

Rangkuman Materi Perkuliahan Machine Learning Kelompok 17

Shanaya Balghis Riyona¹, Thoriqurrahman Akrami²,

^{1,2} Program Studi Teknik Informatika, STMIK Tazkia

E-mail :

241552010012.shanaya@student.stmik.tazkia.ac.id¹

241552010014.thoriqurrahman@student.stmik.tazkia.ac.id²

Abstrak

Dokumen ini merupakan rangkuman komprehensif materi perkuliahan Machine Learning yang disampaikan melalui serangkaian video pertemuan. Materi mencakup pengenalan profesi di bidang data dan AI, definisi formal machine learning menurut Tom Mitchell, filosofi data dan pengetahuan, hingga perkembangan teknologi seperti deep learning dan generative AI. Pembahasan berlanjut pada konsep dasar dan lanjutan, seperti perbedaan pendekatan deduktif dan induktif, supervised dan unsupervised learning, serta pemodelan matematika melalui function approximation. Selain itu, dokumen ini mengulas algoritma-algoritma penting seperti Decision Tree (ID3), Gaussian Mixture Models (GMM), dan algoritma Expectation-Maximization (EM), lengkap dengan penjelasan konseptual, matematis, dan implementasi teknis. Tidak hanya aspek teknis, rangkuman juga memuat wawasan tentang etika, regulasi, kepatuhan (compliance), strategi tugas besar, penggunaan GitHub sebagai portofolio, serta pentingnya komunitas dan networking dalam karier IT. Secara keseluruhan, dokumen ini bertujuan menjadi bahan belajar terstruktur yang menghubungkan teori, praktik, dan realitas industri dalam bidang machine learning dan kecerdasan buatan.

Kata kunci : *Machine Learning, Artificial Intelligence, Deep Learning, Data dan Pengetahuan, Supervised Learning, Unsupervised Learning, Decision Tree (ID3), Gaussian Mixture Model (GMM), Expectation-Maximization (EM), Data Science, Compliance dan Etika Data, GitHub dan Portofolio, Pipeline Machine Learning, Generative AI.*

Summary of Machine Learning Lecture Material Group 17

Abstract

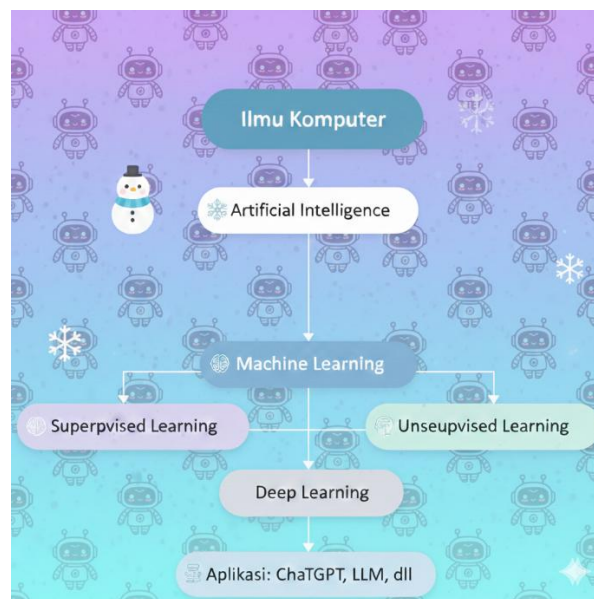
This document is a comprehensive summary of Machine Learning course materials delivered through a series of lecture videos. The content covers an introduction to careers in data and AI, the formal definition of machine learning according to Tom Mitchell, the philosophy of data and knowledge, and the development of technologies such as deep learning and generative AI. The discussion continues with fundamental and advanced concepts, including the difference between deductive and inductive approaches, supervised and unsupervised learning, and mathematical modeling through function approximation. In addition, this document reviews important algorithms such as Decision Tree (ID3), Gaussian Mixture Models (GMM), and the Expectation-Maximization (EM) algorithm, complete with conceptual, mathematical, and technical implementation explanations. Beyond technical aspects, the summary also includes insights into ethics, regulation, compliance, final project strategies, the use of GitHub as a portfolio, and the importance of community and networking in an IT career. Overall, this document aims to serve as a structured learning resource that connects theory, practice, and real-world industry applications in the field of machine learning and artificial intelligence.

Keywords : *Machine Learning, Artificial Intelligence, Deep Learning, Data and Knowledge, Supervised Learning, Unsupervised Learning, Decision Tree (ID3), Gaussian Mixture Model (GMM), Expectation-Maximization (EM), Data Science, Data Ethics and Compliance, GitHub and Portfolio, Machine Learning Pipeline, Generative AI.*

BAB 1 Pengantar Machine Learning & Prospek Karir

1. Deskripsi Singkat Materi

Video pertama merupakan sesi pembuka perkuliahan yang membahas pengenalan machine learning, ruang lingkup profesi di bidang data dan kecerdasan buatan, serta gambaran tren teknologi di masa depan. Materi ini menjadi dasar pemahaman sebelum memasuki pembahasan teknis pada pertemuan-pertemuan selanjutnya.



Gambar 1 Taksonomi Pengantar Machine Learning

2. Penjelasan Detail Materi

2.1 Profesi di Bidang Data dan AI

Pada bagian awal, dijelaskan peta karier yang relevan dengan mata kuliah machine learning. Beberapa profesi utama yang diperkenalkan meliputi data scientist yang berfokus pada eksperimen dan penemuan insight dari data, AI atau machine learning engineer yang bertugas mengimplementasikan model ke dalam sistem atau aplikasi, data analyst yang menganalisis data untuk kebutuhan bisnis, serta data engineer yang menangani infrastruktur dan alur data, meskipun profesi terakhir tidak menjadi fokus utama dalam mata kuliah ini.

Profesi-profesi tersebut dinilai masih sangat dibutuhkan oleh industri dan relatif aman dari ancaman otomatisasi, karena sistem AI masih memerlukan peran manusia dalam perancangan, pengawasan, dan pengambilan keputusan. Oleh karena itu, tujuan perkuliahan ini tidak hanya agar mahasiswa memahami teori, tetapi juga mampu mengimplementasikan konsep machine learning ke dalam bentuk perangkat lunak.

2.2 Definisi Machine Learning menurut Tom Mitchell

Machine learning didefinisikan berdasarkan teori Tom Mitchell, yaitu suatu komputer dikatakan belajar apabila ia belajar dari pengalaman (experience), untuk menyelesaikan suatu tugas (task), dengan ukuran kinerja tertentu (performance), dan kinerjanya meningkat seiring bertambahnya pengalaman. Konsep ini dijelaskan melalui analogi ujian aljabar, di mana mengerjakan soal merupakan task, nilai ujian merupakan performance, dan proses latihan sebelumnya merupakan experience. Semakin banyak latihan yang dilakukan, seharusnya nilai yang diperoleh semakin baik. Dengan demikian, inti dari machine learning

adalah proses pembelajaran yang diukur secara kuantitatif dan menunjukkan peningkatan kinerja dari waktu ke waktu.

2.3 Filosofi Data dan Pengetahuan

Materi selanjutnya membahas hubungan antara data dan pengetahuan dari sudut pandang filosofis. Data dipandang sebagai fakta atau kebenaran yang menjadi dasar pembentukan pengetahuan. Model, hipotesis, atau teori yang dibangun harus selalu konsisten dengan data.

Sebagai contoh, jika terdapat hipotesis bahwa semua angsa berwarna putih, lalu ditemukan satu angsa berwarna hitam, maka hipotesis tersebut harus direvisi atau ditolak. Data tidak dapat disalahkan, melainkan model atau pengetahuan yang tidak sesuai dengan fakta. Prinsip ini menegaskan bahwa dalam machine learning, kesalahan prediksi menunjukkan bahwa model perlu diperbaiki agar lebih sesuai dengan data.

2.4 Prediksi Tren Teknologi

Pandangan mengenai bidang teknologi yang diprediksi akan berkembang pesat dalam lima tahun ke depan. Bidang pertama adalah keamanan AI (AI security), karena sistem AI dapat dimanipulasi atau diserang sehingga diperlukan ahli yang mampu mengamankannya. Kedua adalah Web3 dan blockchain yang menawarkan konsep desentralisasi dan keamanan data. Ketiga adalah software engineering, karena kemampuan membangun perangkat lunak yang andal tetap menjadi fondasi utama dalam pengembangan teknologi, termasuk sistem berbasis AI.

2.5 Referensi dan Sumber Belajar

Sebagai rujukan utama, mahasiswa diwajibkan menggunakan buku *Machine Learning* karya Tom Mitchell. Selain itu, buku *Artificial Intelligence: A Modern Approach* karya Stuart Russell direkomendasikan sebagai referensi tambahan karena menjadi standar di banyak universitas ternama. Mahasiswa juga didorong untuk tidak takut membaca literatur berbahasa Inggris, karena sebagian besar sumber terbaik mengenai machine learning dan AI tersedia dalam bahasa tersebut.

3. Sample Code / Pseudocode

3.1 Pseudocode Konsep Machine Learning

Input: Data (Experience)
Tentukan tugas (Task)
Tentukan ukuran kinerja (Performance)

Inisialisasi model

Ulangi:

 Latih model dengan data

 Ukur performa model

 Jika performa belum baik:

 Perbaiki model

Sampai performa meningkat

3.2 Contoh Kode Sederhana (Python – Regresi Linear)

```
from sklearn.linear_model import LinearRegression
import numpy as np
```

```
# Data sebagai experience
```

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

```
y = np.array([2, 4, 6, 8])
```

```
# Membuat model
```

```
model = LinearRegression()
```

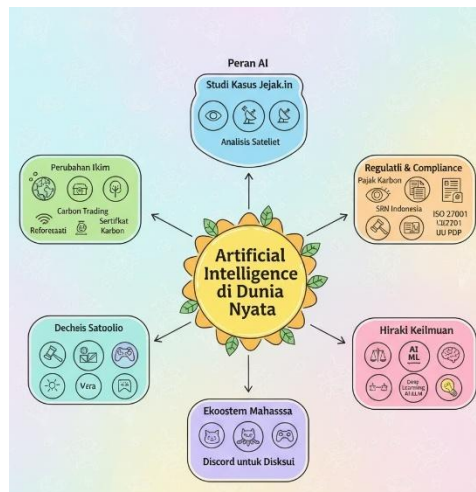
```
model.fit(X, y)
```

```
# Mengukur performa dengan prediksi
prediksi = model.predict([[5]])
print(prediksi)
```

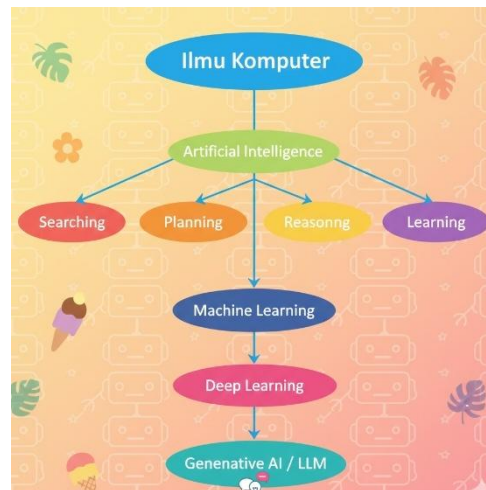
BAB 2 Implementasi AI di Dunia Nyata dan Peta Besar Artificial Intelligence

1. Deskripsi Singkat Materi

Video kedua membahas bagaimana Artificial Intelligence (AI) dan Machine Learning (ML) digunakan dalam dunia nyata melalui studi kasus lingkungan, khususnya pada isu perubahan iklim dan ekonomi karbon. Selain itu, video ini juga menjelaskan peta besar keilmuan AI, mulai dari Artificial Intelligence, Machine Learning, Deep Learning, hingga Generative AI. Mahasiswa diarahkan untuk tidak hanya memahami aspek teknis, tetapi juga memahami domain masalah, regulasi, dan ekosistem kerja profesional.



Gambar 2 Mind Map Implementasi AI



Gambar 3 Taksonomi Implementasi AI

2. Penjelasan Detail Materi

2.1 Studi Kasus Jejak.in dan Ekonomi Karbon

Menjelaskan bahwa pekerjaan di bidang IT tidak hanya berfokus pada kemampuan koding, tetapi juga harus memahami domain masalah dan regulasi. Salah satu contoh nyata adalah studi kasus Jejak.in yang bergerak di bidang ekonomi karbon.

Masalah lingkungan seperti perubahan iklim menyebabkan kenaikan permukaan laut dan mengancam wilayah pesisir, termasuk Jakarta. Hal ini memicu munculnya regulasi global terkait emisi karbon. Dalam sistem carbon trading, perusahaan penghasil emisi harus membeli kredit karbon untuk menyeimbangkan emisi yang mereka hasilkan. Yang dibeli bukanlah tanah atau hutan secara langsung, melainkan sertifikat yang menyatakan bahwa sejumlah karbon telah diserap oleh alam, biasanya melalui reforestasi.

Platform seperti Jejak.in berperan memfasilitasi proses ini, mulai dari pencatatan reforestasi hingga administrasi sertifikat karbon. Namun, masalah terbesar dalam bisnis ini adalah kepercayaan (trust). Muncul pertanyaan: apakah pohon benar-benar ditanam? Apakah lokasinya sesuai klaim?

2.2 Peran AI dalam Validasi

AI digunakan untuk menjawab masalah kepercayaan tersebut. Dengan computer vision, sistem dapat memverifikasi foto tanaman, misalnya membedakan bibit pohon duren dan mangga, atau mendeteksi foto yang telah dimanipulasi. Selain itu, AI juga digunakan untuk menganalisis citra satelit guna memastikan tutupan lahan (land cover). Sistem dapat mengecek apakah lahan yang diklaim memang lahan baru yang direforestasi, bukan hutan lindung yang sudah ada atau perkebunan sawit yang tidak menyerap karbon seefektif hutan alami.

2.3 Regulasi dan Kepatuhan

Dalam konteks global, barang yang masuk ke Eropa akan diperiksa jejak karbonnya. Jika emisinya tinggi, pajak karbonnya juga tinggi. Di Indonesia terdapat Sistem Registri Nasional (SRN) untuk mencatat aktivitas karbon, sementara secara global ada standar seperti Verra. Mahasiswa juga ditekankan untuk memahami standar keamanan seperti ISO 27001 dan aturan perlindungan data seperti UU PDP, karena pengelolaan data menjadi aspek yang sangat sensitif.

2.4 Administrasi dan Ekosistem Mahasiswa

Menekankan pentingnya GitHub bukan hanya sebagai tempat menyimpan kode, tetapi sebagai portofolio karier. Tugas besar harus ditata rapi karena bisa dilihat oleh perusahaan saat mahasiswa melamar kerja. Untuk komunikasi teknis, digunakan Discord agar diskusi lebih terstruktur dibandingkan menggunakan WhatsApp.

2.5 Hirarki Keilmuan Artificial Intelligence

Untuk menghindari kebingungan istilah, di jelaskan peta besar AI. Artificial Intelligence merupakan bidang paling luas yang mencakup searching, planning, reasoning, dan learning. Machine Learning adalah bagian dari AI yang fokus pada learning. Deep Learning merupakan bagian dari ML yang menggunakan jaringan saraf berlapis. Generative AI dan Large Language Model (LLM) adalah pengembangan terbaru dari deep learning yang mampu menghasilkan konten baru.

Mahasiswa diminta merujuk pada buku “Artificial Intelligence: A Modern Approach” karya Stuart Russell dan “Machine Learning” karya Tom Mitchell sebagai referensi utama agar pemahaman yang diperoleh bersifat akademis dan valid.

3. Sample Code / Pseudocode

3.1 Pseudocode Validasi Foto Tanaman dengan AI:

Input: Foto tanaman

Proses:

- Ekstrak fitur visual dari foto
- Bandingkan dengan model tanaman yang tersimpan
- Tentukan jenis tanaman
- Deteksi apakah foto asli atau hasil manipulasi

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

model = load_model("model_tanaman.h5")

img = image.load_img("bibit.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
```

```
prediksi = model.predict(img_array)
kelas = np.argmax(prediksi)
```

```
print("Hasil klasifikasi:", kelas)
```

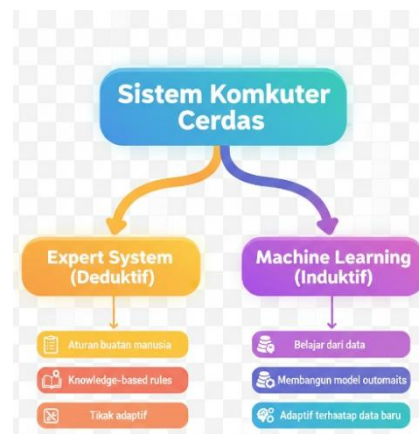
BAB 3 Definisi Filosofis Machine Learning, Paradigma, dan Konsep Data

1. Deskripsi Singkat Materi

Video ketiga membahas dasar filosofis dan konseptual dari machine learning, khususnya perbedaan paradigma antara sistem tradisional berbasis aturan (deduktif) dan machine learning berbasis data (induktif). Selain itu, video ini mengupas secara mendalam definisi formal machine learning menurut Tom Mitchell, hubungan antara data, fakta, dan pengetahuan, serta pandangan matematis machine learning sebagai proses pendekatan fungsi (function approximation). Materi ini menjadi fondasi teoritis penting untuk memahami cara kerja model pembelajaran mesin modern.



Gambar 5 Mind Map Filosofis Machine Learning



Gambar 4 Taksonomi Filosofis Machine Learning

2. Penjelasan Detail Materi

2.1 Pergeseran Paradigma: Deduktif vs Induktif

Menjelaskan bahwa sistem cerdas modern sangat berbeda dengan perangkat lunak tradisional. Pada masa lalu, banyak sistem menggunakan pendekatan deduktif melalui expert system atau sistem pakar. Dalam pendekatan ini, seorang pakar menuliskan aturan secara eksplisit ke dalam komputer. Pengetahuan umum diturunkan menjadi keputusan untuk kasus khusus.

Sebagai contoh di bidang medis, aturan dapat berbunyi: jika suhu tubuh lebih dari 38°C dan terdapat bintik merah, maka pasien mengalami demam berdarah. Pendekatan ini memiliki kelemahan besar, yaitu tidak mampu menangani gejala baru yang tidak tercantum dalam aturan. Jika kondisi tidak sesuai dengan rule yang tersedia, sistem akan gagal total.

Sebaliknya, machine learning menggunakan pendekatan induktif. Komputer tidak diberi aturan secara langsung, melainkan diberikan ribuan hingga jutaan data contoh. Dari data tersebut, sistem secara otomatis mencari pola dan membentuk model atau hipotesis umum. Pendekatan ini jauh lebih fleksibel dan mampu menangani kompleksitas dunia nyata yang sulit dirumuskan manusia secara eksplisit.

2.2 Definisi Machine Learning Menurut Tom Mitchell

Definisi formal machine learning yang digunakan dalam perkuliahan adalah:

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .

Dari definisi ini terdapat tiga komponen utama:

1. **Task (T):** tugas yang harus dikerjakan sistem, seperti klasifikasi email spam, pengenalan wajah, atau bermain catur.
2. **Performance (P):** ukuran keberhasilan, misalnya akurasi, error rate, atau jumlah kemenangan.
3. **Experience (E):** data atau pengalaman yang digunakan untuk belajar, seperti dataset atau riwayat permainan.

Sebuah sistem hanya dapat dikatakan benar-benar “belajar” apabila nilai performanya meningkat seiring bertambahnya pengalaman. Analogi yang diberikan adalah mahasiswa yang belajar untuk ujian. Jika waktu belajar bertambah tetapi nilai justru menurun, maka proses belajar dianggap gagal. Dalam konteks machine learning, kondisi ini dapat terjadi akibat model yang tidak tepat, seperti overfitting atau underfitting.

2.3 Filosofi Data, Fakta, dan Pengetahuan

Dalam perspektif machine learning, data dipandang sebagai fakta atau ground truth, yaitu kebenaran hasil pengamatan. Selama tidak terdapat kesalahan sensor atau noise, data dianggap benar. Pengetahuan atau model bukanlah kebenaran mutlak, melainkan hipotesis yang konsisten terhadap data yang tersedia. Konsep ini sejalan dengan falsifikasi yang dikemukakan oleh Karl Popper.

Sebagai contoh, hipotesis “semua angsa berwarna putih” akan dianggap benar selama data yang diamati menunjukkan angsa putih. Namun, ketika ditemukan satu angsa berwarna hitam, hipotesis tersebut harus direvisi atau ditolak. Yang salah bukan datanya, melainkan hipotesisnya. Dalam machine learning, prinsip ini berarti bahwa jika model menghasilkan prediksi yang salah, maka model tersebut harus diperbaiki, bukan datanya yang dimanipulasi agar sesuai dengan model.

2.4 Konsep Matematis: Function Approximation

Machine learning juga dijelaskan dari sudut pandang matematis sebagai proses pendekatan fungsi. Terdapat sebuah fungsi ideal yang disebut target function $f(x)$, yaitu fungsi asli yang memetakan input ke output dengan sempurna. Fungsi ini pada umumnya tidak diketahui manusia, misalnya fungsi yang menentukan harga saham atau risiko penyakit seseorang.

Model machine learning membangun sebuah fungsi pendekatan yang disebut hypothesis $h(x)$. Tujuan dari proses pembelajaran adalah mencari $h(x)$ yang sedekat mungkin dengan $f(x)$. Perbedaan antara $h(x)$ dan $f(x)$ disebut error. Proses training dalam machine learning merupakan proses optimasi untuk meminimalkan error tersebut sehingga model semakin mendekati perilaku fungsi sebenarnya.

3. Sample Code / Pseudocode

3.1 Pseudocode Paradigma Induktif Machine Learning::

Input: Dataset (Experience)

Tentukan Task

Tentukan Performance metric

Inisialisasi model $h(x)$

Ulangi:

Gunakan data untuk melatih model

Hitung performa model

Hitung error antara prediksi dan nilai sebenarnya

Perbarui parameter model

Sampai error minimum atau performa stabil

3.2 Contoh Kode Sederhana (Function Approximation – Regresi)

```
import numpy as np
from sklearn.linear_model import LinearRegression
```

```
# Experience (data)
X = np.array([[1], [2], [3], [4], [5]])
y = np.array([2, 4, 6, 8, 10])
```

```
# Model sebagai hypothesis h(x)
model = LinearRegression()
model.fit(X, y)
```

```
# Prediksi
prediksi = model.predict([[6]])
print("Prediksi output:", prediksi)
```

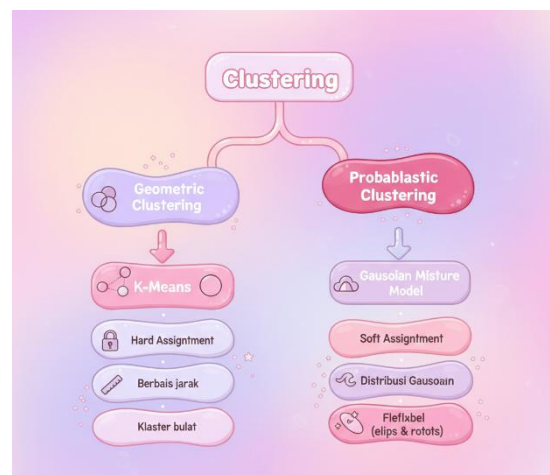
BAB 4 Probabilistic Clustering dengan Gaussian Mixture Models (GMM) dan Algoritma Expectation-Maximization (EM)

1. Deskripsi Singkat Materi

Video keempat membahas teknik clustering berbasis probabilitas menggunakan Gaussian Mixture Models (GMM) serta algoritma Expectation-Maximization (EM) sebagai metode optimasi parameternya. Materi dimulai dari kritik terhadap pendekatan clustering geometris seperti K-Means, lalu beralih ke pendekatan probabilistik yang lebih fleksibel. Selain aspek matematis, video juga menampilkan demonstrasi coding menggunakan Python untuk memvisualisasikan proses pembelajaran GMM pada data multivariat. Di akhir sesi, dosen menekankan pentingnya pemahaman konseptual dibanding sekadar kemampuan menggunakan tools.



Gambar 6 Mind Map Probabilistic Clustering GMM & EM



Gambar 7 Taksonomi Probabilistic Clustering GMM & EM

2. Penjelasan Detail Materi

2.1 Pergeseran Paradigma: Geometris ke Probabilistik

Pendekatan K-Means menggunakan konsep jarak Euclidean dan titik pusat (centroid). Setiap data harus masuk ke satu kluster secara mutlak (hard assignment). Pendekatan ini sederhana, tetapi memiliki asumsi

kuat bahwa semua kluster berbentuk bulat dan berukuran seragam. Jika data berbentuk lonjong atau dua kluster saling menumpuk, K-Means akan menghasilkan pembagian yang tidak akurat.

GMM mengubah sudut pandang ini. Data tidak lagi dianggap hanya sebagai titik kaku, melainkan berasal dari campuran beberapa distribusi probabilitas. Setiap data memiliki peluang menjadi anggota lebih dari satu kluster (soft assignment). Dengan pendekatan ini, bentuk kluster bisa elips dan dapat berotasi mengikuti korelasi antar variabel.

2.2 Anatomi Matematis Gaussian Mixture Models

GMM mengasumsikan bahwa data berasal dari campuran beberapa distribusi normal (Gaussian). Jika histogram data menunjukkan satu puncak, berarti hanya ada satu distribusi. Jika terdapat dua atau lebih puncak, berarti terdapat beberapa sub-populasi yang bercampur.

Setiap kluster dalam GMM didefinisikan oleh tiga parameter utama:

1. **Mean (μ)**
Menentukan lokasi pusat kluster atau puncak distribusi. Ini adalah rata-rata nilai data pada kluster tersebut.
2. **Covariance (Σ)**
Menentukan bentuk dan arah sebaran data.
 - Pada 1 dimensi: berupa varians atau standar deviasi.
 - Pada multi dimensi: berupa matriks kovarians yang memungkinkan kluster berbentuk elips dan dapat berputar.
3. **Mixing Coefficient (π)**
Menunjukkan proporsi kluster terhadap keseluruhan data. Nilai seluruh π harus berjumlah 1.

2.3 Algoritma Expectation-Maximization (EM)

Karena tidak ada label pada data (unsupervised), sistem menghadapi masalah “ayam dan telur”: parameter dibutuhkan untuk menentukan keanggotaan data, tetapi keanggotaan data juga dibutuhkan untuk menghitung parameter.

Algoritma EM menyelesaikan masalah ini secara iteratif:

1. **Initialization**
Menentukan jumlah kluster dan menebak parameter awal (μ , Σ , π) secara acak.
2. **Expectation Step (E-Step)**
Menghitung responsibility (γ), yaitu probabilitas bahwa suatu data berasal dari kluster tertentu.
Nilainya tidak biner, tetapi pecahan, misalnya 0.8 untuk kluster A dan 0.2 untuk kluster B.
3. **Maximization Step (M-Step)**
Parameter diperbarui menggunakan data yang sudah diberi bobot probabilitas:
 - μ baru dihitung sebagai rata-rata tertimbang.
 - Σ baru dihitung dari sebaran tertimbang.
 - π baru dihitung dari proporsi tanggung jawab total.
4. **Evaluasi (Log-Likelihood)**
Setiap iterasi dihitung nilai log-likelihood. Jika masih meningkat signifikan, proses diulang. Jika sudah stabil, algoritma berhenti (konvergen).

2.4 Studi Kasus Multivariat dan Live Coding

Pada data dua dimensi, distribusi Gaussian divisualisasikan sebagai kontur elips. Matriks kovarians memungkinkan elips tersebut miring mengikuti korelasi antar variabel, misalnya nilai Matematika Diskrit dan Machine Learning yang cenderung naik bersama.

Dalam demonstrasi coding, dosen menggunakan Python dan library scikit-learn. Terdapat kendala teknis seperti variabel iterasi yang belum didefinisikan dan kesalahan pemanggilan fungsi visualisasi. Hal ini menjadi pelajaran bahwa pemrograman machine learning tidak hanya soal memanggil fungsi fit(), tetapi juga memahami alur eksekusi dan state variabel untuk debugging.

2.5 Filosofi Pembelajaran Mesin

Perbedaan antara menggunakan AI sebagai komoditas dan memahami AI sebagai ilmu. Menghasilkan kode GMM dengan bantuan AI sangat mudah, tetapi memahami mengapa kurva bergeser, apa arti matriks kovarians, dan bagaimana EM bekerja adalah kompetensi inti seorang sarjana teknik informatika. Oleh karena itu, evaluasi akademik tidak akan menilai sekadar sintaks, melainkan kemampuan analisis terhadap perilaku model.

3. Sample Code / Pseudocode

3.1 Pseudocode Algoritma EM untuk GMM:

Input: Dataset X, jumlah kluster K

Inisialisasi μ , Σ , π secara acak

Ulangi:

E-Step

Untuk setiap data x_n :

 Hitung probabilitas x_n berasal dari tiap kluster k

 Simpan sebagai responsibility γ_{nk}

M-Step

Untuk setiap kluster k:

 Update μ_k dari rata-rata tertimbang data

 Update Σ_k dari kovarians tertimbang data

 Update π_k dari proporsi responsibility

 Hitung log-likelihood

Sampai log-likelihood konvergen

3.2 Contoh Kode dengan scikit-learn:

```
import numpy as np
```

```
from sklearn.mixture import GaussianMixture
```

```
# Data contoh 2 dimensi
```

```
X = np.array([[60, 65], [62, 63], [90, 92], [88, 91], [59, 60], [91, 89]])
```

```
# Inisialisasi GMM dengan 2 kluster
```

```
gmm = GaussianMixture(n_components=2, covariance_type='full', random_state=0)
```

```
gmm.fit(X)
```

```
# Prediksi probabilitas keanggotaan
```

```
proba = gmm.predict_proba(X)
```

```
print("Probabilitas keanggotaan tiap data:")
```

```
print(proba)
```

```
# Prediksi kluster dominan
```

```
labels = gmm.predict(X)
```

```
print("Label kluster:", labels)
```

BAB 5 Decision Tree, Konsep Entropy, dan Algoritma ID3

1. Deskripsi Singkat Materi

Video kelima membahas salah satu algoritma supervised learning yang paling intuitif, yaitu Decision Tree. Materi dimulai dari konsep dasar pohon keputusan sebagai model prediksi berbasis aturan (rule-based), kemudian masuk ke dasar matematis pemilihan atribut terbaik menggunakan konsep entropy dan

information gain. Algoritma yang dijadikan fokus utama adalah ID3 (Iterative Dichotomiser 3), yaitu algoritma klasik pembentuk decision tree berbasis teori informasi.



Gambar 8 Mind Map Decision Tree

2. Penjelasan Detail Materi

2.1 Konsep Dasar Decision Tree

Decision Tree adalah model prediksi berbentuk struktur pohon.

- Root node: atribut pertama yang diuji.
- Internal node: atribut lanjutan.
- Leaf node: hasil keputusan (kelas akhir).

Setiap jalur dari root ke leaf dapat ditulis sebagai aturan:

IF kondisi1 AND kondisi2 THEN kelas = X

Decision Tree banyak digunakan karena:

- Mudah dipahami manusia.
- Bisa diterjemahkan langsung menjadi aturan logika.

2.2 Decision Tree sebagai Supervised Learning

Decision Tree bekerja pada data berlabel.

- Input: fitur (atribut)
- Output: kelas (label)

Contoh:

Dataset mahasiswa

- Fitur: Kehadiran, Nilai Tugas, Nilai UTS
- Label: Lulus / Tidak Lulus

Model belajar dari data latih untuk membuat aturan yang memisahkan kelas sebaik mungkin.

2.3 Masalah Utama: Memilih Atribut Terbaik

Pertanyaan penting: “Atribut mana yang dipakai duluan di root?” Tidak bisa asal pilih, karena urutan atribut sangat memengaruhi kualitas pohon.

Solusinya: gunakan teori informasi.

2.4 Konsep Entropy

Entropy mengukur tingkat ketidakpastian data.

Rumus:

$$\text{Entropy}(S) = -\sum p_i \log_2(p_i)$$

- Jika semua data satu kelas \rightarrow entropy = 0 (pasti).
- Jika data campur rata \rightarrow entropy tinggi (tidak pasti).

Contoh:

Dalam 10 data:

- 5 Lulus
- 5 Tidak Lulus

Entropy tinggi karena sangat tidak pasti.

2.5 Information Gain

Information Gain mengukur seberapa besar penurunan entropy setelah data dipisah oleh suatu atribut.

Rumus:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum |S_v|/|S| \text{Entropy}(S_v)$$

Makna:

- Hitung entropy awal.
- Pecah data berdasarkan atribut A.
- Hitung entropy masing-masing pecahan.
- Atribut terbaik adalah yang memberi penurunan entropy terbesar.

2.6 Algoritma ID3

ID3 adalah algoritma pembentuk decision tree berbasis information gain.

Langkah-langkah:

1. Hitung entropy seluruh data.
2. Untuk setiap atribut, hitung information gain.
3. Pilih atribut dengan gain terbesar sebagai node.
4. Bagi data berdasarkan nilai atribut tersebut.
5. Ulangi proses secara rekursif sampai:
 - Semua data satu kelas, atau
 - Tidak ada atribut tersisa.

2.7 Kelebihan dan Kelemahan

Kelebihan:

- Mudah dipahami.
- Bisa dijelaskan ke non-teknis.
- Cocok untuk data kategorikal.

Kelemahan:

- Mudah overfitting.
- Sensitif terhadap perubahan kecil data.
- Pohon bisa sangat besar dan kompleks.

3. Sample Code / Pseudocode

3.1 Pseudocode ID3:

```
function ID3(dataset, atribut):  
    jika semua data satu kelas:  
        return kelas itu  
    jika atribut kosong:  
        return kelas mayoritas  
  
    pilih atribut A dengan information gain terbesar  
    buat node dengan atribut A  
  
    untuk setiap nilai v pada A:  
        subset = data dengan A = v  
        jika subset kosong:  
            tambahkan leaf dengan kelas mayoritas  
        else:  
            child = ID3(subset, atribut - A)  
            hubungkan child ke node  
  
    return node
```

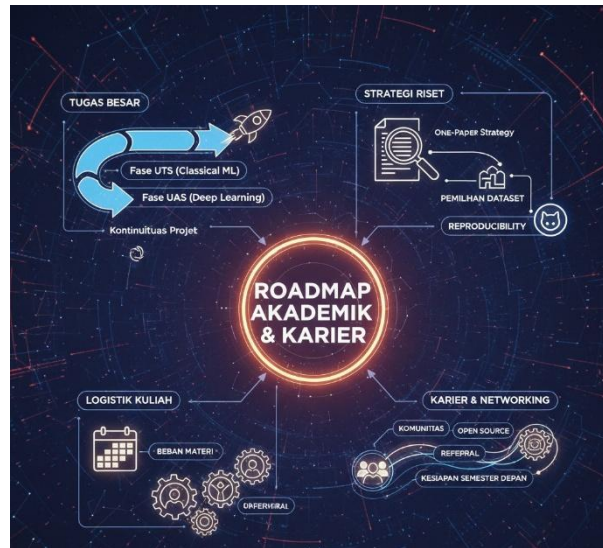
3.2 Contoh Implementasi Python (sketsa)

```
from sklearn.tree import DecisionTreeClassifier  
  
model = DecisionTreeClassifier(criterion="entropy")  
model.fit(X_train, y_train)  
pred = model.predict(X_test)
```

BAB 6 Roadmap Akademik, Strategi Tugas Besar, dan Strategi Karier

1. Deskripsi Singkat Materi

Video keenam tidak lagi berfokus pada algoritma teknis, melainkan pada strategi akademik, perancangan tugas besar, serta persiapan karier di bidang teknologi dan kecerdasan buatan. Dosen menekankan bahwa mata kuliah machine learning bukan hanya soal memahami teori dan koding, tetapi juga tentang bagaimana mahasiswa mengelola proyek, menyusun riset secara sistematis, serta membangun portofolio dan jejaring profesional. Video ini menjadi panduan agar mahasiswa tidak hanya lulus mata kuliah, tetapi juga siap menghadapi dunia industri dan riset.



Gambar 9 Mind Map Roadmap Akademik dan Karier

2. Penjelasan Detail Materi

2.1 Arsitektur Tugas Besar: Kontinuitas UTS ke UAS

Tugas besar dirancang sebagai proyek riset mini yang berlangsung dalam dua tahap yang saling terhubung.

Fase UTS – Classical Machine Learning

Mahasiswa wajib menyelesaikan masalah menggunakan algoritma supervised learning klasik seperti Decision Tree, SVM, Naive Bayes, atau regresi. Tujuannya adalah membangun baseline kinerja model, yaitu tolok ukur awal seberapa baik masalah dapat diselesaikan dengan metode statistik standar.

Fase UAS – Deep Learning

Masalah yang sama sangat disarankan diselesaikan kembali menggunakan pendekatan deep learning seperti neural network, CNN, atau RNN.

Mahasiswa harus menyertakan argumen ilmiah:

- Mengapa deep learning diperlukan?
- Apakah datanya cukup kompleks dan besar?
- Apa hipotesis peningkatan performa dibanding metode klasik?

Dengan pendekatan ini, mahasiswa belajar membandingkan metode dan menganalisis perbedaan hasil secara ilmiah.

2.2 One-Paper Strategy (Strategi Efisiensi Riset)

Strategi agar mahasiswa tidak bekerja dua kali.

Skenario Ideal:

- Sejak UTS memilih dataset yang kompleks.
- Dataset yang sama digunakan untuk UTS dan UAS.
- Laporan hanya satu, tetapi berevolusi.
 - Bab pendahuluan dan dataset tetap.
 - Bab metodologi dan hasil diperluas dengan perbandingan ML klasik vs deep learning.

Skenario Buruk:

- Jika UTS menggunakan dataset terlalu sederhana (misalnya Iris atau Titanic), dataset tersebut tidak cocok untuk deep learning.

- Akibatnya, saat UAS harus mulai dari nol: cari data baru, menulis ulang laporan, dan melakukan eksperimen ulang.

Strategi ini melatih mahasiswa untuk berpikir jangka panjang dalam perencanaan riset.

2.3 Standar Deliverables & Simulasi Software Engineering

Pengumpulan tugas mengikuti standar industri, bukan sekadar kirim file.

Manajemen Repositori GitHub:

- Wajib memiliki satu repositori utama.
- Struktur folder rapi:
 - /UTS
 - /UAS

Aset Wajib (Deliverables):

1. Proposal riset dengan hipotesis ilmiah.
2. Dataset (file mentah atau link sumber resmi).
3. Kode (Jupyter Notebook atau script Python) yang bisa dijalankan ulang.
4. Referensi jurnal ilmiah sebagai dasar teori.
5. Laporan akhir sesuai format jurnal komputer.

Mahasiswa dilatih menjadi peneliti dan software engineer, bukan sekadar pengguna library.

2.4 Hidden Curriculum: Networking dan Realita Industri

Ditekankan bahwa nilai akademik saja tidak cukup.

Kekuatan Komunitas

Banyak pekerjaan terbaik tidak diiklankan secara terbuka, melainkan melalui rekomendasi dari teman atau relasi komunitas.

Ekosistem Open Source:

Komunitas seperti Python ID dan Java User Group sering mengadakan event gratis atau murah. Mahasiswa dianjurkan hadir untuk:

- Belajar langsung dari praktisi.
- Membangun relasi dengan senior industri.

Relasi dipandang sebagai investasi karier jangka panjang.

2.5 Logistik dan Ekspektasi Kesulitan

Machine Learning adalah irisan dari:

- Matematika
- Probabilitas
- Pemrograman

Karena itu, mata kuliah ini berat dan tidak bisa dianggap remeh. Materi ini menjadi fondasi untuk mata kuliah Artificial Intelligence di semester berikutnya. Jika dasar ML lemah, maka materi AI akan jauh lebih sulit.

3. Sample Code / Pseudocode

3.1 Struktur Repositori Tugas Besar:

Project_ML/

```
├── UTS/
│   ├── data/
│   ├── classical_model.ipynb
│   └── proposal.pdf
└── UAS/
```

```

├── data/
│   ├── deep_learning_model.ipynb
│   └── final_report.pdf
└── references/
    ├── paper1.pdf
    └── paper2.pdf
    
```

3.2 Contoh Implementasi Python (sketsa)

Fase UTS

latih_model_klasik(dataset)

hitung_performa -> baseline_score

Fase UAS

latih_model_deep_learning(dataset)

hitung_performa -> dl_score

bandingkan baseline_score dan dl_score

analisis hasil dan jelaskan secara ilmiah

BAB 7 Strategi Tugas Besar, Komunitas, Filosofi Engineering, dan Pipeline Industri

1. Deskripsi Singkat Materi

Video ketujuh membahas keterkaitan antara dunia akademik, dunia industri, dan ekosistem komunitas teknologi. Dosen menjelaskan bagaimana tugas besar dirancang menyerupai proyek riset industri, pentingnya strategi efisiensi pengerjaan, serta peran komunitas dalam membangun karier. Selain itu, video ini menanamkan filosofi dasar perbedaan antara sains dan engineering, serta memperkenalkan machine learning pipeline sebagai siklus hidup proyek di industri. Materi ini menegaskan bahwa machine learning bukan hanya persoalan algoritma, tetapi juga pemahaman bisnis, regulasi, dan penerapan nyata.



Gambar 11 Mind Map Strategi Mahasiswa & Pipeline Industri



Gambar 10 Taksonomi Strategi Mahasiswa & Pipeline Industri

2. Penjelasan Detail Materi

2.1 Grand Design Tugas Besar (UTS & UAS)

Tugas besar dirancang menyerupai proyek riset industri yang dibagi dua fase.

UTS – Machine Learning Klasik

Mahasiswa diminta membangun solusi menggunakan supervised learning klasik seperti Decision Tree, Naive Bayes, SVM, atau regresi. Hasilnya menjadi proof of concept awal dan baseline performa.

UAS – Deep Learning

Solusi yang sama di-upgrade menggunakan deep learning seperti neural network atau CNN. Mahasiswa wajib menjelaskan secara ilmiah mengapa deep learning dibutuhkan dan apa keunggulannya dibanding metode klasik.

One-Paper Strategy

Mahasiswa dianjurkan menggunakan satu kasus yang sama untuk UTS dan UAS. Dengan begitu, laporan cukup dikembangkan, bukan ditulis ulang. Jika sejak awal memilih dataset yang terlalu sederhana, maka saat UAS harus mulai dari nol.

GitHub Workflow

Tugas dikumpulkan melalui GitHub dengan struktur folder UTS dan UAS. Deliverables meliputi proposal dengan hipotesis, kode, referensi jurnal, dan laporan akhir format jurnal komputer.

2.2 Hidden Curriculum: Networking & Komunitas

Dosen menekankan bahwa karier sering ditentukan oleh relasi.

Pentingnya Komunitas

Banyak pekerjaan terbaik tidak didapat lewat lamaran formal, melainkan lewat rekomendasi komunitas.

Event Teknologi

Contoh: PyCon di Universitas Trilogi. Tiket pelajar murah, bahkan dosen menawarkan skema pembayaran kolektif. Event IT biasanya disponsori perusahaan besar sehingga murah tapi fasilitas baik. Mahasiswa didorong hadir bukan hanya untuk materi, tetapi untuk membangun jejaring dengan praktisi senior.

2.3 Filosofi: Science vs. Engineering

Dosen membedakan dua pola pikir:

Science (Ilmuwan) Fokus menjawab pertanyaan: mengapa, bagaimana, kapan suatu fenomena terjadi.

Engineering (Insinyur) Fokus membangun solusi dari hasil sains tersebut. Contoh: dari hasil riset prediksi akademik, engineer membuat aplikasi peringatan dini mahasiswa berisiko drop-out. Mahasiswa informatika diharapkan memiliki mental engineer: tidak hanya paham teori, tetapi mampu mengubahnya menjadi solusi nyata.

2.4 Machine Learning Pipeline (Siklus Hidup Proyek)

Dosen menjelaskan alur kerja ML Engineer:

1. **Data Exploration** Menganalisis statistik dasar dan distribusi data.
2. **Data Analytics & Feature Engineering** Merekayasa fitur agar lebih bermakna bagi model. Tahap ini sangat menentukan kualitas model.
3. **Experimentation (Modeling)** Training, testing, dan tuning parameter.
4. **Development** Mengubah model menjadi service (API, REST, gRPC, library).
5. **Deployment** Menjalankan model di lingkungan produksi. Mahasiswa yang bisa sampai tahap deployment dengan Docker dijanjikan nilai A otomatis.

2.5 Business Understanding & Domain Knowledge

Machine learning adalah matematika terapan. Tanpa memahami konteks bisnis, data hanya menjadi angka kosong.

CRISP-DM

Standar industri yang dimulai dari business understanding.

Studi Kasus Lingkungan

Proyek karbon dan ESG harus dipahami dari sisi regulasi dan tujuan lingkungan (misalnya mencegah Jakarta tenggelam akibat naiknya permukaan laut).

Studi Kasus Fintech

Untuk menganalisis transaksi ATM dan mendeteksi fraud, engineer harus paham alur bisnis keuangan.

2.6 Nasihat Karier: Compliance (Kepatuhan)

Skill Langka (Unicorn Skill)

Banyak orang jago koding, tetapi sedikit yang paham regulasi, ISO, dan keamanan.

Risiko Nyata

Kesalahan di dunia kerja bisa berdampak hukum dan finansial, tidak seperti di kampus yang relatif aman.

Nilai Tinggi di Industri

Menguasai coding sekaligus compliance membuat seseorang sangat bernilai dan bergaji tinggi.

2.7 Arsitektur Teknis: Training vs. Inference

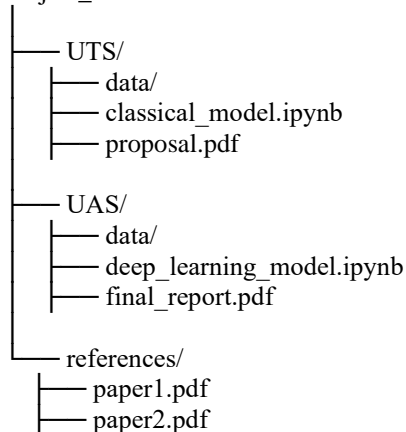
Dosen menjelaskan perbedaan fase belajar dan fase penggunaan model.

- Preprocessing saat training harus sama persis saat inference.
- Jika beda, model akan gagal.
- Pada deep learning, sering ada tahap pre-training sebelum training utama.
- Output akhir pipeline adalah model siap dipakai untuk prediksi nyata.

3. Sample Code / Pseudocode

3.1 Struktur Repositori Tugas Besar:

Project_ML/



BAB 8 Disiplin Akademik dan Validasi Dataset

1. Deskripsi Singkat Materi

Video kedelapan berfokus pada pembentukan mentalitas akademik, etika belajar, serta aspek teknis awal yang sangat krusial dalam proyek machine learning, yaitu pemilihan dan validasi dataset. Dosen menekankan bahwa keberhasilan mahasiswa tidak hanya ditentukan oleh kehadiran formal, tetapi oleh keseriusan belajar dan penguasaan skill nyata. Sesi ini juga diisi dengan bedah kasus nyata dari tugas

mahasiswa yang salah memilih dataset, sehingga menjadi pembelajaran langsung tentang pentingnya memahami struktur data sebelum mulai membangun model.



Gambar 12 Mind Map & Taksonomi Disiplin Akademik dan Validasi Dataset

2. Penjelasan Detail Materi

2.1 Teguran Keras: Etika, Mentalitas, dan Nilai Ijazah

Dosen membuka sesi dengan menegur mahasiswa yang kurang disiplin, seperti mematikan kamera dan tidak responsif saat kuliah online. Menurut dosen, hadir secara formal tanpa belajar sungguh-sungguh adalah kerugian bagi mahasiswa sendiri.

Fenomena “ijazah rasa SMA” muncul ketika lulusan S1 tidak memiliki skill nyata. Dalam kondisi ini, ijazah S1 tidak memberi nilai tambah di dunia kerja karena tidak dibarengi kompetensi. Dunia industri menuntut profesionalisme, kedisiplinan, dan tanggung jawab. Kebiasaan meremehkan perkuliahan akan terbawa ke dunia kerja dan berakibat pada kegagalan karier.

Validasi dari kampus hanya memastikan kehadiran administratif, tetapi validasi kemampuan harus dilakukan oleh mahasiswa sendiri melalui proses belajar yang serius.

2.2 Bedah Kasus Teknis: Kesalahan Fatal Pemilihan Dataset

Dosen melakukan review langsung terhadap dataset yang diajukan mahasiswa. Ditemukan banyak kesalahan mendasar.

Masalah “The Missing Target”

Banyak mahasiswa memilih dataset yang tidak memiliki kolom target (label). Padahal, supervised learning wajib memiliki pasangan:

- X = fitur (input)
- Y = target (output/kunci jawaban)

Jika dataset hanya berisi data mentah tanpa label, maka itu bukan supervised learning, melainkan unsupervised learning. Karena tugas UTS mewajibkan supervised learning, dataset seperti ini langsung dinyatakan tidak valid.

Inkonsistensi Paper dan Dataset

Ada mahasiswa yang melampirkan paper tentang klasifikasi, tetapi dataset yang dipakai tidak memiliki variabel target seperti yang dijelaskan di paper. Ini menunjukkan mahasiswa hanya “asal ambil” tanpa memahami isi paper dan struktur data.

Pelajaran penting: variabel yang disebut dalam paper harus benar-benar ada di dataset yang digunakan.

2.3 Teknis Validasi Data (Data Understanding)

Sebelum mulai coding, mahasiswa harus memahami data.

Cek Metadata dan Deskripsi Dataset

Saat melihat dataset dari Kaggle atau UCI Repository, jangan langsung download. Baca deskripsinya.

Cari kalimat seperti:

- “The target variable is ...”
- “Class attribute is ...”

Identifikasi Tipe Target

Jika target berupa kategori → masalah klasifikasi.

Jika target berupa angka kontinu → masalah regresi.

Jika tidak bisa menentukan kolom target, dataset harus dibuang.

Menangani Data Ambigu

Contoh kolom “Cancel”.

Harus ditentukan apakah “Cancel” adalah target atau hanya fitur.

Jika tujuan proyek adalah memprediksi pembatalan, maka “Cancel” adalah target.

Jika tujuan memprediksi profit, “Cancel” bisa jadi fitur atau malah data bocor (data leakage) yang harus dibuang.

2.4 Instruksi Perbaikan (Corrective Actions)

Dosen memberi ultimatum bagi mahasiswa yang salah dataset:

1. Ganti dataset jika tidak ada target yang jelas.
2. Baca ulang paper referensi dan pahami metodologinya.
3. Pastikan dataset bisa menghasilkan output yang sama seperti di paper.
4. Konsultasi ulang setelah perbaikan agar tidak bekerja sia-sia sampai akhir semester.

3. Sample Code / Pseudocode

3.1 Langkah Validasi Dataset untuk Supervised Learning::

Baca deskripsi dataset

Jika tidak ada kolom target:

Tolak dataset

Tentukan kolom Y (target)

Tentukan kolom X (fitur)

Cek apakah tipe Y:

Kategorikal -> Klasifikasi

Numerik -> Regresi

Jika Y ambigu:

Analisis konteks bisnis

Tentukan peran Y

Jika masih tidak jelas:

Buang dataset

3.2 Contoh Validasi di Python:

```
import pandas as pd

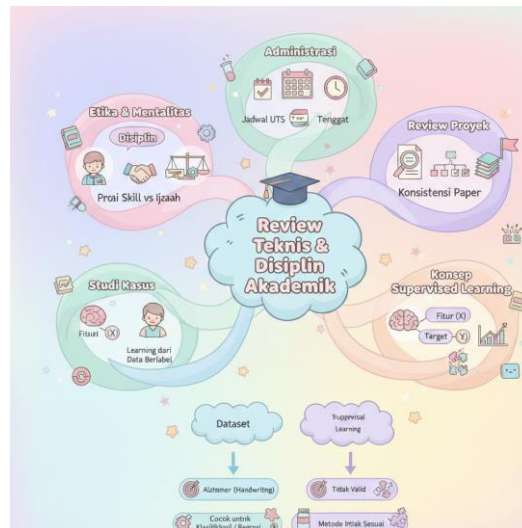
data = pd.read_csv("dataset.csv")
print(data.columns)

target = "label" # misal kolom target
if target not in data.columns:
    print("Dataset tidak valid untuk supervised learning")
else:
    X = data.drop(columns=[target])
    y = data[target]
    print("Dataset valid. Siap untuk supervised learning.")
```

BAB 9 Review Teknis, Disiplin Akademik, dan Validasi Proyek

1. Deskripsi Singkat Materi

Video kesembilan merupakan sesi konsultasi dan evaluasi teknis terhadap proyek mahasiswa. Fokus utama video ini adalah koreksi langsung terhadap kesalahan pemilihan dataset, kesalahan pemahaman konsep supervised learning, serta pembentukan mentalitas profesional. Dosen tidak hanya bertindak sebagai pengajar, tetapi juga sebagai “reviewer industri” yang menguji apakah proyek mahasiswa layak secara akademik dan praktis. Video ini menegaskan bahwa kesalahan di tahap awal (dataset, target, metode) akan menggagalkan seluruh proyek.



Gambar 13 Mind Map Review Teknis dan Validasi Proyek

2. Penjelasan Detail Materi

2.1 Teguran Disiplin dan Nilai Ijazah

Dosen membuka kelas dengan teguran keras terkait etika perkuliahan, seperti kamera mati dan mahasiswa yang tidak responsif. Dosen menekankan bahwa lulus S1 tanpa skill nyata hanya menghasilkan “ijazah rasa SMA”. Di dunia kerja, ijazah hanyalah formalitas, sedangkan kompetensi adalah penentu utama diterima atau tidaknya seseorang.

Kebiasaan tidak disiplin di kampus akan terbawa ke dunia kerja dan berakibat pada kegagalan karier. Kehadiran di kampus bukan hanya soal administrasi, tetapi latihan membentuk sikap profesional.

2.2 Administrasi dan Tenggat Waktu

Dosen mengingatkan bahwa minggu tersebut adalah minggu ke-7 (pengganti), sekaligus periode pengumpulan tugas UTS pada rentang tanggal 14–21. Absensi dilakukan melalui WhatsApp yang dikoordinasi oleh ketua kelas. Hal ini menegaskan bahwa selain teknis, mahasiswa juga harus tertib administrasi.

2.3 Review Teknis Proyek Mahasiswa

Kasus 1: Deteksi Alzheimer dari Tulisan Tangan

Dataset dinilai valid karena memiliki target class yang jelas (Alzheimer: Ya/Tidak). Namun mahasiswa dilarang langsung menggunakan deep learning pada fase UTS. Mereka diminta memakai machine learning klasik terlebih dahulu (misalnya Random Forest) untuk membangun baseline.

Kasus 2: Proposal Salah Konsep

Mahasiswa menyalin seluruh metode dari paper ke proposal. Dosen menegaskan bahwa proposal harus berisi rencana kerja mahasiswa, bukan ringkasan paper. Cukup pilih satu metode yang benar-benar akan digunakan. Masalah lain: paper menggunakan GMM (cenderung unsupervised), padahal tugas UTS wajib supervised. Mahasiswa diminta mengganti metode atau mencari paper lain yang sesuai.

Kasus 3: Dataset Konsumsi Obat (Drug Consumption)

Dataset ini membingungkan karena memiliki banyak kolom target potensial (18 jenis obat). Dosen menjelaskan:

- Kolom demografi = fitur (X).
- Kolom riwayat penggunaan obat = target (Y).

Namun, tidak mungkin memprediksi semua target sekaligus. Mahasiswa harus memilih satu target, misalnya “penggunaan ganja”, lalu menjadikan kolom lain sebagai fitur.

2.4 Konsep Kunci: Target vs. Fitur

Dosen menggunakan analogi Imam Bukhari:

- Data (X): sanad, perawi, isi hadis.
- Label (Y): sahih atau palsu.

Dengan belajar dari data berlabel, Imam Bukhari mampu menilai hadis baru. Mesin juga bekerja demikian: tanpa kolom target (Y), mesin tidak bisa belajar dalam supervised learning.

3. Sample Code / Pseudocode

3.1 Validasi Dataset untuk Supervised Learning::

Baca dataset

Jika tidak ada kolom target:

Tolak dataset

Tentukan kolom Y (target)

Tentukan kolom X (fitur)

Jika metode tidak sesuai (unsupervised dipakai untuk tugas supervised):

Ganti metode atau dataset

3.2 Contoh Penentuan X dan Y di Python:

```
import pandas as pd

data = pd.read_csv("drug_consumption.csv")

# Misal kita ingin memprediksi penggunaan ganja
target = "Cannabis_Use"

X = data.drop(columns=[target])
y = data[target]

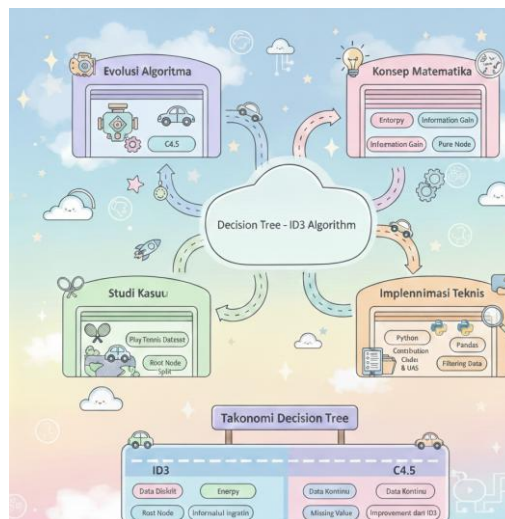
print("Fitur (X):", X.columns)
print("Target (Y):", y.name)
```

BAB 10 Algoritma ID3 dan Matematika di Balik Decision Tree

1. Deskripsi Singkat Materi

Video kesepuluh membahas secara mendalam algoritma pembentuk Decision Tree, khususnya ID3 (Iterative Dichotomiser 3), beserta fondasi matematis yang membuat pohon keputusan tidak sekadar “if-else asal”, tetapi hasil dari perhitungan statistik yang sistematis. Dosen menekankan bahwa memahami logika matematika di balik decision tree jauh lebih penting daripada sekadar memanggil fungsi dari library seperti scikit-learn.

Materi dimulai dari perbandingan ID3 dengan C4.5, lalu masuk ke konsep entropi dan information gain, dilanjutkan dengan studi kasus klasik “Play Tennis Dataset” yang dihitung manual langkah demi langkah. Di akhir, dosen menunjukkan implementasi teknis dengan Python dan Pandas, serta menutup dengan aturan administrasi tugas kelompok melalui GitHub.



Gambar 14 Mind Map Algoritma ID3 dan Matematika di Balik Decision Tree

2. Penjelasan Detail Materi

2.1 ID3 vs C4.5: Evolusi Algoritma

Dosen menjelaskan bahwa ID3 adalah algoritma klasik yang menjadi dasar pembelajaran decision tree. Algoritma ini cocok untuk data kategorikal seperti “Sunny”, “Rainy”, atau “Overcast”. Namun ID3 memiliki keterbatasan besar: sulit menangani data numerik (kontinu) dan tidak ramah terhadap missing value.

C4.5 adalah pengembangan dari ID3 yang lebih fleksibel. Ia mampu membagi data numerik menggunakan threshold (misalnya suhu > 30 atau ≤ 30) dan bisa menangani data kosong. Namun, dalam perkuliahan ini, fokus tetap pada ID3 agar mahasiswa memahami logika dasarnya sebelum naik ke versi yang lebih canggih.

2.2 Entropi: Ukuran Kekacauan Data

Entropi adalah ukuran ketidakpastian dalam sebuah kumpulan data. Semakin campur aduk datanya, semakin tinggi entropinya.

Rumus:

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

Maknanya:

- Jika data berimbang (misalnya 5 “Yes”, 5 “No”), maka entropi tinggi, mendekati 1.
- Jika data seragam (10 “Yes”, 0 “No”), maka entropi mendekati 0. Ini disebut node murni (pure node).

Tujuan decision tree adalah mengubah data yang awalnya “kacau” menjadi kumpulan node yang semakin murni melalui serangkaian pertanyaan.

2.3 Information Gain: Memilih Pertanyaan Terbaik

Information Gain (IG) mengukur seberapa efektif sebuah atribut dalam mengurangi kekacauan data.

Rumus sederhananya:

$$\text{Gain} = \text{Entropi Awal} - \text{Entropi Setelah Dibagi Atribut}$$

Atribut dengan nilai Gain tertinggi akan dipilih sebagai pertanyaan pertama (root node), karena ia paling mampu “merapikan” data.

2.4 Studi Kasus: Play Tennis Dataset

Dataset ini memiliki fitur:

- Outlook
- Temperature
- Humidity
- Wind

Target:

- Play (Yes/No)

Langkah 1: Hitung Entropi Awal

Dosen menghitung berapa jumlah “Yes” dan “No” di seluruh dataset, lalu dimasukkan ke rumus entropi untuk mendapatkan base entropy.

Langkah 2: Hitung Gain Tiap Atribut

Setiap atribut diuji:

1. Outlook → Sunny, Overcast, Rainy
2. Temperature → Hot, Mild, Cool
3. Humidity → High, Normal
4. Wind → Weak, Strong

Untuk setiap atribut:

- Data dipecah sesuai nilai atribut.

- Hitung entropi tiap cabang.
- Hitung rata-rata tertimbang.
- Kurangkan dari entropi awal.

Hasilnya: Outlook memiliki Information Gain terbesar. Maka Outlook dipilih sebagai root node.

Langkah 3: Analisis Cabang

- Cabang Overcast: Semua data bernilai “Yes”.
 - Entropi = 0
 - Tidak perlu dipecah lagi. Ini menjadi leaf node.
- Cabang Sunny dan Rainy: Masih ada campuran Yes dan No.
 - Entropi > 0
 - Lakukan proses yang sama secara rekursif, tapi hanya pada data di cabang tersebut dan dengan fitur yang tersisa.

Proses ini berulang sampai semua cabang menjadi murni atau tidak ada lagi fitur yang bisa dipakai.

2.5 Implementasi dengan Python dan Pandas

Setelah menghitung manual, dosen menunjukkan implementasi logikanya dalam kode.

- Filtering data:
Menggunakan ekspresi seperti:
`df[df['Outlook'] == 'Sunny']`
untuk mengambil subset data sesuai cabang pohon.
- Drop kolom:
Setelah Outlook dipakai sebagai root, kolom tersebut dihapus dari perhitungan berikutnya agar tidak dipilih ulang.
- Automasi:
Kode akan secara otomatis menghitung entropi dan gain untuk menentukan atribut terbaik di setiap level.

Ini menunjukkan bahwa library ML sebenarnya hanya mengotomatisasi logika matematika yang sama seperti yang dihitung manual.

2.6 Implementasi dengan Python dan Pandas

Administrasi: Validasi Kontribusi via GitHub

Di akhir video, dosen memberi instruksi tegas untuk tugas kelompok:

- Penilaian tidak dirata.
- Dosen akan melihat GitHub Insights dan riwayat commit.
- Siapa yang tidak pernah commit dianggap tidak bekerja.

Struktur repositori:

- Satu repositori per kelompok.
- Di dalamnya ada folder UTS dan folder UAS.

Dengan ini, kerja tim menjadi transparan dan adil.

3. Sample Code / Pseudocode

3.1 Pseudocode ID3 Sederhana:

Input: Dataset D, atribut A, target Y

Jika semua data di D memiliki Y yang sama:

Return leaf node dengan nilai Y

Jika A kosong:

Return leaf node dengan nilai mayoritas Y

Hitung Gain untuk tiap atribut di A

Pilih atribut dengan Gain terbesar sebagai node

Untuk tiap nilai atribut:

Ambil subset data

Bangun subtree secara rekursif

3.2 Contoh Filtering di Python:

```
import pandas as pd
```

```
df = pd.read_csv("play_tennis.csv")
```

```
# Memecah data berdasarkan Outlook
```

```
sunny = df[df['Outlook'] == 'Sunny']
```

```
rainy = df[df['Outlook'] == 'Rainy']
```

```
overcast = df[df['Outlook'] == 'Overcast']
```

```
# Setelah Outlook dipakai, kolom itu dibuang
```

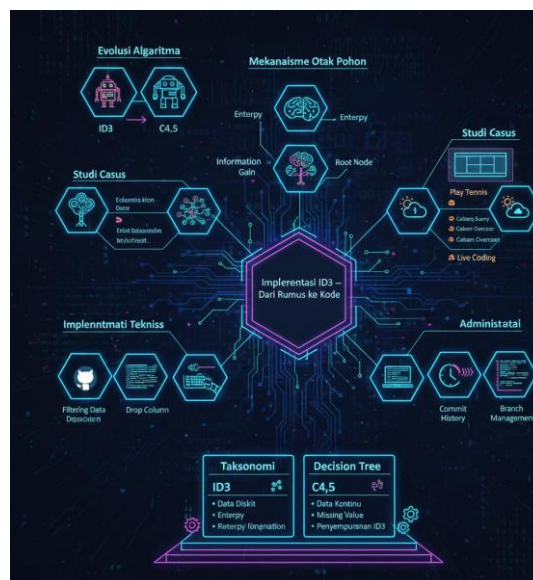
```
sunny = sunny.drop(columns=['Outlook'])
```

BAB 11 Implementasi Teknis Algoritma ID3

1. Deskripsi Singkat Materi

Video kesebelas merupakan kelanjutan langsung dari pembahasan Decision Tree, dengan fokus utama pada implementasi teknis algoritma ID3 menggunakan Python. Jika pada video sebelumnya mahasiswa diajak “berkeringat” menghitung entropi dan information gain secara manual, maka pada video ini dosen menunjukkan bagaimana logika matematika tersebut diterjemahkan ke dalam kode.

Tujuan utama sesi ini bukan sekadar bisa menjalankan program, tetapi memahami hubungan antara rumus statistik dan baris-baris kode. Mahasiswa diajak melihat bahwa setiap filter, drop, dan perhitungan di Pandas sebenarnya adalah representasi langsung dari langkah-langkah teoritis ID3.



Gambar 15 Mind Map Implementasi Teknis Algoritma ID3

2. Penjelasan Detail Materi

2.1 ID3 vs C4.5: Posisi Materi

Dosen mengulang bahwa ID3 dipilih sebagai bahan belajar karena:

- Sederhana dan logikanya transparan.
- Cocok untuk data kategorikal seperti “Yes/No”, “Sunny/Rainy”.

C4.5 disebut sebagai versi lanjutan:

- Bisa menangani data numerik.
- Bisa mengelola missing values.

Namun, mahasiswa diminta tidak “loncat” ke C4.5 atau library instan sebelum benar-benar paham logika ID3.

2.2 Mekanisme Otak Decision Tree

Decision Tree bekerja dengan satu pertanyaan utama:

“Fitur mana yang paling layak dijadikan pertanyaan pertama?”

Untuk menjawabnya digunakan dua konsep:

1. Entropy (H)

Ukuran kekacauan data.

- Campur (5 Yes, 5 No) → Entropy tinggi.
- Seragam (10 Yes, 0 No) → Entropy = 0.

2. Information Gain (IG)

Mengukur seberapa besar suatu fitur mampu menurunkan entropy.

Fitur dengan Gain terbesar = kandidat terbaik jadi root node.

Dengan kata lain, pohon keputusan selalu memilih pertanyaan yang paling “membersihkan” data.

2.3 Studi Kasus: Play Tennis (Lanjutan)

Tahap 1 – Menentukan Root Global

Semua fitur diuji:

- Outlook
- Temperature
- Humidity
- Wind

Setelah dihitung Gain-nya, ternyata:

- Outlook memiliki Gain terbesar.
→ Root node = Outlook.
→ Pertanyaan pertama: “Bagaimana cuacanya?”

Tahap 2 – Analisis Cabang Outlook

- Overcast
Semua data = “Yes”.
→ Entropy = 0.
→ Tidak perlu dipecah lagi, langsung jadi leaf node.
- Sunny
Data masih campur (Yes/No).
→ Harus dipecah lagi.

Tahap 3 – Iterasi di Cabang Sunny

- Fitur Outlook dibuang dari perhitungan.
- Sisa fitur: Temperature, Humidity, Wind.
- Hitung ulang Gain hanya untuk data Sunny.
- Hasil: Humidity punya Gain terbesar.

Maka aturan baru:

Jika Sunny → cek Humidity.

Ini adalah proses rekursif: setiap cabang diperlakukan seperti “masalah baru” sampai datanya murni.

2.4 Implementasi Python Live Coding

Dosen menerjemahkan logika ini ke dalam Pandas.

1. Filtering Data

Untuk memecah data sesuai cabang:

```
sunny = df[df['Outlook'] == 'Sunny']  
rainy = df[df['Outlook'] == 'Rainy']  
overcast = df[df['Outlook'] == 'Overcast']
```

Ini sama seperti membagi dataset berdasarkan nilai atribut.

2. Drop Column

Setelah Outlook jadi root:

```
sunny = sunny.drop(columns=['Outlook'])
```

Supaya fitur yang sudah dipakai tidak ikut dihitung lagi.

3. Rekursif Manual

Urutan yang dilakukan dosen:

1. Hitung entropi awal dataset.
2. Hitung Gain tiap fitur → pilih Outlook.
3. Filter data “Sunny”.
4. Hitung ulang Gain di data Sunny → pilih Humidity.
5. Ulangi sampai semua cabang murni.

Mahasiswa ditunjukkan bahwa walaupun library seperti scikit-learn bisa membuat decision tree dalam satu baris kode, di balik layar ia melakukan proses yang sama persis seperti ini.

2.5 Administrasi & Penilaian GitHub

Di akhir sesi, dosen menekankan aspek profesionalisme dalam kerja kelompok.

- Penilaian tidak rata.
- Dosen akan membuka tab GitHub Insights / Contributors.
- Yang tidak punya commit history dianggap tidak bekerja.

Ciri “penumpang gelap”:

- Tidak pernah commit.
- Hanya 1–2 commit kecil (sekadar typo atau copy-paste).

Solusi:

- Bagi tugas jelas.
- Gunakan branch untuk kerja paralel.
- Pastikan setiap anggota punya kontribusi nyata di kode.

3. Sample Code / Pseudocode

3.1 Pseudocode ID3 Sederhana:

Fungsi BuildTree(Data, Atribut):

Jika semua target sama:

Return Leaf

Jika atribut habis:

Return Leaf mayoritas

Hitung Gain tiap atribut

Pilih atribut dengan Gain terbesar

Untuk tiap nilai atribut:

SubData = filter(Data, nilai)

SubTree = BuildTree(SubData, atribut_sisa)

Hubungkan ke node

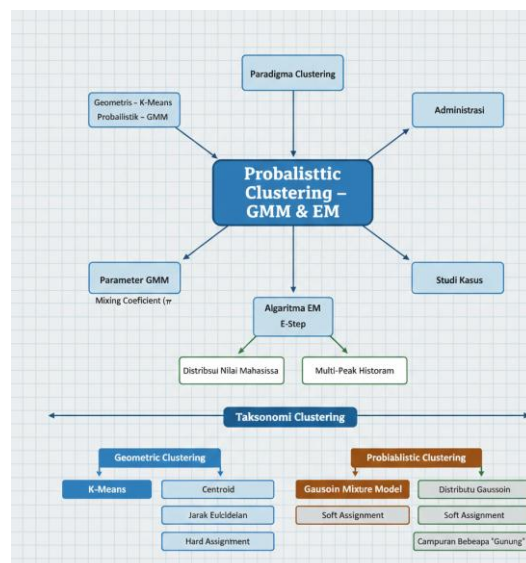
Return Tree

BAB 12 Gaussian Mixture Models (GMM) & Expectation-Maximization Algorithm

1. Deskripsi Singkat Materi

Video kedua belas membahas pendekatan clustering berbasis probabilitas menggunakan Gaussian Mixture Models (GMM) dan algoritma Expectation-Maximization (EM) sebagai mesin optimasinya. Materi ini menjadi lanjutan alami dari pembahasan unsupervised learning, dengan tujuan memperkenalkan cara berpikir yang lebih fleksibel dibanding metode geometris seperti K-Means.

Dosen tidak hanya membahas sisi teknis dan matematis, tetapi juga menggunakan analogi sehari-hari seperti distribusi nilai mahasiswa untuk membantu mahasiswa memahami konsep “campuran distribusi” (mixture). Di akhir sesi, kembali ditekankan bahwa mata kuliah ini berat karena berada di persimpangan matematika, probabilitas, dan pemrograman, serta menjadi fondasi mutlak sebelum masuk ke mata kuliah Artificial Intelligence dan Deep Learning.



Gambar 16 Mind Map GMM & EM

2. Penjelasan Detail Materi

2.1 Paradigma Clustering: Geometris vs Probabilistik

Dosen memulai dengan membandingkan dua cara pandang utama dalam clustering:

K-Means (Geometris):

- Setiap kluster direpresentasikan oleh satu titik pusat (centroid).
- Kedekatan data diukur dengan jarak Euclidean.
- Satu data hanya boleh masuk ke satu kluster (hard clustering).
- Asumsi bentuk kluster: bulat dan simetris.

Masalah muncul ketika data:

- Berbentuk lonjong, miring, atau saling tumpang tindih.
- Memiliki kepadatan berbeda antar kluster.

Dalam kondisi ini, K-Means sering “memaksa” pembagian yang tidak sesuai struktur asli data.

GMM (Probabilistik):

- Kluster dianggap sebagai distribusi Gaussian.
- Data dilihat sebagai tumpukan beberapa “gunung” probabilitas.
- Setiap titik data punya peluang menjadi anggota beberapa kluster (soft clustering).

Dengan pendekatan ini:

- Bentuk kluster bisa elips.
- Bisa miring dan mengikuti korelasi antar fitur.
- Lebih realistis untuk data dunia nyata.

2.2 Komponen Matematika Gaussian Mixture Model

Setiap kluster dalam GMM ditentukan oleh tiga parameter:

1. Mean (μ)

- Titik puncak gunung.
- Lokasi pusat distribusi.
- Rata-rata nilai data dalam kluster tersebut.

2. Covariance (Σ)

- Menentukan lebar, sempit, dan arah sebaran data.
 - Pada 1 dimensi: cukup varians.
 - Pada multi dimensi: matriks kovarians \rightarrow memungkinkan elips yang bisa berotasi.
- Ini membuat GMM jauh lebih fleksibel daripada K-Means yang hanya “bulat”.

3. Mixing Coefficient (π)

- Proporsi kluster terhadap keseluruhan populasi.
- Misal: Kluster A = 60%, Kluster B = 40%.
- Total semua π harus = 1.

2.3 Algoritma Expectation-Maximization (EM)

Masalah utama dalam GMM adalah:

Kita tidak tahu kluster mana milik data, tapi untuk tahu kluster kita butuh parameter, dan untuk tahu parameter kita butuh kluster. Ini disebut masalah “ayam dan telur”.

EM menyelesaikannya secara iteratif:

1. Inisialisasi

- Tebak awal μ , Σ , dan π secara acak.
- Bisa juga memakai K-Means sebagai tebakan awal.

2. E-Step (Expectation)

- Hitung responsibility (γ).
- Untuk tiap data x dan kluster k , hitung: “Seberapa besar peluang data ini berasal dari kluster k ?”
- Output: nilai probabilitas antara 0 dan 1.

3. M-Step (Maximization)

- Update parameter menggunakan data yang sudah diberi bobot probabilitas.
- μ baru = rata-rata tertimbang.
- Σ baru = kovarians tertimbang.
- π baru = proporsi total tanggung jawab kluster.

4. Evaluasi (Log-Likelihood)

- Mengukur seberapa cocok model dengan data.
- Jika peningkatan kecil atau stabil \rightarrow berhenti (konvergen).
- Jika masih naik signifikan \rightarrow ulangi lagi.

Proses ini terus berjalan sampai model “puas” dengan parameternya.

2.4 Studi Kasus: Distribusi Nilai Mahasiswa

Dosen memberi analogi konkret:

- Jika nilai mahasiswa berkisar 0–100 dan histogramnya punya satu puncak besar di sekitar 70 \rightarrow berarti satu distribusi utama.
- Tapi jika ada puncak kecil di sekitar 95 \rightarrow ada sub-kelompok mahasiswa yang sangat pintar.

GMM akan:

- Mendeteksi bahwa data bukan berasal dari satu “gunung”.
- Tapi dari dua “gunung” yang bertumpuk.
- Lalu mencari parameter masing-masing gunung secara otomatis.

Inilah kekuatan GMM:

Tidak memaksakan semua data ke satu pola, tetapi mengakui bahwa populasi bisa terdiri dari beberapa sub-populasi.

Aspek	K-Means	GMM
Pendekatan	Geometris	Probabilistik
Bentuk Kluster	Bulat	Elips, fleksibel
Keanggotaan	Keras (0/1)	Lunak (probabilitas)
Output	Label kluster	Probabilitas per kluster
Cocok untuk	Data sederhana	Data kompleks & tumpang tindih

Tabel 1 Geometris vs Probabilistik

2.5 Administrasi & Penilaian GitHub

Di akhir sesi, dosen kembali mengingatkan:

- Mata kuliah Machine Learning berat karena:
 - Menggabungkan matematika, probabilitas, dan pemrograman.
- Ini adalah prasyarat mutlak sebelum masuk ke mata kuliah Artificial Intelligence di semester berikutnya.
- Jika gagal paham di ML, maka di AI (yang sudah masuk Deep Learning) mahasiswa akan kesulitan jauh lebih besar.

Ada juga diskusi jadwal:

- Dicari waktu pengganti yang lebih manusiawi:

- Jumat pagi/sore, atau
- Online malam hari, supaya mahasiswa belajar dalam kondisi segar, bukan setengah tidur.

3. Sample Code / Pseudocode

3.1 Pseudocode EM untuk GMM

Input: Data X , jumlah kluster K

Inisialisasi μ , Σ , π secara acak

Ulangi:

 # E-Step

 Untuk setiap data x_n :

 Untuk setiap kluster k :

 Hitung $\gamma_{nk} = P(\text{kluster } k | x_n)$

 # M-Step

 Untuk setiap kluster k :

 Update μ_k dari rata-rata tertimbang

 Update Σ_k dari kovarians tertimbang

 Update π_k dari proporsi responsibility

 Hitung log-likelihood

Sampai log-likelihood konvergen

Kesimpulan

Berdasarkan rangkuman seluruh materi perkuliahan Machine Learning dari berbagai video, dapat disimpulkan bahwa pembelajaran tidak hanya berfokus pada penggunaan algoritma, tetapi pada pembentukan cara berpikir ilmiah dan rekayasa dalam memecahkan masalah nyata. Mahasiswa diajak memahami machine learning dari tiga sisi utama: filosofi data dan pengetahuan, dasar matematis, serta implementasi teknis yang dapat diterapkan di dunia industri.

Materi dimulai dari pergeseran paradigma pemrograman deduktif (expert system) menuju pendekatan induktif (machine learning), di mana mesin belajar dari data. Definisi formal Tom Mitchell menegaskan bahwa sebuah sistem dikatakan belajar hanya jika performanya meningkat seiring bertambahnya pengalaman (data). Konsep ini diperkuat dengan pandangan filosofis bahwa data adalah fakta, sedangkan model hanyalah hipotesis yang harus tunduk pada data.

Pada aspek teknis, mahasiswa diperkenalkan dengan berbagai algoritma, mulai dari metode klasik seperti Decision Tree (ID3), K-Means, hingga pendekatan probabilistik seperti Gaussian Mixture Models (GMM) dengan algoritma Expectation-Maximization (EM). Perbandingan antara pendekatan geometris dan probabilistik menunjukkan bahwa tidak ada satu metode yang selalu terbaik; pemilihan algoritma harus disesuaikan dengan karakteristik data dan tujuan masalah.

Selain algoritma, mahasiswa juga dibekali pemahaman tentang pipeline machine learning yang lengkap, mulai dari business understanding, eksplorasi data, feature engineering, modeling, hingga deployment. Proyek UTS dan UAS dirancang menyerupai riset mini agar mahasiswa terbiasa dengan proses ilmiah, penulisan laporan, penggunaan GitHub, serta kerja tim berbasis kontribusi nyata.

Aspek non-teknis juga mendapat penekanan kuat, seperti disiplin, etika akademik, mentalitas profesional, pentingnya komunitas, serta pemahaman regulasi dan compliance. Semua ini menunjukkan bahwa seorang praktisi machine learning tidak cukup hanya jago koding, tetapi juga harus mampu memahami domain masalah, bekerja secara bertanggung jawab, dan siap menghadapi realitas industri.

Secara keseluruhan, rangkaian materi ini membentuk mahasiswa menjadi calon engineer yang tidak hanya mampu membangun model, tetapi juga mampu menganalisis data secara kritis, memilih metode secara tepat, membangun solusi yang dapat digunakan di dunia nyata, serta memiliki sikap profesional dalam karier di bidang teknologi dan kecerdasan buatan.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada dosen pengampu mata kuliah Machine Learning, Bapak Hendri Karisma, S.Kom., M.T., atas bimbingan, arahan, motivasi, serta penjelasan yang sistematis dan inspiratif selama proses perkuliahan dan penyusunan rangkuman ini. Materi yang disampaikan tidak hanya memperkaya pemahaman teknis, tetapi juga membentuk cara berpikir kritis, ilmiah, dan profesional dalam bidang teknologi dan kecerdasan buatan.

Ucapan terima kasih juga disampaikan kepada seluruh pihak yang berkontribusi dalam penyediaan sumber belajar, baik penulis buku rujukan seperti Tom Mitchell dan Stuart Russell, maupun pengelola platform data dan pembelajaran terbuka yang menyediakan dataset, referensi, serta perangkat lunak pendukung. Tanpa dukungan sumber-sumber tersebut, proses belajar, eksplorasi, dan penyusunan rangkuman ini tidak akan berjalan dengan baik.

Terakhir, penulis juga berterima kasih kepada teman-teman sekelas dan komunitas belajar yang telah menjadi ruang diskusi, berbagi pengetahuan, serta saling memotivasi selama proses perkuliahan, sehingga pengalaman belajar Machine Learning menjadi lebih bermakna dan aplikatif.

