**Rangkuman Materi Perkuliahan Machine Learning Kelompok 14**

Fitri Ambar Sari1, Destri Komalasari2

1Program Studi Teknik Informatika, STMIK Tazkia

2 Program Studi Sistem Informasi, STMIK Tazkia

E-mail :

[241552010023.fitriambar@student.stmik.tazkia.ac.id1](mailto:241552010023.fitriambar@student.stmik.tazkia.ac.id1)

‎[241572010023.destri@student.stmik.tazkia.ac.id](mailto:‎241572010023.destri@student.stmik.tazkia.ac.id)[2](mailto:‎241572010023.destri@student.stmik.tazkia.ac.id)

**Abstrak**

Pembelajaran Mesin (Machine Learning) adalah salah satu cabang utama kecerdasan buatan (Artificial Intelligence) dan memainkan peran penting dalam pemrosesan data dan pengambilan keputusan berdasarkan pola. Tulisan ini bertujuan untuk meringkas dan menjelaskan konsep dasar hingga lanjutan dari pelajaran mesin yang dipelajari di perguruan tinggi, mulai dari pengantar ai, jenis pelajaran, hingga algoritma dan model umum. Metode yang digunakan dalam penyusunan ringkasan ini adalah melalui studi literatur dan analisis materi kuliah, yang meliputi pembelajaran terawasi (supervised learning), pembelajaran tak terawasi (unsupervised learning), dan pengantar pembelajaran terdalam (deep learning). Pembahasan berfokus pada alur kerja pembelajaran mesin, proses pembuatan model, dan evaluasi kerja menggunakan pendekatan seperti legresi liniar pohon keputusan (ID3 dan C4.5), dan jaringan saraf. Ringkasan ini juga mencakup pentingnya pra pemrosesan data, pemahaman target dalam suatu dataset, dan hubungan antara pembelajaran mesin dengan kebutuhan dan regulasi industri. Hasil ringkasan ini menunjukan bahwa pemahaman konsep dasar, algoritma dan proses pembelajaran model sangat penting sebelum beralih ke pengembangan sistem berbasis pembelajaran mendalam. Dengan ringkasan ini, diharapkan mahasiswa akan memiliki pemahaman yang komprehensif tentang pembelajaran mesin dan mampu menerapkannya baik secara akademis maupun praktis.

**Kata kunci :** Machine Learning, Supervised Learning, Decision Tree, Neural Network, Deep Learning

**Summary of machine learning lecture material group 14**

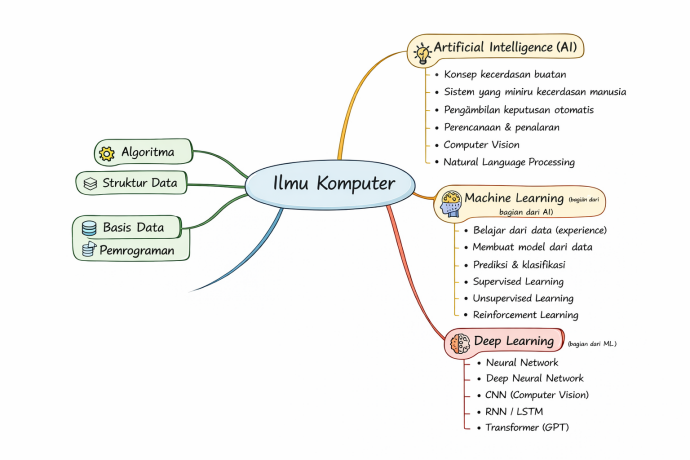
***Abstract***

*Machine Learning is one of the main branches of Artificial Intelligence and plays a crucial role in data processing and decision-making based on patterns. This writing aims to summarize and explain the basic to advanced concepts of Machine Learning that are studied in college, starting from an introduction to AI, types of learning, up to the application of common algorithms and models. The method used in preparing this summary is through literature study and analysis of course materials, which include supervised learning, unsupervised learning, and an introduction to deep learning. The discussion focuses on the workflow of Machine Learning, the model-building process, and performance evaluation using approaches such as linear regression, decision tree (ID3 and C4.5), and neural networks. It also covers the importance of data preprocessing, understanding the target in a dataset, and the connection between Machine Learning and industrial needs and regulations. The results of this summary show that understanding basic concepts, algorithms, and the model learning process is essential before moving into the development of deep learning-based systems. With this summary, it is hoped that students will have a comprehensive understanding of Machine Learning and be able to apply it both academically and practically.*

**Rangkuman Pertemuan 1**

1. **Deskripsi Singkat Materi**

Materi pada pertemuan pertama membahas pengenalan machine learning, konsep dasar kecerdasan buatan, serta ruang lingkup profesi di bidang data dan AI. Selain itu, materi ini juga memberikan gambaran mengenai perkembangan dan tren teknologi di masa depan sebagai bekal sebelum memasuki pembahasan teknis pada pertemuan selanjutnya.



1. **Penjelasan Detail Materi**

Machine Learning merupakan bagian dari kecerdasan buatan (Artificial Intelligence) yang membahas bagaimana komputer bisa belajar dari data dan pengalaman. Dengan machine learning, komputer tidak perlu lagi diprogram satu per satu untuk setiap masalah, tetapi bisa belajar sendiri dari data yang diberikan.

Pada pertemuan pertama, mahasiswa diperkenalkan dengan konsep dasar machine learning serta hubungannya dengan kecerdasan buatan. Dijelaskan bahwa kecerdasan buatan adalah bidang ilmu yang bertujuan untuk membuat sistem komputer memiliki kemampuan seperti manusia, misalnya mampu berpikir, belajar, mengambil keputusan, dan menyelesaikan masalah. Machine learning menjadi salah satu teknologi utama dalam pengembangan AI karena memungkinkan komputer untuk belajar secara otomatis dari data.

Materi juga menjelaskan pengertian machine learning menurut Tom Mitchell, yang menyatakan bahwa sebuah komputer dikatakan belajar jika ia belajar dari pengalaman (experience) untuk mengerjakan suatu tugas (task) dengan ukuran performa tertentu (performance), dan performanya meningkat seiring bertambahnya pengalaman. Dalam konteks machine learning, experience diartikan sebagai data. Data inilah yang digunakan komputer sebagai bahan pembelajaran untuk membentuk pola dan pengetahuan.

Data merupakan kumpulan fakta yang dapat berupa angka, teks, gambar, suara, atau bentuk informasi lainnya. Dari data tersebut, sistem akan membangun model yang digunakan untuk melakukan prediksi, klasifikasi, atau pengambilan keputusan. Oleh karena itu, kualitas data sangat berpengaruh terhadap hasil pembelajaran mesin. Jika data yang digunakan baik dan lengkap, maka model yang dihasilkan juga akan lebih akurat.

Selain membahas konsep dasar, materi pertemuan pertama juga membahas ruang lingkup bidang data dan kecerdasan buatan serta berbagai profesi yang berkaitan dengan machine learning. Beberapa profesi yang diperkenalkan antara lain Data Scientist, Machine Learning Engineer, AI Engineer, dan Data Analyst. Setiap profesi memiliki peran yang berbeda, tetapi saling berkaitan dalam membangun sistem berbasis data dan AI.

Data Scientist berperan dalam mengolah dan menganalisis data untuk menemukan pola serta menghasilkan informasi yang berguna. Machine Learning Engineer dan AI Engineer bertugas membangun serta mengembangkan model machine learning agar dapat digunakan dalam sebuah aplikasi atau sistem. Sementara itu, Data Analyst berfokus pada analisis data dan penyajian hasil dalam bentuk laporan atau visualisasi.

Materi juga membahas perkembangan dan tren teknologi di masa depan yang semakin mengarah pada pemanfaatan kecerdasan buatan di berbagai bidang, seperti kesehatan, pendidikan, perbankan, e-commerce, dan industri. Penggunaan AI diharapkan dapat membantu manusia dalam bekerja lebih cepat, lebih efisien, dan lebih akurat.

Selain AI, dibahas pula pentingnya keamanan sistem (security) dan teknologi blockchain. Keamanan sistem sangat dibutuhkan untuk melindungi data dari kebocoran dan penyalahgunaan. Blockchain diperkenalkan sebagai teknologi yang mampu menyediakan sistem pencatatan data yang aman, transparan, dan sulit untuk dimanipulasi.

Mahasiswa juga diperkenalkan dengan pentingnya penguasaan dasar matematika, statistika, dan algoritma sebagai fondasi utama dalam memahami machine learning. Matematika dan statistika membantu dalam memahami bagaimana data diolah dan bagaimana model bekerja, sedangkan algoritma menjadi dasar dalam proses pembelajaran mesin.

Materi ini menjadi dasar pembelajaran sebelum memasuki pembahasan yang lebih teknis pada pertemuan-pertemuan berikutnya. Dengan memahami konsep dasar machine learning, ruang lingkup bidang AI, serta tren teknologi di masa depan, mahasiswa diharapkan memiliki gambaran yang jelas mengenai peran dan pentingnya machine learning di era digital saat ini.

1. **Sample Code/Pseudcode**

## 3.1 Pseudocode Konsep Machine Learning

Masukkan dataset sebagai data pelatihan

Tentukan tujuan pembelajaran

Tentukan kriteria evaluasi

Buat model awal

Lakukan perulangan:

Jalankan proses training menggunakan data

Hitung hasil evaluasi model

Jika hasil belum sesuai target:

Sesuaikan parameter model

Sampai model mencapai performa terbaik

## 3.2 Contoh Kode Sederhana (Python – Regresi Linear)

import numpy as np

from sklearn.linear\_model import LinearRegression

Data sebagai experience

X = jumlah jam belajar

y = nilai ujian

X = np.array([[2], [4], [6], [8], [10]])

y = np.array([50, 60, 70, 80, 90])

Membuat model regresi linear

model = LinearRegression()

Melatih model menggunakan data

model.fit(X, y)

Melakukan prediksi nilai jika belajar 7 jam

prediksi\_nilai = model.predict([[7]])

print("Prediksi nilai untuk 7 jam belajar:", prediksi\_nilai)

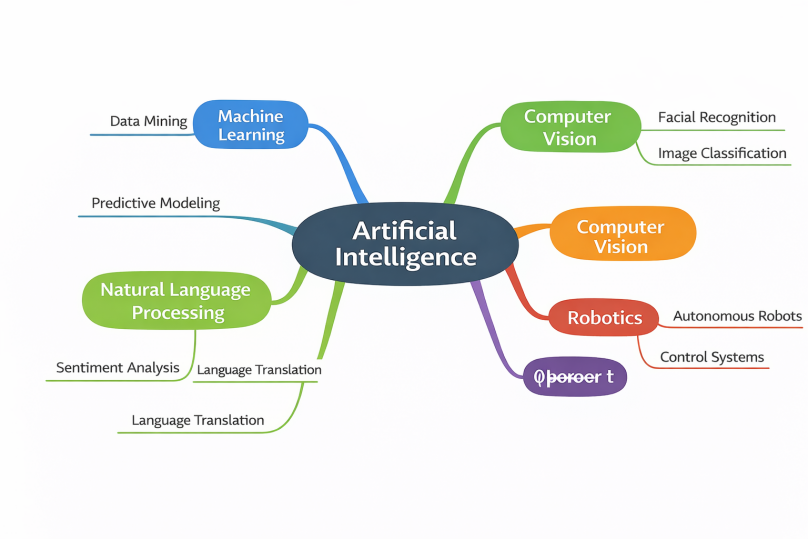
**Rangkuman Pertemuan 2**

1. **Deskripsi Singkat Materi**

Pada pertemuan kedua mata kuliah Machine Learning, pembahasan difokuskan pada bagaimana kecerdasan buatan (Artificial Intelligence/AI) dan machine learning diterapkan dalam dunia nyata, serta bagaimana posisi machine learning dalam peta besar keilmuan AI. Materi ini bertujuan agar mahasiswa tidak hanya memahami algoritma secara teknis, tetapi juga mampu melihat konteks masalah, kebutuhan industri, serta keterkaitan antara teknologi, regulasi, dan dampak sosial.

1. **Penjelasan Detail Materi**

Pembahasan diawali dengan studi kasus implementasi AI di bidang lingkungan, khususnya pada isu perubahan iklim dan ekonomi karbon. Menjelaskan bahwa perubahan iklim merupakan masalah global yang dampaknya sudah terasa langsung, seperti kenaikan permukaan laut yang mengancam wilayah pesisir, termasuk Jakarta. Untuk menekan laju perubahan iklim, muncul regulasi internasional terkait emisi karbon. Dalam konteks ini, perusahaan yang menghasilkan emisi karbon diwajibkan untuk menyeimbangkan emisinya dengan cara membeli kredit karbon. Kredit karbon tersebut merepresentasikan sejumlah karbon yang berhasil diserap oleh alam, misalnya melalui kegiatan reforestasi atau penanaman pohon.



Di sinilah peran teknologi, khususnya AI dan machine learning, menjadi sangat penting. Salah satu contoh nyata yang dibahas adalah platform Jejak.in, yang bergerak di bidang pencatatan dan verifikasi aktivitas karbon. Permasalahan utama dalam bisnis ekonomi karbon bukan hanya soal teknis pencatatan, tetapi juga masalah kepercayaan (trust). Muncul pertanyaan mendasar, seperti apakah pohon benar-benar ditanam, apakah lokasi yang dilaporkan sesuai dengan kenyataan, dan apakah data yang dikirimkan tidak dimanipulasi. Jika kepercayaan ini tidak terjaga, maka seluruh sistem perdagangan karbon akan kehilangan legitimasi.

Untuk menjawab masalah tersebut, AI digunakan sebagai alat bantu validasi. Dengan pendekatan computer vision, sistem dapat menganalisis foto tanaman untuk memastikan bahwa objek yang difoto benar-benar tanaman yang sesuai dengan klaim, misalnya membedakan antara bibit pohon tertentu dengan tanaman lain. Selain itu, AI juga dapat digunakan untuk mendeteksi apakah sebuah foto merupakan foto asli atau hasil manipulasi. Tidak hanya berhenti di level foto, teknologi AI juga diterapkan pada citra satelit untuk menganalisis tutupan lahan (land cover). Dari citra tersebut, sistem dapat memastikan apakah suatu wilayah benar-benar merupakan area reforestasi baru, bukan hutan lama atau perkebunan tertentu yang secara ekologis memiliki daya serap karbon berbeda. Contoh ini menunjukkan bahwa machine learning tidak berdiri sendiri, tetapi selalu berinteraksi dengan domain masalah yang spesifik.

Selanjutnya,bahwa dalam implementasi AI di dunia nyata, pemahaman regulasi dan kepatuhan (compliance) sama pentingnya dengan kemampuan teknis. Dalam konteks global, produk yang masuk ke pasar Eropa, misalnya, akan diperiksa jejak karbonnya. Jika emisi yang dihasilkan tinggi, maka pajak karbon yang dikenakan juga tinggi. Di Indonesia sendiri terdapat Sistem Registri Nasional (SRN) yang digunakan untuk mencatat aktivitas terkait karbon. Selain itu, terdapat pula standar internasional seperti Verra yang mengatur sertifikasi karbon. Dari sisi pengelolaan data, mahasiswa juga diingatkan tentang pentingnya standar keamanan informasi seperti ISO 27001 serta regulasi perlindungan data pribadi seperti Undang-Undang Perlindungan Data Pribadi (UU PDP).

Materi pertemuan kedua juga membahas peta besar keilmuan Artificial Intelligence agar mahasiswa tidak keliru dalam memahami istilah-istilah yang sering digunakan. Artificial Intelligence dijelaskan sebagai bidang paling luas yang mencakup berbagai pendekatan, seperti pencarian (searching), perencanaan (planning), penalaran (reasoning), dan pembelajaran (learning). Machine Learning merupakan bagian dari AI yang secara khusus berfokus pada kemampuan sistem untuk belajar dari data. Di dalam machine learning sendiri terdapat deep learning, yaitu pendekatan yang menggunakan jaringan saraf dengan banyak lapisan (multi-layer neural networks). Perkembangan terbaru dari deep learning melahirkan Generative AI dan Large Language Model (LLM), yang mampu menghasilkan konten baru seperti teks, gambar, atau kode program. Dengan pemetaan ini, mahasiswa diharapkan tidak lagi menyamakan semua istilah AI, ML, dan deep learning sebagai hal yang sama.

Selain aspek teknis dan konseptual, pertemuan ini juga menyinggung aspek akademik dan profesional yang sangat relevan bagi mahasiswa. Dosen menekankan pentingnya penggunaan GitHub sebagai portofolio, bukan sekadar tempat menyimpan kode. Repository GitHub yang rapi, terdokumentasi dengan baik, dan menunjukkan progres kerja yang jelas dapat menjadi nilai tambah ketika mahasiswa melamar kerja. Tugas-tugas yang dikerjakan selama perkuliahan sebaiknya dipandang sebagai bagian dari portofolio, bukan hanya kewajiban akademik semata. Untuk komunikasi dan diskusi teknis, penggunaan platform seperti Discord juga dianjurkan karena lebih terstruktur dibandingkan aplikasi pesan instan biasa.

Sebagai contoh sederhana penerapan konsep yang dibahas, dapat dibayangkan sebuah sistem AI yang digunakan untuk memvalidasi foto tanaman. Ketika sebuah foto diunggah, sistem akan mengekstraksi ciri visual dari gambar tersebut, kemudian membandingkannya dengan model yang telah dilatih sebelumnya. Dari proses tersebut, sistem dapat menentukan jenis tanaman yang ada di foto dan menilai apakah foto tersebut valid atau mencurigakan. Walaupun contoh ini terlihat sederhana, di dunia nyata sistem seperti ini melibatkan banyak aspek, mulai dari kualitas data, pemilihan model, hingga kepatuhan terhadap regulasi data.

1. **Sample Code / Pseudocode**

## 3.1 Pseudocode Validasi Foto Tanaman dengan AI:

Input : Foto Tanaman

Proses :

Lakukan pra-pemrosesan pada foto (resize, normalisasi)

 Ekstrak fitur visual dari foto menggunakan model computer vision

 Bandingkan fitur foto dengan model tanaman yang tersimpan

Tentukan jenis tanaman berdasarkan hasil perbandingan

Deteksi apakah foto merupakan foto asli atau hasil manipulas

Output :

Jenis tanaman

Status valid / tidak valid

**3.2 Contoh Kode Sederhana (Python – Klasifikasi Gambar)**

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

Memuat model klasifikasi tanaman

model = load\_model("model\_tanaman.h5")

Memuat dan menyiapkan gambar

img = image.load\_img("foto\_tanaman.jpg", target\_size=(224, 224))

img\_array = image.img\_to\_array(img)

img\_array = np.expand\_dims(img\_array, axis=0)

img\_array = img\_array / 255.0

Melakukan prediksi

prediksi = model.predict(img\_array)

kelas\_tanaman = np.argmax(prediksi)

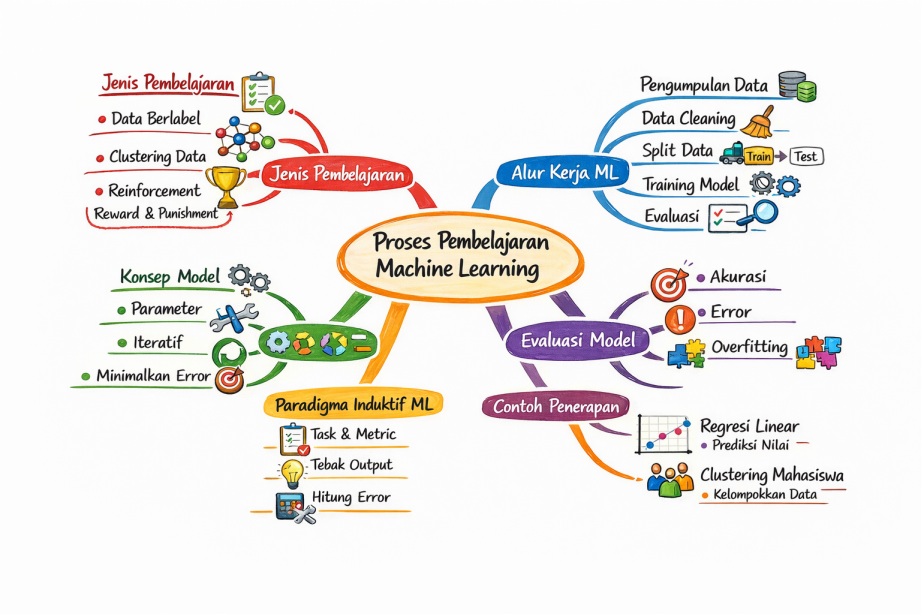
Menampilkan hasil

print("Jenis tanaman terdeteksi:", kelas\_tanaman)

**Rangkuman Pertemuan 3**

1. **Deskripsi Singkat Materi**

Pertemuan ketiga membahas konsep dasar proses pembelajaran dalam machine learning, mulai dari jenis-jenis pembelajaran seperti supervised dan unsupervised learning hingga alur kerja umum dalam membangun model machine learning. Materi ini bertujuan untuk memberikan pemahaman kepada mahasiswa mengenai bagaimana data diproses, digunakan untuk melatih model, serta dievaluasi agar dapat menghasilkan prediksi atau analisis yang baik.



1. **Penjelasan Detail Materi**

Pada pertemuan ketiga mata kuliah Machine Learning, pembahasan mulai masuk ke konsep yang lebih teknis, khususnya mengenai bagaimana proses pembelajaran mesin bekerja secara umum dan bagaimana data digunakan untuk membangun sebuah model. Pada tahap ini mahasiswa diajak untuk memahami bahwa machine learning bukan sekadar menjalankan kode atau library, tetapi merupakan sebuah proses sistematis yang dimulai dari data, dilanjutkan dengan pemodelan, hingga evaluasi hasil. Pemahaman alur ini menjadi sangat penting karena akan selalu digunakan pada pertemuan-pertemuan selanjutnya, baik saat membahas regresi, klasifikasi, maupun deep learning.

Materi pertemuan ketiga diawali dengan penjelasan mengenai jenis-jenis pembelajaran dalam machine learning, yaitu supervised learning, unsupervised learning, dan sekilas pengenalan reinforcement learning. Supervised learning dijelaskan sebagai metode pembelajaran mesin yang menggunakan data berlabel, artinya setiap data sudah memiliki jawaban atau target yang jelas. Contoh paling sederhana adalah data nilai ujian siswa yang dilengkapi dengan jumlah jam belajar. Dalam kasus ini, jam belajar menjadi input, sedangkan nilai ujian menjadi output atau label. Model dilatih menggunakan pasangan data tersebut agar nantinya mampu melakukan prediksi pada data baru yang belum pernah dilihat sebelumnya.

Berbeda dengan supervised learning, unsupervised learning menggunakan data yang tidak memiliki label. Pada metode ini, sistem tidak diberi tahu jawaban yang benar, melainkan diminta untuk menemukan pola atau struktur tersembunyi di dalam data. Contohnya adalah pengelompokan pelanggan berdasarkan perilaku belanja tanpa mengetahui sebelumnya kategori pelanggan tersebut. Dari data transaksi, sistem akan secara otomatis mengelompokkan pelanggan yang memiliki pola mirip. Melalui penjelasan ini, mahasiswa diharapkan dapat memahami perbedaan mendasar antara belajar dengan “guru” (supervised) dan belajar secara mandiri dari pola data (unsupervised).

Selanjutnya, dibahas juga mengenai alur kerja atau workflow dalam machine learning. Proses ini biasanya dimulai dari pengumpulan data, kemudian dilanjutkan dengan pembersihan data (data cleaning) dan pra-pemrosesan (preprocessing). Pada tahap ini, data yang tidak lengkap, duplikat, atau tidak relevan harus ditangani terlebih dahulu agar tidak mengganggu proses pembelajaran model. Dosen menekankan bahwa dalam praktik nyata, sebagian besar waktu pengembangan machine learning justru dihabiskan pada tahap pengolahan data, bukan pada pembuatan modelnya.

Setelah data siap, langkah berikutnya adalah membagi data menjadi data latih (training data) dan data uji (testing data). Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi seberapa baik model tersebut bekerja pada data baru. Pembagian ini penting untuk menghindari kondisi overfitting, yaitu ketika model terlalu “menghafal” data latih sehingga performanya buruk saat diuji pada data lain. Konsep ini menjadi dasar dalam menilai apakah sebuah model benar-benar pintar atau hanya sekadar mengingat data.

Pada pertemuan ini juga dijelaskan secara umum tentang model dalam machine learning. Model dapat dipahami sebagai representasi matematis dari pola yang terdapat dalam data. Ketika model dilatih, sistem akan menyesuaikan parameter-parameter tertentu agar kesalahan prediksi menjadi sekecil mungkin. Proses ini dilakukan secara iteratif, artinya model akan terus diperbaiki hingga mencapai performa yang dianggap optimal. Dari sini mahasiswa diperkenalkan pada konsep evaluasi model, seperti mengukur tingkat kesalahan atau akurasi, meskipun pembahasannya masih bersifat pengantar.

Sebagai contoh penerapan supervised learning, diberikan ilustrasi sederhana mengenai regresi linear. Misalnya, sebuah sistem ingin memprediksi nilai ujian mahasiswa berdasarkan jumlah jam belajar. Dengan menggunakan data historis, model regresi linear akan mencari hubungan antara jam belajar dan nilai ujian. Setelah model dilatih, sistem dapat digunakan untuk memprediksi nilai mahasiswa lain yang memiliki jumlah jam belajar tertentu. Contoh ini menunjukkan bagaimana data masa lalu dapat dimanfaatkan untuk membuat prediksi di masa depan.

Sementara itu, contoh unsupervised learning dapat dilihat pada kasus pengelompokan data mahasiswa berdasarkan IPK dan keaktifan organisasi. Tanpa adanya label seperti “mahasiswa aktif” atau “mahasiswa kurang aktif”, sistem akan mengelompokkan mahasiswa berdasarkan kemiripan karakteristiknya. Hasil pengelompokan ini kemudian dapat digunakan sebagai bahan analisis, misalnya untuk menentukan strategi pembinaan akademik atau kegiatan kampus.

1. **Sample Code / Pseudocode**

## 3.1 Pseudocode Paradigma Induktif Machine Learning:

Input: Dataset sebagai experience (data latih)

Tentukan task

Tentukan performance metric

Inisialisasi model h(x)

Ulangi:

Kasih data ke model biar “belajar”

Suruh model nebak hasil

Bandingin tebakan model sama jawaban asli

Hitung error (seberapa salah tebakan model)

Perbaiki parameter model biar ga ngulang kesalahan yang sama

**3.2 Contoh Kode Sederhana (Function Approximation – Regresi)**

import numpy as np

from sklearn.linear\_model import LinearRegression

Experience (data hasil pengamatan kehidupan nyata)

X = jam nongkrong di kampus

y = tingkat kelelahan

X = np.array([[1], [2], [3], [4], [5]])

y = np.array([10, 20, 30, 40, 50])

Model sebagai hipotesis h(x)

model = LinearRegression()

Model mulai belajar dari pengalaman hidup

model.fit(X, y)

Prediksi kelelahan kalau nongkrong 6 jam

prediksi = model.predict([[6]])

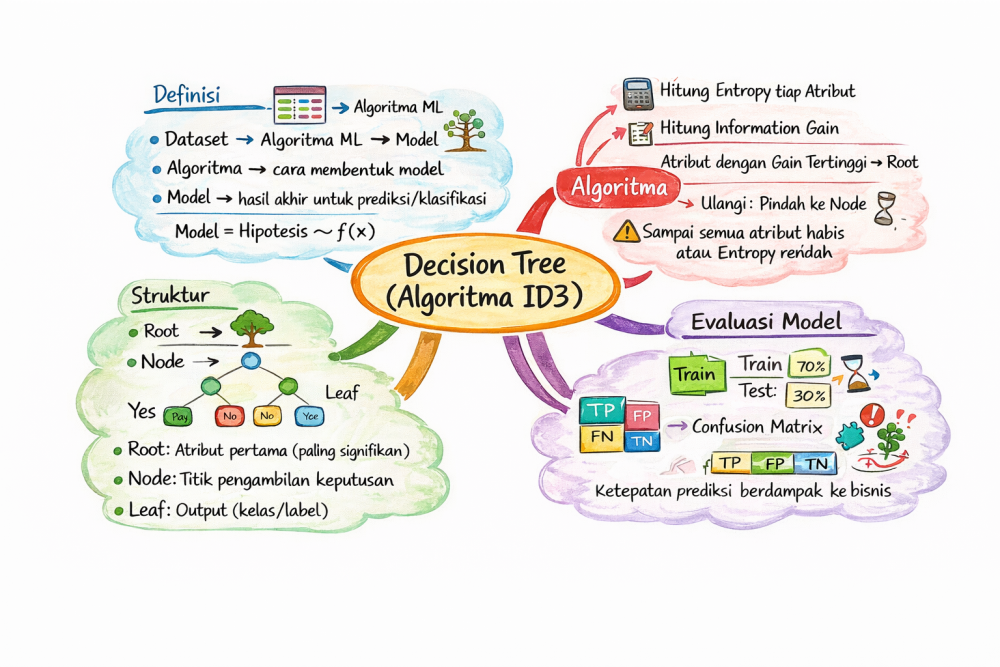
print("Prediksi tingkat kelelahan:", prediksi)

**Rangkuman Pertemuan 4**

Pada pertemuan ini dibahas kembali konsep dasar machine learning, khususnya perbedaan antara metode, algoritma, dan model. Machine learning dipandang sebagai sebuah proses yang menerima data (dataset) lalu menghasilkan model sebagai output. Model ini disebut juga sebagai hipotesis yang berfungsi untuk mendekati fungsi sebenarnya secara stokastik (berbasis probabilitas), bukan deterministik. Algoritma adalah cara membentuk model, sedangkan model adalah hasil akhirnya yang nantinya dapat digunakan untuk melakukan prediksi atau klasifikasi data baru.

Selanjutnya dibahas kembali jenis pendekatan dalam machine learning, yaitu supervised learning, semi-supervised learning, dan reinforcement learning. Fokus utama pertemuan ini adalah supervised learning, khususnya algoritma Decision Tree dengan algoritma populer seperti ID3 dan C4.5. Struktur model Decision Tree dijelaskan sebagai pohon yang memiliki root, node, dan leaf. Root merupakan atribut pertama yang paling signifikan, sedangkan leaf merupakan hasil keputusan (kelas atau label).

Dijelaskan pula bahwa dalam dataset terdapat fitur (atribut) dan label (kelas/target). Fitur adalah variabel input, sedangkan label adalah output yang ingin diprediksi, misalnya “Play Tennis = Yes/No”. Label bisa bersifat diskrit (kategori) atau kontinu (numerik). Dalam supervised learning, dataset harus memiliki label agar model dapat belajar dengan benar.



Proses pembentukan Decision Tree menggunakan algoritma ID3 dilakukan dengan menghitung entropy dan information gain untuk setiap atribut. Entropy digunakan untuk mengukur ketidakpastian data, sedangkan information gain digunakan untuk menentukan atribut mana yang paling berpengaruh dalam membedakan kelas. Atribut dengan gain tertinggi dipilih sebagai root, lalu proses ini diulang secara rekursif hingga terbentuk pohon keputusan.

Setelah model terbentuk, dilakukan proses evaluasi model menggunakan teknik train-test split, misalnya 70% data untuk training dan 30% untuk testing. Hasil prediksi dievaluasi menggunakan confusion matrix untuk melihat jumlah prediksi benar dan salah, serta mengukur akurasi model. Dijelaskan pula dampak kesalahan prediksi terhadap dunia bisnis, terutama pada sistem seperti fraud detection, di mana kesalahan klasifikasi dapat menyebabkan kerugian finansial dan hilangnya kepercayaan pelanggan.

* **Pseudocode Algoritma ID3**

ID3(Examples, Attributes, Target):

Jika semua Examples memiliki kelas yang sama:

return Leaf dengan label kelas tersebut

Jika Attributes kosong:

return Leaf dengan label kelas terbanyak

Hitung Entropy dari Examples

Untuk setiap atribut A dalam Attributes:

Hitung Information Gain(A)

Pilih atribut dengan Information Gain terbesar → A\_best

Jadikan A\_best sebagai root

Untuk setiap nilai v dari A\_best:

Buat cabang untuk nilai v

Ambil subset Examples\_v dengan A\_best = v

Jika Examples\_v kosong:

Tambahkan leaf dengan label kelas terbanyak

Else:

Subtree = ID3(Examples\_v, Attributes - A\_best, Target)

Tambahkan subtree ke cabang

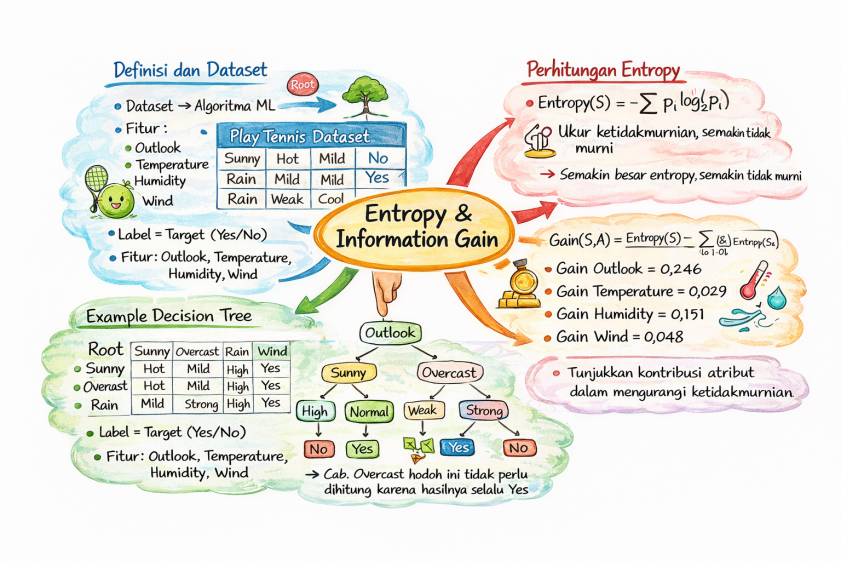
Return Tree

**Rangkuman Pertemuan 5**

Pada pertemuan kelima ini, materi yang dibahas adalah lanjutan dari pertemuan sebelumnya mengenai algoritma dalam machine learning, khususnya tentang pembentukan model dalam bentuk struktur data pohon (tree). Pertemuan ini menjadi pengganti pertemuan minggu sebelumnya yang belum terlaksana, sehingga durasi pembelajaran dilakukan lebih singkat namun tetap membahas inti materi yang penting. Fokus utama pembahasan adalah bagaimana algoritma ID3 bekerja dalam membangun sebuah decision tree berdasarkan data latih yang tersedia.

Di awal pertemuan, dijelaskan kembali konsep dasar dari struktur data tree. Tree merupakan struktur data berbentuk hierarki yang terdiri dari root (akar), node (simpul), dan leaf (daun). Root merupakan node paling atas, sedangkan leaf adalah node paling bawah yang tidak memiliki anak. Setiap node dapat memiliki satu atau lebih anak tergantung jenis tree yang digunakan. Dalam binary tree, setiap node hanya boleh memiliki maksimal dua anak, sedangkan dalam n-ary tree, setiap node bisa memiliki jumlah anak yang bebas.

Dalam implementasi di pemrograman, tree direpresentasikan menggunakan objek atau class. Setiap node menyimpan data, alamat parent (induk), serta daftar children (anak). Konsep alamat memori juga diperkenalkan untuk menggambarkan bagaimana komputer menyimpan setiap node di RAM dan bagaimana relasi antar node dibangun menggunakan referensi alamat tersebut. Dengan cara ini, struktur tree dapat dibentuk seperti rantai yang saling terhubung antar node dari root hingga leaf.



Selanjutnya dijelaskan bahwa dalam machine learning, model juga memiliki berbagai bentuk representasi, seperti model probabilistik, grafikal, neural network, matriks, dan tree. Pada pertemuan ini, fokus diarahkan pada model tree yang digunakan dalam algoritma decision tree. Tree ini dibentuk berdasarkan data latih dan digunakan untuk membantu pengambilan keputusan berdasarkan aturan-aturan yang dihasilkan dari data tersebut.

Dataset yang digunakan sebagai contoh adalah dataset klasik Play Tennis yang sering digunakan dalam pembelajaran machine learning. Dataset ini memiliki empat fitur utama yaitu Outlook, Temperature, Humidity, dan Wind, serta satu label yaitu Play Tennis dengan nilai Yes atau No. Dataset ini digunakan untuk menentukan apakah seseorang akan bermain tenis berdasarkan kondisi cuaca.

Dalam supervised learning, dataset harus memiliki label atau target. Label merupakan kelas atau kategori yang ingin diprediksi, sedangkan fitur adalah atribut yang digunakan sebagai dasar pengambilan keputusan. Pada dataset Play Tennis, labelnya adalah keputusan bermain tenis (Yes/No), sedangkan fitur-fitur lainnya menjadi dasar pertimbangan untuk membuat keputusan tersebut.

Untuk membangun decision tree menggunakan algoritma ID3, diperlukan perhitungan nilai entropy dan information gain. Entropy digunakan untuk mengukur tingkat ketidakmurnian suatu kumpulan data. Semakin besar nilai entropy, maka semakin tidak murni data tersebut. Sebaliknya, semakin kecil nilai entropy, maka data semakin homogen atau seragam.

Entropy dihitung menggunakan rumus:

Dimana pi merupakan proporsi jumlah data pada masing-masing kelas terhadap total data. Pada dataset Play Tennis terdapat dua kelas, yaitu Yes dan No. Dari total 14 data, terdapat 9 data Yes dan 5 data No. Dengan menggunakan rumus entropy, diperoleh nilai entropy total sebesar kurang lebih 0,94 yang menunjukkan bahwa data masih cukup tidak murni. Setelah menghitung entropy total, langkah selanjutnya adalah menghitung information gain untuk setiap atribut. Information gain menunjukkan seberapa besar kontribusi suatu atribut dalam mengurangi ketidakmurnian data. Semakin besar nilai gain, maka atribut tersebut semakin baik digunakan sebagai pemisah data.

Untuk menghitung gain, terlebih dahulu dihitung entropy dari setiap nilai atribut. Sebagai contoh, pada atribut Outlook terdapat tiga nilai yaitu Sunny, Overcast, dan Rain. Masing-masing nilai ini memiliki distribusi kelas Yes dan No yang berbeda, sehingga entropy untuk setiap nilai juga berbeda.

Hasil perhitungan menunjukkan bahwa:

* Gain Outlook = 0,246
* Gain Temperature = 0,029
* Gain Humidity = 0,151
* Gain Wind = 0,048

Dari hasil tersebut, atribut Outlook memiliki nilai gain terbesar, sehingga dipilih sebagai root dari decision tree. Setelah root ditentukan, proses dilanjutkan dengan membangun cabang berdasarkan nilai dari atribut Outlook, yaitu Sunny, Overcast, dan Rain. Untuk cabang Overcast, seluruh data menunjukkan hasil Yes, sehingga langsung ditetapkan sebagai leaf dengan keputusan Yes tanpa perlu perhitungan lanjutan. Untuk cabang Sunny dan Rain, data masih bercampur antara Yes dan No sehingga perlu dilakukan perhitungan ulang menggunakan atribut yang tersisa. Pada cabang Sunny, dilakukan perhitungan gain untuk atribut Temperature, Humidity, dan Wind. Hasil perhitungan menunjukkan bahwa Humidity memiliki gain tertinggi, sehingga dipilih sebagai node berikutnya. Jika Humidity bernilai High maka hasilnya No, sedangkan jika bernilai Normal maka hasilnya Yes. Pada cabang Rain, dilakukan perhitungan serupa dan diperoleh bahwa atribut Wind memiliki gain tertinggi. Jika Wind bernilai Weak maka hasilnya Yes, sedangkan jika bernilai Strong maka hasilnya No.

* **Pseudocode Algoritma ID3**

FUNCTION ID3(Data, Atribut):

IF semua data dalam satu kelas:

RETURN node kelas tersebut

IF atribut kosong:

RETURN node dengan kelas mayoritas

Hitung entropy(Data)

FOR setiap atribut A dalam Atribut:

Hitung entropy setiap nilai A

Hitung Information Gain(A)

Pilih atribut dengan gain terbesar → Root

FOR setiap nilai v pada Root:

Buat cabang

SubData = Data dengan nilai Root = v

Anak = ID3(SubData, Atribut - Root)

Hubungkan Anak ke Root

RETURN Root

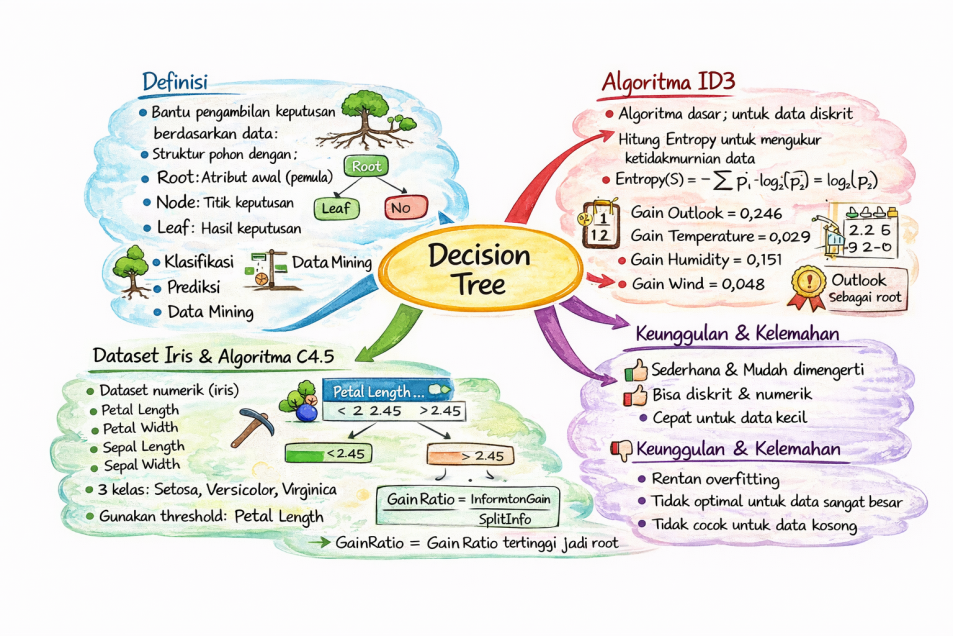
**Rangkuman Pertemuan 6**

Materi yang dibahas pada pertemuan ini adalah tentang Decision Tree (Pohon Keputusan), yang merupakan salah satu algoritma penting dalam bidang machine learning dan data mining. Decision Tree digunakan untuk membantu proses pengambilan keputusan berdasarkan data dengan cara membentuk struktur pohon, di mana setiap node mewakili atribut, setiap cabang mewakili kondisi, dan setiap daun mewakili hasil atau kelas.

Pada dasarnya, tujuan utama dari algoritma decision tree adalah menentukan atribut mana yang paling baik dijadikan sebagai root (akar pohon). Root ini akan menjadi pemisah pertama yang membagi data ke dalam beberapa kelompok berdasarkan nilai atribut tersebut. Setelah root ditentukan, proses ini akan dilanjutkan ke cabang-cabang berikutnya sampai seluruh data dapat diklasifikasikan dengan baik.

Algoritma decision tree memiliki banyak varian, namun yang pertama kali dipelajari adalah ID3 (Iterative Dichotomiser 3). ID3 merupakan algoritma decision tree paling dasar dan biasanya digunakan untuk data yang bersifat diskrit, yaitu data yang nilainya berupa kategori seperti “Sunny”, “Rainy”, “High”, “Normal”, dan bukan berupa angka.

Contoh kasus yang digunakan adalah dataset Play Tennis, yang memiliki target kelas berupa Yes dan No. Artinya, model akan memprediksi apakah seseorang akan bermain tenis atau tidak berdasarkan kondisi cuaca. Dataset ini memiliki 14 data, dengan 9 data berlabel Yes dan 5 data berlabel No.



Untuk menentukan root pada decision tree, digunakan konsep yang disebut dengan Entropy. Entropy digunakan untuk mengukur tingkat kemurnian atau ketidakmurnian suatu kumpulan data. Jika seluruh data dalam satu kelompok memiliki kelas yang sama, maka entropy-nya bernilai 0 yang berarti data tersebut murni. Sebaliknya, jika data terdiri dari campuran beberapa kelas, maka nilai entropy akan semakin besar.

Secara matematis, entropy dirumuskan sebagai:

Entropy(S) = − Σ pi log2(pi)

di mana pi adalah probabilitas masing-masing kelas. Dalam kasus Play Tennis, probabilitas Yes adalah 9/14 dan probabilitas No adalah 5/14. Setelah dimasukkan ke dalam rumus, diperoleh nilai entropy total sekitar 0,94. Nilai ini menunjukkan bahwa data masih cukup campuran sehingga perlu dilakukan pemisahan lebih lanjut.

Setelah entropy total diketahui, langkah berikutnya adalah menghitung Information Gain. Information Gain digunakan untuk mengukur seberapa besar pengaruh suatu atribut dalam membagi data menjadi kelompok yang lebih murni. Semakin besar nilai Information Gain suatu atribut, maka semakin baik atribut tersebut dijadikan sebagai root.

Rumus Information Gain adalah:

Gain(S, A) = Entropy(S) − Σ (|Sv| / |S|) × Entropy(Sv)

Artinya, entropy total dikurangi dengan jumlah entropy setiap subset yang dihasilkan oleh atribut tersebut, dikalikan dengan proporsi jumlah datanya.

Dalam dataset Play Tennis, dilakukan perhitungan Information Gain untuk setiap atribut, seperti Outlook, Temperature, Humidity, dan Wind. Dari hasil perhitungan tersebut, diperoleh bahwa atribut **Outlook** memiliki nilai Information Gain tertinggi. Oleh karena itu, Outlook dipilih sebagai root pertama pada pohon keputusan.

Outlook memiliki tiga nilai, yaitu Sunny, Overcast, dan Rain. Ketika data dibagi berdasarkan Outlook, ditemukan bahwa cabang Overcast menghasilkan data yang seluruhnya berlabel Yes, sehingga entropy-nya bernilai 0 dan dianggap sebagai cabang yang sudah murni.

Sementara itu, cabang Sunny dan Rain masih memiliki campuran Yes dan No, sehingga perlu dilakukan pemisahan lanjutan dengan menghitung entropy dan information gain kembali untuk masing-masing cabang tersebut.

Proses ini dilakukan secara berulang. Setiap kali data masih belum murni, maka dihitung kembali entropy dan information gain untuk atribut yang tersisa. Atribut dengan nilai gain tertinggi akan dijadikan node berikutnya. Proses akan berhenti jika seluruh data pada suatu cabang sudah murni atau jika tidak ada lagi atribut yang bisa digunakan. Dengan demikian, terbentuklah sebuah struktur pohon keputusan yang dapat digunakan untuk melakukan prediksi terhadap data baru.

Namun, algoritma ID3 memiliki keterbatasan karena hanya dapat digunakan untuk data yang bersifat diskrit. Untuk data yang bersifat numerik atau kontinu, seperti panjang kelopak bunga, lebar kelopak bunga, dan sebagainya, digunakan algoritma lain yang lebih canggih, yaitu C4.5.

Sebagai contoh, digunakan dataset Iris, yang merupakan dataset terkenal dalam dunia machine learning. Dataset ini memiliki empat fitur numerik, yaitu sepal length, sepal width, petal length, dan petal width, serta memiliki tiga kelas target, yaitu Setosa, Versicolor, dan Virginica.

Karena fitur pada dataset Iris berupa angka kontinu, maka tidak bisa langsung digunakan seperti pada ID3. Oleh karena itu, C4.5 menggunakan pendekatan berupa threshold atau nilai batas. Nilai numerik akan diubah menjadi dua kategori, misalnya:

Petal Length ≤ 2.45

Petal Length > 2.45

Nilai 2.45 ini diperoleh dari nilai tengah antara nilai minimum dan maksimum suatu fitur. Dengan cara ini, fitur numerik dapat diperlakukan seperti fitur diskrit.

Selain itu, C4.5 tidak menggunakan Information Gain secara langsung, tetapi menggunakan Gain Ratio. Gain Ratio adalah perbandingan antara Information Gain dengan Split Information. Tujuannya adalah untuk menghindari bias terhadap atribut yang memiliki banyak kategori.

Rumus Gain Ratio adalah:

GainRatio = InformationGain / SplitInformationss

Atribut dengan Gain Ratio tertinggi akan dipilih sebagai root.

Dengan metode ini, C4.5 mampu menangani dataset yang lebih kompleks dan lebih realistis, karena sebagian besar data di dunia nyata bersifat numerik.

Dalam praktiknya, perhitungan manual seperti ini cukup panjang dan rumit. Oleh karena itu, biasanya digunakan bantuan komputer dengan bahasa pemrograman Python menggunakan tools bernama **Jupyter** Notebook. Jupyter Notebook memungkinkan kita untuk menulis kode Python, menjalankan perhitungan, dan menampilkan hasil secara langsung dalam bentuk tabel maupun grafik.

Jupyter Notebook sangat sering digunakan dalam dunia data science dan AI untuk melakukan eksplorasi data, analisis, eksperimen model, serta validasi hasil sebelum model diimplementasikan ke dalam aplikasi sesungguhnya.

* **Pseudocode Algoritma Decision Tree (ID3)**

Algorithm ID3(Data, Attributes, Target):

Jika semua data dalam Data memiliki kelas Target yang sama:

Kembalikan node dengan label kelas tersebut

Jika Attributes kosong:

Kembalikan node dengan label kelas mayoritas dalam Data

Hitung entropy(Data)

Untuk setiap atribut A dalam Attributes:

Hitung InformationGain(Data, A)

Pilih atribut terbaik A\_best dengan InformationGain tertinggi

Jadikan A\_best sebagai node

Untuk setiap nilai v dalam A\_best:

Ambil subset Data\_v dari Data dengan nilai A\_best = v

Jika Data\_v kosong:

Tambahkan child node dengan label kelas mayoritas Data

Jika tidak:

Child = ID3(Data\_v, Attributes - A\_best, Target)

Tambahkan Child sebagai cabang dari node A\_best

Kembalikan node A\_best

**Rangkuman Materi Pertemuan 7**

Pada pertemuan ini dibahas kembali materi mengenai algoritma Decision Tree, khususnya algoritma ID3 dan pengembangannya yaitu C4.5. Decision Tree merupakan salah satu metode klasifikasi dalam machine learning yang bekerja dengan cara membentuk pohon keputusan berdasarkan atribut-atribut yang ada pada data. Tujuan utama dari algoritma Decision Tree adalah mencari atribut terbaik yang dijadikan sebagai root (akar pohon) dan selanjutnya membagi data menjadi cabang-cabang hingga diperoleh keputusan akhir berupa kelas atau label.

Algoritma Decision Tree memiliki banyak variasi, namun yang paling dasar dan sering digunakan sebagai pengenalan adalah ID3. Algoritma ID3 bekerja dengan data yang bersifat diskrit atau kategorikal, misalnya data dengan nilai seperti “Sunny”, “Rain”, “Overcast”, “High”, “Normal”, dan sebagainya. Pada contoh kasus yang sering digunakan, yaitu dataset Play Tennis, label kelasnya hanya terdiri dari dua nilai, yaitu “Yes” dan “No”.

Tujuan utama dari algoritma ID3 adalah memilih atribut yang paling berpengaruh dalam menentukan kelas target. Untuk mengetahui atribut mana yang paling berpengaruh, digunakan perhitungan yang disebut dengan entropy dan information gain. Entropy digunakan untuk mengukur tingkat ketidakmurnian atau ketidakpastian suatu kumpulan data. Jika seluruh data berada dalam satu kelas yang sama, maka data tersebut dikatakan murni dan nilai entropinya adalah nol. Sebaliknya, jika data terbagi ke dalam beberapa kelas, maka entropinya semakin besar.

Entropy dihitung menggunakan rumus Entropy((pi), di mana pi adalah probabilitas kemunculan suatu kelas dalam data. Sebagai contoh, jika dalam suatu dataset terdapat 14 data dengan 9 data berlabel “Yes” dan 5 data berlabel “No”, maka probabilitas “Yes” adalah 9/14 dan probabilitas “No” adalah 5/14. Dengan memasukkan nilai tersebut ke dalam rumus entropy, akan diperoleh nilai entropy total sekitar 0,940. Nilai ini menunjukkan bahwa data tersebut belum murni karena masih mengandung dua kelas yang berbeda.

Setelah menghitung entropy total, langkah berikutnya adalah menghitung Information Gain untuk setiap atribut. Information Gain digunakan untuk mengetahui seberapa besar pengaruh suatu atribut dalam membagi data menjadi lebih murni. Rumus Information Gain adalah Gain

(

di mana Sv adalah subset data berdasarkan nilai atribut tertentu, dan |Sv| adalah jumlah data pada subset tersebut.

Dalam kasus Play Tennis, terdapat beberapa atribut seperti Outlook, Humidity, Wind, dan Temperature. Masing-masing atribut memiliki beberapa nilai, misalnya Outlook memiliki nilai Sunny, Overcast, dan Rain. Untuk menghitung gain Outlook, data dibagi menjadi tiga subset berdasarkan nilai tersebut, kemudian dihitung entropy masing-masing subset, lalu dikalikan dengan proporsi datanya terhadap total data.

Hasil perhitungan menunjukkan bahwa atribut Outlook memiliki nilai Information Gain tertinggi dibandingkan atribut lainnya. Oleh karena itu, Outlook dipilih sebagai root atau akar dari decision tree. Setelah root ditentukan, pohon akan bercabang sesuai dengan nilai-nilai yang dimiliki oleh atribut tersebut, yaitu Sunny, Overcast, dan Rain.

Untuk cabang Overcast, seluruh data memiliki label “Yes”, sehingga node ini bersifat murni dan tidak perlu dilakukan perhitungan lanjutan. Artinya, jika Outlook bernilai Overcast, maka hasil prediksi langsung adalah Yes.

Untuk cabang Sunny, data yang tersisa berjumlah lima data dengan dua data berlabel Yes dan tiga data berlabel No. Karena masih terdapat dua kelas, maka perlu dilakukan perhitungan ulang untuk menentukan atribut terbaik berikutnya. Pada subset ini dihitung kembali entropy total khusus data Sunny, kemudian dihitung Information Gain untuk atribut-atribut yang tersisa seperti Humidity, Wind, dan Temperature.

Hasil perhitungan menunjukkan bahwa pada subset Sunny, atribut Humidity memiliki nilai gain tertinggi. Oleh karena itu, Humidity dijadikan node berikutnya. Humidity memiliki dua nilai, yaitu High dan Normal. Dari hasil pembagian, jika Humidity bernilai High maka seluruh datanya berlabel No, dan jika Humidity bernilai Normal maka seluruh datanya berlabel Yes. Karena masing-masing cabang sudah murni, maka proses pembentukan tree pada cabang Sunny selesai.

Untuk cabang Rain, data yang tersisa juga masih mengandung dua kelas sehingga perlu dilakukan perhitungan ulang. Setelah dihitung, atribut Wind memiliki nilai gain tertinggi. Wind memiliki dua nilai, yaitu Weak dan Strong. Jika Wind bernilai Weak maka seluruh datanya berlabel Yes, sedangkan jika Wind bernilai Strong maka seluruh datanya berlabel No. Karena kedua cabang sudah murni, maka proses pembentukan tree juga selesai.

Dengan demikian, terbentuklah pohon keputusan yang lengkap, di mana root-nya adalah Outlook, kemudian cabang Sunny dipecah oleh Humidity, cabang Rain dipecah oleh Wind, dan cabang Overcast langsung menghasilkan keputusan Yes. Inilah bentuk akhir dari Decision Tree menggunakan algoritma ID3.

Namun, algoritma ID3 memiliki keterbatasan karena hanya bisa menangani data diskrit. Jika data yang digunakan berupa data numerik atau kontinu seperti panjang, lebar, berat, atau ukuran lainnya, maka algoritma ID3 tidak dapat langsung digunakan. Untuk mengatasi hal tersebut, digunakan algoritma C4.5 yang merupakan pengembangan dari ID3.

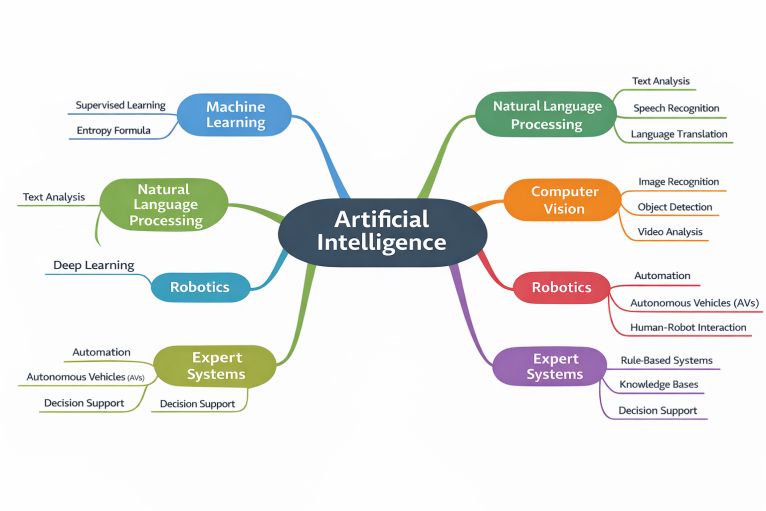
C4.5 tetap menggunakan konsep entropy dan information gain, tetapi menambahkan konsep baru yaitu Split Information dan Gain Ratio. Gain Ratio digunakan untuk menghindari bias terhadap atribut yang memiliki banyak nilai. Gain Ratio dihitung dengan membagi Information Gain dengan Split Information.

Split Information sendiri mengukur bagaimana suatu atribut membagi data tanpa memperhatikan kelasnya. Dengan demikian, Gain Ratio memberikan ukuran yang lebih adil dalam memilih atribut terbaik.

Pada dataset numerik seperti Iris Dataset, fitur-fitur berupa nilai kontinu seperti sepal length, sepal width, petal length, dan petal width. Dataset ini memiliki tiga kelas yaitu Setosa, Versicolor, dan Virginica. Karena fiturnya berupa angka, maka diperlukan proses pemisahan data menggunakan threshold atau nilai batas.

Threshold biasanya diperoleh dari nilai tengah antara nilai minimum dan maksimum suatu atribut, atau berdasarkan analisis statistik. Data kemudian dibagi menjadi dua kelompok, yaitu data dengan nilai lebih kecil atau sama dengan threshold, dan data dengan nilai lebih besar dari threshold. Dengan cara ini, data numerik diubah menjadi dua kategori biner yang dapat diproses oleh decision tree.

Setelah data dibagi berdasarkan threshold, proses perhitungan entropy, information gain, split information, dan gain ratio dilakukan seperti pada ID3, hanya saja sekarang dalam bentuk pembagian numerik. Atribut dengan gain ratio tertinggi dipilih sebagai root, lalu proses diulang untuk setiap cabang hingga data menjadi murni atau memenuhi kondisi berhenti.



Dalam praktik machine learning dan data science, perhitungan manual seperti ini jarang dilakukan secara penuh karena sudah tersedia library yang dapat melakukan semuanya secara otomatis. Namun memahami perhitungan manual sangat penting agar dapat memahami bagaimana algoritma bekerja di balik layar. Untuk melakukan eksperimen dan simulasi model, biasanya digunakan Jupyter Notebook. Jupyter Notebook merupakan sebuah tool interaktif yang memungkinkan pengguna menulis kode Python, menjalankan perhitungan, menampilkan grafik, dan melihat hasil secara langsung dalam satu dokumen. Notebook ini sangat populer di bidang data science dan artificial intelligence karena memudahkan proses eksplorasi data dan pengujian model.

Dalam penggunaan Python untuk data science, sangat disarankan menggunakan virtual environment. Virtual environment memungkinkan kita membuat lingkungan Python terpisah untuk setiap proyek sehingga library yang diinstal tidak bercampur dengan proyek lain. Hal ini membuat sistem lebih rapi, aman, dan mudah dikelola.

Secara umum, alur kerja dalam pengembangan AI biasanya dimulai dari eksplorasi data menggunakan Jupyter Notebook, kemudian dilakukan eksperimen model, lalu setelah model dianggap baik, barulah diimplementasikan ke dalam aplikasi menggunakan pendekatan software engineering.

* **Pseudocode Algoritma Decision Tree (ID3)**

Function Entropy(Data):

Untuk setiap kelas c dalam Target:

Hitung jumlah data Nc

Hitung total data N

Entropy = 0

Untuk setiap kelas c:

p =

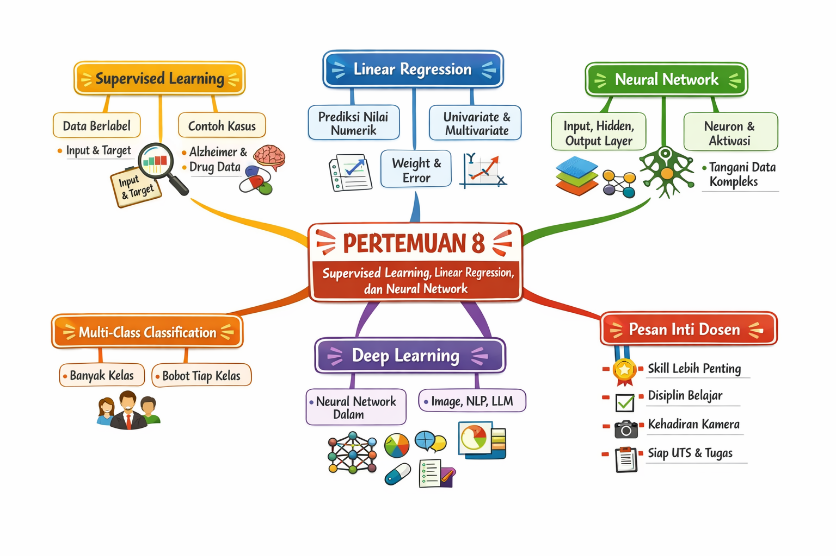
Entropy = Entropy -

Kembalikan nilai Entropy

**Rangkuman Materi Pertemuan 8**

1. **Deskripsi Singkat Materi**

Materi pada pertemuan kedelapan membahas konsep supervised learning, penerapan linear regression (univariate dan multivariate), serta pengantar neural network dan deep learning. Selain itu, pertemuan ini menekankan pentingnya pemahaman target pada dataset, disiplinnya kehadiran, keseriusan belajar, dan pemahaman konsep Machine Learning, khususnya supervised learning, linear regression (univariate & multivariate), serta pengantar neural network.



1. **Penjelasan Detail Materi**

Pada pertemuan kedelapan, dosen menekankan bahwa supervised learning merupakan pendekatan utama yang digunakan dalam mata kuliah ini. Supervised learning adalah metode pembelajaran mesin di mana data latih memiliki label atau target yang jelas. Target inilah yang menjadi acuan model dalam mempelajari pola dari data dan menghasilkan prediksi yang akurat.

Mahasiswa diingatkan untuk memastikan bahwa dataset yang digunakan benar-benar memiliki target. Penggunaan metode unsupervised learning, seperti clustering atau Gaussian Mixture Model (GMM), tidak dianjurkan apabila tujuan penelitian adalah prediksi dengan label yang jelas. Dalam proposal penelitian, mahasiswa diminta untuk fokus pada **s**atu metode utama, meskipun referensi jurnal dapat mencantumkan beberapa metode pembanding.

Materi juga membahas contoh dataset supervised learning, seperti dataset Alzheimer berbasis tulisan tangan dan dataset konsumsi obat-obatan (drug consumption). Pada dataset drug consumption, terdapat banyak kelas target yang merepresentasikan jenis dan frekuensi penggunaan obat. Untuk mempermudah proses pemodelan, dosen menyarankan penyederhanaan target menjadi dua kelas, yaitu pernah menggunakan dan tidak pernah menggunakan, dengan asumsi yang harus dijelaskan secara akademis dalam proposal.

Selanjutnya, dibahas konsep linear regression sebagai salah satu algoritma dasar dalam supervised learning. Linear regression bertujuan untuk memodelkan hubungan antara variabel input dan output dalam bentuk persamaan linear. Pada univariate linear regression, model hanya menggunakan satu variabel input, sedangkan pada multivariate linear regression, model menggunakan lebih dari satu variabel input. Konsep bias dan bobot (weight) diperkenalkan sebagai bagian dari persamaan regresi.

Materi kemudian berkembang ke arah multi**-**class classification dan keterbatasan model linear ketika jumlah kelas dan fitur semakin banyak. Dari permasalahan ini, diperkenalkan konsep neural network, yang terdiri dari input layer, hidden layer, dan output layer. Setiap neuron saling terhubung melalui bobot yang disesuaikan selama proses training untuk meminimalkan kesalahan prediksi.

Neural network menjadi dasar dari deep learning, yaitu model pembelajaran dengan banyak lapisan tersembunyi yang mampu menangkap pola kompleks dalam data. Dosen juga menyinggung perkembangan Large Language Model (LLM**)** dengan jumlah parameter yang sangat besar, serta kebutuhan perangkat keras yang memadai untuk menjalankannya.

Selain materi teknis, pertemuan ini juga menekankan aspek disiplin, kehadiran, dan kesiapan mahasiswa. Dosen mengingatkan bahwa keberhasilan di dunia kerja tidak hanya ditentukan oleh ijazah, tetapi juga oleh pemahaman konsep dan keterampilan praktis yang dimiliki mahasiswa dalam bidang machine learning dan data science.

1. **Pseudocode Konsep Supervised Learning**

Masukkan dataset dengan fitur dan target  
Tentukan metode pembelajaran (misalnya regresi linear)  
Bagi data menjadi data latih dan data uji

Buat model awal

Lakukan proses training menggunakan data latih  
Evaluasi performa model menggunakan data uji

Jika performa belum optimal: Sesuaikan parameter atau model

Ulangi hingga diperoleh performa terbaik

1. **Sample code di python**

import numpy as np

from sklearn.linear\_model import LinearRegression

Data sebagai experience

X = fitur (misalnya jam belajar, jumlah latihan)

y = target (nilai ujian)

Membuat model regresi linear

model = LinearRegression()

Melatih model

model.fit(X, y)

Prediksi

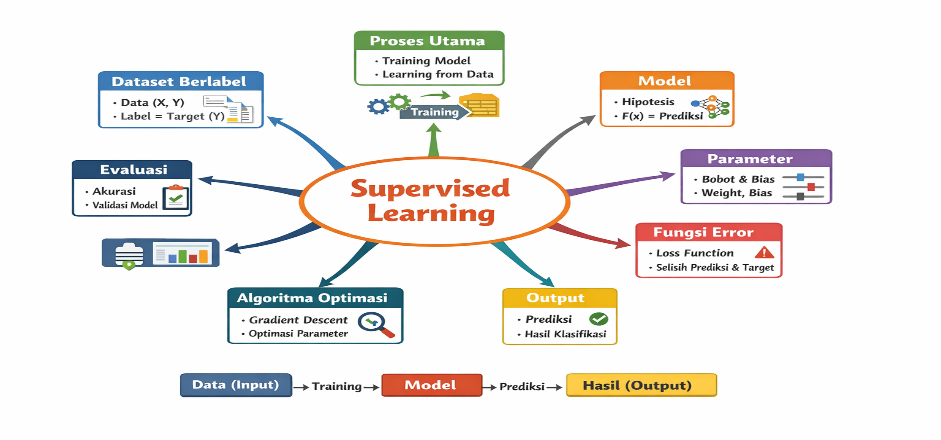
prediksi = model.predict([[5, 2]])

print("Hasil prediksi:", prediksi)

**Rangkuman Materi Pertemuan 9**

1. **Deskripsi Singkat Materi**

Materi pada pertemuan ini membahas keterkaitan antara regresi linear dengan konsep neural network. Pembahasan dimulai dari regresi linear sederhana (univariate), berkembang ke regresi linear multivariate, hingga diperkenalkan representasi model tersebut dalam bentuk neuron dan jaringan saraf tiruan (neural network). Selain itu, materi juga memberikan gambaran awal mengenai hidden layer, bobot (weight), bias, error, serta kompleksitas komputasi dalam neural network.



1. **Penjelasan Detail Materi**

Regresi linear merupakan salah satu model dasar dalam machine learning yang digunakan untuk memodelkan hubungan antara variabel input dan output. Secara matematis, regresi linear sederhana dapat dituliskan dalam bentuk persamaan garis lurus, yaitu f(x) = mx + c, di mana m adalah kemiringan (slope) dan c adalah konstanta atau bias.

Dalam notasi machine learning, persamaan tersebut dapat dituliskan ulang menjadi , di mana w₁ merepresentasikan bobot (weight) dan w₀ merepresentasikan bias. Bias sering kali dituliskan sebagai w₀x₀ dengan nilai x₀ = 1. Dengan demikian, bias dapat diperlakukan sama seperti variabel input lainnya, tetapi dengan nilai tetap.

Pada contoh kasus harga rumah, luas rumah dijadikan sebagai variabel input (x₁), sedangkan harga rumah dijadikan sebagai target atau output (y). Jika kenaikan luas rumah berbanding lurus dengan kenaikan harga, maka model regresi linear dapat digunakan untuk melakukan prediksi harga berdasarkan luas rumah.

Regresi linear sederhana dengan satu variabel input disebut sebagai univariate. Jika jumlah variabel input lebih dari satu, misalnya luas rumah, keberadaan carport, taman, atau fasilitas lainnya, maka model tersebut disebut sebagai regresi linear multivariate. Pada regresi multivariate, fungsi prediksi dapat dituliskan sebagai:

dengan x₀ = 1 sebagai bias, dan x₁ hingga xₙ sebagai variabel input.

Model regresi linear ini dapat direpresentasikan secara visual dalam bentuk neuron. Setiap neuron menerima beberapa input, masing-masing dikalikan dengan bobot tertentu, kemudian dijumlahkan untuk menghasilkan sebuah output. Jika model hanya terdiri dari input layer dan output layer, maka struktur tersebut setara dengan regresi linear.

Selanjutnya, diperkenalkan konsep neural network dengan menambahkan hidden layer di antara input layer dan output layer. Hidden layer berfungsi untuk meningkatkan kemampuan model dalam menangkap pola yang lebih kompleks pada data. Jumlah hidden layer dan jumlah neuron pada setiap layer tidak memiliki aturan baku dan sering disebut sebagai "black magic" karena biasanya ditentukan melalui eksperimen.

Setiap koneksi antar neuron memiliki bobot yang harus dipelajari oleh model. Semakin banyak neuron dan layer yang digunakan, maka semakin banyak pula bobot yang harus dihitung. Hal ini menyebabkan kebutuhan komputasi menjadi lebih besar. Oleh karena itu, proses training neural network umumnya memanfaatkan GPU yang mampu melakukan komputasi matriks secara paralel.

Dalam proses pembelajaran, model akan menghasilkan prediksi (ŷ) yang kemudian dibandingkan dengan nilai target sebenarnya (y). Selisih antara nilai prediksi dan nilai sebenarnya disebut sebagai error atau delta. Untuk menghindari nilai negatif, error sering kali dikuadratkan sehingga menghasilkan Mean Squared Error (MSE).

Untuk mencari nilai bobot terbaik, digunakan algoritma optimasi seperti Stochastic Gradient Descent (SGD). Algoritma ini bekerja dengan cara memperbarui bobot secara bertahap berdasarkan error yang dihasilkan, hingga diperoleh model dengan error yang minimal.

Materi ini menegaskan bahwa neural network pada dasarnya merupakan pengembangan dari regresi linear. Perbedaannya terletak pada jumlah layer dan kompleksitas perhitungan yang digunakan. Dengan memahami regresi linear dan representasinya sebagai neuron, mahasiswa diharapkan memiliki fondasi yang kuat sebelum mempelajari neural network dan deep learning secara lebih mendalam.

1. **Pseudocode Konsep Supervised Learning**

Mulai

Siapkan dataset yang memiliki pasangan data input dan label target

Pisahkan dataset menjadi data latih dan data uji

Tentukan model yang akan digunakan

Tentukan parameter awal model (bobot dan bias)

Tentukan fungsi error untuk mengukur kesalahan prediksi Selama proses pelatihan berlangsung:

Hitung hasil prediksi berdasarkan data input

Bandingkan hasil prediksi dengan nilai target

Hitung besar kesalahan (error)

Perbarui bobot dan bias untuk memperkecil error

Ulangi proses pelatihan hingga error minimum atau jumlah iterasi tercapai

Uji model menggunakan data uji

Evaluasi performa model

Selesai

1. **Sample code di python**

Import library yang dibutuhkan

import numpy as np

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

* Menyiapkan dataset (Supervised Learning ada input dan target)

x = fitur (misalnya luas rumah)

y = target (harga rumah)

* Membagi data menjadi data latih dan data uji
* Membuat model Linear Regression

model = LinearRegression()

* Melatih model menggunakan data latih

model.fit(X\_train, y\_train)

* Melakukan prediksi

y\_pred = model.predict(X\_test)

* Evaluasi model

mse = mean\_squared\_error(y\_test, y\_pred)

* Menampilkan hasil

print("Bobot (weight):", model.coef\_)

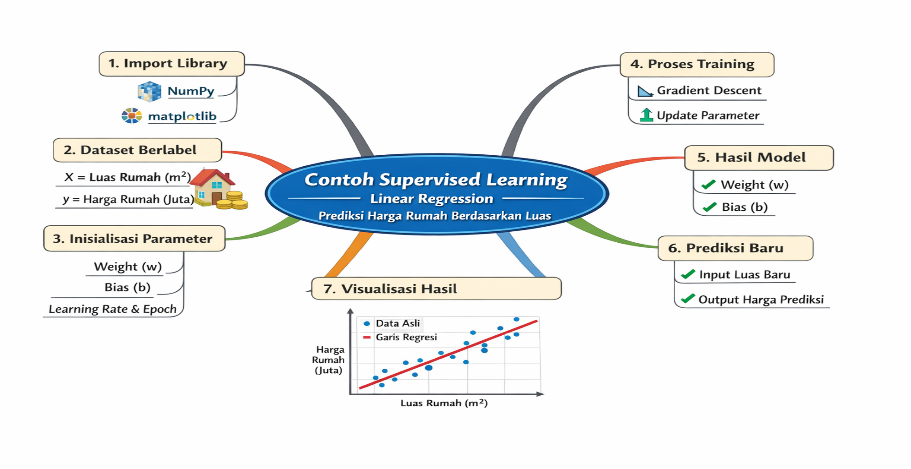
print("Bias:", model.intercept\_)

print("Mean Squared Error:", mse)

**Rangkuman Materi Pertemuan 10**

1. **Deskripsi Singkat Materi**

Pada pertemuan ini dibahas konsep lanjutan dalam machine learning yang meliputi supervised learning, regresi, neural network (perceptron dan multilayer perceptron), serta tahapan penting sebelum proses pelatihan model yaitu data preprocessing. Materi juga menekankan pentingnya pemahaman fundamental, proses training model, serta penerapan machine learning di dunia industri, khususnya dalam kasus nyata seperti fraud detection dan pengolahan data tidak terstruktur.



1. **Penjelasan Detail Materi**

Pada awal pertemuan, dibahas kembali konsep supervised learning, yaitu metode pembelajaran mesin yang menggunakan dataset berlabel. Beberapa algoritma yang termasuk dalam supervised learning antara lain linear regression, decision tree, dan neural network. Linear regression digunakan untuk memodelkan hubungan antara variabel input dan output secara linear, baik dalam bentuk univariate (satu variabel input) maupun multivariate (lebih dari satu variabel input).

Dalam regresi univariate, model hanya memiliki satu variabel input, sedangkan pada regresi multivariate, jumlah variabel input bisa lebih dari dua dan dinyatakan sebagai . Pada model ini, x0x\_0x0​ selalu bernilai 1 dan berfungsi sebagai bias. Output model diperoleh dari hasil penjumlahan perkalian antara setiap input dan bobotnya, yang kemudian dapat dimasukkan ke dalam fungsi aktivasi seperti sigmoid. Fungsi sigmoid menghasilkan nilai output dalam rentang 0 hingga 1 sehingga sering digunakan pada kasus klasifikasi.

Materi kemudian dilanjutkan dengan pembahasan neural network, khususnya feedforward neural network dan multilayer perceptron (MLP). Neural network terdiri dari tiga komponen utama, yaitu input layer, hidden layer, dan output layer. Pada multilayer perceptron, terdapat lebih dari satu hidden layer yang memungkinkan model mempelajari pola data yang lebih kompleks. Setiap neuron pada suatu layer terhubung dengan neuron pada layer berikutnya melalui bobot (weights), dan proses perhitungan dilakukan menggunakan operasi penjumlahan berbobot yang diikuti oleh fungsi aktivasi.

Proses pembelajaran pada neural network dilakukan melalui training, salah satunya menggunakan metode stochastic gradient descent, yang berfungsi untuk memperbarui bobot agar nilai error semakin kecil dan model dapat mencapai kondisi konvergen. Hasil dari proses training ini adalah sebuah model yang direpresentasikan sebagai fungsi f(x)f(x)f(x), yang selanjutnya dapat dideploy ke dalam sebuah aplikasi untuk melakukan prediksi terhadap data baru.

Materi selanjutnya membahas alur kerja machine learning secara umum, dimulai dari dataset yang dimasukkan ke dalam proses training hingga menghasilkan model, kemudian model tersebut diimplementasikan (deployment) ke dalam sistem atau aplikasi nyata. Dalam praktiknya, data yang dihadapi di dunia nyata sering kali tidak ideal karena bersifat campuran, terdiri dari data numerik, kategorikal, teks, serta sering mengandung nilai yang hilang atau tidak konsisten.

Oleh karena itu, sebelum melakukan training model, diperlukan tahapan data preprocessing. Preprocessing mencakup beberapa proses penting, antara lain data cleaning, data transformation, dan feature engineering. Data cleaning dilakukan untuk menangani missing value dan outlier. Missing value dapat ditangani dengan beberapa cara, seperti mengganti dengan nilai rata-rata, nilai minimum atau maksimum, atau menggunakan model prediktif. Outlier adalah data yang memiliki nilai ekstrem dan dapat mempengaruhi kinerja model jika tidak ditangani dengan baik.

Data transformation mencakup proses normalisasi, seperti min-max scaling, yang bertujuan menyamakan rentang nilai data menjadi antara 0 dan 1. Normalisasi sangat penting karena perbedaan skala data yang terlalu jauh dapat memperlambat proses training dan menyebabkan model sulit mencapai konvergensi. Selain itu, dilakukan pula data encoding, yaitu mengubah data kategorikal menjadi data numerik, misalnya dengan label encoding atau one-hot encoding.

Selain preprocessing, dibahas pula konsep feature engineering, yaitu proses menciptakan fitur baru dari data yang sudah ada untuk meningkatkan performa model. Contohnya adalah menghitung jarak antara alamat pengiriman dan alamat penagihan dalam kasus fraud detection, atau mengekstrak umur dari tanggal lahir.

Materi juga menyinggung permasalahan imbalanced dataset, yaitu kondisi ketika jumlah data pada satu kelas jauh lebih banyak dibandingkan kelas lainnya. Untuk mengatasi hal ini, dapat digunakan teknik upsampling, downsampling, atau kombinasi keduanya agar distribusi data menjadi lebih seimbang.

Pada bagian akhir, dosen memperkenalkan konsep deep learning, yaitu pengembangan neural network dengan arsitektur yang jauh lebih besar dan kompleks. Deep learning memiliki banyak hidden layer dan mampu mengekstrak fitur secara otomatis dari data, terutama pada data tidak terstruktur seperti gambar dan teks. Beberapa algoritma deep learning yang disebutkan antara lain Convolutional Neural Network (CNN) untuk pengolahan citra, Long Short-Term Memory (LSTM) untuk data time series, serta Generative Adversarial Network (GAN) untuk menghasilkan data baru.

Secara keseluruhan, materi ini menekankan bahwa pemahaman konsep dasar, preprocessing data, dan pemilihan algoritma yang tepat sangat berpengaruh terhadap keberhasilan model machine learning. Proses pembelajaran tidak berhenti pada pembuatan model, tetapi juga mencakup evaluasi, deployment, dan pengembangan berkelanjutan sesuai dengan kebutuhan dunia nyata.

1. **Pseudocode Konsep Supervised Learning**

Siapkan dataset berlabel yang terdiri dari data masukan (X) dan target (Y)

Lakukan pembagian data menjadi data latih dan data uji

Tentukan model supervised learning yang akan digunakan

Inisialisasi parameter model (bobot dan bias)

Tentukan fungsi untuk mengukur kesalahan prediksi

Lakukan proses pelatihan model:

Masukkan data latih ke dalam model

Hitung nilai prediksi berdasarkan parameter saat ini

Bandingkan hasil prediksi dengan nilai target

Hitung nilai kesalahan (error)

Perbarui parameter model untuk meminimalkan kesalahan

Ulangi proses pelatihan hingga kriteria berhenti terpenuhi  
(jumlah iterasi tercapai atau kesalahan minimum)

Uji model menggunakan data uji

Hitung dan analisis performa model

Selesai

1. **Sample code di python**

import numpy as np

import matplotlib.pyplot as plt

X = luas rumah (m2)

y = harga rumah (juta)

Prediksi

y\_pred = w \* X + b

Hitung error

error = y\_pred - y

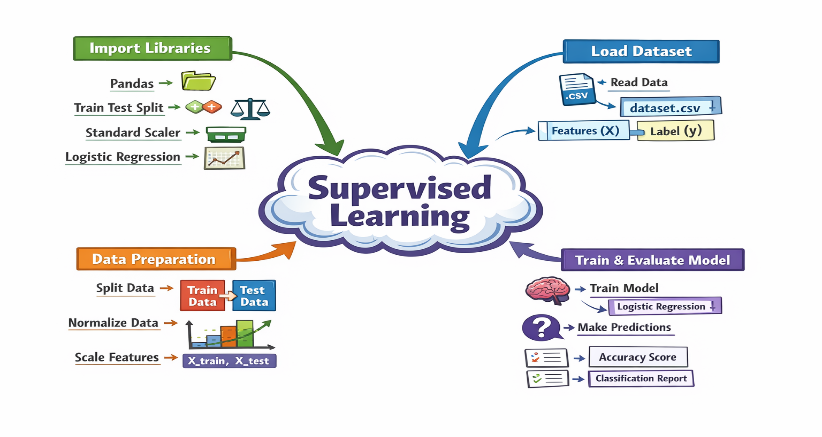
Update parameter

dw =

**Rangkuman Materi Pertemuan 11**

1. **Deskripsi Singkat Materi**

Materi ini merupakan rangkuman dari seluruh pembahasan yang disampaikan selama perkuliahan, mencakup konsep dasar machine learning, supervised learning, unsupervised learning, hingga penerapan deep learning dalam tugas besar UTS dan UAS. Rangkuman ini disusun dengan format naratif akademik sebagai landasan konseptual sebelum memasuki pembahasan teknis dan implementatif.



1. **Penjelasan Detail Materi**

Machine learning merupakan bagian dari kecerdasan buatan (Artificial Intelligence) yang mempelajari bagaimana komputer dapat belajar dari data tanpa harus diprogram secara eksplisit untuk setiap permasalahan. Dalam machine learning, data berperan sebagai pengalaman (experience) yang digunakan sistem untuk membangun model guna melakukan prediksi, klasifikasi, maupun pengambilan keputusan.

Dalam konteks pembelajaran, machine learning dibagi menjadi beberapa pendekatan utama, yaitu supervised learning, unsupervised learning, dan deep learning. Masing-masing pendekatan memiliki karakteristik, tujuan, serta metode yang berbeda, namun saling berkaitan dalam proses analisis data.

**2.1 Supervised Learning**

Supervised learning adalah metode machine learning yang menggunakan data berlabel, yaitu data yang memiliki target atau kelas. Model dilatih dengan pasangan data input (X) dan output (y) sehingga sistem dapat mempelajari hubungan antara keduanya. Pendekatan ini umum digunakan untuk permasalahan klasifikasi dan regresi.

Pada tugas besar UTS, mahasiswa diwajibkan menerapkan supervised learning sebagai dasar pemahaman machine learning. Proyek UTS mencakup penyusunan proposal, pemilihan dataset, studi literatur, implementasi model menggunakan Python atau Jupyter Notebook, serta penyusunan laporan akhir dalam format jurnal komputer. Seluruh proses dan hasil pekerjaan dikumpulkan dalam satu repository GitHub.

**2.2 Unsupervised Learning**

Unsupervised learning merupakan metode machine learning yang tidak menggunakan data berlabel. Pada pendekatan ini, sistem hanya diberikan data fitur tanpa informasi kelas atau target. Tujuan utama unsupervised learning adalah menemukan pola tersembunyi, struktur alami, atau hubungan tertentu di dalam data.

Salah satu fungsi utama unsupervised learning adalah clustering, yaitu proses pengelompokan data berdasarkan tingkat kemiripan karakteristik. Kelompok yang terbentuk tidak memiliki makna kelas secara langsung dan perlu dianalisis lebih lanjut melalui proses profiling.

Selain clustering, unsupervised learning juga berperan dalam transforming data dan dimensionality reduction. Reduksi dimensi bertujuan untuk mengurangi jumlah fitur agar data lebih mudah dianalisis dan divisualisasikan tanpa menghilangkan informasi penting. Algoritma yang umum digunakan antara lain Principal Component Analysis (PCA) dan autoencoder.

Algoritma unsupervised learning yang sering digunakan antara lain K-Means dan Gaussian Mixture Model (GMM). K-Means bekerja berdasarkan jarak ke pusat cluster (centroid), sedangkan GMM menggunakan pendekatan probabilistik dengan asumsi data berasal dari beberapa distribusi Gaussian. Pemahaman unsupervised learning sangat penting sebagai tahap eksplorasi data dan preprocessing sebelum menerapkan supervised learning maupun deep learning.

**2.3 Deep Learning**

Deep learning merupakan pengembangan lebih lanjut dari machine learning yang menggunakan jaringan saraf tiruan (Artificial Neural Network) dengan banyak lapisan (hidden layers). Pendekatan ini mampu mempelajari pola yang kompleks dan non-linear, terutama pada data berskala besar.

Pada tugas besar UAS, mahasiswa diwajibkan mengembangkan proyek UTS dengan menambahkan pendekatan deep learning. Proposal UAS merupakan pembaruan dari proposal UTS dengan penekanan pada alasan pemilihan metode deep learning, hipotesis penelitian, serta perbandingan dengan metode supervised learning sebelumnya.

Mahasiswa juga diwajibkan mengimplementasikan model deep learning, menyusun laporan akhir dalam format jurnal komputer, serta mengunggah seluruh berkas ke repository GitHub yang sama dengan UTS.

**2.4 Keterkaitan Materi dengan Tugas Besar**

Seluruh materi yang dibahas, mulai dari konsep dasar machine learning, supervised learning, unsupervised learning, hingga deep learning, saling terintegrasi dalam pelaksanaan tugas besar UTS dan UAS. Mahasiswa diharapkan mampu memilih metode yang sesuai dengan karakteristik dataset, menjelaskan proses analisis secara ilmiah, serta membandingkan performa metode supervised learning dan deep learning secara objektif.

Materi ini menjadi fondasi penting sebelum memasuki pembahasan teknis lanjutan, sehingga mahasiswa memiliki pemahaman konseptual yang kuat mengenai peran dan penerapan machine learning di berbagai bidang.

1. **Pseudocode Konsep Supervised Learning**

Mulai

Input dataset yang memiliki label (data berlabel)

Pisahkan dataset menjadi:

* Data fitur (X)
* Data target atau label (Y)

Lakukan preprocessing data:

* Menghapus atau mengisi data yang hilang
* Normalisasi atau standarisasi data jika diperlukan

Bagi dataset menjadi dua bagian:

* Data latih (training data)
* Data uji (testing data)

Pilih algoritma supervised learning

(contoh: Regresi Linear, K-Nearest Neighbor, Decision Tree)

Latih model menggunakan data latih

Gunakan model untuk memprediksi data uji

Bandingkan hasil prediksi dengan nilai target sebenarnya

Hitung performa model menggunakan metrik evaluasi

(akurasi, presisi, recall, atau error)

Jika hasil belum optimal:

* Lakukan penyesuaian parameter
* Ulangi proses pelatihan dan pengujian

Simpan model terbaik

Gunakan model untuk memprediksi data baru

Selesai

1. Sample code di python

Contoh Supervised Learning menggunakan Python

1. Import library yang dibutuhkan

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report

2. Load dataset

Contoh dataset sederhana (misalnya data kesehatan)

data = pd.read\_csv("dataset.csv")

3. Pisahkan fitur dan label

X = data.drop("target", axis=1) # fitur

y = data["target"]

4. Bagi data menjadi data latih dan data uji

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.2, random\_state=42

)

5. Normalisasi data

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

6. Inisialisasi model supervised learning

model = LogisticRegression()

7.Latih model

model.fit(X\_train, y\_train)

8. Lakukan prediksi

y\_pred = model.predict(X\_test)

9. Evaluasi model

accuracy = accuracy\_score(y\_test, y\_pred)

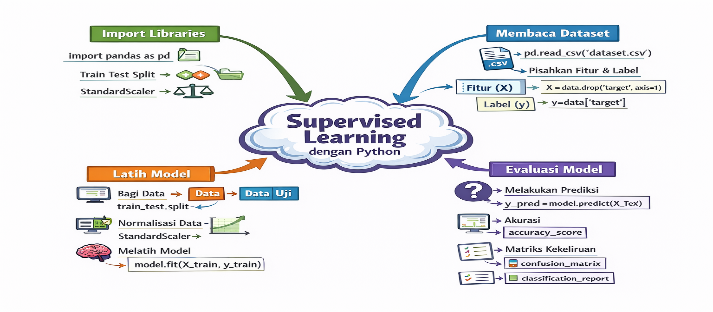
print("Akurasi Model:", accuracy)

print(classification\_report(y\_test, y\_pred))

**Rangkuman Materi Pertemuan 12**

1. **Deskripsi Singkat Materi**

Bab ini merupakan rangkuman dari seluruh ketikan dan penjelasan lisan yang disampaikan dalam sesi perkuliahan. Materi difokuskan pada praktik dan pemahaman konseptual machine learning, khususnya supervised learning dan unsupervised learning, disertai pembahasan teknis seperti preprocessing data, encoding, evaluasi model, clustering, serta Gaussian Mixture Model. Rangkuman disusun dalam bentuk naratif akademik sebagaimana format pada dokumen "Bab 1 Pengenalan Machine Learning".



1. **Penjelasan Detail Materi**

Pada perkuliahan ini dibahas proses pengerjaan proyek machine learning mahasiswa, mulai dari tahap eksplorasi data, preprocessing, pemodelan, hingga evaluasi hasil. Salah satu tahap penting yang ditekankan adalah preprocessing data, termasuk proses encoding data kategorikal menjadi numerik. Encoding dilakukan dengan mengubah nilai kategorikal menjadi representasi angka, umumnya 0 dan 1, agar data dapat diproses oleh algoritma machine learning. Selain itu, digunakan pula proses normalisasi atau standarisasi data dengan Standard Scaler untuk menyamakan skala fitur.

Model yang digunakan mahasiswa bervariasi, antara lain Logistic Regression, Decision Tree, Random Forest, dan Support Vector Machine (SVM). Perbedaan model ini dijelaskan secara konseptual, termasuk alasan pemilihan model dan pengaruhnya terhadap akurasi. Evaluasi model dilakukan menggunakan metrik seperti akurasi dan confusion matrix. Confusion matrix dijelaskan sebagai alat untuk melihat kesalahan prediksi model, misalnya jumlah data yang salah diklasifikasikan pada kelas tertentu.

Mahasiswa juga diperkenalkan dengan hyperparameter tuning sebagai bagian dari optimasi model. Hyperparameter tuning bertujuan untuk meningkatkan performa model dengan menyesuaikan parameter tertentu, seperti jumlah hidden layer, jumlah neuron, atau parameter lain yang tidak dipelajari langsung oleh model. Proses ini dilakukan melalui beberapa percobaan hingga diperoleh hasil akurasi terbaik.

Selain supervised learning, perkuliahan juga membahas unsupervised learning, khususnya clustering. Clustering merupakan proses pengelompokan data tanpa label dengan tujuan menemukan struktur atau pola alami dalam data. Dua algoritma utama yang dibahas adalah K-Means dan Gaussian Mixture Model (GMM). K-Means menggunakan konsep jarak antar data untuk membentuk cluster berdasarkan kedekatan terhadap centroid. Nilai K pada K-Means menunjukkan jumlah cluster yang ingin dibentuk, bukan jumlah data terdekat seperti pada K-Nearest Neighbor (KNN).

Proses K-Means dimulai dengan pemilihan centroid secara acak, kemudian menghitung jarak setiap data ke centroid menggunakan fungsi jarak, seperti Euclidean Distance atau Manhattan Distance. Data akan dikelompokkan ke cluster dengan jarak terdekat, dan centroid akan diperbarui hingga posisinya stabil. Model K-Means menyimpan informasi berupa posisi centroid sebagai representasi cluster.

Gaussian Mixture Model (GMM) diperkenalkan sebagai pendekatan clustering berbasis probabilistik. GMM mengasumsikan bahwa data berasal dari beberapa distribusi Gaussian (normal). Setiap distribusi direpresentasikan oleh parameter mean (μ) dan standar deviasi (σ). Proses pembelajaran pada GMM menggunakan metode Expectation Maximization (EM), yang terdiri dari dua tahap utama, yaitu tahap expectation untuk menghitung probabilitas keanggotaan data terhadap setiap cluster, dan tahap maximization untuk memperbarui parameter distribusi agar likelihood data semakin besar.

Distribusi data menjadi konsep penting dalam unsupervised learning. Distribusi Gaussian atau distribusi normal digambarkan sebagai bentuk gunung yang menunjukkan frekuensi kemunculan nilai data. Melalui distribusi ini, dapat diperkirakan jumlah cluster atau kelompok yang terbentuk secara alami. Visualisasi distribusi data membantu manusia memahami pola data, meskipun komputer sendiri bekerja berdasarkan parameter statistik, bukan visual.

Dalam kasus data berdimensi tinggi, visualisasi menjadi sulit dilakukan. Oleh karena itu, digunakan teknik dimensionality reduction untuk mengurangi jumlah fitur tanpa menghilangkan informasi penting. Metode yang dibahas antara lain Principal Component Analysis (PCA) dan Autoencoder. Teknik ini sering digunakan sebelum proses clustering atau sebagai tahap eksplorasi data.

Perkuliahan juga menekankan pentingnya pemahaman konsep, bukan sekadar menyalin kode atau menggunakan bantuan AI. Mahasiswa diharapkan mampu menjelaskan alasan pemilihan metode, fungsi preprocessing, serta interpretasi hasil model. Pemahaman ini penting sebagai bekal dalam dunia kerja, khususnya bagi mahasiswa yang ingin berkarier sebagai Data Scientist, Machine Learning Engineer, atau AI Engineer.

Materi ini menjadi penguatan konsep sebelum memasuki tahap lanjutan, yaitu penerapan deep learning pada tugas UAS. Dengan memahami supervised learning, unsupervised learning, serta dasar statistik dan optimasi model, mahasiswa diharapkan memiliki fondasi yang kuat dalam pengembangan sistem berbasis machine learning.

1. **Pseudocode Konsep Supervised Learning**

Mulai

Input dataset berlabel

Pisahkan dataset menjadi:

Data latih (training data)

Data uji (testing data)

Lakukan preprocessing data:

Tangani data kosong

Lakukan normalisasi atau standarisasi

Lakukan encoding jika terdapat data kategorikal

Inisialisasi model supervised learning

Latih model menggunakan data latih

Gunakan model untuk melakukan prediksi pada data uji

Evaluasi hasil prediksi menggunakan metrik evaluasi

Akurasi

Presisi

Recall

F1-Score

Jika performa model belum baik:

Lakukan penyesuaian parameter

Latih ulang model

Tampilkan hasil evaluasi model

Selesi

1. **Sample code di python**

1. Import library yang dibutuhkan

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

2. Membaca dataset

Dataset harus memiliki kolom target (label)

data = pd.read\_csv("dataset.csv")

3. Memisahkan fitur dan label

X = data.drop("target", axis=1)

y = data["target"]

4. Membagi data menjadi data latih dan data uji

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.2, random\_state=42

)

5. Normalisasi data

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

6. Inisialisasi model supervised learning

model = LogisticRegression()

7. Melatih model

model.fit(X\_train, y\_train)

8. Melakukan prediksi

y\_pred = model.predict(X\_test)

9. Evaluasi model

print("Akurasi Model :", accuracy\_score(y\_test, y\_pred))

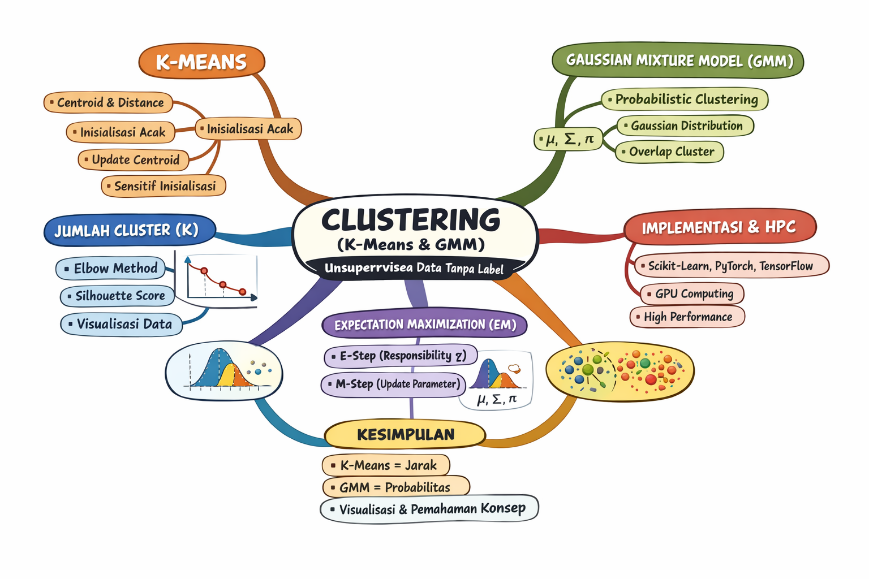
print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(\_

**Rangkuman Materi Pertemuan 13**

1. **Deskripsi Singkat Materi**

Materi pada pertemuan ini membahas konsep clustering sebagai salah satu metode unsupervised learning dalam machine learning. Pembahasan difokuskan pada dua algoritma utama, yaitu K-Means dan Gaussian Mixture Model (GMM). Selain itu, materi juga menjelaskan konsep distribusi normal, probabilitas, serta algoritma Expectation Maximization (EM) yang digunakan dalam GMM, termasuk proses iterasi, konvergensi, dan visualisasi hasil clustering.



1. **Penjelasan Detail Materi**

Clustering merupakan metode pembelajaran mesin tanpa label (unsupervised learning) yang bertujuan untuk mengelompokkan data berdasarkan kemiripan karakteristik. Data yang memiliki sifat mirip akan dikelompokkan dalam satu cluster, sedangkan data yang berbeda akan berada pada cluster yang berbeda.

K-Means Clustering

Algoritma K-Means bekerja dengan konsep centroid, yaitu titik pusat dari suatu cluster. Proses K-Means diawali dengan pemilihan centroid secara acak. Selanjutnya, setiap data dihitung jaraknya terhadap seluruh centroid, dan data akan dimasukkan ke cluster dengan jarak terdekat. Setelah itu, centroid diperbarui dengan menghitung rata-rata seluruh data dalam cluster tersebut. Proses ini dilakukan secara berulang hingga posisi centroid tidak lagi berubah secara signifikan atau telah mencapai kondisi stabil.

K-Means menghasilkan pembagian data yang tegas, di mana setiap data hanya menjadi anggota satu cluster berdasarkan jarak terdekat ke centroid.

Gaussian Mixture Model (GMM)

Berbeda dengan K-Means, Gaussian Mixture Model (GMM) menggunakan pendekatan probabilistik dengan asumsi bahwa data mengikuti distribusi normal (Gaussian). Distribusi normal ditandai dengan data yang berkumpul di sekitar nilai rata-rata dan membentuk kurva menyerupai gunung.

Apabila dalam satu dataset terdapat lebih dari satu puncak distribusi, maka setiap puncak tersebut dianggap sebagai satu Gaussian. Gabungan dari beberapa Gaussian inilah yang disebut sebagai Gaussian Mixture. Oleh karena itu, GMM mampu menangani data yang saling tumpang tindih dan memberikan peluang keanggotaan suatu data terhadap lebih dari satu cluster.

Setiap distribusi Gaussian memiliki parameter utama berupa:

μ (mean) sebagai nilai rata-rata atau puncak distribusi,

σ (standar deviasi) sebagai ukuran sebaran data,

Σ (covariance) untuk data multivariat.

Probabilitas dan Konsep Dasar GMM

Dalam GMM, yang dicari bukan hanya jarak, tetapi probabilitas suatu data berasal dari suatu cluster. Misalnya, untuk sebuah data dengan nilai tertentu, GMM akan menghitung probabilitas data tersebut berasal dari masing-masing distribusi Gaussian yang ada.

Pendekatan ini mirip dengan konsep probabilitas bersyarat, di mana sistem menghitung peluang suatu kejadian berdasarkan kondisi tertentu. Dengan cara ini, GMM mampu memberikan representasi clustering yang lebih fleksibel dibandingkan K-Means.

* Expectation Maximization (EM) Algorithm
* Proses pembelajaran pada GMM menggunakan algoritma Expectation Maximization (EM), yang terdiri dari dua tahap utama:
* Expectation Step (E-Step)  
  Pada tahap ini, dihitung responsibility (γ / gamma), yaitu probabilitas setiap data terhadap masing-masing cluster. Nilai ini menunjukkan seberapa besar kontribusi suatu data terhadap sebuah Gaussian.
* Maximization Step (M-Step)  
  Pada tahap ini, parameter model diperbarui berdasarkan nilai responsibility yang telah dihitung, yaitu:

nilai mean (μ),

covariance (Σ),

dan mixing coefficient (π) yang menunjukkan proporsi data pada setiap cluster.

Proses E-Step dan M-Step dilakukan secara berulang hingga nilai log-likelihood stabil atau perubahan nilainya sangat kecil, yang menandakan bahwa model telah konvergen.

Iterasi dan Konvergensi

Pada awal proses training, nilai mean dan covariance diinisialisasi secara acak sehingga sering kali belum sesuai dengan distribusi data. Seiring berjalannya iterasi, parameter-parameter tersebut akan menyesuaikan diri dengan pola data yang sebenarnya. Iterasi akan berhenti ketika model sudah stabil atau telah mencapai batas maksimum iterasi yang ditentukan.

Penentuan Jumlah Cluster

Jumlah cluster (K) tidak selalu diketahui di awal. Beberapa metode yang dapat digunakan untuk menentukan jumlah cluster yang optimal antara lain:

* visualisasi distribusi data,
* Elbow Method,
* Silhouette Score.

Untuk data berdimensi tinggi, digunakan teknik reduksi dimensi seperti Principal Component Analysis (PCA) atau Autoencoder agar data dapat divisualisasikan dalam dua atau tiga dimensi.

Framework dan Komputasi

Dalam implementasi machine learning, digunakan berbagai framework seperti Scikit-learn, PyTorch, dan TensorFlow. Framework modern memanfaatkan konsep tensor dan komputasi paralel menggunakan GPU untuk mempercepat perhitungan, terutama pada data berukuran besar dan model yang kompleks.

1. **Pseudocode Konsep Supervised Learning**

Masukkan dataset sebagai data pelatihan

Tentukan jumlah cluster (K)

Inisialisasi parameter μ, Σ, dan π secara acak

Lakukan perulangan:

Hitung probabilitas (E-Step)

Perbarui parameter model (M-Step)

Hitung log-likelihood

Hentikan iterasi jika model telah konvergen

1. **Sample code di python**

from sklearn.mixture import GaussianMixture

import numpy as np

Data contoh

X = np.array([[20], [25], [30], [60], [65], [70]])

Membuat model GMM dengan 2 cluster

gmm = GaussianMixture(n\_components=2)

Melatih model

gmm.fit(X)

Prediksi cluster

labels = gmm.predict(X)

print("Hasil clustering:", lab