

RANGKUMAN MATERI PERKULIAHAN MACHINE LEARNING



Dosen Pengampu: Hendri Karisma, S.Kom., M.T

Rackisha Dhia Ezelly Lathief (241572010011)

Rizka Sugiarto (241572010013)

Muhammad Nabil Thoriq (241572010010)

Program Studi Sistem Informasi

Sekolah Tinggi Manajemen dan Ilmu Komputer Tazkia

Jalan Raya Dramaga Km.7, Dramaga, Bogor 16680

23 Januari 2026

1. Pengantar Machine Learning

1.1. Deskripsi Singkat Pembahasan

Pertemuan pertama membahas pengantar Machine Learning sebagai bagian dari Artificial Intelligence serta perannya dalam perkembangan teknologi informasi modern. Materi pada pertemuan ini difokuskan pada pemahaman konseptual mengenai Machine Learning, alasan mengapa teknologi ini berkembang pesat, serta bagaimana Machine Learning digunakan dalam berbagai bidang kehidupan dan industri.

Selain membahas konsep dasar, pertemuan ini juga menjelaskan arah perkuliahan, ruang lingkup materi yang akan dipelajari selama satu semester, serta keterkaitan Machine Learning dengan bidang lain seperti keamanan informasi, rekayasa perangkat lunak, teknologi blockchain, dan Environmental, Social, and Governance (ESG). Melalui pembahasan ini, mahasiswa diharapkan memperoleh gambaran awal mengenai pentingnya Machine Learning dalam dunia akademik maupun dunia kerja.

1.2. Taksonomi Materi

Struktur taksonomi Machine Learning pada pertemuan ini disusun secara hierarkis, dimulai dari konsep paling umum hingga komponen inti dan domain penerapannya. Taksonomi ini bertujuan untuk memperjelas posisi Machine Learning dalam Artificial Intelligence serta keterkaitannya dengan berbagai bidang lain.

- **Artificial Intelligence (AI)**

- Konsep payung besar yang bertujuan menciptakan sistem komputer yang mampu meniru atau menyerupai kecerdasan manusia.

- **Machine Learning (ML)**

- * Sub-bidang dari Artificial Intelligence yang berfokus pada sistem yang mampu belajar dari data.

- * **Proses Pembelajaran (Komponen Inti)**

- **Data** Berperan sebagai pengalaman yang menjadi dasar pembelajaran sistem.
- **Algoritma** Berfungsi sebagai mekanisme pembelajaran yang mengolah data untuk membangun model.
- **Model** Merupakan representasi pengetahuan yang dihasilkan dari proses pembelajaran dan digunakan untuk prediksi atau pengambilan keputusan.

* **Domain Penerapan (Sifat Multidisipliner)**

- Analisis Data
- Sistem Rekomendasi
- Keamanan dan Kepatuhan
- Rekayasa Perangkat Lunak
- Blockchain
- Environmental, Social, and Governance (ESG)

Taksonomi ini menunjukkan bahwa Machine Learning merupakan bagian integral dari Artificial Intelligence yang memanfaatkan data sebagai fondasi pembelajaran dan memiliki keterkaitan erat dengan berbagai domain penerapan di dunia nyata.

1.3. Penjelasan Detail Materi

Machine Learning merupakan salah satu cabang dari Artificial Intelligence yang berfokus pada pengembangan sistem komputer yang mampu belajar dari pengalaman. Berbeda dengan pemrograman konvensional yang mengandalkan aturan eksplisit yang ditentukan oleh manusia, Machine Learning memungkinkan sistem untuk membangun pengetahuan secara otomatis berdasarkan data yang tersedia. Pendekatan ini menjadi sangat relevan seiring meningkatnya volume, variasi, dan kompleksitas data di berbagai sektor.

Definisi Machine Learning yang digunakan dalam perkuliahan ini mengacu pada pendapat Tom Mitchell, yang menyatakan bahwa sebuah program dikatakan belajar dari pengalaman (E) terhadap suatu tugas (T) dan ukuran kinerja (P) apabila kinerjanya pada tugas T meningkat seiring bertambahnya pengalaman E. Definisi ini menekankan bahwa

pembelajaran mesin selalu melibatkan tiga komponen utama, yaitu tugas yang ingin diselesaikan, ukuran kinerja sebagai indikator keberhasilan, dan pengalaman yang diperoleh dari data. Tanpa salah satu dari ketiga komponen tersebut, proses pembelajaran tidak dapat berjalan secara optimal.

Dalam konteks Machine Learning, pengalaman diwujudkan dalam bentuk data. Data dipandang sebagai fakta yang merepresentasikan kondisi dunia nyata, sehingga kualitas data menjadi faktor yang sangat menentukan keberhasilan proses pembelajaran. Data yang tidak lengkap, tidak akurat, atau mengandung bias dapat menghasilkan model yang kurang optimal dan berpotensi menimbulkan kesalahan dalam pengambilan keputusan. Oleh karena itu, pengumpulan, pembersihan, dan pemahaman data merupakan tahap penting dalam pengembangan sistem Machine Learning.

Proses pembelajaran dalam Machine Learning dilakukan melalui mekanisme pengukuran kesalahan atau probabilitas secara berulang. Sistem akan membandingkan hasil prediksi yang dihasilkan oleh model dengan nilai yang sebenarnya, kemudian menyesuaikan parameter model untuk meminimalkan kesalahan tersebut. Proses ini berlangsung secara iteratif dan menyerupai cara manusia belajar melalui latihan dan umpan balik. Semakin banyak data dan pengalaman yang diperoleh, model diharapkan mampu meningkatkan kinerjanya secara bertahap.

Model yang dihasilkan dari proses pembelajaran disimpan dalam memori atau media penyimpanan, sehingga dapat digunakan kembali atau dipindahkan ke sistem lain. Hal ini memungkinkan model Machine Learning bersifat portabel dan dapat diterapkan pada berbagai lingkungan komputasi. Namun demikian, model dengan kompleksitas tinggi sering kali memerlukan sumber daya komputasi yang besar, sehingga biaya komputasi menjadi salah satu tantangan utama dalam penerapan Machine Learning, terutama di negara berkembang.

Dalam kurikulum perkuliahan, Machine Learning ditempatkan sebagai bagian dari jalur utama yang terintegrasi dengan kecerdasan buatan, keamanan, dan rekayasa perangkat lunak. Pada jalur Artificial Intelligence, penekanan diberikan pada pemahaman dasar algoritma pembelajaran mesin dan pendekatan berbasis metode. Mahasiswa tidak hanya dituntut mampu menggunakan alat, tetapi juga memahami prinsip kerja algoritma

yang digunakan.

Pada jalur keamanan, pembahasan mencakup tata kelola, manajemen risiko, dan kepatuhan atau Governance, Risk, and Compliance (GRC). Aspek ini menjadi semakin penting karena banyak sistem Machine Learning memproses data sensitif seperti data keuangan dan kesehatan. Oleh karena itu, penerapan Machine Learning harus memperhatikan isu kepemilikan data, residensi data, kebijakan retensi, serta kepatuhan terhadap regulasi yang berlaku.

Selain jalur utama tersebut, pertemuan ini juga menyoroti domain tambahan yang memiliki potensi besar untuk penelitian dan pengembangan, yaitu Blockchain dan ESG. Teknologi blockchain dipahami sebagai teknologi dasar yang mendukung sistem terdesentralisasi dengan karakteristik utama berupa keterlacakan dan keamanan data. Sementara itu, ESG menjadi bidang yang semakin penting dalam konteks keberlanjutan lingkungan dan tata kelola industri. Penerapan Machine Learning dalam ESG mencakup penghitungan emisi karbon, analisis citra satelit untuk memantau kondisi lingkungan, serta pelaporan keberlanjutan.

Machine Learning juga memiliki peran penting dalam dunia industri dan pasar kerja. Beberapa peran yang relevan antara lain Data Scientist, Machine Learning Engineer, AI Engineer, dan Data Engineer. Namun demikian, penerapan Machine Learning secara luas masih menghadapi kendala biaya komputasi yang tinggi, sehingga peran pakar domain menjadi sangat penting dalam mengarahkan penggunaan teknologi ini secara efektif dan bertanggung jawab.

1.4. Sample Code / Pseudocode

Pada pertemuan pertama ini belum dibahas implementasi kode atau pseudocode secara langsung. Materi difokuskan pada pemahaman konseptual mengenai Machine Learning, termasuk definisi, proses pembelajaran, serta konteks penerapannya dalam dunia akademik dan industri. Implementasi teknis akan dibahas pada pertemuan-pertemuan selanjutnya.

1.5. Ringkasan Bab

Bab ini membahas pengantar Machine Learning sebagai landasan awal dalam memahami pembelajaran mesin. Materi mencakup definisi Machine Learning, perbedaan dengan pemrograman konvensional, peran data dan model dalam proses pembelajaran, serta keterkaitannya dengan berbagai bidang lain. Pemahaman terhadap bab ini menjadi dasar penting sebelum mempelajari algoritma dan teknik Machine Learning secara lebih mendalam pada bab berikutnya.

2. Pendahuluan Machine Learning

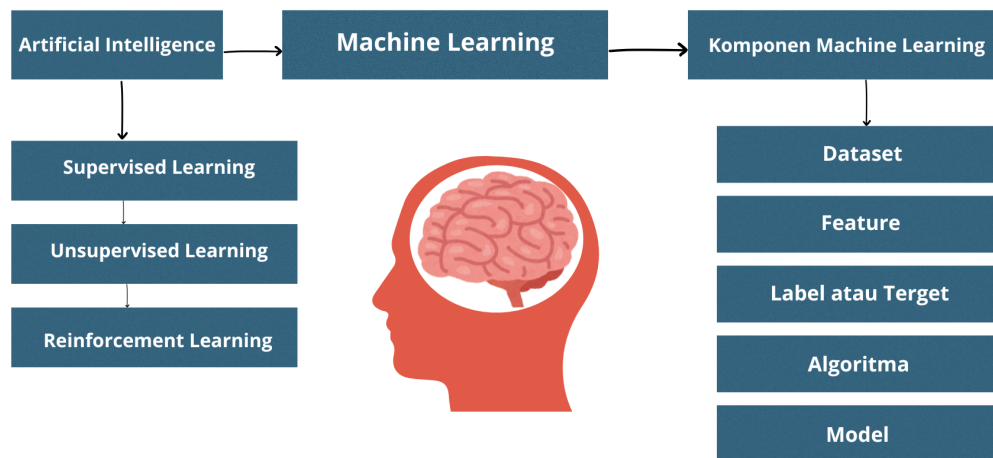
2.1. Deskripsi Singkat Pembahasan

Machine Learning merupakan salah satu cabang dari Artificial Intelligence yang berfokus pada pengembangan sistem komputer agar mampu belajar dari data. Berbeda dengan pemrograman konvensional yang bergantung pada aturan eksplisit, Machine Learning memungkinkan sistem untuk membangun pengetahuan berdasarkan data historis dan pengalaman sebelumnya.

Dalam perkuliahan ini, Machine Learning diperkenalkan sebagai teknologi yang banyak digunakan dalam berbagai bidang seperti sistem rekomendasi, analisis data, deteksi penipuan, serta pengambilan keputusan berbasis data.

2.2. Mind Map Machine Learning

Secara umum, Machine Learning dapat diklasifikasikan ke dalam beberapa kategori utama sebagai berikut:



Mindmap ini menunjukkan bahwa Machine Learning merupakan bagian dari Artificial Intelligence dengan data sebagai fondasi utama dalam proses pembelajaran. 1cm

2.3. Penjelasan Detail Materi

Machine Learning adalah pendekatan komputasi yang memungkinkan sistem komputer untuk mempelajari pola dari data dan menghasilkan prediksi atau keputusan. Sistem Machine Learning membangun model berdasarkan data latih, kemudian menggunakan model tersebut untuk memproses data baru.

Dalam Machine Learning, data berperan sebagai input utama, algoritma digunakan sebagai mekanisme pembelajaran, sedangkan model merupakan hasil dari proses pembelajaran tersebut. Model yang dihasilkan bersifat probabilistik, sehingga hasil prediksi tidak selalu bersifat mutlak.

2.4. Sample Code / Pseudocode

Berikut merupakan pseudocode sederhana yang menggambarkan alur kerja Machine Learning:

Input:

Dataset D

Algorithm A

Process:

Model M = Train(A, D)

Output:

Model M digunakan untuk prediksi

2.5. Ringkasan Bab

Bab ini membahas pengenalan dasar Machine Learning, mulai dari definisi, ruang lingkup, hingga komponen utama yang terlibat dalam proses pembelajaran mesin. Pemahaman terhadap bab ini menjadi dasar penting sebelum mempelajari algoritma Machine Learning secara lebih mendalam pada bab selanjutnya.

3. Penerapan Machine Learning dan Artificial Intelligence pada Proyek Karbon dan ESG

3.1. Deskripsi Singkat Pembahasan

Pertemuan ketiga membahas penerapan Machine Learning dan Artificial Intelligence dalam konteks proyek karbon dan Environmental, Social, and Governance (ESG). Fokus utama pembahasan adalah bagaimana teknologi AI digunakan untuk mendukung validasi proyek karbon, pemantauan kepatuhan, serta pengelolaan data yang berkaitan dengan emisi dan keberlanjutan lingkungan.

Selain itu, pertemuan ini juga menyoroti keterkaitan antara teknologi AI, regulasi, dan mekanisme pasar karbon, baik di tingkat nasional maupun internasional. Mahasiswa diperkenalkan pada contoh implementasi nyata penggunaan AI dalam mendukung

pengambilan keputusan berbasis data pada proyek lingkungan.

3.2. Taksonomi Materi

Penerapan Artificial Intelligence (AI) pada proyek karbon dan Environmental, Social, and Governance (ESG) dapat diklasifikasikan ke dalam beberapa lapisan konsep dan teknologi. Klasifikasi ini disajikan dalam bentuk taksonomi pada Tabel berikut.

Lapisan Konsep	Sub Bidang	Peran dan Fungsi
Artificial Intelligence (AI)	–	Bidang utama yang mencakup kemampuan pencarian, penalaran, perencanaan, dan pembelajaran.
Machine Learning (ML)	–	Sub-bidang AI yang memungkinkan sistem belajar dari data historis dan observasi lapangan.
Machine Learning (ML)	Deep Learning	Digunakan untuk mengolah data kompleks seperti citra satelit dan foto lapangan.
Machine Learning (ML)	Generative AI	Menghasilkan model generatif seperti Large Language Models (LLMs) untuk mendukung analisis dan pelaporan.
Teknologi Pendukung	Blockchain	Menjaga integritas data dan mencegah penyalahgunaan sertifikat karbon.

Taksonomi ini menunjukkan bahwa penerapan AI pada proyek karbon dan ESG bersifat multidisipliner, melibatkan integrasi antara kecerdasan buatan, pembelajaran mesin, teknologi pembelajaran mendalam, model generatif, serta sistem tata kelola data yang andal.

Dalam konteks ini, Deep Learning menjadi bagian dari Machine Learning yang

mendukung pengolahan data kompleks seperti citra satelit dan foto lapangan, sementara Generative AI menghasilkan model-model seperti Large Language Models (LLMs) yang digunakan dalam aplikasi pendukung analisis dan pelaporan. Teknologi blockchain berperan sebagai mekanisme pendukung untuk menjaga integritas data dan mencegah penyalahgunaan sertifikat karbon. Taksonomi ini menunjukkan bahwa penerapan AI pada proyek karbon bersifat multidisipliner dan melibatkan berbagai teknologi yang saling terintegrasi.

3.3. Penjelasan Detail Materi

Proyek karbon merupakan salah satu upaya untuk mengurangi dampak perubahan iklim melalui kegiatan seperti reboisasi, pemulihan fungsi hutan, dan pengelolaan emisi karbon. Dalam konteks pasar karbon, perusahaan dapat berpartisipasi dengan cara mengimbangi emisi yang dihasilkan melalui investasi pada proyek-proyek lingkungan. Teknologi Artificial Intelligence dan Machine Learning memainkan peran penting dalam memastikan bahwa proyek-proyek tersebut berjalan sesuai dengan standar dan klaim yang diajukan dapat diverifikasi secara akurat.

Salah satu penerapan AI dalam proyek karbon adalah validasi penanaman pohon menggunakan teknologi visi komputer. Melalui analisis foto lapangan, sistem berbasis Machine Learning dapat mengenali genus atau spesies pohon yang ditanam serta mendeteksi ketidaksesuaian dengan spesifikasi proyek. Pendekatan ini membantu meningkatkan akurasi verifikasi dan mengurangi ketergantungan pada pemeriksaan manual yang memerlukan waktu dan biaya besar.

Selain validasi visual, Machine Learning juga digunakan dalam pengecekan data geospasial dan tutupan lahan. Data citra satelit dan peta geospasial dianalisis untuk memastikan bahwa proyek karbon dilakukan pada lahan yang sesuai, seperti bukan kawasan hutan lindung atau lahan basah yang tidak diperbolehkan. Proses ini mendukung kepatuhan terhadap regulasi dan standar lingkungan yang berlaku sebelum proyek disetujui dan dijalankan.

Dalam operasional proyek karbon, otomatisasi berbasis AI digunakan untuk me-

antau kepatuhan pada seluruh alur kerja proyek, mulai dari tahap perencanaan hingga pelaporan. Sistem AI membantu mengidentifikasi potensi pelanggaran lebih awal dan memberikan peringatan kepada pengelola proyek. Namun demikian, penerapan teknologi ini juga menghadapi tantangan teknis, seperti kinerja aplikasi web yang lambat pada beberapa bagian alur kerja, yang dapat memengaruhi efisiensi proses validasi.

Aspek kepatuhan dan standar menjadi perhatian utama dalam pengelolaan proyek karbon. Di Indonesia, proyek karbon terdaftar dan diawasi melalui Sistem Registri Nasional (SRN) yang dikelola oleh pemerintah. Selain itu, terdapat standar internasional seperti VERRA dan ISO 27001 yang mengatur aspek keamanan informasi dan integritas data. Teknologi blockchain dimanfaatkan untuk mencegah klaim ganda atau penyalahgunaan sertifikat karbon yang telah diterbitkan, sehingga meningkatkan transparansi dan kepercayaan pasar.

Dalam konteks metrik dan kapasitas, data yang tercatat dalam registri seperti SRN menunjukkan jumlah emisi yang dapat diklaim oleh perusahaan, misalnya dalam satuan ton ekuivalen CO₂ (tCO₂e). Informasi ini menjadi dasar dalam penentuan harga karbon serta pengambilan keputusan bisnis, terutama bagi sektor dengan tingkat emisi tinggi seperti pertambangan dan industri ekstraktif.

Pertemuan ini juga mengaitkan kembali materi dengan dasar-dasar Artificial Intelligence dan Machine Learning yang telah dipelajari sebelumnya. AI dipahami sebagai bidang utama, Machine Learning sebagai bagian dari AI, Deep Learning sebagai bagian dari Machine Learning, dan Generative AI sebagai bagian dari Deep Learning. Model bahasa besar seperti GPT dan Gemini merupakan contoh hasil dari AI generatif yang digunakan dalam berbagai aplikasi, termasuk ChatGPT sebagai antarmuka pengguna. Pemahaman taksonomi ini membantu mahasiswa melihat posisi setiap teknologi dalam ekosistem AI secara menyeluruh.

3.4. Sample Code / Pseudocode

Pada pertemuan ini belum dibahas secara langsung implementasi kode atau pseudocode Machine Learning. Materi difokuskan pada pemahaman konsep, arsitektur sistem, serta

penerapan AI dalam skenario nyata proyek karbon dan ESG. Implementasi teknis akan dibahas pada pertemuan selanjutnya yang membahas algoritma dan metode Machine Learning secara lebih mendalam.

3.5. Ringkasan Bab

Bab ini membahas penerapan Artificial Intelligence dan Machine Learning dalam proyek karbon dan ESG. Materi mencakup peran AI dalam validasi penanaman, analisis data geospasial, pemantauan kepatuhan, serta penggunaan blockchain untuk menjaga integritas sertifikat karbon. Pembahasan ini menunjukkan bahwa Machine Learning memiliki peran strategis dalam mendukung keberlanjutan lingkungan dan tata kelola industri berbasis data.

4. Sinkronisasi Proyek dan Pendalaman Konsep Machine Learning

4.1. Deskripsi Singkat Pembahasan

Bab ini membahas sesi sinkronisasi proyek pembelajaran Machine Learning yang bertujuan untuk menyatukan pemahaman mahasiswa terhadap konsep-konsep fundamental sebelum memasuki tahap implementasi dan pemodelan lanjutan. Materi yang dibahas mencakup paradigma pembelajaran mesin, penalaran induktif dan deduktif, pembentukan data dan pengetahuan, pendekatan pemodelan, pengelolaan dataset dan fitur, taksonomi metode pembelajaran, hingga pengenalan algoritma Fine-S.

Selain aspek teknis, sesi ini juga menekankan koordinasi akademik terkait penggunaan alat pendukung seperti GitHub dan Discord, serta penyelarasan tugas antar kelompok agar proses pembelajaran berjalan terstruktur dan konsisten.

4.2. Taksonomi Materi

Secara konseptual, materi pada bab ini dapat dipetakan ke dalam struktur taksonomi Machine Learning sebagai berikut:

- **Paradigma Pembelajaran Machine Learning**

- Penalaran Induktif
- Penalaran Deduktif

- **Pembentukan Pengetahuan**

- Data
- Informasi
- Pengetahuan / Hipotesis / Model
- Pengetahuan Deterministik dan Stokastik

- **Pendekatan Pemodelan**

- Fungsi Ahli (FX)
- Aproksimasi Model Terpelajar ($F'(X)$)
- Evaluasi Model dan Kesalahan

- **Pengelolaan Dataset dan Fitur**

- Dataset Terstruktur dan Tidak Terstruktur
- Fitur, Atribut, dan Dimensi
- Label dan Target

- **Taksonomi Metode Pembelajaran**

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Reinforcement Learning
- Deep Learning

- **Algoritma Pembelajaran**

- K-Nearest Neighbors

- Support Vector Machine
- K-Means
- Deep Neural Network
- Algoritma Fine-S

Taksonomi ini memperlihatkan hubungan antara konsep dasar, data, pendekatan pembelajaran, hingga algoritma yang digunakan dalam proses Machine Learning.

4.3. Penjelasan Detail Materi

Machine Learning bekerja berdasarkan paradigma pembelajaran induktif, yaitu proses pembentukan pengetahuan dengan cara menggeneralisasi pola dari data-data spesifik. Berbeda dengan sistem deduktif seperti sistem pakar yang menggunakan aturan formal yang telah ditentukan sebelumnya, Machine Learning memungkinkan sistem membangun model secara otomatis berdasarkan pengalaman.

Definisi pembelajaran mesin yang banyak digunakan dikemukakan oleh Tom Mitchell, yang menyatakan bahwa sebuah program komputer dikatakan belajar dari pengalaman (E) terhadap suatu tugas (T) dengan ukuran kinerja (P), apabila kinerjanya meningkat seiring bertambahnya pengalaman. Definisi ini menjadi dasar formal dalam mengevaluasi apakah sebuah sistem benar-benar melakukan proses pembelajaran.

Dalam konteks pembentukan pengetahuan, data dipandang sebagai pengamatan faktual yang dapat diverifikasi. Data tersebut kemudian diolah menjadi informasi ketika diberi konteks, dan selanjutnya membentuk pengetahuan dalam bentuk hipotesis atau model. Pengetahuan yang dihasilkan bersifat stokastik, artinya hasil prediksi bersifat probabilistik dan tidak selalu mutlak benar.

Pendekatan pemodelan dalam Machine Learning bertujuan untuk mendekati fungsi ahli (F_X) melalui model terpelajar $F'(X)$. Model ini dievaluasi dengan membandingkan hasil prediksi terhadap nilai sebenarnya menggunakan metrik kesalahan seperti Mean Squared Error (MSE). Model diterima ketika tingkat kesalahan berada pada ambang yang dapat diterima, meskipun risiko kesalahan tetap ada.

Pengelolaan dataset juga menjadi aspek penting, terutama dalam memahami perbedaan antara data terstruktur dan tidak terstruktur. Dataset seperti *Play Tennis* digunakan sebagai contoh instruksional untuk memperkenalkan konsep fitur, label, dan instansi. Fitur dapat disebut sebagai atribut atau dimensi, sedangkan contoh data lengkap dengan label disebut sebagai titik data.

4.4. Sample Code / Pseudocode Algoritma Fine-S

Algoritma Fine-S merupakan metode pembelajaran dari contoh spesifik menuju hipotesis yang lebih umum. Algoritma ini bertujuan menemukan hipotesis paling spesifik yang konsisten dengan seluruh contoh positif.

```
Input:
    Dataset D dengan fitur dan label

Initialize:
    H = hipotesis paling spesifik (kosong)

For setiap contoh positif x dalam D:
    For setiap fitur i:
        Jika H[i] kosong:
            H[i] = x[i]
        Jika H[i] != x[i]:
            H[i] = '?'

Output:
    Hipotesis H
```

Tanda “?” menunjukkan bahwa fitur tersebut tidak menjadi pembatas dalam menentukan label, sehingga menerima nilai apa pun.

4.5. Ringkasan Bab

Bab ini membahas sinkronisasi pemahaman konsep Machine Learning secara menyeluruh, mulai dari paradigma pembelajaran, pembentukan pengetahuan, pengelolaan dataset, hingga pengenalan algoritma Fine-S. Selain itu, dibahas pula aspek praktis terkait kompleksitas komputasi, keterbatasan sumber daya, serta koordinasi akademik untuk mendukung proses pembelajaran yang efektif. Bab ini menjadi jembatan penting sebelum memasuki materi pemodelan dan implementasi algoritma Machine Learning secara lebih mendalam.

5. Pembelajaran Terawasi dan Pohon Keputusan

5.1. Deskripsi Singkat Pembahasan

Pertemuan ini membahas konsep pembelajaran terawasi (*supervised learning*) dengan fokus pada pemodelan menggunakan pohon keputusan, khususnya algoritma ID3. Materi disajikan dalam bentuk sinkronisasi proyek sekaligus pembaruan status pembelajaran, yang mencakup pemahaman algoritma, evaluasi model, kesiapan alat praktis, serta koordinasi tugas perkuliahan.

Selain aspek teknis, sesi ini juga menekankan pentingnya manajemen proyek akademik, penggunaan GitHub sebagai sarana kolaborasi, serta perencanaan implementasi manual algoritma ID3 pada pertemuan berikutnya.

5.2. Taksonomi Materi

Materi pada pertemuan ini dapat dipetakan dalam taksonomi pembelajaran terawasi sebagai berikut:

- **Pembelajaran Terawasi**
 - Dataset berlabel
 - Target diskrit dan kontinu

- **Model Klasifikasi**

- Pohon Keputusan (ID3, C4.5)
- Naive Bayes
- Support Vector Machine
- Jaringan Saraf Tiruan

- **Evaluasi Model**

- Validasi data latih dan uji
- Akurasi
- Confusion Matrix

- **Peralatan Pendukung**

- Jupyter Notebook
- pandas, NumPy, SciPy
- GitHub dan Discord

Taksonomi ini menunjukkan keterkaitan antara pendekatan pembelajaran terawasi, algoritma klasifikasi, teknik evaluasi, serta infrastruktur pendukung dalam proses implementasi Machine Learning.

5.3. Penjelasan Detail Materi

Pembelajaran terawasi merupakan pendekatan Machine Learning yang menggunakan dataset berlabel, di mana setiap contoh data memiliki target atau kelas yang jelas. Dalam pertemuan ini, mahasiswa mempelajari perbedaan antara pembelajaran terawasi, pembelajaran tanpa pengawasan, dan pembelajaran penguatan, serta memahami bahwa model merupakan hipotesis atau pengetahuan yang dihasilkan dari proses pembelajaran.

Fokus utama sesi ini adalah algoritma pohon keputusan, khususnya ID3. Pohon keputusan digunakan sebagai model yang memetakan input X ke prediksi \hat{y} , dengan tujuan mendekati fungsi sebenarnya $f(x)$. Struktur pohon terdiri dari node akar, node internal, dan daun yang merepresentasikan keputusan akhir.

Algoritma ID3 bekerja dengan memilih atribut terbaik untuk pemisahan data berdasarkan ukuran *information gain*. Proses dimulai dengan menghitung entropi dataset untuk mengukur tingkat ketidakmurnian kelas, kemudian menghitung entropi kondisional pada setiap atribut kandidat. Atribut dengan nilai keuntungan informasi tertinggi dipilih sebagai node pemisah.

Evaluasi model dilakukan melalui pembagian dataset menjadi data latih dan data uji, dengan rasio umum 70% untuk pelatihan dan 30% untuk pengujian. Kinerja model diukur menggunakan akurasi dan confusion matrix, yang mencakup True Positive, True Negative, False Positive, dan False Negative. Dalam konteks aplikasi bisnis seperti deteksi penipuan, kesalahan klasifikasi memiliki dampak signifikan, sehingga penyesuaian ambang keputusan menjadi aspek yang sangat penting.

Pertemuan ini juga membahas risiko underfitting dan overfitting. Model yang terlalu sederhana cenderung gagal menangkap pola data, sedangkan model yang terlalu kompleks berisiko tidak mampu melakukan generalisasi dengan baik. Oleh karena itu, pohon keputusan harus dibangun dengan tingkat kompleksitas yang seimbang.

5.4. Sample Code / Pseudocode Algoritma ID3

Berikut merupakan pseudocode sederhana yang menggambarkan alur kerja algoritma ID3:

```
Input:
    Dataset D dengan atribut dan label kelas

Procedure ID3(D, Attributes):
    Jika semua contoh dalam D memiliki kelas yang sama:
        Return node daun dengan label kelas tersebut
    Jika Attributes kosong:
        Return node daun dengan kelas mayoritas di D
    Hitung entropi D
    Untuk setiap atribut A dalam Attributes:
        Hitung InformationGain(D, A)
```

```
Pilih atribut A_best dengan gain tertinggi
Buat node dengan A_best sebagai akar
Untuk setiap nilai v dari A_best:
    Bentuk subset D_v
    Jika D_v kosong:
        Tambahkan daun dengan kelas mayoritas di D
    Else:
        Panggil ID3(D_v, Attributes - A_best)
Return pohon
```

5.5. Ringkasan Bab

Pertemuan ini membahas pembelajaran terawasi dengan fokus pada algoritma pohon keputusan ID3 sebagai metode klasifikasi yang mudah dipahami dan diimplementasikan. Mahasiswa mempelajari konsep entropi, information gain, serta teknik evaluasi model menggunakan data latih dan uji. Selain itu, dibahas pula pentingnya manajemen proyek akademik melalui penggunaan GitHub dan koordinasi kelas melalui Discord untuk mendukung kelancaran tugas dan implementasi praktis pada pertemuan selanjutnya.

6. Implementasi Pohon Keputusan dan Lingkungan Pengembangan

6.1. Deskripsi Singkat Pembahasan

Pada pertemuan ini dibahas penerapan algoritma pohon keputusan dalam pembelajaran terawasi, khususnya algoritma ID3 dan C4.5. Mahasiswa mempelajari bagaimana sebuah model klasifikasi dibangun melalui perhitungan entropi dan information gain, serta bagaimana algoritma C4.5 memperluas konsep tersebut untuk menangani fitur kontinu menggunakan threshold dan gain ratio.

Selain aspek algoritmik, pertemuan ini juga menekankan pentingnya kesiapan lingkungan pengembangan berbasis Python, penggunaan Jupyter Notebook sebagai media eksperimen ilmiah, serta penerapan praktik kerja profesional melalui penggunaan vir-

tual environment dan GitHub untuk kolaborasi tim.

6.2. Taksonomi Materi

Secara konseptual, materi pada pertemuan ini dapat disusun dalam bentuk taksonomi sebagai berikut:

- **Artificial Intelligence (AI)**
 - Machine Learning
 - * Supervised Learning
 - * Klasifikasi
 - * Pohon Keputusan (ID3, C4.5)
- **Komponen Pendukung**
 - Dataset: Play Tennis, Iris
 - Metode Evaluasi: Entropi, Information Gain, Gain Ratio
 - Lingkungan Pengembangan:
 - * Python
 - * Jupyter Notebook
 - * Virtual Environment (venv)
 - * GitHub

Taksonomi ini menunjukkan bahwa implementasi pohon keputusan tidak berdiri sendiri, melainkan berada dalam ekosistem Machine Learning yang terintegrasi dengan data, algoritma, dan infrastruktur pengembangan.

6.3. Penjelasan Detail Materi

6.3.1 Algoritma ID3

Algoritma ID3 membangun pohon keputusan berdasarkan prinsip pengurangan ketidakpastian kelas. Ketidakpastian ini diukur menggunakan entropi, sedangkan efektivitas su-

atu atribut dalam memisahkan data diukur menggunakan information gain.

Pada dataset Play Tennis, proses dimulai dengan menghitung entropi total dataset yang bernilai sekitar 0,94. Selanjutnya, setiap atribut dievaluasi dan diperoleh bahwa atribut *Outlook* memiliki nilai information gain tertinggi sehingga dipilih sebagai akar pohon keputusan.

Struktur pohon yang dihasilkan menunjukkan bahwa:

- Cabang *Mendung* langsung menghasilkan keputusan *Ya*.
- Cabang *Cerah* dipisahkan oleh atribut *Kelembapan*.
- Cabang *Hujan* dipisahkan oleh atribut *Angin*.

Ketika sebuah cabang telah mencapai kondisi murni, proses pemisahan dihentikan. Hal ini mencerminkan prinsip dasar ID3 dalam membangun model yang sederhana namun efektif.

6.3.2 Algoritma C4.5

Algoritma C4.5 merupakan pengembangan dari ID3 yang dirancang untuk menangani atribut kontinu. C4.5 menggunakan konsep *thresholding* untuk mengubah data numerik menjadi dua kelompok berdasarkan nilai ambang tertentu.

Pada dataset Iris, salah satu fitur yang dianalisis adalah *Petal Length*. Dengan ambang batas sebesar 2,45, data dibagi menjadi dua subset utama, yaitu subset dengan nilai kurang dari atau sama dengan ambang, dan subset dengan nilai lebih besar dari ambang.

Pemilihan atribut pada C4.5 tidak hanya didasarkan pada information gain, tetapi juga pada *gain ratio*, yang memperhitungkan informasi pembagian agar tidak bias terhadap atribut dengan banyak nilai. Hasil analisis menunjukkan bahwa *Petal Length* merupakan pemisah awal terbaik pada dataset Iris.

6.3.3 Lingkungan Pengembangan

Untuk mendukung implementasi algoritma, digunakan Python sebagai bahasa pemrograman utama dengan Jupyter Notebook sebagai lingkungan eksperimen. Jupyter memungkinkan mahasiswa untuk:

- Membaca dan memanipulasi data menggunakan pandas.
- Melakukan perhitungan numerik menggunakan NumPy.
- Menampilkan hasil perhitungan secara interaktif.

Selain itu, penggunaan virtual environment dianjurkan agar setiap proyek memiliki dependensi yang terisolasi. Praktik ini mencerminkan standar profesional dalam pengembangan perangkat lunak dan data science.

Kolaborasi tim difasilitasi melalui GitHub, yang berfungsi sebagai media pengumpulan tugas sekaligus alat untuk melacak kontribusi setiap anggota.

6.4. Sample Code / Pseudocode

Berikut adalah pseudocode sederhana untuk algoritma ID3:

```
Function ID3(Data, Attributes):  
    If all examples have the same class:  
        Return a leaf node with that class  
  
    If Attributes is empty:  
        Return a leaf node with the most common class  
  
    Compute entropy of Data  
    For each attribute in Attributes:  
        Compute Information Gain  
  
    Select attribute with highest Information Gain as  
        BestAttribute
```

```
Create a decision node based on BestAttribute

For each value v of BestAttribute:
    Subset = examples in Data with BestAttribute = v
    ChildNode = ID3(Subset, Attributes - BestAttribute)
    Attach ChildNode to decision node

Return decision node
```

Untuk C4.5, perbedaannya terletak pada penggunaan *threshold* dan *gain ratio*:

```
Function C45(Data, Attributes):
    For each numeric attribute:
        Determine possible thresholds
        Compute Gain Ratio for each threshold

    Select attribute and threshold with highest Gain Ratio
    Split Data based on threshold
    Repeat recursively until stopping criteria met
```

6.5. Ringkasan Bab

Bab ini membahas penerapan algoritma pohon keputusan ID3 dan C4.5 sebagai bagian dari pembelajaran terawasi dalam Machine Learning. Melalui studi kasus dataset Play Tennis dan Iris, mahasiswa memahami bagaimana konsep entropi, information gain, threshold, dan gain ratio digunakan untuk membangun model klasifikasi yang efektif.

Selain aspek algoritmik, bab ini juga menekankan pentingnya kesiapan lingkungan pengembangan, mulai dari penggunaan Python dan Jupyter Notebook hingga penerapan virtual environment dan GitHub. Kombinasi antara pemahaman teori dan praktik ini menjadi landasan penting bagi pengembangan sistem Machine Learning yang profesional dan berkelanjutan.

7. Deskripsi Singkat Pembahasan

Bab ini membahas sinkronisasi proyek kelas, meliputi pengelolaan jadwal dan kehadiran, panduan penyusunan tugas dan makalah penelitian, pemilihan dataset dan alat analisis, serta kemajuan teknis pada materi regresi linier menggunakan Python dan NumPy. Fokus utama bab ini adalah memastikan seluruh kelompok bekerja secara terkoordinasi, mengikuti standar akademik, serta mampu mengimplementasikan metode analisis data secara benar dan efisien.

7.1 Mindmap / Taksonomi Konsep

BAB 7: Sinkronisasi Proyek & Kemajuan Teknis

A. Jadwal & Logistik

Pertemuan ke-6

Sesi pengganti

Kehadiran (absensi)

Tanya jawab via Discord

B. Panduan Tugas & Proyek

Format makalah UNICOM

Pengumpulan via GitHub

Jadwal proposal

Tujuan publikasi

Komposisi tim

Pemilihan metode utama

C. Data & Alat

Pemilihan dataset terbaru

Modalitas data (tabular, teks, gambar)

NumPy vs Pandas

Alur kerja pengolahan data

D. Kemajuan Teknis

Regresi linier sederhana

Solusi analitik

Kesalahan umum data

NumPy vs List Python

Python vs Java

Stochastic Gradient Descent (SGD)

\section*{7.2 Penjelasan Detail}

\subsection*{7.2.1 A. Jadwal dan Logistik}

Pertemuan ini merupakan pertemuan keenam, namun satu sesi sebelumnya te

Mahasiswa diwajibkan mengisi link kehadiran yang dibagikan di chat. Sel

\subsection*{7.2.2 B. Panduan Tugas dan Proyek}

Makalah penelitian harus mengikuti format makalah UNICOM. Hasil kode da

Proposal pertama harus diserahkan paling lambat tanggal 25 Oktober. Mah

Kelas menargetkan publikasi karya di ``Jurnal Komputa'', sehingga kualifi

Setiap kelompok harus memilih satu metode utama dan tidak boleh ada dua

\subsection*{7.2.3 C. Data dan Alat}

Dataset dan referensi yang digunakan harus terbaru, sebaiknya dalam 10

NumPy digunakan untuk perhitungan ilmiah dan bekerja dengan array, seda

Alur kerja yang dianjurkan adalah mengekspor data ke format CSV dan men

\subsection*{7.2.4 D. Kemajuan Teknis}

Regresi linier sederhana memiliki bentuk model:

```
\[
y = mx + c \quad \text{atau} \quad y = w_0 + w_1 x
\]
```

Nilai m (slope) menunjukkan kemiringan garis, sedangkan c (intercep

Solusi analitik (closed-form) digunakan untuk menghitung nilai m dan

Kesalahan umum dalam pengolahan data meliputi tidak melewati baris head

Python bersifat dinamis dan diinterpretasikan sehingga kode mudah dibaca

Stochastic Gradient Descent (SGD) digunakan sebagai metode optimasi num

\section*{7.3 Sample Code / Pseudocode}

\subsection*{7.3.1 Absensi dan Sinkronisasi Jadwal}

```
\begin{lstlisting}
IF student_fills_attendance_form:
    mark_present(student_id)
ELSE:
    mark_absent(student_id)
```

```
ENDIF
```

```
\end{lstlisting}
```

```
\subsection*{7.3.2 Validasi Metode Utama}
```

```
\begin{lstlisting}
```

```
IF chosen_method IN used_methods:
```

```
    reject_choice()
```

```
    request_new_method()
```

```
ELSE:
```

```
    approve_choice()
```

```
    add_to_used_methods(chosen_method)
```

```
ENDIF
```

```
\end{lstlisting}
```

```
\subsection*{7.3.3 Membaca CSV dengan NumPy}
```

```
\begin{lstlisting}[language=Python]
```

```
import numpy as np
```

```
data = np.loadtxt("data.csv", delimiter=",", skiprows=1, dtype=float)
```

```
\end{lstlisting}
```

```
\subsection*{7.3.4 Regresi Linier -- Solusi Analitik}
```

```
\begin{lstlisting}[language=Python]
```

```
import numpy as np
```

```
x = np.array([1, 2, 3, 4, 5], dtype=float)
```

```
y = np.array([2, 4, 5, 4, 5], dtype=float)
```

```

n = len(x)
m = (n * np.sum(x*y) - np.sum(x) * np.sum(y)) / (n * np.sum(x**2) - (np
c = (np.sum(y) - m * np.sum(x)) / n

```

```

y_pred = m * x + c
residuals = y - y_pred
\end{lstlisting}

```

\subsection*{7.3.5 Regresi Linier dengan SGD}

```

\begin{lstlisting}[language=Python]
import numpy as np

x = np.array([1, 2, 3, 4, 5], dtype=float)
y = np.array([2, 4, 5, 4, 5], dtype=float)

m, c = 0.0, 0.0
lr = 0.01
epochs = 1000

```

```

for epoch in range(epochs):
    for i in range(len(x)):
        y_pred = m * x[i] + c
        error = y_pred - y[i]
        m -= lr * error * x[i]
        c -= lr * error
\end{lstlisting}

```

\documentclass[11pt,a4paper]{article}

\usepackage[margin=2.2cm]{geometry}

```

\usepackage{amsmath,amssymb}
\usepackage{hyperref}
\usepackage{enumitem}
\usepackage{xcolor}
\usepackage{listings}
\usepackage{tikz}
\usetikzlibrary{mindmap,trees}

\hypersetup{
    colorlinks=true,
    linkcolor=black,
    urlcolor=blue
}

\begin{document}

% Listings style
\lstdefinestyle{codeStyle}{
    basicstyle=\ttfamily\small,
    breaklines=true,
    frame=single,
    rulecolor=\color{black!30},
    numbers=left,
    numberstyle=\tiny\color{black!50},
    xleftmargin=2em,
    framexleftmargin=1.5em
}
\lstset{style=codeStyle}
\section*{8. Machine Learning dan Decision Tree (Algoritma ID3)}

```

```
\section*{8.1 Deskripsi Singkat Pembahasan}
```

Bab ini membahas konsep dasar Machine Learning, khususnya perbedaan met

```
\section*{8.2 Mindmap / Taksonomi Konsep}
```

```
\begin{itemize}
  \item \textbf{Machine Learning}
    \begin{itemize}
      \item Output: Model
    \end{itemize}
  \item \textbf{Jenis Pembelajaran}
    \begin{itemize}
      \item Supervised
      \item Unsupervised
      \item Semi-supervised
      \item Reinforcement
    \end{itemize}
  \item \textbf{Model}
    \begin{itemize}
      \item Decision Tree
      \item Neural Network
      \item Probabilistic
    \end{itemize}
  \item \textbf{Algoritma}
    \begin{itemize}
      \item ID3
      \item C4.5
      \item Random Forest
      \item Naive Bayes
    \end{itemize}
\end{itemize}
```

```

\item \textbf{Evaluasi}
\begin{itemize}
\item Training{Testing Split
\item Confusion Matrix
\item Akurasi
\item Overfitting dan Underfitting
\end{itemize}
\end{itemize}

```

\section*{8.3 Penjelasan Inti}

Machine Learning menghasilkan model sebagai hipotesis untuk melakukan p

\section*{8.4 Sample Code / Pseudocode}

```

\begin{verbatim}
ID3(examples, attributes):
    Jika semua contoh satu kelas → return kelas
    Jika atribut habis → return kelas mayoritas
    Hitung gain tiap atribut
    Pilih atribut dengan gain terbesar sebagai root
    Untuk tiap nilai atribut:
        Buat cabang dan panggil ID3 secara rekursif

```

\section*{8.5 Rangkuman Inti Bab}

Bab ini menjelaskan bahwa Machine Learning menghasilkan model, Decision

```

\documentclass[11pt,a4paper]{article}

```

```

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[indonesian]{babel}
\usepackage{amsmath, amssymb}
\usepackage{geometry}
\usepackage{enumitem}
\usepackage{hyperref}
\geometry{margin=2.2cm}

\begin{document}

\section*{9. Dari Linear Regression ke Neural Network (Feedforward, MSE)}

\section*{9.1 Deskripsi Singkat Pembahasan}
Materi ini membahas transisi konsep dari \textit{linear regression} (fu

\section*{9.2 Mindmap / Taksonomi (Ringkas)}
\begin{itemize}[leftmargin=1.2em]
  \item \textbf{Linear Regression}
  \begin{itemize}
    \item Persamaan:  $y = mx + c$ 
    \item Notasi ML:  $f(x) = w_1x_1 + w_0$  (bias)
    \item Bias trick:  $x_0 = 1 \rightarrow f(x) = w_0x_0 + w_1x_1$ 
  \end{itemize}
\end{itemize}

\item \textbf{Error dan Loss}
\begin{itemize}
  \item Delta/error:  $\Delta = y - \hat{y}$ 
  \item Kuadrat error:  $(y - \hat{y})^2$  (hindari negatif)
  \item MSE (cost):  $\mathrm{MSE} = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ 
\end{itemize}

```



```

\item \textbf{Multivariate Regression}
\begin{itemize}
  \item Fitur banyak:  $x_1, x_2, \dots, x_n$ 
  \item Model umum:  $f(x) = \sum_{i=0}^n w_{ix_i}$  dengan  $x_0=1$ 
\end{itemize}

\item \textbf{Neural Network (1 Hidden Layer)}
\begin{itemize}
  \item Layer: Input  $\rightarrow$  Hidden  $\rightarrow$  Output
  \item Neuron hidden: kombinasi linear dari input
  \item Output: kombinasi linear dari neuron hidden
  \item Bobot bertambah cepat  $\rightarrow$  komputasi berat
\end{itemize}

\item \textbf{Training (SGD) dan Propagasi}
\begin{itemize}
  \item Inisialisasi bobot random
  \item \textit{Feedforward}: hitung  $\hat{y}$ 
  \item Hitung MSE
  \item Update bobot:  $w \leftarrow w - \alpha \frac{\partial J}{\partial w}$ 
  \item Ulang per epoch hingga MSE kecil/konvergen
\end{itemize}

\item \textbf{Data dan Preprocessing}
\begin{itemize}
  \item Split:  $X$  (fitur) dan  $Y$  (label/target)
  \item Normalisasi (0--1), encoding kategori, shuffle
  \item Istilah: \textit{instance} = 1 baris, \textit{dataset} = kump
\end{itemize}
\end{itemize}

```

\section*{9.3 Penjelasan Detail}

\subsection*{9.3.1 Linear Regression sebagai Neuron Sederhana}

Linear regression adalah fungsi linear:

$$y = mx + c$$

Dalam notasi ML:

$$\hat{y} = f(x) = w_1x_1 + w_0$$

Bias trick:

$$x_0 = 1 \rightarrow \hat{y} = w_0x_0 + w_1x_1$$

\subsection*{9.3.2 Univariate vs Multivariate}

\textbf{Univariate}: hanya 1 fitur.\

\textbf{Multivariate}: banyak fitur:

$$\hat{y} = \sum_{i=0}^n w_ix_i \quad \text{dengan } x_0=1$$

\subsection*{9.3.3 Error (Delta) dan Mean Squared Error (MSE)}

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

\subsection*{9.3.4 Neural Network dengan 1 Hidden Layer}

Input \rightarrow Hidden \rightarrow Output.

$\left[\right.$

$N_j = \sum_{i=0}^d w_{ij} x_i, \quad \text{quad}$

$\hat{y} = \sum_j v_j N_j$

$\left. \right]$

Jumlah bobot meningkat cepat seiring bertambahnya neuron.

$\text{\textbackslash subsection*{9.3.5 Intuisi Training dengan SGD}}$

$\text{\textbackslash begin\{enumerate\}}$

$\text{\textbackslash item}$ Inisialisasi bobot acak

$\text{\textbackslash item}$ Hitung prediksi ($\text{\textbackslash textit\{feedforward\}}$)

$\text{\textbackslash item}$ Hitung error dan MSE

$\text{\textbackslash item}$ Update bobot:

$\left[\right.$

$w \rightarrow w - \alpha \frac{\partial J}{\partial w}$

$\left. \right]$

$\text{\textbackslash item}$ Ulangi per epoch hingga konvergen

$\text{\textbackslash end\{enumerate\}}$

$\text{\textbackslash subsection*{9.3.6 Data, Instance, dan Preprocessing}}$

$\text{\textbackslash begin\{itemize\}}$

$\text{\textbackslash item}$ $\text{\textbackslash textbf\{Instance\}}$: 1 baris data

$\text{\textbackslash item}$ $\text{\textbackslash textbf\{Dataset\}}$: kumpulan instance

$\text{\textbackslash item}$ Proses umum: split X/Y , shuffle, normalisasi, encoding

$\text{\textbackslash end\{itemize\}}$

$\text{\textbackslash section*{9.4 Sample Pseudocode}}$

$\text{\textbackslash subsection*{9.4.1 Feedforward (1 Hidden Layer)}}$

$\text{\textbackslash begin\{verbatim\}}$

For each data:

```
hitung neuron hidden
hitung output  $\hat{y}$ 
hitung error =  $y - \hat{y}$ 
update bobot
```

```
\subsection*{9.4.2 Split Data}
```

```
\begin{verbatim}
```

```
Load CSV
```

```
Shuffle data
```

```
X = fitur
```

```
Y = target
```

```
\section*{9.5 Kesimpulan Inti}
```

```
\begin{itemize}
```

```
  \item Linear regression = neuron paling sederhana.
```

```
  \item Bias dapat dimasukkan lewat  $x_0=1$ .
```

```
  \item Multivariate regression:  $\sum w_{ix_i}$ .
```

```
  \item Neural network menambah hidden layer  $\rightarrow$  bobot dan koefisien.
```

```
  \item Error diukur dengan delta dan MSE.
```

```
  \item Training dilakukan dengan SGD hingga MSE menurun.
```

```
\end{itemize}
```

```
\section*{10. Regression, Neural Network (MLP), Preprocessing, dan Pengujian}
```

```
\section*{10.1 Deskripsi Singkat Pembahasan}
```

```
Bagian ini membahas alur dasar machine learning mulai dari \textit{regresi linier}
```

```
\section*{10.2 Mindmap dan Taksonomi (Struktur Materi)}
```

```
\begin{verbatim}
```

Machine Learning Pipeline

- A. Model Dasar
 - 1. Regression
 - Univariate (1 fitur + bias)
 - Multivariate (banyak fitur + bias)
 - 2. Neural Network (MLP)
 - Feedforward (input -> hidden -> output)
 - Hidden layer bisa banyak
 - Output bisa 1 (binary) atau lebih dari 1 (multi-class)
- B. Training (Membangun model $f(x)$)
 - Inisialisasi bobot (weight)
 - Forward pass (hitung prediksi)
 - Hitung loss (mis. MSE)
 - Update bobot (mis. SGD/gradient descent)
- C. Preprocessing (Sebelum training utama)
 - Encoding (teks/kategori -> angka)
 - Label encoding (0,1,2,...)
 - One-hot encoding (0/1)
 - Normalization/Scaling (min-max, dsb.)
 - Missing value handling
 - Outlier handling
 - Feature engineering
 - Imbalanced data
- D. Deep Learning (Pengantar)
 - Neural network lebih besar (lebih banyak layer/neuron)
 - Bisa ada pretraining
 - CNN untuk gambar
 - NLP: teks harus jadi vektor (embedding)

\section*{10.3 Penjelasan Inti}

\subsection*{10.3.1 Regression (Univariate dan Multivariate)}

\begin{itemize}

\item \textbf{Univariate regression}: satu fitur utama x_1 dan bias

\[

$$z = w_0x_0 + w_1x_1$$

\]

\item \textbf{Multivariate regression}: banyak fitur x_1, x_2, \dots

\[

$$z = \sum_{i=0}^n w_i x_i$$

\]

\item Untuk klasifikasi, nilai z sering dimasukkan ke fungsi aktiva

\end{itemize}

\subsection*{10.3.2 Neural Network (MLP / Feedforward)}

\begin{itemize}

\item Struktur: Input \rightarrow Hidden layer (satu atau banyak) \rightarrow Output.

\item Setiap neuron menghitung jumlah tertimbang lalu aktivasi:

\[

$$a = \sigma\left(\sum_i w_i x_i\right)$$

\]

\item Output bisa:

\begin{itemize}

\item Satu neuron untuk klasifikasi biner.

\item Beberapa neuron untuk klasifikasi multi-kelas.

\end{itemize}

\end{itemize}

\subsection*{10.3.3 Training dan Model $f(x)$ }

```

\begin{itemize}
  \item \textbf{Forward pass}: menghitung prediksi  $\hat{y}$ .
  \item \textbf{Loss/Error}: contoh Mean Squared Error (MSE):
  \[
    \text{MSE} = \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2
  \]
  \item \textbf{Update bobot}: menggunakan metode seperti SGD atau grad
\end{itemize}

```

\subsection*{10.3.4 Preprocessing (Tahap Wajib)}

```

\begin{itemize}
  \item \textbf{Encoding}: mengubah data kategori/teks menjadi angka (1
  \item \textbf{Normalization/Scaling}: menyetarakan skala fitur (misal
  \item \textbf{Missing value handling}: mengisi nilai kosong dengan me
  \item \textbf{Outlier handling}: menangani data ekstrem yang dapat me
  \item \textbf{Feature engineering}: membuat fitur baru yang lebih inf
  \item \textbf{Imbalanced dataset}: menangani perbedaan jumlah kelas (
\end{itemize}

```

\subsection*{10.3.5 Pengantar Deep Learning}

```

\begin{itemize}
  \item Deep learning adalah neural network dengan banyak layer dan neu
  \item CNN umum digunakan untuk data gambar.
  \item Pada NLP, teks harus diubah menjadi vektor atau embedding sebel
\end{itemize}

```

\section*{10.4 Kesimpulan Inti}

```

\begin{itemize}
  \item Regression adalah model dasar untuk membangun fungsi prediksi $
  \item Neural network (MLP) memperluas regression dengan hidden layer.
  \item Training dilakukan dengan forward pass, perhitungan loss, dan u

```

\item Preprocessing wajib karena data dunia nyata tidak selalu bersih

\item Deep learning adalah pengembangan neural network yang lebih bes

\end{itemize}

\section*{11. Machine Learning Engineering, Pipeline, Taksonomi Metode,

\section*{11.1 Deskripsi Singkat Pembahasan}

Bagian ini membahas cara memandang \textit{machine learning} sebagai du

\section*{11.2 Mindmap dan Taksonomi (Struktur Materi)}

\begin{verbatim}

Machine Learning

- Perspektif

- Science: memahami pola, sebab-akibat, prediksi

- Engineering: membangun solusi/sistem ML

- Workflow

- Data Exploration -> Data Analytics -> Experiment -> Development -> I

- Pipeline

- Training: dataset -> preprocessing -> training -> model

- Inference: data baru -> preprocessing sama -> model -> prediksi

- Jenis Data

- Flat (tabular)

- Time series (berurutan waktu)

- Taksonomi Metode

- Supervised

- Unsupervised

- Semi-supervised

- Deep Learning
- Reinforcement Learning
- Ensemble

- Unsupervised (Detail)
 - Clustering: K-Means, GMM
 - Dimension Reduction: PCA, Autoencoder
 - Distance Function: Euclidean, Manhattan

\section*{11.3 Penjelasan Detail}

\subsection*{11.3.1 ML sebagai Science vs Engineering}

\begin{itemize}

\item \textbf{Science}: fokus memahami data, hubungan variabel, pola,

\item \textbf{Engineering}: fokus membangun sistem atau produk yang m

\end{itemize}

\subsection*{11.3.2 Tahapan Membangun Solusi ML}

\begin{enumerate}

\item \textbf{Data Exploration}: memahami struktur data, kualitas, ni

\item \textbf{Data Analytics}: menganalisis distribusi, pola, korelas

\item \textbf{Experiment}:

\begin{itemize}

\item Preprocessing data

\item Feature engineering

\item Pemilihan model

\item Training dan testing

\end{itemize}

\item \textbf{Development}: membangun layanan atau aplikasi berbasis m

\item \textbf{Deployment}: menyiapkan model agar siap digunakan di li

\end{enumerate}

\subsection*{11.3.3 Pipeline: Training vs Inference}

\begin{itemize}

\item \textbf{Training}:

\[

\text{Dataset} \rightarrow \text{Preprocessing} \rightarrow \text{T

\]

\item \textbf{Inference}:

\[

$X_{\{\text{baru}\}}$ \rightarrow Preprocessing (sama) \rightarrow

\]

\item Preprocessing harus sama agar input konsisten dengan model.

\end{itemize}

\subsection*{11.3.4 Flat Table vs Time Series}

\begin{itemize}

\item \textbf{Flat (Tabular)}: setiap baris berdiri sendiri, tidak be

\item \textbf{Time Series}: data berurutan waktu, sering memiliki tre

\end{itemize}

\subsection*{11.3.5 Ensemble}

\begin{itemize}

\item Ensemble menggabungkan banyak model untuk meningkatkan akurasi.

\item Contoh: \textbf{Random Forest} menggunakan banyak decision tree

\end{itemize}

\subsection*{11.3.6 Unsupervised Learning}

\begin{itemize}

\item Tidak memiliki label target.

\item Tujuan utama:

\begin{enumerate}

```

\item Clustering
\item Transformasi data
\item Dimension reduction
\end{enumerate}
\end{itemize}

\subsection*{11.3.7 Dimension Reduction (PCA / Autoencoder)}
\begin{itemize}
\item Mengurangi jumlah fitur agar data lebih ringkas.
\item Sebagian informasi diringkas, tetapi komponen utama tetap diper
\end{itemize}

\subsection*{11.3.8 KNN vs K-Means}
\begin{itemize}
\item \textbf{KNN (Supervised)}:
\begin{itemize}
\item  $K$  = jumlah tetangga terdekat.
\item Prediksi berdasarkan voting mayoritas.
\end{itemize}
\item \textbf{K-Means (Unsupervised)}:
\begin{itemize}
\item  $K$  = jumlah cluster.
\item Proses: assign ke centroid terdekat lalu update centroid samp
\end{itemize}
\end{itemize}

\subsection*{11.3.9 Distance Function}
\begin{itemize}
\item \textbf{Euclidean}:
\begin{aligned}
& D_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}
\end{aligned}
\end{itemize}

```

```

\]
\item \textbf{Manhattan}:
\[
D_M(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^d \lvert x_i - y_i \rvert
\]
\end{itemize}

\subsection*{11.3.10 GMM (Gaussian Mixture Model)}
\begin{itemize}
\item GMM memodelkan data sebagai campuran beberapa distribusi Gaussi.
\item Setiap Gaussian mewakili satu kelompok atau cluster.
\item Evaluasi menggunakan likelihood, bukan error klasifikasi.
\end{itemize}

\section*{11.4 Sample Code / Pseudocode}

\subsection*{11.4.1 Pipeline Training vs Inference}
\begin{verbatim}
# TRAINING
X_train, y_train = load_dataset()
prep = fit_preprocessing(X_train)
X_train_p = prep.transform(X_train)
model = train_model(X_train_p, y_train)
save(prepare, model)

# INFERENCE
X_new = load_new_data()
prep, model = load_saved_objects()
X_new_p = prep.transform(X_new)
y_pred = model.predict(X_new_p)

```

```

\subsection*{11.4.2 K-Means}
\begin{verbatim}
initialize K centroids randomly
repeat until convergence:
    assign each point to nearest centroid
    update each centroid = mean of assigned points
return centroids and cluster labels

```

```

\subsection*{11.4.3 KNN}
\begin{verbatim}
given K and labeled dataset:
for new point x:
    compute distance to all training points
    take K nearest neighbors
    return majority vote of their labels

```

```

\subsection*{11.4.4 GMM}
\begin{verbatim}
choose number of components K
initialize Gaussian parameters
repeat until convergence:
    E-step: compute responsibilities
    M-step: update parameters
return mixture model and soft clusters

```

```

\section*{11.5 Kesimpulan Inti}
\begin{itemize}

```

```

\item ML dapat dipandang sebagai sains dan engineering.
\item Solusi ML dibangun melalui fase eksplorasi hingga deployment.
\item Pipeline training dan inference harus konsisten terutama pada p
\item Data bisa berbentuk tabular atau time series.
\item Unsupervised learning fokus pada clustering dan dimension reduc
\item K-Means dan KNN berbeda tujuan dan pendekatan.
\item GMM melakukan clustering berbasis distribusi Gaussian.
\end{itemize}

```

```

\section*{12. Evaluation Metrics, Overfitting, Underfitting, dan Model

```

```

\section*{12.1 Deskripsi Singkat Pembahasan}

```

Bagian ini membahas bagaimana mengevaluasi performa model machine learn

```

\section*{12.2 Mindmap dan Taksonomi (Struktur Materi)}

```

```

\begin{verbatim}

```

Model Evaluation and Validation

- A. Evaluation Metrics

- Regression:

- MSE (Mean Squared Error)
- RMSE (Root Mean Squared Error)
- MAE (Mean Absolute Error)
- R^2 (Coefficient of Determination)

- Classification:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

- B. Model Quality

- Underfitting:
 - Model terlalu sederhana
 - Tidak menangkap pola data
- Overfitting:
 - Model terlalu kompleks
 - Menghafal data training
 - Performa jelek di data baru
- C. Validation Techniques
 - Train-Test Split
 - Validation Set
 - Cross-Validation (k-fold)
- D. Tujuan Akhir
 - Generalisasi model
 - Performa stabil di data nyata

\section*{12.3 Penjelasan Detail}

\subsection*{1) Evaluation Metrics untuk Regression}

Pada regresi, tujuan utama adalah meminimalkan selisih antara nilai akt

\begin{itemize}

\item \textbf{MSE (Mean Squared Error)}:

\[

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

\]

\item \textbf{RMSE (Root Mean Squared Error)}:

\[

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

\]

```

\item \textbf{MAE (Mean Absolute Error)}:
\[
\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|
\]
\item \textbf{$R^2$ Score}:
    Mengukur seberapa baik model menjelaskan variasi data, dengan nilai a
\end{itemize}

```

```

\subsection*{2) Evaluation Metrics untuk Classification}
Untuk klasifikasi, hasil prediksi biasanya berupa label (0/1 atau kelas)
\begin{itemize}
    \item \textbf{Accuracy}: proporsi prediksi yang benar.
    \item \textbf{Precision}: seberapa tepat prediksi positif.
    \item \textbf{Recall}: seberapa banyak data positif yang berhasil ter
    \item \textbf{F1-Score}: rata-rata harmonik antara precision dan reca
\end{itemize}

```

Semua metrik ini biasanya dirangkum dalam sebuah **confusion matrix**

```

\subsection*{3) Underfitting}
\textit{Underfitting} terjadi ketika model terlalu sederhana sehingga t
Ciri-ciri:
\begin{itemize}
    \item Error training tinggi
    \item Error testing juga tinggi
    \item Model tidak belajar dengan baik
\end{itemize}

```

Contoh: menggunakan regresi linear untuk data yang pola aslinya non-linear

```

\subsection*{4) Overfitting}
\textit{Overfitting} terjadi ketika model terlalu kompleks sehingga san
Ciri-ciri:

```



```

\begin{itemize}
  \item Error training sangat kecil
  \item Error testing besar
  \item Model seperti \menghafal" data training
\end{itemize}

```

Contoh: neural network dengan terlalu banyak neuron dan layer pada data

```

\subsection*{5) Model Validation}

```

Untuk memastikan model mampu melakukan generalisasi, digunakan teknik v

```

\begin{itemize}
  \item \textbf{Train-Test Split}: data dibagi menjadi data latih dan d
  \item \textbf{Validation Set}: sebagian data latih dipisahkan untuk t
  \item \textbf{Cross-Validation (k-fold)}: data dibagi menjadi $k$ bag
\end{itemize}

```

```

\section*{12.4 Sample Pseudocode}

```

```

\subsection*{A) Pseudocode Train-Test Split dan Evaluasi}

```

```

\begin{verbatim}

```

```

Load dataset -> data

```

```

Shuffle data

```

```

Split data:

```

```

    train_data = 80%

```

```

    test_data = 20%

```

```

Train model using train_data

```

```

Predict using test_data -> y_hat

```

```

Compute evaluation metric (e.g., MSE or Accuracy)

```

```

Print metric result

```

```
\subsection*{B) Pseudocode k-Fold Cross Validation}
```

```
\begin{verbatim}
```

```
Load dataset -> data
```

```
Shuffle data
```

```
Choose k
```

```
Split data into k folds
```

```
For each fold i in 1..k:
```

```
    validation_set = fold i
```

```
    training_set = all other folds
```

```
    Train model on training_set
```

```
    Evaluate model on validation_set
```

```
    Store evaluation score
```

```
Average all evaluation scores -> final performance
```

```
\section*{12.5 Kesimpulan Inti}
```

```
\begin{itemize}
```

```
    \item Evaluasi model penting untuk mengukur kualitas prediksi.
```

```
    \item Metrik regresi (MSE, RMSE, MAE,  $R^2$ ) dan klasifikasi (accuracy)
```

```
    \item Underfitting berarti model terlalu sederhana; overfitting berarti
```

```
    \item Model validation seperti train-test split dan cross-validation
```

```
    \item Tujuan utama adalah mendapatkan model yang seimbang: cukup kompleks
```

```
\end{itemize}
```

```
% =====
```

```
% BAB 13
```

% =====

\section*{13. Clustering (K-Means) dan Gaussian Mixture Model}

\section*{13.1 Deskripsi Singkat Pembahasan}

Materi ini membahas inti \textit{unsupervised learning} untuk \textbf{c

(1) \textbf{K-Means} yang mengelompokkan data berdasarkan \textbf{jarak

(2) \textbf{Gaussian Mixture Model (GMM)} yang memodelkan data sebagai

Dibahas pula perbedaan konsep K pada KNN vs K-Means, pilihan \textit{c

\section*{13.2 Mindmap dan Taksonomi}

\begin{itemize}[leftmargin=1.2em]

\item {Unsupervised Learning}

\begin{itemize}

\item Clustering tanpa label

\item Interpretasi hasil cluster

\end{itemize}

\item \textbf{KNN vs K-Means}

\begin{itemize}

\item KNN: supervised, berbasis tetangga

\item K-Means: unsupervised, berbasis cluster

\end{itemize}

\item \textbf{K-Means}

\begin{itemize}

\item Inisialisasi centroid

\item Assignment dan update

\end{itemize}

\item \textbf{Distance Function}

```

\begin{itemize}
  \item Euclidean
  \item Manhattan
\end{itemize}

\item \textbf{GMM}
\begin{itemize}
  \item Campuran Gaussian
  \item Parameter  $\mu$ ,  $\sigma$ ,  $\pi$ 
\end{itemize}

\item \textbf{Dimensionality Reduction}
\begin{itemize}
  \item PCA
  \item Autoencoder
\end{itemize}
\end{itemize}

```

\section*{13.3 Clustering (Unsupervised)}

Clustering bertujuan mengelompokkan data berdasarkan kemiripan tanpa la

\section*{13.4 KNN vs K-Means}

KNN menggunakan K tetangga terdekat untuk voting kelas, sedangkan K-M

\section*{13.5 K-Means: Assignment dan Update Centroid}

Langkah utama K-Means:

```

\begin{enumerate}[leftmargin=1.5em]
  \item Menentukan jumlah cluster  $K$ 
  \item Inisialisasi centroid
  \item Assignment ke centroid terdekat
  \item Update centroid

```

```

\item Iterasi sampai stabil
\end{enumerate}

```

\section*{13.6 Distance Function}

```

\[
d_{\text{euclid}}(x,y)=\sqrt{\sum_{j=1}^D (x_j-y_j)^2}
\]

```

```

\[
d_{\text{manhattan}}(x,y)=\sum_{j=1}^D |x_j-y_j|
\]

```

\section*{13.7 Stopping Criteria}

Algoritma berhenti jika centroid stabil, keanggotaan tidak berubah, atau

\section*{13.8 Distribusi Data}

Distribusi menggambarkan frekuensi nilai fitur dan sering memiliki satu

\section*{13.9 Gaussian Mixture Model (GMM)}

```

\[
p(x)=\sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)
\]

```

\section*{13.10 Expectation-Maximization (EM)}

EM terdiri dari:

```

\begin{itemize}
\item \textbf{E-step}: hitung responsibility
\item \textbf{M-step}: update parameter
\end{itemize}

```

\section*{13.11 Dimensionality Reduction}

Digunakan untuk mengurangi jumlah fitur:

```
\begin{itemize}
```

```
  \item PCA
```

```
  \item Autoencoder
```

```
\end{itemize}
```

```
\section*{13.12 Sample Code / Pseudocode}
```

```
\textbf{Pseudocode K-Means}
```

```
\begin{verbatim}
```

```
Init centroid
```

```
Assignment ke centroid terdekat
```

```
Update centroid
```

```
Ulangi sampai stabil
```

```
\textbf{Pseudocode EM}
```

```
\begin{verbatim}
```

```
E-step: hitung responsibility
```

```
M-step: update parameter
```

```
Ulangi sampai likelihood stabil
```

```
\section*{13.13 Kesimpulan Inti}
```

```
\begin{itemize}
```

```
  \item Clustering merupakan unsupervised learning.
```

```
  \item K-Means berbasis jarak dan centroid.
```

```
  \item GMM bersifat probabilistik dan dioptimalkan dengan EM.
```

```
  \item PCA dan autoencoder membantu menangani fitur berdimensi tinggi.
```

```
\end{itemize}
```

```

% =====
\section*{14. Clustering: K-Means, GMM, EM, Tensor, dan Komputasi GPU}

% =====
\section*{14.1 Deskripsi Singkat Pembahasan}
Bab ini membahas metode \textit{unsupervised learning} untuk \textbf{clustering}
khususnya \textbf{K-Means} dan \textbf{Gaussian Mixture Model (GMM)}.
K-Means mengelompokkan data berdasarkan jarak ke centroid, sedangkan GMM
memodelkan data sebagai campuran beberapa distribusi Gaussian dan dioptimalkan
menggunakan algoritma \textbf{Expectation-Maximization (EM)}.
Selain itu dibahas konsep \textbf{tensor} sebagai representasi data multivariat
serta peran \textbf{GPU} dalam mempercepat komputasi matriks dan tensor
pada machine learning.

% =====
\section*{14.2 Mindmap dan Taksonomi Materi}
\begin{itemize}[leftmargin=1.2em]

\item \textbf{Unsupervised Learning}
\begin{itemize}
\item Clustering (tanpa label)
\end{itemize}

\item \textbf{K-Means}
\begin{itemize}
\item Berbasis jarak ke centroid
\item Assignment dan update centroid
\end{itemize}

\item \textbf{Gaussian Mixture Model (GMM)}
\begin{itemize}

```

```

\item Berbasis probabilitas
\item Campuran beberapa Gaussian
\item Optimasi menggunakan EM
\end{itemize}

\end{itemize}

% =====
\section*{14.3 K-Means: Konsep dan Proses Iteratif}
K-Means adalah metode clustering berbasis jarak.
Algoritma ini memulai dengan memilih centroid awal secara acak,
kemudian mengelompokkan setiap data ke centroid terdekat.
Setelah itu, centroid diperbarui sebagai rata-rata anggota cluster.
Proses ini diulang sampai centroid stabil atau iterasi maksimum tercapai

% =====
\section*{14.4 Gaussian Mixture Model (GMM)}
GMM memandang data sebagai campuran beberapa distribusi Gaussian.
Setiap Gaussian merepresentasikan satu kelompok data dengan parameter
mean ( $\mu$ ), kovarians ( $\Sigma$ ), dan bobot ( $\pi$ ).

Model GMM dirumuskan sebagai:
\[\begin{aligned} p(\mathbf{x}) &= \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k), \\ \sum_{k=1}^K \pi_k &= 1 \end{aligned}\]

% =====
\section*{14.5 Expectation-Maximization (EM)}
EM digunakan untuk melatih GMM dengan memaksimalkan \textit{log-likelihood}

```


Algoritma EM terdiri dari dua langkah:

```
\begin{itemize}
```

```
\item \textbf{E-step}: menghitung probabilitas data berasal dari seti
```

```
\item \textbf{M-step}: memperbarui parameter  $\mu$ ,  $\sigma$ , dan
```

```
\end{itemize}
```

Iterasi dihentikan ketika perubahan $\log\text{-likelihood}$ sangat kecil

```
% =====
```

```
\section*{14.6 Reduksi Dimensi}
```

Pada data berdimensi tinggi, clustering dan visualisasi menjadi sulit.

Reduksi dimensi seperti **PCA** dan **Autoencoder**

digunakan untuk memadatkan fitur dengan tetap mempertahankan informasi meskipun sebagian informasi dapat hilang.

```
% =====
```

```
\section*{14.7 Tensor dan Komputasi GPU}
```

Tensor adalah generalisasi array ke dimensi lebih tinggi.

Sebagian besar komputasi machine learning melibatkan operasi matriks dan

GPU memiliki banyak core sehingga mampu melakukan komputasi paralel

dan mempercepat proses training model dibanding CPU.

```
% =====
```

```
\section*{14.8 Pseudocode K-Means}
```

```
\begin{verbatim}
```

```
Input: data X, jumlah cluster K
```

```
Output: centroid dan cluster
```

```
Init centroid secara acak
```

```
Ulangi:
```

```
    Assignment data ke centroid terdekat
```

```
    Update centroid sebagai rata-rata cluster
```

Sampai centroid stabil

14.9 Pseudocode EM untuk GMM

Input: data X , jumlah Gaussian K

Output: parameter π , μ , Σ

Init parameter

Ulangi:

 E-step: hitung responsibility

 M-step: update parameter

Sampai log-likelihood stabil

14.10 Kesimpulan Inti

- K-Means adalah clustering berbasis jarak dan centroid.
- GMM adalah clustering berbasis probabilitas dan distribusi Gaussian.
- EM digunakan untuk melatih GMM melalui E-step dan M-step.
- Reduksi dimensi membantu analisis data berdimensi tinggi.
- Tensor dan GPU berperan penting dalam komputasi machine learning modern.

Penutup

Dokumen ini merangkum konsep-konsep inti dalam machine learning, mulai dari dasar regresi, neural network, evaluasi model, hingga metode *unsupervised learning* seperti K-Means dan Gaussian Mixture Model (GMM).

Diharapkan rangkuman ini dapat membantu pembaca memahami alur berpikir machine learning secara terstruktur, baik dari sisi teori maupun implementasi, serta menjadi referensi ringkas untuk pembelajaran dan pengembangan model di dunia nyata.