika tersendiri. Topik matematika yang terkait dengan hal tersebut yaitu Aljabar Linear. Pada matriks, bilangan dalam *array* disebut *entries*. Gunakan huruf kapital untuk menunjukkan matriks dan huruf kecil untuk menunjukkan *entries*. Lambang *entries* yang terdapat pada baris ditulis dengan i sedangkan *entries* pada kolom dilambangkan j. Ukuran matriks berdasarkan banyaknya jumlah baris m (horizontal) dan jumlah kolom n (vertical). Matriks yang hanya mempunyai satu baris disebut matriks baris (*row matrix* atau *row vektor)*. Sedangkan matriks yang hanya mempunyai satu kolom disebut matriks kolom (*column matrix* atau *column vektor*). Misal, matrik A dengan entries a_{ij} atau bisa ditulis $(A)_{ij}$ berukuran 2 x 3, angka 2 menunjukkan banyaknya jumlah baris, dan angka 3 menunjukkan banyaknya jumlah kolom, sehingga matriks A bisa ditulis.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \tag{6.1}$$

Entries a_{11} berarti menunjukkan baris ke-1 kolom ke-1 pada matriks A, a_{21} berarti menunjukkan baris ke-2 kolom ke-1 pada matriks A. Jika Matriks A mempunyai jumlah baris sama dengan jumlah kolom (m = n) disebut matrik persegi berorde n, dengan diagonal utama matriks A tersebut $a_{11}, a_{22}, \dots, a_{nn}$.

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{6.2}$$

Jika matriks persegi dengan diagonal utamanya mempunyai nilai = 1, maka disebut dengan matriks Identitas (I).

$$I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \tag{6.3}$$

6.2.2 Operasi Matriks

Operasi aritmatika pada matriks dapat berupa, penjumlahan, pengurangan dan perkalian. Berikut beberapa hal yang berkaitan dengan operasi pada matriks.

1. Dua matriks didefinisikan sama, ketika ukurannya sama dan *entries* yang bersesuaian sama.

   Contoh :

   $A = \begin{bmatrix} 1 & 2 \\ 3 & x \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$

   Matriks A berukuran 2 x 2 = Matriks B berukuran 2 x 2

   Sehingga x = 5

   $C = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 4 & 5 \end{bmatrix}$

   Matriks A berukuran 2 x 2 ≠ Matriks C berukuran 2 x 3

2. Jika Matriks A dan Matriks B berukuran sama, maka operasi aritmatika penjumlahan dapat dilakukan dengan menjumlahkan *entries* yang bersesuain pada Matriks A dan Matriks B. Selain itu, operasi aritmatika pengurangan dapat dilakukan dengan mengurangkan *entries* yang bersesuaian pada matriks A dan Matriks B. Tetapi kalau matriks A dan matriks B berukuran beda, operasi aritmatika penjumlahan dan pengurangan pada matriks A dan matriks B tidak bisa dilakukan.

   Contoh :

   $$A = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 2 & 3 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}$$

   $$A + B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 6 & 8 \\ 6 & 8 \end{bmatrix}$$

   $$B - A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ -2 & -2 \end{bmatrix}$$

   A + C , B + C, A – C, B – C undefined dikarenakan matriks C berbeda ukuran dengan matriks A dan matriks B.

3. JIka matriks A dikalikan dengan bilangan skalar c, maka setiap *entries* pada matriks A dikalikan dengan skalar c.

   Contoh :

   $$A = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 2 & 3 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}$$

   $$2A = 2 \times \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 6 \\ 8 & 10 \end{bmatrix}$$

   $$-3B = -3 \times \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} -6 & -9 \\ -12 & -15 \\ -6 & -9 \end{bmatrix}$$

   $$5C = 5 \times \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 20 \\ 15 & 5 \end{bmatrix}$$

4. Jika matriks A berukuran m x n, maka *transpose* matriks A (A^T) berukuran n x m yang dihasilkan dengan menukarkan baris dengan kolom.

   Contoh :

   $$A = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

5. Jika dan hanya jika matriks A merupakan matriks persegi, maka *trace* matrik A dinotasikan tr(A) dan didefinisikan sebagai jumlah *entries* diagonal utama matriks A.

   Contoh :

   $$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 4 & 2 \\ 1 & 3 & 5 \end{bmatrix} \qquad tr(A) = 1 + 4 + 5 = 10$$

6. Jika Matriks A berukuran m x r dan matriks B berukuran r x n, maka hasil kali matriks A dengan matriks B yaitu matriks AB akan berukuran m x n. Ketika mencari *entries* baris i serta kolom j pada matriks AB, pilihlah baris i dari matriks A dan kolom j matriks B, lalu kalikan entries-entries yang bersesuaian, kemudian jumlahkan untuk mendapatkan hasilnya.

   Contoh :

   Jika Matriks A berukuran 2 x 3 dan Matriks B berukuran 3 x 4, maka Matrik AB berukuran 2 x 4

   $$A = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 6 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 4 & 1 & 4 & 3 \\ 0 & -1 & 3 & 1 \\ 2 & 7 & 5 & 2 \end{bmatrix}$$

   $$A \times B = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 6 & 0 \end{bmatrix} \begin{bmatrix} 4 & 1 & 4 & 3 \\ 0 & -1 & 3 & 1 \\ 2 & 7 & 5 & 2 \end{bmatrix}$$

$\sum$(*Entries* baris ke-1 x *Entries* kolom ke-1) = 1x4 + 2x0 + 4x2 = 12

∑(*Entries* baris ke -1 x *Entries* kolom ke-2) = 1x1 + 2x-1+ 4x7 = 27

∑(*Entries* baris ke -1 x *Entries* kolom ke-3) = 1x4 + 2x3 + 4x5 = 30

∑(*Entries* baris ke -1 x *Entries* kolom ke-4) = 1x3 + 2x1 + 4x2 = 13

∑(*Entries* baris ke-2 x *Entries* kolom ke-1) = 2x4 + 6x0 + 0x2 = 8

∑(*Entries* baris ke-2 x *Entries* kolom ke-2) = 2x1 + 6x-1 + 0x7 = -4

∑(*Entries* baris ke-2 x *Entries* kolom ke-3) = 2x4 + 6x3 + 0x5 = 26

∑(*Entries* baris ke-2 x *Entries* kolom ke-4) = 2x3 + 6x1 + 0x2 = 12

$$A \times B = \begin{bmatrix} 12 & 27 & 30 & 13 \\ 8 & -4 & 26 & 12 \end{bmatrix}$$

7. Asumsikan bahwa ukuran matriks sedemikian rupa, sehingga operasi-operasi matrik berikut dapat dilakukan.

a.	A + B = B + A Penjumlahan	Sifat Komutatif
b.	A + (B + C) = (A + B) + C Penjumlahan	Sifat Asosiatif
c.	A (BC) = (AB) C	Sifat Asosiatif Perkalian
d.	A(B ± C) = AB ± AC	Distribusi sebelah kiri
e.	(B ± C)A = BA ± CA	Distribusi sebelah kanan
f.	a(B ± C) = aB ± aC	
g.	(a ± b)C = aC ± bC	
h.	a(bC) = (ab)C	
i.	a(BC) = (aB)C = B(aC)	

6.3 Sistem Persamaan Linear

6.3.1 Pengenalan Persamaan Linear

Biasanya matrik muncul berupa tabel data numerik dari suatu pengamatan atau dalam bentuk konteks matematika. Misalnya, informasi mengenai sistem persamaan berikut yang diwujudkan dalam bentuk matriks.

Sistem Persamaan : Bentuk Matriks :

$x - y = 1$ $\begin{bmatrix} 1 & -1 & 1 \\ 2 & 1 & 6 \end{bmatrix}$

$2x + y = 6$

Solusi sistem persamaan di atas dapat diperoleh dengan melakukan operasi yang sesuai pada matriks tersebut. Sebagai mana diketahui bahwa dalam dua dimensi sebuah garis pada sistem koordinat x-y dapat direpresentasikan oleh sebuah persamaan dalam bentuk:

$$ax + by = c \quad \text{yakni } a, b \neq 0$$

Persamaan dalam bentuk di atas merupakan contoh persamaan linear dengan variable x dan y. Sedangkan untuk tiga dimensi sebuah garis pada sitem koordinat xyz dapat direpresentasikan oleh sebuah persamaan dalam bentuk :

$$ax + by + cz = d \quad \text{yakni } a,b, c \neq 0$$

Persamaan dalam bentuk di atas merupakan contoh persamaan linear dengan variable x, y, dan z. Secara umum bentuk persamaaan linear untuk n variabel x_1, x_2, x_3,… x_n dapat ditulis sebagai berikut:

$$a_1x_1 + a_2x_2 + a_3x_3 + \ldots a_nx_n = b \qquad (6.4)$$

dengan $a_1, a_2, a_3, \ldots, a_n$ dan b konstan, serta $a_1, a_2, a_3, \ldots, a_n \neq 0$

dengan b = 0, persamaan (6.4) menjadi:

$$a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_nx_n = 0 \qquad (6.5)$$

Persamaan (6.5) disebut persamaan linear homogen dengan variabel $x_1, x_2, x_3, \ldots, x_n$

Contoh Persamaan Linear :

a. $3y = 7 - x$

b. $-3x_3 + x_4 = 2x_2 - x_1$

c. $6z + 2 = 2y - x$

Contoh Persamaan Non Linear :

a. $x + 3y^2 - 4 = 0$

b. $\text{Sin } x + y = 0$

c. $3x + 2y = 5 + xy$

6.3.2 Sistem Persamaan Linear

Himpunan persamaan linear berhingga disebut sistem persamaan linear atau sistem linear, dan variabelnya disebut *unknowns* (yang tidak diketahui). Contoh sistem persamaan linear dengan *unknowns* x dan y :

$$5x + y - 3 = 0$$

$$2x - y - 4 = 0$$

Contoh sistem persamaan linear dengan *unknowns* x_1, x_2, dan x_3 :

$$3x_3 + 1 = x_2 - 4x_1$$

$$3x_1 + x_2 + 9x_3 + 4 = 0$$

Secara umum bentuk sistem linear untuk m persamaan dan n variabel yang tidak diketahui (*unknowns*) $x_1, x_2, x_3, \ldots, x_n$ dapat ditulis sebagai berikut :

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \qquad (6.6)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \qquad \vdots \qquad \vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m$$

Koefisien a_{ij} dari variabel yang tidak diketahui (*unknowns*) menunjukkan lokasinya pada sistem persamaan linear tersebut. Misal, a_{12} menunjukkan persamaan ke-1 dan mengalikannya dengan *unknowns* x_2. Solusi sistem linear n variabel yang tidak diketahui (*unknowns*) $x_1, x_2, x_3, \dots, x_n$ yaitu barisan bilangan $s_1, s_2, s_3, \dots, s_n$ hasil substitusi dan dapat ditulis $(s_1, s_2, s_3, \dots, s_n)$

$$x_1 = s_1 \qquad x_2 = s_2 \qquad x_3 = s_3 \qquad x_n = s_n$$

Sistem linear dua variabel yang tidak diketahui (*unknowns*) berhubungan dengan perpotongan garis. Misal, untuk sistem linear berikut:

$$a_1x + b_1y = c_1 \qquad (6.7)$$
$$a_2x + b_2y = c_2$$

dengan grafik persamaannya adalah garis pada bidang xy. Setiap solusi (x,y) dari sistem linear dengan dua variabel yang tidak diketahui (*unknowns*) merupakan titik potong garis persamaan tersebut. Ada tiga kemungkinan yang akan terjadi, yaitu :

1. Garis-garis yang berbeda sejajar, tidak ada perpotongan garis, akibatnya tidak ada solusi.

   Contoh :

   $x + y = 4 \qquad 3x + 3y = 6$

2. Garis-garis berpotongan hanya satu titik, akibatnya mempunyai tepat satu solusi.

   Contoh :

   $x - y = 1 \qquad 2x + y = 6$

3. Garis-garis yang berimpitan (terdapat banyak sekali titik berpotongan atau titik-titik pada garis yang sama), akibatnya mempunyai banyak solusi.

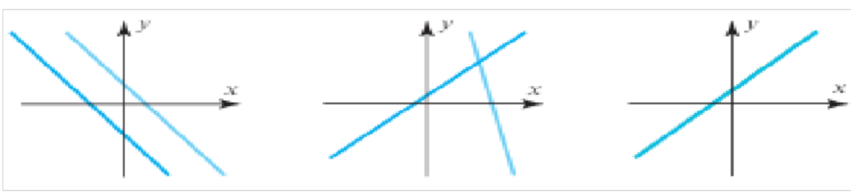
   Contoh :

   $4x - 2y = 1 \qquad 16x - 8y = 4$

Secara umum sistem linear dikatakan konsisten, jika sistem tersebut mempunyai paling sedikit satu solusi, dan tidak konsisten jika sistem tersebut tidak mempunyai solusi. Sistem linear konsisten untuk sistem linear yang terdiri dari dua persamaan dengan dua variabel yang tidak dikatahui (*unknowns*). Gambar 6.1 akan memperjelas hal tersebut. Hal yang sama berlaku juga untuk tiga persamaan dengan tiga variabel yang tidak diketahui (*unknowns*).

$$
\begin{aligned}
a_1x + b_1y + c_1z &= d_1 \\
a_2x + b_2y + c_2z &= d_2 \\
a_3x + b_3y + c_3z &= d_3
\end{aligned}
\tag{6.8}
$$

Ketika grafik persamaan berbentuk bidang pada koordinat xyz. Setiap solusi (x,y,z) dari sistem linear dengan tiga variabel yang tidak diketahui (*unknowns*) merupakan titik-titik yang bersesuaian dengan ketiga bidang yang berpotongan. Ada tiga kemungkinan, yaitu tidak ada solusi, tepat satu solusi dan banyak solusi. Gambar 6.2 akan memperjelas hal tersebut.



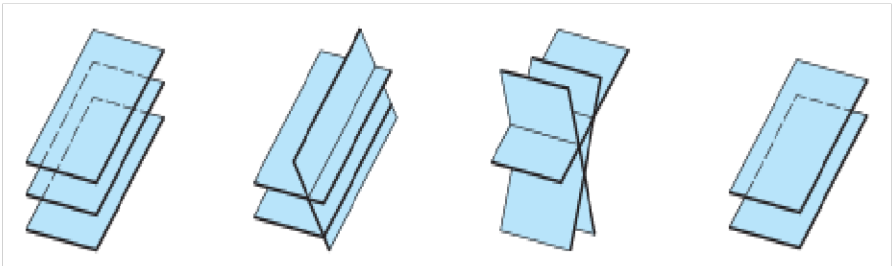
Tidak ada solusi (sistem linear tidak konsisten)

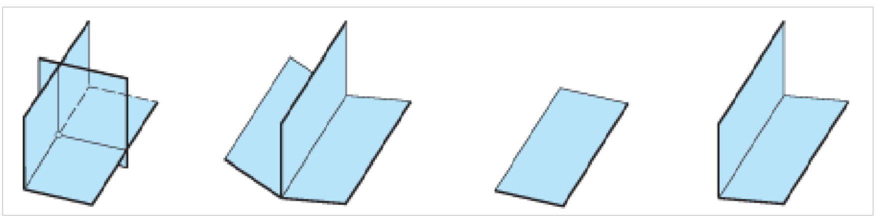
Tepat satu solusi (sistem linear konsisten)

Banyak solusi (sistem linear konsisten)

Gambar 6.1 Sistem Linear dengan Dua Persamaan Linear

Sumber Anton & Rorres, 11th Edition





Gambar 6.2 Sistem Linear dengan Tiga Persamaan Linear

Sumber Anton & Rorres, 11th Edition

Operasi Aritmatika perkalian matriks memiliki peranan penting dalam penerapan mencari solusi sistem linear. Pada saat jumlah persamaan linear dan jumlah variabel yang tidak diketahui (*unknowns*) makin banyak maka mencari solusi sistem linear menjadi lebih kompleks. Supaya perhitungan mencari solusi lebih mudah, maka perlu menyederhanakan notasi dan menstandarkan prosedurnya. Di sini digunakan matriks untuk mencari solusi sistem linear. Istilah matriks biasanya digunakan dalam matematika untuk menunjukkan susunan angka yang berbentuk persegi panjang. Secara umum bentuk sistem linear untuk m persamaan linear dan n variabel yang tidak diketahui (*unknowns*) $x_1, x_2, x_3, \dots, x_n$ sebagai berikut:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + \dots + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + \dots + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & a_{m3}x_3 & + \dots + & a_{mn}x_n & = & b_m \end{array}$$

dapat ditulis dalam bentuk ***augmented matrix*** menjadi :

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \cdots a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} \cdots a_{2n} & b_2 \\ \vdots & \vdots & \vdots \quad \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} \cdots a_{mn} & b_m \end{bmatrix} \tag{6.9}$$

Selain itu, dengan menggunakan definisi bahwa dua matrik dikatakan sama, jika dan hanya jika ukurannya sama dan *entries-entries* yang bersesuaian sama. Sehingga sistem linear untuk m persamaan liniear dan n variabel yang tidak diketahui (unknowns) dapat ditulis menjadi persamaan matriks kolom (m x 1).

$$\begin{bmatrix} a_{11}x_1 & a_{12}x_2 & a_{13}x_3 \cdots a_{1n}x_n \\ a_{21}x_1 & a_{22}x_2 & a_{23}x_3 \cdots a_{2n}x_n \\ \vdots & \vdots & \vdots \quad\quad \vdots \\ a_{m1}x_1 & a_{m2}x_2 & a_{m3}x_3 \cdots a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{6.10}$$

Matrik berukuran (size) m x n Matrik berukuran

(size) m x 1

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{6.11}$$

Matriks berukuran m x n n x 1 m x 1

Sehingga bisa ditulis A x = b, dengan matriks A sebagai koefisien matriks pada sistem linear tersebut. *Augmented matrix* dapat ditulis dengan menggabungkan *entries* matriks A dengan b, dan disimpan di kolom terakhir, menjadi seperti di bawah ini.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \cdots a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} \cdots a_{2n} & b_2 \\ \vdots & \vdots & \vdots \quad\quad \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} \cdots a_{mn} & b_m \end{bmatrix} \tag{6.12}$$

Metode dasar untuk menyelesaikan sistem linear yaitu dengan melakukan operasi aljabar dengan tidak mengubah himpunan solusi untuk menghasilkan sistem yang makin sederhana dan konsisten. Sistem linear dalam bentuk *augmented matrix* di atas dapat diselesaikan dengan menggunakan operasi baris dasar (*elementary row operations*) sebagai berikut :

- Kalikan baris dengan konstanta tak nol
- Tukarkan dua baris
- Tambahkan baris yang telah dikalikan dengan konstanta dengan baris lainnya.

Berikut contoh menyelesaikan sistem linear dengan tiga persamaan dan tiga variabel yang tidak diketahui menggunakan bentuk *augmented matrix* dan operasi baris dasar (*elementary row operations*).

Sistem Linear	Operasi Baris Dasar	*Augmented Matrix*
$x + y + 2z = 9$ $2x + 4y - 3z = 1$ $3x + 6y - 5z = 0$	-2 x baris ke-1 + baris ke-2	$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 2 & 4 & -3 & 1 \\ 3 & 6 & -5 & 0 \end{bmatrix}$
$x + y + 2z = 9$ $2y - 7z = -17$ $3x + 6y - 5z = 0$	-3 x baris ke-1 + baris ke-3	$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 2 & -7 & -17 \\ 3 & 6 & -5 & 0 \end{bmatrix}$
$x + y + 2z = 9$ $2y - 7z = -17$ $3y - 11z = -27$	1/2 x baris ke-2	$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 2 & -7 & -17 \\ 0 & 3 & -11 & -27 \end{bmatrix}$
$x + y + 2z = 9$ $y - 7/2z = -17/2$ $3y - 11z = -27$	-3 x baris ke-2 + baris ke-3	$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -7/2 & -17/2 \\ 0 & 3 & -11 & -27 \end{bmatrix}$
$x + y + 2z = 9$ $y - 7/2z = -17/2$ $- 1/2z = -3/2$	-2 x baris ke-3	$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -7/2 & -17/2 \\ 0 & 0 & -1/2 & -3/2 \end{bmatrix}$
$x + y + 2z = 9$ $y- 7/2z = -17/2$ $z = 3$	-1 x baris ke-2 + baris ke-1	$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -7/2 & -17/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$

x + 11/2z = 35/2 y - 7/2z = -17/2 z = 3	-11/2 x baris ke-3 + baris ke-1	$\begin{bmatrix} 1 & 0 & 11/2 & 35/2 \\ 0 & 1 & -7/2 & -17/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$
x = 1 y = 2 z = 3	7/2 x baris ke-3 + baris ke-2	$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$

Jadi, solusinya x = 1, y = 2, dan z = 3 untuk sistem linear

$$x + y + 2z = 9$$

$$2x + 4y - 3z = 1$$

$$3x + 6y - 5z = 0$$

6.4 Determinan

6.4.1 Menghitung Determinan

Sistem linear tertentu dapat dicari penyelesaiannya yaitu menggunakan konsep determinan, yaitu persamaan linear tersebut dibuat bentuk matriks terlebih dahulu. Misal, jika matrik A dibentuk dari dua persamaan linear dengan ukuran 2 x 2 seperti di bawah ini :

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

determinan matriks A dapat ditulis

$$\det(A) = \det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \qquad (6.13)$$

Matriks A tersebut mempunyai invers (A^{-1}) jika dan hanya jika det (A) = $a_{11}a_{22} - a_{12}a_{21} \neq 0$, sehingga invers dari matriks A (A^{-1}) dapat dicari menggunakan rumus berikut:

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad (6.14)$$

Rumus tersebut bisa diuji kevalidannya, ketika matriks A dikalikan dengan inversi matriks A (A^{-1}) akan menghasilkan matriks Identitas (I).

$$A.A^{-1} = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.15}$$

Jika matriks A dan matriks B mempunyai inversi, maka inversi dari matriks AB dapat ditulis:

$$(AB)^{-1} = B^{-1}A^{-1} \tag{6.16}$$

Jika matriks A dan matriks B merupakan matriks persegi berukuran sama, maka berlaku:

$$\det(AB) = \det(A)\det(B) \tag{6.17}$$

6.4.2 Kofaktor

Jika matriks A merupakan matriks persegi, maka minor *entries* a_{ij} dinotasikan M_{ij}, dan didefinisikan sebagai determinan sub matriks tersisa, setelah baris ke-i dan kolom ke-j dihapus dari matriks A. Nilai $(-1)^{i+j}M_{ij}$ ditulis sebagai C_{ij} dan disebut Ko~~p~~faktor dengan *entries* a_{ij}.

Contoh :

$$A = \begin{bmatrix} 3 & 1 & -4 \\ 2 & 5 & 6 \\ 1 & 4 & 8 \end{bmatrix}$$

Minor *entries* a_{11} ditulis M_{11}, baris ke-1 dan kolom ke-1 pada matriks A dicoret, sehingga determinan sub matriksnya menjadi :

$$M_{11} = \begin{vmatrix} \cancel{3} & \cancel{1} & \cancel{-4} \\ \cancel{2} & 5 & 6 \\ \cancel{1} & 4 & 8 \end{vmatrix} = \begin{vmatrix} 5 & 6 \\ 4 & 8 \end{vmatrix} = 40 - 24 = 16$$

Kofaktor dengan *entries* a_{11} dapat ditulis C_{11} yaitu :

$$C_{11} = (-1)^{1+1} M_{11} = (-1)^2 . 16 = 1 \;.\; 16 = 16$$

6.4.3 Hubungan Bilangan Skalar dan Determinan Matriks

Beberapa kemungkinan hubungan det(A) dengan det(B) yang akan terjadi untuk matriks A, matriks B dan matriks C yang berukuran n x n dengan k bilangan sembarang skalar, antara lain:

1. $\det(kA) = k^n \det(A)$
2. $\det(A + B) \neq \det(A) + \det(B)$
3. $\det(C) = \det(A) + \det(B)$
4. $\det(AB) = \det(A)\det(B)$
5. $\det(A^{-1}) = \frac{1}{\det(A)}$
6. Transpose dari matrik kopaktor disebut adjoint dari matrik A dan ditulis adj(A).
7. Hubungan invers matriks A dengan adjoint matrik A

$$A^{-1} = \frac{1}{\det(A)}\, adj(A)$$

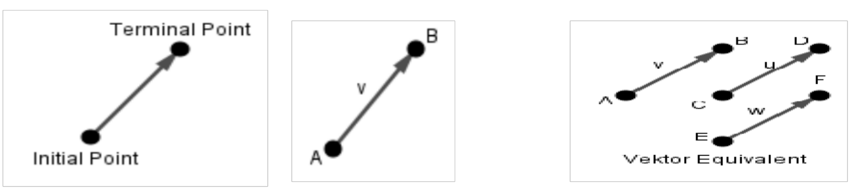
6.5 Ruang Vektor

6.5.1 Pengenalan Vektor

Besaran menurut para ahli fisika dan insinyur dibedakan menjadi dua, antara lain besaran skalar dan besaran vektor. Besaran Skalar yaitu besaran yang hanya memiliki nilai numerik saja, sedangkan besaran vektor yaitu besaran yang memiliki nilai dan deskripsi arah secara lengkap. Contoh yang termasuk besaran skalar: suhu, panjang, waktu, kecepatan. Contoh besaran vektor kelajuan, gaya, perpindahan, tekanan, medan magnet. Walaupun besaran skalar dan vektor berasal dari keteknikan dan fisika, namun demikian dalam buku ini dibahas supaya dapat membangun struktur matematika yang bisa diterapkan pada bidang komputer, pemograman dan *Artificial Intelligence* (AI) atau kecerdasan buatan. Pokok materi utama pada aljabar linear yaitu matriks dan vektor. Matrik sudah dijelaskan pada sub bab sebelumnya. Pada sub bab selanjutnya akan diulas mengenai sifat-sifat dasar vektor pada dua dimensi (2-*space*), tiga dimensi (3-*space*) dan R^n dimensi. Hal yang perlu diperhatikan pada geometri vektor antara lain; arah mata panah menentukan arah vektor; panjang panah menentukan nilai atau besarnya vektor; ekor anak panah disebut titik awal (*initial point*); ujung anak panah disebut titik akhir (*terminal point*). Pada buku ini vektor simbol vektor ditulis dengan huruf kecil tebal misal, **a**, **b**, **c**, **v**, **w**, **x**. Sedangkan skalar ditulis dengan huruf kecil miring misal, *a*, *b*, *c*, *v*, *w*, *x* . Pada saat kita akan menunjukkan bahwa pada suatu vektor **v** mempunyai titik awal A dan titik akhir B maka dapat dinotasikan:

$$\boldsymbol{v} = \overrightarrow{AB} \tag{6.18}$$

Vektor dengan panjang dan arah yang sama disebut dengan vektor yang *equivalent*. Misal, vektor **v** = **u** = **w** seperti pada Gambar 6.3 berikut yang menunjukkan mengenai geometri vektor dengan dibuat menggunakan Aplikasi Geogebra.

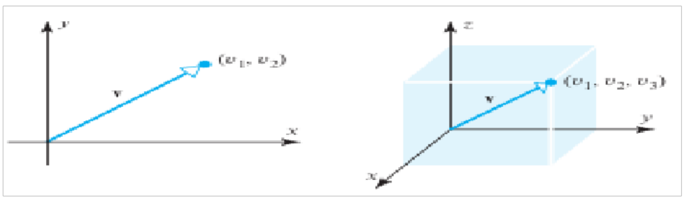


Gambar 6.3 Geometri Vektor

Sumber Penulis

6.5.2 Vektor dalam Sistem Koordinat

Perhitungan vektor berdimensi dua, tiga dan berdimensi n (R^n) akan lebih mudah dipahami dan dicari solusinya dengan sistem koordinat. Misal, jika vektor **v** yang *initial point*nya pada titik *origin* (0,0) berada pada dimensi dua $\mathbf{v} = (v_1, v_2)$ dengan komponen v_1, v_2 dan dimensi tiga $\mathbf{v} = (v_1, v_2, v_3)$ dengan komponen v_1, v_2, v_3 dapat dilihat dalam sistem koordinat berikut pada Gambar 6.4.



Gambar 6.4 Vektor dalam Sistem Koordinat

Sumber Anton & Rorres, 11th Edition

Dapat ditulis bentuk vektor yang dibatasi koma pada dimensi n R^n dengan notasi:

$$\mathbf{v} = (v_1, v_2, \ldots, v_n) \tag{6.19}$$

Ketika bentuk vektor tersebut tidak dibatasi koma, maka dapat ditulis

$$\text{vektor baris} \quad \mathbf{v} = [v_1 \quad v_2 \quad \cdots \quad v_n] \tag{6.20}$$

$$\text{vektor kolom} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{6.21}$$

Jika suatu vektor titik awal (*initial point*) tidak pada titik *origin*, maka dapat ditulis $\overrightarrow{P_1P_2}$, yaitu untuk vektor dimensi dua (2- *space*) titik awal $P_1(x_1, y_1)$ dan titik akhir $P_2(x_2, y_2)$. Sedangkan jika vektor tersebut berdimensi tiga (3-*space*) titik awal $P_1(x_1,y_1,z_1)$ dan titik akhir $P_2(x_2,y_2,z_2)$. Sehingga komponen $\overrightarrow{P_1P_2}$ dapat ditulis sebagai berikut :

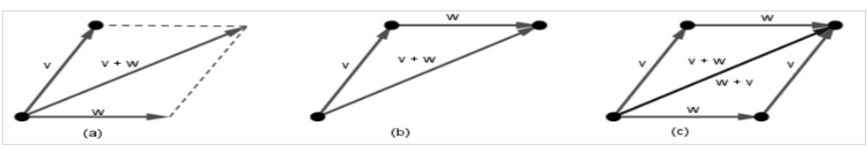
$\overrightarrow{P_1P_2} = (x_2 - x_1, y_2 - y_1)$ komponen vektor 2-*space*

$\overrightarrow{P_1P_2} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$ komponen vektor 3-*space*

6.5.3 Penjumlahan dan Pengurangan Vektor

Operasi penjumlahan dua buah vektor berdimensi 2 dan berdimensi 3 mengikuti aturan jajaran genjang (Gambar 6.5a) dan aturan segitiga (Gambar 6.5b). Sifat komutatif berlaku pada penjumlahan dua buah vektor, hal tersebut dapat direkonstruksi dengan aturan jajaran genjang dan aturan segitiga (Gambar 6.5c).

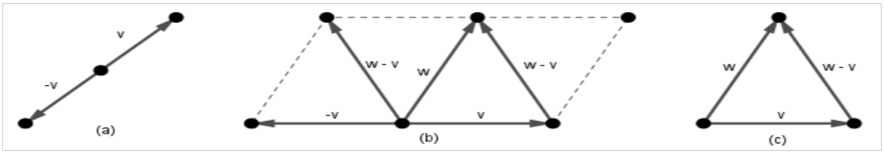
$$\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v} \tag{6.22}$$



Gambar 6.5 Penjumlahan Vektor. **Sumber Penulis**

Pada aritmatika, operasi pengurangan dalam bentuk operasi penjumlah dapat ditulis a – b = a + (-b), hal tersebut berlaku juga pada operasi pengurangan pada vektor. Misal, suatu vektor **v** merupakan vektor negatif dan ditulis **–v**, dimana panjangnya sama dengan vektor **v** namun arahnya yang berlawanan (Gambar 6.6a), operasi pengurangannya direkonstruksi dengan aturan jajaran genjang (Gambar 6.6b atau 6.6c) dan dapat ditulis

$$\mathbf{w} - \mathbf{v} = \mathbf{w} + (-\mathbf{v}) \tag{6.23}$$



Gambar 6.6 Pengurangan Vektor

Sumber Penulis

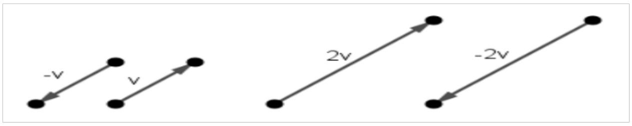
Jika **v** ($v_1, v_2, \ldots, v_n$) dan **w** ($w_1, w_2, \ldots, w_n$) merupakan vektor pada dimensi n (R^n), maka dapat didefinisikan penjumlahan dan pengurangan vektor sebagai berikut :

$$\mathbf{w} + \mathbf{v} = \mathbf{v} + \mathbf{w} = (v_1+w_1, v_2+w_2, \ldots, v_n+w_n) \tag{6.24}$$

$$\mathbf{w} - \mathbf{v} = \mathbf{w} + (-\mathbf{v}) = (w_1-v_1, w_2-v_2, \ldots, w_n-v_n) \tag{6.25}$$

6.5.4 Perkalian Skalar dengan Vektor

Pada waktu akan mengubah panjang vektor dan arah, maka harus dikalikan bilangan skalar. Jika **v** merupakan vektor tak nol di ruang dimensi dua (R^2) dan dimensi tiga (R^3), sedangkan k bilangan skalar tak nol, maka dapat didefinisikan hasil kali skalar (*skalar product*) k dengan **v** menjadi vektor yang panjangnya |k| dikali v dengan arah sama dengan vektor **v** jika k positif, dan berlawanan arah dengan vektor **v** jika k negatif. Selain itu, jika k = 0 atau **v** = 0, maka k**v** = 0. Lebih jelasnya dapat dilihat pada Gambar 6.7 berikut



Gambar 6.7 Perkalian Skalar dengan Vektor

Sumber Penulis

Jika **v** (v_1, v_2, … , v_n), merupakan vektor pada dimensi n (R^n) dan k bilangan skalar, maka kita dapat mendefinisikan k**v** = (kv_1, kv_2, … , kv_n)

6.5.5 Kombinasi Linear

Operasi aritmatika pada vektor penjumlahan, pengurangan dan perkalian skalar sering digunakan secara bersamaan (*combination*) membentuk vektor baru. Jika k_1, k_2, … , k_n bilangan skalar yang disebut koefisien dan **w** vektor kombinasi linear dengan dimensi R^n dari vektors, $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \ldots, \mathbf{v_n}$ dalam dimensi R^n. Maka dapat ditulis bentuk kombinasi linear sebagai berikut:

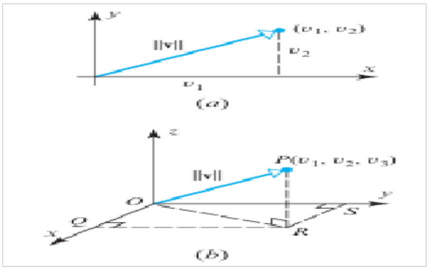
$$\mathbf{w} = k_1\mathbf{v_1} + k_2\mathbf{v_2} + \ldots + k_n\mathbf{v_n} \qquad (6.26)$$

6.5.6 Panjang Vektor, Dot Product dan Cross Product

Panjang vektor *(norm)* **v** ditulis $\|\boldsymbol{v}\|$. Vektor **v** (v_1,v_2) di R^2 dan **v** (v_1,v_2,v_3) di R^3 dengan initial point di titik origin (0,0) panjang vektor $\|\boldsymbol{v}\|$ dapat ditentukan dengan menggunakan teorema Pythagoras seperti pada Gambar 6.8a (R^2) dan Gambar 6.8b (R^3).

Sehingga dapat ditulis $\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2}$ dan $\|v\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$. Secara umum, jika vektor v (v_1,v_2, … , v_n) dalam dimensi R^n maka panjang vektor (norm) $\|\boldsymbol{v}\|$ didefinisikan dengan rumus berikut:

$$\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} \tag{6.27}$$



Gambar 6.8 Panjang Vektor dengan Pytagoras

Sumber Anton & Rorres, 11th Edition

Jika vektor **u** dan **v** bukan vektor nol, berada dalam dimensi R^2 dan R^3, serta θ merupakan sudut diantara vektor **u** dengan vektor **v**, maka hasil kali titik (*Dot Product/Euclidean Inner Product*) vektor **u** dengan vektor **v** yang ditulis **u**.**v** dan didefinisikan:

$$\mathbf{u}.\mathbf{v} = \|\boldsymbol{u}\|\ \|\boldsymbol{v}\| \cos\theta \tag{6.28}$$

Sehingga secara umum jika vektor **u**(u_1, u_2, … , u_n) dan vektor **v**(v_1, v_2, … , v_n) pada dimensi R^n. Hasil kali titik **u**.**v** didefinisikan

$$\mathbf{u}.\mathbf{v} = u_1v_1 + u_2v_2 + \ldots + u_nv_n \tag{6.29}$$

Hasil kali titik (*Dot product*) dua buah vektor akan menghasilkan skalar sedangkan hasil kali silang (*cross product*) menghasilkan vektor. Pada buku ini untuk *cross product* penulis batasi sampai ruang vektor dimensi 3 (R^3). Jika vektor **u**(u_1, u_2, u_3) dan vektor **v**(v_1, v_2, v_3) maka hasil kali silang vektor **u** dengan vektor **v** ditulis **u** x **v** dan didefinisikan.

$$\mathbf{u} \times \mathbf{v} = (u_2v_3 - u_3v_2,\ u_3v_1 - u_1v_3,\ u_1v_2 - u_2v_1) \qquad (6.30)$$

dalam bentuk determinan

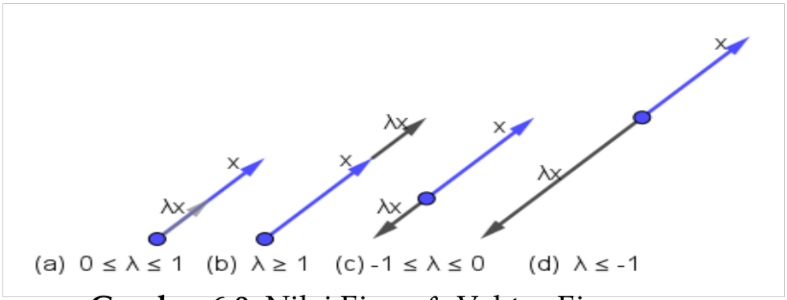
$$\mathbf{u} \times \mathbf{v} = \left(\begin{vmatrix} u_2 & u_3 \\ v_2 & v_3 \end{vmatrix}, -\begin{vmatrix} u_1 & u_3 \\ v_1 & v_3 \end{vmatrix}, \begin{vmatrix} u_1 & u_2 \\ v_1 & v_2 \end{vmatrix} \right) \qquad (6.31)$$

6.6 Nilai Eigen (*Eigenvalue*) dan Vektor Eigen (*Eigenvactor*)

Istilah Eigen diperkenalkan sekitar tahun 1900 di Jerman yaitu ketika mempelajari gerak rotasi, mengklasifikasikan berbagai jenis permukaan dan menjelaskan solusi persamaan differensial tertentu. Penerapan mengenai eigen ini pada matriks dan transformasi matrik saat bermanfaat untuk beberapa bidang ilmu pengetahuan, diantaranya: komputer grafis dan mekanika kuantum. Kata eigen berarti: khas, milik, kharakteristik, individu. Selanjutnya, Misal A merupakan matriks berukuran n x n, kemudian **x** merupakan vektor tak nol pada dimensi R^n dapat disebut vektor eigen (*eigenvektor*) dari matrik A (atau operator matrik T_A), jika A**x** adalah perkalian skalar dengan vektor **x.**

$$A\mathbf{x} = \lambda\mathbf{x} \qquad (6.32)$$

yaitu untuk skalar λ disebut nilai eigen (*eigenvalue*) dari matrik A (T_A), sehingga **x** merupakan vektor eigen yang sesuai dengan nilai eigen (λ). Besaran dan arah bergantung pada tanda dan nilai eigen λ yang bersesuaian dengan vektor eigen (**x**), pembalikan arah terjadi jika nilai eigen λ negatif.



Gambar 6.9 Nilai Eigen & Vektor Eigen

Sumber Penulis

Contoh: Diketahui matrik A dengan vektor eigen **x** sebagai berikut

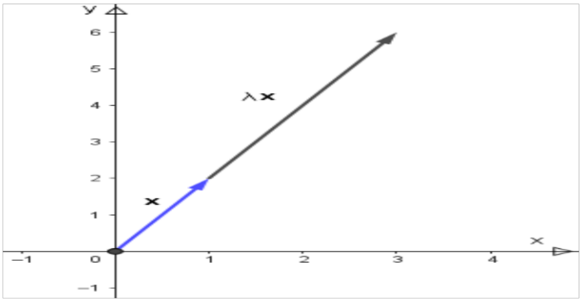
$$A = \begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Perkalian matriks A dengan vektor eigen **x** menghasilkan nilai eigen λ yang dikalikan dengan vektor eigen **x**.

$$A\mathbf{x} = \lambda\mathbf{x}$$

$$\begin{bmatrix} 3 & 0 \\ 8 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix} = 3\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Jadi nilai eigennya $\lambda = 3$



Gambar 6.10 Grafik Nilai Eigen & Vektor Eigen

Sumber Penulis

6.7 Transformasi Linear

Transformasi matriks merupakan fungsi-fungsi khusus yang berasal dari perkalian matriks yang mendasar untuk mempelajari transformasi linear. Penerapan transformasi matriks sangat berguna pada bidang teknik, fisika ataupun sosial.

Sebagaimana diketahui bahwa fungsi merupakan suatu aturan yang mengasosiasikan setiap elemen pada himpunan A (*domain*) satu dan hanya satu-satunya elemen himpunan B (*codomain*). Jika f merupakan sebuah fungsi dengan *domain* R^n dan *codomain* R^m, lalu disebut fungsi f adalah transformasi R^n menjadi R^m atau fungsi tersebut memetakan R^n ke R^m, hal tersebut dapat ditulis:

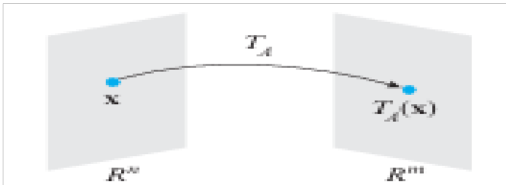
$$f : R^n \rightarrow R^m \tag{6.33}$$

Berdasarkan konsep fungsi di atas, dapat ditulis transformasi matriks (atau matriks operator jika m = n) sebagai berikut:

$$T_A : R^n \rightarrow R^m \tag{6.34}$$

T_A menunjukkan terjadinya transformasi hasil dari perkalian vektor di R^n menjadi R^m dengan matriks A yang berukuran m x n menjadi pemetaan yang berbentuk :

$$T_A(\mathbf{x}) = A\mathbf{x} \tag{6.35}$$



Gambar 6.11 Transformasi Matrik A di Rn ke Rm

Sumber Anton & Rorres, 11th Edition

Transformasi nol T_0 terjadi hasil dari perkalian vektor di Rn menjadi R^m dengan matrik 0 yang berukuran m x n sebagai berikut

$$T_0(\mathbf{x}) = 0\mathbf{x} = \mathbf{0} \qquad (6.36)$$

Transformasi identitas T_I atau disebut juga operator identitas, terjadi hasil dari perkalian vektor di R^n menjadi R^m (m = n), dengan matrik Identitas I yang berukuran n x n sebagai berikut:

$$T_I(\mathbf{x}) = I\mathbf{x} = \mathbf{x} \qquad (6.37)$$

Transformasi matriks $T : R^n \rightarrow R^m$ jika dan hanya jika untuk semua vektor **u** dan **v** serta skalar k, berlaku hubungan sebagai berikut :

1. $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$ Sifat Aditif
2. $T(k\mathbf{u}) = k\,T(\mathbf{u})$ Sifat Homogenitas

Sifat aditif dan sifat homogenitas tersebut disebut kondisi linear, transformasi yang memenuhi kondisi tersebut disebut transformasi linear. Setiap transformasi matriks dari R^n ke R^m merupakan transformasi linear begitu juga sebaliknya, setiap transformasi linear dari R^n ke R^m merupakan transformasi matriks. Sifat aditif dan sifat homogenitas transformasi matriks merupakan titik awal ketika kita akan mendefinisikan trasformasi linear umum. Misal, Transformasi linear $T : V \rightarrow W$ dapat digunakan dalam kombinasi untuk menunjukkan bahwa jika $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ adalah vektor-vektor di V dan $k_1, k_2, \ldots, k_r$ beberapa skalar sehingga dapat ditulis:

$$T(k_1\mathbf{v}_1 + k_2\mathbf{v}_2 + \ldots + k_r\mathbf{v}_r) = k_1T(\mathbf{v}_1) + k_2T(\mathbf{v}_2) + \ldots + k_rT(\mathbf{v}_r) \quad (6.38)$$

Bab 7

Memahami Kompleksitas Komputasi: Konsep, Model Deterministik dan Non-Deterministik, Serta Aplikasi Praktis

7.1 Kompleksitas Kompuasi

Kompleksitas komputasi adalah salah satu bidang yang paling penting dalam ilmu komputer, yang berfokus pada mempelajari seberapa efisien suatu algoritma dapat memecahkan masalah tertentu. Kompleksitas ini dapat diukur dari berbagai aspek, seperti waktu yang dibutuhkan (kompleksitas waktu) dan ruang memori yang diperlukan (kompleksitas ruang).

Kompleksitas waktu mengukur jumlah langkah yang diperlukan oleh sebuah algoritma untuk menyelesaikan masalah sebagai fungsi dari ukuran inputnya. Misalnya, jika dimiliki algoritma dengan kompleksitas waktu $O(n)$, ini berarti bahwa waktu yang dibutuhkan oleh algoritma tersebut berbanding lurus dengan ukuran input n. Contoh lain adalah algoritma dengan kompleksitas waktu $O(n^2)$, yang menunjukkan bahwa waktu yang diperlukan meningkat secara kuadrat seiring dengan peningkatan ukuran input.

Kompleksitas ruang mengukur jumlah memori yang diperlukan oleh sebuah algoritma selama eksekusinya sebagai fungsi dari ukuran input. Misalnya, algoritma dengan kompleksitas ruang $O(1)$ hanya memerlukan jumlah memori yang tetap, terlepas dari ukuran inputnya. Sebaliknya, algoritma dengan kompleksitas ruang $O(n)$ memerlukan memori yang berbanding lurus dengan ukuran input n.

Pemahaman yang mendalam tentang kompleksitas komputasi sangat penting untuk merancang algoritma yang efisien dan optimal. Algoritma yang efisien tidak hanya mempercepat proses komputasi, tetapi juga mengurangi penggunaan sumber daya, yang sangat penting dalam lingkungan dengan sumber daya terbatas. Misalnya, dalam aplikasi real-time atau sistem dengan daya komputasi rendah, algoritma yang efisien dapat membuat perbedaan antara kegagalan dan keberhasilan.

Selain itu, pemahaman tentang kompleksitas komputasi membantu dalam memilih algoritma yang tepat untuk masalah tertentu. Dalam beberapa kasus, algoritma sederhana dengan kompleksitas waktu yang lebih tinggi mungkin lebih cocok dibandingkan algoritma yang lebih kompleks tetapi lebih cepat, tergantung pada konteks dan kebutuhan spesifik.

Sebagai contoh penerapan, pertimbangkan masalah pencarian elemen dalam sebuah array. Algoritma pencarian linier memiliki kompleksitas waktu $O(n)$, yang berarti waktu yang dibutuhkan untuk mencari elemen dalam *array* berukuran n adalah sebanding dengan n. Di sisi lain, algoritma pencarian biner memiliki kompleksitas waktu $O(\log n)$, yang jauh lebih efisien untuk *array* yang sudah terurut.

Dalam dunia nyata, berbagai masalah komputasi seperti penjadwalan, optimasi jaringan, analisis data besar, dan pengenalan pola sangat bergantung pada algoritma yang efisien. Oleh karena itu, studi tentang kompleksitas komputasi bukan hanya bidang teoretis, tetapi juga memiliki aplikasi praktis yang luas. Dengan demikian, memahami dan mengukur kompleksitas komputasi adalah langkah fundamental dalam pengembangan teknologi informasi dan ilmu komputer, membantu kita untuk terus meningkatkan efisiensi dan kinerja sistem komputasi.

7.2 Mesin *Turing* Deterministik (DTM)

Komponen Utama:

1. Unit Kontrol: Menyimpan sejumlah keadaan terbatas.
2. Unit Memori: Pita tak terbatas dibagi menjadi sel-sel yang menyimpan simbol.
3. Kepala Baca/Tulis: Membaca, menulis, dan bergerak di sepanjang pita.

Langkah Operasi DTM:

1. Membaca simbol di sel yang dipindai.
2. Menulis simbol baru di sel tersebut.
3. Memindahkan kepala pita (kiri atau kanan).
4. Mengubah keadaan kontrol berdasarkan keadaan dan simbol yang dibaca.

Definisi Formal:

Q : Set keadaan.

$q0$: Keadaan awal.

F : Set keadaan penerimaan.

Σ : Set simbol input.

Γ : Set simbol pita (termasuk simbol kosong B).

Δ : Fungsi transisi, δ: $(Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$.

Konfigurasi:

Konfigurasi α dari *MT M* adalah elemen (q, x, y) dari $Q \times \Gamma * \times \Gamma *$, dengan q adalah keadaan saat ini, x dan y adalah simbol di pita, dan kepala pita memindai simbol pertama dari y.

Fungsi Transisi:

1. Kasus 1: $\delta(q1, s1) = (q2, s2, L)$

   Jika $x1 = \lambda$, maka $s3 = B$; jika tidak, $x1 = x2s3$.

   $(q1, x1, y1) \vdash (q2, \ell(x2), r(s3s2y2))$.

2. Kasus 2: $\delta(q1, s1) = (q2, s2, R)$

   $(q1, x1, y1) \vdash (q2, \ell(x1s2), r(y2))$.

3. Kasus 3: δ(q1, s1) tidak terdefinisi.

Penerimaan String:

MT M menerima string w jika urutan konfigurasi dari $\alpha0$ ke αn dengan $\alpha0 = (q0, \lambda, w)$ dan keadaan akhir $\alpha n = (q, x,$ y) dengan $q \in F$.

Contoh:

Mesin M menerima string dalam $L = \{a\text{^}i\ b\text{^}j : 0 \leq i \leq j\}$.

Keadaan: $Q = \{q0, q1, q2, q3, q4, q5\}$

1. Keadaan Awal: $q0$
2. Keadaan Penerimaan: $q5$
3. Simbol Input: $\Sigma = \{a, b\}$
4. Simbol Pita: $\Gamma = \{a, b,$ c, $B\}$

Fungsi Transisi δ:

δ	**a**	**b**	**c**	**B**
q0	(q1, c, R)	(q4, B, R)	(q0, B, R)	(q0, B, R)
q1	(q1, a, R)	(q2, b, R)		
q2	(q3, c, L)	(q2, c, R)	(q2, B, R)	
q3	(q3, a, L)	(q3, b, L)	(q3, c, L)	(q0, B, R)
q4	(q4, a, R)	(q4, B, R)	(q5, B, R)	

Proses:

Input "abaa":

$q0abaa \vdash cq1baa \vdash cbq2aa \vdash cq3bca \vdash q3cbca \vdash \mathrm{q}3Bcbca \vdash q0cbca \vdash q0bca \vdash q4ca \vdash q4a \vdash q4B \vdash q5$.

DTM menerima string dalam w jika, dan hanya jika, w dalam L.

7.3 Mesin Turing Nondeterministik (NTM)

Mesin Turing deterministik (DTM) hanya memiliki satu langkah yang dapat diambil dari setiap konfigurasi, menghasilkan satu konfigurasi berikutnya. Sebaliknya, Mesin Turing nondeterministik (NTM) bisa memiliki beberapa langkah dari suatu konfigurasi, menghasilkan beberapa konfigurasi berikutnya. Sebuah NTM M didefinisikan oleh keadaan Q, keadaan awal q0, keadaan penerimaan F, simbol input Σ, simbol pita Γ (termasuk simbol kosong B), dan relasi transisi Δ. Relasi transisi Δ adalah himpunan bagian dari $(Q - F) \times \Gamma \times Q \times \Gamma \times \{L,R\}$, yang memungkinkan berbagai langkah potensial. Konfigurasi berikut dari sebuah NTM pada input *w* diwakili oleh pohon perhitungan di mana setiap simpul adalah konfigurasi dan anak-anaknya adalah konfigurasi berikutnya. Sebuah NTM berhenti pada *w* jika ada jalur terbatas dalam pohon perhitungannya, dan menerima *w* jika jalur ini berakhir pada keadaan penerimaan. Bahasa yang diterima oleh sebuah NTM adalah himpunan string yang diterima oleh NTM tersebut.

NTM M dapat menghitung sebuah fungsi jika ada jalur perhitungan yang menghasilkan nilai tertentu dan setiap jalur penerimaan menghasilkan nilai yang sama. Teorema Church-Turing menyatakan bahwa DTM cukup kuat untuk mensimulasikan NTM, meskipun membutuhkan lebih banyak sumber daya. Teorema menunjukkan bahwa semua bahasa yang diterima oleh NTM adalah rekursif *enumerable*, dan semua fungsi yang dihitung oleh NTM adalah rekursif parsial. Simulasi NTM oleh DTM melibatkan penggunaan pita tambahan untuk menyimpan informasi simulasi dan menghapus sisa dari tahap sebelumnya. DTM kemudian menyalin input, menghasilkan string dalam urutan leksikografis, dan mensimulasikan langkah-langkah NTM berdasarkan simbol yang dihasilkan. Algoritma nondeterministik sering dijelaskan sebagai menebak-dan-memverifikasi, ketika langkah-langkah nondeterministik menebak jalur perhitungan, dan subrutin deterministik memverifikasi jalur tersebut.

7.4 *NP*

Kelas kompleksitas *NP* mencakup masalah yang dapat diverifikasi dalam waktu polinomial oleh sebuah mesin Turing deterministik jika diberikan sebuah "saksi" atau "sertifikat." Secara formal, sebuah bahasa *A* berada dalam *NP* jika dan hanya jika ada sebuah bahasa *B* dalam *P* dan sebuah fungsi polinomial *p* sehingga untuk setiap *instance x*:

$$x \in A\big(\exists y, |y| \leq p(|x|)\big)\langle x, y\rangle \in B \tag{7.1}$$

Bukti Theorem 2.1

(⇐) Misalkan $A = L(M)$ di mana *M* adalah mesin Turing nondeterministik dengan batas waktu polinomial $p(n)$. Kita mendefinisikan sebuah mesin Turing deterministik M^* yang menerima pasangan $<x, y>$. M^* mensimulasikan *M* pada input *x* menggunakan langkah-langkah yang ditentukan oleh *y*, dan menerima jika simulasi menerima dalam waktu polinomial. Maka, $x \in L(M)$ jika dan hanya jika ada *y* sedemikian sehingga $< x, y > \in L(M^*)$.

(⇒) Jika *A* dapat direpresentasikan oleh set *B* dan fungsi *p* seperti dalam (2.1), maka kita bisa mendesain mesin Turing nondeterministik *M* untuk *A* yang pertama membuat $p(|x|)$ gerakan nondeterministik untuk menulis *string* *y*, dan kemudian memverifikasi bahwa $< x, y > \in B$ menggunakan mesin Turing deterministik untuk B.

Contoh-Contoh Masalah dalam NP

1. *SATISFIABILITY* (SAT):

   Definisi: Diberikan sebuah formula *Boolean* ϕ, tentukan apakah ada penugasan nilai variabel yang membuat ϕ bernilai benar.

   Verifikasi: Untuk setiap formula Boolean ϕ dengan panjang n dan penugasan τ pada variabel-variabelnya, kita bisa memeriksa deterministik dalam waktu $O(n^2)$ apakah τ memuaskan ϕ. Oleh karena itu, SAT berada dalam *NP* karena kita bisa menebak penugasan τ dan memeriksanya dalam waktu polinomial.

2. HAMILTONIAN CIRCUIT (HC):

   Definisi: Diberikan sebuah graf G, tentukan apakah ada sebuah siklus yang melewati setiap simpul tepat satu kali.

   Verifikasi: Kita bisa menebak urutan simpul dan memverifikasi apakah urutan tersebut membentuk siklus Hamiltonian dalam waktu polinomial. Verifikasi mencakup memeriksa apakah urutan tersebut adalah permutasi dari semua simpul dan apakah setiap pasangan simpul yang bertetangga dalam urutan tersebut terhubung oleh sebuah tepi dalam graf.

3. INTEGER PROGRAMMING (*IP*):

   Definisi: Diberikan sebuah matriks bilangan bulat A dan sebuah vektor b, tentukan apakah ada sebuah vektor bilangan bulat x sehingga $Ax \geq b$.

   Verifikasi: Kita bisa menebak sebuah solusi bilangan bulat x dan memverifikasi ketidaksamaan $Ax \geq b$ dalam waktu polinomial. Namun, kita juga harus memastikan bahwa jika masalah memiliki solusi, maka ada solusi yang panjangnya dibatasi secara polinomial oleh ukuran input.

Lema Penting untuk *IP*

a. Lema 2.5: Jika B adalah submatriks persegi dari A, maka $|\det B| \leq (\alpha q)^q$, di mana α adalah nilai absolut maksimum elemen dalam A dan q adalah ordo dari B.

b. Lema 2.6: Jika *rank* dari A adalah $r < m$, maka ada vektor nonzero z sehingga $Az = 0$ dan nilai absolut dari setiap komponen z tidak melebihi $(\alpha q)^q$.

c. Lema 2.7: Jika $Ax \geq b$ memiliki solusi bilangan bulat, maka ada solusi bilangan bulat x' yang elemen-elemennya adalah bilangan bulat dengan panjang paling banyak $2(\alpha q)^{2q+2}$.

Lema-lema ini menunjukkan bahwa solusi bilangan bulat untuk IP memiliki panjang yang dibatasi secara polinomial oleh ukuran input. Hal ini memastikan bahwa masalah IP berada dalam NP karena bisa ditebak solusi bilangan bulat dan memverifikasi ketidaksamaan dalam waktu polinomial.

7.5 Teorema Cook

Konsep *reducibility* pertama kali dikembangkan dalam teori rekursi. *Reducibility* (≤r) adalah relasi biner pada bahasa yang memenuhi sifat refleksivitas dan transitivitas, sehingga membentuk urutan parsial pada kelas semua bahasa. *Reducibility many-one* dalam waktu polinomial (≤Pm) diperkenalkan untuk mengukur kesulitan masalah dalam kelas kompleksitas tertentu. Jika A ≤Pm B, artinya ada fungsi komputabel f yang memetakan setiap elemen dalam A ke B dalam waktu polinomial. *Reducibility* ini memenuhi sifat refleksivitas dan transitivitas.

Himpunan kompleksitas seperti P, NP, dan PSPACE tertutup di bawah ≤Pm. Artinya, jika A ≤Pm B dan B ∈ C, maka A juga termasuk dalam C. Sebuah himpunan B disebut ≤Pm-hard untuk kelas C jika setiap himpunan dalam C dapat direduksi ke B, dan disebut ≤Pm-complete jika B juga termasuk dalam C. Himpunan NP-complete adalah masalah paling sulit dalam NP yang tidak termasuk dalam P jika dan hanya jika P ≠ NP.

Masalah *halting* terbatas waktu (BHP) adalah masalah pertama yang terbukti NP-complete. BHP menentukan apakah mesin Turing nondeterministik (NTM) menerima masukan dalam sejumlah langkah tertentu. Teorema Cook menyatakan bahwa masalah *satisfiability* (SAT) adalah NP-complete. Buktinya menunjukkan bahwa setiap masalah A dalam NP dapat direduksi ke SAT dalam waktu polinomial. Dengan kata lain, jika SAT dapat diselesaikan dalam waktu polinomial, maka setiap masalah dalam NP juga dapat diselesaikan dalam waktu polinomial, yang berarti P = NP.

Proses pembuktian Teorema Cook melibatkan konstruksi formula Boolean Fx dari variabel yang mewakili konfigurasi NTM. Fx dibangun sedemikian rupa sehingga Fx satisfiable jika dan hanya jika NTM menerima input dalam waktu polinomial. Pembuktian ini menunjukkan bahwa SAT adalah masalah yang sangat penting dan sulit dalam teori kompleksitas. Keberadaan algoritma polinomial untuk SAT akan menyelesaikan semua masalah dalam NP dengan waktu polinomial.

Untuk mempermudah pembuktian masalah NP-complete lainnya, SAT dapat diubah menjadi bentuk normal konjungtif (CNF) atau 3-CNF, yang terdiri dari klausa dengan tiga literal. 3-SAT, yaitu SAT dalam bentuk 3-CNF, juga terbukti NP-complete, yang mempermudah reduksi dari masalah lain ke 3-SAT.

7.6 Masalah *NP*-Lengkap Lainnya

Konsep NP-kelengkapan sangat penting dengan ribuan masalah NP-lengkap yang ada dalam berbagai bidang seperti ilmu komputer, matematika diskrit, dan penelitian operasi. Secara teoritis, semua masalah ini bisa dibuktikan NP-lengkap dengan mereduksi SAT ke masalah tersebut. Namun, secara praktis lebih mudah membuktikan masalah NP-lengkap baru dari masalah NP-lengkap lain yang memiliki struktur serupa.

Dalam bagian ini, dipelajari beberapa masalah NP-lengkap terkenal:

1. *Vertex Cover* (*VC*): Diberikan graf $G = (V,E)$ dan bilangan bulat $K \geq 0$, tentukan apakah G memiliki *vertex cover* dengan ukuran paling banyak K.

   Teorema 2.14: *VC* adalah *NP*-lengkap.

   Bukti: *VC* berada dalam *NP* dan dapat dibuktikan *NP*-lengkap dengan mereduksi 3-SAT ke *VC*.

2. *Clique*: Diberikan graf $G = (V,E)$ dan bilangan bulat $K \geq 0$, tentukan apakah G memiliki klik dengan ukuran K.

   Independent Set (IS) : Diberikan graf $G = (V,E)$ dan bilangan bulat $K \geq 0$, tentukan apakah G memiliki himpunan independen dengan ukuran K.

   Teorema 2.15: Clique dan IS adalah NP-lengkap.

   Bukti: IS adalah NP-lengkap karena dapat direduksi dari VC, dan Clique adalah NP-lengkap karena dapat direduksi dari IS.

3. *Hamiltonian Circuit* (HC):

   Teorema 2.16: HC adalah NP-lengkap.

   Bukti: HC adalah NP-lengkap dengan mereduksi 3-SAT ke HC.

4. *Integer Programming* (*IP*):

   Teorema 2.17: IP adalah NP-lengkap.

   Bukti: IP adalah NP-lengkap dengan mereduksi 3-SAT ke IP.

5. *Subset Sum* (*SS*): Diberikan daftar bilangan bulat L = (a1, a2, …, an) dan bilangan bulat S, tentukan apakah ada subset $I \subseteq \{1, 2, \ldots, n\}$ sehingga $\sum_{i \in I} a_i = S$.

   Teorema 2.18: *SS* adalah *NP*-lengkap.

   Bukti: SS adalah NP-lengkap dengan mereduksi 3-SAT ke SS.

Bukti dan penjelasan masing-masing teorema melibatkan reduksi dari masalah NP-lengkap lain seperti 3-SAT dan menunjukkan bahwa keputusan untuk masalah-masalah tersebut dapat dilakukan dalam waktu polinomial.

7.7 Polinomial-Time Turing Reducibility

Bagian ini membahas reduksi Turing dalam waktu polinomial, bentuk reduksi yang lebih lemah dibandingkan dengan reduksi *polinomial-time many-one*. Reduksi *Turing* dalam waktu polinomial juga mempertahankan keanggotaan dalam kelas P, dan dapat diterapkan pada masalah pencarian (fungsi).

Secara intuitif, sebuah masalah A dikatakan *Turing-reducible* ke masalah B jika ada algoritma M untuk A yang dapat mengajukan beberapa pertanyaan mengenai keanggotaan di set B selama komputasi. Jika total waktu yang digunakan oleh M pada *input* x, tanpa waktu *querying*, dibatasi oleh polinomial p($|x|$), dan panjang setiap *query* yang diajukan juga dibatasi oleh p($|x|$), maka A dikatakan *Turing-reducible* dalam waktu polinomial ke B dan dilambangkan dengan $A \leq_P^T B$.

Contoh-contoh yang menggambarkan konsep ini adalah:

1. Untuk setiap set A, $A \leq_P^T A - complement$. Ini dilakukan dengan mengajukan oracle A apakah input x ada dalam A atau tidak, lalu membalikkan jawabannya. Ini menunjukkan bahwa reduksi $\leq_P^T$. bisa lebih lemah dibandingkan dengan reduksi $\leq_P^M$. jika $NP \neq$ coNP.
2. Untuk masalah *NP-complete CLIQUE*, kita mendefinisikan variasi masalahnya sebagai *EXACT-CLIQUE*, yang mengharuskan menentukan apakah ukuran maksimum klik dari graf G adalah K. Meskipun tidak jelas apakah *EXACT-CLIQUE* termasuk *NP*, kita menunjukkan bahwa *CLIQUE* dan *EXACT-CLIQUE* saling *Turing-reducible* dalam waktu *polinomial*, sehingga keduanya termasuk dalam P atau tidak termasuk P.
3. *MAX-CLIQUE* adalah *Turing-reducible* dalam waktu polinomial ke *CLIQUE* dan sebaliknya. Algoritma yang dijelaskan untuk *MAX-CLIQUE* menunjukkan bagaimana masalah ini dapat diselesaikan menggunakan *CLIQUE* dengan melakukan *query* terhadap berbagai ukuran K.

Bagian ini juga mendefinisikan mesin *Turing oracle* sebagai model formal untuk reduksi *Turing* dalam waktu *polinomial*. Mesin *Turing oracle* memiliki *tape query* tambahan dan dua status tambahan untuk menangani *query* ke fungsi atau *set oracle*. Proses komputasi dilakukan dengan memasukkan *string query* ke *tape query*, dan *oracle* menggantikan string tersebut dengan hasilnya.

Definisi reduksi Turing dalam waktu polinomial adalah sebagai berikut:

1. Sebuah fungsi parsial f Turing-reducible ke fungsi total g jika ada mesin *Turing oracle* M yang menghitung fungsi f menggunakan *oracle* g.
2. Sebuah set A *Turing-reducible* ke set B jika karakteristik fungsi χ_A Turing-reducible ke χ_B.

Proposisi-proposisi yang dibahas menunjukkan bahwa reduksi $\leq_P^T$. bersifat refleksif dan transitif, serta lebih lemah dibandingkan reduksi $\leq_P^m$.. Selain itu, kelas *P* dan *PSPACE* tertutup di bawah $\leq_P^T$, dan *NP* = *coNP* jika dan hanya jika *NP* tertutup di bawah $\leq_P^T$.

Terakhir, bagian ini mencatat bahwa kompleksitas ruang mesin *Turing oracle* lebih sulit didefinisikan dibandingkan kompleksitas waktu, terutama dalam hal penggunaan tape query. Pengukuran kompleksitas ruang diatur dengan cara alternatif di mana tape query dianggap sebagai tape yang hanya dapat ditulis dan tidak dihitung dalam pengukuran ruang.

7.8 Masalah Optimasi *NP-Complete*

Masalah optimasi kombinatorial sering kali merupakan masalah pencarian NP-hard. Untuk membuktikan *NP-hardness*, kita menunjukkan bahwa masalah keputusan terkait adalah ≤*P* m-complete untuk NP, dan kemudian membuktikan bahwa pencarian solusi optimal untuk masalah tersebut adalah ≤P.

T-equivalent dengan masalah keputusan terkait. Namun, dalam banyak aplikasi, solusi mendekati optimal sudah memadai. *NP-hardness* dari masalah optimasi tidak selalu berarti bahwa aproksimasi masalah tersebut juga *NP-hard*. Kami menunjukkan bahwa untuk beberapa masalah optimasi *NP-complete*, versi aproksimasinya juga *NP-hard*, sementara untuk masalah lainnya, aproksimasi waktu-*polynomial* mungkin dicapai.

Diperkenalkan kerangka umum untuk menangani masalah aproksimasi. Biasanya, masalah optimasi Π memiliki struktur umum yaitu untuk setiap instance input x, terdapat sejumlah solusi y untuk x. Tujuan dari masalah Π adalah menemukan solusi y sehingga nilai $v(y)$ dimaksimalkan (atau diminimalkan). Sebagai contoh, masalah *MAX-CLIQUE* cocok dengan kerangka ini yaitu input adalah graf *G*, solusi adalah klik

C dalam G, nilai $v(C)$ adalah jumlah *vertex* dalam C, dan masalahnya adalah menemukan klik terbesar dalam graf G.

Untuk masalah maksimalisasi Π dengan rasio aproksimasi $r > 1$, versi aproksimasi dari Π didefinisikan sebagai r-APPROX- Π, yaitu, menemukan solusi y dengan $v(y) = \frac{v*(x)}{\tau}$, dengan $v^*(x)$ adalah nilai maksimum dari semua solusi untuk x. Untuk masalah minimisasi Π, versi aproksimasinya dengan rasio r adalah menemukan solusi y dengan $v(y) \leq r \,.\, v^*(x)$.

Kemudian dipertimbangkan masalah optimasi terkenal: *Traveling Salesman Problem* (TSP), yang meminta tur terpendek dari n kota pada peta. Versi keputusan dari TSP adalah menentukan apakah ada *tur* (*Hamiltonian circuit*) dengan biaya total kurang dari atau sama dengan K. TSP terbukti *NP-complete* dengan reduksi sederhana dari *HC* (*Hamiltonian Circuit*) ke TSP. Selain itu, masalah pencarian *TSP* juga *NP-hard*, dan aproksimasi untuk TSP juga NP-hard untuk setiap rasio $r > 1$.

TSP yang memenuhi ketentuan ketidaksetaraan segitiga (*triangle inequality*) juga *NP-complete*. Namun, untuk versi aproksimasi TSP yang memenuhi ketidaksetaraan segitiga, algoritma aproksimasi polinomial ada untuk semua rasio $r \geq \frac{3}{2}$. Apakah rasio ini bisa ditingkatkan masih merupakan pertanyaan terbuka. Versi Euclidean TSP pada bidang dua dimensi juga memiliki algoritma aproksimasi polinomial untuk semua rasio $r > 1$.

Selanjutnya, diperkenalkan hasil bahwa aproksimasi $\frac{3}{2}$-APPROX-TSP pada graf yang memenuhi ketidaksetaraan segitiga dapat diselesaikan dalam waktu polinomial. Digunakan teknik yang melibatkan pohon merentang minimum (Minimum Spanning Tree, MST) dan pencocokan minimum (Minimum Matching) untuk mencapai hasil ini.

Terakhir, dikaji masalah *knapsack*, yang merupakan contoh masalah dengan FPTAS (*Fully Polynomial-Time*

Approximation Scheme). Untuk masalah *knapsack*, algoritma aproksimasi FPTAS dapat memberikan solusi mendekati optimal dalam waktu polinomial terhadap ukuran input *n* dan parameter *W*.

1. Kompleksitas Komputasi
    - Kompleksitas Waktu dan Kompleksitas Ruang adalah metrik utama untuk mengevaluasi efisiensi algoritma. Kompleksitas waktu mengukur berapa lama algoritma bekerja sebagai fungsi dari ukuran input, sedangkan kompleksitas ruang mengukur seberapa banyak memori yang digunakan.
    - Algoritma yang efisien penting untuk aplikasi nyata di mana sumber daya terbatas, seperti dalam sistem real-time.
2. Mesin Turing Deterministik (DTM)
    - DTM memiliki unit kontrol, unit memori, dan kepala baca/tulis yang dapat bergerak di sepanjang pita memori. Operasinya meliputi membaca, menulis, dan memindahkan kepala pita.
    - Definisi Formal dan Fungsi Transisi DTM menyediakan cara untuk formal dan sistematik dalam menentukan hasil dari algoritma berdasarkan konfigurasi dan transisi.
3. Mesin Turing Nondeterministik (NTM)
    - Berbeda dengan DTM, NTM dapat memiliki beberapa langkah transisi dari satu konfigurasi, memungkinkan lebih banyak kemungkinan konfigurasi berikutnya.
    - Teorema *Church-Turing* menunjukkan bahwa DTM dapat mensimulasikan NTM dengan lebih banyak sumber daya, mengarah pada pemahaman bahwa

NTM dan DTM memiliki kekuatan komputasi yang setara dalam hal fungsi yang dapat dihitung.

4. Kompleksitas Kelas NP

    - NP (*Nondeterministic Polynomial time*) mencakup masalah yang solusinya dapat diverifikasi dalam waktu polinomial. Masalah dalam NP dapat memiliki sertifikat verifikasi yang memudahkan proses verifikasi solusi.
    - Masalah NP-*Complete* seperti SAT, HC, IP, dan SS merupakan masalah yang paling sulit dalam NP. Teorema Cook menyatakan bahwa SAT adalah NP-*complete*, mengindikasikan bahwa setiap masalah dalam NP dapat direduksi ke SAT dalam waktu polinomial.
    - Teorema menunjukkan bahwa masalah-masalah ini, jika dapat diselesaikan dalam waktu polinomial, berarti P = NP.

5. Reduksi Turing dalam Waktu Polinomial

    - Reduksi Turing dalam waktu polinomial ($\leq_p^T$) mengukur kekuatan relatif masalah berdasarkan kemampuan mereka untuk saling mengajukan pertanyaan tentang keanggotaan dalam waktu polinomial.
    - Contoh termasuk pengurangan antara CLIQUE dan MAX-CLIQUE serta masalah TSP. Reduksi ini membantu dalam memahami hubungan antar masalah dalam hal kompleksitas mereka.

6. Masalah Optimasi NP-Complete

    - Masalah optimasi sering kali adalah NP-*hard*, dan membuktikan NP-*hardness* melibatkan pengurangan dari masalah keputusan terkait.

- Contoh* masalah optimasi termasuk *Traveling Salesman Problem* (TSP) dan Knapsack, ketika TSP memiliki solusi aproksimasi dengan rasio tertentu, sementara Knapsack dapat diselesaikan dengan algoritma FPTAS.

Bab 8

Integrasi Matematika dengan Teknologi

8.1 Perkembangan Matematika

Matematika telah berkembang dari alat praktis untuk perhitungan sehari-hari menjadi disiplin ilmu yang sangat abstrak dan kompleks. Penemuan dan inovasi dalam matematika telah memungkinkan perkembangan teknologi dan sains, serta pemahaman yang lebih dalam tentang alam semesta. Matematika terus menjadi bidang yang dinamis dengan banyak area penelitian baru yang terus berkembang.

Mulanya bangsa Mesir mengembangkan matematika untuk kebutuhan praktis seperti pengukuran lahan dan konstruksi piramida. Mereka menggunakan sistem bilangan desimal dan hieroglif untuk menghitung. Matematika Mesir Kuno menunjukkan bagaimana masyarakat kuno mengembangkan dan menerapkan konsep matematika untuk memenuhi kebutuhan praktis mereka. Ini juga mencerminkan kecerdikan dan kemampuan mereka dalam mengelola sumber daya, membangun monumen, dan mengatur kehidupan sehari-hari mereka dengan bantuan matematika.

Pada abad pertengahan, ilmuwan Muslim seperti Al-Khwarizmi dan Omar Khayyam membuat kontribusi signifikan dalam aljabar, trigonometri, dan astronomi. Al-Khwarizmi menulis kitab yang menjadi dasar kata "algebra". Sementara di

India Matematikawan seperti Aryabhata dan Brahmagupta mengembangkan konsep nol, sistem bilangan desimal, dan metode kalkulasi yang kompleks.

Perkembangan aplikasi matematika di zaman renaisans dan awal modern, matematikawan seperti Fibonacci memperkenalkan sistem bilangan Hindu-Arab ke Eropa. Pada abad ke-17, tokoh-tokoh seperti Descartes, Fermat, dan Pascal mengembangkan konsep-konsep dasar dalam aljabar dan probabilitas. Isaac Newton dan Gottfried Wilhelm Leibniz secara independen mengembangkan kalkulus, yang menjadi alat penting dalam matematika dan ilmu fisika.

Perkembangan matematika modern terjadi di abad ke-20 hingga sekarang. Perkembangan besar terjadi dalam analisis, topologi, dan geometri diferensial. Matematikawan seperti Kurt Gödel, Alan Turing, dan John von Neumann membuat terobosan dalam logika, teori komputer, dan teori game. Era komputer membawa perkembangan baru dalam matematika komputasi, termasuk algoritma, kriptografi, dan kecerdasan buatan. Matematika terus berkembang dalam aplikasi ke bidang fisika, teknik, ekonomi, biologi, dan ilmu sosial. Model matematika digunakan untuk memecahkan masalah kompleks di dunia nyata.

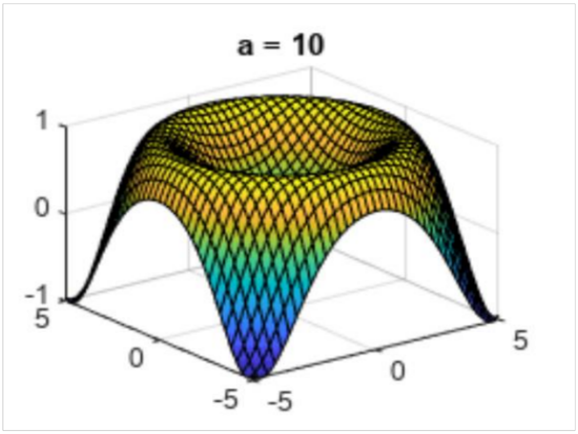
8.2 Penggunaan Software Matematika

Software matematika memainkan peran penting dalam membantu para ilmuwan, insinyur, dan matematikawan untuk memecahkan masalah kompleks, memvisualisasikan data, dan melakukan perhitungan yang rumit. Berikut adalah beberapa *software* matematika yang umum digunakan beserta penggunaannya:

8.2.1. Matlab

Matlab (*Matrix Laboratory*) merupakan salah satu aplikasi yang banyak digunakan untuk kegiatan analisis dan komputasi numerik. Aplikasi ini menggunakan bahasa pemrograman matematika lanjutan yang beroperasi dalam bentuk matriks (Cahyono, 2013). Sebelum aplikasi ini menjadi produk komersial dari perusahaan Mathworks Inc, pada awalnya program ini merupakan *interface* untuk koleksi rutin-rutin numerik yang berasal dari LINPACK dan EISPACK, dan selanjutnya dikembangkan dengan mengaplikasikan bahasa FORTRAN.

Pemanfaatan perangkat lunak MATLAB akan menghasilkan penyelesaian aljabar linear yang cepat dan akurat dibandingkan dengan perhitungan manual. Selain itu, hasil pengolahan datanya dapat disajikan secara visual dalam bentuk dua dimensi maupun tiga dimensi. Gambar 8.1 menunjukkan visual grafik persamaan $\sin((x^2 + y^2)/a)$ dengan $a = 10$.



Gambar 8.1 Grafik tiga dimensi pada aplikasi MATLAB

Sumber: MathWorks®

8.2.2. Aplikasi R

Aplikasi R menjadi salah satu program yang dapat diakses dengan mudah tanpa perlu membayar karena berada di bawah GPL (*General Public License*). Aplikasi yang merupakan wujud kolaborasi riset dari pakar statistik di seluruh dunia ini dapat digunakan untuk kegiatan analisis data (Sihombing et al., 2019). Dengan kemudahan aksesnya, aplikasi ini dimanfaatkan dalam penyelesaian soal kalkulus yang menunjang kemampuan pemrograman mahasiswa (Fauzan & Mahara, 2020). R menyediakan beragam teknik statistik (pemodelan linier dan nonlinier, uji statistik klasik, analisis deret waktu, klasifikasi, pengelompokan, dan grafis.

8.3 Kalkulus dan Aljabar dalam Teknologi

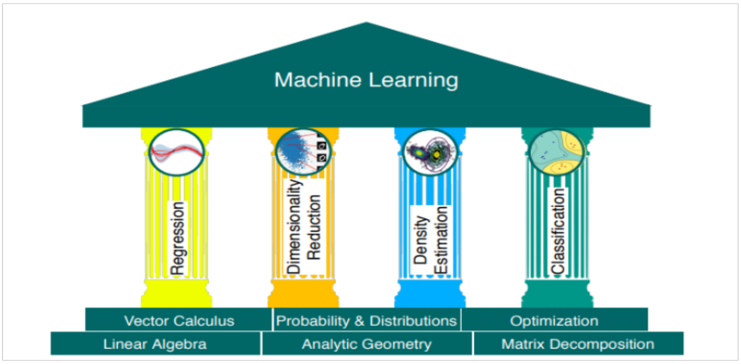
Perkembangan matematika, terutama dalam bidang kalkulus dan aljabar, telah mengalami evolusi yang signifikan sepanjang sejarah (Irmayanti et al., 2021). Berikut adalah beberapa cara kalkulus telah mempengaruhi teknologi.

1. Analisis Struktur dan Material: Kalkulus digunakan untuk menghitung tegangan, regangan, dan deformasi dalam struktur bangunan dan mesin.
2. Dinamika Fluida**:** Dalam teknik sipil dan mekanik, kalkulus digunakan untuk memodelkan aliran fluida dalam pipa, jembatan, pesawat terbang, dan mobil.
3. Sistem Kontrol: Kalkulus membantu dalam desain sistem kontrol otomatis, seperti dalam robotika dan sistem kendali penerbangan.
4. Gerak dan Gaya: Hukum Newton dan persamaan gerak memerlukan kalkulus untuk memodelkan dan memprediksi perilaku objek.
5. Elektromagnetisme: Persamaan Maxwell yang mengatur listrik dan magnetisme merupakan persamaan diferensial parsial yang memerlukan kalkulus untuk diselesaikan.
6. Relativitas: Teori relativitas umum Einstein menggunakan kalkulus diferensial dan integral untuk menjelaskan kelengkungan ruang-waktu.

7. Pemrosesan Sinyal: Kalkulus digunakan dalam analisis sinyal digital dan analog, seperti dalam audio, gambar, dan video.
8. Pembelajaran Mesin: Algoritma optimasi dan backpropagation dalam jaringan saraf tiruan menggunakan kalkulus diferensial.
9. Grafik Komputer: Kalkulus digunakan dalam rendering gambar, simulasi fisika, dan animasi komputer.
10. Optimasi Portofolio: Kalkulus digunakan untuk memaksimalkan keuntungan dan meminimalkan risiko dalam investasi.
11. Model Penetapan Harga Opsi: Model matematika seperti Black-Scholes menggunakan kalkulus untuk menentukan harga opsi.
12. Analisis Ekonomi: Kalkulus digunakan dalam analisis marginal dan teori ekonomi untuk memahami hubungan antara variabel ekonomi.
13. Model Populasi: Persamaan diferensial digunakan untuk memodelkan pertumbuhan populasi dan interaksi spesies.
14. Farmakokinetik: Kalkulus digunakan untuk memodelkan bagaimana obat diserap, didistribusikan, dimetabolisme, dan dieliminasi oleh tubuh.
15. Pemodelan Penyakit: Kalkulus membantu memodelkan penyebaran penyakit menular dan efek intervensi kesehatan.

8.4 Machine Learning

Machine learning adalah cabang dari kecerdasan buatan (AI) yang berfokus pada pengembangan algoritma yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit (Patil et al., 2023). Gambar 8.2 memperlihatkan empat pilar pada *machine learning* (Peter et al., 2019).



Gambar 8.2 Empat pilar *machine learning*

Pemahaman operasi dasar aljabar linear menjadi hal yang wajib dimiliki bagi siapa saja yang ingin menjadi ahli dalam *machine learning*. Aljabar linear memberikan kerangka kerja pada kegiatan manipulasi data yang ditampilkan dalam bentuk vector dan matriks. Dalam *machine learning,* operasi aljabar linear dimanfaatkan secara luas pada beberapa tahap yakni pengolahan data hingga *training* dan evaluasi model yang dikembangkan.

Dimensionality reduction atau pengurangan dimensi adalah tugas mengurangi jumlah fitur dalam kumpulan data. Dalam tugas *machine learning* seperti regresi atau klasifikasi, sering kali memiliki terlalu banyak variabel yang harus dikerjakan. Variabel-variabel ini disebut juga fitur. Semakin tinggi jumlah fitur maka semakin sulit untuk memodelkannya. Selain itu, beberapa fitur ini bisa jadi tidak bermanfaat, sehingga menambah gangguan pada kumpulan data. Oleh karena itu, fitur perlu dikurangi.

Estimasi kepadatan adalah salah satu hal dasar yang dijumpai dalam statistik dan *machine learning*. Estimasi kepadatan bertujuan untuk menemukan distribusi probabilitas yang menggambarkan data tertentu. Salah satu teknik untuk proses estimasi kepadatan adalah Model campuran Gaussian.

Klasifikasi (*classification*) didefinisikan sebagai proses pengenalan, pemahaman, dan pengelompokan objek dan ide ke dalam kategori yang telah ditentukan sebelumnya. Dengan bantuan kumpulan data pelatihan yang telah dikategorikan sebelumnya, klasifikasi dalam program *machine learning* memanfaatkan berbagai algoritma untuk mengklasifikasikan kumpulan data masa depan ke dalam kategori yang relevan dan relevan.

8.4.1 Supervised Learning

Jenis *machine learning* ini melatih algoritma menggunakan data yang sudah diberi label. Model belajar untuk memetakan *input* ke *output* berdasarkan data latih**.** Contoh Algoritma yang digunakan meliputi regresi linear, regresi logistik, decision trees, random forests, support vector machines, *neural networks*. Implementasi *supervised learning* seperti **k**lasifikasi email spam, prediksi harga rumah, pengenalan suara.

8.4.2 Unsupervised Learning

Algoritma pada *unsupervised learning* bekerja dengan data yang tidak berlabel dan mencoba menemukan struktur tersembunyi dalam data. Contoh algoritma seperti *K-means clustering, hierarchical clustering, principal component analysis (PCA), autoencoders*. Aplikasi jenis *machine learning* ini seperti segmentasi pelanggan, deteksi anomali, dan kompresi data.

8.4.3 Semi-Supervised Learning

Semi-Supervised Learning merupakan metode yang menggunakan sejumlah kecil data berlabel dan sejumlah besar data tidak berlabel untuk pelatihan. Tujuan dari pembelajaran semi-supervisi adalah untuk mempelajari suatu fungsi yang secara akurat dapat memprediksi variabel keluaran berdasarkan variabel masukan, serupa dengan pembelajaran yang diawasi.

Namun, tidak seperti pembelajaran yang diawasi, algoritma dilatih pada kumpulan data yang berisi data berlabel dan tidak berlabel.

8.4.4 Reinforcement Learning

Algoritma belajar melalui percobaan dan kesalahan dengan mendapatkan umpan balik berupa *reward* atau *punishment.* Contoh Algoritma yang banyak diterapkan antara lain *Q-learning*, *deep Q-networks* (DQN), *policy gradients*. Algoritma ini dipakai pada aplikasi pengendalian robot, permainan video, pengoptimalan portofolio.

8.5 Optimisasi

Optimisasi adalah cabang matematika dan ilmu komputer yang berkaitan dengan menemukan solusi terbaik (optimal) dari suatu masalah di antara sekumpulan solusi yang mungkin. Optimisasi memiliki berbagai aplikasi dalam ilmu pengetahuan, teknik, ekonomi, dan bisnis (Widayanti, 2017). Metode ini disebut-sebut sebagai alat yang sangat kuat yang memungkinkan pengambilan keputusan yang lebih baik di berbagai bidang dengan meningkatkan efisiensi, mengurangi biaya, dan meningkatkan kinerja.

8.6 Pemodelan dan Simulasi

8.6.1 Simulasi Fisika

Simulasi fisika adalah teknik untuk memodelkan dan memprediksi perilaku sistem fisik melalui penggunaan komputer. Teknik ini sangat penting dalam berbagai bidang ilmu pengetahuan dan teknik, memungkinkan para peneliti dan insinyur untuk memahami fenomena kompleks, menguji hipotesis, dan merancang sistem baru tanpa harus melakukan eksperimen fisik yang mahal atau tidak praktis.

Metode Elemen Hingga menggunakan algoritma dengan membagi domain masalah menjadi elemen-elemen kecil dan menyelesaikan persamaan diferensial parsial yang mengatur perilaku fisik pada setiap elemen. Metode ini diterapkan pada analisis struktur, simulasi termal, dan elektromagnetisme. Metode volume hingga memiliki algoritma dengan membagi domain menjadi volume kontrol kecil dan menyelesaikan integral dari persamaan konservasi.

Metode Dinamika Molekuler merupakan simulasi gerakan atom dan molekul berdasarkan hukum Newton dan potensi interaksi antar partikel. Metode ini diterapkan pada simulasi material, biologi molekuler, kimia. Berbeda dengan metode lain, metode Monte Carlo Menggunakan statistik dan sampling acak untuk menyelesaikan masalah fisika seperti fisika statistik, pemodelan radiasi, dan optimisasi.

8.6.2 Klimatologi

Matematika memiliki peran penting dalam klimatologi, terutama dalam pemodelan, analisis, dan prediksi fenomena iklim. Pemodelan iklim melibatkan penggunaan persamaan matematika untuk merepresentasikan proses fisik dan kimia yang terjadi dalam sistem iklim Bumi. Beberapa metode matematika yang digunakan seperti persamaan diferensial parsial, motode numerik dan pemodelan stokastik.

Kegiatan analisis data iklim melibatkan matematika untuk mengolah data yang dikumpulkan dari berbagai sumber, seperti satelit, stasiun cuaca, dan model iklim. Statistik dan Probabilitas digunakan untuk menganalisis tren, variabilitas, dan ekstrem iklim. Analisis deret waktu digunakan untuk menganalisis data iklim yang dikumpulkan secara berurutan sepanjang waktu. Analisis spektral, transformasi Fourier, dan analisis komponen utama (Principal Component Analysis, PCA) untuk mengidentifikasi pola temporal dan spasial.

Matematika memungkinkan prediksi kondisi iklim jangka pendek dan proyeksi perubahan iklim jangka panjang. Model Prediksi numerik cuaca menggunakan model matematika untuk memprediksi kondisi cuaca dalam jangka pendek. Contoh penerapannya adalah model atmosfer global dan regional yang memecahkan persamaan diferensial untuk dinamika atmosfer. Berbeda dengan model prediksi numerik, model iklim global digunakan untuk merepresentasikan interaksi kompleks antara atmosfer, lautan, daratan, dan es untuk memproyeksikan perubahan iklim jangka panjang. Implementasi model iklim global seperti pada *Model Coupled Model Intercomparison Project* (CMIP) yang digunakan dalam laporan *Intergovernmental Panel on Climate Change* (IPCC).

8.7 Statistika dan Analisis Data

Matematika memainkan peran mendasar dalam statistika dan analisis data, serta menyediakan alat dan kerangka kerja untuk mengumpulkan, menganalisis, dan menafsirkan data. Matematika menyediakan fondasi teoritis dan praktis yang diperlukan untuk melakukan analisis statistik dan data, memungkinkan kita untuk mengambil keputusan yang lebih baik berdasarkan data yang tersedia. Beberapa konsep matematika pada statistika adalah aljabar linear, kalkulus, teori probabilitas, matematika diskrit, teori angka dan analisis ril. Dengan memahami konsep-konsep matematika ini, analisis statistik dapat dilakukan dengan lebih akurat dan efisien.

8.8 Visułalisasi Data

Visualisasi data adalah teknik untuk mengubah data menjadi representasi grafis atau visual agar lebih mudah dipahami dan dianalisis. Ini sangat penting dalam mengidentifikasi pola, tren, dan anomali dalam data yang mungkin tidak terlihat melalui analisis data mentah. Data yang diolah oleh alat komputasi akan ditampilkan dalam beberapa bentuk antara lain grafik batang, grafik lurus, grafik lingkaran, histogram, *scatter plot*, *heatmap*, *box plot*, *tree map*, *graph* dan diagram jaringan. Visualisasi data menjadi alat yang kuat untuk memahami dan menyampaikan informasi yang kompleks. Dengan menggunakan teknik visualisasi yang tepat, data dapat diubah menjadi wawasan yang dapat ditindaklanjuti.

Bab 9

Penerapan Konsep Matematika dalam Penelitian

Pengenalan matematika bukan dimulai dari kehidupan sekolah, namun kehidupan sehari-hari, yang mana adanya kegiatan proses menghitung dalam proses kehidupan misalnya belanja atau menyatakan banyak atau tidaknya barang yang dimiliki. Dalam dunia sekolah, proses matematika yang dilakukan sehari-hari mengalami progress kearah konsep yaitu adanya sekumpulan rumus yang diperoleh dari masalah-masalah kehidupan. Menurut Tymoczko (1986) dan Ernest (1991), matematika diawali dengan aktivitas manusia yang melibatkan pemecahan situasi problematis. Dalam menemukan respon atau solusi terhadap permasalahan eksternal dan internal tersebut, objek matematika semakin muncul dan berkembang. Kemudian permasalahan matematika dan penyelesaiannya dibagikan pada lembaga atau kolektif tertentu yang terlibat dalam mempelajari permasalahan tersebut. Kemudian simbol-simbol digunakan untuk mengungkapkan situasi-masalah dan solusi yang ditemukan sehingga simbol-simbol ini menjadi bersifat komunikatif antara ahli matematik dan masyarakat. Ketika sistem ini diterima maka terbentuk konsep matematika. Jadi konsep matematika adalah pandangan dari pemikiran yang abstrak, digunakan untuk mengelompokkan suatu masalah atau kejadian, misalnya angka satu yang merupakan pandangan abstrak dalam perhitungan.

Seperti penjelasan sebelumnya, konsep matematika erat kaitannya dengan masalah sehari-hari. Ahmad (2021) menyatakan bahwa ada konsep matematika yang umumnya digunakan pada ekonomi adalah variabel, konstanta, koefisien, dan parameter; persamaan dan pertidaksamaan; konsep dan teori himpunan; sistem bilangan real; aturan pangkat, akar, pemfaktoran, serta; pecahan, desimal dan persentase. Matematika dapat digunakan sebagai prediksi suatu fenomena seperti keuntungan atau kerugian, jumlah penduduk bahkan penyebaran penyakit.

Matematika adalah alat yang ampuh untuk pemahaman dan komunikasi global. Nst,dkk (2023) menyatakan matematika dapat digunakan pada algoritma komputer dengan logika matematika, teori peluang, teori graf dapat digunakan pada penyusunan kode bahasa pemograman. Konsep matematika juga digunakan dalam bidang industi terutama saat ini ini ketika industri berkembang. Kartono (2003) menjelaskan bahwa konsep matematika dapat digunakan dalam penyelesaian masalah dan solusi yang disarankan lewat analisa menggunakan matematika. Masalah industri yang sangat kompleks, dimulai dari perencanaan bahan baku yang memperhitungkan hasil produksi, biaya produksi dan keuntungan; bagaimana mesin bekerja dengan seeifisien mungkin dan bagaimana memasarkan hasil produksi dengan mempertimbahkan permintaan. Penggunaan konsep matematika ini dapat diaplikasikan pada berbagai bidang karena dikaji terlebih dahulu lewat penelitian yang analisanya tidak lepas dari matematika, menganalisis data, menguji hipotesis, dan menarik kesimpulan yang valid berdasarkan bukti empiris.

Saat ini matematika dimasukkan dalam kurikulum pendidikan STEM, karena dianggap penting dalam mencari solusi dair masalah tantangan global yang ada. Matematika dianggap dasar ilmu yang digunakan untuk membentuk pemikiran yang sistematis dan kritis. Konsep matematika melalui metode berpikir matematis dapat mengarahkan kita (juga mahasiswa atau siswa) secara sadar atau tidak sadar membetuk pola pikir yang mampu memunculkan ide dan menganalisanya lewat teori-teori yang ada. Memecahkan masalah kompleks yang melibatkan banyak variabel dan batasan dan merancang solusi inovatif.

9.1 Konsep Matematika

Berikut ini diberikan konsep matematika yang sering digunakan sebagai dasar penelitian.

9.1.1 Logika Matematika

Logika matematika sering dikaitkan dengan penalaran, yang digunakan untuk pembuktian logis dengan mengevaluasi nilai kebenaran suatu pernyataan. Pada logika matematika terdapat berbagai aturan yang berbeda untuk membedakan hasil logika benar dan salah. Logika matematika dapat digunakan pada penelitian bidang komputer, mulai dari desain sirkuit digital hingga konstruksi program komputer dan verifikasi kebenaran program.

9.1.2 Operator Logika Matematika

1. Konjungsi: menghubungkan dua pernyataan yang disimbolkan dengan "$\wedge$" (DAN), kesimpulan pernyataan bernilai benar jika kedua pernyataan benar dan salah jika salah satu pernyataan salah.

2. Disjungsi: menghubungkan dua pernyataan yang disimbolkan dengan "$\vee$" (ATAU), kesimpulan pernyataan bernilai benar jika kedua pernyataan salah maka kesimpulan

bernilai salah dan jika salah satu pernyataan bernilai benar maka kesimpulan bernilai benar.

3. Negasi: operator yang menyatakan kebalikan "~" dari suatu pernyataan.

9.1.3 Rule Logika matematika

1. Hukum komutatif

$$m \wedge n \equiv n \wedge m \qquad (9.1)$$

$$m \vee n \equiv n \vee m$$

2. Hukum Asosiatif

$$(m \wedge n) \wedge a \equiv m(n \wedge a) \qquad (9.2)$$

$$(m \vee n) \vee a \equiv m(n \vee a)$$

3. Hukum Distribusi

$$m \wedge (n \vee a) \equiv (m \wedge n) \vee (m \wedge a) \qquad (9.3)$$

$$m \vee (n \wedge a) \equiv (m \vee n) \wedge (m \vee a)$$

4. De morgan laws

$$\neg(m \wedge n) \equiv \neg m \vee \neg n \qquad (9.4)$$

$$\neg(m \vee n) \equiv \neg m \wedge \neg n$$

9.2 Teori Peluang

Sering didengar kata untung-untungan ketika dihadapkan dengan pilihan. Misalnya ketika ada dua orang teman berjanji jika mendapat kepala ketika melemparkan koin maka akan mentraktir makan, dan ketika dilempar, yang muncul adalah ekor. Namun sebenarnya, hal yang dilakukan ini adalah berhubungan dengan peluang, yaitu adanya peristiwa acak yang tidak dapat ditentukan, namun hasilnya dapat berupa kemungkinan. Prediksi tentang penyakit genetik pada anak yang diturunkan dari orang tuanya dapat menggunakan teori peluang yang dikenal dengan frekuensi relatif. Pada teori peluang perlu dilakukan beberapa kali eksperiman acak untuk mendapatkan kemungkinan hasil. Himpunan semua kemungkinan hasil dari eksperimen acak itu dikenal dengan ruang sampel. Contoh pada pelemparan koin sebelumnya, maka ruang sampelnya adalah {kepala, ekor}. Probabilitas (peluang) suatu kejadian berada antara 0 dan 1, karena jumlah hasil yang diinginkan tidak akan melebihi jumlah total hasil suatu peristiwa.

$$\boldsymbol{P(A) = \frac{n(A)}{n(S)}} \tag{9.5}$$

keterangan:

P(A): peluang kejadian A

n(A): peluang anggota kejadian A

n(S): banyaknya titik sampel

9.3 Dasar Statistika

Pada dasarnya pengetahuan mengenai dunia dan isinya termasuk fenomena-fenomena yang terjadi adalah hasil pekerjaan peneliti. Awalnya dimulai dari pemikiran mengenai sekitar yang hanya berdasarkan pengetahuan sederhana saja sehingga muncul asumsi, kemudian semakin tertariknya terhadap hal tersebut, maka diberikan waktu dan dilakukan eksperimen-eksperimen. Namun tidak mungkin dipelajari keseluruhan dari hal yang ingin diketahui lebih dalam, karena itu diperlukan sampel dan diharapkan sampel ini dapat mewakili target. Mengumpulkan data, menganalisa dan mengkomunikasikannya lewat kesimpulan disebut metode statistika. Berikut ini pada umumnya hal-hal yang perlu dipelajari sebagai dasar statistika:

9.3.1 Variabel dan Konstanta

Sesuatu yang belum pasti dalam hal ini kemungkinan berubah maka disebut variabel, sebaliknya yang tidak berubah disebut konstanta. Misalnya banyaknya peminat yang mendaftar ke suatu sekolah, karena bervariasi dapat disebut variabel. Adapun variabel ini dapat dibedakan menjadi variabel kuantitatif yaitu data-data berupa numerik dan variabel kualitatif contohnya warna rambut, jenis kelamin. Pada variabel kuantitatif, terdapat data ordinal, kategorial, interval dan rasio. Ketika data dikategorikan dan tidak berurut disebut data kategorial, contoh jenis kelamin laki-laki dan perempuan. Untuk data yang diurut maka disebut data ordinal. Data ordinal dikelompokkan dengan jarak yang sama disebut data interval sedangkan dikelompokkan dengan perbandingan yang sama disebut data rasio.

9.3.2 Populasi dan Sampel

Kumpulan data dikumpulkan dari rangkaian prosedur yang telah ditentukan. Populasi merupakan kumpulan data yang diambil dari semua anggota kelompok misalnya populasi suatu negara, sedangkan sampel merupakan kelompok tertentu yang dikumpulkan datanya dari suatu populasi, sehingga ukuran sampel lebih kecil dari ukuran populasi. Ada kalanya penelitian membutuhkan data populasi misalnya data ipk lulusan mahasiswa tahun 2022 di suatu kampus, data ini dapat diperoleh dari seluruh alumni yang lulus tahun 2022, namun ketika ingin mengetahui pendapatan suatu provinsi, terlalu sulit mendapatkan informasi dari setiap warga, karena itu dilakukan pengambilan sample untuk mengambil sampel dari warga provinsi tersebut. Pada teknik pengambilan sampel dikenal probalilitas ketika sampel didapat secara acak, dan jika tidak dilakukan acak misalnya berdasarkan kategori disebut non-probability sampling. Pengambilan sampel ini sangatlah efektif, karena penyimpanannya mudah dibandingkan data populasi yang besar, biaya yang dikeluarkan juga lebih sedikit, dan ada kalanya kesulitan mengaskes data populasi. Berikut ini rumus populasi dan sampel, misalkan n banyaknya populasi dan n-1 besarnya sampel (https://byjus.com/maths/population-and-sample/):

$$\text{Population MAD} = \frac{1}{n}\sum_{\mathrm{i}=1}^{n} |x_i - \overline{x}| \tag{9.6}$$

$$\text{Population Variance} = (\sigma x)^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2 \tag{9.7}$$

$$\text{Sample MAD} = \frac{1}{n-1}\sum_{i=1}^{n} |x_i - \overline{x}| \tag{9.8}$$

$$\text{Sample Variance} = (Sx)^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \overline{x})^2 \tag{9.9}$$

$$\text{Population Standard Deviation} = \sigma x = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2} \tag{9.10}$$

$$\text{Sample Standard Deviation } = Sx \qquad (9.11)$$

$$= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

9.4 Penelitian

Kata "penelitian" berasal dari kata Perancis kuno "recerchier" yang berarti mencari dan mencari lagi. Secara harfiah berarti mengulangi pencarian sesuatu dan secara implisit mengasumsikan bahwa pencarian sebelumnya tidak menyeluruh dan lengkap dalam arti bahwa masih ada ruang untuk perbaikan. Penelitian melibatkan tiga hal dasar:

1. Pengumpulan data: Mengacu pada mengamati, mengukur, dan mencatat informasi.
2. Analisis data: Mengacu pada pengaturan dan pengorganisasian data yang dikumpulkan sehingga kita dapat melakukannya mencari tahu apa signifikansinya dan menggeneralisasikannya.
3. Penulisan laporan : Merupakan bagian yang tidak terpisahkan dan merupakan hasil akhir dari suatu penelitian. Tujuannya adalah untuk menyampaikan informasi yang terkandung di dalamnya kepada pembaca atau khalayak.

Penelitian adalah suatu proses yang melaluinya berusaha untuk mencapai jawaban atas pertanyaan, penyelesaian masalah, atau pemahaman yang lebih baik tentang suatu fenomena secara sistematis dan dengan dukungan data. Proses ini memiliki delapan karakteristik berbeda (Kabir, 2016):

1. Berasal dari suatu pertanyaan atau masalah.
2. Membutuhkan artikulasi tujuan yang jelas.
3. Mengikuti rencana prosedur tertentu.

4. Biasanya membagi masalah pokok menjadi sub-sub masalah yang lebih mudah dikelola.
5. Dipandu oleh masalah penelitian, pertanyaan, atau hipotesis tertentu.
6. Menerima asumsi kritis tertentu.
7. Membutuhkan pengumpulan dan interpretasi data dalam upaya menyelesaikan permasalahan tersebut mengawali penelitian tersebut.
8. Berdasarkan sifatnya, bersifat siklus; atau lebih tepatnya, heliks.

Penelitian adalah penyelidikan ilmiah yang bertujuan untuk mempelajari fakta-fakta baru, menguji ide-ide, dll. Ini adalah pengumpulan sistematis, analisis dan interpretasi data untuk menghasilkan pengetahuan baru dan menjawab pertanyaan tertentu atau memecahkan masalah. Karakteristik penelitian (Degu, 2006):

1 Menuntut pernyataan masalah yang jelas

2 Membutuhkan sebuah rencana (bukan berarti "mencari" sesuatu tanpa tujuan dengan harapan Anda akan menemukan solusi)

3 Hal ini didasarkan pada data yang ada, menggunakan temuan positif dan negative

4 Data baru harus dikumpulkan sesuai kebutuhan dan diorganisasikan sedemikian rupa sehingga dapat menjawab pertanyaan penelitian

Penelitian adalah pencarian informasi dan pengetahuan baru secara sistematis. Ini mencakup topik-topik di setiap bidang ilmu pengetahuan dan persepsi tentang ruang lingkup dan aktivitasnya tidak terbatas. Pada umumnya pada penelitian, ide yang awal dituangkan dalam bentuk hipotesa, yang akan dicari kebenarannya melalui pengujian *variable independent* dan *dependent*. Hipotesis adalah pernyataan tentang hubungan antara dua variabel atau lebih, yang merupakan asumsi sebagai jawaban awal atas pertanyaan yang membantu memandu proses penelitian. Pertimbangkan sebuah penelitian yang dirancang untuk menguji hubungan antara kecanduan game online dan hasil tes ujian mahasiswa. Hipotesisnya mungkin: "Penelitian ini dirancang untuk menilai hipotesis bahwa orang yang kecanduan game online akan mendapatkan hasil ujian yang lebih buruk dibandingkan orang yang tidak kurang tidur." Adapun metode penelitian yang umumnya digunakan adalah sebagai berikut:

1. Munculnya pertanyaan mengenai suatu fenomena dan mencari latar belakang dari masalah tersebut.
2. Membentuk hipotesis
3. Menyusun metode penelitian
4. Pengumpulan data
5. Menganalisa data
6. Menarik kesimpulan berdasarkan hasil analisa data dan menghubungkan dengan hipotesa

Ada kesulitan yang dirasakan peneliti dalam menyusun hipotesa, diakibatkan kurangnya pengamatan atau pengetahuan mengenai pertanyaan tersebut, bahkan ada ide pertanyaan tersebut sudah diteliti pihak lain. Perlu dipahami, bagaimana menyusun hipotesa : Kumpulkan sebanyak mungkin pengamatan tentang suatu topik atau masalah. Evaluasi pengamatan ini dan cari kemungkinan penyebab masalahnya. Buatlah daftar kemungkinan penjelasan yang mungkin ingin dielajahi. Setelah mengembangkan beberapa kemungkinan hipotesis, pikirkan cara agar Anda dapat mengkonfirmasi atau menyangkal setiap hipotesis melalui eksperimen. Hal ini dikenal dengan istilah falsifiabilitas (https://www.verywellmind.com/what-is-a-hypothesis-2795239).

Penelitian kuantitatif berhubungan dengan data-data yang dapat diukur, yaitu topik penelitian atau pertanyaan sudah dirumuskan sebelum penelitian dilakukan. Penelitian kuantitatif biasanya berkaitan dengan ide hubungan antara variable baik *dependent* maupun *independent*, dapat berupa deskripsi karakteristik variable dnegan data terukur seperti melihat perbedaan bahkan membandingkan antar kelompok. Penelitian kualitatif berbeda dengan penelitian kuantitatif. Penelitian kualitatif menggunakan data non-numerik, lebih susah mengukurnya, sehingga dilakukan kategori dalam mendeskripsikan data. Karena itu penelitian kualitatif harus terus dikaji dan dirumuskan ulang, terutama dalam menyajikan berbagai perspektif.

Ada beragam tujuan untuk mengembangkan pertanyaan penelitian kualitatif. Pertanyaan-pertanyaan tersebut dapat berfungsi dalam beberapa cara (Barroga & Matanguihan, 2022) , seperti untuk:

1. mengidentifikasi dan mendeskripsikan kondisi yang ada (pertanyaan penelitian kontekstual);
2. mendeskripsikan suatu fenomena (pertanyaan penelitian deskriptif);
3. menilai efektivitas metode, rotocol, teori, atau prosedur yang ada (pertanyaan penelitian evaluasi);
4. mengkaji suatu fenomena atau menganalisis alasan atau hubungan antar subjek atau fenomena (pertanyaan penelitian eksplanatif); atau
5. fokus pada aspek-aspek yang belum diketahui dari suatu topik tertentu (pertanyaan penelitian eksploratif).

Bab 10

Peran Matematika dalam Perkembangan Kecerdasan Buatan di Masa Depan

10.1 Matematika sebagai Landasan Algoritma dalam Kecerdasan Buatan

Algoritma kecerdasan buatan adalah inti dari kecerdasan buatan, membantu sistem AI memecahkan masalah dan membuat keputusan tanpa campur tangan manusia. Algoritma ini juga dapat belajar dari pola dan memproses banyak data. AI telah mengubah industri dan meningkatkan efisiensi banyak aplikasi. Ini termasuk algoritma pembelajaran mesin, seperti pohon keputusan dan jaringan saraf, dan algoritma pemrosesan bahasa alami, seperti analisis sentimen dan pengenalan ucapan.

Untuk mengembangkan algoritma AI yang efektif, Anda harus memiliki pemahaman yang kuat tentang dasar matematika. Untuk membuat model yang dapat menafsirkan data dengan benar dan membuat keputusan yang tepat, konsep seperti statistik, aljabar linear, kalkulus, dan teori probabilitas sangat penting. Insinyur AI mungkin menghadapi kesulitan dalam mengoptimalkan algoritma mereka, yang dapat menyebabkan prediksi yang tidak akurat atau pemecahan masalah yang tidak efisien jika mereka tidak memiliki latar belakang matematika yang solid. Pengembang dapat memastikan bahwa sistem AI mereka berfungsi dengan baik dan berkontribusi pada kemajuan umum kecerdasan buatan dengan memahami prinsip-prinsip matematika ini.

Oleh karena itu, sangat penting bagi para insinyur AI untuk mendedikasikan waktu dan upaya untuk mengasah keterampilan matematika mereka guna menciptakan algoritma AI yang sukses. Dengan memahami prinsip-prinsip dasar matematika, para pengembang dapat dengan yakin menavigasi kompleksitas pengembangan AI dan menghasilkan solusi yang andal dan inovatif. Pada akhirnya, tujuan bab ini adalah untuk menekankan pentingnya kemahiran matematika dalam bidang kecerdasan buatan dan menyoroti bagaimana hal itu secara langsung memengaruhi kualitas dan fungsionalitas sistem AI.

10.2 Statistik dan Probabilitas dalam Pembelajaran Mesin

Praktik pembelajaran mesin kontemporer sering kali melibatkan penerapan algoritma deterministik dan intensif komputasi untuk meminimalkan kriteria kecocokan antara data diskriminan dan sampel secara berulang. Sering kali hanya sedikit yang tertarik menggunakan probabilitas untuk memodelkan ketidakpastian dalam masalah dan statistik untuk mengkarakterisasi perilaku prediktor yang berasal dari data, dengan penekanan pada komputasi dan pengodean. Oleh karena itu, sedikit yang dapat dinyatakan tentang kinerja pada data masa depan, di luar mungkin hitungan kesalahan sederhana pada set pengujian tertentu. Dalam hal ini, pengetahuan yang diberikan oleh metode komputasi deterministik tidak terkait erat dengan dunia nyata dan, khususnya, peristiwa masa depan. Hubungan ini memerlukan pemodelan probabilistik dan inferensi statistik yang ketat serta pemahaman tentang peran komputasi yang tepat dan apresiasi terhadap masalah epistemologis (Braga-Neto & Dougherty, 2020).

Bagaimana seseorang mengetahui bahwa mereka memiliki model prediktif yang sangat terkait dengan dunia nyata dan kejadian di masa mendatang? Dalam klasifikasi, model bersifat prediktif jika tingkat kesalahan klasifikasi kecil. Bagaimana kita bisa mengetahui bahwa kita memiliki pengklasifikasi prediktif? Hanya dengan memperkirakan kesalahan klasifikasi, kita dapat menjawab pertanyaan ini secara praktis dengan menggunakan aturan estimasi kesalahan yang diterapkan pada data pelatihan, serangkaian data uji yang berbeda, atau campuran data pelatihan dan uji. Kriteria validitas harus digunakan untuk mengukur keakuratan aturan estimasi kesalahan. Kriteria validitas yang paling umum adalah kesalahan akar-rata-kuadrat antara estimasi dan kesalahan sebenarnya. Distribusi fitur-label, yaitu distribusi probabilitas gabungan antara vektor fitur dan label, merupakan dasar untuk penetapan validitas klasifikasi. Aturan estimasi kesalahan saat ini, seperti validasi silang atau estimasi kesalahan set pengujian, dapat digunakan. Namun, metode ini hanya akan memberikan angka yang berkaitan dengan data pelatihan dan pengujian sampel yang digunakan, dan tidak akan memiliki hubungan yang dapat diukur dengan kinerja pengklasifikasi di masa mendatang. Mungkin ada yang mengatakan bahwa mereka mengharapkan proporsi kesalahan pada aplikasi berikutnya akan sesuai dengan estimasi; namun, pernyataan ini tidak dapat diukur dalam hal prediksi versus observasi, sehingga tidak ilmiah. (Braga-Neto & Dougherty, 2020).

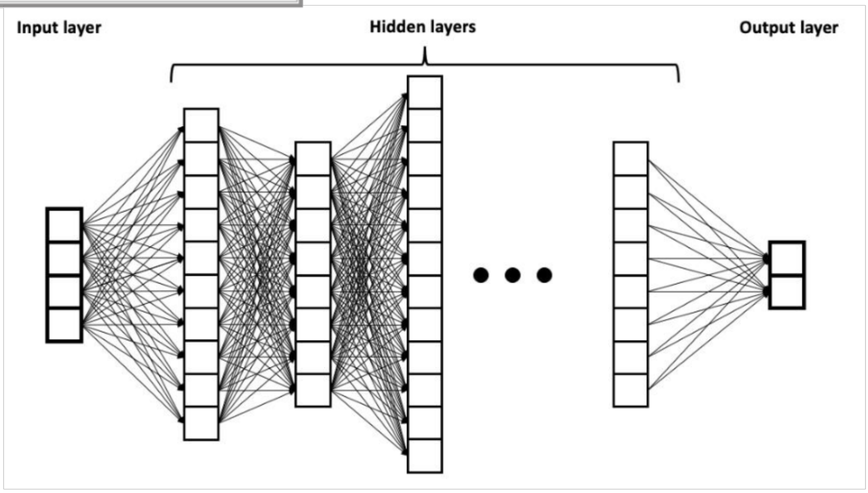
10.3 Aljabar Linier dalam Jaringan Syaraf Tiruan

Jaringan saraf telah menjadi landasan kecerdasan buatan modern, yang merevolusi berbagai bidang termasuk visi komputer, pemrosesan bahasa alami, dan pembelajaran penguatan. Jaringan ini, yang terinspirasi oleh struktur dan fungsi otak manusia, terdiri dari lapisan neuron yang saling terhubung yang mampu mempelajari pola dan hubungan kompleks dari data (Divya, 2024).

Meskipun jaringan saraf sangat baik dalam banyak tugas, memahami cara kerja internalnya dan mengoptimalkan kinerjanya masih sulit. Dengan menggunakan metode aljabar linier dalam jaringan saraf, inti dari teori jaringan saraf terletak pada aljabar linier, cabang matematika yang berkaitan dengan ruang vektor dan transformasi linier. Ini memberikan kerangka kerja yang kuat untuk menganalisis perilaku jaringan saraf, menafsirkan keputusannya, dan meningkatkan kemampuan mereka. Kita dapat menggambarkan struktur dan dinamika jaringan saraf dengan matriks dan vektor. (Divya, 2024).

Jaringan saraf terdiri dari neuron dan tepi dengan neuron yang biasanya tersusun dalam lapisan dan tepi terarah yang menghubungkan neuron dari satu lapisan ke lapisan berikutnya. Kita dapat menganggap neuron sebagai variabel dengan nilai yang ditetapkan yang kita hitung melalui "propagasi maju" yang akan didefinisikan nanti. Mereka juga disebut unit aktivasi. Kita dapat menganggap tepi sebagai variabel yang nilainya menunjukkan seberapa kuat satu neuron memengaruhi neuron lain. Bobot akan berfungsi sebagai jenis skalar untuk neuron yang diterimanya.

Nilai tepi ini akan digunakan untuk mendefinisikan fungsi yang mengambil nilai neuron dalam satu lapisan dan menggunakannya untuk mendefinisikan nilai di lapisan berikutnya. Akan ada sejumlah lapisan yang telah ditentukan sebelumnya dalam jaringan dan unit aktivasi di lapisan terakhir akan menandakan sesuatu tentang data yang dimasukkan ke dalam jaringan saraf. Ilustrasi Jaringan Saraf Tiruan Dalam (*Deep Artificial Neural Network*) bisa dilihat pada Gambar 10.1.



Gambar 10.1 Struktur Jaringan Saraf Tiruan Dalam (*Deep Artificial Neural Network*)
Sumber: Wikipedia

10.4 Kalkulus pada Algoritma Optimisasi

Kalkulus matematika adalah cabang matematika yang mempelajari laju perubahan dan akumulasi kuantitas. Dalam bidang algoritma pengoptimalan kecerdasan buatan, kalkulus memainkan peran penting dalam menentukan solusi optimal dengan menganalisis dan mengoptimalkan fungsi yang kompleks. Dengan memanfaatkan kalkulus, algoritma dapat menavigasi data dalam jumlah besar secara efisien untuk menemukan solusi yang paling efisien dan efektif untuk berbagai masalah. Integrasi kalkulus dalam algoritma AI ini memungkinkan prediksi yang lebih akurat, kecepatan pemrosesan yang lebih cepat, dan pada akhirnya, sistem AI yang lebih canggih dan canggih.

Secara keseluruhan, penerapan kalkulus dalam algoritma pengoptimalan memainkan peran penting dalam meningkatkan efisiensi dan efektivitas sistem kecerdasan buatan. Dengan memanfaatkan turunan untuk menemukan titik minimum atau maksimum, algoritma AI dapat disetel dengan baik untuk mencapai kinerja yang optimal. Selain itu, penggunaan integral dalam menghitung area di bawah kurva untuk pengoptimalan memungkinkan proses pengambilan keputusan yang lebih akurat dan tepat dalam teknologi AI. Oleh karena itu, jelas bahwa kalkulus tidak hanya penting dalam membentuk masa depan kecerdasan buatan tetapi juga penting dalam mendorong kemajuan dan terobosan di bidang tersebut.

Salah satu contoh kalkulus dalam algoritma pengoptimalan adalah penggunaan penurunan gradien dalam melatih jaringan saraf. Dengan menghitung gradien fungsi kerugian sehubungan dengan bobot jaringan, penyesuaian dapat dilakukan untuk meminimalkan kesalahan dan meningkatkan kinerja model. Contoh lain adalah dalam algoritma pembelajaran penguatan, —ketika kalkulus digunakan untuk menghitung kebijakan optimal bagi agen untuk memaksimalkan imbalannya dari waktu ke waktu. Aplikasi dunia nyata ini menunjukkan dampak signifikan kalkulus dalam meningkatkan teknologi AI dan mendorong batasan dari apa yang mungkin.

Kalkulus memainkan peran penting dalam meningkatkan efisiensi dan keakuratan algoritma pengoptimalan dengan menyediakan kerangka kerja untuk memahami bagaimana perubahan kecil dalam parameter dapat memengaruhi kinerja keseluruhan suatu sistem. Dengan memanfaatkan kalkulus, algoritma pengoptimalan dapat menavigasi ruang kompleks berdimensi tinggi secara lebih efektif untuk menemukan solusi optimal. Hal ini tidak hanya mempercepat proses pembelajaran tetapi juga memungkinkan penyesuaian yang lebih tepat untuk dilakukan, yang mengarah pada kinerja dan hasil yang lebih baik dalam sistem AI. Pada akhirnya, integrasi kalkulus dalam algoritma pengoptimalan sangat penting untuk mendorong batasan dari apa yang dapat dicapai oleh teknologi AI.

10.5 Pemodelan Matematika untuk Implementasi Keceredasan Buatan

Pemodelan matematika sangat penting untuk memajukan teknologi AI karena menyediakan kerangka kerja untuk memahami dan mengoptimalkan sistem yang kompleks. Dengan menggunakan model matematika, peneliti dapat menguji berbagai skenario, mengevaluasi kinerja algoritma, dan menyempurnakan sistem AI untuk hasil yang optimal. Selain itu, pemodelan matematika membantu menjembatani kesenjangan antara konsep teoritis dan aplikasi praktis, yang memungkinkan teknologi AI menjadi lebih efisien, akurat, dan mudah beradaptasi dengan lingkungan yang berubah. Intinya, integrasi pemodelan matematika dalam aplikasi AI sangat penting untuk mendorong batasan kecerdasan buatan dan membuka potensi penuhnya.

Dengan menggabungkan pemodelan matematika ke dalam sistem AI, peneliti dan pengembang dapat memperoleh pemahaman yang lebih mendalam tentang algoritme yang kompleks dan mengoptimalkan kinerjanya dalam situasi dunia nyata. Hal ini memungkinkan kemajuan dalam pembelajaran mesin, visi komputer, pemrosesan bahasa alami, dan domain AI lainnya, yang pada akhirnya mengarah pada inovasi inovatif yang dapat merevolusi industri dan meningkatkan kualitas hidup individu di seluruh dunia. Sebagai kesimpulan, penggunaan pemodelan matematika sangat penting untuk mendorong evolusi kecerdasan buatan dan mendorongnya menuju masa depan di mana mesin cerdas benar-benar dapat meniru kemampuan kognitif manusia.

10.6 Analisis Data dan Pengenalan Pola

Matematika memainkan peran penting dalam pengenalan pola dan analisis data karena menyediakan berbagai alat dan teknik yang diperlukan untuk memahami kumpulan data yang besar dan kompleks, dari analisis statistik hingga aljabar linear.

Matematika juga membantu kita memahami pola, tren, dan hubungan dalam data yang sulit diidentifikasi. Sangat sulit untuk mendapatkan informasi penting dari volume data yang besar yang dibuat setiap hari jika Anda tidak memiliki dasar matematika yang solid.

Pengembangan algoritma pembelajaran mesin, yang sangat bergantung pada teori probabilitas, pengoptimalan, dan lainnya dari matematika. Selain itu, peneliti sekarang dapat memproses dan menganalisis data dengan lebih baik berkat kemajuan dalam matematika komputasional, yang menghasilkan hasil yang lebih cepat dan akurat. Peran matematika dalam analisis data dan pengenalan pola akan semakin penting seiring dengan kemajuan teknologi.

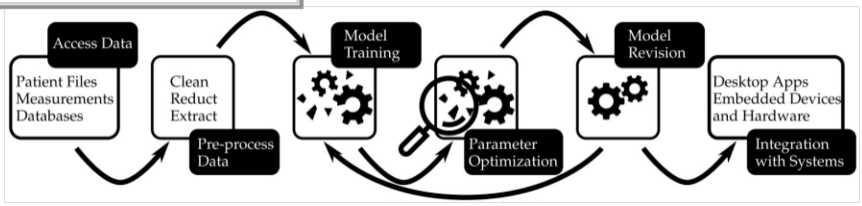
Di masa mendatang, akan dilihat model matematika yang lebih canggih dikembangkan untuk menangani sejumlah besar data yang dihasilkan. Selain meningkatkan akurasi prediksi, model ini akan membantu mengidentifikasi pola dan hubungan yang tersembunyi dalam data yang dikumpulkan. Matematika akan terus menjadi yang terdepan dalam kemajuan analisis data dan pengenalan pola, memainkan peran penting dalam membentuk masa depan teknologi dan proses pengambilan keputusan. Ini karena bidang analisis data terus berkembang.

Fokus pada kebutuhan penelitian dan inovasi matematika yang berkelanjutan untuk mengimbangi kompleksitas analisis data dan tantangan pengenalan pola yang terus berkembang. Secara keseluruhan, bidang ini memiliki masa depan yang cerah dalam matematika, karena ada banyak peluang untuk bekerja sama dan berkembang. Matematikawan dapat terus memainkan peran penting dalam membuka potensi analisis data dan pengenalan pola di berbagai industri dan bidang dengan tetap menjadi yang terdepan dan mengadopsi pendekatan interdisipliner.

10.7 Pemodelan Prediktif dan Peramalan

Pemodelan prediktif melibatkan penggunaan metode matematika atau komputasi untuk membuat model yang dapat memperkirakan hasil di masa mendatang. Sementara model berbasis persamaan digunakan untuk pendekatan pertama, teknik simulasi diperlukan untuk pendekatan kedua. Pemodelan prediktif memiliki banyak aplikasi dalam bidang kedokteran, seperti pengambilan keputusan klinis dan uji klinis; namun, potensinya sebagian besar belum dimanfaatkan di bidang ini karena berbagai tantangan. Penerapan teknik ini pada domain medis sangat menantang karena berkaitan dengan sifat dinamis dari disiplin ilmu ini dan kompleksitas populasi pasien yang dirawat dalam pengaturan perawatan kesehatan modern. Lebih jauh, pengembangan dan penerapan model prediktif yang efektif memerlukan pemahaman mendalam tentang data yang digunakan beserta sumber daya yang memadai untuk mendukung pengembangan dan penerapan model (Toma, 2023).

Pemodelan prediktif melibatkan penggunaan metode matematika dan komputasi untuk memperkirakan kejadian atau hasil di masa mendatang. Metode ini mencakup berbagai teknik, seperti analisis regresi, pohon keputusan, hutan acak, jaringan saraf, dan mesin vektor pendukung. Algoritma digunakan oleh metode ini untuk analisis data dan pembuatan model yang dapat memprediksi hasil berdasarkan pola dan hubungan yang ditemukan dalam data. Contoh alur pengembangan model prediktif bisa dilihat pada Gambar 10.2.



Gambar 10.2 Contoh Alur Pengembangan Model Prediktif
Sumber: (Toma, 2023)

10.8 Optimasi dan Pengambilan Keputusan

Teknik optimasi menjadi kompas yang memandu para eksekutif melalui labirin pengambilan keputusan. Kendala, ketidakpastian, dan banyak variabel tidak lagi menjadi batu sandungan; mereka menjadi titik data dalam model matematika yang dirancang untuk mengidentifikasi keputusan yang paling menguntungkan yang selaras dengan tujuan organisasi. Ini tentang mengubah kemungkinan abstrak menjadi pilihan konkret yang didukung oleh wawasan berbasis data. Namun, kekuatan ekonomi matematika melampaui proses pengambilan keputusan langsung. Ini adalah alat visioner yang memungkinkan perusahaan untuk mengintip masa depan. Dengan memanfaatkan teori ekonomi, metode statistik, dan model canggih, para pengambil keputusan dapat memperkirakan hasil dan mengantisipasi tantangan. Ini seperti memiliki bola kristal yang mengungkapkan konsekuensi potensial, yang memungkinkan organisasi untuk secara proaktif menyusun strategi dan beradaptasi. Ekonomi matematika menjadi orkestrasi diam di balik setiap langkah strategis, mengubah pengambilan keputusan dari seni menjadi sains. Ini bukan hanya tentang membuat pilihan; ini tentang membuat pilihan yang tepat, yaitu pilihan yang didasarkan pada data, dipandu oleh optimasi, dan diperkuat dengan pandangan ke depan untuk menavigasi tarian rumit lanskap bisnis.

Berbagai peristiwa dalam perekonomian saling berhubungan satu sama lain, sehingga akan saling mempengaruhi peristiwa tersebut. Misalnya; Jika pendapatan naik, pola konsumsi juga naik, harga produk naik, maka pola permintaan akan turun. Berbagai peristiwa ekonomi tersebut dapat dinyatakan dengan perubahan nilai variabel. Variabel adalah sesuatu yang nilainya berubah-ubah, misalnya biaya, harga, kuantitas, pendapatan, suku bunga dan sebagainya. Matematika memegang peranan penting dalam menganalisis berbagai peristiwa ekonomi. Dengan menggunakan matematika sebagai alat analisis, maka dapat diperoleh hasil analisis yang konkret, mudah digunakan sebagai dasar perencanaan, alat kontrol, dan dasar untuk melakukan evaluasi. Banyak sekali penggunaan matematika dalam analisis kuantitatif, yaitu analisis yang memberikan hasil dalam bentuk angka-angka.

Dasar dari setiap bisnis yang sukses adalah matematika, khususnya matematika bisnis. Bidang ini menerapkan konsep matematika pada situasi bisnis praktis, dengan fokus pada area utama seperti laba, rugi, bunga, dan rumus keuangan. Alat-alat ini memberdayakan bisnis untuk mengelola tugas-tugas seperti menghitung margin laba, mengoptimalkan inventaris, dan menetapkan tarif pajak secara efektif. Matematika bisnis terkait erat dengan statistik, yang memberikan wawasan berharga untuk memecahkan tantangan bisnis. Ketika uang atau barang dipertukarkan, memahami laba, rugi, dan persentase serta diskon terkait menjadi sangat penting. Meskipun matematika murni mungkin bukan persyaratan utama, penalaran matematika yang kuat dan beberapa rumus utama sangat penting untuk menavigasi lanskap keuangan bisnis apa pun.

Perhitungan matematika bisnis merupakan elemen integral dalam seluruh proses pengambilan keputusan bisnis. Dari analisis keuangan hingga manajemen risiko, perhitungan matematika memberikan landasan yang kuat bagi para pemimpin bisnis untuk membuat keputusan yang akurat, efektif, dan berorientasi pada tujuan. Seiring kemajuan teknologi, penerapan algoritma dan analisis data semakin memperkuat peran matematika dalam membantu perusahaan menghadapi tantangan dan memanfaatkan peluang untuk pertumbuhan yang berkelanjutan (Ardanta et al., 2024).

DAFTAR PUSTAKA

Adam, J., 2011. Mathematics in Nature: Modeling Patterns in the Natural World. Princeton University Press.

Ahmad, Abdan Matin. 2021. Konsep-Konsep Dasar Matematika dalam Ekonomi. MEGA: Jurnal Pendidikan Matematika, 2(1).

Ali, S. et al. (2023) 'Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence', Information Fusion, 99(March), p. 101805. Available at: https://doi.org/10.1016/j.inffus.2023.101805.

Allgöwer, F. & Zheng, A. (eds.), 2012. Nonlinear Model Predictive Control. Vol. 26. Birkhäuser.

Amatriain, X., 2013. Big & personal: data and models behind Netflix recommendations. Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications. DOI: 10.1145/2501221.2501222.

Andi, T., Andriyani, W. and Putranto, B.P.D. (2023) 'Deep Learning Architecture for Stock Price Prediction', Journal of Intelligent Software Systems, 7(2). Available at: https://doi.org/10.3390/drones7020078.

Andriyani, W. et al. (2024) Matematika Lanjut. Edited by Y.S. Nugraha and Q. Nurlaila. Available at: https://toharmedia.co.id/product/matematika-lanjut/.

Anshori, M., Haris, M. S. and Teja Kusuma, W. (2023) 'Penerapan Backpropagation Neural Network (BPNN) Untuk Prediksi Kecanduan Smartphone Pada Remaja', Cices, 9(2), pp. 192–202. doi: 10.33050/cices.v9i2.2701.

Anton, H., 1997. Aljabar Linear Elementer (terjemahan). Penerbit Erlangga, Jakarta.

Anton, H., Rorres, C., 2014. Elementary Linear Algebra: Application Version, 11th Edition. John Willey and Sons, New York.

Anwarudin, A. et al. (2022) 'The Prediction on the Students' Graduation Timeliness Using Naive Bayes Classification and K-Nearest Neighbor', Journal of Intelligent Software Systems, 1(1), p. 75. Available at: https://doi.org/10.26798/jiss.v1i1.597.

Ardanta, M. A., Fauzi, A., Patimah, P., Khadijah, F., & Todo, Y. (2024). Optimization of Business Decision Accuracy through the Application of Mathematical Economics. 2(4), 866–885.

Argisitawan, A. et al. (2022) 'Determining the Target of Independent Graduation for Beneficiary Families of the Hopeful Family Program', Journal of Intelligent Software Systems, 1(1), p. 55. Available at: https://doi.org/10.26798/jiss.v1i1.601.

Arora, S. and Barak, B., 2009. Computational Complexity: A Modern Approach. Cambridge: Cambridge University Press. ISBN-13 978-0-521-42426-4 (print), ISBN-13 978-0-511-53075-3

Barroga, Edward dan Glafera Janet Metanguihan. 2022. A Practical Guide to Writing Quantitative and Qualitative Research Question and Hypotheses in Scholarly Articles. J. Korean Med. Sci. 37(16).

Berger, J. O. (2013). Statistical decision theory and Bayesian analysis. Springer Science & Business Media.

Berman, G., Chubb, J. and Williams, K. (2023) 'The Use of AI in Science,Technology, Engineering,and Maths - Key themes identified in a breadth-focused literature review', (March).

Boateng, E.Y., Otoo, J. and Abaye, D.A. (2020) 'Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review', Journal of Data Analysis and Information Processing, 8(4), pp. 341-357. DOI: 10.4236/jdaip.2020.84020.

Braga-Neto, U. M., & Dougherty, E. R. (2020). Machine Learning Requires Probability and Statistics [Perspectives]. IEEE Signal Processing Magazine, 37(4), 118–122. https://doi.org/10.1109/MSP.2020.2985385

Brink, H., Richards, J. & Fetherolf, M., 2016. Real-World Machine Learning. Simon and Schuster.

Broekhuizen, T. et al. (2023) 'AI for managing open innovation: Opportunities, challenges, and a research agenda', Journal of Business Research, 167(July), p. 114196. Available at: https://doi.org/10.1016/j.jbusres.2023.114196.

Bulut, A.S. and Kuzu, O. (2023) 'Mathematical Reasoning Skills as a Predictive of Number Sense', International Journal of Progressive Education, 19(5), pp. 172–185. Available at: https://doi.org/10.29329/ijpe.2023.603.11.

Cahyono, B. (2013). Penggunaan Software Matrix Laboratory (Matlab) dalam Pembelajaran Aljabar Linear. Jurnal PHENOMENON, 1(1), 45–60.

Čajić, E. (2024) 'Application of mathematics in artificial intelligence', in.

Carlo, C. M. (2004). Markov chain monte carlo and gibbs sampling. Lecture notes for EEB, 581(540), 3.

Chatterjee, R. (2020) 'Fundamental Concepts of Artificial Intelligence and Its Applications', Journal of Mathematical Problems, Equations and Statistics, 1(2), pp. 13–24. Available at: www.mathematicaljournal.com.

Che, C. et al. (2024) 'Intelligent robotic control system based on computer vision technology', Applied and Computational Engineering, 64(1), pp. 142–147. Available at: https://doi.org/10.54254/2755-2721/64/20241373.

Dangeti, P. (2017). Statistics for machine learning. Packt Publishing Ltd.

Degu, Getu and Tegbar Yigzaw. 2006. Lecture Notes: Research Methodoly. University of Gondar.

Deolika, A., Kusrini and Luthfi, E. T. (2019) 'Analisis Pembobotan Kata Pada Klasifikasi Text Mining', Jurnal Teknologi Informasi, 3(2), pp. 179–184. doi: 10.36294/jurti.v3i2.1077.

Divya, R. (2024). Linear Algebraic Methods In Neural Networks. International Journal of Engineering Research & Technology (IJERT), 12(01).

Du, D.-Z. & Ko, K.-I., 2014. Theory of Computational Complexity. 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc. ISBN 978-1-118-30608-6.

Eriana, E.S. and Zein, D.A. (2023) 'Artificial Intelligence', Angewandte Chemie International Edition, 6(11), 951–952., p. 26.

Ernest, P. 1991. The philosophy of mathematics education. London: The Falmer Press.

Fatimah, D.D.S. and Rahmawati, E. (2021) 'Penggunaan Metode Decision Tree dalam Rancang Bangun Sistem Prediksi untuk Kelulusan Mahasiswa', Jurnal Algoritma, 18(2), pp. 553-561.

Fauzan, A., & Mahara, D. O. (2020). Implementasi Software R pada Mata Kuliah Kalkulus I untuk Menunjang Kemampuan Pemrograman Mahasiswa. Refleksi Pembelajaran Inovatif, 2(1), 279–286.

Fauzi, A. et al. (2022) 'Price Intelligence Using K-Means Clustering and Linear Regression, Case Study of Store Dk Nutritionindo at Tokopedia', Journal of Intelligent Software Systems, 1(1), p. 27. Available at: https://doi.org/10.26798/jiss.v1i1.602.

Ferreira, D. (2022) 'Mathematical Language Processing: Deep Learning Representations and Inference over Mathmatical Text'.

Friedrich, S. et al. (2022) 'Is there a role for statistics in artificial intelligence?', Advances in Data Analysis and Classification, 16(4), pp. 823–846. Available at: https://doi.org/10.1007/s11634-021-00455-6.

Ghosh, A. (2022) 'Prediction of Diabetes Using Random Forest And xgboost Classifiers', International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), 12(1), pp. 19–28.

Hidayat, R. and Astuti, T. (2020) 'Diagnosis Preeklamsia pada Ibu Hamil Berdasarkan Algoritme K- Nearest Neighbour', 14(2).

https://byjus.com/maths/population-and-sample/

https://www.verywellmind.com/what-is-a-hypothesis-2795239

Hu, L.Y. et al. (2016) 'The distance function effect on k-nearest neighbor classification for medical datasets', SpringerPlus, 5, pp. 1-9.

Ilias, S. et al. (2016) 'Feature Extraction of Autism Gait Data Using Principal Component Analysis and Linear Discriminant Analysis', 2016 IEEE Industrial Electronics and Applications Conference (IEACon), pp. 275–279. doi: 10.1109/IEACON.2016.8067391.

Imrona, M., 2013. Aljabar Linear Dasar, Edisi Kedua. Penerbit Erlangga, Jakarta.

Iriyanta, K., Putranto, B.P.D. and Andriyani, W. (2023) 'Iot Based Soil Moisture Monitoring and Soil Moisture Prediction Using Linear Regression (Case Study of Vinca Plants)', Journal of Intelligent Software Systems, 2(1), p. 1. Available at: https://doi.org/10.26798/jiss.v2i1.929.

Irmayanti, Henra, K., Asnita, A. U., Munaji, Riaddin, D., Fitriani, Junaedi, Resi, F. B., Setiawan, J., & Dahlan, T. (2021). Teori dan Aplikasi Kalkulus Dasar.

Jia, F., Mingyang, L. and Yueen, L. (2023) 'Investigating the Fundamental Mathematical Logic Relationship and Driving Mode in Artificial Intelligence', pp. 359–364. Available at: https://doi.org/10.32474/CTCSA.2023.03.000155.

Jinwei Han and Michaline Kamber (2006) Data Mining, south asia edition : Concept and Technology. Morgan Kaufmann.

Kartono, 2003. Aplikasi Matematika Dalam Perencanaan dan Pengendalian di Bidang Industri. Jurnal Matematika dan Komputer. 6(1), pp. 49-58.

Kufel, J. et al. (2023) 'What Is Machine Learning, Artificial Neural Networks and Deep Learning?—Examples of Practical Applications in Medicine', Diagnostics, 13(15). Available at: https://doi.org/10.3390/diagnostics13152582.

Kuhn, M. & Johnson, K., 2016. Applied Predictive Modeling. Springer. DOI: 10.1007/978-1-4614-6849-3.

Kumar Yadav, D. (2021) 'Three Magical Words of Mathematics', 9(8), pp. 265–274. Available at: www.ijcrt.org.

Kutyniok, G. (2023) 'The mathematics of artificial intelligence', pp. 5118–5139. Available at: https://doi.org/10.4171/icm2022/141.

Lámer, J., Cymbalak, D. and Jakab, F. (2013) 'Computer vision based object recognition principles in education', ICETA

2013 - 11th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings, (October 2013), pp. 253–257. Available at: https://doi.org/10.1109/iceta.2013.6674439.

Lantz, B., 2019. Machine Learning with R: Expert Techniques for Predictive Modeling. Packt Publishing Ltd.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Leon, S. J., 2001. Aljabar Linear dan Aplikasinya (terjemahan). Penerbit Erlangga, Jakarta.

Lishania, I., Goejantoro, R. and Nasution, Y. N. (2019) 'Perbandingan Klasifikasi Metode Naive Bayes dan Metode Decision Tree Algoritma (J48) pada Pasien Penderita Penyakit Stroke di RSUD Abdul Wahab Sjahranie Samarinda', Jurnal Eksponensial, 10(2), pp. 135–142. Available at: http://jurnal.fmipa.unmul.ac.id/index.php/exponensial/article/view/571.

Lonang, S. and Normawati, D. (2022) 'Klasifikasi Status Stunting Pada Balita Menggunakan K-Nearest Neighbor Dengan Feature Selection Backward Elimination', Jurnal Media Informatika Budidarma, 6(1), p. 49. Available at: https://doi.org/10.30865/mib.v6i1.3312.

Marsudi & Marjono, 2012. Aljabar Linear. Universitas Brawijaya Press, Malang.

Mathworks. 2024. Diambil dari https://www.mathworks.com/discovery/data-visualization.html pada tanggal 06 Agustus 2024

Mazur, J.E. (2006) 'Mathematical Models and the Experimental Analysis of Behavior', Journal of the Experimental

Analysis of Behavior, 85(2), pp. 275–291. Available at: https://doi.org/10.1901/jeab.2006.65-05.

Mehdiyev, N., Majlatow, M. and Fettke, P. (2024) 'Quantifying and explaining machine learning uncertainty in predictive process monitoring: an operations research perspective', Annals of Operations Research [Preprint]. Available at: https://doi.org/10.1007/s10479-024-05943-4.

Miklós, I., 2019. Computational Complexity of Counting and Sampling. Boca Raton, FL: CRC Press, Taylor & Francis Group. ISBN-13 978-1-138-03557-7 (Paperback), ISBN-13 978-1-138-07083-7.

Miles, J., & Shevlin, M. (2000). Applying regression and correlation: A guide for students and researchers.

Mohana, M. (2024) 'Natural Language Processing Presented by', (April). Available at: https://doi.org/10.13140/RG.2.2.13534.04169.

Muhrial, H. et al. (2022) 'Data Warehouse to Support the Decision Using Vikor Method', Journal of Intelligent Software Systems, 1(2), p. 153. Available at: https://doi.org/10.26798/jiss.v1i2.767.

Normawati, D. (2014) Kajian Teknik-teknik Data Mining untuk Diagnosis Penyakit Jantung Koroner. Proceedings of Conference on Information Technology and Electrical Engineering. Available at: https://www.researchgate.net/profile/Hanifah_Rahmi_Fajrin/publication/331772158_ISI_Proceedings_CITEE_2014/links/5c8b5e9f92851c1df941bb98/ISI-Proceedings-CITEE-2014.pdf#page=96.

Nst, Zulaini Masruro, Miranda Meylissa Siadari, Intan Julia Sari Saragih, Ika Okta Kirana, Zulia Almaida Siregar. 2023. Penerapan Matematika Algoritma dalam Bidang

Komputer. FARABI: Jurnal Matematika dan Pendidikan Matematika. 6(2), pp. 180-191.

Pareek, P., et al., 2022. Predicting music popularity using machine learning algorithm and music metrics available in Spotify. J. Dev. Econ. Manag. Res. Stud. (JDMS), 9, pp.10-19.

Patil, M. B., Jadhav., S., Talekar, S., & Bag, V. V. (2023). The role of Mathematics in Machine Learning. Journal of Data Acquisition and Processing, 38(1), 1062–1073. https://doi.org/10.5281/zenodo.7702430

Patro, S. G. K. and Kumar, K. (no date) 'Normalization : A Preprocessing Stage', International Advanced Research Journal in Science, Engineering and Technology. doi: 10.17148/IARJSET.2015.2305.

Peter, M., Aldo, D. A., Cheng, F., & Ong, S. (2019). Mathematics For Machine Learning. https://mml-book.com.

Pratomo, C.H. and Andriyani, W. (2023) 'Mushroom Image Classification', Journal of Intelligent Software Systems, 2(1), pp. 24–26.

Putra, R.F. et al. (2024) Algoritma Pembelajaran Mesin (Dasar, Teknik, dan Aplikasi). Available at: www.buku.sonpedia.com.

Raftery, A. E., Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1995). Hypothesis testing and model. Markov chain Monte Carlo in practice, 1, 165-87.

Roh Bintang Jaya, M. et al. (2024) 'Dynamic Bitrate Adjustment in Web-based Video Streaming Applications Using HTTP Live Streaming (HLS)', Journal of Intelligent Software Systems, 3(1), p. 13. Available at: https://doi.org/10.26798/jiss.v3i1.1344.

Ross, S. M. (2014). Introduction to probability models. Academic press.

Rubinstein, R. Y., & Kroese, D. P. (2016). Simulation and the Monte Carlo method. John Wiley & Sons.

Sandag, G.A. & Manueke, A.M., 2020. Predictive models for popularity of solo and group singers in Spotify using decision tree. 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS). DOI:10.1109/icoris50180.2020.9320838.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3), 160.

Sarker, I.H. (2021) 'Machine Learning: Algorithms, Real-World Applications and Research Directions', SN Computer Science, 2(3), pp. 1–21. Available at: https://doi.org/10.1007/s42979-021-00592-x.

Sarker, I.H., 2021. Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective. SN Computer Science, 2(5), p.377. DOI: 10.1007/s42979-021-00765-8.

Semadi, I.N.O., Kristomo, D. and Purnomosidi, B. (2023) 'A Decision Model to Support the Selection of SENKOM Personnel Using the Profile Matching Method with the Capability of Cyber Security', Journal of Intelligent Software Systems, 2(2), p. 1. Available at: https://doi.org/10.26798/jiss.v2i2.1135.

Seoni, S. et al. (2023) 'Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023)', Computers in Biology and Medicine, 165(August), p. 107441. Available at: https://doi.org/10.1016/j.compbiomed.2023.107441.

Shapiro, L. and Stockman, G. (2000) Cmputer Vision, The University of Washington and Michigan State University. Available at:

http://repositorio.unan.edu.ni/2986/1/5624.pdf%0Ahttp://fiskal.kemenkeu.go.id/ejournal%0Ahttp://dx.doi.org/10.1016/j.cirp.2016.06.001%0Ahttp://dx.doi.org/10.1016/j.powtec.2016.12.055%0Ahttps://doi.org/10.1016/j.ijfatigue.2019.02.006%0Ahttps://doi.org/10.1.

Shokrzade, A., Ramezani, M., Akhlaghian Tab, F. and Abdulla Mohammad, M. (2021) 'A novel extreme learning machine based kNN classification method for dealing with big data', Expert Systems with Applications, 183, 115293. DOI:10.1016/j.eswa.2021.115293.

Sibarani, M., 2013. Aljabar Linier, Edisi Kedua. Penerbit RajaGrafindo Persada, Jakarta.

Sihombing, R. E., Rachmatin, D., & Dahlan, J. A. (2019). Program Aplikasi Bahasa R Untuk Pengelompokan Objek Menggunakan Metode K-Medoids Clustering. Jurnal EurekaMatika, 7(1).

Soori, M., Arezoo, B. and Dastres, R. (2023) 'Artificial intelligence, machine learning and deep learning in advanced robotics, a review', Cognitive Robotics, 3(March), pp. 54–70. Available at: https://doi.org/10.1016/j.cogr.2023.04.001.

Stigler, S. M. (2002). Statistics on the table: The history of statistical concepts and methods. Harvard University Press.

Sumarlin, T. (2023). Statistik Probabilitas. Penerbit Yayasan Prima Agus Teknik, 1-110.

Suthaharan, S. (2016) 'Decision Tree Learning', in Integrated Series in Information Systems, pp. 237–269. DOI:10.1007/978-1-4899-7641-3_10.

Suyanto (2017) Data Mining Untuk Klasifikasi dan Klasterisasi Data. Penerbit Informatika.

Syafitri Hidayatul AA, Yuita Arum S, A. A. (2018) 'Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes', Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(9), pp. 2546–2554.

Taherdoost, H. and Mitra Madanchian (2023) 'Decision Making: Models, Processes, Techniques', Cloud Computing and Data Science, (August), pp. 1–14. Available at: https://doi.org/10.37256/ccds.5120233284.

Toma, M. (2023). Predictive Modeling in Medicine. 590–601.

Triaji, B. et al. (2022) 'Building a Knowledge Graph on Video Transcript Text Data', Journal of Intelligent Software Systems, 1(1), p. 1. Available at: https://doi.org/10.26798/jiss.v1i1.585.

Tripathy, A., Agrawal, A. and Rath, S. K. (2015) 'Classification of Sentimental Reviews Using Machine Learning Techniques', International Conference on Recent Trends in Computing 2015, 57, pp. 821–829. doi: 10.1016/j.procs.2015.07.523.

Tsang, A., Smith, K. et al. (2009) 'Decision trees for uncertain data', IEEE Transactions on Knowledge and Data Engineering, 23(1), pp. 64-78. DOI: 10.1109/TKDE.2009.175.

Tymoczko, T. 1986. New directions in the philosopy of mathematics. Boston: Kirkhauser.

Wati, R.A., Irsyad, H. and Rivan, M.E. Al (2020) 'Klasifikasi Pneumonia Menggunakan Metode Support Vector Machine', Jurnal Algoritme, 1(1), pp. 21–32.

Weinan, E. et al. (2020) 'Towards a Mathematical Understanding of Neural Network-Based Machine Learning: What We Know and What We Don't', CSIAM Transactions on

Applied Mathematics, 1(4), pp. 561–615. Available at: https://doi.org/10.4208/csiam-am.SO-2020-0002.

Wibowo, H. W., Hasbi, M. and Anshori, M. (2024) 'Use Discriminant Analysis to Identify Eroticism-Related Terms in The Lyrics of Dangdut Songs', 1(1), pp. 17–22.

Widayanti, E. (2017). Pemodelan Matematika Dalam Optimalisasi Produk Pengolahan Susu Segar. 4(2).

Wijayanti, I.E., Wahyuni, S., & Susanti, Y., 2018. Dasar-dasar Aljabar Linear dan Penggunaannya dalam Berbagai Bidang. UGM Press, Yogyakarta.

Witten, I.H. et al. (2016) Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Wu, T. et al. (2022) 'Intrusion detection system combined enhanced random forest with SMOTE algorithm', Eurasip Journal on Advances in Signal Processing, 2022(1). doi: 10.1186/s13634-022-00871-6.

Wu, Y. (2014). Principal Component Analysis and Linear Discriminant Analysis. Electrical Engineering and Computer Science, Northwestern University, Evanston, wykład.

Xie, Y. (2015) 'A Fault Diagnosis Approach Using SVM with Data Dimension Reduction by PCA and LDA Method', 2015 Chinese Automation Congress (CAC), pp. 869–874. doi: 10.1109/CAC.2015.7382620.

Ying, S. et al. (2021) 'An improved KNN-based efficient log anomaly detection method with automatically labeled samples', ACM Transactions on Knowledge Discovery from Data (TKDD), 15(3), pp. 1-22.

Zhang, Z. (2016) 'Introduction to machine learning: k-nearest neighbors', Annals of Translational Medicine, 4(11). DOI: 10.21037/atm.2016.03.37.

Zhongwen, Z. and Huanghuang, G. (2017) 'Visualization Study of High-Dimensional Data Classification Based on PCA-SVM', 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), pp. 346–349. doi: 10.1109/DSC.2017.57.

Zhou, H., Wang, X. and Zhu, R. (2022) 'Feature selection based on mutual information with correlation coefficient', Applied Intelligence, 52(5), pp. 5457–5474. doi: 10.1007/s10489-021-02524-x.

Zhu, C., Idemudia, C. U. and Feng, W. (2019) 'Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques', Informatics in Medicine Unlocked, 17(March), p. 100179. doi: 10.1016/j.imu.2019.100179.

PROFIL PENULIS

Dr. Widyastuti Andriyani, S.Kom., M.Kom lahir di Karanganyar, Surakarta pada Tanggal 17 Maret. Dengan latar belakang pendidikan Sarjana, Magister dan Doktor di bidang komputer, telah memperoleh pengalaman yang luas dalam dunia Teknologi Informasi. Sejak tahun 2019, telah menjadi dosen di Program Studi Magister Teknologi Informasi, Universitas Teknologi Digital Indonesia di Yogyakarta, yang berdedikasi, terus menerus melakukan riset, dengan fokus khusus pada bidang Sistem Cerdas. Riset ini mencakup berbagai aspek dalam pengembangan dan penerapan teknologi cerdas untuk meningkatkan kualitas kehidupan dan efisiensi dalam berbagai sector untuk menghadirkan inovasi baru yang dapat mengoptimalkan berbagai proses dan sistem dengan memanfaatkan kecerdasan buatan dan teknologi terkini. Dengan motivasi ini, terus berupaya untuk menjadi agen perubahan dalam mendorong kemajuan teknologi informasi dan menginspirasi generasi muda dan rekan sejawat untuk mengeksplorasi dan memanfaatkan potensi teknologi untuk kebaikan bersama.

Mochammad Anshori, M.Kom. lahir di kota Banyuwangi, Jawa Timur, tanggal 02 Maret 1994. Lulus Sarjarana pada tahun 2017 dan meraih gelar Magister Komputer di tahun 2020 di Universitas Brawijaya. Memulai karir mandiri menjadi seorang *freelancer* dan kini aktif menjadi dosen Informatika di Institut Teknologi, Sains, dan Kesehatan RS.DR. Soepraoen Kesdam V/BRW, Malang. Selain mengajar juga melaksasnakan tridharma perguruan tinggi dengan melaksanakan penelitian dan pengabdian kepada masyarakat. Fokus penelitiannya adalah di bidang Informatika khususnya *Bioinformatics* dan *Healthinformatics*. Sedangkan dalam pelaksanaan pengabdian kepada masyarakat cenderung menyajikan wawasan baru tentang kecerdasan buatan dan implementasi praktisnya. Ia juga pernah beberapa kali menjadi trainer pada kegiatan Digitalent yang diprakarsai oleh Kominfo dengan tema Pelatihan Data Science for High School Teacher. Pernah juga menjadi trainer pada pelatihan *Mobile Application Programming*.

Dwi Normawati, S.T., M.Eng. lahir di Kota Bengkulu pada tanggal 4 Agustus. Ia lulus pada tahun 2015 hingga mendapat gelar Master of Engineering (M.Eng) dari Magister Teknologi Informasi di Universitas Gadjah Madha, Yogyakarta. Saat ini tercatat sebagai dosen tetap dan mengajar di Program Studi Informatika, Universitas Ahmad Dahlan, Yogyakarta. Dia juga mengajar sebagai Trainer di Program Digital Talent Scholarship, Kominfo Program Vocational School Graduate Academy (VSGA) pada skema Associate Data Scientist sejak tahun 2022. Selain mengajar ia aktif dalam kegiatan tridarma lainnya diantaranya ialah penelitian dan pengabdian. Beberapa penelitian yang berhasil didanai oleh Ristekdikti dan hibah internal dari tahun 2019 hingga sekarang berjudul Seleksi Fitur Menggunakan Penambangan Data Berbasis Variable Precision Rough Set (VPRS) (2016), Implementasi K-Fold Cross Validation Pada Seleksi Fitur berbasis Rule Untuk Diagnosis Penyakit Jantung Koroner menggungkan metode VPRS (2019), Studi dan Implementasi Datamining untuk Penyakit Pre-eklamsia pada Ibu Hamil (2020), Sistem Klasifikasi Datamining Untuk Prediksi Dini Penyakit Pre-eklamsia Pada Ibu Hamil (2021), Analsis dan Perancangan Sistem Pendukung Keputusan Prioritas Penanganan Pasien Penyakit Stunting dengan Metode TOPSIS (2022), Analsis Clustering Penyakit Stunting pada Balita Menggunakan Algoritma K-Means (2023), dan Analisis pengelompokkan data Usaha Mikro, Kecil dan Menengah (UMKM) menggunakan metode K-Means dengan Clustering Purity dan Bee Colony Optimization (2024).

Risqy Siwi Pradini, S.S.T., M.Kom lahir di Trenggalek pada tanggal 28 Maret. Ia menyelesaikan pendidikan D4 dari Politeknik Negeri Malang, lalu melanjutkan studi S2 Ilmu Komputer di Universitas Brawijaya. Saat ini ia menjadi dosen tetap di Program Studi S1 Informatika ITSK RS dr. Soepraoen. Selain mengajar ia aktif dalam kegiatan Tridarma lainnya, diantaranya ialah melakukan berbagai macam penelitian di bidang Sistem Informasi dan *Artificial Intelligence*. Selain itu, ia juga aktif melakukan pelatihan tentang *Artificial Intelligence* yang ditujukan kepada siswa SMK. Saat ini, ia diamanahi sebagai *Chief of Editor* pada jurnal *Journal of Enhanced Studies in Informatics and Computer Application* (JESICA) dan menjadi *reviewer* aktif di beberapa jurnal nasional yang terakreditasi.

Mohamad Zaenudin, S.Pd., M.Sc.Eng. telah menjadi salah satu Dosen di Universitas Global Jakarta dengan status *Senior Lecturer*. Beliau menyelesaikan pendidikan S1 di Universitas Negeri Jakarta pada program studi S1 Pendidikan Teknik Elektronika dan melanjutkan studi S2 di Management and Science University dengan fokus pada bidang ilmu material. Selama perjalanan akademisnya, Mohamad Zaenudin telah meneliti dan menyelesaikan tesisnya mengenai simulasi dinamika molekuler pada fenomena welding difusi material Aluminium dan Nikel. Pengalaman riset ini memberikan kontribusi besar dalam pemahaman tentang material dan teknik pemrosesannya, yang menjadi fokus utama dalam pengabdian beliau di Universitas Global Jakarta. Sebagai seorang dosen, Mohamad Zaenudin telah mengajar berbagai mata kuliah, termasuk Matematika Teknik, Fisika, dan sekarang fokus pada mata kuliah keteknikan mesin, terutama pada bidang material, seperti Material Teknik, Metalurgi Fisik, dan Pemilihan Bahan dan Proses. Selain aktif dalam pengajaran, beliau juga aktif dalam berbagai kegiatan riset dan pengabdian kepada masyarakat. Sebagai capaian, beliau telah menerbitkan berbagai artikel ilmiah di bidang teknik mesin dan material pada jurnal internasional terindeks bereputasi dan jurnal nasional terindeks serta menjadi *reviewer* pada jurnal-jurnal tersebut. Prestasinya dalam riset telah diakui dengan diraihnya berbagai kesempatan hibah dengan sumber pendanaan dari RISTEKDIKTI DRTPM dan BELMAWA. Selain itu, Mohamad Zaenudin juga turut berkontribusi dalam pengembangan pendidikan tinggi di

Indonesia sebagai seorang *reviewer* pada Program *Matching Fund* tahun 2023 dan Program Dana Padanan tahun 2024, Dewan Pendidikan Tinggi, Dirjen Pendidikan Tinggi, Kemendikbud-Ristek.

Kontak: mzaenudin@jgu.ac.id

Profil Google Scholar: https://bit.ly/GS-MZaenudin

Profil *Researchgate*: https://bit.ly/RG-MZaenudin

Profil Scopus: https://bit.ly/SC-MZaenudin

Muhammad Iqbal Harisuddin, S.T., M.Pd. lahir di Subang, 19 Agustus 1981. Menyelesaikan Program S1 Teknik Kimia di ITENAS Bandung tahun 2005 dan program S2 Magister Pendidikan Matematika di Universitas Pasundan tahun 2014. Merupakan penulis aktif dan sudah menerbitkan buku; **Asyiknya Belajar Matematika dengan Geogebra** diterbitkan Deepublish tahun 2019; **Secuil Esensi Berpikir Kreatif dan Motivasi Belajar Siswa** diterbitkan Pantera Publishing tahun 2019; **Berpikir Kreatif Motivasi Kemandirian Belajar Siswa** diterbitkan Deepublish tahun 2023. Hasil penelitian penulis yang dipublish di Jurnal diantaranya; Kemampuan Pemecahan Masalah Matematis dan Kemandirian Belajar Siswa dengan PJJ Dimasa Covid-19, Jurnal Teorema: teori dan riset matematika tahun 2021; Implementation of Accelerated Learning to Improve Mathematics Communication Ability, Jurnal Pendidikan MIPA tahun 2022. Profesi sekarang penulis sebagai Dosen Pendidikan Matematika Universitas Mandiri Subang. Sebagai dosen, penulis mengampu mata kuliah yaitu; Geometri Analitik; Aplikasi

Matematika, Sains, Teknologi dan Rekayasa; Statistika Dasar; Program Linear; Persamaan Differensial; dan Metode Numerik.

M. Syauqi Haris, M.Kom. lahir di Kota Lamongan Jawa Timur pada 22 Februari 1985. Lulus dari Sarjana Ilmu Komputer Universitas Brawijaya pada tahun 2010 dan lulus dari Magister Ilmu Komputer Universitas Brawijaya pada tahun 2020. Berpengalaman lebih 10 tahun dalam industri pengembangan perangkat lunak dan saat ini aktif sebagai Kepala Program Studi S1 Informatika di Institut Teknologi, Sains, dan Kesehatan (ITSK) RS dr. Soepraoen Malang. Selain menjadi dosen, penulis aktif sebagai CEO PT Narasumber Teknologi Indonesia, sebuah startup di bidang konsultan teknologi informasi dan digital marketing bagi koperasi dan UMKM sehingga seringkali diundang sebagai narasumber oleh berbagai instansi negeri maupun swasta sebagai trainer di berbagai acara bimbingan teknis maupun pelatihan bagi koperasi dan UMKM. Dalam penelitian, penulis aktif melakukan riset di bidang rekayasa perangkat lunak dan pembelajaran mesin. Pada tahun 2022, menerima hibah Penelitian Dosen Pemula (PDP) dengan judul Prediksi Prevalensi Stunting di Provinsi Jawa Timur dengan menggunakan Algoritma *Machine Learning.*

Astuty, S.Pd., M.Eng.

Penulis lahir di Sidrap tanggal 11 Februari 1991. Setelah menyelesaikan pendidikan S1 pada Jurusan Pendidikan Teknik Elektro, penulis melanjutkan S2 pada Jurusan Teknik Elektro. Penulis saat ini aktif sebagai dosen tetap pada Program Studi Teknik Listrik dan Instalasi, Akademi Komunitas Industri Manufaktur Bantaeng. Penelitian yang dilakukan fokus pada bidang sistem tenaga listrik yang meliputi pembangkitan energi listrik dan sistem transmisi listrik. Penulis merupakan salah satu staff pengajar di Politeknik Teknologi kimia Industri Medan, suatu kampus negeri di bawah Kementerian Perindustrian. Menyelesaikan pendidikan Sarjana di Universitas Negeri Medan jurusan Pendidikan Matematika pada tahun 2015 dan magister di Universitas Sumatera Utara jurusan Matematika. Awal mulai mengajar dimulai dari les privat, mengajar di UPH Medan, hingga akhirnya lulus CPNS 2018 pada 24 desember 2018 di PTKI Medan. Penulis terus meningkatkan pengetahuan matematika dan menulis berbagai jurnal dan merambah ke buku. Matematika merupakan ilmu yang bisa dibilang abstrak sehingga dalam memahaminya diperlukan imajinasi dan tidak lepas bantuan dari Sang Pencipta.

Anna Angela Sitinjak merupakan salah satu staff pengajar di Politeknik Teknologi kimia Industri Medan, suatu kampus negeri di bawah Kementerian Perindustrian. Menyelesaikan pendidikan Sarjana di Universitas Negeri Medan jurusan Pendidikan Matematika pada tahun 2015 dan magister di Universitas Sumatera Utara jurusan Matematika. Awal mulai mengajar dimulai dari les privat, mengajar di UPH Medan, hingga akhirnya lulus CPNS 2018 pada 24 desember 2018 di PTKI Medan. Penulis terus meningkatkan pengetahuan matematika dan menulis berbagai jurnal dan merambah ke buku. Matematika merupakan ilmu yang bisa dibilang abstrak sehingga dalam memahaminya diperlukan imajinasi dan tidak lepas bantuan dari Sang Pencipta.

Wahyu Teja Kusuma, S.Kom., M.Kom. lahir di Kota Pasuruan pada tanggal 17 Mei 1992. Ia mendapat gelar Magister Komputer di Universitas Brawijaya pada tahun 2019. Saat ini ia tercatat sebagai dosen tetap di ITSK SOEPRAOEN Malang dengan legalitas keahlian dan kompetensi BNSP. Selain mengajar ia aktif dalam kegiatan tridarma lainnya diantaranya ialah penelitian dan pengabdian yang telah menghasilkan beberapa Paten, Hak Cipta, Prestasi Internasional, dan Prestasi Nasional. Saat ini ia pun diamanahi untuk menjadi Wakil Dekan Fakultas Sains dan Teknologi, mengembangkan program studi S1 Informatika di ITSK SOEPRAOEN, mengembangkan TUK BNSP Prodi S1 Informatika ITSK SOEPRAOEN LSP Teknologi Digital, mengembangkan publisher JESICA: Journal of Enhanced Studies in Sistem Informatics and Computer Applications, menjadi reviewer dan editorial board jurnal nasional.

E-BOOK
TOHAR MEDIA

MATEMATIKA PADA KECERDASAN BUATAN

Di era yang ditandai dengan kemajuan teknologi yang cepat, matematika telah menjadi infrastruktur kritis yang mendukung berbagai aspek kecerdasan buatan (AI). Buku ini membahas peran penting matematika dalam pengolahan data, pemodelan algoritma pembelajaran mesin, dan pengembangan model prediktif, yang tersaji pada Bab berikut:

Bab 1 Peran Penting Matematika dalam Kecerdasan Buatan

Bab 2 Matematika dalam Pengolahan Data

Bab 3 Pemodelan Matematika untuk Algoritma *Machine Learning*

Bab 4 Matematika dalam Pengembangan Model Prediktif

Bab 5 Konsep Matematika pada Statistik dan Probabilitas

Bab 6 Konsep Matematika pada Aljabar Linear

Bab 7 Memahami Kompleksitas Komputasi: Konsep, Model Deterministik dan Non-Deterministik, Serta Aplikasi Praktis

Bab 8 Integrasi Matematika dengan Teknologi

Bab 9 Penerapan Konsep Matematika dalam Penelitian

Bab 10 Peran Matematika dalam Perkembangan Kecerdasan Buatan di Masa Depan

Dengan adanya teknologi AI yang terus berkembang, buku ini mengeksplorasi bagaimana matematika dapat diterapkan dalam penelitian untuk mendapatkan wawasan baru dan inovasi. Masa depan matematika dalam kecerdasan buatan yang ditawarkan di bab-bab akhir buku ini menunjukkan bagaimana organisasi yang mengadopsi pendekatan ini tidak hanya meningkatkan efisiensi operasional mereka tetapi juga memperkuat posisi kompetitif di segala sektor kehidupan modern.

TOHAR MEDIA
No Anggota IKAPI : 022/SSL/2019
Workshop : JL. Adiyaksa Baru, Ruko Zamrud Blok I no 9
Redaksi : JL. Muhktar dg Tompo Kabupaten Gowa
Perumahan Nayla Regency Blok D No 25
Telp. (0411) 8987659 Hp. 085299993635
https://toharmedia.co.id

ISBN 978-623-8705-33-7
9 786238 705337