



FAKULTAS ILMU KOMPUTER
UNIVERSITAS MEGA BUANA PALOPO
S1 SARJANA SISTEM INFORMASI

MODUL DATA MINING DAN MACHINE LEARNING BERBASIS PYTHON

Modul praktikum ini dirancang untuk memberikan pemahaman konsep, keterampilan analisis data, serta penerapan algoritma data mining dan machine learning menggunakan Python.



MATA KULIAH
Data Mining dan Machine Learning



KODE MATA KULIAH
SIP2311



BOBOT (SKS)
3 SKS



SEMESTER
6

```
import pandas as pd
import numpy as np
from sklearn.model_selection import
from sklearn.ensemble import
from sklearn.metrics import
import matplotlib.pyplot as plt
import seaborn as sns
```



*Menggali Data,
Membangun Wawasan,
Mendukung Keputusan Berbasis Data*

IDENTITAS MODUL

Mata Kuliah	Data Mining dan Machine Learning
Kode Mata Kuliah	SIP2311
Bobot	3 SKS
Program Studi	S1 Sistem Informasi
Fakultas	Fakultas Ilmu Komputer
Perguruan Tinggi	Universitas Mega Buana Palopo
Penyusun	1. Abdul Malik, S.Kom., M.Cs. 2. Fahmi Kurniawan, S.Kom., M.M. 3. M. Hasanuddin, S.Kom., M.Kom.
Tahun	2026
Bahasa Pemrograman	Python
Lingkungan Praktikum	Google Colab/Jupyter Notebook

LEMBAR PENGESAHAN
MODUL PRAKTIKUM DATA MINING DAN MACHINE LEARNING
BERBASIS PYTHON

Modul Praktikum Data Mining dan Machine Learning Berbasis Python ini telah diperiksa, ditelaah, dan disahkan untuk digunakan sebagai pedoman dalam pelaksanaan kegiatan praktikum pada:

Program Studi	S1 – Sistem Informasi
Fakultas	Fakultas Ilmu Komputer
Perguruan Tinggi	Universitas Mega Buana Palopo
Mata Kuliah	Data Mining dan Machine Learning
Kode Mata Kuliah	SIP2311
Semester	Genap
Tahun Akademik	2025/2026

Palopo, 13 April 2026

Disusun oleh,
Dosen Pengembang Modul Praktikum



Abdul Malik, S.Kom., M.Cs.
NIDN. 0910049004

Diperiksa dan Direkomendasikan oleh
Ketua Program Studi Sistem Informasi




M. Hasanuddin, S.Kom., M.Kom
NIDN. 0927098905

Menyetujui
Plh. Dekan Fakultas Ilmu Komputer




Abdul Malik, S.Kom., M.C.s
NIDN. 0910049004

KATA PENGANTAR

Modul Praktikum Data Mining dan Machine Learning Berbasis Python disusun sebagai panduan kegiatan praktikum bagi mahasiswa Program Studi S1 Sistem Informasi. Modul ini dikembangkan berdasarkan Rencana Pembelajaran Semester mata kuliah Data Mining dan Machine Learning dan diarahkan untuk mendukung pencapaian kemampuan mahasiswa dalam pengolahan data, pemodelan, evaluasi, visualisasi, interpretasi, dan penyelesaian proyek berbasis data.

Kegiatan praktikum menggunakan Python dengan dukungan Google Colab atau Jupyter Notebook, pandas, NumPy, Matplotlib, dan scikit-learn. Melalui pendekatan praktis, mahasiswa diharapkan mampu menghubungkan konsep yang dipelajari di kelas dengan permasalahan nyata dalam bidang Sistem Informasi.

Modul ini juga menekankan dokumentasi proses analisis, integritas akademik, etika penggunaan data, privasi, bias model, serta tanggung jawab dalam menyampaikan hasil dan rekomendasi.

Palopo, 2026

Penulis

DAFTAR ISI

SAMPUL	i
IDENTITAS MODUL.....	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Deskripsi Mata Kuliah	4
1.3 Tujuan Penyusunan Modul	6
1.4 Sasaran Pengguna	7
1.5 Prasyarat Kompetensi	9
1.6 Ruang Lingkup Modul.....	13
BAB II CAPAIAN PEMBELAJARAN DAN PETA PRAKTIKUM.....	22
2.1 Capaian Pembelajaran Lulus (CPL).....	22
2.2 Capaian Pembelajaran Mata Kuliah (CPMK).....	27
2.3 Sub-Capaian Pembelajaran Mata Kuliah (Sub-CPMK)	32
2.4 Pemetaan CPMK dan Sub-CPMK terhadap Modul Praktikum.....	39
2.5 Peta Alur Pembelajaran.....	48
BAB III PETUNJUK PENGGUNAAN MODUL	56
3.1 Petunjuk Penggunaan untuk Mahasiswa.....	56
3.2 Petunjuk Penggunaan untuk Dosen atau Asisten Praktikum	63
3.3 Tata Tertib Praktikum	72
3.4 Integritas Akademik dan Etika Penggunaan AI.....	73
3.5 Pengumpulan Tugas melalui LMS Edlink.....	74
BAB IV PERSIAPAN LINGKUNGAN PRAKTIKUM	76
4.1 Kebutuhan Perangkat Keras.....	76

4.2	Perangkat Lunak dan Platform.....	76
4.3	Pengenalan Google Colab.....	77
4.4	Struktur Notebook Praktikum.....	77
4.5	Mengimpor Library.....	78
4.6	Membaca Dataset.....	79
4.7	Mengunggah dan Mengelola Dataset.....	79
4.8	Pemeriksaan Awal Dataset	80
4.9	Mengenal pandas dan NumPy	80
4.10	Pengenalan Matplotlib	81
4.11	Pengenalan scikit-learn	81
4.12	Penanganan Kesalahan Dasar	81
4.13	Penanganan Kesalahan Dasar	82
4.14	Kesiapan Sebelum Memulai Modul Praktikum.....	82
MODUL 1 PENGENALAN PYTHON DAN GOOGLE COLAB.....		83
MODUL 2 CRISP-DM DAN PERUMUSAN MASALAH ANALITIK		95
MODUL 3 JENIS DATA DAN EXPLORATORY DATA ANALYSIS		108
MODUL 4 PEMBERSIHAN DATA		123
MODUL 5 TRANSFORMASI DATA DAN PERSIAPAN DATA UNTUK PEMODELAN.....		151
MODUL 6 KLASIFIKASI DENGAN K-NEAREST NEIGHBORS, DECISION TREE, DAN NAIVE BAYES		176
MODUL 7 REGRESI LINEAR UNTUK PREDIKSI DATA BISNIS		201
MODUL 8 CLUSTERING MENGGUNAKAN ALGORITMA K-MEANS		227
DAFTAR PUSTAKA.....		266

DAFTAR TABEL

Tabel 1. Hubungan CPL dengan Aktivitas Modul	26
Tabel 2. Hubungan CPMK dan Kegiatan Praktikum	32
Tabel 3. Hubungan CPMK dan Sub-CPMK	38
Tabel 4. Pemetaan Modul Praktikum	39
Tabel 5. Matriks Modul Praktikum dan CPMK	47
Tabel 6. Hubungan Tahapan CPMK	54
Tabel 7. Tipe Data Dasar Python.....	86
Tabel 8. Operator Aritmatika Python	87
Tabel 9. Operator Perbandingan.....	88
Tabel 10. Lembar Kerja CRISP-DM.....	102
Tabel 11. Contoh Dataset dan Atribut	110
Tabel 11. Contoh Struktur Dataset Klasifikasi	181

DAFTAR GAMBAR

Gambar 1. Alur Pembelajaran Modul.....	48
Gambar 2. Alur Pembersihan Data.....	129
Gambar 3. Alur Transformasi Data	155
Gambar 4. Alur Praktikum Klasifikasi	180
Gambar 5. Proses Pemodelan Klasifikasi.....	198
Gambar 6. Alur Analisis Clustering	233

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah mendorong organisasi untuk menghasilkan dan menyimpan data dalam jumlah yang semakin besar. Data tersebut dapat berasal dari transaksi bisnis, aktivitas pelanggan, layanan digital, sistem akademik, media sosial, maupun berbagai sistem informasi yang digunakan dalam kegiatan operasional organisasi. Namun, data yang tersedia tidak serta-merta memberikan manfaat apabila tidak diolah, dianalisis, dan diinterpretasikan secara tepat. Oleh karena itu, dibutuhkan metode yang mampu mengubah data menjadi informasi, pola, wawasan, dan rekomendasi yang mendukung pengambilan keputusan (Fawcett, 2013; Han, Pei and Tong, 2022).

Data mining merupakan proses penggalian pola dan pengetahuan dari sekumpulan data dengan menggunakan teknik analisis tertentu. Sementara itu, machine learning merupakan pendekatan yang memungkinkan komputer mempelajari pola dari data untuk menghasilkan prediksi, klasifikasi, atau pengelompokan berdasarkan data yang tersedia. Dalam bidang Sistem Informasi, kedua bidang tersebut memiliki peran penting karena dapat digunakan untuk mendukung analisis pelanggan, prediksi penjualan, segmentasi pasar, deteksi risiko, evaluasi layanan, dan berbagai kebutuhan pengambilan keputusan berbasis data (Géron, 2019; Han, Pei and Tong, 2022). Scikit-learn sendiri mendukung pendekatan supervised dan unsupervised learning serta menyediakan fasilitas untuk preprocessing, pemodelan, pemilihan model, dan evaluasi model (scikit-learn Developers, 2026).

Mata kuliah Data Mining dan Machine Learning dirancang untuk membekali mahasiswa dengan kemampuan memahami konsep, tahapan, metode, dan penerapan analisis data dalam penyelesaian masalah di bidang Sistem Informasi. Materi yang dipelajari mencakup pemahaman data, eksplorasi data, pembersihan data, transformasi data, seleksi fitur, pembagian dataset, klasifikasi, regresi, clustering, evaluasi model, visualisasi, dan interpretasi hasil. Mahasiswa juga diarahkan untuk menghasilkan wawasan serta rekomendasi keputusan berdasarkan hasil pemodelan data. Cakupan

tersebut secara eksplisit tercantum dalam RPS mata kuliah Data Mining dan Machine Learning..

Penguasaan konsep saja belum cukup untuk mencapai kompetensi tersebut. Mahasiswa perlu memperoleh pengalaman langsung dalam mengolah dataset, menulis kode program, membangun model, mengevaluasi hasil, dan menyusun interpretasi. Pendekatan praktis penting dalam pembelajaran machine learning karena proses pengembangan model melibatkan rangkaian aktivitas mulai dari penyiapan data, pemodelan, prediksi, evaluasi, hingga perbandingan hasil (Géron, 2019). Kegiatan praktikum menjadi sarana bagi mahasiswa untuk menghubungkan teori dengan studi kasus sehingga mereka dapat memahami keterkaitan antara permasalahan bisnis, karakteristik data, pemilihan metode, hasil pemodelan, dan rekomendasi yang dihasilkan.

Python dipilih sebagai bahasa pemrograman utama dalam modul ini karena didukung oleh ekosistem pustaka yang luas untuk komputasi numerik, pengolahan data, visualisasi, dan machine learning. NumPy menyediakan fasilitas dasar untuk komputasi ilmiah dan operasi array multidimensi, sedangkan pandas menyediakan struktur data serta perangkat analisis data yang dirancang untuk pengolahan data secara praktis. Scikit-learn menyediakan fasilitas untuk preprocessing, supervised learning, unsupervised learning, model selection, dan model evaluation. Dengan dukungan tersebut, Python dapat digunakan dalam berbagai tahapan analisis data mulai dari manipulasi data hingga pengembangan dan evaluasi model machine learning (McKinney, 2017; Géron, 2019).

Kegiatan praktikum dapat dilaksanakan melalui Jupyter Notebook atau Google Colab sehingga kode program, narasi, data, visualisasi, dan hasil analisis dapat disajikan dalam satu dokumen komputasional. Jupyter Notebook dirancang sebagai dokumen yang dapat menggabungkan kode, penjelasan berbasis teks, data, visualisasi, dan keluaran interaktif dalam satu lingkungan kerja (Project Jupyter, 2026). Karakteristik tersebut mendukung proses pembelajaran karena mahasiswa dapat mendokumentasikan tidak hanya kode yang digunakan, tetapi juga tahapan analisis dan interpretasi hasil.

Penggunaan Google Colab memberikan kemudahan bagi mahasiswa untuk menjalankan notebook melalui peramban tanpa harus melakukan instalasi dan konfigurasi lingkungan komputasi secara lokal. Google menjelaskan bahwa Colab merupakan layanan Jupyter Notebook yang dihosting, tidak membutuhkan proses penyiapan awal, dan ditujukan antara lain untuk machine learning, data science, dan

pendidikan (Google, 2026). Notebook Colab juga dapat disimpan di Google Drive, dibagikan, maupun dimuat dari GitHub. Dalam pelaksanaan mata kuliah, hasil praktikum selanjutnya dapat dikumpulkan dan didiskusikan melalui LMS Edlink sebagaimana media pembelajaran daring yang ditetapkan dalam RPS.

Modul praktikum ini disusun sebagai panduan sistematis bagi mahasiswa dalam mempelajari dan menerapkan data mining serta machine learning menggunakan Python. Materi praktikum dirancang mengikuti tahapan pembelajaran mulai dari pengenalan lingkungan pemrograman, perumusan masalah analitik, eksplorasi dan pembersihan data, transformasi data, pemodelan klasifikasi dan regresi, clustering, evaluasi model, visualisasi, hingga penyelesaian proyek akhir. Susunan tersebut diselaraskan dengan tahapan pembelajaran, Sub-CPMK, materi, dan aktivitas praktikum yang terdapat dalam RPS. Setiap kegiatan dilengkapi dengan tujuan, dasar teori, dataset, langkah kerja, kode program, tugas, pertanyaan analisis, dan rubrik penilaian agar mahasiswa dapat mengikuti proses pembelajaran secara bertahap dan terdokumentasi.

Selain kemampuan teknis, pembelajaran data mining dan machine learning juga harus memperhatikan aspek dokumentasi, integritas akademik, etika penggunaan data, privasi, bias model, dan dampak sosial. Sistem berbasis AI dan machine learning dapat menghasilkan risiko yang berkaitan dengan bias, diskriminasi, privasi, transparansi, dan akuntabilitas apabila dikembangkan atau digunakan tanpa pertimbangan yang memadai. UNESCO menempatkan privasi dan perlindungan data, keadilan dan nondiskriminasi, transparansi, akuntabilitas, serta pengawasan manusia sebagai prinsip penting dalam penggunaan AI yang bertanggung jawab (UNESCO, 2022). Oleh karena itu, model yang memiliki performa tinggi belum tentu menghasilkan keputusan yang adil atau sesuai dengan konteks permasalahan. Mahasiswa perlu dilatih untuk tidak hanya berfokus pada ketepatan implementasi algoritma, tetapi juga memahami keterbatasan data, risiko kesalahan model, potensi bias, serta konsekuensi penggunaan hasil analisis dalam pengambilan keputusan.

Aspek tanggung jawab tersebut juga menjadi bagian dari capaian mata kuliah. RPS menekankan bahwa mahasiswa harus mampu menyelesaikan proyek data mining dan machine learning secara kolaboratif, terdokumentasi, bertanggung jawab, serta memperhatikan etika, privasi, bias model, dan dampak sosial maupun lingkungan. Dengan demikian, praktikum tidak hanya diarahkan pada pencapaian keterampilan

pemrograman, tetapi juga pada pembentukan kemampuan analitis dan sikap profesional dalam penggunaan data.

Dengan adanya **Modul Praktikum Data Mining dan Machine Learning Berbasis Python**, diharapkan proses pembelajaran menjadi lebih terarah, terstruktur, dan berorientasi pada keterampilan. Modul ini diharapkan dapat membantu mahasiswa mengembangkan kemampuan dalam mengolah data, menerapkan algoritma, mengevaluasi model, memvisualisasikan hasil, serta mengomunikasikan wawasan dan rekomendasi berbasis data. Pada akhirnya, mahasiswa diharapkan mampu menyelesaikan proyek data mining dan machine learning secara kolaboratif, terdokumentasi, bertanggung jawab, dan relevan dengan kebutuhan organisasi sebagaimana kompetensi yang ditetapkan dalam RPS.

1.2 Deskripsi Mata Kuliah

Mata kuliah Data Mining dan Machine Learning merupakan mata kuliah yang membahas konsep, tahapan, metode, dan penerapan penggalian pengetahuan dari data untuk mendukung penyelesaian masalah serta pengambilan keputusan dalam bidang Sistem Informasi. Mata kuliah ini memiliki bobot 3 SKS dan ditempuh pada semester 6 Program Studi S1 Sistem Informasi.

Pembelajaran dalam mata kuliah ini mencakup proses pengolahan data secara menyeluruh, mulai dari pemahaman permasalahan, pengenalan karakteristik data, eksplorasi data, pembersihan data, transformasi data, seleksi fitur, hingga pembagian dataset untuk kebutuhan pemodelan. Mahasiswa mempelajari bagaimana data mentah dipersiapkan agar dapat digunakan secara tepat dalam proses analisis dan pengembangan model machine learning.

Pada bagian pemodelan, mahasiswa mempelajari metode supervised learning dan unsupervised learning. Metode supervised learning digunakan untuk menyelesaikan permasalahan klasifikasi dan regresi, sedangkan metode unsupervised learning digunakan untuk menemukan kelompok atau pola dalam data melalui teknik clustering dan pengenalan association rule mining. Algoritma yang dipelajari meliputi k-Nearest Neighbors, Decision Tree, Naive Bayes, regresi linear, K-Means, clustering hierarkis, serta teknik penemuan pola asosiasi.

Mahasiswa juga mempelajari evaluasi kinerja model dengan menggunakan ukuran yang sesuai. Evaluasi model klasifikasi mencakup confusion matrix, accuracy, precision,

recall, specificity, F1-score, ROC, dan AUC. Evaluasi model regresi mencakup MAE, MSE, RMSE, dan koefisien determinasi atau R^2 , sedangkan evaluasi clustering menggunakan ukuran seperti silhouette score. Selain itu, mahasiswa diperkenalkan pada cross-validation, perbandingan model, pemilihan model terbaik, dan pengantar penyetelan hyperparameter.

Kegiatan pembelajaran dilaksanakan melalui ceramah interaktif, demonstrasi, diskusi, studi kasus, praktikum, latihan terbimbing, pembelajaran berbasis kasus, dan pembelajaran berbasis proyek. Praktikum dilakukan menggunakan Python dengan dukungan Google Colab atau Jupyter Notebook. Pustaka yang digunakan meliputi pandas dan NumPy untuk pengolahan data, Matplotlib untuk visualisasi, serta scikit-learn untuk praproses, pemodelan, evaluasi, dan pemilihan model. Hasil praktikum didokumentasikan dalam bentuk notebook, laporan, visualisasi, dan proyek yang dikumpulkan melalui LMS Edlink.

Mata kuliah ini tidak hanya menekankan kemampuan menjalankan algoritma, tetapi juga kemampuan memahami konteks permasalahan, memilih metode yang sesuai, mengevaluasi hasil secara kritis, dan menginterpretasikan keluaran model menjadi wawasan serta rekomendasi keputusan. Mahasiswa diharapkan mampu menjelaskan hubungan antara masalah bisnis, karakteristik data, proses pemodelan, hasil evaluasi, dan implikasi keputusan yang dihasilkan.

Selain aspek teknis, mata kuliah ini menekankan pentingnya dokumentasi, integritas akademik, etika penggunaan data, perlindungan privasi, identifikasi bias model, serta kesadaran terhadap dampak sosial dan lingkungan dari penerapan machine learning. Mahasiswa diarahkan untuk menggunakan data dan model secara bertanggung jawab serta mampu menjelaskan keterbatasan dari analisis yang dilakukan.

Pada akhir perkuliahan, mahasiswa menyelesaikan proyek data mining dan machine learning secara kolaboratif. Proyek tersebut mencakup perumusan masalah, pemilihan dan pemahaman dataset, eksplorasi serta praproses data, penerapan algoritma, evaluasi dan perbandingan model, visualisasi hasil, interpretasi, penyusunan rekomendasi, dokumentasi, serta presentasi. Melalui rangkaian pembelajaran tersebut, mahasiswa diharapkan memiliki kemampuan untuk menghasilkan solusi berbasis data yang relevan, dapat dipertanggungjawabkan, dan mendukung pengambilan keputusan dalam organisasi.

1.3 Tujuan Penyusunan Modul

Modul Praktikum Data Mining dan Machine Learning Berbasis Python disusun untuk mendukung pelaksanaan pembelajaran praktikum yang terarah, sistematis, dan selaras dengan capaian pembelajaran mata kuliah. Modul ini menjadi panduan bagi mahasiswa dalam menerapkan konsep data mining dan machine learning melalui kegiatan praktik menggunakan Python, Google Colab, pandas, NumPy, Matplotlib, dan scikit-learn.

Secara umum, penyusunan modul ini bertujuan membantu mahasiswa memahami alur kerja analisis data mulai dari perumusan masalah, pemahaman karakteristik data, eksplorasi, pembersihan, transformasi, pemodelan, evaluasi, visualisasi, hingga interpretasi hasil. Alur tersebut sesuai dengan cakupan materi dan capaian pembelajaran dalam RPS mata kuliah Data Mining dan Machine Learning.

Secara khusus, modul ini disusun untuk:

1. memberikan panduan praktikum yang terstruktur dan mudah diikuti oleh mahasiswa;
2. membantu mahasiswa menguasai penggunaan Python sebagai alat pengolahan dan analisis data;
3. membimbing mahasiswa dalam menggunakan Google Colab atau Jupyter Notebook sebagai lingkungan praktikum;
4. melatih mahasiswa melakukan eksplorasi, pembersihan, transformasi, seleksi fitur, dan pembagian dataset;
5. membimbing mahasiswa menerapkan algoritma klasifikasi, regresi, clustering, dan association rule mining;
6. melatih mahasiswa mengevaluasi serta membandingkan kinerja model menggunakan metrik yang sesuai;
7. mengembangkan kemampuan mahasiswa dalam memvisualisasikan dan menginterpretasikan hasil pemodelan;
8. melatih mahasiswa menyusun wawasan dan rekomendasi keputusan berbasis data;
9. membiasakan mahasiswa mendokumentasikan proses analisis, kode program, hasil eksperimen, dan kesimpulan secara sistematis;

10. membimbing mahasiswa menyelesaikan proyek data mining dan machine learning secara kolaboratif dan bertanggung jawab;
11. menanamkan kesadaran mengenai etika penggunaan data, privasi, bias model, integritas akademik, dan dampak sosial penerapan machine learning;
12. mendukung proses penilaian praktikum melalui tugas, notebook, laporan, kuis, presentasi, dan proyek akhir.

Modul ini juga disusun untuk mengurangi kesenjangan antara pemahaman konseptual dan keterampilan praktis. Mahasiswa tidak hanya diarahkan untuk mengetahui cara kerja suatu algoritma, tetapi juga dilatih untuk menentukan kapan algoritma tersebut digunakan, bagaimana data harus dipersiapkan, bagaimana hasil model dievaluasi, serta bagaimana hasil analisis dikomunikasikan dalam konteks permasalahan organisasi.

Melalui penggunaan studi kasus dan dataset yang relevan, mahasiswa diharapkan mampu memahami bahwa keberhasilan proses data mining dan machine learning tidak hanya ditentukan oleh performa model. Kualitas data, ketepatan perumusan masalah, pemilihan metode, konsistensi eksperimen, kualitas dokumentasi, dan ketepatan interpretasi juga menjadi bagian penting dalam menghasilkan solusi yang dapat dipertanggungjawabkan.

Dengan demikian, modul ini diharapkan dapat menjadi sarana pembelajaran yang membantu mahasiswa mencapai CPMK, khususnya dalam menyiapkan data, menerapkan algoritma supervised dan unsupervised learning, mengevaluasi dan menginterpretasikan hasil, serta menyelesaikan proyek data mining dan machine learning secara kolaboratif, terdokumentasi, dan etis.

1.4 Sasaran Pengguna

Modul Praktikum Data Mining dan Machine Learning Berbasis Python disusun terutama untuk mahasiswa Program Studi S1 Sistem Informasi yang menempuh mata kuliah Data Mining dan Machine Learning pada semester 6. Modul ini dirancang sebagai panduan pembelajaran praktis yang membantu mahasiswa mengembangkan keterampilan pengolahan data, pemodelan, evaluasi, visualisasi, dan interpretasi hasil menggunakan Python.

Sasaran utama penggunaan modul ini meliputi:

a. Mahasiswa Program Studi S1 Sistem Informasi

Mahasiswa merupakan pengguna utama modul. Melalui modul ini, mahasiswa diharapkan dapat mengikuti kegiatan praktikum secara bertahap, mulai dari pengenalan lingkungan kerja Python hingga penyelesaian proyek data mining dan machine learning. Modul membantu mahasiswa memahami langkah kerja, menjalankan kode, mengolah dataset, membaca hasil, serta menyusun kesimpulan dan rekomendasi.

b. Dosen pengampu mata kuliah

Modul dapat digunakan dosen sebagai pedoman pelaksanaan praktikum, demonstrasi, pemberian tugas, diskusi, dan evaluasi hasil belajar. Struktur modul yang sistematis memudahkan dosen menjaga keselarasan antara kegiatan praktikum, CPMK, Sub-CPMK, materi pembelajaran, dan bentuk penilaian yang ditetapkan dalam RPS.

c. Asisten praktikum atau pendamping pembelajaran

Modul dapat membantu asisten praktikum dalam memberikan arahan teknis, menjelaskan langkah kerja, membantu menangani kesalahan kode, serta mendampingi mahasiswa selama kegiatan praktik. Adanya prosedur yang seragam juga membantu menjaga konsistensi pendampingan antarkelas atau antarkelompok.

d. Mahasiswa atau pembelajar mandiri

Selain digunakan dalam perkuliahan formal, modul ini dapat dimanfaatkan oleh pembelajar yang ingin mempelajari dasar data mining dan machine learning menggunakan Python secara mandiri. Namun, penggunaan mandiri tetap membutuhkan pemahaman dasar mengenai pemrograman, statistika, dan pengolahan data.

e. Program studi atau unit pengelola pembelajaran

Modul dapat digunakan sebagai salah satu perangkat pembelajaran untuk mendukung pelaksanaan kurikulum, dokumentasi akademik, penjaminan mutu, dan evaluasi ketercapaian pembelajaran pada mata kuliah Data Mining dan Machine Learning.

Pengguna modul diharapkan memiliki kemampuan dasar dalam mengoperasikan komputer, mengakses internet, menggunakan peramban, mengelola berkas, dan mengikuti instruksi praktikum. Kemampuan awal dalam pemrograman dan statistika akan sangat membantu, tetapi modul tetap dirancang secara bertahap agar mahasiswa dapat menyesuaikan diri dengan penggunaan Python, Google Colab, pandas, NumPy, Matplotlib, dan scikit-learn.

Dalam pelaksanaannya, modul dapat digunakan pada pembelajaran luring, daring, maupun kombinasi keduanya. Kegiatan praktik dapat dilaksanakan di laboratorium komputer atau menggunakan laptop pribadi. Notebook, dataset, laporan, dan tugas praktikum dapat didistribusikan serta dikumpulkan melalui LMS Edlink sebagaimana media pembelajaran daring yang tercantum dalam RPS.

Dengan sasaran pengguna tersebut, modul diharapkan tidak hanya menjadi kumpulan langkah teknis, tetapi juga menjadi media pembelajaran yang membantu mahasiswa belajar secara lebih mandiri, terarah, kolaboratif, dan bertanggung jawab.

1.5 Prasyarat Kompetensi

Agar dapat mengikuti kegiatan praktikum dalam modul ini dengan baik, mahasiswa diharapkan telah memiliki sejumlah kompetensi dasar yang mendukung proses pembelajaran Data Mining dan Machine Learning. Kompetensi awal tersebut diperlukan karena kegiatan praktikum tidak hanya melibatkan penggunaan perangkat lunak, tetapi juga mencakup pemahaman data, penalaran analitis, pemrograman, serta interpretasi hasil.

Pada RPS, bagian mata kuliah prasyarat belum dicantumkan secara khusus. Oleh karena itu, prasyarat dalam modul ini diposisikan sebagai kompetensi awal yang disarankan, bukan sebagai ketentuan administratif mata kuliah.

Kompetensi awal yang diharapkan meliputi beberapa aspek berikut:

a. **Kemampuan Dasar Pengoperasian Komputer**

Mahasiswa diharapkan mampu:

- 1) mengoperasikan komputer atau laptop;
- 2) menggunakan sistem operasi;
- 3) membuat, menyimpan, memindahkan, dan mengelola berkas;
- 4) menggunakan peramba web;

- 5) mengunduh dan mengunggah berkas;
- 6) menggunakan aplikasi berbasis internet;
- 7) mengelola folder untuk menyimpan dataset, notebook, dan laporan.

Kemampuan ini penting karena sebagian besar praktikum dilakukan menggunakan Google Colab, Jupyter Notebook, LMS Edlink, dan layanan penyimpanan berbasis cloud.

b. Pemahaman Dasar Pemrograman

Mahasiswa sebaiknya telah memahami konsep dasar pemrograman, seperti:

- 1) Variabel;
- 2) Tipe data;
- 3) Operator;
- 4) Percabangan;
- 5) Perulangan;
- 6) Fungsi;
- 7) Struktur data sederhana;
- 8) Input dan output;
- 9) Logika algoritma.

Pemahaman tersebut akan membantu mahasiswa mengikuti kode Python yang digunakan dalam proses eksplorasi, pembersihan, transformasi, pemodelan, dan evaluasi data. Meskipun demikian, modul ini tetap menyediakan pengantar Python pada bagian awal praktikum untuk menyegarkan kembali kemampuan dasar mahasiswa.

c. Pemahaman Dasar Statistika

Mahasiswa diharapkan memahami konsep statistika dasar, antara lain:

- 1) Populasi dan sampel;
- 2) Jenis data;
- 3) Skala pengukuran;
- 4) Nilai rata-rata;
- 5) Median;
- 6) Modus;
- 7) Nilai minimum dan maksimum;

- 8) Rentang data;
- 9) Varians;
- 10) Simpangan baku;
- 11) Distribusi data;
- 12) Korelasi;
- 13) Interpretasi tabel dan grafik.

Pengetahuan statistika diperlukan untuk memahami karakteristik dataset, melakukan exploratory data analysis, membaca distribusi data, menemukan pencilaan, serta menginterpretasikan hasil evaluasi model.

d. Pemahaman Dasar Basis Data

Mahasiswa sebaiknya memiliki pemahaman awal mengenai;

- 1) Tabel;
- 2) Baris dan kolom;
- 3) Atribut;
- 4) Record;
- 5) Tipe data;
- 6) Nilai kosong;
- 7) Kunci atau identitas data;
- 8) Relasi sederhana antara identitas;
- 9) Format data tabular.

Pemahaman ini membantu mahasiswa mengenali struktur dataset yang disimpan dalam format CSV, Excel, atau bentuk tabel lainnya.

e. Kemampuan Membaca Data dan Visualisasi

Mahasiswa diharapkan mampu membaca dan memahami:

- 1) Table data;
- 2) Diagram batang;
- 3) Diagram garis;
- 4) Histogram;
- 5) Diagram pencar;
- 6) Diagram lingkaran;

- 7) Grafik perbandingan;
- 8) Ringkasan statistik.

Kemampuan tersebut dibutuhkan untuk melakukan eksplorasi data dan menginterpretasikan hasil pemodelan yang disajikan dalam bentuk visual.

f. Kemampuan Berpikir Logis dan Analisis

Kegiatan data mining dan machine learning memerlukan kemampuan untuk:

- 1) Mengenali masalah;
- 2) Mengidentifikasi kebutuhan data;
- 3) Membedakan variable input dan target;
- 4) Memilih prosedur yang sesuai;
- 5) Membandingkan beberapa hasil;
- 6) Menarik Kesimpulan berdasarkan bukti;
- 7) Menjelaskan hubungan antara hasil dan konteks permasalahan.

Mahasiswa tidak hanya diharapkan mampu menjalankan kode, tetapi juga memahami alasan penggunaan metode dan dampaknya terhadap hasil analisis.

g. Kemampuan Dasar Menggunakan Internet dan Layanan Daring

Karena praktikum memanfaatkan Google Colab dan LMS Edlink, mahasiswa diharapkan mampu:

- 1) Menggunakan akun google;
- 2) Membuka dan meyimpan notebook di Google Drive;
- 3) Mengunggah dataset;
- 4) Membagikan tautan notebook;
- 5) Mengakses materi pada LMS Edlink;
- 6) Mengunggah tugas;
- 7) Mengikuti diskusi dan kuis daring.

Penggunaan Google Colab, Jupyter Notebook, GitHub, dan LMS Edlink telah dicantumkan dalam RPS sebagai bagian dari media pembelajaran mata kuliah.

h. Kemampuan Dokumentasi dan Penulisan Akademik

Mahasiswa diharapkan mampu:

- 1) Menulis identitas praktikum;
- 2) Menjelaskan langkah kerja;

- 3) Menuliskan sumber dataset;
- 4) Memberikan komentar pada kode;
- 5) Menyajikan hasil dalam tabel atau grafik;
- 6) Menuliskan analisis;
- 7) Membuat kesimpulan;
- 8) Mencantumkan referensi.

Kemampuan dokumentasi menjadi penting karena hasil praktikum tidak hanya dinilai berdasarkan kode yang berhasil dijalankan, tetapi juga berdasarkan keterbatasan keterbacaan notebook, ketepatan interpretasi, dan kualitas pelaporan.

i. Sikap dan Etika Akademik

Mahasiswa diharapkan memiliki sikap:

- 1) Jujur dalam mengerjakan tugas;
- 2) Bertanggung jawab terhadap hasil analisis;
- 3) Menghargai kontribusi anggota kelompok;
- 4) Mencantumkan sumber kode dan data;
- 5) Menjaga kerahasiaan data;
- 6) Tidak memanipulasi hasil;
- 7) Tidak melakukan plagiasi;
- 8) Memperhatikan aspek privasi dan bias data.

Hal tersebut sejalan dengan RPS yang menekankan dokumentasi, integritas akademik, etika penggunaan data, privasi, bias model, serta dampak social penerapan machine learning.

Apabila terdapat mahasiswa yang belum menguasai seluruh kompetensi awal tersebut, dosen dapat memberikan materi penyegaran, latihan pendahuluan, atau sumber belajar tambahan sebelum masuk ke praktikum inti. Dengan dukungan tersebut, mahasiswa diharapkan dapat mengikuti seluruh tahapan modul secara lebih efektif dan mencapai capaian pembelajaran yang telah ditetapkan.

1.6 Ruang Lingkup Modul

Modul Praktikum Data Mining dan Machine Learning Berbasis Python mencakup serangkaian kegiatan praktikum yang dirancang untuk membimbing mahasiswa

memahami dan menerapkan proses analisis data secara bertahap, mulai dari pengenalan lingkungan pemrograman hingga penyelesaian proyek berbasis data. Ruang lingkup modul disusun selaras dengan bahan kajian, capaian pembelajaran, dan aktivitas praktikum yang tercantum dalam RPS mata kuliah Data Mining dan Machine Learning.

Secara umum, ruang lingkup modul meliputi beberapa bagian berikut:

a. Pengenalan Lingkungan Praktikum Berbasis Python

Bagian awal modul membahas lingkungan kerja yang digunakan selama praktikum, khususnya Python, Google Colab, dan Jupyter Notebook. Mahasiswa diperkenalkan pada cara membuat dan menjalankan notebook, menulis kode, menambahkan narasi, mengunggah dataset, serta menyimpan hasil praktikum.

Pada bagian ini juga diperkenalkan penggunaan pustaka utama, yaitu:

- 1) NumPy untuk komputasi numerik;
- 2) Pandas untuk manipulasi dan analisis data;
- 3) Matplotlib untuk visualisasi data;
- 4) Scikit-learn untuk praproses, pemodelan, evaluasi, dan pemilihan model.

Penggunaan perangkat tersebut sesuai dengan media pembelajaran yang telah ditetapkan dalam RPS.

b. Kosep Data Mining dan Machine Learning

Modul membahas konsep dasar data mining dan machine learning sebagai landasan sebalum mahasiswa melakukan pemodelan. Cakupan materi meliputi:

- 1) Pengertian data mining;
- 2) Pengertian machine learning;
- 3) Hubungan antara artificial intelligence, machine learning, data mining, dan data science;
- 4) Tujuan analisis data;
- 5) Contoh penerapan dalam bidang Sistem Informasi;
- 6) Jenis permasalahan analitik.

Bagian ini memberikan pemahaman konseptual agar mahasiswa tidak hanya menjalankan algoritma, tetapi juga memahami konteks penggunaannya.

c. Proses Penemuan Pengetahuan dari Data

Mahasiswa diperkenalkan pada tahapan proses analisis data melalui pendekatan:

- 1) Knowledge Discovery in Database atau KDD;
- 2) Cross Industry Standard Process for Data Mining atau CRIPS-DM;
- 3) Alur kerja proyek machine learning.

Mahasiswa diarahkan untuk memahami bagaimana masalah bisnis diterjemahkan menjadi masalah analitik, bagaimana kebutuhan data ditentukan, serta bagaimana hasil analisis dikembalikan menjadi rekomendasi yang relevan bagi organisasi.

d. Pemahaman dan Eksplorasi Data

Ruang lingkup eksplorasi data mencakup:

- 1) Jenis data;
- 2) Atribut;
- 3) Skala pengukuran;
- 4) Struktur dataset;
- 5) Tipe data;
- 6) Kualitas data;
- 7) Statistika data;
- 8) Distribusi data;
- 9) Hubungan antarvariabel;
- 10) Visualisasi awal.

Mahasiswa menggunakan pandas dan Matplotlib untuk memahami karakteristik dataset sebelum melakukan proses pemodelan.

e. Pembersihan Data

Modul membahas berbagai permasalahan kualitas data yang umum ditemukan dalam dataset, antara lain:

- 1) data hilang;
- 2) data duplikat;
- 3) inkonsistensi nilai;
- 4) kesalahan format;
- 5) pencilan;
- 6) ketidaksesuaian tipe data.

Mahasiswa diarahkan untuk mengidentifikasi permasalahan tersebut dan menerapkan teknik pembersihan yang sesuai serta mendokumentasikan perubahan yang dilakukan.

f. Transformasi dan Persiapan Data

Setelah data dibersihkan, mahasiswa mempelajari proses transformasi dan persiapan dataset untuk pemodelan. Materi meliputi:

- 1) encoding data kategorikal;
- 2) normalisasi;
- 3) standardisasi;
- 4) seleksi fitur;
- 5) reduksi fitur;
- 6) pemisahan fitur dan target;
- 7) pembagian training set dan test set;
- 8) pengenalan validation set;
- 9) cross-validation.

Tahapan ini menjadi bagian penting karena kualitas data masukan akan memengaruhi kinerja model yang dibangun.

g. Supervised Learning untuk Klasifikasi

Pada bagian klasifikasi, mahasiswa mempelajari konsep supervised learning untuk memprediksi kelas atau kategori tertentu.

Algoritma yang dibahas meliputi:

- 1) k-Nearest Neighbors;
- 2) Decision Tree;
- 3) Naive Bayes.

Mahasiswa melakukan proses:

- 1) pemilihan fitur dan target;
- 2) pembagian dataset;
- 3) pelatihan model;
- 4) prediksi;
- 5) evaluasi awal;
- 6) interpretasi hasil.

Materi tersebut sesuai dengan cakupan klasifikasi pada RPS.

h. Supervised Learning untuk Regresi

Bagian regresi digunakan untuk menyelesaikan permasalahan prediksi dengan target numerik.

Materi meliputi:

- 1) konsep regresi;
- 2) regresi linear;
- 3) variabel bebas dan variabel terikat;
- 4) koefisien regresi;
- 5) nilai prediksi;
- 6) interpretasi hubungan antarvariabel;
- 7) pengantar metode regresi lainnya.

Mahasiswa dilatih untuk tidak hanya menghasilkan nilai prediksi, tetapi juga memahami makna model dalam konteks data yang dianalisis.

i. Unsupervised Learning dan Clustering

Pada bagian unsupervised learning, mahasiswa mempelajari teknik pengelompokan data tanpa menggunakan label target.

Materi meliputi:

- 1) konsep clustering;
- 2) algoritma K-Means;
- 3) centroid;
- 4) pemilihan jumlah cluster;
- 5) analisis karakteristik cluster;
- 6) clustering hierarkis;
- 7) dendrogram;
- 8) evaluasi hasil clustering.

Mahasiswa kemudian menginterpretasikan hasil pengelompokan menjadi informasi yang relevan, misalnya untuk segmentasi pelanggan atau pengelompokan karakteristik tertentu.

j. Association Rule Mining

Modul juga memperkenalkan teknik penemuan pola hubungan dalam data transaksi melalui association rule mining.

Materi mencakup:

- 1) data transaksi;
- 2) itemset;
- 3) support;
- 4) confidence;
- 5) lift;
- 6) interpretasi aturan asosiasi.

Mahasiswa diarahkan untuk memahami bagaimana pola hubungan antaritem dapat digunakan sebagai dasar penyusunan rekomendasi atau strategi bisnis.

k. Evaluasi Model Klasifikasi

Mahasiswa mempelajari berbagai metrik evaluasi klasifikasi, yaitu:

- 1) confusion matrix;
- 2) accuracy;
- 3) precision;
- 4) recall;
- 5) specificity;
- 6) F1-score;
- 7) ROC;
- 8) AUC.

Mahasiswa dilatih untuk memilih metrik yang sesuai dengan karakteristik masalah, bukan hanya menggunakan accuracy sebagai satu-satunya ukuran performa.

l. Evaluasi Model Regresi dan Clustering

Untuk regresi, modul mencakup:

- 1) Mean Absolute Error;
- 2) Mean Squared Error;
- 3) Root Mean Squared Error;
- 4) R^2 .

Untuk clustering, salah satu ukuran yang digunakan adalah:

- silhouette score.

Mahasiswa membandingkan hasil beberapa model dan menentukan metode yang lebih sesuai berdasarkan bukti hasil eksperimen.

m. Perbandingan dan Pemilihan Model

Selain membangun model, mahasiswa diarahkan untuk melakukan eksperimen dan perbandingan.

Materi meliputi:

- 1) perbandingan beberapa algoritma;
- 2) cross-validation;
- 3) konsistensi eksperimen;
- 4) pemilihan model terbaik;
- 5) pengantar hyperparameter tuning.

Fokus bagian ini adalah melatih mahasiswa membuat keputusan berdasarkan hasil evaluasi, bukan berdasarkan asumsi atau preferensi terhadap algoritma tertentu.

n. Visualisasi dan Interpretasi Hasil

Mahasiswa mempelajari cara menyajikan hasil analisis dan pemodelan melalui visualisasi yang informatif.

Ruang lingkupnya meliputi:

- 1) visualisasi distribusi;
- 2) visualisasi pola;
- 3) visualisasi prediksi;
- 4) visualisasi cluster;
- 5) visualisasi hasil evaluasi;
- 6) visualisasi fitur penting;
- 7) interpretasi keluaran model.

Hasil pemodelan kemudian diterjemahkan menjadi wawasan dan rekomendasi yang dapat dipahami oleh pihak yang tidak selalu memiliki latar belakang teknis

o. Etika, Privasi, Bias, dan Integrasi Akademik

Modul tidak hanya membahas aspek teknis, tetapi juga menempatkan etika sebagai bagian dari proses analisis.

Materi meliputi:

- 1) penggunaan data secara bertanggung jawab;
- 2) privasi data;
- 3) bias pada dataset;
- 4) bias model;
- 5) keterbatasan hasil analisis;
- 6) integritas akademik;
- 7) pencantuman sumber data dan kode;
- 8) dokumentasi eksperimen;
- 9) dampak sosial penerapan machine learning.

Aspek tersebut merupakan bagian penting dalam RPS, khususnya pada penyelesaian proyek secara bertanggung jawab dan memperhatikan dampak sosial.

p. **Proyek Data Mining dan Machine Learning**

Bagian akhir modul mencakup kegiatan proyek yang mengintegrasikan seluruh keterampilan yang telah dipelajari.

Mahasiswa melaksanakan tahapan:

- 1) menentukan permasalahan;
- 2) memilih atau memperoleh dataset;
- 3) memahami karakteristik data;
- 4) melakukan eksplorasi;
- 5) membersihkan dan mentransformasi data;
- 6) memilih metode;
- 7) membangun model;
- 8) mengevaluasi model;
- 9) membandingkan hasil;
- 10) melakukan visualisasi;
- 11) menginterpretasikan hasil;
- 12) menyusun rekomendasi;
- 13) mendokumentasikan proses;
- 14) mempresentasikan hasil.

Proyek dilaksanakan secara kolaboratif dan diarahkan pada permasalahan yang relevan dengan bidang Sistem Informasi.

BAB II

CAPAIAN PEMBELAJARAN DAN PETA PRAKTIKUM

Bab ini menjelaskan capaian pembelajaran yang menjadi dasar penyusunan seluruh kegiatan dalam Modul Praktikum Data Mining dan Machine Learning Berbasis Python. Capaian pembelajaran digunakan untuk memastikan bahwa setiap kegiatan praktikum tidak hanya berorientasi pada kemampuan menjalankan kode, tetapi juga mengembangkan kemampuan analitis, pemecahan masalah, evaluasi model, interpretasi hasil, dokumentasi, kerja sama, dan tanggung jawab dalam penggunaan data.

Penyusunan capaian pembelajaran pada modul ini mengacu pada Rencana Pembelajaran Semester mata kuliah Data Mining dan Machine Learning. Dalam RPS tersebut, mata kuliah dibebankan pada sejumlah Capaian Pembelajaran Lulusan, kemudian dijabarkan ke dalam empat Capaian Pembelajaran Mata Kuliah dan beberapa Sub-CPMK yang menjadi dasar aktivitas pembelajaran selama satu semester.

Pemetaan capaian pembelajaran dalam modul berfungsi untuk menunjukkan keterkaitan antara kompetensi yang harus dicapai mahasiswa dengan materi, kegiatan praktikum, tugas, dan proyek yang dikerjakan. Dengan demikian, setiap modul praktikum memiliki tujuan yang jelas dan dapat dievaluasi berdasarkan indikator yang telah ditetapkan.

2.1 Capaian Pembelajaran Lulus (CPL)

Capaian Pembelajaran Lulusan atau CPL merupakan kemampuan yang diharapkan dimiliki mahasiswa sebagai bagian dari profil lulusan Program Studi S1 Sistem Informasi. Pada mata kuliah Data Mining dan Machine Learning, terdapat beberapa CPL yang dibebankan dan menjadi dasar penyusunan aktivitas pembelajaran.

Berdasarkan RPS, CPL yang dibebankan pada mata kuliah ini meliputi aspek keterampilan khusus, keterampilan umum, pengetahuan, dan sikap.

a. Keterampilan Khusus (KK)

KK-3: Mampu mentransformasikan data bisnis menjadi wawasan (*insight*) strategis melalui teknik pelaporan dan visualisasi data.

Capaian ini berkaitan langsung dengan kemampuan mahasiswa dalam mengolah data menjadi informasi yang memiliki nilai bagi organisasi. Dalam kegiatan praktikum, pencapaian KK3 dikembangkan melalui aktivitas:

1. eksplorasi data;
2. analisis statistik deskriptif;
3. pemodelan data;
4. evaluasi hasil;
5. visualisasi;
6. interpretasi;
7. penyusunan wawasan;
8. penyusunan rekomendasi keputusan.

Mahasiswa tidak hanya diminta menghasilkan grafik atau nilai evaluasi model, tetapi juga menjelaskan makna hasil tersebut dalam konteks permasalahan yang sedang dianalisis.

Sebagai contoh, pada kegiatan clustering, mahasiswa tidak hanya menghasilkan kelompok data, tetapi juga perlu menjelaskan karakteristik masing-masing cluster dan kemungkinan pemanfaatannya dalam pengambilan keputusan.

b. Keterampilan Umum (KU)

KU-1: Mampu mengambil keputusan secara tepat dalam konteks penyelesaian masalah di bidang Sistem Informasi berdasarkan hasil analisis data dan informasi.

Capaian ini dikembangkan melalui kegiatan praktikum yang menuntut mahasiswa memilih pendekatan analisis berdasarkan karakteristik permasalahan dan hasil eksperimen.

Implementasinya dalam modul antara lain:

1. menentukan jenis analisis yang sesuai;
2. menentukan algoritma berdasarkan karakteristik data;
3. membandingkan beberapa model;

4. memilih metrik evaluasi;
5. menentukan model yang lebih sesuai;
6. menyusun rekomendasi berdasarkan hasil analisis.

Mahasiswa diarahkan untuk membuat keputusan berdasarkan bukti yang diperoleh dari data, bukan sekadar berdasarkan asumsi.

Sebagai contoh, ketika membandingkan beberapa model klasifikasi, mahasiswa perlu menggunakan hasil accuracy, precision, recall, F1-score, dan metrik lain yang relevan sebelum menentukan model yang dianggap paling sesuai dengan permasalahan.

KU-2: Mampu mendokumentasikan, menyimpan, mengamankan, dan menemukan kembali data untuk menjamin kesahihan dan mencegah plagiasi.

Capaian ini dikembangkan melalui kebiasaan dokumentasi selama kegiatan praktikum.

Mahasiswa diwajibkan mendokumentasikan:

1. sumber dataset;
2. proses pembersihan data;
3. transformasi yang dilakukan;
4. kode program;
5. parameter model;
6. hasil eksperimen;
7. visualisasi;
8. interpretasi;
9. referensi yang digunakan.

Notebook praktikum menjadi salah satu media utama untuk mendokumentasikan proses tersebut. Mahasiswa juga diarahkan menggunakan penamaan berkas yang sistematis, menyimpan versi pekerjaan dengan baik, serta mencantumkan sumber data maupun kode yang diadaptasi dari referensi lain. Dalam proyek kelompok, penggunaan GitHub dapat dimanfaatkan untuk membantu dokumentasi perubahan dan kontribusi setiap anggota kelompok.

c. Pengetahuan (P)

P-3: Menguasai metode analisis data, statistika bisnis, dan teknik visualisasi informasi untuk mendukung keputusan strategis.

Capaian ini menjadi salah satu landasan utama kegiatan praktikum.

Mahasiswa mempelajari dan menerapkan beberapa metode, antara lain:

1. statistika deskriptif;
2. exploratory data analysis;
3. klasifikasi;
4. regresi;
5. clustering;
6. association rule mining;
7. evaluasi model;
8. visualisasi data;
9. interpretasi model.

Penguasaan metode tersebut diharapkan membantu mahasiswa memahami hubungan antara teknik analisis dan kebutuhan pengambilan keputusan. Dalam praktikum, mahasiswa diarahkan untuk memahami alasan penggunaan metode, bukan hanya menghafalkan sintaks Python.

Sebagai contoh, mahasiswa perlu memahami kapan sebuah permasalahan lebih tepat diselesaikan menggunakan klasifikasi, regresi, atau clustering sebelum memilih algoritma yang akan digunakan.

d. Sikap

S-3: Mampu bekerja sama dan memiliki kepekaan sosial serta kepedulian terhadap masyarakat dan lingkungan dalam setiap proyek Sistem Informasi.

Capaian sikap dikembangkan terutama melalui kegiatan berbasis proyek dan kerja kelompok.

Mahasiswa diarahkan untuk:

1. bekerja sama dalam pembagian tugas;
2. bertanggung jawab terhadap kontribusi masing-masing;
3. berkomunikasi secara profesional;
4. menghargai pendapat anggota kelompok;

5. mendokumentasikan kontribusi;
6. mempertimbangkan dampak penggunaan data dan model.

Dalam proyek data mining dan machine learning, mahasiswa juga perlu memperhatikan:

1. privasi data;
2. penggunaan data secara bertanggung jawab;
3. potensi bias;
4. risiko kesalahan model;
5. dampak keputusan berbasis model;
6. konsekuensi sosial penerapan machine learning.

Hal tersebut penting karena hasil analisis data dapat memengaruhi individu, kelompok, maupun keputusan organisasi.

e. Implementasi CPL dalam Kegiatan Praktikum

Keempat aspek CPL tersebut tidak diterapkan secara terpisah. Setiap kegiatan praktikum dapat mengembangkan beberapa aspek kemampuan secara bersamaan.

Sebagai contoh, pada praktikum klasifikasi, mahasiswa melakukan proses:

1. menyiapkan dataset;
2. membangun beberapa model;
3. mengevaluasi hasil;
4. memilih model berdasarkan bukti;
5. mendokumentasikan proses;
6. menginterpretasikan hasil.

Kegiatan tersebut secara bersamaan mengembangkan kemampuan analisis data, pengambilan keputusan, dokumentasi, dan tanggung jawab terhadap hasil analisis. Hubungan umum CPL dengan aktivitas dalam modul dapat dirangkum pada Tabel 2.1.

Tabel 1. Hubungan CPL dengan Aktivitas Modul

CPL	Fokus Kompetensi	Implementasi dalam Modul Praktikum
KK3	Transformasi data menjadi insight	EDA, visualisasi, interpretasi, rekomendasi

KU1	Pengambilan keputusan berbasis data	Pemilihan metode, evaluasi, perbandingan model
KU2	Dokumentasi dan integritas data	Notebook, sumber data, dokumentasi eksperimen, GitHub
P3	Metode analisis dan visualisasi	Statistika, klasifikasi, regresi, clustering, evaluasi
S3	Kolaborasi dan tanggung jawab sosial	Proyek kelompok, etika data, privasi, bias model

Melalui kegiatan yang terintegrasi tersebut, mahasiswa diharapkan tidak hanya memiliki keterampilan teknis dalam menggunakan Python, tetapi juga mampu menghubungkan hasil analisis dengan kebutuhan organisasi, mengambil keputusan berdasarkan data, mendokumentasikan proses secara benar, serta menggunakan data dan model secara bertanggung jawab.

Selanjutnya, CPL tersebut dijabarkan lebih spesifik melalui **Capaian Pembelajaran Mata Kuliah (CPMK)** yang menjadi sasaran utama pembelajaran dalam mata kuliah Data Mining dan Machine Learning.

2.2 Capaian Pembelajaran Mata Kuliah (CPMK)

Capaian Pembelajaran Mata Kuliah (CPMK) merupakan kemampuan yang diharapkan dikuasai mahasiswa setelah menyelesaikan proses pembelajaran pada mata kuliah Data Mining dan Machine Learning. CPMK menjadi acuan dalam penyusunan materi, kegiatan praktikum, penugasan, evaluasi, dan proyek akhir dalam Modul Praktikum Data Mining dan Machine Learning Berbasis Python.

Berdasarkan RPS, mata kuliah ini memiliki empat CPMK utama yang mencakup penguasaan konsep, kemampuan menyiapkan dan memodelkan data, kemampuan mengevaluasi dan menginterpretasikan hasil, serta kemampuan menyelesaikan proyek secara kolaboratif dan bertanggung jawab.

a. CPMK-1: Mampu menjelaskan konsep, tahapan, metode, dan penerapan data mining serta machine learning dalam penyelesaian masalah Sistem Informasi.

CPMK-1 berfokus pada penguasaan konsep dasar yang menjadi landasan sebelum mahasiswa melakukan kegiatan pemodelan. Mahasiswa diharapkan memahami posisi data mining dan machine learning dalam pengolahan data serta

mampu menghubungkannya dengan permasalahan yang terdapat dalam bidang Sistem Informasi.

Dalam modul praktikum, CPMK-1 dikembangkan melalui kegiatan yang mencakup:

- 1) pengenalan konsep data mining dan machine learning;
- 2) pemahaman hubungan antara artificial intelligence, machine learning, data mining, dan data science;
- 3) pengenalan proses KDD dan CRISP-DM;
- 4) identifikasi jenis dan karakteristik data;
- 5) pengenalan kualitas data;
- 6) exploratory data analysis;
- 7) identifikasi karakteristik supervised learning dan unsupervised learning;
- 8) perumusan masalah bisnis menjadi masalah analitik.

Mahasiswa tidak hanya diharapkan mampu menjelaskan konsep secara teoritis, tetapi juga dapat mengenali bentuk penerapannya pada kasus nyata dalam organisasi.

Sebagai contoh, mahasiswa harus mampu menentukan apakah suatu permasalahan lebih tepat diselesaikan menggunakan klasifikasi, regresi, clustering, atau pendekatan lain berdasarkan karakteristik target dan tujuan analisis.

b. CPMK-2: Mampu menyiapkan data dan menerapkan algoritma supervised serta unsupervised learning untuk menemukan pola, melakukan klasifikasi, dan menghasilkan prediksi.

CPMK-2 menekankan keterampilan teknis mahasiswa dalam menyiapkan dataset dan mengimplementasikan metode machine learning.

Kemampuan yang dikembangkan meliputi:

- 1) pengumpulan dan pemahaman dataset;
- 2) pembersihan data;
- 3) penanganan missing value;
- 4) penghapusan data duplikat;
- 5) penanganan inkonsistensi dan pencilan;

- 6) transformasi data;
- 7) encoding data kategorikal;
- 8) normalisasi dan standardisasi;
- 9) seleksi fitur;
- 10) pembagian dataset;
- 11) penerapan algoritma klasifikasi;
- 12) penerapan algoritma regresi;
- 13) penerapan algoritma clustering;
- 14) pengenalan teknik penemuan pola atau association rule mining.

Pada bagian supervised learning, mahasiswa menerapkan algoritma seperti:

- 1) k-Nearest Neighbors;
- 2) Decision Tree;
- 3) Naive Bayes;
- 4) regresi linear.

Sementara itu, pada bagian unsupervised learning, mahasiswa mempelajari dan menerapkan:

- 1) K-Means;
- 2) clustering hierarkis;
- 3) association rule mining.

Kegiatan praktikum pada CPMK-2 diarahkan agar mahasiswa mampu melakukan proses analisis secara runtut. Mahasiswa tidak hanya menjalankan algoritma, tetapi juga harus mampu menjelaskan alasan penggunaan teknik praproses dan metode pemodelan tertentu.

c. CPMK-3: Mampu mengevaluasi, membandingkan, memvisualisasikan, dan menginterpretasikan hasil model untuk menghasilkan wawasan serta rekomendasi keputusan berbasis data.

Bagian ini menjelaskan perangkat lunak dan perangkat keras yang dibutuhkan untuk melaksanakan praktikum. Umumnya alat yang digunakan meliputi:

CPMK-3 berfokus pada kemampuan mahasiswa setelah model berhasil dibangun. Mahasiswa diharapkan mampu menilai kualitas model secara objektif

dan menerjemahkan hasil teknis menjadi informasi yang dapat digunakan dalam pengambilan keputusan.

Kemampuan yang dikembangkan meliputi:

- 1) menghitung dan menginterpretasikan metrik evaluasi klasifikasi;
- 2) mengevaluasi model regresi;
- 3) mengevaluasi hasil clustering;
- 4) melakukan cross-validation;
- 5) membandingkan beberapa model;
- 6) menentukan model yang lebih sesuai;
- 7) melakukan pengantar hyperparameter tuning;
- 8) membuat visualisasi hasil;
- 9) menginterpretasikan pola dan prediksi;
- 10) menyusun wawasan;
- 11) menyusun rekomendasi berbasis data.

Untuk model klasifikasi, metrik yang digunakan meliputi:

- 1) confusion matrix;
- 2) accuracy;
- 3) precision;
- 4) recall;
- 5) specificity;
- 6) F1-score;
- 7) ROC;
- 8) AUC.

Untuk model regresi, evaluasi mencakup:

- 1) Mean Absolute Error (MAE);
- 2) Mean Squared Error (MSE);
- 3) Root Mean Squared Error (RMSE);
- 4) koefisien determinasi (R^2).

Sedangkan untuk clustering, salah satu ukuran yang digunakan adalah silhouette score. Pencapaian CPMK-3 tidak berhenti pada perolehan nilai evaluasi.

Mahasiswa harus mampu menjelaskan makna dari nilai tersebut dan menghubungkannya dengan konteks masalah.

Sebagai contoh, mahasiswa tidak cukup menyatakan bahwa sebuah model memiliki accuracy tertinggi. Mahasiswa harus mempertimbangkan metrik lain yang relevan dan memberikan argumentasi mengapa model tersebut dipilih.

d. CPMK-4: Mampu menyelesaikan proyek data mining dan machine learning secara kolaboratif, terdokumentasi, bertanggung jawab, serta memperhatikan dampak sosial dan lingkungan.

CPMK-4 merupakan integrasi dari kemampuan yang dikembangkan pada CPMK sebelumnya. Mahasiswa diarahkan untuk menyelesaikan sebuah proyek data mining dan machine learning secara utuh mulai dari identifikasi masalah hingga penyampaian hasil.

Kemampuan yang dikembangkan meliputi:

- 1) merumuskan masalah;
- 2) menentukan tujuan analisis;
- 3) memilih dataset;
- 4) menentukan metode;
- 5) menyusun rencana kerja;
- 6) membagi tugas kelompok;
- 7) melakukan praproses data;
- 8) membangun dan mengevaluasi model;
- 9) menginterpretasikan hasil;
- 10) menyusun rekomendasi;
- 11) mendokumentasikan eksperimen;
- 12) mempresentasikan hasil;
- 13) mempertimbangkan etika penggunaan data;
- 14) mempertimbangkan privasi;
- 15) mengidentifikasi potensi bias model;
- 16) mempertimbangkan dampak sosial dan lingkungan.

Pada CPMK ini, mahasiswa juga dituntut untuk menunjukkan kemampuan bekerja sama secara efektif. Setiap anggota kelompok diharapkan memiliki kontribusi yang jelas dan dapat ditelusuri melalui dokumentasi pekerjaan.

Proyek akhir tidak hanya dinilai berdasarkan performa model, tetapi juga berdasarkan kualitas perumusan masalah, kualitas data, ketepatan metode, evaluasi, interpretasi, rekomendasi, dokumentasi, etika, presentasi, dan kerja sama. Hal tersebut sejalan dengan bentuk evaluasi proyek akhir yang tercantum dalam RPS.

e. Hubungan CPMK dengan Kegiatan Praktikum

Keempat CPMK disusun secara bertahap dan saling berkaitan. CPMK-1 menjadi landasan konseptual, CPMK-2 menekankan implementasi dan pemodelan, CPMK-3 menekankan evaluasi serta interpretasi, sedangkan CPMK-4 mengintegrasikan seluruh kemampuan tersebut dalam bentuk proyek.

Hubungan tersebut dapat dirangkum dalam Tabel 2.2

Tabel 2. Hubungan CPMK dan Kegiatan Praktikum

CPMK	Fokus Utama	Bentuk Aktivitas Praktikum
CPMK-1	Konsep dan pemahaman masalah	Konsep dasar, CRISP-DM, identifikasi data, EDA
CPMK-2	Penyiapan dan pemodelan data	Cleaning, transformasi, klasifikasi, regresi, clustering
CPMK-3	Evaluasi dan interpretasi	Evaluasi model, perbandingan, visualisasi, insight
CPMK-4	Integrasi dan proyek	Proposal, implementasi proyek, dokumentasi, presentasi

2.3 Sub-Capaian Pembelajaran Mata Kuliah (Sub-CPMK)

Sub-Capaian Pembelajaran Mata Kuliah (Sub-CPMK) merupakan penjabaran yang lebih spesifik dari CPMK dan digunakan sebagai dasar dalam merancang aktivitas pembelajaran serta kegiatan praktikum pada setiap tahap perkuliahan. Sub-CPMK membantu memastikan bahwa kemampuan mahasiswa berkembang secara bertahap, mulai dari pemahaman konsep, penyiapan data, penerapan algoritma, evaluasi hasil, hingga penyelesaian proyek.

Berdasarkan RPS mata kuliah Data Mining dan Machine Learning, terdapat delapan Sub-CPMK yang menjadi dasar pelaksanaan pembelajaran dan praktikum selama satu semester.

a. Sub-CPMK-1.1: Mampu menjelaskan konsep, tujuan, ruang lingkup, tahapan, dan penerapan data mining serta machine learning dalam bidang Sistem Informasi.

Sub-CPMK ini menjadi kemampuan dasar yang harus dikuasai sebelum mahasiswa masuk ke tahap analisis dan pemodelan data. Mahasiswa diarahkan untuk memahami:

- 1) konsep dasar data mining;
- 2) konsep dasar machine learning;
- 3) hubungan data mining dengan artificial intelligence, machine learning, dan data science;
- 4) tujuan penggunaan data mining dan machine learning;
- 5) tahapan KDD;
- 6) tahapan CRISP-DM;
- 7) contoh penerapan dalam organisasi;
- 8) transformasi masalah bisnis menjadi masalah analitik.

Dalam kegiatan praktikum, Sub-CPMK-1.1 dapat dicapai melalui analisis studi kasus, identifikasi masalah bisnis, penyusunan tujuan analisis, dan pemetaan tahapan CRISP-DM.

b. Sub-CPMK-1.2: Mampu menjelaskan jenis data, teknik eksplorasi data, serta karakteristik metode supervised dan unsupervised learning.

Sub-CPMK ini berfokus pada pemahaman karakteristik dataset sebelum proses pemodelan dilakukan.

Mahasiswa diharapkan mampu:

- 1) mengenali jenis data;
- 2) mengidentifikasi atribut;
- 3) memahami skala pengukuran;
- 4) memeriksa tipe data;
- 5) menilai kualitas data;
- 6) melakukan statistika deskriptif;

- 7) melakukan exploratory data analysis;
- 8) membedakan supervised learning dan unsupervised learning;
- 9) menentukan pendekatan pembelajaran yang sesuai berdasarkan karakteristik permasalahan.

Keterampilan ini dikembangkan melalui penggunaan Python dan pandas untuk membaca, memeriksa, merangkum, dan mengeksplorasi dataset.

c. Sub-CPMK-2.1: Mampu melakukan pengumpulan, pembersihan, transformasi, integrasi, seleksi fitur, dan pembagian dataset untuk proses pemodelan.

Sub-CPMK-2.1 menekankan kemampuan mahasiswa dalam menyiapkan data yang layak digunakan untuk machine learning.

Kegiatan yang dikembangkan meliputi:

- 1) mengidentifikasi missing value;
- 2) menangani data duplikat;
- 3) memperbaiki inkonsistensi;
- 4) menangani pencilan;
- 5) memperbaiki kesalahan format;
- 6) melakukan encoding data kategorikal;
- 7) melakukan normalisasi;
- 8) melakukan standardisasi;
- 9) memilih fitur;
- 10) memisahkan fitur dan target;
- 11) membagi dataset menjadi data latih dan data uji.

Mahasiswa juga diarahkan untuk mendokumentasikan setiap proses perubahan data sehingga tahapan praproses dapat ditelusuri kembali.

d. Sub-CPMK-2.2: Mampu menerapkan algoritma klasifikasi dan regresi untuk menyelesaikan permasalahan prediksi pada data bisnis.

Sub-CPMK ini berfokus pada penerapan supervised learning.

Pada bagian klasifikasi, mahasiswa mempelajari dan menerapkan:

- 1) k-Nearest Neighbors;
- 2) Decision Tree;
- 3) Naive Bayes.

Pada bagian regresi, mahasiswa mempelajari:

- 1) konsep prediksi numerik;
- 2) regresi linear;
- 3) pembacaan koefisien;
- 4) interpretasi hasil prediksi.

Mahasiswa tidak hanya menjalankan algoritma, tetapi juga diharapkan mampu menentukan fitur, target, data latih, data uji, serta menjelaskan hasil prediksi dalam konteks permasalahan.

e. Sub-CPMK-2.3: Mampu menerapkan algoritma clustering dan teknik penemuan pola untuk mengidentifikasi kelompok atau hubungan dalam data.

Sub-CPMK ini mengembangkan kemampuan mahasiswa dalam menerapkan unsupervised learning dan teknik penemuan pola.

Materi praktikum mencakup:

- 1) konsep clustering;
- 2) algoritma K-Means;
- 3) pemilihan jumlah cluster;
- 4) analisis karakteristik cluster;
- 5) clustering hierarkis;
- 6) evaluasi clustering;
- 7) association rule mining;
- 8) support;
- 9) confidence;
- 10) lift.

Mahasiswa diharapkan mampu menjelaskan arti kelompok atau pola yang dihasilkan dalam konteks data yang dianalisis.

f. Sub-CPMK-3.1: Mampu mengevaluasi dan membandingkan kinerja model menggunakan ukuran evaluasi yang sesuai.

Sub-CPMK ini menekankan kemampuan mahasiswa dalam menentukan kualitas hasil pemodelan.

Evaluasi klasifikasi mencakup:

- 1) confusion matrix;
- 2) accuracy;
- 3) precision;
- 4) recall;
- 5) specificity;
- 6) F1-score;
- 7) ROC;
- 8) AUC.

Evaluasi regresi mencakup:

- 1) MAE;
- 2) MSE;
- 3) RMSE;
- 4) R^2 .

Evaluasi clustering mencakup penggunaan silhouette score.

Mahasiswa juga diperkenalkan pada:

- 1) cross-validation;
- 2) perbandingan model;
- 3) konsistensi eksperimen;
- 4) pemilihan model terbaik;
- 5) pengantar hyperparameter tuning.

Pada tahap ini, mahasiswa diharapkan mampu membuat keputusan berdasarkan hasil evaluasi yang dapat dibuktikan.

- g. Sub-CPMK-3.2: Mampu memvisualisasikan, menginterpretasikan, dan mengomunikasikan hasil pemodelan menjadi wawasan serta rekomendasi keputusan berbasis data.**

Sub-CPMK ini mengarahkan mahasiswa untuk mengubah keluaran teknis menjadi informasi yang memiliki nilai dalam pengambilan keputusan.

Kemampuan yang dikembangkan mencakup:

- 1) memilih jenis visualisasi yang sesuai;
- 2) menampilkan hasil prediksi;
- 3) memvisualisasikan cluster;
- 4) menampilkan fitur penting;
- 5) menjelaskan hasil evaluasi;
- 6) mengidentifikasi pola penting;
- 7) menyusun insight;
- 8) merumuskan rekomendasi;
- 9) menyampaikan hasil secara lisan maupun tertulis.

Mahasiswa perlu memahami bahwa hasil pemodelan menjadi lebih bermakna apabila dapat dijelaskan dan dikomunikasikan dalam bahasa yang sesuai dengan konteks organisasi.

- h. Sub-CPMK-4.1: Mampu merancang dan menyelesaikan proyek data mining dan machine learning secara kolaboratif, terdokumentasi, bertanggung jawab, serta memperhatikan etika dan dampak sosial.**

Sub-CPMK-4.1 merupakan tahap integrasi seluruh kompetensi yang telah dipelajari.

Mahasiswa diharapkan mampu:

- 1) mengidentifikasi masalah;
- 2) menentukan tujuan proyek;
- 3) memilih dataset;
- 4) menentukan metode;
- 5) menyusun rencana kerja;

- 6) membagi tugas anggota kelompok;
- 7) melakukan eksplorasi dan praproses data;
- 8) membangun model;
- 9) mengevaluasi model;
- 10) memvisualisasikan hasil;
- 11) menginterpretasikan hasil;
- 12) menyusun rekomendasi;
- 13) mendokumentasikan seluruh eksperimen;
- 14) mempresentasikan proyek;
- 15) mempertimbangkan privasi data;
- 16) mengidentifikasi kemungkinan bias;
- 17) mempertimbangkan dampak sosial dari penggunaan model.

Dalam RPS, kemampuan ini dikembangkan pada tahap perancangan proposal, implementasi proyek, presentasi kemajuan, hingga penyelesaian dan presentasi proyek akhir.

i. Hubungan CPMK dan Sub-CPMK

Hubungan antara CPMK dan Sub-CPMK dapat dirangkum pada Tabel 2.3

Tabel 3. Hubungan CPMK dan Sub-CPMK

CPMK	Sub-CPMK	Fokus Kemampuan
CPMK-1	Sub-CPMK-1.1	Konsep, tahapan, dan penerapan data mining dan machine learning
CPMK-1	Sub-CPMK-1.2	Jenis data, EDA, supervised dan unsupervised learning
CPMK-2	Sub-CPMK-2.1	Pembersihan, transformasi, dan persiapan data
CPMK-2	Sub-CPMK-2.2	Klasifikasi dan regresi
CPMK-2	Sub-CPMK-2.3	Clustering dan penemuan pola
CPMK-3	Sub-CPMK-3.1	Evaluasi dan perbandingan model
CPMK-3	Sub-CPMK-3.2	Visualisasi, interpretasi, dan rekomendasi
CPMK-4	Sub-CPMK-4.1	Proyek, kolaborasi, dokumentasi, dan etika

Susunan Sub-CPMK tersebut menunjukkan perkembangan kemampuan mahasiswa secara berjenjang, dari pemahaman konsep menuju penerapan metode, evaluasi

hasil, interpretasi, hingga integrasi seluruh tahapan dalam bentuk proyek data mining dan machine learning.

2.4 Pemetaan CPMK dan Sub-CPMK terhadap Modul Praktikum

Pemetaan CPMK dan Sub-CPMK terhadap kegiatan praktikum diperlukan untuk memastikan bahwa setiap aktivitas dalam Modul Praktikum Data Mining dan Machine Learning Berbasis Python memiliki hubungan yang jelas dengan capaian pembelajaran mata kuliah. Pemetaan ini juga membantu menjaga kesinambungan antara materi, aktivitas mahasiswa, produk praktikum, dan bentuk evaluasi.

Berdasarkan RPS, proses pembelajaran mata kuliah Data Mining dan Machine Learning disusun secara bertahap mulai dari pemahaman konsep, eksplorasi dan praproses data, klasifikasi dan regresi, clustering dan penemuan pola, evaluasi model, visualisasi serta interpretasi hasil, hingga penyelesaian proyek.

Dalam modul ini, tahapan tersebut dikembangkan menjadi dua belas modul praktikum dan satu rangkaian proyek akhir.

a. Pemetaan Umum Modul Praktikum

Secara umum pemetaan modul praktikum di uraikan pada Tabel 2.4

Tabel 4. Pemetaan Modul Praktikum

Modul	Topik Praktikum	CPMK	Sub-CPMK	Fokus Kemampuan
1	Pengenalan Python dan Google Colab	Pendukung CPMK-2	Pendukung	Menggunakan lingkungan Python dan notebook sebagai dasar praktikum
2	CRISP-DM dan Perumusan Masalah Analitik	CPMK-1	Sub-CPMK-1.1	Memahami proses data mining dan merumuskan masalah analitik
3	Exploratory Data Analysis dengan pandas	CPMK-1	Sub-CPMK-1.2	Mengidentifikasi karakteristik dan mengeksplorasi dataset
4	Pembersihan Data	CPMK-2	Sub-CPMK-2.1	Menangani masalah kualitas data

5	Transformasi Data dan Persiapan Pemodelan	CPMK-2	Sub-CPMK-2.1	Transformasi, encoding, scaling, seleksi fitur, dan pembagian data
6	Klasifikasi	CPMK-2	Sub-CPMK-2.2	Membangun model klasifikasi
7	Regresi	CPMK-2	Sub-CPMK-2.2	Membangun model prediksi numerik
8	Clustering dengan K-Means	CPMK-2	Sub-CPMK-2.3	Mengelompokkan data dan menginterpretasikan cluster
9	Clustering Hierarkis dan Association Rule Mining	CPMK-2	Sub-CPMK-2.3	Menemukan kelompok dan pola hubungan dalam data
10	Evaluasi Model Klasifikasi	CPMK-3	Sub-CPMK-3.1	Mengukur dan menginterpretasikan performa model klasifikasi
11	Evaluasi Regresi, Clustering, dan Perbandingan Model	CPMK-3	Sub-CPMK-3.1	Membandingkan model dan menentukan model yang sesuai
12	Visualisasi, Interpretasi, dan Rekomendasi	CPMK-3	Sub-CPMK-3.2	Mengubah hasil pemodelan menjadi insight dan rekomendasi
Proyek Akhir	Proyek Data Mining dan Machine Learning	CPMK-4	Sub-CPMK-4.1	Mengintegrasikan seluruh tahapan dalam proyek kolaboratif

Modul 1 merupakan kegiatan pendukung yang ditambahkan dalam modul praktikum untuk membekali mahasiswa dengan keterampilan dasar penggunaan Python dan Google Colab. RPS telah menetapkan Python sebagai bahasa pemrograman utama serta Jupyter Notebook dan Google Colab sebagai lingkungan praktikum, tetapi tidak menetapkan pengenalan Python sebagai Sub-CPMK tersendiri.

b. Pemetaan CPMK-1

CPMK-1 menekankan kemampuan mahasiswa dalam memahami konsep, tahapan, metode, dan penerapan data mining serta machine learning.

CPMK ini terutama dicapai melalui Modul 2 dan Modul 3.

Modul 2 — CRISP-DM dan Perumusan Masalah Analitik

Modul ini mengembangkan Sub-CPMK-1.1, yaitu kemampuan menjelaskan konsep, tujuan, ruang lingkup, tahapan, dan penerapan data mining serta machine learning.

Materi praktikum meliputi:

- 1) konsep KDD;
- 2) tahapan CRISP-DM;
- 3) identifikasi masalah bisnis;
- 4) perumusan tujuan analisis;
- 5) identifikasi kebutuhan data;
- 6) transformasi masalah bisnis menjadi masalah analitik.

RPS pada pertemuan kedua menekankan kemampuan mahasiswa menguraikan tahapan KDD dan CRISP-DM serta merumuskan masalah analitik dari suatu studi kasus bisnis.

Produk praktikum yang dihasilkan dapat berupa:

- 1) lembar analisis studi kasus;
- 2) formulasi masalah bisnis;
- 3) tujuan analisis;
- 4) pemetaan tahapan CRISP-DM;
- 5) rancangan awal proses analisis.

Modul 3 — Exploratory Data Analysis dengan pandas

Modul ini mengembangkan Sub-CPMK-1.2, yaitu kemampuan menjelaskan jenis data, teknik eksplorasi data, serta karakteristik supervised dan unsupervised learning.

Mahasiswa melakukan:

- 1) pembacaan dataset;
- 2) identifikasi atribut;
- 3) pemeriksaan tipe data;
- 4) statistika deskriptif;

- 5) pemeriksaan distribusi;
- 6) visualisasi awal;
- 7) identifikasi masalah kualitas data.

Pada RPS, kegiatan tersebut ditempatkan pada pertemuan ketiga dalam bentuk demonstrasi dan praktikum Python/pandas.

c. Pemetaan CPMK-2

CPMK-2 merupakan bagian terbesar dalam kegiatan praktikum karena berhubungan langsung dengan kemampuan menyiapkan data serta menerapkan algoritma supervised dan unsupervised learning.

CPMK ini dikembangkan melalui Modul 4 hingga Modul 9.

Modul 4 — Pembersihan Data

Modul ini mengembangkan Sub-CPMK-2.1.

Mahasiswa melakukan:

- 1) identifikasi missing value;
- 2) penanganan data kosong;
- 3) identifikasi data duplikat;
- 4) penanganan inkonsistensi;
- 5) deteksi pencilan;
- 6) perbaikan format data.

Produk utama berupa dataset hasil pembersihan dan notebook dokumentasi proses. Hal tersebut sesuai dengan kegiatan pada RPS yang menilai mahasiswa berdasarkan kebenaran prosedur, kualitas dataset hasil pembersihan, serta dokumentasi kode.

Modul 5 — Transformasi Data dan Persiapan Pemodelan

Modul ini masih mengembangkan Sub-CPMK-2.1.

Materi meliputi:

- 1) encoding;
- 2) normalisasi;
- 3) standardisasi;
- 4) seleksi fitur;

- 5) reduksi fitur;
- 6) pemisahan fitur dan target;
- 7) pembagian data latih dan data uji.

Mahasiswa diharapkan menghasilkan dataset yang telah siap digunakan untuk pemodelan.

Modul 6 — Klasifikasi

Modul ini mengembangkan Sub-CPMK-2.2.

Algoritma yang digunakan sesuai RPS meliputi:

- 1) k-Nearest Neighbors;
- 2) Decision Tree;
- 3) Naive Bayes.

Mahasiswa dilatih untuk:

- 1) menentukan fitur dan target;
- 2) menyiapkan data;
- 3) membuat model;
- 4) melakukan training;
- 5) melakukan prediksi;
- 6) membaca hasil awal model.

Modul 7 — Regresi

Modul ini juga mengembangkan Sub-CPMK-2.2.

Fokus praktikum adalah:

- 1) regresi linear;
- 2) pemodelan target numerik;
- 3) membaca koefisien;
- 4) menghasilkan prediksi;
- 5) menginterpretasikan hasil prediksi.

Pada RPS, mahasiswa dinilai berdasarkan kebenaran model, interpretasi hasil, dan relevansinya terhadap masalah bisnis.

Modul 8 — Clustering dengan K-Means

Modul ini mengembangkan Sub-CPMK-2.3.

Mahasiswa mempelajari:

- 1) konsep unsupervised learning;
- 2) K-Means;
- 3) centroid;
- 4) penentuan jumlah cluster;
- 5) analisis karakteristik cluster.

Hasil praktikum diarahkan agar mahasiswa tidak berhenti pada pemberian label cluster, tetapi mampu menjelaskan karakteristik setiap kelompok yang dihasilkan.

Modul 9 — Clustering Hierarkis dan Association Rule Mining

Modul ini melanjutkan pencapaian Sub-CPMK-2.3.

Materi disesuaikan dengan RPS yang mencakup:

- 1) clustering hierarkis;
- 2) evaluasi hasil clustering;
- 3) association rule mining;
- 4) support;
- 5) confidence;
- 6) lift.

Mahasiswa diarahkan untuk menemukan dan menjelaskan pola hubungan yang relevan dalam konteks data bisnis.

d. Pemetaan CPMK-3

CPMK-3 berfokus pada kemampuan mengevaluasi, membandingkan, memvisualisasikan, serta menginterpretasikan hasil model.

CPMK ini dikembangkan melalui Modul 10, Modul 11, dan Modul 12.

Modul 10 — Evaluasi Model Klasifikasi

Modul ini mengembangkan Sub-CPMK-3.1.

Mahasiswa mempelajari:

- 1) confusion matrix;
- 2) accuracy;

- 3) precision;
- 4) recall;
- 5) specificity;
- 6) F1-score;
- 7) ROC;
- 8) AUC.

Mahasiswa dituntut tidak hanya menghitung metrik, tetapi juga memilih dan menginterpretasikan metrik yang paling relevan dengan permasalahan.

Modul 11 — Evaluasi Regresi, Clustering, dan Perbandingan Model

Modul ini juga mengembangkan Sub-CPMK-3.1.

Materinya mencakup:

- 1) MAE;
- 2) MSE;
- 3) RMSE;
- 4) R^2 ;
- 5) silhouette score;
- 6) cross-validation;
- 7) perbandingan model;
- 8) pengantar hyperparameter tuning.

Mahasiswa melakukan eksperimen menggunakan beberapa model kemudian menyusun argumentasi mengenai model yang lebih sesuai berdasarkan hasil evaluasi.

Modul 12 — Visualisasi, Interpretasi, dan Rekomendasi

Modul ini mengembangkan Sub-CPMK-3.2.

Aktivitas mahasiswa meliputi:

- 1) memilih bentuk visualisasi;
- 2) menyajikan hasil pemodelan;
- 3) mengidentifikasi temuan penting;
- 4) menjelaskan makna hasil;

- 5) menyusun insight;
- 6) membuat rekomendasi keputusan berbasis data.

Pada tahap ini, hasil teknis machine learning mulai diterjemahkan menjadi informasi yang dapat digunakan dalam konteks organisasi.

e. Pemetaan CPMK-4

CPMK-4 dikembangkan melalui rangkaian proyek akhir dan menjadi bentuk integrasi kompetensi dari seluruh modul sebelumnya.

Proyek Akhir Data Mining dan Machine Learning

Proyek mengembangkan Sub-CPMK-4.1, yaitu kemampuan merancang dan menyelesaikan proyek secara kolaboratif, terdokumentasi, bertanggung jawab, serta memperhatikan etika dan dampak sosial.

Proyek mencakup rangkaian kegiatan:

- 1) identifikasi permasalahan;
- 2) perumusan tujuan analisis;
- 3) pemilihan dataset;
- 4) eksplorasi data;
- 5) pembersihan data;
- 6) transformasi dan persiapan data;
- 7) pemilihan metode;
- 8) pembangunan model;
- 9) evaluasi dan perbandingan model;
- 10) visualisasi;
- 11) interpretasi;
- 12) penyusunan rekomendasi;
- 13) pembahasan aspek etika, privasi, dan bias;
- 14) dokumentasi;
- 15) presentasi hasil.

Dalam RPS, tahapan proyek mulai terlihat secara eksplisit pada pertemuan 14 melalui penyusunan proposal, dilanjutkan implementasi dan presentasi kemajuan

pada pertemuan 15, kemudian diakhiri dengan penyelesaian dan presentasi proyek pada pertemuan 16.

f. Matriks Keterkaitan Modul Praktikum dan CPMK

Hubungan antara masing-masing modul dengan CPMK divisualisasikan Tabel 2.5.

Tabel 5. Matriks Modul Praktikum dan CPMK

Modul Praktikum	CPMK-1	CPMK-2	CPMK-3	CPMK-4
Modul 1 – Python dan Google Colab		√*		
Modul 2 – CRISP-DM	√			
Modul 3 – EDA	√	√		
Modul 4 – Pembersihan Data		√		
Modul 5 – Transformasi Data		√		
Modul 6 – Klasifikasi		√		
Modul 7 – Regresi		√		
Modul 8 – K-Means		√		
Modul 9 – Clustering Hierarkis dan Association Rule		√		
Modul 10 – Evaluasi Klasifikasi			√	
Modul 11 – Evaluasi dan Perbandingan Model			√	
Modul 12 – Visualisasi dan Interpretasi			√	
Proyek Akhir	√	√	√	√

Keterangan: √ menunjukkan keterkaitan utama. Tanda √* pada Modul 1 menunjukkan kompetensi pendukung untuk pelaksanaan CPMK-2, bukan Sub-CPMK tersendiri dalam RPS.

Matriks tersebut menunjukkan bahwa proses pembelajaran tidak berdiri sebagai kegiatan yang terpisah. Kompetensi pada satu tahap menjadi dasar untuk tahap berikutnya. Pada proyek akhir, keempat CPMK digunakan secara terintegrasi sehingga mahasiswa harus mampu menghubungkan pemahaman masalah, pengolahan data, pemodelan, evaluasi, interpretasi, dan penyusunan rekomendasi dalam satu proses analisis yang utuh.

Dengan pemetaan ini, seluruh kegiatan praktikum dalam modul dapat diarahkan pada proses pembelajaran yang progresif dan terukur serta tetap memiliki keterkaitan langsung dengan capaian pembelajaran yang ditetapkan dalam RPS.

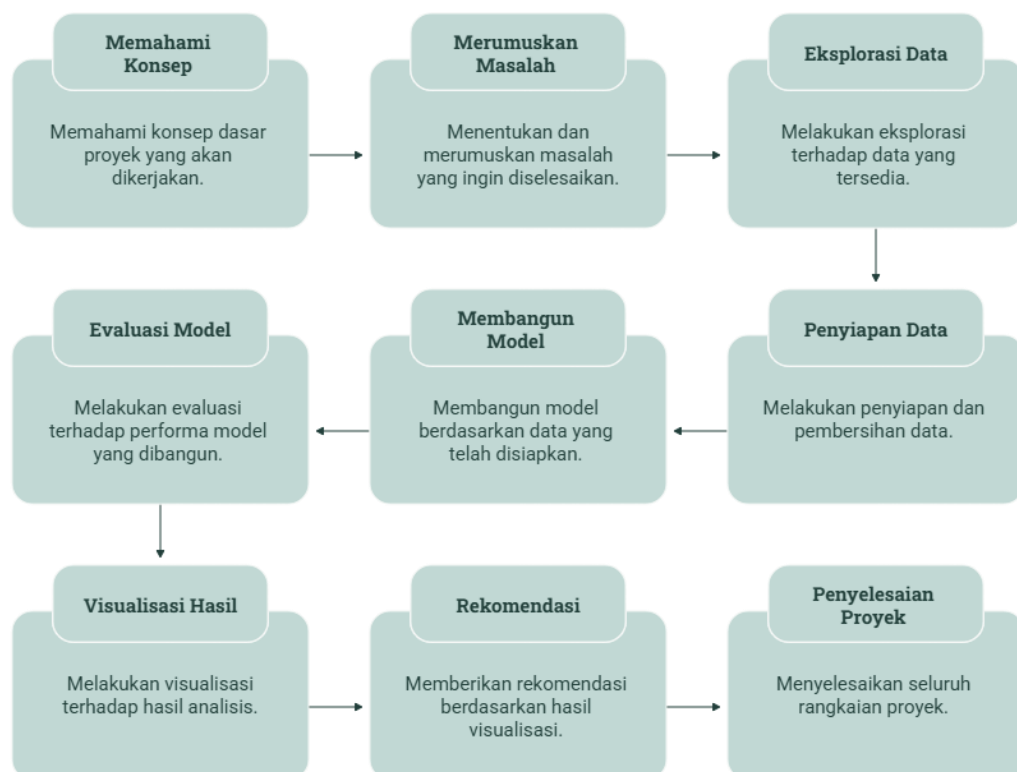
2.5 Peta Alur Pembelajaran

Peta alur pembelajaran menggambarkan urutan pengembangan kompetensi mahasiswa selama mengikuti Modul Praktikum Data Mining dan Machine Learning Berbasis Python. Alur ini disusun secara progresif agar mahasiswa tidak langsung masuk ke tahap pemodelan, tetapi terlebih dahulu memahami masalah, mengenali karakteristik data, menyiapkan data, memilih metode, mengevaluasi hasil, hingga menyusun rekomendasi dan menyelesaikan proyek.

Urutan tersebut sejalan dengan struktur pembelajaran pada RPS yang dimulai dari konsep dasar dan CRISP-DM, dilanjutkan eksplorasi dan praproses data, klasifikasi, regresi, clustering, evaluasi model, visualisasi, interpretasi hasil, dan proyek akhir.

Secara umum, alur pembelajaran dalam modul ini dapat digambarkan pada Gambar 2.1

Gambar 1. Alur Pembelajaran Modul



Tahap 1 Memahami Konsep dan Permasalahan

Tahap pertama membangun pemahaman dasar mahasiswa mengenai data mining dan machine learning sebelum melakukan kegiatan teknis.

Mahasiswa mempelajari:

- 1) konsep data mining;
- 2) konsep machine learning;
- 3) supervised dan unsupervised learning;
- 4) hubungan data mining dengan artificial intelligence dan data science;
- 5) KDD;
- 6) CRISP-DM;
- 7) penerapan data mining dalam bidang Sistem Informasi.

Pada tahap ini, mahasiswa juga mulai belajar membedakan antara masalah bisnis dan masalah analitik.

Sebagai contoh, pernyataan:

“Perusahaan mengalami peningkatan pelanggan yang berhenti menggunakan layanan”

merupakan masalah bisnis.

Masalah tersebut kemudian dapat diubah menjadi masalah analitik:

“Membangun model klasifikasi untuk memprediksi pelanggan yang berpotensi berhenti menggunakan layanan.”

Kemampuan tersebut menjadi dasar sebelum mahasiswa memilih dataset atau algoritma.

Tahap ini terutama berkaitan dengan CPMK-1 dan Sub-CPMK-1.1.

Tahap 2 Memahami dan Mengeksplorasi Data

Setelah memahami masalah, mahasiswa diarahkan untuk mengenali dataset yang akan digunakan.

Kegiatan yang dilakukan meliputi:

- 1) membaca dataset;
- 2) mengenali atribut;
- 3) memeriksa tipe data;

- 4) menentukan fitur dan target;
- 5) menghitung statistik deskriptif;
- 6) memeriksa distribusi data;
- 7) membuat visualisasi awal;
- 8) mengidentifikasi masalah kualitas data.

Tahap ini penting karena mahasiswa perlu memahami data sebelum memutuskan teknik analisis atau algoritma yang digunakan.

Pada tahap ini, mahasiswa mulai menggunakan Python dan pandas secara lebih intensif.

Tahap ini berkaitan terutama dengan Sub-CPMK-1.2.

Tahap 3 Membersihkan dan Menyiapkan Data

Dataset yang diperoleh dalam kondisi nyata sering mengandung berbagai masalah yang dapat memengaruhi hasil analisis.

Mahasiswa melakukan proses:

- 1) menangani missing value;
- 2) menghapus duplikasi;
- 3) memperbaiki inkonsistensi;
- 4) menangani pencilan;
- 5) memperbaiki format;
- 6) melakukan encoding;
- 7) melakukan normalisasi;
- 8) melakukan standardisasi;
- 9) melakukan seleksi fitur;
- 10) membagi data menjadi training set dan test set.

Tahap ini menghasilkan dataset yang lebih siap digunakan untuk pemodelan.

Mahasiswa juga perlu memahami bahwa teknik praproses tidak dapat diterapkan secara sembarangan. Setiap perubahan terhadap data harus memiliki alasan dan didokumentasikan dengan baik.

Tahap ini mendukung pencapaian **CPMK-2 dan Sub-CPMK-2.1**.

Tahap 4 Membangun Model

Setelah dataset siap, mahasiswa mulai menerapkan algoritma machine learning.

Pada supervised learning, mahasiswa melakukan:

- 1) klasifikasi menggunakan k-Nearest Neighbors;
- 2) klasifikasi menggunakan Decision Tree;
- 3) klasifikasi menggunakan Naive Bayes;
- 4) regresi menggunakan regresi linear.

Pada unsupervised learning, mahasiswa melakukan:

- 1) clustering menggunakan K-Means;
- 2) clustering hierarkis;
- 3) pengenalan association rule mining.

Kegiatan ini mencakup:

- 1) menentukan fitur dan target;
- 2) memilih algoritma;
- 3) melatih model;
- 4) menghasilkan prediksi atau cluster;
- 5) menyimpan hasil eksperimen.

Tahap ini berkaitan dengan Sub-CPMK-2.2 dan Sub-CPMK-2.3.

Tahap 5 Mengevaluasi dan Membandingkan Model

Model yang telah dibangun perlu dievaluasi untuk mengetahui kualitas dan kesesuaiannya dengan permasalahan.

Untuk klasifikasi, mahasiswa menggunakan:

- 1) confusion matrix;
- 2) accuracy;
- 3) precision;
- 4) recall;
- 5) specificity;
- 6) F1-score;
- 7) ROC;
- 8) AUC.

Untuk regresi, digunakan:

- 1) MAE;
- 2) MSE;
- 3) RMSE;
- 4) R^2 .

Untuk clustering, mahasiswa mempelajari:

- silhouette score.

Mahasiswa juga melakukan:

- 1) cross-validation;
- 2) perbandingan beberapa model;
- 3) pemilihan model;
- 4) pengantar hyperparameter tuning.

Mahasiswa diarahkan untuk menyusun keputusan berdasarkan hasil evaluasi, bukan sekadar memilih algoritma yang menghasilkan angka tertinggi pada satu metrik.

Tahap ini berhubungan langsung dengan **CPMK-3 dan Sub-CPMK-3.1**.

Tahap 6 Memvisualisasikan dan Menginterpretasikan Hasil

Hasil pemodelan kemudian disajikan dalam bentuk yang lebih mudah dipahami.

Mahasiswa melakukan:

- 1) visualisasi distribusi data;
- 2) visualisasi hasil prediksi;
- 3) visualisasi cluster;
- 4) visualisasi performa model;
- 5) penyajian fitur penting;
- 6) identifikasi pola utama.

Setelah visualisasi dibuat, mahasiswa harus mampu menjelaskan:

- 1) apa yang ditunjukkan oleh hasil;
- 2) pola apa yang ditemukan;
- 3) faktor apa yang berpengaruh;
- 4) keterbatasan model;
- 5) kemungkinan implikasi hasil.

Tahap ini mengembangkan kemampuan mahasiswa untuk mengubah hasil teknis menjadi informasi yang dapat dipahami oleh pengguna nonteknis.

Tahap ini berkaitan dengan Sub-CPMK-3.2.

Tahap 7 Menyusun Wawasan dan Rekomendasi

Interpretasi hasil harus dilanjutkan dengan penyusunan wawasan dan rekomendasi.

Mahasiswa diarahkan untuk membedakan antara:

- 1) hasil analisis, yaitu keluaran teknis;
- 2) wawasan, yaitu pemahaman penting yang diperoleh dari hasil;
- 3) rekomendasi, yaitu tindakan yang dapat dipertimbangkan berdasarkan wawasan tersebut.

Contoh:

Hasil model:

Pelanggan dengan frekuensi transaksi rendah memiliki kemungkinan churn lebih tinggi.

Wawasan:

Aktivitas transaksi yang rendah dapat menjadi indikator awal berkurangnya keterlibatan pelanggan.

Rekomendasi:

Organisasi dapat membuat program retensi atau promosi khusus untuk pelanggan dengan tingkat aktivitas yang mulai menurun.

Tahap ini menjadi bagian penting dari CPMK-3, karena tujuan akhir analisis bukan hanya menghasilkan model, tetapi mendukung keputusan berbasis data.

Tahap 8 Menyelesaikan Proyek Terintegrasi

Tahap akhir merupakan integrasi seluruh kompetensi melalui proyek data mining dan machine learning.

Mahasiswa melaksanakan:

- 1) identifikasi masalah;
- 2) perumusan tujuan;
- 3) pemilihan dataset;
- 4) eksplorasi data;

- 5) praproses;
- 6) pemodelan;
- 7) evaluasi;
- 8) visualisasi;
- 9) interpretasi;
- 10) penyusunan rekomendasi;
- 11) dokumentasi;
- 12) presentasi.

Selain aspek teknis, mahasiswa juga wajib mempertimbangkan:

- 1) etika penggunaan data;
- 2) privasi;
- 3) bias model;
- 4) keterbatasan hasil;
- 5) dampak sosial;
- 6) tanggung jawab penggunaan model.

Dalam RPS, proyek dikembangkan mulai dari penyusunan proposal pada pertemuan 14, implementasi dan demonstrasi pada pertemuan 15, hingga penyelesaian serta presentasi proyek akhir pada pertemuan 16.

Hubungan Tahapan dengan CPMK

Peta keterkaitan tahapan pembelajaran dengan CPMK dapat dirangkum pada Tabel 2.6.

Tabel 6. Hubungan Tahapan CPMK

Tahap	Aktivitas Utama	CPMK
1	Memahami konsep dan merumuskan masalah	CPMK-1
2	Memahami dan mengeksplorasi data	CPMK-1
3	Membersihkan dan menyiapkan data	CPMK-2
4	Membangun model	CPMK-2
5	Mengevaluasi dan membandingkan model	CPMK-3
6	Memvisualisasikan dan menginterpretasikan hasil	CPMK-3
7	Menyusun wawasan dan rekomendasi	CPMK-3
8	Menyelesaikan proyek terintegrasi	CPMK-4

Peta alur tersebut menunjukkan bahwa kompetensi mahasiswa dibangun secara berjenjang. Mahasiswa tidak hanya diarahkan untuk **mampu membuat model**, tetapi mengikuti satu siklus analisis yang lengkap mulai dari memahami masalah hingga menghasilkan rekomendasi yang dapat dipertanggungjawabkan.

BAB III

PETUNJUK PENGGUNAAN MODUL

Bab ini menjelaskan tata cara penggunaan Modul Praktikum Data Mining dan Machine Learning Berbasis Python agar kegiatan praktikum dapat dilaksanakan secara terarah, konsisten, dan sesuai dengan capaian pembelajaran mata kuliah. Petunjuk penggunaan mencakup peran mahasiswa, dosen atau asisten, tata tertib praktikum, ketentuan dokumentasi, integritas akademik, serta mekanisme pengumpulan tugas melalui LMS Edlink.

Modul ini dirancang untuk digunakan dalam kegiatan pembelajaran luring maupun daring dengan memanfaatkan Python, Google Colab atau Jupyter Notebook, pandas, NumPy, Matplotlib, scikit-learn, GitHub, serta LMS Edlink sebagaimana tercantum dalam RPS mata kuliah.

3.1 Petunjuk Penggunaan untuk Mahasiswa

Mahasiswa merupakan pengguna utama modul ini. Setiap kegiatan praktikum disusun secara bertahap sehingga mahasiswa perlu mengikuti urutan kegiatan yang telah ditetapkan. Praktikum tidak hanya menekankan keberhasilan menjalankan kode Python, tetapi juga pemahaman terhadap proses analisis, ketepatan pemilihan metode, interpretasi hasil, dokumentasi, dan penyusunan kesimpulan.

Sebelum mengikuti setiap kegiatan praktikum, mahasiswa perlu memahami tujuan, materi, dataset, dan hasil yang diharapkan. Setiap modul sebaiknya dikerjakan secara berurutan karena kompetensi pada satu praktikum menjadi dasar bagi praktikum berikutnya.

3.1.1. Persiapan Sebelum Praktikum

Sebelum kegiatan praktikum dimulai, mahasiswa diharapkan melakukan beberapa persiapan berikut:

- 1) membaca tujuan dan capaian pembelajaran praktikum;
- 2) mempelajari dasar teori yang terdapat pada modul;
- 3) menyiapkan laptop atau komputer yang dapat digunakan selama praktikum;

- 4) memastikan tersedia koneksi internet yang memadai;
- 5) memiliki akun Google yang aktif untuk mengakses Google Colab dan Google Drive;
- 6) memastikan dapat mengakses LMS Edlink;
- 7) mengunduh atau menyiapkan dataset yang digunakan;
- 8) membaca ketentuan tugas dan format pengumpulan;
- 9) menyiapkan folder khusus untuk menyimpan notebook, dataset, dan hasil praktikum.

RPS menetapkan laptop atau komputer dengan prosesor minimal dual-core, RAM minimal 4 GB dan disarankan 8 GB, ruang penyimpanan kosong minimal 5 GB, serta koneksi internet sebagai perangkat pendukung pembelajaran.

3.1.2. Membaca Tujuan Praktikum

Pada awal setiap modul terdapat tujuan praktikum yang menjelaskan kemampuan yang diharapkan setelah kegiatan selesai.

Mahasiswa perlu membaca bagian tersebut sebelum menjalankan kode agar memahami:

- 1) apa yang akan dipelajari;
- 2) keterampilan apa yang harus dikuasai;
- 3) hasil apa yang harus dihasilkan;
- 4) hubungan praktikum dengan CPMK dan Sub-CPMK.

Sebagai contoh, pada praktikum pembersihan data, mahasiswa tidak hanya diharapkan mampu menjalankan fungsi Python, tetapi juga mampu menentukan permasalahan kualitas data dan memilih teknik pembersihan yang sesuai.

3.1.3. Mempelajari Dasar Teori

Dasar teori pada setiap modul memberikan konsep yang diperlukan untuk memahami kegiatan praktikum. Mahasiswa disarankan membaca bagian ini sebelum mengikuti langkah kerja.

Ketika mempelajari dasar teori, mahasiswa perlu memahami:

- 1) istilah penting;
- 2) tujuan suatu metode;
- 3) karakteristik metode;

- 4) kondisi penggunaan;
- 5) kelebihan dan keterbatasan;
- 6) hubungan metode dengan permasalahan yang dianalisis.

Mahasiswa tidak disarankan hanya menyalin kode tanpa memahami konsep yang mendasarinya.

3.1.4. Menyiapkan Notebook Praktikum

Notebook digunakan sebagai dokumen utama pelaksanaan praktikum.

Setiap notebook setidaknya memuat:

1. identitas mahasiswa;
2. nomor dan judul praktikum;
3. tujuan praktikum;
4. import library;
5. pemuatan dataset;
6. tahapan pengolahan data;
7. kode program;
8. output program;
9. visualisasi;
10. interpretasi hasil;
11. jawaban pertanyaan analisis;
12. kesimpulan.

Mahasiswa dianjurkan menggunakan kombinasi **code cell** dan **text/Markdown cell** sehingga notebook tidak hanya berisi kode, tetapi juga memberikan penjelasan mengenai proses yang dilakukan.

3.1.5. Mengikuti Langkah Kerja Secara Berurutan

Langkah kerja pada setiap modul disusun untuk membentuk alur analisis yang sistematis. Oleh karena itu, mahasiswa perlu mengikuti tahapan yang diberikan secara berurutan.

Sebagai contoh, pada praktikum pemodelan, urutan kegiatan dapat meliputi: membaca dataset, menentukan fitur dan target, melakukan praproses, membagi dataset, membangun model, melakukan prediksi, dan terakhir mengevaluasi hasil.

Melompati salah satu tahap dapat menyebabkan hasil analisis tidak sesuai atau menimbulkan kesalahan pada tahapan berikutnya.

3.1.6. Menjalankan dan Memahami Kode Program

Mahasiswa perlu menjalankan setiap bagian kode secara mandiri dan memperhatikan output yang dihasilkan.

Untuk setiap kode, mahasiswa diharapkan memahami:

- 1) tujuan kode;
- 2) fungsi atau library yang digunakan;
- 3) input yang diberikan;
- 4) proses yang dilakukan;
- 5) output yang dihasilkan.

Apabila terjadi kesalahan, mahasiswa perlu membaca pesan kesalahan (error message) sebelum melakukan perbaikan.

Mahasiswa tidak dianjurkan langsung mengganti kode tanpa mengetahui penyebab kesalahan karena proses menemukan dan memperbaiki kesalahan merupakan bagian dari pembelajaran praktikum.

3.1.7. Mendokumentasikan Hasil Praktikum

Setiap tahapan analisis perlu didokumentasikan secara jelas.

Dokumentasi minimal mencakup:

- 1) sumber dataset;
- 2) kondisi awal data;
- 3) teknik pembersihan yang digunakan;
- 4) transformasi yang dilakukan;
- 5) algoritma dan parameter;
- 6) hasil model;
- 7) nilai evaluasi;
- 8) visualisasi;
- 9) interpretasi;
- 10) kesimpulan.

Dokumentasi merupakan aspek penting dalam mata kuliah karena RPS menekankan kemampuan mahasiswa dalam mendokumentasikan proses, menjaga kesahihan data, serta mencegah plagiasi.

3.1.8. Menjawab Pertanyaan Analisis

Setiap praktikum dapat dilengkapi dengan pertanyaan analisis yang bertujuan mengukur pemahaman mahasiswa terhadap hasil eksperimen.

Jawaban tidak cukup berupa:

“Model A lebih baik.”

Mahasiswa perlu memberikan alasan berdasarkan data atau hasil evaluasi, misalnya:

“Model A dipilih karena memperoleh nilai recall yang lebih tinggi. Pada kasus ini, kemampuan mendeteksi kelas positif lebih diprioritaskan dibandingkan peningkatan accuracy secara keseluruhan.”

Dengan demikian, mahasiswa dilatih menyusun argumentasi berdasarkan bukti.

3.1.9. Menulis Interpretasi Hasil

Interpretasi merupakan bagian penting dari setiap praktikum.

Mahasiswa perlu menjelaskan:

- 1) apa arti hasil yang diperoleh;
- 2) pola apa yang ditemukan;
- 3) apakah hasil sesuai dengan dugaan awal;
- 4) apa faktor yang memengaruhi hasil;
- 5) apa keterbatasan analisis;
- 6) bagaimana hasil dapat digunakan dalam konteks permasalahan.

Mahasiswa perlu menghindari hanya menuliskan ulang angka yang terdapat pada output tanpa menjelaskan maknanya.

3.1.10. Menyusun Kesimpulan Praktikum

Pada akhir praktikum, mahasiswa menyusun kesimpulan berdasarkan tujuan dan hasil kegiatan.

Kesimpulan sebaiknya menjawab:

- 1) apa yang telah dilakukan;
- 2) hasil utama yang diperoleh;
- 3) apa yang dipelajari;
- 4) apakah tujuan praktikum tercapai;
- 5) apa implikasi dari hasil tersebut.

Kesimpulan harus berasal dari hasil praktikum yang benar-benar dilakukan, bukan sekadar mengulang teori.

3.1.11. Mengerjakan Tugas Mandiri

Beberapa modul dilengkapi tugas atau eksperimen mandiri untuk mengembangkan kemampuan mahasiswa lebih lanjut.

Tugas dapat berupa:

- 1) menggunakan dataset berbeda;
- 2) mengubah parameter algoritma;
- 3) membandingkan beberapa metode;
- 4) melakukan visualisasi tambahan;
- 5) memperbaiki kualitas model;
- 6) memberikan interpretasi;
- 7) menyusun rekomendasi.

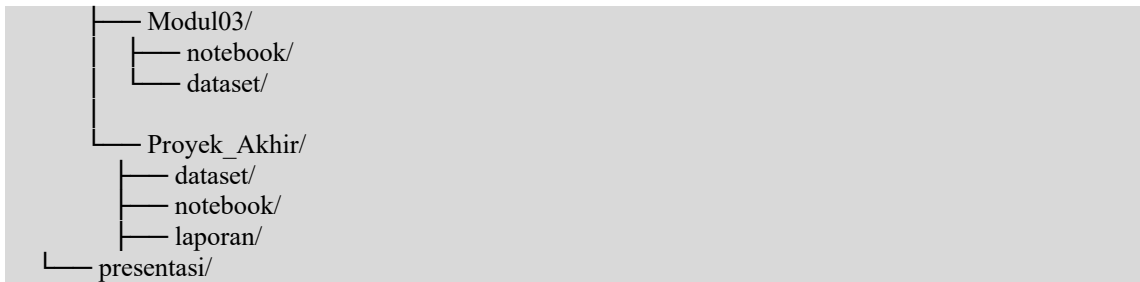
Mahasiswa diharapkan melakukan eksperimen sendiri dan mencatat perubahan serta hasil yang diperoleh.

3.1.12. Menyimpan Berkas Praktikum

Mahasiswa perlu membiasakan diri melakukan penyimpanan dan pengelolaan berkas secara sistematis.

Struktur penyimpanan yang disarankan:





Pengelolaan berkas yang baik membantu mahasiswa menemukan kembali dataset, kode, maupun hasil eksperimen ketika diperlukan.

3.1.13. Penamaan Berkas

Untuk memudahkan pengelolaan dan pemeriksaan tugas, mahasiswa harus mengikuti pola penamaan berkas yang ditentukan.

Contoh notebook individu:

NIM_Nama_Modul03_EDA.ipynb

Contoh laporan:

NIM_Nama_Modul03_EDA.pdf

Contoh tugas kelompok:

Kelompok03_Proyek_DataMining.ipynb

Penamaan yang konsisten membantu proses dokumentasi dan pengumpulan tugas

3.1.14. Pengumpulan melalui LMS Edlink

RPS menetapkan LMS Edlink sebagai media distribusi materi, pengumpulan tugas, diskusi, kuis, dan aktivitas pembelajaran daring.

Sebelum mengumpulkan tugas, mahasiswa perlu memastikan bahwa:

- 1) berkas dapat dibuka;
- 2) notebook telah dijalankan dengan benar;
- 3) output penting telah muncul;
- 4) identitas mahasiswa telah dituliskan;
- 5) jawaban analisis telah lengkap;
- 6) nama berkas telah sesuai;
- 7) tugas dikumpulkan sebelum batas waktu.

Setelah mengunggah berkas, mahasiswa perlu memeriksa kembali status pengumpulan untuk memastikan tugas telah berhasil dikirim.

3.1.15. Melakukan Refleksi Pembelajaran

Pada akhir kegiatan, mahasiswa dianjurkan melakukan refleksi singkat dengan menjawab pertanyaan seperti:

- 1) konsep apa yang paling dipahami?
- 2) bagian apa yang masih sulit?
- 3) kesalahan apa yang ditemukan selama praktikum?
- 4) bagaimana kesalahan tersebut diperbaiki?
- 5) bagaimana metode yang dipelajari dapat diterapkan pada kasus lain?

Refleksi membantu mahasiswa mengenali perkembangan kompetensinya dari satu praktikum ke praktikum berikutnya.

Dengan mengikuti petunjuk tersebut, mahasiswa diharapkan dapat menggunakan modul secara lebih mandiri dan sistematis. Praktikum tidak hanya menjadi kegiatan menjalankan kode Python, tetapi menjadi proses belajar untuk memahami masalah, mengolah data, membangun dan mengevaluasi model, menginterpretasikan hasil, serta menghasilkan rekomendasi berbasis data.

3.2 Petunjuk Penggunaan untuk Dosen atau Asisten Praktikum

Dosen dan asisten praktikum memiliki peran penting dalam memastikan penggunaan Modul Praktikum Data Mining dan Machine Learning Berbasis Python berjalan sesuai dengan capaian pembelajaran yang telah ditetapkan. Modul ini dapat digunakan sebagai panduan untuk merencanakan demonstrasi, mendampingi mahasiswa selama praktikum, memantau kemajuan, memberikan umpan balik, dan melakukan penilaian.

RPS mata kuliah Data Mining dan Machine Learning menempatkan demonstrasi, praktikum, diskusi, studi kasus, case-based learning, project-based learning, serta penggunaan Google Colab dan LMS Edlink sebagai bagian dari pelaksanaan pembelajaran. Oleh karena itu, dosen dan asisten perlu menggunakan modul ini secara fleksibel namun tetap konsisten dengan CPMK, Sub-CPMK, indikator penilaian, dan materi yang telah ditetapkan.

3.2.1. Persiapan Sebelum Praktikum

Sebelum kegiatan praktikum dilaksanakan, dosen atau asisten perlu memastikan bahwa seluruh perangkat pembelajaran telah tersedia.

Persiapan yang perlu dilakukan meliputi:

- 1) membaca tujuan dan Sub-CPMK praktikum;
- 2) memeriksa kesesuaian materi dengan RPS;
- 3) menyiapkan notebook praktikum;
- 4) menyiapkan dataset;
- 5) menguji seluruh kode sebelum digunakan mahasiswa;
- 6) memastikan library yang dibutuhkan dapat dijalankan;
- 7) menyiapkan instruksi tugas;
- 8) menyiapkan rubrik penilaian;
- 9) memastikan materi tersedia pada LMS Edlink;
- 10) menyiapkan alternatif apabila terjadi kendala teknis.

Pengujian notebook sebelum praktikum penting dilakukan agar kode, dataset, dan keluaran yang digunakan sesuai dengan tujuan pembelajaran.

3.2.2. Kendali Integral

Kendali integral akan berguna bagi sistem yang memiliki kesalahan keadaan keadaan stabil tidak nol (*error steady state*) (Hidayat, 2009). Jika sebuah plant tidak memiliki unsur integrator ($1/s$), pengontrol proposional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan stabilnya nol. Dengan pengontrol integral, respon sistem dapat diperbaiki sehingga kesalahan keadaan stabilnya nol.

Pengontrol integral memiliki karakteristik seperti halnya sebuah integral. Keluaran sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran pengontrol ini merupakan penjumlahan yang terus menerus dari perubahan masukannya.

Pengontrol integral mempunyai beberapa karakteristik berikut ini:

- a. Keluaran pengontrol membutuhkan selang waktu tertentu, sehingga pengontrol integral cenderung memperlambat respon.
- b. Ketika sinyal kesalahan berharga nol, keluaran pengontrol akan bertahan pada nilai sebelumnya.

- c. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .
- d. Konstanta integral (K_i) yang berharga besar akan mempercepat hilangnya offset. Tetapi semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi pada sistem tersebut.

3.2.3. Memberikan Pengantar Konseptual

Sebelum demonstrasi kode, dosen atau asisten perlu memberikan pengantar singkat mengenai konsep utama yang akan digunakan.

Pengantar tidak perlu menggantikan pembelajaran teori, tetapi harus membantu mahasiswa memahami:

- 1) tujuan suatu metode;
- 2) input yang diperlukan;
- 3) proses utama;
- 4) output yang dihasilkan;
- 5) kondisi penggunaannya;
- 6) keterbatasannya.

Sebagai contoh, sebelum praktikum K-Means, mahasiswa perlu memahami bahwa algoritma tersebut digunakan untuk mengelompokkan data berdasarkan kemiripan, bukan untuk melakukan klasifikasi terhadap label yang telah diketahui.

3.2.4. Melakukan Demostrasi Secara Bertahap

Demonstrasi kode sebaiknya dilakukan secara bertahap, bukan menjalankan seluruh notebook sekaligus.

Urutan demonstrasi dapat dilakukan sebagai berikut:

- 1) menjelaskan tujuan kode;
- 2) menunjukkan sintaks;
- 3) menjalankan kode;
- 4) mengamati output;
- 5) menjelaskan arti output;
- 6) memberikan kesempatan mahasiswa mencoba;
- 7) mendiskusikan kemungkinan kesalahan.

3.2.5. Menggunakan Pendekatan *Guided Practice*

Pada tahap awal, mahasiswa dapat diberikan langkah kerja yang lebih terstruktur. Setelah mahasiswa memahami pola pengerjaan, tingkat bantuan dapat dikurangi secara bertahap.

Pendekatan dapat dibagi menjadi:

- 1) demonstrasi, dosen menunjukkan proses;
- 2) guided practice, mahasiswa mengerjakan dengan panduan;
- 3) independent practice, mahasiswa menyelesaikan tugas dengan bantuan minimal;
- 4) project work, mahasiswa menentukan sebagian besar keputusan analisis secara mandiri.

Pendekatan bertahap tersebut sejalan dengan aktivitas pada RPS yang mencakup demonstrasi, guided practice, praktikum mandiri terbimbing, diskusi, dan project-based learning.

3.2.6. Mengarahkan Mahasiswa pada Proses Berpikir Analitis

Dalam kegiatan praktikum, dosen atau asisten sebaiknya tidak langsung memberikan jawaban ketika mahasiswa mengalami masalah.

Mahasiswa dapat diarahkan melalui pertanyaan seperti:

- 1) Apa tujuan tahap ini?
- 2) Bagaimana kondisi dataset sebelum diproses?
- 3) Mengapa teknik tersebut dipilih?
- 4) Apa yang ditunjukkan oleh output?
- 5) Apakah hasilnya masuk akal?
- 6) Metrik mana yang lebih relevan?
- 7) Apakah ada kemungkinan data leakage?
- 8) Apa yang akan terjadi jika parameter diubah?

Pertanyaan semacam ini membantu mahasiswa mengembangkan kemampuan analitis dan pemecahan masalah.

3.2.7. Membimbing Penanganan Kesalahan Program

Kesalahan kode merupakan bagian alami dari pembelajaran pemrograman.

Dosen atau asisten perlu membimbing mahasiswa untuk:

- 1) membaca pesan error;
- 2) menemukan baris yang menyebabkan masalah;
- 3) memeriksa nama variabel;
- 4) memeriksa nama kolom;
- 5) memeriksa tipe data;
- 6) memeriksa lokasi dataset;
- 7) memeriksa library;
- 8) melakukan perbaikan;
- 9) menjalankan kembali kode.

Mahasiswa sebaiknya tidak dibiasakan menerima kode yang telah diperbaiki tanpa memahami penyebab kesalahan.

3.2.8. Memantau Proses Praktikum

Selama praktikum berlangsung, dosen atau asisten perlu memantau proses kerja mahasiswa, bukan hanya hasil akhir.

Hal yang perlu diamati antara lain:

- 1) kemampuan mengikuti prosedur;
- 2) pemahaman kode;
- 3) kemampuan membaca output;
- 4) ketepatan pengolahan data;
- 5) kemampuan menyelesaikan error;
- 6) kualitas diskusi;
- 7) dokumentasi;
- 8) keterlibatan dalam kelompok.

Pemantauan proses penting karena indikator pada RPS tidak hanya menilai hasil, tetapi juga kebenaran prosedur, dokumentasi kode, interpretasi, dan kualitas analisis.

3.2.9. Memberikan Umpan Balik

Umpan balik sebaiknya diberikan secara spesifik dan berorientasi pada perbaikan.

Umpan balik dapat mencakup:

- 1) kesalahan pada prosedur;
- 2) ketidaktepatan pemilihan metode;
- 3) dokumentasi yang belum lengkap;
- 4) interpretasi yang terlalu umum;
- 5) visualisasi yang kurang tepat;
- 6) kesimpulan yang belum didukung hasil;
- 7) kode yang sulit dibaca.

Dosen atau asisten perlu membedakan antara kesalahan sintaks, kesalahan prosedur, dan kesalahan interpretasi karena masing-masing membutuhkan pendekatan perbaikan yang berbeda.

3.2.10. Menggunakan Dataset Secara Terencana

Dataset yang digunakan perlu dipilih sesuai dengan tingkat kesulitan dan tujuan praktikum.

Untuk praktikum awal, disarankan menggunakan dataset yang:

- 1) tidak terlalu besar;
- 2) memiliki struktur sederhana;
- 3) jumlah atribut terbatas;
- 4) mudah dipahami;
- 5) memiliki permasalahan yang jelas.

Pada tahap lanjut, dataset dapat dibuat lebih kompleks dengan:

- 1) missing value;
- 2) data kategorikal;
- 3) pencilan;
- 4) distribusi tidak seimbang;
- 5) jumlah fitur lebih banyak.

Untuk proyek akhir, mahasiswa dapat diberikan kebebasan lebih besar dalam memilih dataset dengan persetujuan dosen.

3.2.11. Memastikan Reprodusibilitas Eksperimen

Dosen dan asisten perlu membiasakan mahasiswa melakukan eksperimen yang dapat diulang.

Hal yang dapat dilakukan antara lain:

- 1) menggunakan `random_state` ketika diperlukan;
- 2) mencatat parameter model;
- 3) mendokumentasikan versi dataset;
- 4) menyimpan tahapan preprocessing;
- 5) mencatat hasil eksperimen;
- 6) tidak menghapus keluaran penting.

Reprodusibilitas menjadi bagian penting dalam dokumentasi dan penjaminan kesahihan hasil analisis.

3.2.12. Mengarahkan Dokumentasi Notebook

Notebook mahasiswa sebaiknya tidak hanya berisi kode dan output.

Dosen atau asisten perlu memastikan mahasiswa menambahkan:

- 1) penjelasan tujuan;
- 2) deskripsi dataset;
- 3) komentar kode;
- 4) alasan pemilihan metode;
- 5) interpretasi output;
- 6) pembahasan;
- 7) kesimpulan.

Notebook yang baik harus dapat dibaca kembali dan dipahami tanpa harus memperoleh penjelasan langsung dari penulisnya.

3.2.13. Mengelola Praktikum Kelompok

Pada praktikum atau proyek yang dikerjakan secara kelompok, dosen perlu memastikan adanya pembagian tugas yang jelas.

Pembagian tugas dapat mencakup:

- 1) pengumpulan data;
- 2) eksplorasi;

- 3) preprocessing;
- 4) pemodelan;
- 5) evaluasi;
- 6) visualisasi;
- 7) dokumentasi;
- 8) presentasi.

Namun, setiap anggota kelompok tetap harus memahami keseluruhan proyek.

Untuk mengurangi ketimpangan kontribusi, dosen dapat menggunakan:

- 1) log aktivitas;
- 2) catatan kontribusi;
- 3) GitHub;
- 4) penilaian teman sebaya;
- 5) tanya jawab individu;
- 6) pembagian bagian presentasi.

3.2.14. Menekankan Etika Penggunaan Data

Dosen atau asisten perlu mengingatkan mahasiswa bahwa penggunaan dataset memiliki konsekuensi etis.

Mahasiswa perlu diarahkan untuk:

- 1) menggunakan data yang diperoleh secara sah;
- 2) mencantumkan sumber;
- 3) tidak mengekspos data pribadi;
- 4) melakukan anonimisasi bila diperlukan;
- 5) mengenali potensi bias;
- 6) tidak memanipulasi hasil;
- 7) menjelaskan keterbatasan analisis.

RPS secara eksplisit menempatkan etika data, privasi, bias model, dan dampak sosial sebagai bagian dari proses pembelajaran dan proyek.

3.2.15. Melakukan Penilaian Berdasarkan Rubrik

Penilaian praktikum perlu dilakukan berdasarkan rubrik yang telah ditetapkan.

Aspek yang dapat digunakan meliputi:

- 1) ketepatan prosedur;

- 2) kualitas kode;
- 3) ketepatan hasil;
- 4) kualitas analisis;
- 5) interpretasi;
- 6) dokumentasi;
- 7) kesimpulan;
- 8) kerapian;
- 9) ketepatan pengumpulan.

Penggunaan rubrik membantu menjaga konsistensi penilaian antarpraktikum dan antarkelompok.

3.2.16. Mengelola Aktivitas melalui LMS Edlink

LMS Edlink dapat digunakan untuk:

- 1) mendistribusikan modul;
- 2) membagikan dataset;
- 3) menyediakan notebook awal;
- 4) memberikan pengumuman;
- 5) menyelenggarakan kuis;
- 6) membuka forum diskusi;
- 7) menerima tugas;
- 8) memberikan umpan balik.

Penggunaan LMS Edlink dalam modul ini mengikuti media pembelajaran daring yang tercantum dalam RPS.

3.2.17. Melakukan Evaluasi dan Perbaikan Praktikum

Setelah kegiatan selesai, dosen atau asisten perlu melakukan evaluasi terhadap pelaksanaan praktikum.

Hal yang dapat dievaluasi meliputi:

- 1) apakah tujuan tercapai;
- 2) apakah durasi mencukupi;
- 3) bagian mana yang paling sulit;
- 4) error apa yang paling sering muncul;

- 5) apakah dataset sesuai;
- 6) apakah instruksi cukup jelas;
- 7) apakah tugas terlalu mudah atau terlalu sulit;
- 8) apakah rubrik sudah sesuai.

Hasil evaluasi dapat digunakan untuk memperbaiki pelaksanaan praktikum pada semester berikutnya.

3.3 Tata Tertib Praktikum

Tata tertib praktikum disusun untuk mendukung proses pembelajaran yang tertib, aman, dan bertanggung jawab. Mahasiswa wajib mengikuti kegiatan praktikum sesuai jadwal, hadir tepat waktu, melakukan presensi, serta mengikuti kegiatan dari awal sampai selesai. Kehadiran dan partisipasi merupakan bagian dari aktivitas pembelajaran yang juga tercantum dalam RPS.

Sebelum praktikum dimulai, mahasiswa perlu menyiapkan perangkat yang diperlukan, seperti laptop atau komputer, koneksi internet, akun Google, akses ke Google Colab atau Jupyter Notebook, dataset, serta LMS Edlink. Selama praktikum, mahasiswa harus mengikuti petunjuk modul, menjalankan kode secara mandiri, memahami output yang dihasilkan, dan mendokumentasikan setiap tahapan analisis.

Dataset dan kode program harus digunakan secara bertanggung jawab. Sumber data maupun kode yang diadaptasi dari referensi lain wajib dicantumkan. Mahasiswa tidak diperkenankan memanipulasi data atau hasil eksperimen, menyalin pekerjaan mahasiswa lain, maupun mengumpulkan pekerjaan yang tidak dipahami. Pada tugas kelompok, setiap anggota harus memiliki kontribusi yang jelas dan memahami keseluruhan proses pekerjaan.

Penggunaan AI generatif mengikuti ketentuan dosen pengampu. Jika diperbolehkan, AI digunakan sebagai alat bantu belajar, bukan untuk menggantikan proses berpikir mahasiswa. Setiap kode, analisis, dan hasil yang diperoleh tetap harus diverifikasi dan dapat dipertanggungjawabkan.

Notebook atau laporan praktikum harus memuat proses kerja, kode, output, visualisasi, interpretasi, dan kesimpulan secara sistematis. Tugas dikumpulkan melalui LMS Edlink sesuai format dan batas waktu yang ditentukan. Sebelum mengumpulkan, mahasiswa perlu memastikan berkas dapat dibuka, identitas telah tercantum, dan seluruh bagian yang diwajibkan telah lengkap.

Dalam seluruh kegiatan praktikum, mahasiswa juga wajib memperhatikan integritas akademik, privasi data, potensi bias, keterbatasan model, serta dampak penggunaan hasil analisis. Hal tersebut sejalan dengan RPS yang menekankan dokumentasi, etika penggunaan data, privasi, bias model, dan tanggung jawab sosial dalam penerapan machine learning.

3.4 Integritas Akademik dan Etika Penggunaan AI

Integritas akademik merupakan prinsip penting dalam pelaksanaan praktikum Data Mining dan Machine Learning. Setiap mahasiswa bertanggung jawab memastikan bahwa kode, analisis, laporan, visualisasi, dan kesimpulan yang dikumpulkan merupakan hasil pekerjaan yang dapat dipertanggungjawabkan. RPS juga menekankan kemampuan mahasiswa dalam mendokumentasikan proses, menjaga kesahihan data, mencegah plagiasi, serta memperhatikan etika penggunaan data dan dampak sosial penerapan machine learning.

Mahasiswa wajib mencantumkan sumber dataset, referensi, serta kode yang diadaptasi dari buku, dokumentasi, repositori, atau sumber lain. Menyalin notebook, laporan, atau hasil pekerjaan mahasiswa lain tanpa atribusi tidak diperkenankan. Manipulasi data, pemalsuan output, atau penyajian hasil eksperimen yang tidak benar juga termasuk pelanggaran terhadap integritas akademik.

Penggunaan AI generatif dalam praktikum mengikuti kebijakan dosen pengampu. Apabila diperbolehkan, AI dapat digunakan sebagai alat bantu untuk memahami konsep, menjelaskan pesan kesalahan, memberikan contoh kode, atau membantu mengeksplorasi alternatif penyelesaian. Namun, AI tidak boleh digunakan untuk menggantikan seluruh proses analisis dan pemikiran mahasiswa.

Setiap keluaran yang diperoleh dari AI harus diperiksa kembali. Mahasiswa tetap wajib memahami kode yang digunakan, menguji program secara mandiri, memverifikasi hasil, serta memastikan bahwa interpretasi dan kesimpulan sesuai dengan data yang dianalisis. Keluaran AI yang tidak diverifikasi berpotensi mengandung kesalahan, ketidaksesuaian konteks, atau penjelasan yang tidak tepat.

Apabila AI digunakan dalam penyusunan tugas atau proyek, penggunaannya harus dinyatakan secara transparan sesuai ketentuan dosen. Mahasiswa tetap bertanggung jawab penuh terhadap kualitas dan kebenaran pekerjaan yang dikumpulkan. Dalam

konteks data mining dan machine learning, etika penggunaan AI juga mencakup perhatian terhadap privasi data, bias, keterbatasan model, dan dampak hasil analisis. Mahasiswa perlu menghindari penggunaan data pribadi atau sensitif tanpa izin serta tidak memasukkan data rahasia ke layanan AI yang tidak ditujukan untuk penyimpanan data tersebut.

Penggunaan AI dalam praktikum diarahkan untuk mendukung proses belajar secara kritis, bukan menggantikannya. Prinsip utama yang harus dijaga adalah transparansi, verifikasi, atribusi, tanggung jawab, dan kejujuran akademik.

3.5 Pengumpulan Tugas melalui LMS Edlink

. Pengumpulan tugas praktikum dilakukan melalui LMS Edlink sesuai dengan jadwal, jenis tugas, dan ketentuan yang ditetapkan oleh dosen pengampu. Penggunaan LMS Edlink dimaksudkan untuk memudahkan distribusi materi, pengumpulan tugas, pemberian umpan balik, kuis, diskusi, serta dokumentasi aktivitas pembelajaran. Hal ini sejalan dengan media pembelajaran daring yang tercantum dalam RPS mata kuliah Data Mining dan Machine Learning.

Setiap mahasiswa bertanggung jawab memastikan bahwa tugas yang dikumpulkan merupakan versi final dan telah memenuhi seluruh komponen yang dipersyaratkan. Notebook praktikum harus dapat dibuka dan dijalankan dengan baik, memuat identitas mahasiswa, kode program, output, visualisasi, interpretasi hasil, serta kesimpulan sesuai instruksi pada masing-masing kegiatan praktikum.

Berkas tugas diberi nama secara konsisten agar mudah diidentifikasi. Format penamaan dapat menggunakan pola:

```
NIM_Nama_ModulXX_JudulPraktikum.ipynb
```

Apabila laporan dalam format PDF juga diminta, penamaannya mengikuti pola yang sama. Untuk tugas kelompok, nama berkas dapat menggunakan nomor atau nama kelompok sesuai ketentuan dosen.

Sebelum melakukan pengumpulan, mahasiswa perlu memeriksa kembali kelengkapan tugas, memastikan seluruh cell penting pada notebook telah dijalankan, serta memastikan tidak terdapat kesalahan yang menyebabkan hasil utama tidak muncul. Sumber dataset, referensi, dan kode yang diadaptasi dari sumber lain juga harus dicantumkan dengan benar.

Tugas harus diunggah melalui ruang pengumpulan yang telah disediakan pada LMS Edlink sebelum batas waktu yang ditentukan. Setelah proses unggah selesai, mahasiswa disarankan memeriksa kembali status pengumpulan untuk memastikan berkas telah tersimpan dengan benar. Kesalahan memilih berkas atau mengunggah versi yang belum selesai menjadi tanggung jawab mahasiswa.

Ketentuan mengenai keterlambatan, pengumpulan susulan, revisi, atau perbaikan tugas mengikuti kebijakan dosen pengampu. Apabila kesempatan revisi diberikan, mahasiswa diharapkan menggunakan umpan balik yang diterima untuk memperbaiki kode, prosedur analisis, interpretasi, dokumentasi, maupun kesimpulan.

LMS Edlink tidak hanya berfungsi sebagai sarana pengumpulan berkas, tetapi juga menjadi bagian dari dokumentasi proses pembelajaran praktikum yang terstruktur, tertib, dan dapat ditelusuri kembali.

BAB IV

PERSIAPAN LINGKUNGAN PRAKTIKUM

Persiapan lingkungan praktikum merupakan tahap awal yang perlu dilakukan sebelum mahasiswa melaksanakan kegiatan Data Mining dan Machine Learning berbasis Python. Lingkungan praktikum yang terstandar akan membantu mahasiswa menjalankan kode secara konsisten, mengelola dataset dengan baik, serta mendokumentasikan proses analisis dalam bentuk notebook komputasional.

Berdasarkan RPS, perangkat lunak yang digunakan dalam mata kuliah ini meliputi Python, Jupyter Notebook, Google Colab, pandas, NumPy, Matplotlib, scikit-learn, Microsoft Excel atau LibreOffice Calc, GitHub, serta LMS Edlink. Sementara itu, perangkat keras yang disarankan berupa laptop atau komputer dengan prosesor minimal dual-core, RAM minimal 4 GB dan disarankan 8 GB, ruang penyimpanan kosong minimal 5 GB, serta koneksi internet.

Bab ini menjelaskan perangkat yang digunakan, cara menyiapkan Google Colab, struktur notebook praktikum, pengelolaan dataset, penggunaan library utama, serta penanganan beberapa kesalahan dasar yang umum terjadi selama praktikum.

4.1 Kebutuhan Perangkat Keras

Kegiatan praktikum dapat dilakukan menggunakan laptop atau komputer pribadi maupun fasilitas laboratorium komputer. Perangkat yang digunakan tidak harus memiliki spesifikasi tinggi karena sebagian besar kegiatan dapat dijalankan melalui Google Colab. Spesifikasi perangkat yang disarankan mengikuti RPS, yaitu prosesor minimal dual-core, RAM minimal 4 GB dan disarankan 8 GB, ruang penyimpanan kosong minimal 5 GB, serta koneksi internet yang stabil.

Untuk dataset berukuran kecil hingga menengah, spesifikasi tersebut pada umumnya cukup digunakan untuk kegiatan eksplorasi data, preprocessing, klasifikasi, regresi, clustering, dan evaluasi model. Jika diperlukan pemrosesan dataset yang lebih besar, mahasiswa dapat menggunakan layanan komputasi yang tersedia di laboratorium.

4.2 Perangkat Lunak dan Platform

Praktikum menggunakan beberapa perangkat lunak dan platform yang saling mendukung. Python digunakan sebagai bahasa pemrograman utama untuk pengolahan

data dan implementasi algoritma machine learning. Jupyter Notebook dan Google Colab digunakan sebagai lingkungan komputasional untuk menulis kode, narasi, visualisasi, serta hasil analisis dalam satu dokumen.

pandas dan NumPy digunakan untuk manipulasi, transformasi, dan analisis data. Matplotlib digunakan untuk membuat visualisasi, sedangkan scikit-learn digunakan untuk preprocessing, klasifikasi, regresi, clustering, evaluasi, dan pemilihan model. GitHub dapat digunakan untuk penyimpanan dan pengelolaan versi berkas proyek, sedangkan LMS Edlink digunakan untuk distribusi materi, pengumpulan tugas, kuis, diskusi, dan aktivitas pembelajaran daring.

Microsoft Excel atau LibreOffice Calc dapat digunakan untuk pemeriksaan awal dataset, terutama untuk melihat struktur tabel, nama atribut, format data, dan kemungkinan masalah sederhana sebelum data diproses menggunakan Python.

4.3 Pengenalan Google Colab

Google Colab digunakan sebagai lingkungan praktikum utama karena dapat dijalankan langsung melalui peramban tanpa memerlukan instalasi Python secara lokal. Mahasiswa cukup menggunakan akun Google untuk membuat, membuka, menjalankan, dan menyimpan notebook. Dalam Google Colab, notebook terdiri atas dua komponen utama, yaitu code cell dan text cell. Code cell digunakan untuk menulis dan menjalankan kode Python, sedangkan text cell digunakan untuk memberikan penjelasan, judul, interpretasi, atau catatan menggunakan format Markdown.

Pada awal penggunaan, mahasiswa perlu memahami beberapa aktivitas dasar, seperti membuat notebook baru, mengganti nama notebook, menjalankan cell, menambah cell, menghapus cell, menyimpan notebook, mengunduh notebook, serta membuka kembali notebook dari Google Drive.

4.4 Struktur Notebook Praktikum

Setiap notebook praktikum sebaiknya memiliki struktur yang konsisten agar mudah dibaca, diperiksa, dan digunakan kembali. Struktur dasar yang digunakan dalam modul ini meliputi:

1. judul praktikum;
2. identitas mahasiswa;
3. tujuan praktikum;

4. deskripsi dataset;
5. import library;
6. pemuatan data;
7. tahapan analisis atau pemodelan;
8. visualisasi;
9. evaluasi;
10. interpretasi;
11. kesimpulan.

Contoh struktur notebook:

```
# MODUL 03 - EXPLORATORY DATA ANALYSIS

Nama :
NIM :
Kelas :

## Tujuan Praktikum

## Deskripsi Dataset

## Import Library

## Membaca Dataset

## Eksplorasi Data

## Visualisasi

## Analisis Hasil

## Kesimpulan
```

Struktur tersebut membantu mahasiswa mendokumentasikan proses secara sistematis dan selaras dengan aspek dokumentasi yang ditekankan dalam RPS.

4.5 Mengimpor Library

Sebelum melakukan pengolahan data, library yang diperlukan perlu diimpor ke dalam notebook. Contoh library dasar yang digunakan adalah:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Pada tahap machine learning, library dari scikit-learn akan ditambahkan sesuai kebutuhan. Sebagai contoh:

```
from sklearn.model_selection import train_test_split
```

Library tidak harus seluruhnya diimpor pada awal notebook. Sebaiknya hanya library yang diperlukan dalam praktikum tersebut yang digunakan agar kode lebih sederhana dan mudah dipahami.

4.6 Membaca Dataset

Dataset yang digunakan dalam praktikum umumnya berbentuk CSV atau Excel. pandas menyediakan fungsi untuk membaca data tersebut ke dalam struktur DataFrame.

Contoh membaca file CSV:

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

Untuk menampilkan lima baris pertama:

```
df.head()
```

Dataset juga dapat dibaca dari file Excel menggunakan:

```
df = pd.read_excel("data.xlsx")
```

Setelah data berhasil dibaca, mahasiswa perlu memeriksa struktur data sebelum melakukan analisis lebih lanjut.

4.7 Mengunggah dan Mengelola Dataset

Dataset dapat dimasukkan ke Google Colab melalui beberapa cara, seperti mengunggah file secara langsung atau mengakses file yang tersimpan di Google Drive.

Untuk praktikum sederhana, file dapat diunggah langsung ke sesi Colab. Namun, file yang diunggah secara langsung dapat hilang ketika sesi berakhir. Oleh karena itu, penyimpanan melalui Google Drive lebih disarankan jika dataset akan digunakan kembali.

Mahasiswa juga dianjurkan memisahkan dataset asli dan dataset hasil pengolahan. Dataset asli sebaiknya tidak ditimpa agar proses analisis dapat dilacak dan diulang kembali jika diperlukan.

Struktur penyimpanan dapat dibuat seperti berikut:

```
Praktikum_DataMining/
├─ dataset_raw/
├─ dataset_processed/
├─ notebook/
└─ hasil/
```

4.8 Pemeriksaan Awal Dataset

Setelah dataset berhasil dibaca, mahasiswa perlu melakukan pemeriksaan awal sebelum masuk ke tahap analisis.

Beberapa fungsi dasar yang sering digunakan antara lain:

```
df.head()
df.tail()
df.shape
df.info()
df.describe()
```

Fungsi tersebut digunakan untuk melihat isi awal data, jumlah baris dan kolom, tipe data, informasi nilai kosong, serta statistik deskriptif. Pemeriksaan awal membantu mahasiswa memahami karakteristik dataset dan mendeteksi kemungkinan masalah yang perlu ditangani pada tahap preprocessing.

4.9 Mengenal pandas dan NumPy

pandas digunakan sebagai library utama untuk pengelolaan data tabular. Struktur data yang paling sering digunakan adalah DataFrame, yang menyerupai tabel dengan baris dan kolom.

Contoh memilih satu kolom:

```
df["umur"]
```

Contoh memilih beberapa kolom:

```
df[["umur", "pendapatan"]]
```

Contoh filtering data:

```
df[df["umur"] > 25]
```

NumPy digunakan untuk operasi numerik dan manipulasi array. Dalam praktikum machine learning, NumPy sering digunakan secara langsung maupun tidak langsung melalui pandas dan scikit-learn.

4.10 Pengenalan Matplotlib

Matplotlib digunakan untuk membuat visualisasi data dan hasil pemodelan.

Contoh visualisasi sederhana:

```
plt.hist(df["umur"])
plt.xlabel("Umur")
plt.ylabel("Frekuensi")
plt.show()
```

Visualisasi membantu mahasiswa memahami distribusi, pola, hubungan antarvariabel, serta hasil model. Pada modul berikutnya, penggunaan visualisasi akan dikembangkan sesuai tujuan analisis.

4.11 Pengenalan scikit-learn

scikit-learn digunakan untuk berbagai tahapan machine learning, mulai dari preprocessing hingga evaluasi model.

Alur umum penggunaan model dalam scikit-learn dapat diringkas menjadi:

```
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Perintah `fit()` digunakan untuk melatih model menggunakan data latih, sedangkan `predict()` digunakan untuk menghasilkan prediksi terhadap data uji.

Pada modul berikutnya, scikit-learn akan digunakan untuk klasifikasi, regresi, clustering, evaluasi model, cross-validation, dan pemilihan model sesuai cakupan RPS.

4.12 Penanganan Kesalahan Dasar

Notebook perlu disimpan secara berkala agar pekerjaan tidak hilang. Jika menggunakan Google Colab, notebook dapat disimpan di Google Drive dan diunduh dalam format `.ipynb`.

Mahasiswa sebaiknya menggunakan nama berkas yang konsisten, misalnya:

```
NIM_Nama_Modul03_EDA.ipynb
```

Apabila hasil praktikum perlu dikumpulkan dalam format PDF, mahasiswa dapat menyesuaikannya dengan instruksi dosen.

Output seperti dataset hasil pembersihan, grafik, atau file lain juga perlu disimpan pada folder yang sesuai agar dapat digunakan pada praktikum berikutnya.

4.13 Penanganan Kesalahan Dasar

Kesalahan saat menjalankan kode merupakan bagian dari proses belajar. Mahasiswa perlu membiasakan diri membaca pesan kesalahan sebelum melakukan perbaikan.

Beberapa kesalahan yang sering muncul antara lain:

- `FileNotFoundError`, yang biasanya terjadi karena file atau lokasi dataset tidak ditemukan.
- `NameError`, yang dapat terjadi karena variabel belum dibuat atau terjadi kesalahan penulisan nama variabel.
- `KeyError`, yang biasanya muncul ketika nama kolom yang digunakan tidak sesuai dengan nama kolom pada dataset.
- `TypeError` atau `ValueError`, yang dapat muncul karena tipe data tidak sesuai dengan operasi yang dilakukan.

Jika terjadi kesalahan, mahasiswa disarankan memeriksa pesan error, lokasi baris yang bermasalah, nama file, nama variabel, nama kolom, dan tipe data sebelum melakukan perubahan pada kode.

4.14 Kesiapan Sebelum Memulai Modul Praktikum

Sebelum memasuki Modul Praktikum 1, mahasiswa diharapkan telah memastikan bahwa Google Colab atau Jupyter Notebook dapat digunakan, dataset dapat dibaca, library utama dapat diimpor, dan notebook dapat disimpan dengan benar.

Kesiapan tersebut menjadi dasar agar kegiatan praktikum selanjutnya dapat berfokus pada pemahaman data, penerapan algoritma, evaluasi model, dan interpretasi hasil, bukan pada masalah konfigurasi lingkungan kerja.

MODUL 1

PENGENALAN PYTHON DAN GOOGLE COLAB

Modul pertama disusun sebagai tahap orientasi sebelum mahasiswa memasuki kegiatan eksplorasi data, praproses, pemodelan, dan evaluasi machine learning. RPS menetapkan Python sebagai bahasa pemrograman utama serta Jupyter Notebook dan Google Colab sebagai lingkungan praktikum. Karena itu, mahasiswa perlu terlebih dahulu memahami cara menggunakan lingkungan tersebut dan mengenal sintaks dasar Python yang akan digunakan pada modul-modul berikutnya.

1. Identitas Praktikum

Judul Praktikum	: Pengenalan Python dan Google Colab
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SKS
Lingkungan Praktikum	: Google Colab/Jupyter Notebook
Bahasa Pemrograman	: Python

Modul ini merupakan materi pendukung untuk menyiapkan keterampilan teknis mahasiswa sebelum memasuki Sub-CPMK inti pada RPS.

2. Capaian dan Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- 1) menggunakan Google Colab sebagai lingkungan pemrograman;
- 2) membuat, membuka, menjalankan, dan menyimpan notebook;
- 3) memahami perbedaan code cell dan text cell;
- 4) menggunakan variabel dan tipe data dasar Python;
- 5) menggunakan operator aritmatika, perbandingan, dan logika;
- 6) menggunakan struktur data sederhana seperti list dan dictionary;
- 7) menerapkan percabangan dan perulangan;
- 8) membuat fungsi sederhana;
- 9) menampilkan hasil pengolahan data sederhana;
- 10) mendokumentasikan kode dan hasil praktikum dalam notebook.

3. Dasar Teori

a. Python

Python merupakan bahasa pemrograman yang digunakan sebagai bahasa utama dalam praktikum Data Mining dan Machine Learning. Dalam RPS, Python digunakan untuk pengolahan data dan implementasi machine learning, sedangkan pustaka seperti pandas, NumPy, Matplotlib, dan scikit-learn digunakan pada tahapan analisis yang lebih lanjut.

Pada modul ini, penggunaan Python masih difokuskan pada sintaks dasar yang akan menjadi fondasi bagi praktikum selanjutnya. Mahasiswa perlu memahami variabel, tipe data, operator, struktur data, percabangan, perulangan, dan fungsi sebelum menggunakannya untuk mengolah dataset.

b. Google Colab

Google Colab merupakan lingkungan berbasis notebook yang dapat digunakan untuk menjalankan kode Python melalui peramban. Notebook memungkinkan kode, penjelasan, dan output disimpan dalam satu dokumen.

Dalam satu notebook terdapat dua jenis cell yang paling sering digunakan, yaitu:

- 1) **Code cell**, untuk menulis dan menjalankan kode Python.
- 2) **Text cell**, untuk menulis penjelasan, judul, interpretasi, dan dokumentasi menggunakan Markdown.

Penggunaan notebook sangat sesuai untuk praktikum karena kode dan hasil analisis dapat dibaca secara berurutan dan didokumentasikan dalam satu berkas.

4. Alat dan Bahan

Perangkat yang digunakan pada praktikum ini meliputi:

- 1) laptop atau komputer;
- 2) koneksi internet;
- 3) akun Google;
- 4) Google Colab atau Jupyter Notebook;
- 5) peramban web;
- 6) LMS Edlink untuk pengumpulan hasil praktikum.

Pada praktikum ini belum diperlukan dataset eksternal karena latihan menggunakan data sederhana yang dibuat langsung dalam Python.

5. Langkah Praktikum

a. Membuka Google Colab

Buka Google Colab menggunakan akun Google, kemudian buat notebook baru.

Ubah nama notebook menjadi:

```
NIM_Nama_Modul01_PythonDasar.ipynb
```

Pada bagian awal notebook, tambahkan *text cell* yang berisi identitas:

```
# MODUL 1 - PENGENALAN PYTHON DAN GOOGLE COLAB  
  
Nama      :  
NIM       :  
Kelas    :
```

b. Menjalankan Program Pertama

Pada *code cell*, tuliskan:

```
print("Selamat datang di Praktikum Data Mining dan Machine  
Learning")
```

Jalankan cell dan amati hasilnya.

Fungsi `print()` digunakan untuk menampilkan teks atau nilai ke layar.

6. Variabel dan Tipe Data

Variabel digunakan untuk menyimpan suatu nilai.

Contoh:

```
nama = "Andi"  
umur = 20  
ipk = 3.75  
aktif = True  
  
print(nama)  
print(umur)  
print(ipk)  
print(aktif)
```

Tipe data dasar Python yang sering digunakan disajikan pada Tabel 7.

Tabel 7. Tipe Data Dasar Python

Tipe Data	Contoh	Keterangan
str	"Andi"	Teks
int	20	Bilangan bulat
float	3.75	Bilangan desimal
bool	True	Nilai logika

Untuk memeriksa tipe data dapat digunakan:

```
print (type (nama) )
print (type (umur) )
print (type (ipk) )
print (type (aktif) )
```

Latihan 1

Buat variabel yang menyimpan:

- a) nama mahasiswa;
- b) NIM;
- c) semester;
- d) nilai rata-rata;
- e) status aktif.

Kemudian tampilkan seluruh nilai tersebut menggunakan print().

7. Operator Aritmatika

Python dapat digunakan untuk melakukan operasi matematika.

```
a = 10
b = 4

print(a + b)
print(a - b)
print(a * b)
print(a / b)
print(a // b)
print(a % b)
print(a ** b)
```

Operator aritmatika yang umum digunakan dalam Python, disajikan pada Tabel 8.

Tabel 8. Operator Aritmatika Python

Operator	Fungsi
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
//	Pembagian bulat
%	Sisa pembagian
**	Pangkat

Latihan 2

Diberikan nilai:

```
nilai_tugas = 85
nilai_uts = 80
nilai_uas = 90
```

Hitung nilai akhir menggunakan komposisi:

- tugas 30%;
- UTS 30%;
- UAS 40%.

Contoh:

```
nilai_akhir = (
    nilai_tugas * 0.30 +
    nilai_uts * 0.30 +
    nilai_uas * 0.40
)

print("Nilai akhir:", nilai_akhir)
```

8. Operator Perbandingan dan Logika

Operator perbandingan digunakan untuk membandingkan dua nilai.

```
nilai = 80

print(nilai > 75)
print(nilai < 75)
print(nilai == 80)
print(nilai != 80)
```

Operator perbandingan antara lain.

Tabel 9. Operator Perbandingan

Operator	Keterangan
>	lebih besar
<	lebih kecil
>=	lebih besar atau sama dengan
<=	lebih kecil atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Sedangkan Operator logika meliputi:

- 1) and;
- 2) or;
- 3) not.

Contoh:

```
nilai = 85
kehadiran = 90
print(nilai >= 75 and kehadiran >= 80)
```

9. Percabangan

Percabangan digunakan ketika program harus mengambil keputusan berdasarkan kondisi tertentu.

Contoh:

```
nilai = 82

if nilai >= 75:
    print("Lulus")
else:
    print("Tidak Lulus")
```

Percabangan juga dapat memiliki beberapa kondisi:

```
nilai = 85
if nilai >= 90:
    print("Sangat Baik")
elif nilai >= 80:
    print("Baik")
elif nilai >= 70:
    print("Cukup")
else:
    print("Perlu Perbaikan")
```

Latihan 3

Buat program sederhana yang menentukan kategori nilai mahasiswa berdasarkan nilai akhir.

10. Struktur Data List

List digunakan untuk menyimpan beberapa nilai dalam satu variabel.

```
nilai = [80, 75, 90, 85, 70]
print(nilai)
```

Mengakses isi list:

```
print(nilai[0])
print(nilai[2])
```

Menghitung jumlah data:

```
print(len(nilai))
```

Menghitung rata-rata:

```
rata_rata = sum(nilai) / len(nilai)
print("Rata-rata:", rata_rata)
```

Menambahkan data:

```
nilai.append(88)
print(nilai)
```

Latihan 4

Gunakan list berikut:

```
penjualan = [120, 150, 135, 180, 210]
```

Hitung:

- jumlah periode;
- total penjualan;
- rata-rata penjualan;
- nilai tertinggi;
- nilai terendah.

11. Dictionary

Dictionary menyimpan data dalam bentuk pasangan key dan value.

Contoh:

```
mahasiswa = {
    "nama": "Andi",
    "nim": "20260001",
    "nilai": 85
}
print(mahasiswa)
```

Mengakses nilai:

```
print(mahasiswa["nama"])
print(mahasiswa["nilai"])
```

Mengubah nilai:

```
mahasiswa["nilai"] = 90
print(mahasiswa)
```

Dictionary akan banyak membantu ketika data memiliki atribut yang berbeda tetapi saling berhubungan.

12. Perulangan

Perulangan digunakan untuk menjalankan instruksi secara berulang.

Contoh:

```
nilai = [80, 75, 90, 85]
for n in nilai:
    print(n)
```

Perulangan dengan range():

```
for i in range(5):
    print(i)
```

Contoh menghitung total secara manual:

```
nilai = [80, 75, 90, 85]
total = 0
for n in nilai:
    total = total + n
print("Total:", total)
```

Latihan 5

Diberikan:

```
penjualan = [100, 120, 90, 150, 175, 200]
```

Gunakan perulangan untuk menampilkan:

```
Penjualan periode 1 = 100
Penjualan periode 2 = 120
...
```

13. Fungsi

Fungsi digunakan untuk mengelompokkan kode agar dapat digunakan kembali.

Contoh:

```
def hitung_rata_rata(data):
    return sum(data) / len(data)
```

Kemudian:

```
nilai = [80, 75, 90, 85]
hasil = hitung_rata_rata(nilai)
print("Rata-rata:", hasil)
```

Contoh fungsi dengan dua parameter:

```
def hitung_total(harga, jumlah):
    return harga * jumlah

total = hitung_total(50000, 3)
print("Total:", total)
```

Latihan 6

Buat fungsi bernama:

```
hitung_nilai_akhir()
```

yang menerima nilai tugas, UTS, dan UAS kemudian mengembalikan nilai akhir.

14. Latihan Terintegrasi

Gunakan data berikut:

```
nilai_mahasiswa = [78, 85, 90, 72, 88, 95, 65, 80]
```

Lakukan analisis sederhana untuk memperoleh:

1. jumlah mahasiswa;
2. nilai rata-rata;
3. nilai tertinggi;
4. nilai terendah;
5. jumlah mahasiswa dengan nilai minimal 75;
6. jumlah mahasiswa dengan nilai kurang dari 75.

Contoh awal:

```
jumlah_mahasiswa = len(nilai_mahasiswa)
rata_rata = sum(nilai_mahasiswa) / jumlah_mahasiswa

print("Jumlah mahasiswa:", jumlah_mahasiswa)
print("Rata-rata:", rata_rata)
```

Selanjutnya, mahasiswa diminta melengkapi analisis menggunakan percabangan dan perulangan.

15. Pengenalan Library

Python memiliki berbagai library yang akan digunakan pada praktikum berikutnya.

Cobalah menjalankan:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Kemudian periksa versi pandas:

```
print(pd.__version__)
```

Pada modul berikutnya, pandas akan digunakan untuk membaca dan mengolah dataset, NumPy untuk operasi numerik, Matplotlib untuk visualisasi, dan scikit-learn untuk machine learning. Penggunaan library tersebut sesuai dengan perangkat pembelajaran yang tercantum dalam RPS.

16. Tugas Praktikum

Buat sebuah notebook Python untuk mengolah data penjualan berikut:

```
penjualan = [125, 150, 175, 120, 200, 225, 190, 210]
```

Notebook harus menghasilkan:

- 1) jumlah periode penjualan;
- 2) total penjualan;
- 3) rata-rata penjualan;
- 4) penjualan tertinggi;
- 5) penjualan terendah;
- 6) jumlah periode dengan penjualan di atas rata-rata;
- 7) kategori setiap periode:
 - a. **Tinggi**, jika penjualan lebih besar atau sama dengan rata-rata;
 - b. **Rendah**, jika penjualan di bawah rata-rata.

Mahasiswa diminta menggunakan:

- a. variabel;
- b. list;
- c. operator;
- d. percabangan;
- e. perulangan;
- f. minimal satu fungsi.

Hasil tidak hanya ditampilkan dalam bentuk kode, tetapi juga disertai penjelasan singkat mengenai hasil yang diperoleh.

17. Pertanyaan Analisis

Jawab pertanyaan berikut pada text cell di akhir notebook:

- 1) Apa perbedaan variabel dengan nilai yang disimpan di dalamnya?
- 2) Apa fungsi list dalam Python?
- 3) Mengapa percabangan diperlukan dalam pengolahan data?
- 4) Apa keuntungan menggunakan fungsi dibandingkan menuliskan kode yang sama berulang kali?
- 5) Mengapa dokumentasi dalam notebook penting dalam praktikum Data Mining dan Machine Learning?
- 6) Dari hasil tugas penjualan, apa kesimpulan yang dapat diperoleh mengenai pola penjualan?

18. Hasil yang Diharapkan

Setelah menyelesaikan praktikum, mahasiswa diharapkan memiliki notebook yang:

- 1) dapat dijalankan tanpa kesalahan;
- 2) memiliki struktur yang jelas;
- 3) menggunakan sintaks dasar Python dengan benar;
- 4) menghasilkan perhitungan sesuai instruksi;
- 5) memuat penjelasan hasil;
- 6) memuat jawaban pertanyaan analisis;
- 7) memuat kesimpulan.

19. Kesimpulan Praktikum

Pada akhir kegiatan, mahasiswa menuliskan kesimpulan dengan bahasa sendiri. Kesimpulan sekurang-kurangnya menjelaskan keterampilan yang telah diperoleh dalam

menggunakan Google Colab dan Python serta keterkaitan kemampuan tersebut dengan kegiatan analisis data pada praktikum berikutnya.

Contoh arah kesimpulan:

Praktikum ini memberikan pemahaman dasar mengenai penggunaan Google Colab dan sintaks Python, meliputi variabel, tipe data, operator, list, dictionary, percabangan, perulangan, dan fungsi. Konsep-konsep tersebut menjadi dasar dalam pengolahan dataset serta implementasi analisis data dan machine learning pada praktikum selanjutnya.

20. Rubrik Penilaian

Rubrik penilain yang digunakan pada modul ini yaitu:

Aspek Penilaian	Bobot
Ketepatan penggunaan sintaks Python	25%
Ketepatan hasil perhitungan	20%
Penggunaan percabangan, perulangan, dan fungsi	20%
Struktur dan dokumentasi notebook	15%
Jawaban pertanyaan analisis	10%
Kesimpulan	10%
Total	100%

Penilaian tidak hanya didasarkan pada keberhasilan kode berjalan, tetapi juga pada pemahaman, dokumentasi, serta kemampuan mahasiswa menjelaskan hasil yang diperoleh.

Referensi

1. McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.
2. Dokumentasi resmi Python.
3. Dokumentasi resmi Jupyter.
4. Dokumentasi Google Colab.
5. Dokumentasi pandas dan NumPy.

MODUL 2

CRISP-DM DAN PERUMUSAN MASALAH ANALITIK

Modul ini membahas tahapan awal dalam proses data mining dan machine learning, yaitu memahami permasalahan, menentukan tujuan analisis, serta menyusun alur penyelesaian menggunakan pendekatan CRISP-DM. Dalam RPS, materi ini ditempatkan pada tahap awal pembelajaran dan menekankan kemampuan mahasiswa dalam menguraikan tahapan KDD dan CRISP-DM serta merumuskan masalah analitik dari studi kasus bisnis.

1. Identitas Praktikum

Judul Praktikum : CRISP-DM dan Perumusan Masalah Analitik
Mata Kuliah : Data Mining dan Machine Learning
Bobot Mata Kuliah : 3 SKS
CPMK : CPMK-1
Sub-CPMK : Sub-CPMK-1.1

Sub-CPMK yang dicapai adalah:

Mampu menjelaskan konsep, tujuan, ruang lingkup, tahapan, dan penerapan data mining serta machine learning dalam bidang Sistem Informasi.

2. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- 1) menjelaskan tahapan proses KDD dan CRISP-DM;
- 2) mengidentifikasi masalah bisnis dari suatu studi kasus;
- 3) membedakan masalah bisnis dan masalah analitik;
- 4) merumuskan tujuan analisis berdasarkan kebutuhan organisasi;
- 5) menentukan jenis data yang diperlukan;
- 6) mengidentifikasi kemungkinan variabel target dan fitur;
- 7) menentukan pendekatan analitik yang sesuai;
- 8) menyusun rancangan awal proyek data mining atau machine learning.

3. Dasar Teori

3.1. Proses Data Mining

Data mining tidak dimulai dari pemilihan algoritma. Prosesnya diawali dengan memahami permasalahan yang ingin diselesaikan, memahami data yang tersedia, menyiapkan data, melakukan pemodelan, mengevaluasi hasil, kemudian menggunakan hasil tersebut untuk mendukung tujuan organisasi.

Dalam RPS, proses penemuan pengetahuan dari data dibahas melalui pendekatan KDD dan CRISP-DM, termasuk proses merumuskan masalah bisnis dan mengubahnya menjadi masalah analitik.

3.2. Knowledge Discovery in Databases

Knowledge Discovery in Databases atau KDD merupakan proses untuk menemukan pengetahuan yang berguna dari sekumpulan data.

Secara umum, tahapan yang dipelajari dalam modul ini meliputi:

- 1) **Selection**, yaitu memilih data yang relevan dengan permasalahan.
- 2) **Preprocessing**, yaitu menangani masalah kualitas data seperti nilai hilang, duplikasi, atau inkonsistensi.
- 3) **Transformation**, yaitu mengubah data menjadi bentuk yang sesuai untuk analisis.
- 4) **Data Mining**, yaitu menerapkan metode untuk menemukan pola atau membangun model.
- 5) **Interpretation/Evaluation**, yaitu menilai dan menginterpretasikan hasil yang diperoleh.

KDD memberikan gambaran bahwa data mining merupakan bagian dari proses yang lebih luas, bukan satu-satunya tahapan dalam analisis data.

3.3. CRISP-DM

CRISP-DM atau Cross-Industry Standard Process for Data Mining merupakan kerangka kerja yang dapat digunakan untuk mengorganisasi tahapan proyek data mining.

Dalam modul ini, CRISP-DM terdiri atas enam tahapan utama:

- 1) Business Understanding
- 2) Data Understanding
- 3) Data Preparation
- 4) Modeling
- 5) Evaluation
- 6) Deployment

Tahapan tersebut tidak selalu bersifat linear. Pada praktiknya, hasil dari satu tahap dapat menyebabkan proses kembali ke tahap sebelumnya untuk melakukan penyesuaian.

3.4. Business Understanding

Tahap pertama berfokus pada pemahaman masalah dari perspektif organisasi atau bisnis.

Pertanyaan yang perlu dijawab antara lain:

- 1) Apa masalah yang sedang dihadapi?
- 2) Mengapa masalah tersebut penting?
- 3) Siapa yang membutuhkan hasil analisis?
- 4) Keputusan apa yang akan didukung?
- 5) Bagaimana keberhasilan analisis akan dinilai?

Contoh:

Sebuah perusahaan telekomunikasi mengalami peningkatan jumlah pelanggan yang berhenti menggunakan layanan.

Masalah tersebut belum merupakan masalah machine learning. Mahasiswa perlu memahami terlebih dahulu tujuan organisasi, misalnya mengurangi tingkat kehilangan pelanggan.

3.5. Data Understanding

Tahap ini bertujuan memahami data yang tersedia.

Kegiatan dapat mencakup:

- 1) mengidentifikasi sumber data;

- 2) memahami atribut;
- 3) menentukan jumlah data;
- 4) memeriksa kualitas data;
- 5) mengidentifikasi target;
- 6) melakukan eksplorasi awal.

Contoh data yang mungkin dibutuhkan untuk kasus pelanggan meliputi:

- 1) lama berlangganan;
- 2) jenis layanan;
- 3) tagihan bulanan;
- 4) frekuensi penggunaan;
- 5) keluhan;
- 6) metode pembayaran;
- 7) status berhenti berlangganan.

3.6. Data Preparation

Pada tahap ini, data dipersiapkan agar dapat digunakan dalam proses pemodelan.

Aktivitas dapat meliputi:

- 1) pembersihan data;
- 2) penanganan missing value;
- 3) transformasi;
- 4) encoding;
- 5) standardisasi;
- 6) seleksi fitur;
- 7) pembagian dataset.

Tahap ini akan dipelajari lebih rinci pada modul praktikum berikutnya.

3.7. Modeling

Modeling merupakan tahap penerapan algoritma terhadap data yang telah dipersiapkan. Pemilihan algoritma disesuaikan dengan jenis masalah.

Permasalahan	Pendekatan
Memprediksi pelanggan berhenti atau tidak	Klasifikasi
Memprediksi jumlah penjualan	Regresi

Mengelompokkan pelanggan	Clustering
Menemukan pola pembelian	Association Rule

Algoritma yang akan digunakan pada modul berikutnya meliputi KNN, Decision Tree, Naive Bayes, regresi linear, K-Means, serta teknik clustering dan penemuan pola sesuai RPS.

3.8. Evaluation

Model yang telah dibuat perlu dievaluasi untuk mengetahui apakah hasilnya sesuai dengan tujuan analisis.

Evaluasi dapat mempertimbangkan:

- 1) performa model;
- 2) kesesuaian metrik;
- 3) konsistensi hasil;
- 4) relevansi terhadap permasalahan;
- 5) kemungkinan keterbatasan.

Pada tahap selanjutnya mahasiswa akan menggunakan metrik seperti accuracy, precision, recall, F1-score, MAE, RMSE, R^2 , dan silhouette score sesuai jenis model.

3.9. Deployment

Deployment merupakan tahap penggunaan hasil analisis.

Deployment tidak selalu berarti membuat aplikasi. Dalam konteks pembelajaran, hasil analisis dapat disampaikan dalam bentuk:

- 1) laporan;
- 2) dashboard;
- 3) visualisasi;
- 4) rekomendasi;
- 5) presentasi;
- 6) model prediksi;
- 7) prototipe.

Tujuan utamanya adalah memastikan hasil analisis dapat digunakan untuk mendukung kebutuhan organisasi.

4. Masalah Bisnis dan Masalah Analitik

Salah satu kemampuan penting dalam data mining adalah menerjemahkan masalah bisnis menjadi masalah analitik.

Masalah bisnis menjelaskan kondisi atau kebutuhan organisasi.

Contoh:

Penjualan perusahaan mengalami penurunan dalam beberapa bulan terakhir.

Masalah analitik menjelaskan bagaimana data digunakan untuk membantu memahami atau menyelesaikan masalah tersebut.

Contoh:

Mengidentifikasi faktor yang berhubungan dengan penurunan penjualan dan membangun model untuk memprediksi penjualan periode berikutnya.

Perumusan masalah analitik membantu menentukan data, metode, dan evaluasi yang diperlukan.

5. Menentukan Jenis Permasalahan Machine Learning

Setelah masalah dirumuskan, mahasiswa perlu mengidentifikasi jenis analisis yang sesuai.

Klasifikasi Digunakan ketika target berbentuk kategori.

Contoh:

- 1) pelanggan churn atau tidak;
- 2) kredit layak atau tidak;
- 3) mahasiswa lulus tepat waktu atau tidak.

Regresi Digunakan ketika target berupa nilai numerik.

Contoh:

- 1) prediksi penjualan;
- 2) prediksi harga;
- 3) prediksi jumlah transaksi.

Clustering Digunakan untuk mengelompokkan objek berdasarkan kemiripan tanpa label target.

Contoh:

- 1) segmentasi pelanggan;
- 2) pengelompokan produk;
- 3) pengelompokan wilayah.

Association Rule Mining Digunakan untuk menemukan pola hubungan antaritem atau kejadian.

Contoh:

pelanggan yang membeli produk A juga cenderung membeli produk B.

6. Studi Kasus Praktikum

Perhatikan studi kasus berikut.

Sebuah perusahaan layanan internet mengalami peningkatan jumlah pelanggan yang menghentikan langganan. Manajemen ingin mengetahui karakteristik pelanggan yang memiliki risiko berhenti dan berharap hasil analisis dapat digunakan sebagai dasar penyusunan program retensi pelanggan.

Berdasarkan kasus tersebut, mahasiswa diminta melakukan analisis awal menggunakan kerangka CRISP-DM.

Tahap 1. Business Understanding

Identifikasi:

Masalah bisnis:

Meningkatnya jumlah pelanggan yang menghentikan layanan.

Tujuan bisnis:

Mengurangi jumlah pelanggan yang berhenti berlangganan.

Pertanyaan bisnis:

Karakteristik pelanggan seperti apa yang lebih berpotensi berhenti menggunakan layanan?

Tahap 2. Perumusan Masalah Analitik

Masalah analitik dapat dirumuskan menjadi:

Membangun model klasifikasi yang dapat mengidentifikasi pelanggan yang berpotensi berhenti berlangganan berdasarkan karakteristik pelanggan dan riwayat penggunaan layanan.

Target:

Churn

Contoh fitur potensial:

Tenure
MonthlyCharges
ContractType
InternetService
PaymentMethod
SupportCalls
TotalCharges

7. Lembar Kerja CRISP-DM

Mahasiswa mengisi tabel berikut berdasarkan studi kasus.

Tabel 10. Lembar Kerja CRISP-DM

Tahap CRISP-DM	Hasil Analisis
Business Understanding	...
Data Understanding	...
Data Preparation	...
Modeling	...
Evaluation	...
Deployment	...

Untuk *Business Understanding*, jelaskan:

- 1) masalah utama;
- 2) tujuan organisasi;
- 3) pihak yang membutuhkan hasil;
- 4) indikator keberhasilan.

Untuk *Data Understanding*, jelaskan:

- 1) data yang diperlukan;
- 2) sumber data;
- 3) variabel target;
- 4) kemungkinan fitur.

Untuk *Data Preparation*, identifikasi kemungkinan:

- 1) missing value;
- 2) duplikasi;
- 3) data kategorikal;
- 4) pencilan;
- 5) transformasi.

Untuk *Modeling*, tentukan:

- 1) jenis masalah;
- 2) algoritma yang mungkin digunakan;
- 3) alasan pemilihan.

Untuk *Evaluation*, tentukan:

- 1) cara mengevaluasi model;
- 2) hasil apa yang dianggap baik.

Untuk *Deployment*, jelaskan bagaimana hasil dapat digunakan organisasi.

8. Latihan Perumusan Masalah

Perhatikan kasus berikut.

Kasus A. Prediksi Penjualan

Sebuah toko ingin memperkirakan jumlah penjualan produk pada bulan berikutnya berdasarkan data transaksi sebelumnya.

Tentukan:

- 1) masalah bisnis;
- 2) masalah analitik;
- 3) jenis machine learning;
- 4) target;
- 5) fitur potensial;
- 6) metode evaluasi.

Kasus B. Segmentasi Pelanggan

Sebuah perusahaan memiliki data pelanggan, tetapi belum mengetahui kelompok karakteristik pelanggannya.

Tentukan:

- 1) tujuan bisnis;
- 2) masalah analitik;
- 3) jenis machine learning;
- 4) data yang diperlukan;
- 5) kemungkinan manfaat hasil.

Kasus C. Risiko Kelulusan

Sebuah perguruan tinggi ingin mengidentifikasi mahasiswa yang berpotensi tidak menyelesaikan studi tepat waktu.

Tentukan:

- 1) masalah bisnis;
- 2) masalah analitik;
- 3) target;
- 4) fitur potensial;
- 5) metode yang dapat digunakan;
- 6) risiko etika yang perlu diperhatikan.

9. Aktivitas Kelompok

Mahasiswa dibagi menjadi kelompok kecil. Setiap kelompok memilih satu permasalahan dalam bidang Sistem Informasi, misalnya:

- 1) pelayanan pelanggan;
- 2) penjualan;
- 3) pemasaran;
- 4) akademik;
- 5) layanan kesehatan;
- 6) transaksi;
- 7) manajemen persediaan;
- 8) layanan publik.

Kelompok kemudian menyusun:

- 1) deskripsi organisasi atau konteks;
- 2) masalah bisnis;
- 3) tujuan analisis;
- 4) masalah analitik;
- 5) jenis machine learning;
- 6) kebutuhan dataset;
- 7) target dan fitur potensial;
- 8) tahapan CRISP-DM;
- 9) hasil yang diharapkan.

Hasil analisis dipresentasikan secara singkat dan didiskusikan bersama.

10. Tugas Praktikum

Pilih satu studi kasus yang relevan dengan bidang Sistem Informasi. Susun analisis awal proyek menggunakan pendekatan CRISP-DM.

Dokumen tugas sekurang-kurangnya memuat:

- a. Judul studi kasus
- b. Latar belakang singkat
- c. Masalah bisnis
- d. Tujuan bisnis
- e. Masalah analitik
- f. Jenis pendekatan machine learning
- g. Dataset yang dibutuhkan
- h. Target dan fitur potensial
- i. Rancangan tahapan CRISP-DM
- j. Indikator keberhasilan
- k. Potensi manfaat hasil analisis
- l. Potensi risiko atau pertimbangan etika

Tugas dapat disusun dalam notebook atau dokumen sesuai instruksi dosen dan dikumpulkan melalui LMS Edlink.

11. Pertanyaan Analisis

Jawab pertanyaan berikut:

Mengapa proses data mining sebaiknya tidak langsung dimulai dari pemilihan algoritma?

- 1) Apa perbedaan utama antara masalah bisnis dan masalah analitik?
- 2) Mengapa Business Understanding penting dalam CRISP-DM?
- 3) Apa hubungan antara jenis target dengan pemilihan metode machine learning?
- 4) Mengapa tahapan CRISP-DM dapat bersifat iteratif?
- 5) Apa risiko yang mungkin terjadi jika model memiliki performa tinggi tetapi tidak sesuai dengan tujuan bisnis?
- 6) Pada studi kasus yang dikerjakan, apa pertimbangan etika yang perlu diperhatikan?

12. Hasil yang Diharapkan

Setelah menyelesaikan praktikum, mahasiswa diharapkan mampu menghasilkan dokumen analisis yang menunjukkan hubungan antara:

masalah bisnis → tujuan analisis → data → metode → evaluasi → pemanfaatan hasil

Mahasiswa juga diharapkan mampu menjelaskan mengapa metode tertentu dipilih dan bagaimana hasil analisis dapat mendukung pengambilan keputusan.

13. Kesimpulan Praktikum

Pada akhir praktikum, mahasiswa menuliskan kesimpulan berdasarkan kegiatan yang telah dilakukan.

Kesimpulan sekurang-kurangnya menjelaskan:

- a. pentingnya memahami masalah sebelum melakukan pemodelan;
- b. fungsi CRISP-DM dalam mengorganisasi proyek;
- c. perbedaan masalah bisnis dan masalah analitik;
- d. hubungan antara tujuan, data, metode, dan evaluasi.

Contoh arah kesimpulan:

CRISP-DM memberikan kerangka kerja yang sistematis untuk menyelesaikan proyek data mining mulai dari memahami kebutuhan organisasi hingga penggunaan hasil analisis. Perumusan masalah analitik yang tepat menjadi dasar dalam menentukan dataset, target, fitur, metode pemodelan, dan cara evaluasi yang sesuai.

14. Rubrik Penilaian

Aspek Penilaian	Bobot
Ketepatan identifikasi masalah bisnis	20%
Ketepatan perumusan masalah analitik	20%
Kelengkapan tahapan CRISP-DM	20%
Kesesuaian data dan metode	15%
Argumentasi dan interpretasi	15%
Dokumentasi dan penyajian	10%
Total	100%

Referensi

1. Han, J., Pei, J., & Tong, H. (2022). Data Mining: Concepts and Techniques (4th ed.). Morgan Kaufmann.
2. Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly Media.
3. Géron, A. (2023). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.

MODUL 3

JENIS DATA DAN EXPLORATORY DATA ANALYSIS

Modul ini membahas proses Exploratory Data Analysis (EDA) sebagai tahap awal untuk memahami karakteristik dataset sebelum dilakukan pembersihan, transformasi, dan pemodelan. Pada tahap ini, mahasiswa mempelajari cara membaca dataset, mengenali struktur data, mengidentifikasi jenis atribut, memeriksa kualitas data, menghitung statistik deskriptif, serta membuat visualisasi awal menggunakan Python dan pandas.

Dalam RPS, kegiatan ini berkaitan dengan kemampuan mahasiswa dalam mengidentifikasi jenis data, masalah kualitas data, statistik deskriptif, serta menentukan pendekatan pembelajaran yang sesuai. Kegiatan pembelajaran dilakukan melalui demonstrasi, praktikum Python/pandas, dan diskusi hasil eksplorasi.

1. Identitas Praktikum

Judul Praktikum	: Exploratory Data Analysis dengan pandas
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SKS
CPMK	: CPMK-1
Sub-CPMK	: Sub-CPMK-1.2
Lingkungan Praktikum	: Google Colab/Jupyter Notebook
Library Utama	: pandas, NumPy, Matplotlib

Sub-CPMK yang dicapai adalah:

Mampu menjelaskan jenis data, teknik eksplorasi data, serta karakteristik metode *supervised* dan *unsupervised learning*.

2. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- a. membaca dataset menggunakan pandas;
- b. mengidentifikasi jumlah baris dan kolom pada dataset;
- c. mengenali nama atribut dan tipe data;
- d. membedakan data numerik dan kategorikal;
- e. menghitung statistik deskriptif;

- f. memeriksa nilai kosong dan data duplikat;
- g. menganalisis distribusi data;
- h. mengidentifikasi kemungkinan pencilan;
- i. membuat visualisasi eksploratif sederhana;
- j. menyusun interpretasi awal terhadap karakteristik dataset.

3. Dasar Teori

3.10. Exploratory Data Analysis

Exploratory Data Analysis atau EDA merupakan proses awal untuk memahami struktur, karakteristik, pola, dan kualitas suatu dataset sebelum dilakukan pemodelan lebih lanjut. EDA membantu analis memperoleh gambaran mengenai kondisi data serta menemukan permasalahan yang perlu diperhatikan pada tahap preprocessing.

Melalui EDA, beberapa pertanyaan awal yang dapat dijawab antara lain:

- 1) Berapa jumlah observasi dan atribut?
- 2) Apa saja variabel yang tersedia?
- 3) Apa tipe data setiap atribut?
- 4) Apakah terdapat nilai kosong?
- 5) Apakah terdapat data duplikat?
- 6) Bagaimana distribusi nilai pada setiap variabel?
- 7) Apakah terdapat nilai ekstrem atau pencilan?
- 8) Apakah terdapat hubungan antarvariabel?
- 9) Bagaimana distribusi variabel target?

Pemahaman tersebut penting karena kualitas pemodelan machine learning sangat dipengaruhi oleh karakteristik dan kualitas dataset yang digunakan.

3.11. Dataset dan Atribut

Dataset merupakan kumpulan data yang disusun untuk tujuan tertentu. Dalam bentuk tabular, dataset biasanya terdiri atas baris dan kolom.

Baris mewakili satu observasi atau record, sedangkan kolom mewakili atribut atau variabel.

Tabel 11. Contoh Dataset dan Atribut

ID	Umur	Pendapatan	Jenis Kelamin	Churn
1	24	3500000	L	Tidak
2	31	5200000	P	Ya
3	28	4100000	L	Tidak

Pada dataset dan atribut pada tabel tersebut menunjukkan, setiap baris merupakan seorang pelanggan, sedangkan Umur, Pendapatan, Jenis Kelamin, dan Churn merupakan atribut.

3.12. Jenis Data

Secara umum, data dapat dikelompokkan menjadi data numerik dan kategorikal.

- a. **Data numerik** merupakan data berbentuk angka yang dapat digunakan dalam operasi matematika.

Contoh:

- 1) umur;
- 2) pendapatan;
- 3) jumlah transaksi;
- 4) nilai ujian.

- b. **Data numerik** dapat berupa diskrit maupun kontinu.

- c. **Data kategorikal** merupakan data yang merepresentasikan kelompok atau kategori tertentu.

Contoh:

- 1) jenis kelamin;
- 2) status pelanggan;
- 3) jenis kontrak;
- 4) metode pembayaran.

Pemahaman terhadap jenis data diperlukan karena proses analisis dan visualisasi akan berbeda untuk setiap tipe variabel.

4. Persiapan Praktikum

Buat notebook baru di Google Colab dengan nama:

```
NIM_Nama_Modul03_EDA.ipynb
```

Tambahkan identitas pada bagian awal notebook:

```
# MODUL 3 - EXPLORATORY DATA ANALYSIS

Nama :
NIM :
Kelas :
```

Selanjutnya, impor library yang digunakan:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

5. Dataset Praktikum

Untuk kegiatan praktikum, gunakan dataset pelanggan sederhana dengan atribut yang dapat menggambarkan karakteristik pengguna layanan.

Contoh struktur dataset:

Atribut	Keterangan
CustomerID	Identitas pelanggan
Age	Umur pelanggan
Gender	Jenis kelamin
Tenure	Lama berlangganan
MonthlyCharges	Tagihan bulanan
TotalCharges	Total pembayaran
Contract	Jenis kontrak
PaymentMethod	Metode pembayaran
Churn	Status berhenti berlangganan

Dataset dapat diberikan oleh dosen dalam format CSV dan diunggah ke Google Colab.

6. Membaca Dataset dengan pandas

Gunakan fungsi `read_csv()` untuk membaca dataset:

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

Tampilkan lima data pertama:

```
df.head()
```

Tampilkan lima data terakhir:

```
df.tail()
```

Tujuannya adalah memperoleh gambaran awal mengenai isi dataset.

7. Mengenali Ukuran Dataset

Gunakan:

```
df.shape
```

Output berbentuk:

```
(jumlah_baris, jumlah_kolom)
```

Contoh:

```
(1000, 9)
```

berarti dataset memiliki 1.000 baris dan 9 kolom.

Jumlah baris dapat diperiksa dengan:

```
len(df)
```

Sedangkan nama seluruh kolom dapat dilihat menggunakan:

```
df.columns
```

Latihan 1

Catat:

- a. jumlah baris;
- b. jumlah kolom;
- c. nama seluruh atribut.

Tuliskan hasilnya dalam text cell.

8. Memeriksa Struktur dan Tipe Data

Gunakan:

```
df.info()
```

Fungsi `info()` menampilkan informasi mengenai:

- a. nama kolom;
- b. jumlah data non-null;
- c. tipe data;
- d. penggunaan memori.

Untuk melihat tipe data setiap kolom:

```
df.dtypes
```

Mahasiswa kemudian mengidentifikasi atribut mana yang termasuk:

- a. numerik;
- b. kategorikal;
- c. identitas;
- d. target.

Latihan 2

Buat tabel sederhana:

Atribut	Tipe Data Python	Jenis Data	Peran
Age	int64	Numerik	Fitur
Gender	object	Kategorikal	Fitur
...	...	...	...

9. Statistik Deskriptif

Statistik deskriptif digunakan untuk memperoleh ringkasan karakteristik data numerik.

Gunakan:

```
df.describe()
```

Output umumnya memuat:

- 1) count;
- 2) mean;
- 3) std;
- 4) min;
- 5) 25%;
- 6) 50%;
- 7) 75%;
- 8) max.

Untuk satu kolom:

```
df["Age"].describe()
```

Rata-rata:

```
df["Age"].mean()
```

Median:

```
df["Age"].median()
```

Nilai minimum:

```
df["Age"].min()
```

Nilai maksimum:

```
df["Age"].max()
```

Simpangan baku:

```
df["Age"].std()
```

Latihan 3

Hitung statistik berikut untuk `MonthlyCharges`:

- 1) rata-rata;
- 2) median;
- 3) minimum;
- 4) maksimum;
- 5) simpangan baku.

Kemudian jelaskan secara singkat apa yang dapat disimpulkan dari hasil tersebut.

10. Eksplorasi Data Kategorikal

Untuk melihat nilai unik pada suatu kolom:

```
df["Contract"].unique()
```

Untuk melihat jumlah kategori:

```
df["Contract"].nunique()
```

Untuk melihat frekuensi setiap kategori:

```
df["Contract"].value_counts()
```

Proporsi kategori dapat dilihat menggunakan:

```
df["Contract"].value_counts(normalize=True)
```

Contoh untuk variabel target:

```
df["Churn"].value_counts()
```

Latihan 4

Analisis:

- 1) distribusi Gender;
- 2) distribusi Contract;
- 3) distribusi PaymentMethod;
- 4) distribusi Churn.

Tuliskan kategori yang paling banyak dan paling sedikit pada setiap variabel.

11. Memeriksa Missing Value

Nilai kosong dapat memengaruhi proses analisis dan pemodelan.

Gunakan:

```
df.isnull().sum()
```

Untuk melihat persentase missing value:

```
missing_percentage = df.isnull().mean() * 100
print(missing_percentage)
```

Apabila terdapat missing value, pada tahap EDA mahasiswa cukup mengidentifikasi dan mendokumentasikannya. Penanganannya akan dibahas secara khusus pada Modul 4.

Latihan 5

Jawab:

- a. atribut mana yang memiliki missing value?
- b. berapa jumlahnya?
- c. berapa persentasenya?
- d. apakah atribut tersebut perlu diperhatikan pada tahap preprocessing?

12. Memeriksa Data Duplikat

Gunakan:

```
df.duplicated().sum()
```

Kode tersebut menghitung jumlah baris yang terduplikasi.

Untuk melihat data duplikat:

```
df[df.duplicated()]
```

Pada tahap ini, mahasiswa hanya melakukan identifikasi. Penghapusan data duplikat akan dibahas pada Modul 4.

13. Memeriksa Distribusi Data Numerik

Histogram dapat digunakan untuk melihat bentuk distribusi data.

Contoh:

```
plt.hist(df["Age"], bins=10)
plt.xlabel("Umur")
plt.ylabel("Frekuensi")
plt.title("Distribusi Umur Pelanggan")
plt.show()
```

Contoh lain:

```
plt.hist(df["MonthlyCharges"], bins=10)
plt.xlabel("Tagihan Bulanan")
plt.ylabel("Frekuensi")
plt.title("Distribusi Tagihan Bulanan")
plt.show()
```

Mahasiswa perlu memperhatikan apakah distribusi:

- relatif simetris;
- condong ke kiri;
- condong ke kanan;
- memiliki nilai ekstrem.

14. Visualisasi Data Kategorikal

Distribusi kategori dapat divisualisasikan menggunakan diagram batang.

Contoh:

```
df["Churn"].value_counts().plot(kind="bar")

plt.xlabel("Status Churn")
plt.ylabel("Jumlah Pelanggan")
plt.title("Distribusi Status Churn")
plt.show()
```

Contoh untuk kontrak:

```
df["Contract"].value_counts().plot(kind="bar")

plt.xlabel("Jenis Kontrak")
plt.ylabel("Jumlah Pelanggan")
plt.title("Distribusi Jenis Kontrak")
plt.show()
```

Latihan 6

Buat visualisasi untuk:

- Gender;
- Contract;
- Churn.

Berikan interpretasi singkat pada setiap grafik.

15. Mengidentifikasi Pencilan Secara Awal

Pencilan atau outlier merupakan data yang memiliki nilai jauh berbeda dari sebagian besar observasi lainnya.

Visualisasi awal dapat dilakukan menggunakan boxplot:

```
plt.boxplot(df["MonthlyCharges"].dropna())

plt.ylabel("Monthly Charges")
plt.title("Boxplot Monthly Charges")
plt.show()
```

Pada tahap EDA, mahasiswa belum harus menghapus pencilan. Mahasiswa hanya perlu mengenali keberadaannya dan mempertimbangkan apakah nilai tersebut wajar atau membutuhkan pemeriksaan lebih lanjut.

16. Menganalisis Hubungan Antarvariabel

Untuk dua variabel numerik, scatter plot dapat digunakan.

Contoh:

```
plt.scatter(df["Tenure"], df["TotalCharges"])

plt.xlabel("Tenure")
plt.ylabel("Total Charges")
plt.title("Hubungan Tenure dan Total Charges")
plt.show()
```

Mahasiswa dapat mengamati apakah:

- terdapat kecenderungan hubungan;
- hubungan bersifat positif atau negatif;
- terdapat kelompok tertentu;
- terdapat nilai ekstrem.

17. Korelasi Data Numerik

Korelasi dapat digunakan untuk melihat tingkat hubungan linier antarvariabel numerik.

Pilih data numerik:

```
numeric_df = df.select_dtypes(include=["number"])
```

Kemudian:

```
correlation = numeric_df.corr()
print(correlation)
```

Mahasiswa perlu memahami bahwa korelasi tidak selalu menunjukkan hubungan sebab-akibat.

Latihan 7

Identifikasi:

- dua variabel dengan korelasi positif paling kuat;
- dua variabel dengan korelasi paling lemah;
- kemungkinan alasan hubungan tersebut.

18. Eksplorasi Variabel Target

Apabila dataset memiliki target seperti Churn, distribusi target perlu diperiksa.

```
df["Churn"].value_counts()
```

Kemudian:

```
df["Churn"].value_counts(normalize=True) * 100
```

Mahasiswa perlu memperhatikan apakah distribusi kelas relatif seimbang atau tidak.

Contoh interpretasi:

Jika 85% pelanggan termasuk kelas Tidak Churn dan hanya 15% Churn, maka dataset memiliki distribusi kelas yang tidak seimbang. Kondisi tersebut perlu diperhatikan pada tahap pemodelan dan evaluasi.

19. Ringkasan Hasil EDA

Setelah seluruh eksplorasi dilakukan, buat ringkasan hasil menggunakan tabel berikut:

Aspek	Temuan
Jumlah data	...
Jumlah atribut	...
Atribut numerik	...
Atribut kategorikal	...
Target	...
Missing value	...
Data duplikat	...
Potensi outlier	...
Distribusi target	...
Pola penting	...

Ringkasan ini menjadi dasar untuk menentukan langkah pada proses pembersihan data.

20. Tugas Praktikum

Gunakan dataset yang diberikan oleh dosen dan lakukan EDA secara lengkap.

Notebook tugas minimal memuat:

- 1) identitas dataset;
- 2) jumlah baris dan kolom;
- 3) nama atribut;

- 4) tipe data;
- 5) identifikasi variabel numerik dan kategorikal;
- 6) statistik deskriptif;
- 7) pemeriksaan missing value;
- 8) pemeriksaan data duplikat;
- 9) distribusi minimal dua variabel numerik;
- 10) distribusi minimal dua variabel kategorikal;
- 11) analisis variabel target;
- 12) minimal satu analisis hubungan antarvariabel;
- 13) identifikasi awal pencilan;
- 14) ringkasan temuan;
- 15) kesimpulan.

Setiap output harus disertai penjelasan singkat mengenai makna hasil yang diperoleh.

21. Pertanyaan Analisis

Jawab pertanyaan berikut pada bagian akhir notebook:

- 1) Mengapa EDA perlu dilakukan sebelum preprocessing dan pemodelan?
- 2) Apa perbedaan data numerik dan kategorikal?
- 3) Apa informasi yang diperoleh dari `df.info()`?
- 4) Mengapa statistik rata-rata saja tidak selalu cukup untuk memahami suatu variabel?
- 5) Apa dampak missing value terhadap proses analisis?
- 6) Mengapa data duplikat perlu diidentifikasi?
- 7) Apa fungsi histogram dalam EDA?
- 8) Bagaimana cara mengenali kemungkinan outlier?
- 9) Mengapa distribusi variabel target perlu diperiksa?
- 10) Apa tiga temuan paling penting dari dataset yang dianalisis?

22. Hasil yang Diharapkan

Setelah menyelesaikan praktikum, mahasiswa diharapkan menghasilkan notebook yang mampu menunjukkan:

- a. struktur dan karakteristik dataset;
- b. tipe setiap atribut;

- c. kondisi kualitas data;
- d. statistik deskriptif;
- e. distribusi variabel;
- f. hubungan awal antarvariabel;
- g. kondisi variabel target;
- h. potensi masalah yang perlu ditangani pada tahap berikutnya.

Hasil EDA harus dapat digunakan sebagai dasar untuk menentukan tindakan pada proses pembersihan dan transformasi data.

23. Kesimpulan Praktikum

Pada akhir praktikum, mahasiswa menyusun kesimpulan berdasarkan hasil eksplorasi dataset yang digunakan.

Kesimpulan sebaiknya memuat:

- a. gambaran umum dataset;
- b. kondisi kualitas data;
- c. karakteristik variabel utama;
- d. temuan pola awal;
- e. permasalahan yang perlu ditangani pada tahap preprocessing.

Contoh arah kesimpulan:

Berdasarkan hasil Exploratory Data Analysis, dataset dapat dipahami melalui pemeriksaan struktur, tipe atribut, statistik deskriptif, distribusi data, serta kualitas data. Beberapa permasalahan seperti missing value, data duplikat, dan kemungkinan pencilan perlu ditangani sebelum proses pemodelan. Hasil eksplorasi juga memberikan gambaran awal mengenai pola pada data dan distribusi variabel target.

24. Rubrik Penilaian

Aspek Penilaian	Bobot
Ketepatan identifikasi struktur dan jenis data	20%
Kelengkapan statistik deskriptif	15%
Ketepatan identifikasi kualitas data	20%
Kualitas visualisasi	15%
Interpretasi hasil eksplorasi	20%
Dokumentasi dan kesimpulan	10%
Total	100%

Rubrik ini diselaraskan dengan indikator pada RPS yang menilai lembar kerja eksplorasi berdasarkan ketepatan identifikasi, kelengkapan analisis, dan interpretasi statistik.

Referensi

1. Han, J., Pei, J., & Tong, H. (2022). Data Mining: Concepts and Techniques (4th ed.). Morgan Kaufmann.
2. McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.
3. Géron, A. (2023). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.
4. Dokumentasi resmi pandas.
5. Dokumentasi resmi NumPy.
6. Dokumentasi resmi Matplotlib.

MODUL 4

PEMBERSIHAN DATA

1. Identitas Praktikum

Judul Praktikum	: Pembersihan Data
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SKS
CPMK	: CPMK-2 — Mampu menyiapkan data dan menerapkan algoritma supervised serta unsupervised learning untuk menemukan pola, melakukan klasifikasi, dan menghasilkan prediksi.
Sub-CPMK	: SUB-CPMK-2.1 — Mampu melakukan pengumpulan, pembersihan, transformasi, integrasi, seleksi fitur, dan pembagian dataset untuk proses pemodelan.
Lingkungan Praktikum	: Google Colab dengan Python
Library Utama	NumPy, dan Matplotlib

Sub-CPMK yang dicapai adalah:

Mahasiswa mampu mengidentifikasi masalah kualitas data dan menerapkan teknik pembersihan yang sesuai terhadap data hilang (missing value), data duplikat, inkonsistensi data, pencilan (outlier), dan kesalahan format sehingga menghasilkan dataset yang lebih bersih untuk digunakan pada tahap persiapan data berikutnya.

2. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- 1) mengidentifikasi permasalahan kualitas data pada dataset;
- 2) mendeteksi data hilang (missing value) dan menentukan teknik penanganan yang sesuai;
- 3) mengidentifikasi serta menangani data duplikat;
- 4) menemukan dan memperbaiki inkonsistensi data;
- 5) mengidentifikasi kesalahan format pada atribut dataset;
- 6) mendeteksi pencilan (outlier) pada data numerik menggunakan teknik yang sesuai;

- 7) menentukan tindakan yang tepat terhadap outlier berdasarkan karakteristik dan konteks data; dan
- 8) menghasilkan dataset yang lebih bersih dan terdokumentasi sebagai dasar untuk tahap persiapan data berikutnya.

Tujuan praktikum ini disusun sesuai indikator RPS pada pertemuan ke-4, yaitu ketepatan mahasiswa dalam menemukan masalah kualitas data dan menerapkan teknik pembersihan yang sesuai. Cakupan materinya meliputi penanganan data hilang, data duplikat, inkonsistensi, pencilan, dan kesalahan format.

3. Dasar Teori

2.1. Pembersihan Data

Data yang diperoleh dari proses pengumpulan belum tentu berada dalam kondisi yang siap digunakan untuk analisis maupun pemodelan. Dataset dapat mengandung nilai yang hilang, data yang tercatat berulang, perbedaan cara penulisan, format yang tidak sesuai, atau nilai yang jauh berbeda dari sebagian besar observasi lainnya. Kondisi tersebut dapat memengaruhi hasil analisis dan kualitas model yang dibangun.

Pembersihan data (data cleaning) merupakan proses mengidentifikasi dan memperbaiki permasalahan kualitas data agar data menjadi lebih akurat, konsisten, lengkap, dan dapat digunakan pada tahap analisis berikutnya. Dalam alur CRISP-DM, kegiatan pembersihan data merupakan bagian dari tahap Data Preparation. Pembersihan data bukan sekadar menghapus data yang dianggap bermasalah. Setiap tindakan perlu mempertimbangkan karakteristik atribut, konteks permasalahan, jumlah data yang terdampak, dan kemungkinan pengaruh perubahan tersebut terhadap hasil analisis.

Pada praktikum ini, pembersihan data difokuskan pada lima permasalahan utama, yaitu data hilang (missing value), data duplikat, inkonsistensi data, kesalahan format, dan pencilan (outlier).

2.2. Kualitas Data

Kualitas data menggambarkan sejauh mana data dapat digunakan secara tepat untuk tujuan analisis. Dataset yang memiliki banyak kesalahan dapat menghasilkan informasi

yang keliru dan selanjutnya memengaruhi keputusan yang diambil berdasarkan hasil analisis tersebut.

Beberapa aspek yang perlu diperhatikan dalam pemeriksaan kualitas data meliputi kelengkapan, konsistensi, validitas, dan keunikan data. Kelengkapan berkaitan dengan tersedia atau tidaknya nilai pada setiap atribut. Konsistensi berkaitan dengan keseragaman representasi data. Validitas menunjukkan apakah suatu nilai sesuai dengan aturan atau domain atribut, sedangkan keunikan berkaitan dengan keberadaan observasi yang tercatat lebih dari satu kali.

Sebagai contoh, sebuah dataset pelanggan dapat memiliki permasalahan seperti berikut:

Permasalahan	Contoh
Data hilang	Nilai usia atau pendapatan tidak tersedia
Data duplikat	Satu pelanggan tercatat dua kali
Inkonsistensi	Palopo, PALOPO, dan palopo
Kesalahan format	Tanggal ditulis dengan format berbeda
Outlier	Nilai transaksi jauh lebih besar dibandingkan mayoritas data

Permasalahan tersebut perlu diidentifikasi sebelum dataset digunakan pada tahapan berikutnya.

2.3. Data Hilang (*Missing Value*)

Missing value merupakan kondisi ketika suatu atribut tidak memiliki nilai pada satu atau beberapa observasi. Dalam dataset, nilai yang hilang dapat muncul karena data tidak diisi saat pengumpulan, kesalahan pencatatan, kegagalan sistem, proses integrasi data yang tidak sempurna, atau informasi memang tidak tersedia.

Dalam Python menggunakan pandas, nilai yang hilang biasanya direpresentasikan sebagai NaN (Not a Number) atau None. Namun, dalam dataset nyata, nilai hilang juga dapat ditulis menggunakan simbol atau teks tertentu seperti -, ?, N/A, null, atau bahkan berupa sel kosong. Oleh karena itu, pemeriksaan nilai hilang tidak hanya bergantung pada satu bentuk representasi.

Secara umum, penanganan missing value dapat dilakukan dengan dua pendekatan utama, yaitu menghapus data atau melakukan imputasi. Penghapusan dapat dilakukan terhadap baris atau kolom tertentu apabila jumlah data yang hilang cukup besar atau

atribut tersebut tidak memberikan informasi penting. Namun, penghapusan yang terlalu banyak dapat menyebabkan hilangnya informasi yang masih berguna.

Imputasi dilakukan dengan mengganti nilai yang hilang menggunakan nilai tertentu. Pada data numerik, nilai pengganti dapat berupa rata-rata (mean) atau median. Median biasanya lebih tahan terhadap pengaruh nilai ekstrem. Pada data kategorikal, nilai yang hilang dapat diganti menggunakan modus atau kategori tertentu berdasarkan konteks data. Pemilihan teknik tidak boleh dilakukan hanya karena suatu metode mudah diterapkan. Mahasiswa perlu mempertimbangkan distribusi data, proporsi nilai yang hilang, serta arti atribut tersebut dalam permasalahan yang sedang dianalisis.

2.4. Data Duplikat

Data duplikat terjadi ketika sebuah observasi tercatat lebih dari satu kali dalam dataset. Duplikasi dapat muncul akibat proses entri data yang berulang, penggabungan beberapa sumber data, atau kesalahan dalam proses penyimpanan.

Keberadaan data duplikat dapat memberikan bobot yang tidak semestinya terhadap observasi tertentu. Misalnya, apabila sebuah transaksi tercatat dua kali, nilai total transaksi yang diperoleh dari analisis dapat menjadi lebih besar daripada keadaan sebenarnya.

Meskipun demikian, tidak setiap baris yang memiliki nilai sama dapat langsung dianggap sebagai kesalahan. Dua pelanggan berbeda dapat mempunyai usia, jenis kelamin, dan kota yang sama. Demikian pula, dua transaksi dapat memiliki nilai yang sama tetapi terjadi pada waktu yang berbeda.

Karena itu, identifikasi duplikasi perlu memperhatikan atribut yang menjadi pembeda, seperti ID pelanggan, kode transaksi, waktu transaksi, atau atribut unik lainnya. Penghapusan dilakukan setelah dipastikan bahwa observasi tersebut benar-benar merupakan pencatatan berulang yang tidak diperlukan.

2.5. Inkonsistensi Data

Inkonsistensi terjadi ketika informasi yang seharusnya mempunyai makna yang sama direpresentasikan dengan cara yang berbeda. Kondisi ini banyak ditemukan pada atribut kategorikal atau data yang berasal dari beberapa sumber.

Sebagai contoh, nama kota yang sama dapat ditulis sebagai:

Palopo, palopo, PALOPO, atau Kota Palopo.

Pada atribut jenis kelamin dapat ditemukan nilai seperti:

L, Laki-laki, Pria, atau Male.

Walaupun nilai-nilai tersebut mungkin memiliki arti yang sama, komputer akan membacanya sebagai kategori yang berbeda apabila tidak dilakukan penyeragaman. Pembersihan terhadap data yang tidak konsisten dilakukan dengan menentukan representasi yang seragam. Proses ini dapat berupa penyeragaman penggunaan huruf besar dan kecil, penghilangan spasi yang tidak diperlukan, perbaikan salah ketik, atau penyatuan beberapa variasi kategori ke dalam satu kategori yang sama. Penyeragaman perlu dilakukan dengan memahami makna data agar kategori yang sebenarnya berbeda tidak digabungkan secara keliru.

2.6. Kesalahan Format Data

Kesalahan format terjadi ketika representasi atau tipe suatu nilai tidak sesuai dengan karakteristik atribut yang seharusnya. Permasalahan ini sering ditemukan pada data tanggal, angka, mata uang, nomor identitas, maupun atribut kategorikal.

Sebagai contoh, sebuah atribut tanggal dapat memiliki beberapa bentuk berikut:

12/08/2026, 2026-08-12, dan 12 Agustus 2026.

Contoh lain adalah nilai pendapatan yang tersimpan sebagai:

5000000, Rp5.000.000, atau 5 juta.

Bagi manusia, nilai tersebut dapat dipahami memiliki arti yang sama. Namun, dalam proses komputasi, perbedaan format dapat menyebabkan atribut dikenali dengan tipe data yang berbeda dan menyulitkan proses analisis. Pembersihan format dilakukan dengan menentukan bentuk representasi yang konsisten dan sesuai dengan tipe data yang diperlukan. Misalnya, data tanggal dikonversi menjadi tipe `datetime`, sedangkan data numerik dibersihkan dari simbol mata uang atau karakter lain sebelum dikonversi menjadi tipe numerik.

Pemeriksaan format juga perlu mempertimbangkan validitas nilai. Sebagai contoh, umur bernilai negatif atau persentase di atas batas yang diperbolehkan dapat menunjukkan adanya kesalahan data meskipun tipe datanya sudah numerik.

2.7. Pencarian (*Outlier*)

Outlier merupakan observasi yang memiliki nilai sangat berbeda dibandingkan sebagian besar observasi dalam dataset. Penculan dapat disebabkan oleh kesalahan pencatatan, kesalahan pengukuran, proses pengumpulan data, atau memang merupakan kejadian nyata yang jarang terjadi.

Sebagai contoh, apabila sebagian besar transaksi pelanggan berada pada rentang Rp100.000 sampai Rp2.000.000, kemudian terdapat sebuah transaksi sebesar Rp100.000.000, nilai tersebut dapat dianggap sebagai kandidat outlier. Namun, nilai tersebut belum tentu merupakan kesalahan. Transaksi tersebut mungkin benar-benar dilakukan oleh pelanggan tertentu.

Salah satu metode yang umum digunakan untuk mengidentifikasi outlier adalah Interquartile Range (IQR). IQR menggambarkan rentang antara kuartil pertama (Q_1) dan kuartil ketiga (Q_3).

$$IQR = Q_3 - Q_1$$

Batas bawah ditentukan dengan:

$$Batas\ Bawah = Q_1 - 1.5 \times IQR$$

sedangkan batas atas ditentukan dengan:

$$Batas\ Atas = Q_3 + 1.5 \times IQR$$

Observasi yang berada di bawah batas bawah atau di atas batas atas dapat dianggap sebagai kandidat outlier. Selain perhitungan statistik, keberadaan outlier dapat diamati secara visual menggunakan boxplot. Visualisasi membantu melihat posisi nilai ekstrem terhadap distribusi data secara keseluruhan.

Hal terpenting dalam menangani outlier adalah memahami bahwa tidak semua outlier harus dihapus. Sebelum mengambil tindakan, perlu diperiksa apakah nilai tersebut merupakan kesalahan atau justru observasi valid yang membawa informasi penting.

2.8. Prinsip Pengambilan Keputusan dalam Pembersihan Data

Proses pembersihan data harus dilakukan secara sistematis dan dapat dijelaskan. Mahasiswa perlu mengetahui kondisi dataset sebelum dilakukan perubahan, tindakan yang diterapkan, alasan pemilihan tindakan tersebut, serta kondisi dataset setelah proses pembersihan.

Secara umum, alur pembersihan data dalam praktikum ini adalah:



Gambar 2. Alur Pembersihan Data

Pendekatan tersebut penting karena tujuan pembersihan data bukan membuat dataset terlihat sempurna, melainkan menghasilkan data yang lebih valid, konsisten, dan sesuai dengan kebutuhan analisis. Setiap perubahan juga perlu didokumentasikan dalam notebook. Dokumentasi memungkinkan proses pembersihan ditelusuri kembali dan membantu memastikan bahwa dataset hasil pembersihan dapat dipertanggungjawabkan sebelum digunakan pada tahap transformasi dan pemodelan.

4. Alat dan Bahan

Praktikum pembersihan data dilaksanakan menggunakan lingkungan komputasi berbasis Python. Perangkat dan bahan yang digunakan adalah sebagai berikut.

a. Perangkat Keras

Mahasiswa menggunakan laptop atau komputer dengan spesifikasi minimal:

- 1) prosesor dual-core;
- 2) RAM minimal 4 GB, disarankan 8 GB;
- 3) ruang penyimpanan kosong minimal 5 GB; dan
- 4) koneksi internet yang stabil.

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam praktikum meliputi:

- 1) Google Colab, sebagai lingkungan utama untuk menulis dan menjalankan notebook Python;
- 2) Python, sebagai bahasa pemrograman untuk proses pembersihan dan analisis data;

- 3) pandas, untuk membaca, memeriksa, memanipulasi, dan membersihkan dataset;
- 4) NumPy, untuk mendukung pengolahan data numerik;
- 5) Matplotlib, untuk visualisasi sederhana terutama dalam membantu identifikasi pencilan (outlier);
- 6) LMS Edlink, untuk mengakses materi, tutorial, serta mengumpulkan notebook hasil praktikum.

Penggunaan perangkat tersebut sejalan dengan media pembelajaran yang tercantum dalam RPS, yaitu Python, Jupyter/Google Colab, pandas, NumPy, Matplotlib, dan LMS Edlink.

c. Bahan Praktikum

Bahan utama yang digunakan adalah dataset praktikum yang mengandung beberapa permasalahan kualitas data, seperti:

- 1) data hilang (missing value);
- 2) data duplikat;
- 3) inkonsistensi penulisan;
- 4) kesalahan format; dan
- 5) pencilan (outlier).

Dataset tersebut digunakan sebagai bahan latihan untuk melakukan proses identifikasi masalah, menentukan teknik pembersihan, menerapkan proses pembersihan, dan memeriksa kembali hasilnya.

d. File Kerja

Mahasiswa membuat notebook Google Colab dengan nama:

```
Modul_04_Pembersihan_Data_NIM_Nama.ipynb
```

Notebook digunakan untuk mendokumentasikan seluruh proses praktikum, mulai dari pemeriksaan awal data, identifikasi permasalahan kualitas data, penerapan teknik pembersihan, hingga evaluasi hasil.

Dataset hasil pembersihan disimpan dalam format:

```
dataset_bersih.csv
```

File notebook dan dataset hasil pembersihan digunakan sebagai bukti proses praktikum dan dikumpulkan melalui LMS Edlink sesuai ketentuan yang berlaku.

5. Langkah Praktikum

Pada bagian ini, mahasiswa melakukan proses pembersihan data secara bertahap menggunakan Google Colab. Praktikum mengikuti pola memeriksa data → menemukan masalah kualitas data → memahami penyebab masalah → menentukan teknik penanganan → menerapkan pembersihan → memeriksa kembali hasilnya.

RPS menetapkan bahwa kegiatan pada pertemuan ini dilaksanakan melalui demonstrasi, guided practice, dan pemecahan masalah, dengan fokus pada data hilang, data duplikat, inkonsistensi, pencilan, dan kesalahan format.

a. Menyiapkan Notebook Google Colab

Buka Google Colab, kemudian buat notebook baru dengan nama:

```
Modul_04_Pembersihan_Data_NIM_Nama.ipynb
```

Pada sel pertama, tuliskan identitas praktikum dalam bentuk Markdown, meliputi:

- 1) nama;
- 2) NIM;
- 3) kelas;
- 4) mata kuliah; dan
- 5) judul praktikum.

Selanjutnya, panggil library yang akan digunakan.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

b. Memuat Dataset

Unggah dataset praktikum ke Google Colab, kemudian baca dataset menggunakan pandas.

```
df = pd.read_csv('/content/dataset.csv')
```

Tampilkan lima baris pertama.

```
df.head()
```

Periksa ukuran dataset.

```
df.shape
```

Tampilkan informasi struktur dataset.

```
df.info()
```

Periksa statistik deskriptif awal.

```
df.describe(include='all')
```

Pada tahap ini, mahasiswa belum melakukan perubahan terhadap dataset. Tujuannya adalah memperoleh gambaran awal mengenai struktur dan kondisi data. Catatan Analisis:

Tuliskan jumlah baris, jumlah kolom, tipe data setiap atribut, serta temuan awal mengenai kemungkinan masalah kualitas data.

c. Membuat Salinan Dataset

Sebelum melakukan pembersihan, buat salinan dataset agar data awal tetap tersedia.

```
df_awal = df.copy()
```

Salinan ini digunakan untuk membandingkan kondisi dataset sebelum dan sesudah proses pembersihan.

d. Memeriksa Data Hilang

Periksa jumlah nilai yang hilang pada setiap kolom.

```
df.isnull().sum()
```

Hitung persentase nilai hilang.

```
missing = pd.DataFrame({
    'Jumlah Missing': df.isnull().sum(),
    'Persentase (%)': df.isnull().sum() / len(df) * 100
})
missing
```

Jika dataset menggunakan simbol tertentu sebagai penanda nilai kosong, periksa pula nilai unik pada kolom yang dicurigai.

```
df['nama_kolom'].unique()
```

Analisis:

Identifikasi kolom yang memiliki missing value. Jelaskan kemungkinan penyebabnya dan apakah nilai tersebut perlu dihapus atau diimputasi.

e. Menangani Data Hilang

Penanganan dilakukan berdasarkan karakteristik atribut.

Untuk data numerik, salah satu pendekatan yang dapat digunakan adalah median.

```
df['nama_kolom'] = df['nama_kolom'].fillna(
    df['nama_kolom'].median()
)
```

Untuk data kategorikal, dapat digunakan modus.

```
df['nama_kategori'] = df['nama_kategori'].fillna(
    df['nama_kategori'].mode()[0]
)
```

Jika terdapat baris yang memang tidak layak dipertahankan, baris tersebut dapat dihapus.

```
df = df.dropna(subset=['nama_kolom'])
```

Setelah itu, periksa kembali nilai hilang.

```
df.isnull().sum()
```

Analisis:

Jelaskan metode yang dipilih dan alasan penggunaannya. Jangan hanya menuliskan bahwa nilai kosong telah diisi.

f. Memeriksa Data Duplikat

Hitung jumlah baris duplikat.

```
df.duplicated().sum()
```

Tampilkan baris duplikat.

```
df[df.duplicated()]
```

Jika dataset memiliki atribut identitas tertentu, pemeriksaan dapat dilakukan berdasarkan atribut tersebut.

```
df[df.duplicated(subset=['id'], keep=False)]
```

Sebelum menghapus data, pastikan bahwa baris tersebut memang merupakan duplikasi yang tidak diperlukan.

g. Menangani Data Duplikat

Jika duplikasi telah dipastikan sebagai kesalahan, hapus menggunakan:

```
df = df.drop_duplicates()
```

Periksa kembali.

```
df.duplicated().sum()
```

Bandingkan ukuran dataset.

```
print('Ukuran awal :', df_awal.shape)
print('Ukuran saat ini:', df.shape)
```

Analisis:

Tuliskan jumlah data duplikat yang ditemukan dan jumlah observasi yang dihapus.

h. Memeriksa Inkonsistensi Data

Identifikasi atribut kategorikal.

```
df.select_dtypes(include='object').columns
```

Periksa variasi nilai kategori.

```
df['nama_kategori'].value_counts(dropna=False)
```

Amati apakah terdapat penulisan yang berbeda tetapi memiliki makna yang sama, misalnya:

Palopo, palopo, PALOPO

atau:

L, Laki-laki, Pria

Gunakan pula:

```
df['nama_kategori'].unique()
```

untuk melihat seluruh variasi nilai.

i. Menangani Inkonsistensi Data

Salah satu cara paling sederhana adalah menstandarkan format teks.

```
df['nama_kategori'] = (
    df['nama_kategori']
    .str.strip()
    .str.lower()
)
```

Jika terdapat beberapa nilai yang berbeda tetapi memiliki arti sama, lakukan pemetaan.

```
pemetaan = {
    'l': 'laki-laki',
    'pria': 'laki-laki',
    'male': 'laki-laki',
    'p': 'perempuan',
    'wanita': 'perempuan',
    'female': 'perempuan'
}

df['jenis_kelamin'] = df['jenis_kelamin'].replace(pemetaan)
```

Periksa kembali hasilnya.

```
df['jenis_kelamin'].value_counts()
```

Analisis:

Tuliskan bentuk inkonsistensi yang ditemukan dan perubahan yang dilakukan.

j. Memeriksa Kesalahan Format

Periksa tipe data setiap atribut.

```
df.dtypes
```

Perhatikan apakah atribut numerik masih terbaca sebagai object, atau atribut tanggal belum dikenali sebagai datetime.

Sebagai contoh, periksa kolom tanggal.

```
df['tanggal'].head()
```

Jika diperlukan, ubah menjadi tipe tanggal.

```
df['tanggal'] = pd.to_datetime(
    df['tanggal'],
    errors='coerce'
)
```

Untuk data numerik yang masih mengandung simbol atau karakter lain, bersihkan terlebih dahulu.

```
df['pendapatan'] = (
    df['pendapatan']
    .astype(str)
    .str.replace('Rp', '', regex=False)
    .str.replace('.', '', regex=False)
    .str.strip()
)
```

Kemudian ubah menjadi numerik.

```
df['pendapatan'] = pd.to_numeric(
    df['pendapatan'],
    errors='coerce'
)
```

Periksa kembali tipe data.

```
df.dtypes
```

Analisis:

Identifikasi kolom yang memiliki masalah format dan jelaskan perubahan tipe data yang dilakukan.

k. Memeriksa Nilai Tidak Valid

Selain format, periksa apakah terdapat nilai yang secara logis tidak sesuai.

Contoh:

```
df[df['umur'] < 0]
```

atau:

```
df[df['umur'] > 100]
```

Untuk atribut lain, batas valid disesuaikan dengan konteks dataset.

Jika nilai tidak valid ditemukan, tentukan apakah data perlu diperbaiki, dihapus, atau diperlakukan sebagai nilai hilang.

l. Mengidentifikasi Outlier dengan Boxplot

Pilih atribut numerik yang relevan.

```
plt.figure(figsize=(8, 4))
plt.boxplot(df['nama_kolom'].dropna())
plt.title('Boxplot Nama Kolom')
plt.ylabel('Nilai')
plt.show()
```

Amati apakah terdapat titik yang berada jauh dari sebagian besar data.

Visualisasi hanya digunakan sebagai indikasi awal. Keputusan terhadap outlier tidak boleh hanya berdasarkan tampilan grafik.

m. Mengidentifikasi Outlier dengan IQR

Hitung kuartil pertama dan ketiga.

```
Q1 = df['nama_kolom'].quantile(0.25)
Q3 = df['nama_kolom'].quantile(0.75)

IQR = Q3 - Q1
```

Hitung batas bawah dan batas atas.

```
batas_bawah = Q1 - 1.5 * IQR
batas_atas = Q3 + 1.5 * IQR

print('Q1          : ', Q1)
print('Q3          : ', Q3)
print('IQR         : ', IQR)
print('Batas bawah : ', batas_bawah)
print('Batas atas  : ', batas_atas)
```

Tampilkan kandidat outlier.

```
outlier = df[
    (df['nama_kolom'] < batas_bawah) |
    (df['nama_kolom'] > batas_atas)
]
outlier
```

Hitung jumlahnya.

```
len(outlier)
```

n. Menentukan Tindakan terhadap Outlier

Setelah kandidat outlier ditemukan, jangan langsung menghapusnya.

Periksa terlebih dahulu:

- 1) apakah nilainya masih masuk akal;
- 2) apakah terdapat indikasi salah input;
- 3) apakah nilai tersebut merupakan kejadian yang memang valid; dan
- 4) apakah keberadaannya relevan terhadap tujuan analisis.

Jika outlier dipastikan berasal dari kesalahan data, observasi dapat dihapus.

```
df = df[
    (df['nama_kolom'] >= batas_bawah) &
    (df['nama_kolom'] <= batas_atas)
]
```

Namun, apabila nilai tersebut valid, data dapat tetap dipertahankan.

Analisis:

Tuliskan jumlah outlier, kemungkinan penyebabnya, dan keputusan yang diambil.

o. Memeriksa Kembali Dataset

Setelah seluruh proses pembersihan selesai, lakukan pemeriksaan ulang.

```
df.info()
```

Periksa nilai hilang.

```
df.isnull().sum()
```

Periksa data duplikat.

```
df.duplicated().sum()
```

Periksa statistik deskriptif.

```
df.describe(include='all')
```

Tujuannya adalah memastikan bahwa proses pembersihan tidak menimbulkan masalah baru.

p. Membandingkan Data Sebelum dan Sesudah

Buat ringkasan hasil pembersihan.

Aspek	Sebelum Pembersihan	Sesudah Pembersihan
Jumlah baris	...	...
Jumlah kolom	...	...
Missing value	...	...
Data duplikat	...	...
Inkonsistensi kategori	...	...
Kesalahan format	...	...
Outlier yang ditangani	...	...

Selanjutnya, tuliskan interpretasi singkat mengenai perubahan kualitas dataset setelah proses pembersihan.

q. Menyimpan Dataset Bersih

Simpan dataset hasil pembersihan.

```
df.to_csv(
    'dataset_bersih.csv',
    index=False
)
```

Pastikan file berhasil dibuat dan dapat digunakan kembali pada praktikum selanjutnya.

r. Menuliskan Kesimpulan Praktikum

Pada bagian akhir notebook, tuliskan kesimpulan singkat yang menjawab tiga hal berikut:

- 1) masalah kualitas data apa saja yang ditemukan;
- 2) teknik pembersihan apa yang diterapkan; dan
- 3) bagaimana kondisi dataset setelah proses pembersihan.

Kesimpulan harus berdasarkan hasil praktikum, bukan sekadar mengulang teori.

Hasil utama praktikum adalah notebook yang terdokumentasi dengan baik dan dataset hasil pembersihan, sesuai dengan kriteria pada RPS yang menilai kebenaran prosedur, kualitas dataset hasil pembersihan, dan dokumentasi kode.

6. Latihan Praktikum

Latihan praktikum bertujuan memperkuat kemampuan mahasiswa dalam mengidentifikasi permasalahan kualitas data dan menentukan teknik pembersihan yang sesuai. Pada bagian ini, mahasiswa bekerja lebih mandiri berdasarkan konsep dan langkah yang telah dipraktikkan pada bagian sebelumnya.

Gunakan dataset yang disediakan, kemudian lakukan kegiatan berikut.

a. Pemeriksaan Awal Dataset

Lakukan pemeriksaan terhadap dataset untuk mengetahui:

- 1) jumlah baris dan kolom;
- 2) nama dan tipe data setiap atribut;
- 3) jumlah nilai yang hilang;
- 4) kemungkinan adanya data duplikat;
- 5) variasi nilai pada atribut kategorikal; dan
- 6) statistik deskriptif atribut numerik.

Tuliskan minimal tiga temuan awal mengenai kualitas dataset.

b. Analisis Data Hilang

Identifikasi seluruh atribut yang memiliki *missing value*.

Untuk setiap atribut yang memiliki nilai hilang, lengkapi tabel berikut.

Atribut	Jumlah Missing	Persentase	Teknik Penanganan	Alasan
...	...	...	...	...
...	...	...	...	...

Terapkan teknik penanganan yang telah dipilih, kemudian periksa kembali kondisi *missing value*.

c. Analisis Data Duplikat

Periksa apakah terdapat data duplikat pada dataset.

Jika ditemukan, tentukan apakah baris tersebut benar-benar merupakan duplikasi berdasarkan atribut yang tersedia.

Tuliskan:

- 1) jumlah data duplikat yang ditemukan;
- 2) atribut yang digunakan untuk menentukan duplikasi;
- 3) keputusan terhadap data tersebut; dan
- 4) jumlah data setelah proses penanganan.

d. Analisis Inkonsistensi Data

Pilih minimal dua atribut kategorikal, kemudian periksa seluruh variasi nilainya.

Identifikasi nilai yang memiliki arti sama tetapi ditulis dalam bentuk berbeda.

Contoh:

Palopo, PALOPO, dan palopo

Tentukan bentuk penulisan yang akan digunakan sebagai standar, kemudian lakukan penyeragaman.

Dokumentasikan kondisi data sebelum dan sesudah dilakukan pembersihan.

e. Analisis Kesalahan Format

Periksa tipe data seluruh atribut dan identifikasi atribut yang formatnya belum sesuai.

Pilih minimal satu atribut yang memerlukan perbaikan format, misalnya:

- 1) tanggal;
- 2) nilai numerik yang tersimpan sebagai teks;
- 3) nilai mata uang;
- 4) persentase; atau
- 5) atribut lain yang sesuai dengan dataset.

Lakukan perbaikan kemudian periksa kembali tipe datanya.

Jelaskan alasan perubahan format yang dilakukan.

f. Analisis Nilai Tidak Valid

Identifikasi kemungkinan nilai yang tidak masuk akal berdasarkan konteks atribut.

Contohnya:

- 1) usia bernilai negatif;
- 2) jumlah transaksi negatif;
- 3) persentase berada di luar rentang yang semestinya; atau
- 4) nilai lain yang melanggar aturan logis dataset.

Jika ditemukan, jelaskan keputusan yang diambil terhadap nilai tersebut.

g. Analisis Outlier

Pilih minimal **dua atribut numerik**.

Untuk setiap atribut:

1. buat *boxplot*;
2. hitung (Q_1), (Q_3), dan IQR;
3. tentukan batas bawah dan batas atas;
4. identifikasi jumlah kandidat *outlier*; dan
5. tampilkan observasi yang teridentifikasi sebagai *outlier*.

Lengkapi ringkasan berikut.

Atribut	Q_1	Q_3	IQR	Jumlah Outlier	Keputusan
...	...	...	...	...	...
...	...	...	...	...	...

Jelaskan apakah *outlier* tersebut dipertahankan atau ditangani. Keputusan harus didasarkan pada konteks data, bukan hanya karena nilainya berada di luar batas IQR.

h. Evaluasi Hasil Pembersihan

Setelah seluruh proses selesai, bandingkan kondisi dataset sebelum dan sesudah pembersihan.

Indikator	Sebelum	Sesudah
Jumlah observasi	...	...
Missing value	...	...
Data duplikat	...	...
Inkonsistensi yang ditemukan	...	...
Kesalahan format	...	...
Nilai tidak valid	...	...
Outlier yang ditangani	...	...

Berdasarkan hasil tersebut, tuliskan **3–5 kalimat analisis** mengenai perubahan kualitas dataset.

i. Refleksi

Jawab secara singkat pertanyaan berikut.

- 1) Permasalahan kualitas data apa yang paling banyak ditemukan?
- 2) Teknik pembersihan apa yang paling banyak digunakan?
- 3) Apakah ada data yang sengaja dipertahankan meskipun teridentifikasi sebagai outlier? Jelaskan.
- 4) Apa risiko jika proses pembersihan dilakukan tanpa memahami konteks data?
- 5) Apakah dataset hasil pembersihan sudah cukup layak untuk dilanjutkan ke tahap transformasi data? Jelaskan alasannya.

Latihan ini menekankan ketepatan menemukan masalah kualitas data dan menerapkan teknik pembersihan yang sesuai, sebagaimana indikator pembelajaran pada pertemuan ke-4 dalam RPS.

7. Tugas Praktikum

a. Deskripsi Tugas

Lakukan proses pembersihan data (data cleaning) terhadap dataset yang diberikan menggunakan Google Colab. Mahasiswa diminta mengidentifikasi permasalahan kualitas data, menentukan teknik penanganan yang tepat, menerapkan proses pembersihan, dan mengevaluasi kondisi dataset setelah dibersihkan.

Pengerjaan harus mengikuti alur:

memeriksa data → menemukan masalah kualitas data → memahami penyebab masalah → menentukan teknik penanganan → menerapkan pembersihan → memeriksa kembali hasilnya.

Setiap tindakan pembersihan harus disertai alasan. Hindari menghapus atau mengubah data hanya karena suatu nilai terlihat berbeda tanpa terlebih dahulu memahami konteks dan karakteristik atribut.

b. Ketentuan Pengerjaan

Gunakan dataset yang diberikan pada praktikum. Notebook yang dikumpulkan harus memuat proses berikut:

- a. pemeriksaan awal struktur dan karakteristik dataset;
- b. identifikasi data hilang (missing value);
- c. penanganan data hilang menggunakan teknik yang sesuai;
- d. pemeriksaan dan penanganan data duplikat;
- e. identifikasi dan penanganan inkonsistensi data;
- f. pemeriksaan serta perbaikan kesalahan format;
- g. pemeriksaan nilai yang tidak valid;
- h. identifikasi outlier menggunakan boxplot dan metode IQR;
- i. keputusan penanganan outlier berdasarkan konteks data;
- j. pemeriksaan kembali dataset setelah proses pembersihan;
- k. perbandingan kondisi dataset sebelum dan sesudah pembersihan; dan
- l. penyimpanan dataset hasil pembersihan.

Kode harus disertai penjelasan menggunakan sel Markdown sehingga alur pengerjaan dapat dipahami dan ditelusuri kembali.

c. Dokumentasi Temuan

Buat tabel ringkasan permasalahan kualitas data yang ditemukan.

No.	Permasalahan	Atribut/ Bagian Data	Jumlah Data Terdampak	Tindakan	Alasan
1	Missing value	...	...	...	...
2	Data duplikat	...	...	...	...
3	Inkonsistensi	...	...	...	...
4	Kesalahan format	...	...	...	...
5	Nilai tidak valid	...	...	...	...
6	Outlier	...	...	...	...

Apabila salah satu jenis permasalahan tidak ditemukan pada dataset, tuliskan **“Tidak ditemukan”** dan sertakan bukti hasil pemeriksaan.

d. Perbandingan Dataset

Setelah proses pembersihan selesai, buat tabel perbandingan kondisi dataset.

Indikator	Dataset Awal	Dataset Bersih	Perubahan
Jumlah baris	...	...	...
Jumlah kolom	...	...	...
Missing value	...	...	...
Data duplikat	...	...	...
Inkonsistensi data	...	...	...
Kesalahan format	...	...	...
Nilai tidak valid	...	...	...
Outlier yang ditangani	...	...	...

Berikan interpretasi singkat terhadap perubahan yang terjadi. Jelaskan apakah proses pembersihan meningkatkan kualitas dataset dan apakah terdapat informasi yang hilang akibat tindakan pembersihan.

e. Kesimpulan

Pada bagian akhir notebook, buat kesimpulan dalam bentuk narasi singkat yang mencakup:

- 1) kondisi awal dataset;
- 2) permasalahan kualitas data yang paling dominan;
- 3) teknik pembersihan yang digunakan;
- 4) keputusan penting yang diambil selama proses pembersihan; dan
- 5) kondisi dataset setelah pembersihan.

Kesimpulan tidak cukup hanya menyatakan bahwa data telah dibersihkan. Jelaskan mengapa dataset hasil pembersihan dianggap lebih layak untuk digunakan pada tahap selanjutnya.

f. Berkas yang Dikumpulkan

Mahasiswa mengumpulkan dua berkas:

- 1) notebook Google Colab dengan nama:
`Modul_04_Pembersihan_Data_NIM_Nama.ipynb`
- 2) dataset hasil pembersihan dengan nama:
`dataset_bersih_NIM.csv`

Pastikan notebook dapat dijalankan kembali tanpa kesalahan dan seluruh keluaran penting telah ditampilkan sebelum dikumpulkan.

g. Pengumpulan

Notebook dan dataset hasil pembersihan dikumpulkan melalui LMS Edlink sesuai jadwal yang telah ditentukan.

Penilaian mengacu pada indikator RPS, yaitu ketepatan menemukan masalah kualitas data dan menerapkan teknik pembersihan yang sesuai. Kualitas pekerjaan terutama dilihat dari kebenaran prosedur, kualitas dataset hasil pembersihan, dan dokumentasi kode.

8. Pertanyaan Evaluasi

Jawab pertanyaan berikut berdasarkan konsep yang telah dipelajari dan hasil praktikum yang dilakukan.

- 1) Apa yang dimaksud dengan pembersihan data (data cleaning), dan mengapa tahap ini penting sebelum data digunakan untuk proses pemodelan?
- 2) Sebutkan lima jenis permasalahan kualitas data yang dipelajari dalam praktikum ini.
- 3) Apa yang dimaksud dengan missing value? Sebutkan beberapa kemungkinan penyebab munculnya missing value dalam sebuah dataset.
- 4) Jelaskan perbedaan antara menghapus data yang memiliki missing value dan melakukan imputasi. Dalam kondisi apa masing-masing pendekatan dapat digunakan?
- 5) Mengapa penggunaan median dapat lebih sesuai daripada rata-rata (mean) untuk mengganti missing value pada data numerik tertentu?
- 6) Mengapa data duplikat perlu diperiksa sebelum dihapus? Berikan contoh kondisi ketika dua baris yang terlihat sama belum tentu merupakan data duplikat.
- 7) Perhatikan nilai kategori berikut:
Palopo, PALOPO, palopo, Palopo
Apakah kondisi tersebut termasuk permasalahan kualitas data? Jelaskan bagaimana cara menanganinya.
- 8) Apa perbedaan antara **inkonsistensi data** dan **kesalahan format data**? Berikan masing-masing satu contoh.
- 9) Sebuah atribut pendapatan memiliki nilai berikut:
Rp 2.500.000, 3000000, Rp 4.000.000, 5000000
Jelaskan permasalahan yang terdapat pada data tersebut dan langkah yang dapat dilakukan agar atribut dapat dianalisis sebagai data numerik.
- 10) Mengapa pemeriksaan tipe data (`dtypes`) penting dalam proses pembersihan data?
- 11) Apa yang dimaksud dengan nilai **tidak valid**? Berikan dua contoh nilai yang secara teknis dapat tersimpan dalam dataset tetapi tidak masuk akal berdasarkan konteksnya.
- 12) Apa yang dimaksud dengan **outlier**? Apakah setiap outlier menunjukkan kesalahan data? Jelaskan.
- 13) Jelaskan fungsi (Q_1), (Q_3), dan IQR dalam proses identifikasi outlier.

14) Jika diketahui:

$$Q_1 = 20$$

$$Q_3 = 40$$

hitung:

- a. nilai IQR;
 - b. batas bawah; dan
 - c. batas atas untuk mendeteksi outlier menggunakan aturan ($1.5 \times \text{IQR}$).
- 15) Sebuah nilai berada di luar batas IQR, tetapi setelah diperiksa ternyata nilai tersebut merupakan transaksi bisnis yang benar-benar terjadi. Apa tindakan yang sebaiknya dilakukan? Jelaskan alasannya.
- 16) Mengapa keputusan untuk menghapus, memperbaiki, atau mempertahankan suatu data harus mempertimbangkan konteks data, bukan hanya hasil perhitungan statistik?
- 17) Mengapa dataset perlu diperiksa kembali setelah seluruh proses pembersihan selesai?
- 18) Apa manfaat menyimpan salinan dataset awal sebelum melakukan proses pembersihan?
- 19) Mengapa setiap proses pembersihan data perlu didokumentasikan dalam notebook?
- 20) Berdasarkan hasil praktikum yang telah dilakukan, jelaskan secara singkat:
- a. masalah kualitas data yang paling dominan ditemukan;
 - b. teknik pembersihan yang digunakan;
 - c. perubahan utama pada dataset setelah dibersihkan; dan
 - d. apakah dataset hasil pembersihan sudah layak dilanjutkan ke tahap persiapan data berikutnya.

Pertanyaan Reflektif

Tuliskan satu paragraf singkat untuk menjawab pertanyaan berikut:

“Apa risiko yang dapat terjadi apabila seorang analis langsung menghapus setiap data yang dianggap bermasalah tanpa memahami penyebab dan konteks data tersebut?”

Jawaban harus dikaitkan dengan pengalaman selama melakukan praktikum pembersihan data.

Pertanyaan evaluasi ini mendukung indikator pembelajaran pada pertemuan ke-4 dalam RPS, yaitu kemampuan menemukan masalah kualitas data dan menerapkan teknik pembersihan secara tepat.

9. Kesimpulan Praktikum

Melalui praktikum ini, mahasiswa telah mempelajari proses pembersihan data sebagai bagian penting dalam tahap Data Preparation. Proses pembersihan dilakukan dengan cara mengidentifikasi permasalahan kualitas data, memahami penyebabnya, menentukan teknik penanganan yang sesuai, menerapkan pembersihan, dan memeriksa kembali hasilnya.

Permasalahan kualitas data yang ditangani dalam praktikum ini meliputi:

- 1) data hilang (missing value);
- 2) data duplikat;
- 3) inkonsistensi data;
- 4) kesalahan format;
- 5) nilai tidak valid; dan
- 6) pencilan (outlier).

Mahasiswa juga telah menggunakan pandas, NumPy, dan Matplotlib dalam Google Colab untuk memeriksa, membersihkan, dan mendokumentasikan perubahan pada dataset.

Hasil akhir praktikum berupa dataset yang lebih bersih, konsisten, dan layak digunakan pada tahap persiapan data berikutnya. Selain menghasilkan dataset yang telah dibersihkan, mahasiswa diharapkan mampu menjelaskan alasan setiap tindakan yang dilakukan sehingga proses pembersihan dapat dipertanggungjawabkan.

Capaian ini sesuai dengan fokus pertemuan ke-4 pada RPS, yaitu kemampuan menemukan masalah kualitas data dan menerapkan teknik pembersihan yang sesuai terhadap data hilang, data duplikat, inkonsistensi, pencilan, dan kesalahan format.

Referensi

1. Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann–Elsevier.
2. Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
3. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O'Reilly Media.
4. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
5. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
6. McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly Media.
7. Dokumentasi resmi Python, pandas, Matplotlib, Jupyter, dan scikit-learn digunakan sebagai referensi teknis dalam pelaksanaan praktikum.

MODUL 5

TRANSFORMASI DATA DAN PERSIAPAN DATA UNTUK PEMODELAN

1. Identitas Praktikum

Judul Praktikum	: Transformasi Data dan Persiapan Data untuk Pemodelan
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SKS
CPMK	: CPMK-2 — Mampu menyiapkan data dan menerapkan algoritma supervised serta unsupervised learning untuk
Sub-CPMK	: Sub-CPMK-2.1 — Mampu melakukan pengumpulan, pembersihan, transformasi, integrasi, seleksi fitur, dan pembagian dataset untuk proses pemodelan.
Lingkungan Praktikum	: Google Colab / Jupyter Notebook
Library Utama	: pandas, NumPy, scikit-learn

Sub-CPMK yang Dicapai

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu menerapkan tahapan transformasi data untuk menghasilkan dataset yang siap digunakan dalam proses pemodelan machine learning. Kegiatan praktikum mencakup pengubahan data kategorikal menjadi representasi numerik, normalisasi dan standarisasi data numerik, pemilihan fitur yang relevan, serta pembagian dataset menjadi data latih dan data uji secara tepat.

Praktikum ini diarahkan agar mahasiswa tidak hanya mampu menjalankan fungsi transformasi data menggunakan Python, tetapi juga mampu menjelaskan alasan pemilihan teknik transformasi berdasarkan karakteristik dataset dan kebutuhan algoritma yang akan digunakan.

2. Tujuan Praktikum

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

- 1) menjelaskan tujuan transformasi data dalam tahapan persiapan data untuk pemodelan machine learning;

- 2) mengidentifikasi atribut numerik dan kategorikal yang memerlukan transformasi;
- 3) menerapkan teknik encoding untuk mengubah data kategorikal menjadi bentuk numerik yang sesuai;
- 4) menerapkan teknik normalisasi dan standardisasi pada fitur numerik sesuai karakteristik data dan kebutuhan pemodelan;
- 5) melakukan pemilihan fitur yang relevan untuk mengurangi atribut yang tidak diperlukan dalam proses pemodelan;
- 6) membagi dataset menjadi data latih (training set) dan data uji (test set) secara tepat;
- 7) memeriksa kembali hasil transformasi untuk memastikan dataset telah berada dalam bentuk yang siap digunakan pada proses pemodelan;
- 8) menjelaskan alasan pemilihan teknik transformasi yang digunakan berdasarkan karakteristik dataset.

Melalui praktikum ini, mahasiswa diharapkan tidak hanya mampu menjalankan fungsi transformasi menggunakan Python dan scikit-learn, tetapi juga memahami bahwa setiap teknik transformasi harus dipilih berdasarkan jenis data, tujuan analisis, dan kebutuhan algoritma machine learning yang akan digunakan.

3. Dasar Teori

3.1. Transformasi Data

Transformasi data merupakan tahap praproses yang bertujuan mengubah bentuk atau representasi data agar sesuai dengan kebutuhan proses analisis dan pemodelan machine learning. Data hasil pembersihan belum tentu dapat langsung digunakan oleh algoritma karena masih mungkin mengandung atribut kategorikal, perbedaan skala nilai yang besar, atau fitur yang tidak relevan.

Pada tahap ini, data dipersiapkan agar memiliki representasi yang lebih sesuai untuk proses pembelajaran model. Transformasi yang dilakukan dapat mencakup encoding data kategorikal, normalisasi, standardisasi, seleksi fitur, reduksi fitur, serta pembagian dataset menjadi data latih dan data uji. Materi tersebut sesuai dengan cakupan pertemuan ke-5 pada RPS.

3.2. Encoding Data Kategorikal

Sebagian besar algoritma machine learning bekerja menggunakan nilai numerik. Oleh karena itu, atribut kategorikal perlu diubah menjadi representasi angka sebelum digunakan dalam proses pemodelan.

Beberapa teknik yang umum digunakan adalah:

- a. Label Encoding, yaitu memberikan kode numerik pada setiap kategori.
- b. One-Hot Encoding, yaitu mengubah setiap kategori menjadi atribut baru yang berisi nilai biner 0 atau 1.

Pemilihan teknik encoding perlu memperhatikan karakteristik atribut. Label encoding lebih sesuai ketika kategori memiliki urutan tertentu, sedangkan one-hot encoding lebih sesuai untuk kategori yang tidak memiliki hubungan ordinal.

3.3. Normalisasi Data

Normalisasi digunakan untuk mengubah rentang nilai suatu fitur ke dalam skala tertentu. Salah satu teknik yang umum digunakan adalah Min-Max Scaling, yang mengubah nilai ke rentang 0 sampai 1.

Secara umum, normalisasi dapat dinyatakan sebagai:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

dengan:

- x = nilai asli,
- $x_{\min}$ = nilai minimum,
- $x_{\max}$ = nilai maksimum,
- x' = nilai setelah normalisasi.

Normalisasi membantu mengurangi pengaruh perbedaan skala antarfitur, khususnya pada algoritma yang menggunakan perhitungan jarak.

3.4. Standardisasi Data

Standardisasi mengubah nilai suatu fitur sehingga memiliki rata-rata mendekati 0 dan standar deviasi mendekati 1. Transformasi ini umumnya dilakukan menggunakan z-score:

$$z = \frac{x-\mu}{\sigma}]$$

dengan:

- x = nilai asli,
- μ = rata-rata fitur,
- σ = standar deviasi,
- z = nilai setelah standardisasi.

Berbeda dengan normalisasi, standardisasi tidak membatasi nilai ke rentang tertentu. Teknik ini digunakan ketika skala antarfitur perlu diseragamkan tanpa mempertahankan batas minimum dan maksimum tertentu.

3.5. Seleksi Fitur

Seleksi fitur merupakan proses memilih atribut yang paling relevan terhadap tujuan pemodelan. Fitur yang tidak relevan, redundan, atau tidak memberikan informasi yang berarti dapat mengurangi efisiensi proses pembelajaran model.

Seleksi fitur bertujuan untuk:

- 1) mengurangi kompleksitas dataset;
- 2) mempercepat proses pelatihan model;
- 3) mengurangi kemungkinan model mempelajari informasi yang tidak relevan; dan
- 4) meningkatkan kemudahan interpretasi hasil pemodelan.

Dalam praktikum, mahasiswa perlu membedakan antara atribut yang digunakan sebagai fitur (X) dan atribut yang digunakan sebagai target (y).

3.6. Reduksi Fitur

Selain memilih fitur tertentu, jumlah dimensi data dapat dikurangi dengan teknik reduksi fitur. Reduksi fitur digunakan ketika dataset memiliki banyak atribut dan sebagian informasi dapat direpresentasikan dalam jumlah dimensi yang lebih sedikit.

Pada Modul 5, konsep reduksi fitur diperkenalkan sebagai bagian dari persiapan dataset. Implementasinya dapat dibatasi pada pemahaman dasar agar fokus utama praktikum tetap pada transformasi dan persiapan data sebelum pemodelan.

3.7. Pembagian Dataset

Dataset yang telah ditransformasi perlu dipisahkan menjadi data yang digunakan untuk melatih model dan data yang digunakan untuk menguji kemampuan model pada data yang tidak digunakan selama proses pelatihan.

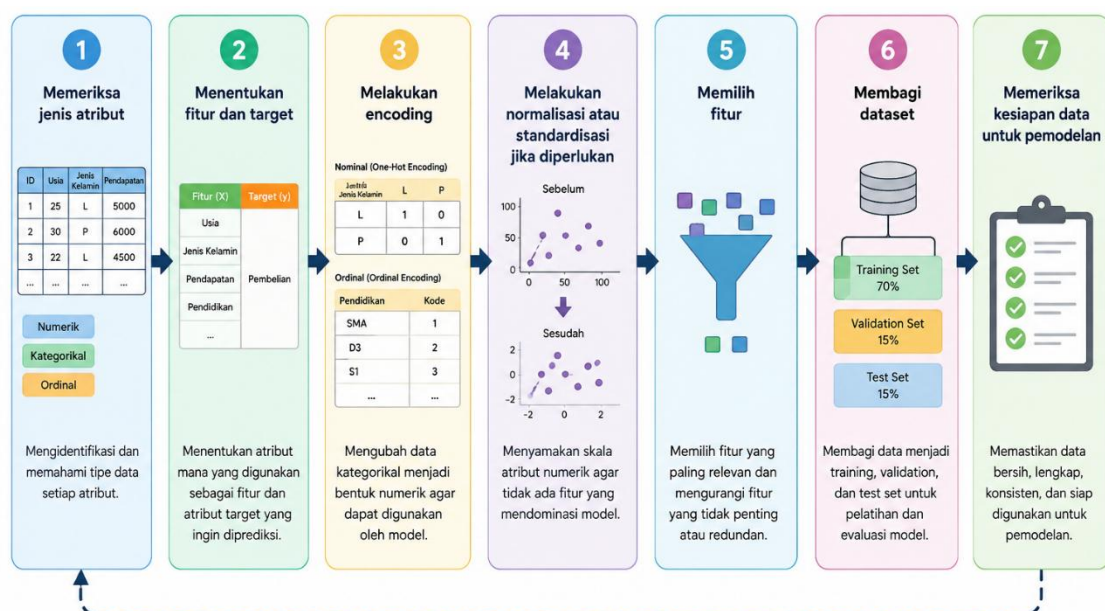
Pembagian utama terdiri atas:

- 1) Training set, yaitu bagian dataset yang digunakan untuk melatih model.
- 2) Test set, yaitu bagian dataset yang digunakan untuk menguji kinerja model setelah proses pelatihan.

Salah satu pembagian yang umum digunakan dalam praktikum adalah 80% data latih dan 20% data uji, meskipun proporsi dapat disesuaikan dengan ukuran dan karakteristik dataset. Dalam scikit-learn, pembagian tersebut dapat dilakukan menggunakan fungsi `train_test_split()`.

3.8. Alur Transformasi Data

Secara umum, tahapan transformasi data digambarkan pada Gambar 2.



Gambar 3. Alur Transformasi Data

Alur tranformasi data pada Gambar 3 membantu memastikan bahwa keputusan transformasi dilakukan secara sistematis dan terdokumentasi. Dataset akhir yang

dihasilkan diharapkan telah memiliki format yang sesuai untuk digunakan pada tahapan pemodelan machine learning berikutnya.

4. Persiapan Praktikum

Sebelum memulai praktikum, pastikan dataset yang digunakan telah melalui tahap pemeriksaan dan pembersihan data sebagaimana dilakukan pada modul sebelumnya. Dataset yang akan digunakan pada Modul 5 selanjutnya dipersiapkan untuk proses transformasi sehingga dapat digunakan pada tahap pemodelan machine learning.

a. Lingkungan Praktikum

Praktikum dilaksanakan menggunakan Google Colab sebagai lingkungan utama. Mahasiswa perlu memastikan notebook dapat menjalankan Python dan library yang diperlukan.

Library utama yang digunakan meliputi:

- 1) pandas untuk membaca dan memanipulasi dataset;
- 2) NumPy untuk operasi numerik;
- 3) scikit-learn untuk encoding, scaling, seleksi fitur, dan pembagian dataset.

Library dapat dipanggil pada awal notebook sesuai kebutuhan praktikum.

b. Dataset Praktikum

Dataset yang digunakan sebaiknya memiliki kombinasi atribut numerik dan kategorikal agar mahasiswa dapat mempraktikkan beberapa bentuk transformasi data.

Sebelum melakukan transformasi, mahasiswa perlu mengenali:

- 1) jumlah baris dan kolom;
- 2) nama setiap atribut;
- 3) tipe data setiap atribut;
- 4) fitur numerik;
- 5) fitur kategorikal;
- 6) atribut yang akan menjadi target; dan
- 7) atribut yang tidak diperlukan dalam pemodelan.

Dataset yang digunakan pada tahap ini diasumsikan telah melalui proses pembersihan sehingga fokus praktikum diarahkan pada transformasi dan persiapan data untuk pemodelan.

c. Pemeriksaan Awal Dataset

Sebelum menerapkan transformasi, lakukan pemeriksaan awal terhadap dataset. Pemeriksaan dapat dilakukan menggunakan beberapa fungsi pandas, seperti:

```
df.head()
df.shape
df.info()
df.describe()
```

Pemeriksaan ini digunakan untuk memahami kembali struktur dataset dan menentukan teknik transformasi yang diperlukan.

Perhatikan terutama apakah terdapat:

- 1) atribut kategorikal yang harus diubah menjadi numerik;
- 2) fitur numerik dengan skala yang sangat berbeda;
- 3) atribut yang tidak relevan terhadap tujuan pemodelan;
- 4) atribut target yang perlu dipisahkan dari fitur.

d. Menentukan Fitur dan Target

Dataset untuk supervised learning umumnya dipisahkan menjadi dua bagian utama:

- X (features), yaitu sekumpulan atribut yang digunakan sebagai masukan model;
- y (target), yaitu atribut yang ingin diprediksi oleh model.

Contoh struktur pemisahan:

```
X = df.drop(columns=['target'])
y = df['target']
```

Nama kolom target harus disesuaikan dengan dataset yang digunakan.

Pemisahan fitur dan target perlu dilakukan secara tepat karena kolom target tidak boleh diperlakukan sebagai fitur masukan.

e. Rencana Transformasi

Sebelum menulis kode transformasi, mahasiswa perlu menentukan tindakan yang akan dilakukan pada setiap kelompok atribut.

Kondisi Data	Teknik yang Dipertimbangkan
Data kategorikal nominal	One-Hot Encoding
Data kategorikal ordinal	Encoding sesuai urutan kategori
Data numerik dengan perbedaan rentang besar	Normalisasi atau standardisasi
Fitur tidak relevan	Seleksi atau penghapusan fitur
Dataset siap untuk pemodelan	Pembagian training set dan test set

Teknik transformasi tidak harus diterapkan pada seluruh atribut. Pemilihan metode perlu didasarkan pada **tipe data, karakteristik dataset, dan kebutuhan proses pemodelan**.

5. Langkah Praktikum

Praktikum dilakukan secara bertahap untuk mengubah dataset hasil pembersihan menjadi dataset yang siap digunakan pada proses pemodelan. Setiap tahap perlu dijalankan dan didokumentasikan pada notebook Google Colab.

a. Mengimpor Library

Jalankan library yang diperlukan untuk proses transformasi data.

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder,
StandardScaler, MinMaxScaler
```

Library pandas digunakan untuk manipulasi data, sedangkan komponen dari scikit-learn digunakan untuk transformasi dan pembagian dataset.

b. Memuat Dataset

Muat dataset yang telah melalui tahap pembersihan pada Modul 4.

```
df = pd.read_csv('/content/dataset_bersih.csv')
df.head()
```

Periksa ukuran dan struktur dataset.

```
print("Ukuran dataset:", df.shape)
df.info()
```

Pastikan tidak terdapat masalah kualitas data utama yang belum ditangani sebelum melanjutkan ke tahap transformasi.

c. Mengidentifikasi Jenis Atribut

Pisahkan atribut berdasarkan tipe datanya.

```
kolom_numerik=df.select_dtypes(include=['int64',
'float64']).columns
kolom_kategorikal = df.select_dtypes(include=['object',
'category']).columns

print("Kolom numerik:")
print(kolom_numerik)

print("\nKolom kategorikal:")
print(kolom_kategorikal)
```

Periksa nilai unik pada atribut kategorikal.

```
for kolom in kolom_kategorikal:
    print(f"\n{kolom}")
    print(df[kolom].value_counts())
```

Catat atribut mana yang memerlukan encoding dan atribut numerik mana yang mungkin memerlukan scaling.

d. Menentukan Fitur dan Target

Tentukan kolom yang berperan sebagai target berdasarkan tujuan analisis.

Sebagai contoh:

```
X = df.drop(columns=['target'])
y = df['target']
```

Ganti nama target sesuai dengan kolom target pada dataset yang digunakan.

Periksa hasil pemisahan.

```
print("Ukuran fitur:", X.shape)
print("Ukuran target:", y.shape)
```

Pada tahap ini, pastikan kolom identitas yang tidak relevan, seperti nomor urut atau ID unik, tidak digunakan sebagai fitur tanpa alasan analitis yang jelas.

e. Melakukan Encoding Data Kategorikal

Atribut kategorikal perlu diubah menjadi bentuk numerik agar dapat digunakan oleh algoritma machine learning.

Salah satu pendekatan yang dapat digunakan adalah one-hot encoding dengan `pandas`.

```
X_encoded = pd.get_dummies(X, drop_first=True)
X_encoded.head()
```

Periksa perubahan jumlah fitur.

```
print("Jumlah fitur sebelum encoding:", X.shape[1])
print("Jumlah fitur setelah encoding:",
      X_encoded.shape[1])
```

Amati bagaimana setiap kategori diubah menjadi kolom numerik baru.

f. Membagi Data Menjadi Training Set dan Test Set

Setelah fitur dan target ditentukan, bagi dataset menjadi data latih dan data uji.

```
X_train, X_test, y_train, y_test = train_test_split(
    X_encoded,
    y,
    test_size=0.2,
    random_state=42
)
```

Periksa ukuran setiap bagian.

```
print("X_train:", X_train.shape)
print("X_test :", X_test.shape)
print("y_train:", y_train.shape)
print("y_test :", y_test.shape)
```

Dengan `test_size=0.2`, sekitar 80% data digunakan sebagai data latih dan 20% sebagai data uji. Parameter `random_state=42` digunakan agar pembagian data dapat direproduksi ketika notebook dijalankan kembali.

g. Melakukan Normalisasi

Jika fitur numerik memiliki rentang nilai yang sangat berbeda, normalisasi dapat dilakukan menggunakan `MinMaxScaler`.

```
scaler_minmax = MinMaxScaler()
X_train_normalized = scaler_minmax.fit_transform(X_train)
X_test_normalized = scaler_minmax.transform(X_test)
```

Hasil transformasi dapat dikembalikan ke bentuk `DataFrame`.

```
X_train_normalized = pd.DataFrame(
    X_train_normalized,
    columns=X_train.columns,
    index=X_train.index
)

X_test_normalized = pd.DataFrame(
    X_test_normalized,
    columns=X_test.columns,
    index=X_test.index
)
```

Periksa hasil normalisasi.

```
X_train_normalized.head()
```

Amati perubahan rentang nilai sebelum dan sesudah normalisasi.

h. Melakukan Standardisasi

Sebagai alternatif normalisasi, standardisasi dapat dilakukan menggunakan `StandardScaler`.

```

scaler_standard = StandardScaler()

X_train_standard = scaler_standard.fit_transform(X_train)
X_test_standard = scaler_standard.transform(X_test)

Ubah kembali hasilnya menjadi DataFrame.

X_train_standard = pd.DataFrame(
    X_train_standard,
    columns=X_train.columns,
    index=X_train.index
)

X_test_standard = pd.DataFrame(
    X_test_standard,
    columns=X_test.columns,
    index=X_test.index
)

```

Periksa statistik setelah standardisasi.

```
X_train_standard.describe()
```

Bandingkan hasil normalisasi dan standardisasi, kemudian jelaskan perbedaan yang terlihat.

i. Melakukan Seleksi Fitur Sederhana

Periksa kembali fitur yang tersedia setelah proses encoding.

```
print(X_encoded.columns.tolist())
```

Identifikasi fitur yang:

- 1) tidak relevan dengan tujuan analisis;
- 2) hanya berfungsi sebagai identitas;
- 3) memiliki informasi yang sama atau hampir sama dengan fitur lain;
- 4) tidak diperlukan dalam proses pemodelan.

Jika terdapat fitur yang tidak relevan, fitur tersebut dapat dihapus.

```

X_selected = X_encoded.drop(
    columns=['nama_fitur_yang_dihapus'],
    errors='ignore'
)

```

Nama fitur harus disesuaikan dengan dataset yang digunakan.

Setiap fitur yang dihapus harus disertai alasan yang dapat dijelaskan secara analitis.

j. Menghindari Data Leakage

Transformasi yang mempelajari parameter dari data, seperti normalisasi dan standardisasi, harus diterapkan dengan hati-hati.

Gunakan:

```
scaler.fit(X_train)
```

pada data latih, kemudian:

```
scaler.transform(X_train)
scaler.transform(X_test)
```

pada data latih dan data uji.

Jangan melakukan:

```
scaler.fit_transform(X_encoded)
```

sebelum membagi dataset apabila transformasi tersebut menggunakan informasi statistik dari keseluruhan data.

Prinsip ini penting agar informasi dari data uji tidak ikut digunakan ketika menyiapkan data latih.

k. Memeriksa Dataset Hasil Transformasi

Lakukan pemeriksaan akhir terhadap data yang telah ditransformasi.

```
print("Missing value X_train:")
print(X_train.isnull().sum().sum())

print("\nTipe data:")
print(X_train.dtypes.value_counts())

print("\nUkuran data:")
print("X_train:", X_train.shape)
print("X_test :", X_test.shape)
```

Pastikan bahwa:

- 1) fitur yang akan digunakan telah berbentuk numerik;
- 2) tidak terdapat nilai hilang yang tidak ditangani;
- 3) fitur dan target telah dipisahkan;
- 4) data latih dan data uji memiliki struktur fitur yang sama;
- 5) teknik scaling telah diterapkan secara konsisten; dan
- 6) dataset telah siap digunakan pada tahap pemodelan.

l. Menyimpan Dataset Hasil Transformasi

Dataset hasil transformasi dapat disimpan agar dapat digunakan kembali pada praktikum berikutnya.

Contoh:

```
X_train.to_csv('/content/X_train.csv', index=False)
X_test.to_csv('/content/X_test.csv', index=False)
```

```
y_train.to_csv('/content/y_train.csv', index=False)
y_test.to_csv('/content/y_test.csv', index=False)
```

Jika menggunakan hasil standardisasi, simpan versi tersebut dengan nama yang jelas.

```
X_train_standard.to_csv(
    '/content/X_train_standard.csv',
    index=False
)

X_test_standard.to_csv(
    '/content/X_test_standard.csv',
    index=False
)
```

m. Menyusun Kesimpulan Praktikum

Pada bagian akhir notebook, tuliskan kesimpulan singkat yang menjelaskan:

- 1) kondisi dataset sebelum transformasi;
- 2) teknik encoding yang digunakan;
- 3) apakah normalisasi atau standardisasi diperlukan;
- 4) fitur yang dipilih atau dihapus beserta alasannya;

- 5) proporsi pembagian data latih dan data uji; dan
- 6) kondisi akhir dataset setelah proses transformasi.

Kesimpulan tidak hanya berisi daftar proses yang dilakukan, tetapi juga menjelaskan alasan pemilihan teknik transformasi dan bagaimana transformasi tersebut mempersiapkan data untuk proses pemodelan.

6. Analisis Hasil Praktikum

Setelah seluruh tahapan transformasi data dilakukan, mahasiswa perlu menganalisis hasil yang diperoleh. Analisis tidak cukup hanya menunjukkan bahwa kode berhasil dijalankan, tetapi juga harus menjelaskan perubahan yang terjadi pada dataset dan alasan mengapa transformasi tersebut diperlukan.

a. Analisis Encoding Data Kategorikal

Bandingkan struktur data sebelum dan sesudah proses encoding. Perhatikan perubahan jumlah kolom, representasi kategori, serta kemungkinan bertambahnya jumlah fitur setelah menggunakan one-hot encoding.

Mahasiswa perlu menjelaskan:

- 1) atribut kategorikal yang ditransformasikan;
- 2) teknik encoding yang digunakan;
- 3) perubahan jumlah fitur setelah proses encoding;
- 4) alasan pemilihan teknik encoding tersebut.

b. Analisis Normalisasi dan Standardisasi

Bandingkan nilai fitur sebelum dan sesudah proses scaling. Amati perubahan rentang nilai, rata-rata, dan penyebaran data.

Gunakan statistik deskriptif untuk membantu analisis:

```
print("Sebelum transformasi:")
display(X_train.describe())

print("Setelah normalisasi:")
display(X_train_normalized.describe())

print("Setelah standardisasi:")
display(X_train_standard.describe())
```

Mahasiswa perlu menjelaskan perbedaan hasil antara normalisasi dan standardisasi serta menentukan teknik mana yang lebih sesuai untuk dataset yang digunakan.

c. Analisis Seleksi Fitur

Periksa kembali fitur yang digunakan dalam dataset akhir. Jika terdapat fitur yang dihapus, jelaskan alasan penghapusannya.

Analisis dapat mempertimbangkan beberapa hal, seperti:

- 1) fitur tidak berhubungan dengan tujuan analisis;
- 2) fitur hanya berfungsi sebagai identitas;
- 3) fitur memiliki informasi yang berulang;
- 4) fitur berpotensi tidak memberikan kontribusi terhadap pemodelan.

Mahasiswa tidak dianjurkan menghapus fitur hanya karena jumlahnya banyak. Keputusan seleksi fitur harus didasarkan pada alasan yang dapat dijelaskan.

d. Analisis Pembagian Dataset

Periksa proporsi data latih dan data uji yang dihasilkan.

Contoh:

```
total_data = len(X_train) + len(X_test)

persen_train = len(X_train) / total_data * 100
persen_test = len(X_test) / total_data * 100

print(f>Data latih : {persen_train:.2f}%")
print(f>Data uji : {persen_test:.2f}%")
```

Mahasiswa perlu memastikan bahwa pembagian dataset telah sesuai dengan proporsi yang ditentukan dan bahwa data uji tidak digunakan untuk mempelajari parameter transformasi.

e. Pemeriksaan Kesiapan Dataset

Dataset dapat dianggap siap untuk proses pemodelan apabila memenuhi kondisi berikut:

- 1) fitur telah berada dalam bentuk numerik;
- 2) tidak terdapat nilai hilang yang belum ditangani;
- 3) atribut target telah dipisahkan dari fitur;
- 4) fitur yang tidak relevan telah ditinjau;

- 5) data latih dan data uji memiliki struktur fitur yang sama;
- 6) proses scaling dilakukan berdasarkan data latih;
- 7) tidak terjadi penggunaan informasi data uji dalam proses persiapan data.

Pemeriksaan sederhana dapat dilakukan dengan:

```
print("Jumlah missing value X_train:",
      X_train.isnull().sum().sum())

print("Jumlah missing value X_test:",
      X_test.isnull().sum().sum())

print("Jumlah fitur X_train:",
      X_train.shape[1])

print("Jumlah fitur X_test:",
      X_test.shape[1])
```

f. Ringkasan Hasil Transformasi

Mahasiswa menyusun ringkasan proses dalam bentuk tabel berikut.

Tahap	Kondisi Awal	Teknik yang Digunakan	Hasil
Encoding	Data kategorikal	...	...
Scaling	Perbedaan skala fitur	...	...
Seleksi fitur	Terdapat fitur yang perlu ditinjau	...	...
Pembagian dataset	Dataset belum dipisahkan	Train-test split	...

Isi tabel berdasarkan hasil praktikum yang diperoleh.

g. Interpretasi Akhir

Tuliskan interpretasi akhir dalam bentuk narasi singkat dengan menjawab pertanyaan berikut:

1. Transformasi apa saja yang diterapkan pada dataset?
2. Mengapa transformasi tersebut diperlukan?
3. Apa perubahan utama yang terjadi pada dataset?
4. Apakah seluruh fitur telah siap digunakan dalam pemodelan?
5. Bagaimana proses yang dilakukan mencegah terjadinya **data leakage**?

6. Apakah terdapat fitur yang dihapus atau dipertahankan secara khusus? Jelaskan alasannya.
7. Apakah dataset hasil akhir sudah layak digunakan pada praktikum pemodelan berikutnya?

Analisis harus didasarkan pada hasil pengamatan terhadap dataset dan keluaran kode, bukan hanya menjelaskan kembali definisi teknik transformasi.

7. Tugas Praktikum

Setelah menyelesaikan seluruh langkah praktikum, mahasiswa mengerjakan tugas individu menggunakan dataset yang diberikan atau dataset lain yang disetujui dosen. Tugas ini bertujuan mengukur kemampuan dalam memilih dan menerapkan teknik transformasi data secara tepat hingga menghasilkan dataset yang siap digunakan untuk proses pemodelan.

a. Instruksi Tugas

Lakukan proses persiapan data dengan tahapan berikut:

1. muat dataset ke dalam Google Colab;
2. identifikasi atribut numerik, kategorikal, fitur, dan target;
3. lakukan encoding terhadap atribut kategorikal yang membutuhkan transformasi;
4. tentukan apakah fitur numerik memerlukan normalisasi atau standardisasi;
5. terapkan teknik scaling yang dipilih dan jelaskan alasannya;
6. lakukan seleksi fitur sederhana jika terdapat atribut yang tidak relevan;
7. pisahkan dataset menjadi data latih dan data uji;
8. pastikan transformasi yang mempelajari parameter hanya menggunakan data latih agar tidak terjadi **data leakage**;
9. periksa kembali struktur, tipe data, missing value, dan ukuran dataset hasil transformasi; dan
10. simpulkan apakah dataset telah siap digunakan untuk proses pemodelan machine learning.

b. Hasil yang Harus Ditampilkan

Notebook tugas minimal menampilkan:

- 1) informasi awal dataset;
- 2) daftar fitur numerik dan kategorikal;
- 3) fitur dan target yang digunakan;
- 4) hasil encoding;
- 5) hasil normalisasi atau standardisasi;
- 6) fitur yang dipertahankan atau dihapus;
- 7) ukuran X_{train} , X_{test} , y_{train} , dan y_{test} ;
- 8) pemeriksaan kondisi akhir dataset; dan
- 9) interpretasi singkat terhadap hasil transformasi.

c. Pertanyaan Analisis

Jawab pertanyaan berikut pada bagian akhir notebook.

- 1) Atribut mana yang memerlukan encoding dan mengapa?
- 2) Teknik encoding apa yang digunakan? Jelaskan alasan pemilihannya.
- 3) Apakah seluruh fitur numerik perlu dinormalisasi atau distandardisasi? Jelaskan.
- 4) Teknik scaling apa yang dipilih dan mengapa teknik tersebut sesuai dengan dataset?
- 5) Apakah terdapat fitur yang dihapus? Jika ada, jelaskan alasannya.
- 6) Berapa proporsi data latih dan data uji yang digunakan?
- 7) Mengapa proses scaling sebaiknya di-*fit* menggunakan data latih?
- 8) Apa risiko jika informasi dari data uji digunakan pada tahap persiapan data?
- 9) Apa perubahan utama yang terjadi pada dataset setelah transformasi?
- 10) Apakah dataset akhir telah siap digunakan untuk pemodelan? Jelaskan berdasarkan hasil pemeriksaan.

d. Ketentuan Pengumpulan

Tugas dikumpulkan dalam bentuk **Google Colab Notebook (.ipynb)** melalui LMS Edlink.

Notebook harus disusun secara sistematis dengan memuat:

- 1) judul dan identitas mahasiswa;

- 2) kode program;
- 3) keluaran setiap tahap;
- 4) penjelasan singkat menggunakan Markdown;
- 5) jawaban pertanyaan analisis; dan
- 6) kesimpulan akhir.

Nama berkas dapat menggunakan format:

`Modul5_NIM_Nama.ipynb`

Pastikan notebook dapat dijalankan kembali tanpa kesalahan sebelum dikumpulkan.

e. Kriteria Penilaian

Penilaian tugas mengacu pada indikator pertemuan ke-5 pada RPS, yaitu ketepatan memilih dan menerapkan teknik transformasi serta kemampuan menghasilkan dataset yang siap untuk pemodelan. Aspek yang dinilai mencakup:

Aspek Penilaian	Bobot
Ketepatan identifikasi fitur dan target	15%
Ketepatan penerapan encoding	15%
Ketepatan normalisasi/standardisasi	20%
Seleksi fitur dan alasan pemilihan	15%
Pembagian data latih dan data uji	15%
Struktur dan keterbacaan kode	10%
Analisis dan kesimpulan	10%
Total	100%

Keputusan transformasi harus disertai alasan yang sesuai dengan karakteristik data. Penilaian tidak hanya didasarkan pada keberhasilan menjalankan kode, tetapi juga pada kemampuan menjelaskan mengapa suatu teknik dipilih.

8. Latihan Mandiri

Latihan mandiri diberikan untuk memperkuat pemahaman mahasiswa terhadap proses transformasi data sebelum masuk ke tahap pemodelan. Latihan dapat dikerjakan menggunakan dataset yang sama dengan praktikum utama atau dataset lain yang memiliki kombinasi atribut numerik dan kategorikal.

Latihan 1 — Identifikasi Kebutuhan Transformasi

Perhatikan struktur dataset yang digunakan, kemudian tentukan:

- 1) atribut mana yang termasuk numerik;
- 2) atribut mana yang termasuk kategorikal;
- 3) atribut mana yang berperan sebagai target;
- 4) atribut mana yang tidak relevan untuk pemodelan;
- 5) atribut mana yang memerlukan encoding;
- 6) atribut mana yang berpotensi memerlukan normalisasi atau standardisasi.

Sajikan hasil identifikasi dalam bentuk tabel.

Atribut	Tipe Data	Peran	Transformasi yang Diperlukan	Alasan
...	...	...	...	...
...	...	...	...	...

Latihan 2 — Membandingkan Hasil Encoding

Pilih minimal satu atribut kategorikal, kemudian lakukan transformasi menggunakan teknik encoding yang sesuai.

Tampilkan:

- 1) data sebelum encoding;
- 2) data setelah encoding;
- 3) perubahan jumlah kolom; dan
- 4) penjelasan singkat mengenai perubahan yang terjadi.

Tuliskan alasan mengapa teknik encoding tersebut dipilih.

Latihan 3 — Membandingkan Normalisasi dan Standardisasi

Pilih minimal dua fitur numerik yang memiliki rentang nilai berbeda.

Lakukan:

- 1) normalisasi menggunakan MinMaxScaler;
- 2) standardisasi menggunakan StandardScaler;
- 3) tampilkan statistik deskriptif sebelum dan sesudah transformasi.

Contoh pemeriksaan:

```
print("Data asli")
display(X_train.describe())

print("Normalisasi")
display(X_train_normalized.describe())

print("Standardisasi")
display(X_train_standard.describe())
```

Jawab pertanyaan berikut:

- 1) Apa perubahan yang terjadi pada rentang nilai setelah normalisasi?
- 2) Apa perubahan yang terjadi pada rata-rata dan penyebaran data setelah standardisasi?
- 3) Teknik mana yang lebih sesuai untuk data yang digunakan? Jelaskan.

Latihan 4 — Eksperimen Pembagian Dataset

Lakukan pembagian dataset dengan beberapa proporsi berbeda, misalnya:

- a) 70% training dan 30% testing;
- b) 80% training dan 20% testing;
- c) 90% training dan 10% testing.

Contoh:

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    Y,
    test_size=0.2,
    random_state=42
)
```

Catat jumlah data pada setiap pembagian.

Proporsi Training–Testing	Jumlah Training	Jumlah Testing
70%–30%	...	...
80%–20%	...	...
90%–10%	...	...

Jelaskan bagaimana perubahan proporsi tersebut memengaruhi jumlah data yang tersedia untuk proses pelatihan dan pengujian.

Latihan 5 — Memeriksa Data Leakage

Bandingkan dua skenario berikut:

Skenario A

Scaling dilakukan pada seluruh dataset sebelum pembagian training dan testing.

Skenario B

Dataset dibagi terlebih dahulu, kemudian scaler di-*fit* menggunakan data training dan diterapkan pada data training serta testing.

Jelaskan:

- 1) perbedaan kedua skenario;
- 2) skenario mana yang lebih tepat;
- 3) mengapa penggunaan informasi dari data testing dalam proses persiapan data dapat menyebabkan data leakage.

Latihan 6 — Evaluasi Kesiapan Dataset

Periksa kembali dataset akhir dan buat checklist berikut.

- 1) Seluruh fitur yang digunakan telah ditentukan.
- 2) Target telah dipisahkan dari fitur.
- 3) Atribut kategorikal telah ditransformasikan sesuai kebutuhan.
- 4) Fitur numerik telah ditinjau untuk kebutuhan scaling.
- 5) Fitur yang tidak relevan telah ditinjau.
- 6) Data training dan testing telah dipisahkan.
- 7) Transformasi berbasis statistik hanya di-*fit* pada data training.
- 8) Tidak terdapat missing value yang belum ditangani.
- 9) Struktur fitur pada data training dan testing konsisten.
- 10) Dataset siap digunakan untuk pemodelan.

Refleksi Singkat

Tuliskan refleksi singkat sebanyak 1–2 paragraf mengenai hal berikut:

- 1) transformasi apa yang paling memengaruhi bentuk dataset;
- 2) teknik apa yang membutuhkan pertimbangan paling besar;
- 3) kesalahan apa yang berpotensi terjadi jika transformasi dilakukan tanpa memahami karakteristik data; dan

- 4) mengapa tahap persiapan data penting sebelum membangun model machine learning.

9. Rangkuman

Transformasi data merupakan tahapan penting dalam persiapan dataset sebelum digunakan pada proses pemodelan machine learning. Dataset yang telah dibersihkan belum tentu dapat langsung digunakan oleh algoritma karena masih dapat mengandung atribut kategorikal, perbedaan skala antarfitur, maupun fitur yang kurang relevan.

Pada praktikum ini, proses persiapan data dilakukan melalui beberapa tahapan utama, yaitu:

- 1) **mengidentifikasi jenis atribut**, sehingga dapat diketahui fitur numerik, kategorikal, target, serta atribut yang tidak diperlukan;
- 2) **melakukan encoding**, yaitu mengubah data kategorikal menjadi representasi numerik yang dapat diproses oleh algoritma machine learning;
- 3) **melakukan normalisasi atau standardisasi** untuk menyesuaikan skala fitur numerik apabila diperlukan;
- 4) **melakukan seleksi fitur** dengan mempertimbangkan relevansi atribut terhadap tujuan pemodelan;
- 5) **memisahkan fitur dan target** agar struktur data sesuai dengan kebutuhan supervised learning;
- 6) **membagi dataset menjadi training set dan test set** agar tersedia data untuk proses pelatihan dan pengujian model;
- 7) **mencegah data leakage** dengan memastikan parameter transformasi dipelajari hanya dari data training; dan
- 8) **memeriksa kembali dataset hasil transformasi** untuk memastikan data telah siap digunakan dalam proses pemodelan.

Normalisasi dan standardisasi memiliki tujuan yang sama dalam mengurangi pengaruh perbedaan skala antarfitur, tetapi menghasilkan karakteristik data yang berbeda. Normalisasi umumnya mengubah nilai ke rentang tertentu, sedangkan standardisasi mengubah distribusi fitur sehingga memiliki rata-rata mendekati 0 dan standar deviasi mendekati 1. Pemilihan teknik harus disesuaikan dengan karakteristik dataset dan kebutuhan algoritma.

Encoding juga tidak dapat diterapkan dengan cara yang sama pada seluruh atribut kategorikal. Teknik yang digunakan harus mempertimbangkan apakah suatu kategori memiliki urutan atau hanya berupa label nominal. Demikian pula, seleksi fitur tidak sekadar mengurangi jumlah kolom, tetapi harus didasarkan pada relevansi fitur terhadap masalah yang akan diselesaikan.

Salah satu prinsip penting yang perlu diperhatikan adalah urutan proses transformasi. Dataset sebaiknya dibagi menjadi data training dan testing sebelum menerapkan transformasi yang mempelajari parameter statistik data. Scaler di-*fit* menggunakan data training, kemudian digunakan untuk mentransformasi data training dan testing. Prosedur ini membantu mencegah informasi dari data testing masuk ke dalam proses persiapan model.

Referensi

1. Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann–Elsevier.
2. Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
3. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O'Reilly Media.
4. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
5. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
6. McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly Media.
7. Dokumentasi resmi **Python**, **pandas**, **NumPy**, **Jupyter**, dan **scikit-learn** sebagai referensi teknis untuk pengolahan, transformasi, pembagian dataset, dan persiapan data untuk pemodelan machine learning.

MODUL 6

KLASIFIKASI DENGAN K-NEAREST NEIGHBORS, DECISION TREE, DAN NAIVE BAYES

1. Identitas Praktikum

Judul Praktikum	: Klasifikasi dengan K-Nearest Neighbors, Decision Tree, dan Naive Bayes
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SK
CPMK	: CPMK-2 — Mampu menyiapkan data dan menerapkan algoritma supervised serta unsupervised learning untuk menemukan pola, melakukan klasifikasi, dan menghasilkan prediksi.
Sub-CPMK	: SUB-CPMK-2.2 — Mampu menerapkan algoritma klasifikasi dan regresi untuk menyelesaikan permasalahan prediksi pada data bisnis.
Lingkungan Praktikum	: Google Colab / Jupyter Notebook
Library Utama	: pandas, NumPy, Matplotlib, dan scikit-learn

Sub-CPMK yang Dicapai

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu menerapkan algoritma klasifikasi untuk menyelesaikan permasalahan prediksi pada data bisnis. Mahasiswa melakukan proses penyiapan data, menentukan variabel fitur dan target, membagi dataset menjadi data latih dan data uji, membangun model klasifikasi, serta menghasilkan prediksi menggunakan algoritma K-Nearest Neighbors (KNN), Decision Tree, dan Naive Bayes.

Praktikum juga diarahkan agar mahasiswa mampu memahami mekanisme dasar masing-masing algoritma serta menginterpretasikan keluaran awal model sebagai dasar untuk melakukan evaluasi dan perbandingan model pada praktikum berikutnya.

2. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- a) menjelaskan konsep dasar klasifikasi sebagai salah satu pendekatan supervised learning;
- b) memahami mekanisme kerja algoritma K-Nearest Neighbors (KNN), Decision Tree, dan Naive Bayes;
- c) menyiapkan dataset klasifikasi dengan menentukan variabel fitur dan variabel target;
- d) membagi dataset menjadi data latih (training data) dan data uji (testing data) untuk proses pemodelan;
- e) membangun model klasifikasi menggunakan KNN, Decision Tree, dan Naive Bayes dengan library scikit-learn;
- f) melakukan proses pelatihan model dan menghasilkan prediksi terhadap data uji;
- g) membaca serta menginterpretasikan keluaran awal dari model klasifikasi;
- h) membandingkan karakteristik dasar hasil klasifikasi dari beberapa algoritma sebagai pengantar menuju evaluasi model pada praktikum berikutnya; dan
- i) mendokumentasikan proses praktikum secara sistematis melalui notebook Google Colab/Jupyter Notebook.

3. Dasar Teori

Klasifikasi merupakan salah satu teknik dalam supervised learning yang digunakan untuk memprediksi kelas atau kategori dari suatu data berdasarkan pola yang dipelajari dari data berlabel. Pada proses klasifikasi, dataset umumnya terdiri atas sejumlah fitur (*features*) sebagai variabel masukan dan sebuah target (label) sebagai kelas yang akan diprediksi. Dalam RPS, praktikum pada pertemuan ke-6 berfokus pada konsep klasifikasi, algoritma K-Nearest Neighbors, Decision Tree, dan Naive Bayes, serta proses pelatihan dan prediksi model.

3.1. Konsep Dasar Klasifikasi

Secara umum, proses klasifikasi terdiri atas beberapa tahapan, yaitu menyiapkan data, menentukan fitur dan target, membagi dataset menjadi data latih dan data uji, melatih model menggunakan data latih, kemudian menggunakan model tersebut untuk memprediksi kelas pada data yang belum pernah dilihat sebelumnya.

Data latih digunakan oleh algoritma untuk mempelajari pola hubungan antara fitur dan label. Setelah proses pelatihan selesai, model digunakan untuk menghasilkan prediksi terhadap data uji. Hasil prediksi tersebut kemudian dapat dibandingkan dengan label sebenarnya untuk mengetahui seberapa baik model bekerja.

Dalam konteks data bisnis, klasifikasi dapat digunakan untuk berbagai kebutuhan seperti memprediksi apakah pelanggan akan berhenti menggunakan layanan, menentukan apakah suatu transaksi termasuk normal atau mencurigakan, mengklasifikasikan kelayakan kredit, atau memprediksi kategori respons pelanggan.

3.2. K-Nearest Neighbors

K-Nearest Neighbors (KNN) merupakan algoritma klasifikasi yang menentukan kelas suatu data berdasarkan kelas dari sejumlah data terdekat di sekitarnya. Nilai K menunjukkan jumlah tetangga terdekat yang digunakan dalam proses pengambilan keputusan. Jika $K = 5$, misalnya, maka model akan melihat lima data terdekat dari data yang akan diprediksi. Kelas yang paling banyak muncul di antara lima tetangga tersebut menjadi hasil prediksi.

Secara sederhana, proses KNN meliputi:

- a) menentukan nilai K;
- b) menghitung jarak antara data baru dengan seluruh data latih;
- c) memilih K data dengan jarak terdekat;
- d) melihat kelas dari tetangga terdekat tersebut; dan
- e) menentukan kelas berdasarkan kelas yang paling dominan.

KNN sangat dipengaruhi oleh skala data. Oleh karena itu, normalisasi atau standardisasi fitur sering diperlukan agar satu fitur dengan nilai yang besar tidak mendominasi perhitungan jarak.

3.3. Decision Tree

Decision Tree adalah algoritma yang melakukan klasifikasi menggunakan struktur menyerupai pohon keputusan. Model membagi data berdasarkan kondisi tertentu pada fitur hingga diperoleh kelompok data yang semakin homogen.

Struktur dasar Decision Tree terdiri atas:

- a) root node, yaitu titik awal pembagian data;
- b) internal node, yaitu kondisi atau pengujian terhadap suatu fitur;
- c) branch, yaitu hasil dari suatu kondisi; dan
- d) leaf node, yaitu keputusan akhir atau kelas hasil prediksi.

Decision Tree relatif mudah dipahami karena proses pengambilan keputusan dapat divisualisasikan sebagai serangkaian aturan. Sebagai contoh, model dapat menghasilkan aturan seperti:

Jika pendapatan > nilai tertentu dan status pelanggan aktif, maka kelas = loyal.

Algoritma ini dapat bekerja pada data dengan hubungan yang tidak selalu linear dan tidak memerlukan proses penskalaan fitur seperti KNN.

3.4. Naive Bayes

Naive Bayes merupakan algoritma klasifikasi berbasis probabilitas. Algoritma ini menggunakan prinsip Teorema Bayes untuk menghitung kemungkinan suatu data termasuk ke dalam kelas tertentu.

Secara umum, Naive Bayes menggunakan hubungan:

$$P(\text{Kelas} \mid \text{Fitur}) \propto P(\text{Fitur} \mid \text{Kelas}) \times P(\text{Kelas})$$

Model kemudian memilih kelas dengan probabilitas terbesar sebagai hasil prediksi.

Naive Bayes disebut naive karena mengasumsikan bahwa fitur-fitur yang digunakan bersifat independen satu sama lain terhadap kelas target. Walaupun asumsi ini tidak selalu sepenuhnya terpenuhi pada data nyata, algoritma ini tetap dapat memberikan hasil yang baik pada berbagai permasalahan klasifikasi.

3.5. Training dan Testing Data

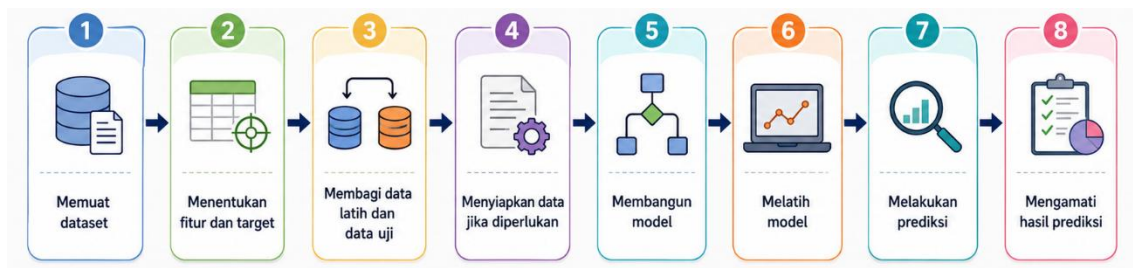
Sebelum model klasifikasi dibangun, dataset biasanya dibagi menjadi dua bagian utama:

- 1) Training data digunakan untuk melatih model agar dapat mempelajari pola dari data.
- 2) Testing data digunakan untuk menguji kemampuan model dalam menghasilkan prediksi terhadap data yang tidak digunakan selama pelatihan.

Pembagian ini penting agar performa model tidak hanya dinilai dari data yang sudah pernah dipelajari. Dalam praktikum menggunakan scikit-learn, pembagian dataset dapat dilakukan dengan fungsi `train_test_split()`.

3.6. Alur Dasar Pemodelan Klasifikasi

Alur umum praktikum klasifikasi dapat diringkas sebagai berikut:



Gambar 4. Alur Praktikum Klasifikasi

Pada Modul 6, perhatian utama diberikan pada proses membangun, melatih, dan menghasilkan prediksi dengan KNN, Decision Tree, dan Naive Bayes. Evaluasi model secara lebih mendalam menggunakan confusion matrix, accuracy, precision, recall, F1-score, ROC, dan AUC dibahas pada bagian pembelajaran berikutnya sesuai urutan materi dalam RPS.

4. Persiapan Praktikum

Sebelum membangun model klasifikasi, mahasiswa perlu memastikan lingkungan praktikum, library, dan dataset telah siap digunakan. Pada Modul 6, implementasi dilakukan menggunakan **Python** melalui **Google Colab** atau **Jupyter Notebook**, dengan **scikit-learn** sebagai library utama untuk membangun model klasifikasi. Penggunaan Google Colab dan scikit-learn sesuai dengan media pembelajaran yang ditetapkan dalam RPS.

a. Perangkat dan Lingkungan Praktikum

Perangkat yang digunakan dalam praktikum meliputi:

- 1) laptop atau komputer yang terhubung ke internet;
- 2) Google Colab atau Jupyter Notebook;
- 3) browser web;
- 4) akun Google untuk mengakses Google Colab; dan
- 5) LMS Edlink untuk mengakses materi dan mengumpulkan hasil praktikum.

Notebook praktikum sebaiknya disusun secara sistematis dengan memisahkan bagian kode, hasil eksekusi, visualisasi, serta interpretasi singkat terhadap hasil yang diperoleh.

b. Library yang Digunakan

Library utama yang digunakan adalah:

- 1) **pandas**, untuk membaca dan mengelola dataset;
- 2) **NumPy**, untuk mendukung operasi numerik;
- 3) **Matplotlib**, untuk visualisasi data atau hasil analisis;
- 4) **scikit-learn**, untuk pembagian dataset, preprocessing, pembangunan model, dan proses prediksi.

Library yang diperlukan dapat dipanggil pada awal notebook sebagai berikut:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
```

c. Dataset Praktikum

Dataset yang digunakan harus memiliki **variabel target berbentuk kategori atau kelas**, karena tujuan praktikum adalah membangun model klasifikasi.

Secara umum, dataset terdiri atas:

- 1) **fitur (X)**, yaitu variabel yang digunakan sebagai masukan model; dan
- 2) **target (y)**, yaitu kelas atau kategori yang ingin diprediksi.

Contoh struktur sederhana dataset klasifikasi pada Tabel 11.

Tabel 11. Contoh Strktur Dataset Klasifikasi

Usia	Pendapatan	Lama_Berlangganan	Jumlah_Transaksi	Status
25	3500000	12	8	Tidak Loyal
34	6500000	36	24	Loyal
29	4200000	18	12	Tidak Loyal
41	8000000	48	35	Loyal

Contoh dataset klasifikasi Pada Tabel 11, terdapat variabel: Usia, Pendapatan, Lama_Berlangganan, dan Jumlah_Transaksi berperan sebagai fitur, sedangkan Status merupakan target yang akan diprediksi.

d. Memuat dan Memeriksa Dataset

Dataset dapat dimuat menggunakan pandas.

```
df = pd.read_csv('dataset.csv')
df.head()
Setelah dataset berhasil dimuat, lakukan pemeriksaan awal.
df.shape
df.info()
df.describe()
df.isnull().sum()
```

Pemeriksaan tersebut digunakan untuk memastikan jumlah data, tipe variabel, distribusi awal data numerik, serta keberadaan nilai yang hilang sebelum model dibangun.

e. Menentukan Fitur dan Target

Fitur dan target harus dipisahkan sebelum proses pemodelan.

Sebagai contoh:

```
X = df.drop('Status', axis=1)
y = df['Status']
```

Variabel X berisi seluruh fitur yang digunakan untuk melakukan prediksi, sedangkan y berisi kelas target.

Mahasiswa perlu memastikan bahwa target yang dipilih benar-benar sesuai dengan permasalahan klasifikasi yang akan diselesaikan.

f. Membagi Data Latih dan Data Uji

Dataset kemudian dibagi menjadi **training set** dan **testing set**.

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

Pada contoh tersebut, 80% data digunakan untuk proses pelatihan dan 20% digunakan sebagai data uji. Penggunaan `random_state` membantu menghasilkan pembagian data yang konsisten ketika notebook dijalankan kembali.

Parameter `stratify=y` digunakan agar proporsi kelas pada data latih dan data uji tetap mendekati proporsi kelas pada dataset awal.

g. Standardisasi Data untuk KNN

KNN menentukan kedekatan data berdasarkan jarak sehingga perbedaan skala antarfitur dapat memengaruhi hasil model. Oleh karena itu, fitur numerik yang digunakan untuk KNN perlu dipertimbangkan untuk distandardisasi.

```
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

`fit_transform()` diterapkan pada data latih, sedangkan data uji hanya dikenai `transform()`. Dengan cara tersebut, informasi dari data uji tidak digunakan ketika parameter standardisasi dipelajari.

Setelah seluruh persiapan selesai, dataset telah siap digunakan untuk membangun tiga model klasifikasi utama pada praktikum ini:

K-Nearest Neighbors → Decision Tree → Naive Bayes

Tahap pembangunan dan pelatihan ketiga model tersebut dilakukan pada **Bagian 5. Langkah Praktikum**.

5. Langkah Praktikum

Pada bagian ini mahasiswa akan membangun tiga model klasifikasi, yaitu K-Nearest Neighbors (KNN), Decision Tree, dan Naive Bayes. Fokus praktikum adalah memahami alur implementasi model, proses pelatihan, dan menghasilkan prediksi sebagaimana materi pertemuan ke-6 pada RPS.

a. Mengimpor Library

Jalankan kode berikut pada bagian awal notebook.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.naive_bayes import GaussianNB
```

b. Memuat Dataset

Muat dataset yang digunakan dalam praktikum.

```
df = pd.read_csv('dataset.csv')
df.head()
Periksa struktur dataset.
df.info()
```

Periksa ukuran dataset.

```
df.shape
```

Pastikan tidak terdapat masalah data yang belum ditangani.

```
df.isnull().sum()
```

Jika masih ditemukan data hilang, tipe data yang tidak sesuai, atau masalah kualitas data lainnya, lakukan penanganan terlebih dahulu sebelum melanjutkan proses pemodelan.

c. Menentukan Fitur dan Target

Pisahkan variabel fitur dan target.

```
X = df.drop('Status', axis=1)
y = df['Status']
```

Periksa fitur yang digunakan.

```
X.head()
```

Periksa distribusi kelas target.

```
y.value_counts()
```

Mahasiswa perlu memahami bahwa variabel X merupakan data masukan bagi model, sedangkan y merupakan kelas yang akan diprediksi.

d. Membagi Data Latih dan Data Uji

Gunakan `train_test_split()` untuk membagi dataset.

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

Periksa ukuran masing-masing bagian data.

```
print("X_train :", X_train.shape)
print("X_test  :", X_test.shape)
print("y_train :", y_train.shape)
print("y_test  :", y_test.shape)
```

Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk menghasilkan prediksi terhadap data yang tidak ikut digunakan selama proses pelatihan.

e. Membangun Model K-Nearest Neighbors

KNN menggunakan kedekatan antarobjek sebagai dasar penentuan kelas. Karena algoritma ini bergantung pada perhitungan jarak, fitur numerik perlu berada pada skala yang relatif sebanding.

Lakukan standardisasi terlebih dahulu.

```
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Selanjutnya, buat model KNN dengan nilai $K = 5$.

```
knn = KNeighborsClassifier(n_neighbors=5)
```

Latih model menggunakan data latih.

```
knn.fit(X_train_scaled, y_train)
```

Lakukan prediksi terhadap data uji.

```
y_pred_knn = knn.predict(X_test_scaled)
Tampilkan beberapa hasil prediksi.
hasil_knn = pd.DataFrame({
    'Aktual': y_test.values,
    'Prediksi_KNN': y_pred_knn
})

hasil_knn.head(10)
```

Amati apakah kelas hasil prediksi sama atau berbeda dengan kelas aktual.

Eksperimen Nilai K

Cobalah beberapa nilai K.

```
for k in [3, 5, 7, 9]:
    model = KNeighborsClassifier(n_neighbors=k)
    model.fit(X_train_scaled, y_train)
    prediksi = model.predict(X_test_scaled)

    print("K =", k)
    print(prediksi[:10])
    print()
```

Catat perubahan hasil prediksi ketika nilai K diubah.

f. Membangun Model Decision Tree

Berbeda dengan KNN, Decision Tree membentuk aturan keputusan berdasarkan fitur pada dataset.

Buat model Decision Tree.

```
dt = DecisionTreeClassifier(
    max_depth=4,
    random_state=42
)
```

Latih model.

```
dt.fit(X_train, y_train)
```

Lakukan prediksi.

```
y_pred_dt = dt.predict(X_test)
Tampilkan hasil prediksi.
hasil_dt = pd.DataFrame({
    'Aktual': y_test.values,
    'Prediksi_DecisionTree': y_pred_dt
})

hasil_dt.head(10)
```

g. Memvisualisasikan Decision Tree

Salah satu kelebihan Decision Tree adalah struktur keputusan model dapat divisualisasikan.

```
plt.figure(figsize=(18, 8))

plot_tree(
    dt,
    feature_names=X.columns,
    class_names=[str(k) for k in dt.classes_],
    filled=True
)

plt.show()
```

Perhatikan:

- 1) fitur yang digunakan pada node awal;
- 2) kondisi pembagian data;
- 3) kelas yang dihasilkan pada leaf node; dan
- 4) kedalaman pohon yang terbentuk.

Mahasiswa tidak hanya diminta melihat gambar pohon, tetapi juga mencoba membaca satu alur keputusan dari root node sampai leaf node.

h. Mengamati Feature Importance Decision Tree

Decision Tree dapat memberikan gambaran mengenai fitur yang memiliki kontribusi relatif lebih besar dalam proses pembentukan pohon.

```
importance = pd.DataFrame({
    'Fitur': X.columns,
    'Importance': dt.feature_importances_
```

```

}))

importance.sort_values(
    by='Importance',
    ascending=False
)

```

Amati fitur mana yang memiliki nilai importance paling tinggi.

Pada tahap ini, nilai tersebut cukup digunakan sebagai bahan pengamatan awal terhadap perilaku model. Interpretasi model secara lebih mendalam akan dibahas pada bagian pembelajaran berikutnya.

i. Membangun Model Naive Bayes

Model ketiga yang digunakan adalah **Gaussian Naive Bayes**.

Buat model.

```
nb = GaussianNB()
```

Latih model menggunakan data latih.

```
nb.fit(X_train, y_train)
```

Lakukan prediksi.

```

y_pred_nb = nb.predict(X_test)
Tampilkan hasilnya.
hasil_nb = pd.DataFrame({
    'Aktual': y_test.values,
    'Prediksi_NaiveBayes': y_pred_nb
})

hasil_nb.head(10)

```

Naive Bayes menghasilkan kelas berdasarkan probabilitas. Probabilitas prediksi dapat dilihat menggunakan:

```

probabilitas_nb = nb.predict_proba(X_test)

probabilitas_nb[:5]

```

Untuk melihat urutan kelas yang berhubungan dengan kolom probabilitas:

```
nb.classes_
```

Perhatikan bagaimana setiap data memiliki probabilitas untuk masing-masing kelas dan bagaimana model menentukan kelas dengan probabilitas terbesar.

j. Membandingkan Prediksi Ketiga Model

Gabungkan hasil prediksi ketiga algoritma agar lebih mudah diamati.

```
perbandingan = pd.DataFrame({
    'Aktual': y_test.values,
    'KNN': y_pred_knn,
    'Decision_Tree': y_pred_dt,
    'Naive_Bayes': y_pred_nb
})

perbandingan.head(15)
```

Amati beberapa kondisi berikut:

- 1) data yang diprediksi sama oleh ketiga model;
- 2) data yang menghasilkan prediksi berbeda;
- 3) model yang paling sering menghasilkan kelas yang sama dengan data aktual; dan
- 4) kemungkinan penyebab perbedaan prediksi berdasarkan karakteristik masing-masing algoritma.

Pada Modul 6, perbandingan ini masih bersifat **pengamatan awal**. Penentuan model yang benar-benar lebih baik tidak cukup dilakukan hanya dengan melihat beberapa prediksi. Evaluasi menggunakan metrik seperti accuracy, precision, recall, F1-score, confusion matrix, ROC, dan AUC ditempatkan pada materi evaluasi model sesuai urutan RPS.

k. Melakukan Prediksi Data Baru

Setelah model terbentuk, mahasiswa dapat mencoba memasukkan satu data baru.

Contoh:

```
data_baru = pd.DataFrame({
    'Usia': [32],
    'Pendapatan': [5500000],
    'Lama_Berlangganan': [24],
    'Jumlah_Transaksi': [18]
})
```

Untuk KNN, data baru perlu ditransformasikan menggunakan scaler yang telah dibangun dari data latih.

```
data_baru_scaled = scaler.transform(data_baru)

prediksi_knn_baru = knn.predict(data_baru_scaled)

print("Prediksi KNN:", prediksi_knn_baru[0])
Untuk Decision Tree:
prediksi_dt_baru = dt.predict(data_baru)

print("Prediksi Decision Tree:", prediksi_dt_baru[0])
Untuk Naive Bayes:
prediksi_nb_baru = nb.predict(data_baru)

print("Prediksi Naive Bayes:", prediksi_nb_baru[0])
```

Bandingkan hasil prediksi dari ketiga algoritma.

I. Mendokumentasikan Hasil Praktikum

Pada setiap tahap, mahasiswa perlu menambahkan penjelasan singkat menggunakan **Markdown Cell**, minimal mencakup:

- 1) dataset dan target yang digunakan;
- 2) fitur yang digunakan untuk membangun model;
- 3) alasan dilakukan standardisasi pada KNN;
- 4) nilai K yang digunakan;
- 5) parameter utama Decision Tree;
- 6) hasil prediksi masing-masing algoritma;
- 7) perbedaan hasil prediksi yang ditemukan; dan
- 8) kesimpulan awal mengenai karakteristik KNN, Decision Tree, dan Naive Bayes.

Notebook sebaiknya memperlihatkan proses analisis secara runtut sehingga kode, keluaran, dan interpretasi dapat ditelusuri dengan mudah.

6. Tugas Praktikum

Tugas praktikum pada Modul 6 dirancang untuk mengukur kemampuan mahasiswa dalam membangun dan menerapkan model klasifikasi menggunakan **K-Nearest Neighbors, Decision Tree, dan Naive Bayes**. Penugasan ini selaras dengan RPS

pertemuan ke-6, yang menilai kebenaran implementasi, kesesuaian algoritma, dan interpretasi keluaran awal model.

a. Deskripsi Tugas

Gunakan sebuah dataset klasifikasi yang telah disiapkan oleh dosen atau dataset lain yang sesuai dengan konteks bisnis dan memiliki target kategorikal.

Lakukan seluruh proses klasifikasi mulai dari persiapan data sampai menghasilkan prediksi menggunakan tiga algoritma berikut:

- 1) K-Nearest Neighbors;
- 2) Decision Tree; dan
- 3) Naive Bayes.

Seluruh proses dikerjakan menggunakan **Google Colab** atau **Jupyter Notebook** dan harus disertai penjelasan singkat pada setiap tahapan.

b. Ketentuan Pengerjaan

Mahasiswa diminta melakukan kegiatan berikut:

- 1) Memuat dataset dan menampilkan lima baris pertama data.
- 2) Menjelaskan secara singkat isi dan konteks dataset.
- 3) Menentukan fitur dan target yang digunakan dalam proses klasifikasi.
- 4) Menampilkan distribusi kelas target.
- 5) Membagi dataset menjadi data latih dan data uji.
- 6) Melakukan standardisasi fitur yang digunakan pada model KNN.
- 7) Membangun model KNN menggunakan minimal tiga variasi nilai K.
- 8) Mengamati perubahan hasil prediksi ketika nilai K diubah.
- 9) Membangun model Decision Tree.
- 10) Menampilkan visualisasi pohon keputusan.
- 11) Mengidentifikasi fitur yang memiliki nilai feature importance tertinggi.
- 12) Membangun model Gaussian Naive Bayes.
- 13) Menghasilkan prediksi menggunakan ketiga algoritma.
- 14) Menampilkan hasil aktual dan hasil prediksi dalam sebuah tabel.
- 15) Membandingkan hasil prediksi ketiga model.
- 16) Melakukan prediksi terhadap minimal tiga data baru.
- 17) Menuliskan interpretasi dan kesimpulan berdasarkan hasil praktikum.

c. **Pertanyaan Analisis**

Jawab pertanyaan berikut pada bagian akhir notebook.

- 1) **Apa yang dimaksud dengan klasifikasi dalam supervised learning?**
Jelaskan berdasarkan proses yang dilakukan pada praktikum.
- 2) **Mengapa data pada KNN perlu distandardisasi?**
Hubungkan jawaban dengan cara kerja KNN yang menggunakan perhitungan jarak.
- 3) **Apa pengaruh perubahan nilai K terhadap hasil prediksi KNN?**
Gunakan hasil eksperimen yang telah dilakukan sebagai dasar penjelasan.
- 4) **Bagaimana Decision Tree menentukan kelas suatu data?**
Jelaskan dengan mengambil satu jalur keputusan dari visualisasi pohon yang dihasilkan.
- 5) **Fitur apa yang memiliki nilai feature importance tertinggi pada Decision Tree?**
Jelaskan makna temuan tersebut dalam konteks dataset yang digunakan.
- 6) **Bagaimana Naive Bayes menentukan kelas hasil prediksi?**
Jelaskan berdasarkan probabilitas yang dihasilkan oleh model.
- 7) **Apakah ketiga algoritma selalu menghasilkan prediksi yang sama?**
Tunjukkan contoh data yang menghasilkan prediksi sama dan, jika ditemukan, contoh data yang menghasilkan prediksi berbeda.
- 8) **Mengapa hasil prediksi dari KNN, Decision Tree, dan Naive Bayes dapat berbeda walaupun menggunakan dataset yang sama?**
Hubungkan jawaban dengan mekanisme dasar masing-masing algoritma.

d. **Format Hasil Praktikum**

Notebook yang dikumpulkan minimal memiliki struktur berikut:

- 1) **Identitas Praktikum**
Nama, NIM, kelas, dan tanggal praktikum.
- 2) **Deskripsi Dataset**
Penjelasan singkat dataset, fitur, dan target.
- 3) **Persiapan Data**
Pemuatan data, pemeriksaan awal, pemisahan fitur dan target, serta pembagian data.

4) **Model K-Nearest Neighbors**

Standardisasi, pembangunan model, variasi nilai K, prediksi, dan interpretasi.

5) **Model Decision Tree**

Pembangunan model, prediksi, visualisasi pohon, feature importance, dan interpretasi.

6) **Model Naive Bayes**

Pembangunan model, prediksi, probabilitas prediksi, dan interpretasi.

7) **Perbandingan Hasil Prediksi**

Tabel perbandingan antara kelas aktual dan hasil prediksi ketiga algoritma.

8) **Prediksi Data Baru**

Hasil prediksi terhadap minimal tiga data baru.

9) **Jawaban Pertanyaan Analisis**

Jawaban terhadap seluruh pertanyaan pada Bagian 6.3.

10) **Kesimpulan**

Ringkasan hasil praktikum dan pemahaman yang diperoleh.

e. Ketentuan Pengumpulan

Hasil praktikum dikumpulkan melalui **LMS Edlink** dalam bentuk notebook .ipynb atau tautan Google Colab sesuai arahan dosen. RPS juga menetapkan Google Colab dan LMS Edlink sebagai lingkungan praktik serta media pengumpulan aktivitas pembelajaran.

Sebelum dikumpulkan, pastikan:

- 1) seluruh cell kode dapat dijalankan;
- 2) tidak terdapat error pada notebook;
- 3) keluaran program tetap ditampilkan;
- 4) setiap tahapan memiliki penjelasan atau interpretasi;
- 5) nama file mengikuti format yang ditentukan dosen; dan
- 6) tautan Google Colab, jika digunakan, memiliki hak akses yang sesuai.

Format nama file yang dapat digunakan:

`Modul6_NIM_Nama.ipynb`

Contoh:

`Modul6_231001_AndiSaputra.ipynb`

7. Hasil yang Diharapkan

Setelah menyelesaikan Modul 6, mahasiswa diharapkan mampu menunjukkan pemahaman dan keterampilan awal dalam membangun model klasifikasi menggunakan **K-Nearest Neighbors, Decision Tree, dan Naive Bayes**. Hasil praktikum tidak hanya berupa kode yang berhasil dijalankan, tetapi juga harus memperlihatkan bahwa mahasiswa memahami proses pelatihan, prediksi, serta karakteristik dasar dari masing-masing algoritma.

Secara khusus, hasil yang diharapkan meliputi:

1) Dataset siap digunakan untuk pemodelan

Dataset telah diperiksa, fitur dan target telah ditentukan dengan benar, serta data telah dibagi menjadi data latih dan data uji.

2) Model K-Nearest Neighbors berhasil dibangun

Mahasiswa mampu melakukan standardisasi data, menentukan nilai K, melatih model, dan menghasilkan prediksi terhadap data uji.

3) Model Decision Tree berhasil dibangun

Mahasiswa mampu melatih model Decision Tree, menghasilkan prediksi, serta membaca struktur dasar pohon keputusan yang terbentuk.

4) Model Naive Bayes berhasil dibangun

Mahasiswa mampu menerapkan Gaussian Naive Bayes dan memahami bahwa proses klasifikasi dilakukan berdasarkan probabilitas masing-masing kelas.

5) Hasil prediksi dapat dibandingkan

Mahasiswa mampu menampilkan hasil aktual dan hasil prediksi ketiga algoritma dalam satu tabel sehingga persamaan dan perbedaan hasil dapat diamati.

6) Perbedaan karakteristik algoritma dapat dijelaskan

Mahasiswa mampu menjelaskan secara sederhana bahwa:

- a) KNN menentukan kelas berdasarkan kedekatan data;
- b) Decision Tree menentukan kelas melalui aturan keputusan;

c) Naive Bayes menentukan kelas berdasarkan probabilitas.

7) **Prediksi terhadap data baru dapat dilakukan**

Mahasiswa mampu menggunakan model yang telah dilatih untuk memprediksi kelas dari data baru yang belum pernah digunakan dalam proses pelatihan.

8) **Interpretasi awal hasil model dapat disusun**

Mahasiswa mampu menuliskan penjelasan singkat mengenai hasil prediksi dan memberikan alasan yang logis terhadap perbedaan hasil antaralgoritma.

9) **Notebook terdokumentasi dengan baik**

Notebook berisi kode, keluaran, visualisasi, dan penjelasan yang tersusun secara sistematis sehingga proses analisis dapat ditelusuri kembali.

Indikator Keberhasilan Praktikum

Praktikum dapat dinyatakan berhasil apabila mahasiswa mampu:

- 1) menjalankan seluruh tahapan tanpa error yang menghambat proses utama;
- 2) menghasilkan model KNN, Decision Tree, dan Naive Bayes;
- 3) menghasilkan prediksi pada data uji;
- 4) menunjukkan hasil perbandingan prediksi;
- 5) menjelaskan mekanisme dasar ketiga algoritma;
- 6) melakukan prediksi pada data baru; dan
- 7) memberikan interpretasi awal terhadap keluaran model.

Capaian tersebut sejalan dengan indikator pada RPS pertemuan ke-6, yaitu ketepatan menjelaskan mekanisme algoritma dan membangun model klasifikasi, dengan penilaian yang menekankan **kebenaran implementasi, kesesuaian algoritma, dan interpretasi keluaran awal**. Pada tahap ini, mahasiswa belum dituntut melakukan evaluasi komprehensif menggunakan confusion matrix, precision, recall, F1-score, ROC, atau AUC karena materi tersebut ditempatkan pada bagian evaluasi model di pertemuan berikutnya dalam RPS.

8. Evaluasi dan Refleksi

Bagian evaluasi dan refleksi digunakan untuk membantu mahasiswa menilai kembali proses praktikum yang telah dilakukan serta mengidentifikasi pemahaman yang sudah dikuasai dan bagian yang masih perlu diperbaiki.

a. Evaluasi Pemahaman

Jawab pertanyaan berikut secara singkat dan jelas berdasarkan hasil praktikum.

- 1) Apa perbedaan utama antara K-Nearest Neighbors, Decision Tree, dan Naive Bayes dalam menentukan kelas suatu data?
- 2) Mengapa KNN lebih sensitif terhadap skala fitur dibandingkan Decision Tree?
- 3) Apa yang dapat terjadi jika nilai K pada KNN terlalu kecil?
- 4) Apa yang dapat terjadi jika nilai K terlalu besar?
- 5) Mengapa Decision Tree relatif mudah diinterpretasikan?
- 6) Apa fungsi `max_depth` pada Decision Tree?
- 7) Apa asumsi dasar yang digunakan oleh Naive Bayes terhadap hubungan antarfitur?
- 8) Mengapa hasil prediksi ketiga algoritma dapat berbeda walaupun menggunakan data latih dan data uji yang sama?
- 9) Apa fungsi data latih dan data uji dalam proses klasifikasi?
- 10) Mengapa model sebaiknya diuji menggunakan data yang tidak digunakan pada saat proses pelatihan?

b. Refleksi Praktikum

Tuliskan refleksi singkat berdasarkan pengalaman selama praktikum dengan menjawab pertanyaan berikut.

- 1) Algoritma mana yang paling mudah dipahami? Jelaskan alasannya.
- 2) Algoritma mana yang menurut hasil pengamatan menghasilkan prediksi paling konsisten pada dataset yang digunakan?
- 3) Tahapan apa yang paling sulit dilakukan selama praktikum?
- 4) Apakah perubahan parameter model memengaruhi hasil prediksi? Berikan satu contoh dari eksperimen yang dilakukan.
- 5) Apa pemahaman baru yang diperoleh setelah membandingkan KNN, Decision Tree, dan Naive Bayes?
- 6) Jika praktikum diulangi menggunakan dataset lain, bagian apa yang perlu lebih diperhatikan?

c. Refleksi terhadap Proses Pemodelan

Mahasiswa perlu memahami bahwa membangun model klasifikasi bukan sekadar menjalankan fungsi `fit()` dan `predict()`. Hasil model sangat dipengaruhi oleh kondisi dataset, fitur yang digunakan, proses penyiapan data, serta karakteristik algoritma.

Pada praktikum ini, mahasiswa telah melihat bahwa:

data yang sama → algoritma berbeda → mekanisme belajar berbeda → prediksi dapat berbeda

Karena itu, pemilihan algoritma tidak seharusnya hanya didasarkan pada kemudahan implementasi. Model perlu dianalisis dan, pada tahap berikutnya, dievaluasi menggunakan ukuran kinerja yang sesuai sebelum digunakan sebagai dasar pengambilan keputusan.

d. Pemeriksaan Mandiri

Sebelum mengumpulkan hasil praktikum, mahasiswa dapat melakukan pemeriksaan mandiri berikut.

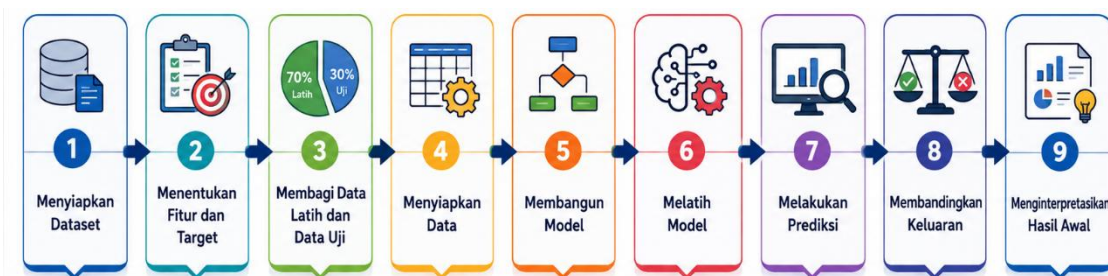
- 1) Dataset berhasil dimuat dan diperiksa.
- 2) Fitur dan target telah ditentukan dengan benar.
- 3) Dataset telah dibagi menjadi data latih dan data uji.
- 4) Standardisasi telah dilakukan untuk KNN.
- 5) Model KNN berhasil dilatih dan menghasilkan prediksi.
- 6) Minimal tiga nilai K telah dicoba.
- 7) Model Decision Tree berhasil dilatih dan menghasilkan prediksi.
- 8) Pohon keputusan berhasil divisualisasikan.
- 9) Feature importance telah diperiksa.
- 10) Model Naive Bayes berhasil dilatih dan menghasilkan prediksi.
- 11) Probabilitas hasil prediksi Naive Bayes telah diamati.
- 12) Hasil prediksi ketiga algoritma telah dibandingkan.
- 13) Prediksi terhadap data baru telah dilakukan.
- 14) Pertanyaan analisis telah dijawab.
- 15) Refleksi praktikum telah dituliskan.
- 16) Notebook dapat dijalankan tanpa error pada proses utama.
- 17) Notebook telah diberi penjelasan menggunakan Markdown.

Evaluasi dan refleksi ini mendukung indikator RPS pertemuan ke-6 yang menekankan kemampuan mahasiswa dalam menjelaskan mekanisme algoritma, membangun model klasifikasi, serta menginterpretasikan keluaran awal dari model yang digunakan.

9. Rangkuman

Klasifikasi merupakan teknik **supervised learning** yang digunakan untuk memprediksi kelas atau kategori berdasarkan pola yang dipelajari dari data berlabel. Pada Modul 6, proses klasifikasi dilakukan menggunakan tiga algoritma, yaitu **K-Nearest Neighbors (KNN)**, **Decision Tree**, dan **Naive Bayes**, sesuai materi pembelajaran pada pertemuan ke-6 dalam RPS.

Secara umum, proses pemodelan klasifikasi yang telah dilakukan meliputi:



Gambar 5. Proses Pemodelan Klasifikasi

Beberapa konsep penting yang diperoleh dari praktikum ini adalah sebagai berikut.

1. **K-Nearest Neighbors (KNN)** menentukan kelas suatu data berdasarkan kelas dari sejumlah tetangga terdekat. Nilai **K** menentukan jumlah tetangga yang digunakan dalam proses prediksi. Karena menggunakan perhitungan jarak, KNN sensitif terhadap perbedaan skala fitur sehingga standardisasi data perlu diperhatikan.
2. **Decision Tree** menentukan kelas melalui serangkaian aturan keputusan yang terbentuk dari proses pembagian data. Struktur pohon membuat proses pengambilan keputusan relatif mudah diamati dan dijelaskan. Parameter seperti `max_depth` dapat digunakan untuk mengatur kompleksitas pohon.
3. **Naive Bayes** melakukan klasifikasi berdasarkan probabilitas suatu data termasuk ke dalam masing-masing kelas. Model kemudian menentukan kelas berdasarkan probabilitas yang diperoleh dari fitur data.

4. **Training data** digunakan untuk melatih model, sedangkan **testing data** digunakan untuk melihat kemampuan model menghasilkan prediksi terhadap data yang tidak digunakan ketika model dibangun.
5. Algoritma yang berbeda dapat menghasilkan prediksi yang berbeda meskipun menggunakan dataset yang sama. Perbedaan tersebut terjadi karena setiap algoritma menggunakan mekanisme yang berbeda dalam mempelajari pola dari data.
6. Proses klasifikasi tidak berhenti setelah perintah `fit()` dan `predict()` berhasil dijalankan. Hasil model perlu diamati dan diinterpretasikan agar keluaran algoritma dapat dipahami dalam konteks permasalahan yang dianalisis.
7. Perbandingan hasil prediksi pada Modul 6 masih merupakan **analisis awal**. Penentuan model yang lebih baik memerlukan metode evaluasi yang sesuai. Dalam RPS, evaluasi klasifikasi menggunakan confusion matrix, accuracy, precision, recall, specificity, F1-score, ROC, dan AUC ditempatkan pada materi evaluasi model pada pertemuan selanjutnya.

Melalui praktikum ini, mahasiswa diharapkan telah memperoleh dasar keterampilan untuk menerapkan algoritma klasifikasi menggunakan **scikit-learn** dan memahami bahwa pemilihan model perlu mempertimbangkan karakteristik data, mekanisme algoritma, serta hasil yang diperoleh. Pemahaman tersebut menjadi dasar untuk mempelajari tahap berikutnya dalam machine learning, yaitu membangun model prediksi numerik menggunakan **algoritma regresi** serta melakukan evaluasi model secara lebih sistematis.

Referensi

1. Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann–Elsevier.
2. Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
3. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O'Reilly Media.

4. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
5. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
6. McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly Media.
7. Python Software Foundation. *Python Documentation*.
8. pandas Development Team. *pandas Documentation*.
9. NumPy Developers. *NumPy Documentation*.
10. Matplotlib Development Team. *Matplotlib Documentation*.
11. scikit-learn Developers. *scikit-learn Documentation*.
12. Project Jupyter. *Jupyter Documentation*.

MODUL 7

REGRESI LINEAR UNTUK PREDIKSI DATA BISNIS

1. Identitas Praktikum

Judul Praktikum	: Regresi Linear untuk Prediksi Data Bisnis
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SKS
CPMK	: CPMK-2 — Mampu menyiapkan data dan menerapkan algoritma supervised serta unsupervised learning untuk menemukan pola, melakukan klasifikasi, dan menghasilkan prediksi.
Sub-CPMK	: SUB-CPMK-2.2 — Mampu menerapkan algoritma klasifikasi dan regresi untuk menyelesaikan permasalahan prediksi pada data bisnis.
Lingkungan Praktikum	: Google Colab / Jupyter Notebook
Library Utama	: pandas, NumPy, Matplotlib, dan scikit-learn

Sub-CPMK yang dicapai adalah:

Mahasiswa mampu membangun model regresi untuk memprediksi nilai numerik pada data bisnis, membaca koefisien model, serta menginterpretasikan hasil prediksi dalam konteks permasalahan yang dianalisis.

Pada praktikum ini, mahasiswa akan mempelajari penerapan regresi linear sebagai salah satu metode supervised learning untuk memodelkan hubungan antara satu atau lebih variabel prediktor dengan variabel target numerik. Praktikum diarahkan pada proses menyiapkan data, menentukan variabel fitur dan target, membagi dataset, melatih model regresi linear, menghasilkan prediksi, serta menginterpretasikan koefisien dan keluaran model. Implementasi dilakukan menggunakan Python dan scikit-learn pada Google Colab. Hasil praktikum berupa notebook yang mendokumentasikan proses pemodelan dan penjelasan hasil analisis, sesuai dengan aktivitas pembelajaran pada RPS.

2. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- 1) menjelaskan konsep dasar regresi sebagai metode *supervised learning* untuk memprediksi nilai numerik;
- 2) mengidentifikasi variabel fitur dan variabel target pada permasalahan regresi;
- 3) menyiapkan dataset untuk proses pemodelan regresi;
- 4) membagi dataset menjadi data latih dan data uji;
- 5) membangun model **Linear Regression** menggunakan library scikit-learn;
- 6) menggunakan model yang telah dilatih untuk menghasilkan nilai prediksi;
- 7) membaca dan menjelaskan nilai koefisien serta *intercept* dari model regresi;
- 8) membandingkan nilai aktual dengan nilai hasil prediksi;
- 9) menginterpretasikan hasil model dalam konteks permasalahan bisnis yang dianalisis; dan
- 10) mendokumentasikan seluruh proses pemodelan dan hasil analisis secara sistematis dalam Google Colab.

Praktikum ini menekankan kemampuan mahasiswa untuk tidak hanya menjalankan algoritma regresi, tetapi juga memahami hubungan antara variabel, membaca keluaran model, serta menghubungkan hasil prediksi dengan masalah bisnis yang menjadi konteks analisis.

3. Dasar Teori

3.1. Konsep Regresi

Regresi merupakan salah satu metode *supervised learning* yang digunakan untuk mempelajari hubungan antara variabel masukan (*feature*) dan variabel target yang berbentuk numerik. Tujuan utama regresi adalah membangun model yang dapat digunakan untuk memperkirakan atau memprediksi suatu nilai berdasarkan pola yang terdapat pada data historis.

Dalam konteks bisnis, regresi dapat digunakan untuk berbagai kebutuhan, seperti memprediksi penjualan, memperkirakan harga produk, memproyeksikan pendapatan, memperkirakan jumlah permintaan, atau memprediksi biaya berdasarkan sejumlah faktor yang memengaruhinya.

Secara umum, proses regresi melibatkan data yang memiliki:

- a. **variabel fitur (x)**, yaitu variabel yang digunakan sebagai prediktor;
- b. **variabel target (y)**, yaitu nilai numerik yang ingin diprediksi.

Model regresi mempelajari hubungan antara fitur dan target dari data latih, kemudian menggunakan hubungan tersebut untuk menghasilkan prediksi pada data baru.

3.2. Regresi Linear

Regresi linear merupakan salah satu metode regresi yang paling dasar. Metode ini mengasumsikan adanya hubungan linear antara variabel prediktor dengan variabel target. Untuk satu variabel prediktor, bentuk umum regresi linear dapat dituliskan sebagai:

$$y = b_0 + b_1 x$$

dengan:

- a. y = nilai target yang diprediksi;
- b. x = nilai variabel prediktor;
- c. b_0 = *intercept* atau konstanta;
- d. b_1 = koefisien regresi.

Nilai *intercept* menunjukkan nilai prediksi ketika nilai variabel prediktor sama dengan nol. Sementara itu, nilai koefisien menunjukkan besarnya perubahan nilai target ketika variabel prediktor mengalami perubahan satu satuan.

Jika digunakan lebih dari satu variabel prediktor, model regresi linear dapat dituliskan sebagai:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

Setiap koefisien menggambarkan arah dan besarnya hubungan antara masing-masing fitur dengan variabel target.

3.3. Koefisien dan Intercept

Koefisien merupakan salah satu bagian penting dalam interpretasi model regresi linear. Koefisien bernilai **positif** menunjukkan bahwa peningkatan nilai fitur cenderung diikuti peningkatan nilai target. Sebaliknya, koefisien bernilai **negatif** menunjukkan bahwa peningkatan nilai fitur cenderung diikuti penurunan nilai target.

Sebagai contoh, jika diperoleh model:

$$\text{Penjualan} = 50 + 3 (\text{Promosi})$$

maka dapat diinterpretasikan bahwa:

- a. nilai *intercept* sebesar 50 menunjukkan nilai prediksi penjualan ketika variabel promosi bernilai 0;
- b. koefisien promosi sebesar 3 menunjukkan bahwa setiap kenaikan satu satuan pada promosi diperkirakan meningkatkan nilai penjualan sebesar 3 satuan.

Interpretasi koefisien tetap harus mempertimbangkan konteks dan satuan dari masing-masing variabel.

1. Data Latih dan Data Uji

Sebelum model dibangun, dataset umumnya dibagi menjadi dua bagian utama:

- a. **Data latih (*training set*)** digunakan untuk melatih model agar dapat mempelajari pola hubungan antara fitur dan target.
- b. **Data uji (*testing set*)** digunakan untuk menguji kemampuan model dalam menghasilkan prediksi terhadap data yang tidak digunakan pada saat pelatihan.

Pembagian ini penting agar performa model tidak hanya dinilai berdasarkan kemampuannya mengingat data latih, tetapi juga berdasarkan kemampuannya melakukan prediksi terhadap data baru.

Pada implementasi menggunakan scikit-learn, pembagian data dapat dilakukan dengan fungsi `train_test_split()`.

2. Proses Pelatihan dan Prediksi

Secara umum, tahapan pembangunan model regresi linear meliputi:

- a. menyiapkan dataset;
- b. menentukan variabel fitur dan target;
- c. membagi data menjadi data latih dan data uji;
- d. membuat objek model regresi;
- e. melatih model menggunakan data latih;

- f. menghasilkan prediksi terhadap data uji;
- g. membandingkan nilai aktual dengan hasil prediksi;
- h. membaca koefisien dan *intercept* model;
- i. menginterpretasikan hasil sesuai konteks permasalahan bisnis.

Pada scikit-learn, model regresi linear dapat dibuat menggunakan kelas `LinearRegression`.

3. Nilai Aktual dan Nilai Prediksi

Setelah proses pelatihan selesai, model digunakan untuk menghasilkan nilai prediksi. Nilai tersebut kemudian dapat dibandingkan dengan nilai aktual pada data uji. Perbandingan ini membantu mahasiswa memahami seberapa dekat hasil prediksi model terhadap kondisi sebenarnya. Pada tahap berikutnya, kualitas prediksi dapat diukur lebih lanjut menggunakan metrik evaluasi regresi.

Dalam Modul 7 ini, fokus utama adalah pada pembangunan model dan interpretasi hasil prediksi, sesuai dengan materi pertemuan ke-7 pada RPS. Evaluasi regresi secara lebih khusus, termasuk MAE, MSE, RMSE, dan (R^2), dibahas pada materi evaluasi model pada pertemuan berikutnya.

4. Implementasi Regresi Linear dengan Scikit-Learn

Library scikit-learn menyediakan kelas `LinearRegression` untuk membangun model regresi linear. Struktur dasar implementasinya adalah:

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
Setelah model dilatih, nilai intercept dan koefisien
dapat diperiksa melalui:
print(model.intercept_)
print(model.coef_)
```

Hasil tersebut kemudian digunakan untuk membantu memahami hubungan yang dipelajari model antara fitur dan target.

4. Persiapan Praktikum

Sebelum memulai praktikum regresi linear, mahasiswa perlu memastikan lingkungan kerja, library, dan dataset telah siap digunakan. Praktikum dilakukan menggunakan **Google Colab** agar proses pemodelan dapat dilakukan tanpa instalasi perangkat lunak tambahan.

1. Lingkungan Praktikum

Perangkat dan perangkat lunak yang digunakan dalam praktikum ini adalah:

- laptop atau komputer yang terhubung ke internet;
- browser web;
- akun Google;
- Google Colab;
- Python;
- pandas untuk pengolahan data;
- NumPy untuk operasi numerik;
- Matplotlib untuk visualisasi;
- scikit-learn untuk pembagian dataset dan pembangunan model regresi.

Mahasiswa membuat satu notebook Google Colab dan memberi nama sesuai format yang ditentukan pada kegiatan praktikum.

2. Menyiapkan Notebook Google Colab

Buka Google Colab melalui browser, kemudian buat notebook baru. Pada bagian awal notebook, tuliskan identitas praktikum menggunakan *Markdown*, minimal memuat:

```
MODUL 7 – REGRESI LINEAR UNTUK PREDIKSI DATA BISNIS
Nama      :
NIM       :
Kelas    :
Mata Kuliah: Data Mining dan Machine Learning
```

Notebook digunakan untuk menyimpan kode program, keluaran analisis, visualisasi, dan interpretasi hasil praktikum secara terstruktur.

3. Mengimpor Library

Pada sel kode pertama, impor library yang diperlukan.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Jalankan sel tersebut dan pastikan tidak terdapat pesan kesalahan.

Library yang digunakan memiliki fungsi utama sebagai berikut:

Library	Fungsi
pandas	Membaca dan mengolah dataset
NumPy	Mendukung operasi numerik
Matplotlib	Membuat visualisasi data dan hasil prediksi
train_test_split	Membagi dataset menjadi data latih dan data uji
LinearRegression	Membuat dan melatih model regresi linear

4. Menyiapkan Dataset

Dataset yang digunakan pada Modul 7 harus memiliki **variabel target berbentuk numerik**, karena tujuan regresi adalah menghasilkan prediksi berupa nilai kontinu.

Contoh permasalahan yang sesuai antara lain:

- memprediksi nilai penjualan berdasarkan biaya promosi;
- memprediksi pendapatan berdasarkan jumlah transaksi;
- memprediksi harga berdasarkan karakteristik produk;
- memprediksi jumlah permintaan berdasarkan faktor-faktor tertentu.

Dataset minimal memiliki satu atau lebih variabel fitur dan satu variabel target numerik.

Contoh struktur sederhana dataset:

Biaya_Promosi	Jumlah_Pelanggan	Penjualan
2.5	120	35.0
3.0	135	39.5
4.2	160	48.0
5.1	185	56.5
6.0	210	64.0

Pada contoh tersebut:

- Biaya_Promosi dan Jumlah_Pelanggan dapat digunakan sebagai **fitur (X)**;
- Penjualan digunakan sebagai **target (y)**.

5. Memuat Dataset ke Google Colab

Apabila dataset tersedia dalam format CSV, berkas dapat diunggah ke Google Colab kemudian dibaca menggunakan pandas.

```
from google.colab import files

uploaded = files.upload()
Setelah file berhasil diunggah, dataset dapat dibaca dengan:
df = pd.read_csv('nama_dataset.csv')
Ganti nama_dataset.csv sesuai nama file yang digunakan.
Tampilkan lima baris pertama dataset untuk memastikan data
berhasil dibaca.
df.head()
```

6. Pemeriksaan Awal Dataset

Sebelum membangun model, lakukan pemeriksaan awal terhadap dataset.

```
df.shape
```

Kode tersebut digunakan untuk mengetahui jumlah baris dan kolom.

Selanjutnya, periksa nama kolom:

```
df.columns
```

Periksa tipe data:

```
df.info()
```

Kemudian tampilkan statistik deskriptif:

```
df.describe()
```

Mahasiswa perlu memastikan bahwa variabel yang akan digunakan sebagai fitur dan target memiliki tipe data yang sesuai untuk proses regresi.

7. Memeriksa Data Hilang

Lakukan pemeriksaan terhadap nilai yang hilang menggunakan:

```
df.isnull().sum()
```

Apabila terdapat data hilang atau masalah kualitas data lainnya, lakukan penanganan terlebih dahulu dengan mengacu pada teknik pembersihan data yang telah dipelajari pada modul sebelumnya.

Dataset yang digunakan untuk pemodelan sebaiknya sudah berada dalam kondisi siap analisis.

8. Menentukan Fitur dan Target

Sebelum melatih model, tentukan dengan jelas:

Fitur (X)

Variabel yang digunakan oleh model untuk melakukan prediksi.

Target (y)

Variabel numerik yang ingin diprediksi.

Sebagai contoh:

```
X = df[['Biaya_Promosi', 'Jumlah_Pelanggan']]
y = df['Penjualan']
```

Pemilihan fitur dan target harus sesuai dengan tujuan analisis yang telah ditentukan.

9. Alur Persiapan Praktikum

Secara ringkas, tahapan persiapan sebelum pembangunan model adalah:

membuka Google Colab → mengimpor library → memuat dataset → memeriksa struktur data → memeriksa kualitas data → menentukan fitur dan target → memastikan dataset siap untuk pemodelan.

Setelah seluruh tahapan persiapan selesai dan dataset telah siap digunakan, praktikum dapat dilanjutkan ke proses pembagian data, pembangunan model regresi linear, prediksi, dan interpretasi hasil.

5. Langkah Praktikum

Pada bagian ini, mahasiswa akan membangun model **regresi linear** untuk memprediksi nilai numerik berdasarkan data bisnis. Tahapan praktikum mengikuti alur pemodelan regresi pada RPS, yaitu menyiapkan data, membangun model, menghasilkan prediksi, membaca koefisien, dan menginterpretasikan hasil.

1. Memuat Dataset

Unggah dataset yang telah disiapkan ke Google Colab.

```
from google.colab import files
uploaded = files.upload()
```

Selanjutnya, baca dataset menggunakan pandas.

```
import pandas as pd

df = pd.read_csv('nama_dataset.csv')
df.head()
```

Ganti `nama_dataset.csv` sesuai nama file yang digunakan.

2. Memeriksa Struktur Dataset

Periksa ukuran, nama kolom, tipe data, dan statistik deskriptif.

```
print("Ukuran dataset:", df.shape)
print("\nNama kolom:")
print(df.columns)

print("\nInformasi dataset:")
df.info()
Tampilkan statistik deskriptif.
df.describe()
Selanjutnya, periksa data hilang.
df.isnull().sum()
```

Pastikan data yang digunakan telah siap untuk proses pemodelan.

3. Menentukan Variabel Fitur dan Target

Tentukan variabel yang digunakan sebagai fitur dan target.

Sebagai contoh, apabila dataset memuat kolom `Biaya_Promosi`, `Jumlah_Pelanggan`, dan `Penjualan`, maka:

```
X = df[['Biaya_Promosi', 'Jumlah_Pelanggan']]
y = df['Penjualan']
```

Pada contoh tersebut:

- a. X merupakan variabel fitur;
- b. y merupakan variabel target;
- c. target harus berupa nilai numerik.

Periksa bentuk data.

```
print("Ukuran X:", X.shape)
print("Ukuran y:", y.shape)
```

4. Membagi Dataset Menjadi Data Latih dan Data Uji

Gunakan `train_test_split()` untuk membagi dataset.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42
)
```

Periksa hasil pembagian data.

```
print("Jumlah data latih :", len(X_train))
print("Jumlah data uji    :", len(X_test))
```

Parameter `test_size=0.2` berarti 20% data digunakan sebagai data uji, sedangkan 80% lainnya digunakan sebagai data latih.

`random_state=42` digunakan agar pembagian dataset dapat direproduksi ketika notebook dijalankan kembali.

5. Membuat Model Regresi Linear

```
Impor kelas LinearRegression.
from sklearn.linear_model import LinearRegression
```

Selanjutnya, buat objek model.

```
model = LinearRegression()
```

Pada tahap ini, objek model telah dibuat tetapi belum mempelajari pola dari dataset.

6. Melatih Model

Latih model menggunakan data latih.

```
model.fit(X_train, y_train)
```

Perintah `fit()` membuat model mempelajari hubungan antara fitur pada `X_train` dengan target pada `y_train`.

Setelah proses pelatihan selesai, model dapat digunakan untuk melakukan prediksi.

7. Membaca Intercept dan Koefisien Model

Tampilkan nilai *intercept*.

```
print("Intercept:", model.intercept_)
Selanjutnya, tampilkan koefisien masing-masing fitur.
print("Koefisien:", model.coef_)
Agar hasil lebih mudah dibaca, buat tabel koefisien.
koefisien = pd.DataFrame({
    'Fitur': X.columns,
    'Koefisien': model.coef_
})

koefisien
```

Mahasiswa kemudian menginterpretasikan setiap koefisien berdasarkan konteks dataset.

Secara umum:

- koefisien positif menunjukkan hubungan searah dengan target;
- koefisien negatif menunjukkan hubungan berlawanan arah dengan target;
- semakin besar nilai absolut koefisien, semakin besar perubahan prediksi target untuk perubahan satu satuan fitur, dengan tetap memperhatikan skala variabel.

8. Menghasilkan Prediksi

Gunakan model yang telah dilatih untuk memprediksi data uji.

```
y_pred = model.predict(X_test)
```

Tampilkan beberapa hasil prediksi.

```
print(y_pred[:10])
```

Nilai `y_pred` merupakan nilai target yang diperkirakan oleh model berdasarkan fitur pada data uji.

9. Membandingkan Nilai Aktual dan Prediksi

Buat tabel perbandingan antara nilai aktual dan nilai prediksi.

```
hasil_prediksi = pd.DataFrame({
    'Aktual': y_test.values,
    'Prediksi': y_pred
})
```

```

hasil_prediksi.head(10)
Tambahkan kolom selisih antara nilai aktual dan
prediksi.
hasil_prediksi['Selisih'] = (
    hasil_prediksi['Aktual'] -
    hasil_prediksi['Prediksi']
)

hasil_prediksi.head(10)

```

Perhatikan apakah hasil prediksi cenderung dekat atau jauh dari nilai aktual.

Pada Modul 7, analisis difokuskan pada pemahaman hasil prediksi. Pengukuran performa model secara formal menggunakan MAE, MSE, RMSE, dan (R^2) akan dibahas lebih lanjut pada materi evaluasi model sesuai RPS.

10. Visualisasi Nilai Aktual dan Prediksi

Buat visualisasi sederhana untuk membandingkan hasil prediksi dengan nilai aktual.

```

import matplotlib.pyplot as plt
plt.scatter(
    hasil_prediksi['Aktual'],
    hasil_prediksi['Prediksi']
)

plt.xlabel('Nilai Aktual')
plt.ylabel('Nilai Prediksi')
plt.title('Perbandingan Nilai Aktual dan Prediksi')
plt.show()

```

Apabila prediksi model cukup mengikuti nilai aktual, titik-titik pada grafik akan menunjukkan pola yang semakin mendekati hubungan diagonal.

Visualisasi ini digunakan sebagai pemeriksaan awal dan bukan sebagai satu-satunya dasar untuk menentukan kualitas model.

11. Melakukan Prediksi pada Data Baru

Model yang telah dilatih dapat digunakan untuk melakukan prediksi terhadap data baru.

Contoh:

```
data_baru = pd.DataFrame({
    'Biaya_Promosi': [5.5],
    'Jumlah_Pelanggan': [200]
})
prediksi_baru = model.predict(data_baru)
print("Hasil prediksi:", prediksi_baru[0])
```

Sesuaikan nama fitur dengan dataset yang digunakan.

Mahasiswa perlu menjelaskan arti nilai prediksi tersebut berdasarkan konteks masalah bisnis.

12. Menginterpretasikan Model

Setelah proses pemodelan selesai, mahasiswa menyusun interpretasi yang menjawab pertanyaan berikut:

1. Apa variabel target yang diprediksi?
2. Fitur apa saja yang digunakan untuk membangun model?
3. Berapa nilai *intercept* yang diperoleh?
4. Berapa koefisien setiap fitur?
5. Apa arti tanda positif atau negatif pada setiap koefisien?
6. Bagaimana perbandingan nilai aktual dan nilai prediksi?
7. Apakah hasil prediksi secara umum mengikuti pola data aktual?
8. Apa makna hasil model tersebut dalam konteks bisnis?

Contoh bentuk interpretasi:

Model regresi menunjukkan bahwa variabel *Biaya_Promosi* memiliki koefisien positif. Hal ini berarti peningkatan biaya promosi, dengan variabel lain dianggap tetap, diikuti oleh peningkatan nilai prediksi penjualan. Model kemudian digunakan untuk memperkirakan nilai penjualan pada data yang belum digunakan saat proses pelatihan.

Interpretasi harus disesuaikan dengan hasil model dan dataset yang benar-benar digunakan.

13. Menyimpan Hasil Prediksi

Hasil perbandingan nilai aktual dan prediksi dapat disimpan dalam file CSV.

```
hasil_prediksi.to_csv(
    'hasil_prediksi_regresi.csv',
    index=False
)
```

Jika diperlukan, file dapat diunduh dari Google Colab.

```
from google.colab import files
files.download('hasil_prediksi_regresi.csv')
```

6. Tugas Praktikum

Tugas praktikum bertujuan mengukur kemampuan mahasiswa dalam membangun model regresi, membaca koefisien, serta menginterpretasikan hasil prediksi sesuai konteks permasalahan bisnis. Hal ini sejalan dengan indikator pertemuan ke-7 pada RPS, yaitu ketepatan membangun model regresi, membaca koefisien, dan menginterpretasikan hasil prediksi.

1. Deskripsi Tugas

Gunakan dataset yang memiliki variabel target numerik, kemudian bangun model Linear Regression menggunakan Python dan `scikit-learn`.

Mahasiswa diminta menyelesaikan seluruh proses analisis mulai dari pemeriksaan dataset hingga interpretasi hasil model.

2. Ketentuan Tugas

Dalam notebook Google Colab, kerjakan kegiatan berikut:

- a) tampilkan dan jelaskan informasi dasar dataset yang digunakan;
- b) identifikasi variabel yang berperan sebagai fitur dan target;
- c) jelaskan alasan pemilihan target berdasarkan konteks masalah;
- d) lakukan pemeriksaan terhadap data hilang dan tipe data;
- e) lakukan penanganan data apabila masih terdapat masalah yang menghambat proses pemodelan;
- f) bagi dataset menjadi data latih dan data uji;
- g) bangun model menggunakan `LinearRegression`;
- h) latih model menggunakan data latih;

- i) tampilkan nilai intercept dan koefisien setiap fitur;
- j) jelaskan arti koefisien yang diperoleh;
- k) lakukan prediksi menggunakan data uji;
- l) buat tabel perbandingan antara nilai aktual dan nilai prediksi;
- m) buat visualisasi yang menunjukkan hubungan antara nilai aktual dan nilai prediksi;
- n) lakukan minimal satu prediksi menggunakan data baru;
- o) tuliskan interpretasi hasil model dalam konteks permasalahan yang dianalisis.

3. Pertanyaan Analisis

Jawab pertanyaan berikut pada bagian akhir notebook:

- a) Apa permasalahan yang ingin diselesaikan menggunakan regresi?
- b) Apa variabel target yang diprediksi?
- c) Fitur apa saja yang digunakan untuk menghasilkan prediksi?
- d) Mengapa variabel tersebut dipilih sebagai fitur?
- e) Berapa nilai intercept model?
- f) Berapa nilai koefisien masing-masing fitur?
- g) Fitur mana yang memiliki koefisien positif dan negatif?
- h) Bagaimana interpretasi setiap koefisien dalam konteks dataset?
- i) Apakah hasil prediksi secara visual cenderung mengikuti nilai aktual?
- j) Berikan satu contoh hasil prediksi dan jelaskan maknanya.
- k) Apa manfaat hasil prediksi tersebut apabila digunakan dalam pengambilan keputusan bisnis?
- l) Apa keterbatasan yang Anda temukan dari model regresi yang dibangun?

4. Format Notebook

Notebook disusun secara sistematis dengan struktur:

A. Identitas Mahasiswa

Nama, NIM, kelas, dan judul praktikum.

B. Deskripsi Dataset

Sumber data, jumlah data, atribut, dan konteks permasalahan.

C. Persiapan Data

Pemeriksaan dataset dan penentuan fitur serta target.

D. Pembagian Dataset

Proses pembentukan data latih dan data uji.

E. Pemodelan Regresi

Pembuatan dan pelatihan model.

F. Koefisien Model

Nilai intercept, koefisien, dan interpretasinya.

G. Prediksi

Hasil prediksi terhadap data uji dan data baru.

H. Visualisasi

Visualisasi nilai aktual dan nilai prediksi.

I. Interpretasi dan Pembahasan

Pembahasan hasil model dalam konteks permasalahan.

J. Kesimpulan

Ringkasan hasil praktikum.

5. Ketentuan Dokumentasi

Setiap bagian kode harus disertai penjelasan menggunakan Markdown. Mahasiswa tidak cukup hanya menampilkan keluaran program, tetapi perlu menjelaskan apa yang dilakukan dan makna dari hasil yang diperoleh.

Contoh:

```
### Interpretasi Koefisien
Koefisien variabel Biaya_Promosi bernilai positif
sebesar 2,45.
Artinya, dengan variabel lainnya dianggap tetap,
setiap peningkatan
satu satuan Biaya_Promosi diperkirakan meningkatkan
nilai Penjualan
sebesar 2,45 satuan.
```

Nilai dan interpretasi harus menyesuaikan hasil pemodelan masing-masing mahasiswa.

6. Produk yang Dikumpulkan

Mahasiswa mengumpulkan:

- a) notebook Google Colab dalam format .ipynb;
- b) dataset yang digunakan apabila dataset berasal dari sumber eksternal;
- c) file hasil prediksi apabila dihasilkan selama praktikum; dan
- d) penjelasan hasil analisis yang telah ditulis di dalam notebook.

Pengumpulan dilakukan melalui LMS Edlink sesuai batas waktu yang ditentukan.

7. Kriteria Penilaian

Tugas praktikum dinilai berdasarkan:

Aspek	Indikator
Persiapan data	Ketepatan membaca, memeriksa, dan menentukan variabel
Pemodelan	Kebenaran proses pembangunan model regresi
Implementasi	Ketepatan penggunaan Python dan scikit-learn
Koefisien	Ketepatan membaca dan menjelaskan koefisien
Prediksi	Ketepatan menghasilkan dan menyajikan hasil prediksi
Interpretasi	Kemampuan menghubungkan keluaran model dengan konteks bisnis
Dokumentasi	Kerapian notebook, penjelasan kode, dan keterlacakan proses

Penilaian menekankan **kebenaran model, interpretasi hasil, dan relevansi terhadap masalah bisnis**, sesuai bentuk penilaian pada RPS untuk pertemuan ke-7.

7. Latihan Mandiri

Latihan mandiri bertujuan memperkuat pemahaman mahasiswa terhadap konsep dan implementasi regresi linear melalui eksplorasi tambahan di luar langkah praktikum utama.

1. Latihan 1 — Regresi dengan Satu Fitur

Pilih satu variabel numerik sebagai fitur dan satu variabel numerik sebagai target.

Lakukan langkah berikut:

- a) tentukan satu variabel fitur (X);
- b) tentukan satu variabel target (y);
- c) bagi data menjadi data latih dan data uji;
- d) bangun model LinearRegression;
- e) tampilkan nilai intercept dan koefisien;
- f) lakukan prediksi pada data uji;
- g) buat visualisasi hubungan antara fitur dan target;
- h) tampilkan garis regresi pada grafik;
- i) jelaskan makna koefisien yang diperoleh.

Contoh visualisasi:

```
plt.scatter(X_test, y_test)
plt.plot(X_test, y_pred)
plt.xlabel('Fitur')
plt.ylabel('Target')
plt.title('Regresi Linear Sederhana')
plt.show()
```

Pastikan bentuk X_test sesuai dengan kode yang digunakan.

2. Latihan 2 — Regresi dengan Beberapa Fitur

Gunakan minimal dua variabel fitur numerik untuk memprediksi satu target numerik.

Contoh:

```
X = df[['Fitur_1', 'Fitur_2']]
y = df['Target']
```

Kemudian:

- a) bangun model regresi;
- b) tampilkan koefisien setiap fitur;
- c) susun tabel koefisien;
- d) identifikasi fitur dengan koefisien positif dan negatif;

- e) jelaskan bagaimana masing-masing fitur berkaitan dengan nilai prediksi target.

Contoh tabel:

```
koefisien = pd.DataFrame({
    'Fitur': X.columns,
    'Koefisien': model.coef_
})

koefisien
```

3. Latihan 3 — Mengubah Proporsi Data Uji

Bangun model menggunakan beberapa proporsi data uji, misalnya:

- a) 20%;
- b) 25%;
- c) 30%.

Contoh:

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.3,
    random_state=42
)
```

Bandingkan hasil prediksi yang dihasilkan.

Jelaskan:

- a) apakah jumlah data latih berubah;
- b) apakah jumlah data uji berubah;
- c) apakah koefisien model berubah;
- d) mengapa pembagian data dapat memengaruhi model yang dihasilkan.

4. Latihan 4 — Mengubah Random State

Gunakan nilai `random_state` yang berbeda, misalnya:

```
random_state=10
random_state=42
random_state=100
```

Bandungkan hasil model yang diperoleh.

Diskusikan mengapa pembagian data yang berbeda dapat menghasilkan koefisien dan prediksi yang berbeda.

5. Latihan 5 — Prediksi Data Baru

Buat minimal tiga data baru yang belum terdapat pada dataset.

Contoh:

```
data_baru = pd.DataFrame({
    'Biaya_Promosi': [4.0, 5.5, 7.0],
    'Jumlah_Pelanggan': [150, 200, 250]
})
hasil_baru = model.predict(data_baru)
```

Tampilkan hasil prediksi:

```
data_baru['Prediksi'] = hasil_baru
data_baru
```

Kemudian jelaskan arti setiap prediksi dalam konteks permasalahan bisnis.

6. Latihan 6 — Analisis Koefisien

Gunakan model yang telah dibangun dan jawab pertanyaan berikut:

- a) fitur mana yang memiliki koefisien terbesar?
- b) fitur mana yang memiliki koefisien terkecil?
- c) adakah fitur yang memiliki koefisien negatif?
- d) bagaimana interpretasi koefisien positif?
- e) bagaimana interpretasi koefisien negatif?
- f) apakah koefisien yang lebih besar selalu berarti fitur tersebut lebih penting?

Pada pertanyaan terakhir, mahasiswa perlu berhati-hati dalam membuat kesimpulan karena nilai koefisien juga dipengaruhi oleh skala variabel.

7. Latihan 7 — Membandingkan Aktual dan Prediksi

Buat tabel perbandingan:

```
perbandingan = pd.DataFrame({
    'Aktual': y_test.values,
    'Prediksi': y_pred
})
perbandingan['Selisih'] = (
    perbandingan['Aktual'] -
    perbandingan['Prediksi']
)
perbandingan.head(10)
```

Kemudian:

- a) temukan prediksi dengan selisih paling kecil;
- b) temukan prediksi dengan selisih paling besar;
- c) jelaskan kemungkinan penyebab perbedaan antara nilai aktual dan prediksi.

8. Pertayaan Refleksi

Setelah menyelesaikan latihan, jawablah secara singkat:

- 1) Apa perbedaan utama antara klasifikasi dan regresi?
- 2) Mengapa regresi digunakan ketika target berupa nilai numerik?
- 3) Apa fungsi `fit()` pada model regresi?
- 4) Apa fungsi `predict()`?
- 5) Apa arti `intercept`?
- 6) Apa arti koefisien regresi?
- 7) Mengapa dataset perlu dibagi menjadi data latih dan data uji?
- 8) Apa yang terjadi jika model hanya diuji menggunakan data yang digunakan untuk pelatihan?
- 9) Mengapa hasil regresi perlu diinterpretasikan dalam konteks permasalahan?
- 10) Dalam kasus bisnis apa Anda akan memilih regresi sebagai metode analisis?

Latihan mandiri ini tetap diarahkan pada capaian pertemuan ke-7 dalam RPS, yaitu kemampuan membangun model regresi, membaca koefisien, serta menginterpretasikan hasil prediksi pada data bisnis.

8. Evaluasi Praktikum

Evaluasi praktikum dilakukan untuk mengukur ketercapaian kemampuan mahasiswa dalam menerapkan regresi untuk menyelesaikan permasalahan prediksi numerik pada data bisnis. Penilaian difokuskan pada kebenaran model, kemampuan membaca koefisien, interpretasi hasil prediksi, serta relevansi hasil terhadap permasalahan bisnis, sesuai indikator penilaian pada pertemuan ke-7 dalam RPS.

1. Bentuk Evaluasi

Evaluasi dilakukan melalui beberapa komponen berikut:

- a) **Notebook praktikum**, yang memuat seluruh proses analisis dan pemodelan.
- b) **Hasil implementasi model regresi**, termasuk proses pelatihan dan prediksi.
- c) **Interpretasi koefisien**, untuk melihat pemahaman mahasiswa terhadap hubungan antara fitur dan target.
- d) **Analisis hasil prediksi**, melalui perbandingan nilai aktual dan prediksi.
- e) **Jawaban pertanyaan analisis dan refleksi**, untuk mengukur pemahaman konsep.
- f) **Kualitas dokumentasi**, mencakup kerapian kode, penjelasan, dan keterlacakan proses.

2. Rubrik Penilaian

No.	Aspek Penilaian	Bobot
1	Ketepatan memahami dan menjelaskan permasalahan regresi	10%
2	Ketepatan menentukan fitur dan target	10%
3	Ketepatan menyiapkan dan membagi dataset	15%
4	Kebenaran implementasi model LinearRegression	20%
5	Ketepatan membaca <i>intercept</i> dan koefisien	15%
6	Ketepatan menghasilkan dan menyajikan prediksi	10%
7	Kedalaman interpretasi hasil dalam konteks bisnis	15%
8	Kerapian dan kelengkapan dokumentasi notebook	5%
	Total	100%

3. Kriteria Penilaian

Sangat Baik

Mahasiswa mampu membangun model regresi secara benar, menjelaskan koefisien secara tepat, menghasilkan prediksi, dan menghubungkan hasil model dengan konteks permasalahan secara logis serta terdokumentasi dengan baik.

Baik

Mahasiswa mampu membangun model dengan benar dan melakukan interpretasi, tetapi masih terdapat kekurangan kecil pada penjelasan atau dokumentasi.

Cukup

Model dapat dijalankan, tetapi masih terdapat kekeliruan pada pemilihan fitur, pembacaan koefisien, atau interpretasi hasil.

Kurang

Implementasi model tidak lengkap atau terdapat kesalahan mendasar dalam proses pemodelan dan interpretasi.

4. Indikator Ketercapaian

Mahasiswa dinyatakan telah mencapai tujuan praktikum apabila mampu:

- 1) menentukan fitur dan target secara tepat;
- 2) membangun model regresi linear menggunakan scikit-learn;
- 3) membaca nilai *intercept* dan koefisien model;
- 4) menghasilkan prediksi pada data uji;
- 5) membandingkan nilai aktual dan hasil prediksi;
- 6) menjelaskan hasil model secara logis dalam konteks permasalahan bisnis;
- 7) mendokumentasikan proses analisis dalam notebook secara sistematis.

5. Catatan Evaluasi

Evaluasi pada Modul 7 belum menekankan penggunaan metrik regresi secara mendalam. Metrik seperti **MAE, MSE, RMSE, dan (R^2)** dibahas pada bagian evaluasi model sesuai urutan materi dalam RPS.

Dengan demikian, penilaian Modul 7 lebih berfokus pada kemampuan mahasiswa dalam **membangun model regresi, memahami keluaran model, dan**

menginterpretasikan hasil prediksi sebelum masuk ke tahap evaluasi performa model secara formal.

9. Ringkasan

Regresi merupakan salah satu metode supervised learning yang digunakan untuk memprediksi nilai numerik berdasarkan pola hubungan antara satu atau lebih variabel fitur dengan variabel target. Pada Modul 7, metode yang digunakan adalah Linear Regression sebagai dasar untuk memahami proses pemodelan prediktif pada data bisnis.

Tahapan utama yang dilakukan dalam praktikum meliputi:

- 1) memahami permasalahan prediksi yang akan diselesaikan;
- 2) memuat dan memeriksa dataset;
- 3) menentukan variabel fitur dan variabel target;
- 4) membagi dataset menjadi data latih dan data uji;
- 5) membangun model menggunakan LinearRegression;
- 6) melatih model menggunakan data latih;
- 7) membaca nilai intercept dan koefisien model;
- 8) menghasilkan prediksi pada data uji;
- 9) membandingkan nilai aktual dengan hasil prediksi;
- 10) melakukan prediksi terhadap data baru; dan
- 11) menginterpretasikan hasil model dalam konteks permasalahan bisnis.

Dalam regresi linear, nilai intercept menunjukkan nilai prediksi target ketika seluruh variabel fitur bernilai nol, sedangkan koefisien menunjukkan arah dan besarnya perubahan prediksi target akibat perubahan satu satuan pada suatu fitur, dengan fitur lainnya dianggap tetap.

Koefisien positif menunjukkan hubungan searah antara fitur dan target, sedangkan koefisien negatif menunjukkan hubungan yang berlawanan arah. Namun, interpretasi koefisien harus dilakukan dengan memperhatikan konteks permasalahan, satuan variabel, dan skala data.

Pembagian dataset menjadi data latih dan data uji diperlukan agar model dapat dipelajari menggunakan sebagian data dan kemudian digunakan untuk menghasilkan prediksi terhadap data yang tidak digunakan dalam proses pelatihan.

Melalui praktikum ini, mahasiswa diharapkan tidak hanya mampu menjalankan kode regresi, tetapi juga memahami bagaimana model dibangun, bagaimana keluaran model dibaca, serta bagaimana hasil prediksi diterjemahkan menjadi informasi yang relevan bagi pengambilan keputusan.

Kemampuan tersebut sesuai dengan fokus SUB-CPMK-2.2 pada RPS, yaitu kemampuan menerapkan algoritma klasifikasi dan regresi untuk menyelesaikan permasalahan prediksi pada data bisnis. Pada pertemuan ke-7, indikator pembelajaran secara khusus menekankan kemampuan membangun model regresi, membaca koefisien, dan menginterpretasikan hasil prediksi.

Referensi

1. Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
2. Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann–Elsevier.
3. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
4. McKinney, W. (2022). *Python for Data Analysis* (3rd ed.). O'Reilly Media.
5. Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O'Reilly Media.
6. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
7. Dokumentasi resmi Python sebagai referensi penggunaan bahasa pemrograman Python dalam pengolahan data.
8. Dokumentasi resmi pandas sebagai referensi manipulasi, pemeriksaan, dan pengolahan dataset.
9. Dokumentasi resmi Matplotlib sebagai referensi visualisasi data dan hasil pemodelan.
10. Dokumentasi resmi scikit-learn sebagai referensi implementasi LinearRegression, train_test_split, proses pelatihan model, dan prediksi data.

MODUL 8

CLUSTERING MENGGUNAKAN ALGORITMA K-MEANS

1. Identitas Praktikum

Judul Praktikum	: Clustering Menggunakan Algoritma K-Means
Mata Kuliah	: Data Mining dan Machine Learning
Bobot Mata Kuliah	: 3 SKS
CPMK	: CPMK-2 — Mampu menyiapkan data dan menerapkan algoritma supervised serta unsupervised learning untuk menemukan pola, melakukan klasifikasi, dan menghasilkan prediksi.
Sub-CPMK	: SUB-CPMK-2.3 — Mampu menerapkan algoritma clustering dan teknik penemuan pola untuk mengidentifikasi kelompok atau hubungan dalam data.
Lingkungan Praktikum	: Google Colab / Jupyter Notebook
Library Utama	: pandas, NumPy, Matplotlib, dan scikit-learn

Sub-CPMK yang dicapai adalah:

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu menerapkan algoritma K-Means Clustering, menentukan jumlah cluster yang sesuai, serta menjelaskan karakteristik setiap cluster yang terbentuk berdasarkan data yang dianalisis.

Praktikum diarahkan pada penggunaan pendekatan unsupervised learning, yaitu proses pembelajaran mesin terhadap data yang tidak memiliki label target. Mahasiswa akan menggunakan K-Means untuk mengelompokkan objek berdasarkan kemiripan karakteristik, kemudian menganalisis hasil pengelompokan tersebut untuk memperoleh pola atau wawasan yang dapat digunakan dalam konteks pengambilan keputusan berbasis data.

2. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

- 1) Menjelaskan konsep dasar unsupervised learning dan perbedaannya dengan supervised learning.

- 2) Menjelaskan konsep clustering dan tujuan pengelompokan data berdasarkan kemiripan karakteristik.
- 3) Menyiapkan dataset yang sesuai untuk proses clustering menggunakan algoritma K-Means.
- 4) Melakukan pemilihan fitur yang relevan dan melakukan standardisasi data sebelum proses clustering apabila diperlukan.
- 5) Menerapkan algoritma K-Means Clustering menggunakan library scikit-learn.
- 6) Menentukan jumlah cluster yang sesuai berdasarkan karakteristik data dan hasil evaluasi.
- 7) Menggunakan metode Elbow sebagai salah satu pendekatan untuk membantu menentukan jumlah cluster.
- 8) Memvisualisasikan hasil clustering untuk memahami pola pengelompokan data.
- 9) Menganalisis karakteristik setiap cluster berdasarkan nilai fitur yang digunakan.
- 10) Menginterpretasikan hasil clustering menjadi informasi atau wawasan yang relevan dengan permasalahan yang dianalisis.

Pada akhir praktikum, mahasiswa tidak hanya diharapkan mampu menjalankan algoritma K-Means, tetapi juga mampu menjelaskan mengapa suatu data berada pada cluster tertentu, bagaimana karakteristik masing-masing cluster, dan informasi apa yang dapat diperoleh dari pola pengelompokan tersebut.

3. Dasar Teori

3.1. *Unsupervised Learning*

Unsupervised learning merupakan pendekatan dalam machine learning yang digunakan untuk menemukan pola, struktur, atau hubungan pada data tanpa menggunakan label target yang telah ditentukan sebelumnya. Dalam pendekatan ini, algoritma berusaha mengenali keteraturan yang terdapat di dalam data berdasarkan karakteristik atau kemiripan antarobjek (Han, Pei, & Tong, 2022; Géron, 2023).

Berbeda dengan supervised learning yang menggunakan pasangan data input dan target selama proses pembelajaran, unsupervised learning bekerja pada data yang tidak memiliki variabel target. Tujuannya bukan untuk memprediksi kelas tertentu, melainkan untuk menemukan struktur tersembunyi yang dapat membantu memahami karakteristik data (Witten et al., 2016).

Salah satu penerapan utama unsupervised learning adalah clustering, yaitu proses mengelompokkan data berdasarkan tingkat kemiripan karakteristik. Dalam konteks Sistem Informasi dan bisnis, clustering dapat diterapkan untuk segmentasi pelanggan, pengelompokan produk, analisis perilaku pengguna, pengelompokan wilayah, dan eksplorasi pola pada data transaksi (Provost & Fawcett, 2013).

3.2. Konsep Clustering

Clustering merupakan teknik pengelompokan data yang bertujuan membagi sekumpulan objek ke dalam beberapa kelompok atau cluster berdasarkan tingkat kemiripan di antara objek tersebut. Objek yang berada dalam cluster yang sama diharapkan memiliki karakteristik yang relatif lebih serupa dibandingkan dengan objek yang berada pada cluster yang berbeda (Han, Pei, & Tong, 2022).

Secara konseptual, proses clustering berusaha memenuhi dua kondisi utama:

- 1) **Kemiripan dalam cluster tinggi**, yaitu objek-objek yang berada pada cluster yang sama memiliki karakteristik yang relatif dekat.
- 2) **Perbedaan antarcluster tinggi**, yaitu objek yang berada pada cluster berbeda memiliki karakteristik yang relatif lebih berbeda.

Berbeda dengan klasifikasi, clustering tidak memerlukan label kelas yang sudah diketahui sebelumnya. Oleh karena itu, setelah proses clustering dilakukan, setiap kelompok perlu dianalisis untuk memahami karakteristik dan makna dari cluster yang terbentuk.

3.3. Algoritma K-Means

K-Means merupakan salah satu algoritma clustering berbasis partisi yang banyak digunakan karena konsepnya relatif sederhana dan dapat diterapkan secara efisien pada berbagai jenis data numerik. Algoritma ini membagi data ke dalam sejumlah K cluster dengan menempatkan setiap objek ke cluster yang memiliki pusat atau centroid terdekat (Han, Pei, & Tong, 2022).

Nilai K menunjukkan jumlah cluster yang ingin dibentuk. Sebagai contoh, apabila ditentukan $K = 3$, algoritma akan membagi data menjadi tiga kelompok berdasarkan kedekatan objek terhadap centroid masing-masing cluster.

Secara umum, proses K-Means dilakukan melalui tahapan berikut (Géron, 2023):

- 1) Menentukan jumlah cluster K.
- 2) Menentukan centroid awal untuk setiap cluster.
- 3) Menghitung jarak setiap objek terhadap seluruh centroid.
- 4) Menempatkan setiap objek pada cluster dengan centroid terdekat.
- 5) Menghitung kembali centroid berdasarkan rata-rata anggota cluster.
- 6) Mengulangi proses penempatan objek dan pembaruan centroid hingga tidak terjadi perubahan yang berarti atau algoritma mencapai kondisi konvergen.

Proses tersebut berlangsung secara iteratif. Pada setiap iterasi, keanggotaan cluster dan posisi centroid dapat berubah hingga diperoleh pembagian kelompok yang relatif stabil.

3.4. Centroid

Centroid merupakan titik pusat yang merepresentasikan suatu cluster. Pada algoritma K-Means, centroid dihitung berdasarkan nilai rata-rata seluruh objek yang menjadi anggota cluster tersebut (Han, Pei, & Tong, 2022).

Jika sebuah cluster memiliki n data, maka centroid untuk suatu fitur dapat dihitung menggunakan:

$$C = \frac{1}{n} \sum_{i=1}^n x_i$$

dengan:

- C = nilai centroid,
- n = jumlah data dalam cluster,
- x_i = nilai data ke-(i).

Pada dataset yang memiliki lebih dari satu fitur, centroid dihitung untuk setiap fitur sehingga membentuk suatu titik dalam ruang multidimensi. Posisi centroid akan diperbarui setiap kali terjadi perubahan anggota cluster.

3.5. Jarak Euclidean

K-Means umumnya menggunakan Euclidean Distance sebagai ukuran kedekatan antara suatu objek dengan centroid cluster. Objek kemudian ditempatkan pada cluster yang memiliki jarak paling kecil terhadap objek tersebut (Han, Pei, & Tong, 2022).

Untuk dua titik (x) dan (c), jarak Euclidean dapat dihitung menggunakan:

$$d(x, c) = \sqrt{\sum_{j=1}^m (x_j - c_j)^2}$$

dengan:

- $d(x, c)$ = jarak antara data x dan centroid,
- x_j = nilai fitur ke-(j) pada suatu data,
- c_j = nilai fitur ke-(j) pada centroid,
- m = jumlah fitur.

Semakin kecil jarak suatu objek terhadap centroid, semakin besar kedekatan karakteristik objek tersebut dengan cluster yang bersangkutan.

3.6. Pentingnya Skala Data

Karena K-Means menggunakan perhitungan jarak, perbedaan skala antarfitur dapat memengaruhi hasil clustering. Fitur dengan rentang nilai yang jauh lebih besar dapat memberikan pengaruh yang lebih dominan terhadap perhitungan jarak dibandingkan fitur dengan rentang nilai yang lebih kecil (Géron, 2023).

Sebagai contoh, sebuah dataset memiliki dua fitur:

- usia dengan rentang 18–60 tahun;
- pendapatan dengan rentang Rp1.000.000–Rp20.000.000.

Apabila kedua fitur digunakan secara langsung, nilai pendapatan berpotensi mendominasi perhitungan jarak. Oleh karena itu, proses normalisasi atau standardisasi sering diperlukan sebelum menjalankan algoritma berbasis jarak (Kuhn & Johnson, 2013).

Salah satu teknik yang umum digunakan adalah standardisasi, yaitu mengubah data sehingga setiap fitur berada pada skala yang lebih sebanding. Dalam scikit-learn, standardisasi dapat dilakukan menggunakan StandardScaler (Géron, 2023):

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

Standardisasi tidak selalu wajib pada semua dataset, tetapi perlu dipertimbangkan apabila fitur memiliki satuan atau rentang nilai yang sangat berbeda.

3.7. Menentukan Jumlah Cluster

Salah satu tantangan dalam penerapan K-Means adalah menentukan nilai K atau jumlah cluster yang sesuai. Jumlah cluster yang terlalu sedikit dapat menyebabkan beberapa kelompok dengan karakteristik berbeda tergabung menjadi satu, sedangkan jumlah cluster yang terlalu banyak dapat menghasilkan pengelompokan yang terlalu terfragmentasi (Han, Pei, & Tong, 2022).

Salah satu pendekatan yang umum digunakan untuk membantu menentukan jumlah cluster adalah Elbow Method. Metode ini membandingkan sejumlah nilai K dengan melihat perubahan nilai inertia atau Within-Cluster Sum of Squares (WCSS).

WCSS mengukur jumlah kuadrat jarak setiap objek terhadap centroid cluster tempat objek tersebut berada. Secara umum dapat dinyatakan sebagai:

$$WCSS = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

dengan:

- K = jumlah cluster,
- C_k = cluster ke-(k),
- x_i = data atau objek ke- i yang berada pada cluster C_k ,
- μ_k = centroid dari cluster ke- k ,
- $\|x_i - \mu_k\|^2$ = kuadrat jarak antara data x_i dan centroid μ_k .

Ketika jumlah cluster meningkat, nilai WCSS umumnya akan menurun. Elbow Method mencari titik ketika penurunan tersebut mulai melambat sehingga membentuk pola menyerupai siku atau elbow. Titik tersebut dapat digunakan sebagai salah satu pertimbangan dalam memilih jumlah cluster (Géron, 2023).

Meskipun demikian, Elbow Method tidak selalu menghasilkan titik siku yang sangat jelas. Oleh karena itu, pemilihan jumlah cluster tetap perlu mempertimbangkan karakteristik dataset dan tujuan analisis.

3.8. Interpretasi Hasil Clustering

Proses clustering tidak berhenti setelah setiap objek memperoleh label cluster. Tahapan penting berikutnya adalah **menginterpretasikan karakteristik masing-masing cluster** agar hasil pemodelan memiliki makna dalam konteks permasalahan yang dianalisis (Provost & Fawcett, 2013). Interpretasi dapat dilakukan dengan membandingkan nilai rata-rata, distribusi, atau karakteristik penting dari setiap fitur pada masing-masing cluster.

Sebagai contoh, hasil segmentasi pelanggan menghasilkan tiga cluster:

Cluster	Pendapatan	Frekuensi Belanja	Interpretasi
Cluster 0	Tinggi	Tinggi	Pelanggan bernilai tinggi
Cluster 1	Sedang	Sedang	Pelanggan reguler
Cluster 2	Rendah	Rendah	Pelanggan dengan aktivitas rendah

Hasil tersebut dapat digunakan untuk memahami perbedaan karakteristik pelanggan dan menjadi dasar untuk menyusun strategi yang lebih sesuai bagi setiap kelompok. Alur analisis clustering dalam praktikum ini digambarkan pada Gambar 3.



Gambar 6. Alur Analisis Clustering

Alur pada Gambar 6 menekankan bahwa keberhasilan clustering tidak hanya ditentukan oleh proses menjalankan algoritma, tetapi juga oleh kualitas persiapan data, pemilihan jumlah cluster, dan kemampuan menginterpretasikan pola yang dihasilkan.

4. Persiapan Praktikum

Sebelum melakukan praktikum clustering menggunakan algoritma K-Means, mahasiswa perlu menyiapkan lingkungan kerja, library, dan dataset yang akan digunakan. Tahap persiapan ini penting agar proses analisis dapat dilakukan secara sistematis dan hasil clustering dapat diinterpretasikan dengan baik.

4.1. Lingkungan Praktikum

Praktikum dapat dilakukan menggunakan **Google Colab** atau **Jupyter Notebook**. Google Colab direkomendasikan karena dapat dijalankan langsung melalui browser tanpa memerlukan instalasi Python secara lokal.

Library utama yang digunakan dalam praktikum meliputi:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

Fungsi masing-masing library Clustering diuraikan pada Tabel

Library	Fungsi
pandas	Membaca, mengelola, dan menganalisis dataset
NumPy	Mendukung operasi numerik
Matplotlib	Membuat visualisasi data dan hasil clustering
StandardScaler	Melakukan standardisasi fitur
KMeans	Membangun model clustering K-Means
silhouette_score	Membantu mengevaluasi kualitas cluster

4.2. Dataset Praktikum

Pada praktikum ini digunakan dataset sederhana mengenai karakteristik pelanggan. Dataset dirancang agar mahasiswa dapat memahami bagaimana algoritma K-Means mengelompokkan pelanggan berdasarkan kesamaan karakteristik.

Contoh atribut yang digunakan adalah:

Atribut	Tipe Data	Keterangan
CustomerID	Integer	Identitas pelanggan
Age	Integer	Usia pelanggan
AnnualIncome	Numerik	Pendapatan tahunan pelanggan
SpendingScore	Numerik	Skor tingkat pengeluaran pelanggan

Variabel `CustomerID` berfungsi sebagai identitas sehingga tidak digunakan sebagai fitur dalam pembentukan cluster. Fitur utama yang digunakan dalam praktikum adalah **AnnualIncome** dan **SpendingScore**.

Contoh data:

CustomerID	Age	AnnualIncome	SpendingScore
1	19	15	39
2	21	15	81
3	20	16	6
4	23	16	77
5	31	17	40
6	22	17	76
7	35	18	6
8	23	18	94
9	64	19	3
10	30	19	72

Dataset dapat disimpan dalam format `.csv`, misalnya: `customer_clustering.csv`

4.3. Memuat Dataset

Dataset dapat dibaca menggunakan `pandas` dengan perintah berikut:

```
df = pd.read_csv("customer_clustering.csv")
df.head()
```

Jika menggunakan Google Colab, dataset dapat diunggah terlebih dahulu menggunakan:

```
from google.colab import files
uploaded = files.upload()
```

Setelah dataset berhasil dimuat, lakukan pemeriksaan awal:

```
df.shape
df.info()
df.describe()
```

Pemeriksaan tersebut bertujuan untuk mengetahui jumlah data, jenis setiap atribut, nilai statistik dasar, serta kemungkinan adanya permasalahan pada dataset.

4.4. Pemeriksaan Kualitas Data

Sebelum melakukan clustering, mahasiswa perlu memastikan bahwa dataset berada dalam kondisi yang layak untuk dianalisis.

Periksa data hilang menggunakan:

```
df.isnull().sum()
```

Periksa data duplikat:

```
df.duplicated().sum()
```

Apabila ditemukan data hilang, duplikasi, kesalahan tipe data, atau nilai yang tidak masuk akal, permasalahan tersebut perlu ditangani sebelum pemodelan.

4.5. Pemilihan Fitur

Tidak semua atribut harus digunakan dalam proses clustering. Pemilihan fitur harus mempertimbangkan tujuan analisis.

Pada praktikum ini digunakan dua fitur:

```
X = df[["AnnualIncome", "SpendingScore"]]
X.head()
```

Pemilihan kedua fitur tersebut memungkinkan mahasiswa mengamati pengelompokan pelanggan berdasarkan kombinasi antara **kemampuan ekonomi** dan **perilaku pengeluaran**.

Visualisasi awal dapat dilakukan menggunakan:

```
plt.scatter(
    X["AnnualIncome"],
    X["SpendingScore"]
)

plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Distribusi Data Pelanggan")
plt.show()
```

Visualisasi awal membantu mahasiswa melihat apakah terdapat kecenderungan pengelompokan tertentu sebelum algoritma K-Means diterapkan.

4.6. Standardisasi Data

Karena K-Means merupakan algoritma berbasis jarak, fitur yang memiliki skala berbeda dapat memberikan pengaruh yang tidak proporsional terhadap pembentukan cluster.

Oleh karena itu, data perlu distandardisasi ketika terdapat perbedaan skala yang cukup besar.

Standardisasi dilakukan menggunakan `StandardScaler`:

```
scaler = StandardScaler()  
X_scaled = scaler.fit_transform(X)
```

Hasil standardisasi dapat diperiksa menggunakan:

```
X_scaled[:5]
```

Setelah proses ini, data telah siap digunakan dalam proses penentuan jumlah cluster dan pemodelan K-Means.

4.7. Hasil yang Diharapkan dari Tahap Persiapan

Setelah menyelesaikan tahap persiapan, mahasiswa seharusnya telah memiliki:

- 1) lingkungan Google Colab atau Jupyter Notebook yang siap digunakan;
- 2) dataset yang berhasil dimuat;
- 3) pemahaman mengenai struktur dan karakteristik dataset;
- 4) dataset yang telah diperiksa kualitasnya;
- 5) fitur yang relevan untuk proses clustering; dan
- 6) data numerik yang telah distandardisasi dan siap digunakan untuk pemodelan K-Means.

Tahap berikutnya adalah melaksanakan proses clustering secara bertahap, mulai dari eksplorasi data, penentuan jumlah cluster menggunakan Elbow Method, pembangunan model K-Means, visualisasi cluster, hingga interpretasi karakteristik setiap kelompok.

5. Langkah Praktikum

Jalankan kode berikut untuk mengimpor library yang digunakan:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

Library tersebut digunakan untuk membaca data, melakukan pengolahan numerik, visualisasi, standardisasi, clustering, dan evaluasi hasil cluster.

5.2 Memuat Dataset

Unggah dataset `customer_clustering.csv` ke Google Colab, kemudian jalankan:

```
df = pd.read_csv("customer_clustering.csv")
df.head()
```

Perintah `head()` digunakan untuk menampilkan lima baris pertama sehingga struktur dataset dapat diperiksa.

Selanjutnya, periksa ukuran dataset:

```
df.shape
```

Periksa informasi setiap atribut:

```
df.info()
```

Kemudian tampilkan statistik deskriptif:

```
df.describe()
```

Dari hasil tersebut, amati jumlah data, tipe atribut, nilai minimum, maksimum, rata-rata, dan distribusi awal setiap fitur.

5.3 Memeriksa Kualitas Data

Periksa apakah terdapat nilai kosong:

```
df.isnull().sum()
```

Kemudian periksa data duplikat:

```
df.duplicated().sum()
```

Jika ditemukan data duplikat, data dapat dihapus dengan:

```
df = df.drop_duplicates()
```

Pastikan atribut yang akan digunakan dalam clustering memiliki tipe data numerik dan tidak mengandung nilai kosong yang dapat mengganggu proses pemodelan.

5.4 Memilih Fitur untuk Clustering

Pada praktikum ini digunakan dua fitur utama, yaitu:

- a. AnnualIncome
- b. SpendingScore

Pilih kedua fitur tersebut:

```
X = df[["AnnualIncome", "SpendingScore"]]
X.head()
```

CustomerID tidak digunakan karena hanya berfungsi sebagai identitas pelanggan, bukan sebagai karakteristik yang relevan untuk menentukan kemiripan antarobjek.

5.5 Memvisualisasikan Distribusi Data

Sebelum melakukan clustering, tampilkan distribusi data menggunakan scatter plot:

```
plt.figure(figsize=(8, 5))
plt.scatter(
    X["AnnualIncome"],
    X["SpendingScore"]
)
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Distribusi Data Pelanggan")
plt.show()
```

Amati pola penyebaran data.

Pertanyaan yang dapat diperhatikan:

- 1) Apakah terlihat kelompok-kelompok tertentu?
- 2) Apakah terdapat data yang sangat jauh dari kelompok lain?
- 3) Apakah hubungan antara pendapatan dan skor pengeluaran terlihat jelas?

Visualisasi awal ini belum menunjukkan cluster secara resmi, tetapi dapat memberikan gambaran mengenai struktur data.

5.6 Melakukan Standardisasi

Karena K-Means menggunakan jarak sebagai dasar pengelompokan, data distandardisasi menggunakan StandardScaler.

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Tampilkan lima data pertama setelah standardisasi:

```
X_scaled[:5]
```

Untuk memudahkan pembacaan, hasil standardisasi dapat diubah kembali menjadi DataFrame:

```
X_scaled_df = pd.DataFrame(
    X_scaled,
    columns=["AnnualIncome", "SpendingScore"]
)

X_scaled_df.head()
```

Setelah standardisasi, masing-masing fitur memiliki skala yang lebih sebanding sehingga tidak ada fitur yang terlalu dominan dalam perhitungan jarak.

5.7 Menentukan Jumlah Cluster dengan Elbow Method

Sebelum membangun model K-Means, jumlah cluster perlu ditentukan.

Pada tahap ini, mahasiswa akan mencoba beberapa nilai K, misalnya dari 1 sampai 10.

```
inertia = []
for k in range(1, 11):
    model = KMeans(
        n_clusters=k,
        random_state=42,
        n_init=10
    )

    model.fit(X_scaled)
    inertia.append(model.inertia_)
```

Nilai `inertia` menunjukkan jumlah kuadrat jarak setiap data terhadap centroid cluster tempat data tersebut berada.

Selanjutnya, visualisasikan nilai inertia:

```
plt.figure(figsize=(8, 5))
plt.plot(
    range(1, 11),
    inertia,
    marker="o"
)

plt.xlabel("Jumlah Cluster (K)")
plt.ylabel("Inertia")
plt.title("Elbow Method")
plt.xticks(range(1, 11))
plt.show()
```

Amati grafik yang dihasilkan.

Pilih nilai K pada bagian grafik yang membentuk pola seperti siku (*elbow*), yaitu ketika penurunan inertia mulai melambat.

Sebagai contoh, apabila titik siku terlihat pada $K = 5$, maka jumlah cluster yang digunakan adalah: `k_optimal = 5`

Nilai tersebut tidak harus selalu sama untuk setiap dataset. Mahasiswa harus menentukan nilai berdasarkan grafik yang dihasilkan.

5.8 Mengevaluasi Kandidat Cluster dengan Silhouette Score

Selain Elbow Method, kualitas cluster juga dapat diamati menggunakan **Silhouette Score**.

Silhouette Score digunakan untuk melihat seberapa baik suatu objek berada pada cluster-nya dibandingkan dengan cluster lain.

Nilainya berada pada rentang:

$$-1 \leq s \leq 1$$

Secara umum:

- nilai mendekati **1** menunjukkan cluster yang semakin baik;
- nilai mendekati **0** menunjukkan cluster saling berdekatan atau tumpang tindih;

- nilai negatif menunjukkan kemungkinan objek ditempatkan pada cluster yang kurang tepat.

Hitung Silhouette Score untuk beberapa nilai K:

```
silhouette_scores = []

for k in range(2, 11):
    model = KMeans(
        n_clusters=k,
        random_state=42,
        n_init=10
    )

    labels = model.fit_predict(X_scaled)

    score = silhouette_score(
        X_scaled,
        labels
    )

    silhouette_scores.append(score)
```

Visualisasikan:

```
plt.figure(figsize=(8, 5))

plt.plot(
    range(2, 11),
    silhouette_scores,
    marker="o"
)

plt.xlabel("Jumlah Cluster (K)")
plt.ylabel("Silhouette Score")
plt.title("Silhouette Score untuk Beberapa Nilai K")
plt.xticks(range(2, 11))
plt.show()
```

Bandingkan hasil Elbow Method dan Silhouette Score sebelum menentukan jumlah cluster yang digunakan.

5.9 Membangun Model K-Means

Setelah jumlah cluster ditentukan, buat model K-Means.

Sebagai contoh digunakan:

```
k_optimal = 5
Bangun model:
kmeans = KMeans(
    n_clusters=k_optimal,
    random_state=42,
    n_init=10
)
```

Lakukan clustering:

```
cluster_labels = kmeans.fit_predict(X_scaled)
```

`fit_predict()` melakukan dua proses sekaligus, yaitu:

1. mempelajari pola data dan membentuk centroid;
2. memberikan label cluster pada setiap data.

Label cluster biasanya dimulai dari 0.

5.10 Menambahkan Label Cluster ke Dataset

Tambahkan hasil clustering ke dataset:

```
df["Cluster"] = cluster_labels
```

Tampilkan beberapa data:

```
df.head()
```

Sekarang setiap pelanggan telah memiliki label cluster.

Sebagai contoh:

CustomerID	AnnualIncome	SpendingScore	Cluster
1	15	39	3
2	15	81	4
3	16	6	3
4	16	77	4
5	17	40	3

Nomor cluster tidak menunjukkan peringkat. Misalnya, Cluster 4 tidak berarti lebih baik daripada Cluster 1. Nomor hanya berfungsi sebagai label kelompok.

5.11 Menghitung Silhouette Score Model

Setelah model akhir dibangun, hitung Silhouette Score:

```
score = silhouette_score(
    X_scaled,
    cluster_labels
)
print("Silhouette Score:", score)
```

Gunakan nilai tersebut sebagai salah satu indikator untuk menilai seberapa baik struktur cluster yang dihasilkan. Interpretasi tidak sebaiknya hanya bergantung pada satu angka. Hasil visualisasi dan karakteristik setiap cluster juga perlu dianalisis.

5.12 Melihat Centroid

Centroid hasil K-Means dapat diperoleh dengan:

```
kmeans.cluster_centers_
```

Nilai tersebut masih berada dalam skala hasil standardisasi.

Agar lebih mudah diinterpretasikan, kembalikan centroid ke skala asli:

```
centroids_original = scaler.inverse_transform(
    kmeans.cluster_centers_
)

centroids_df = pd.DataFrame(
    centroids_original,
    columns=["AnnualIncome", "SpendingScore"]
)

centroids_df
```

Tabel centroid menunjukkan nilai rata-rata representatif dari setiap cluster.

Misalnya:

Cluster	AnnualIncome	SpendingScore
0	55.3	49.5
1	86.5	82.1
2	88.2	17.3
3	26.3	20.9
4	25.7	79.4

Nilai tersebut nantinya digunakan untuk membantu menginterpretasikan karakteristik setiap kelompok.

5.13 Memvisualisasikan Hasil Clustering

Visualisasikan hasil clustering berdasarkan dua fitur yang digunakan:

```
plt.figure(figsize=(9, 6))

plt.scatter(
    X["AnnualIncome"],
    X["SpendingScore"],
    c=df["Cluster"],
    s=50
)

plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Hasil K-Means Clustering")
plt.show()
Agar centroid ikut ditampilkan:
plt.figure(figsize=(9, 6))

plt.scatter(
    X["AnnualIncome"],
    X["SpendingScore"],
```

```

        c=df["Cluster"],
        s=50
    )

plt.scatter(
    centroids_df["AnnualIncome"],
    centroids_df["SpendingScore"],
    marker="X",
    s=250,
    label="Centroid"
)

plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Hasil K-Means Clustering dan Centroid")
plt.legend()
plt.show()

```

Perhatikan posisi masing-masing cluster dan centroid.

5.14 Menghitung Jumlah Anggota Setiap Cluster

Untuk mengetahui jumlah pelanggan pada setiap kelompok:

```
df["Cluster"].value_counts().sort_index()
```

Hasil ini membantu melihat apakah ukuran masing-masing cluster relatif seimbang atau terdapat cluster dengan anggota sangat sedikit.

5.15 Menganalisis Karakteristik Cluster

Hitung nilai rata-rata setiap fitur berdasarkan cluster:

```

cluster_summary = df.groupby("Cluster")[
    ["Age", "AnnualIncome", "SpendingScore"]
].mean()

cluster_summary

```

Jika Age tidak ingin digunakan dalam interpretasi, gunakan:

```
cluster_summary = df.groupby("Cluster") [
    ["AnnualIncome", "SpendingScore"]
].mean()

cluster_summary
```

Hasil tersebut menjadi dasar untuk memahami karakteristik masing-masing cluster.

Sebagai contoh:

Cluster	Annual Income	Spending Score	Interpretasi
0	Sedang	Sedang	Pelanggan reguler
1	Tinggi	Tinggi	Pelanggan bernilai tinggi
2	Tinggi	Rendah	Berpotensi meningkatkan pengeluaran
3	Rendah	Rendah	Pelanggan bernilai relatif rendah
4	Rendah	Tinggi	Pelanggan aktif dengan pendapatan relatif rendah

Nama atau interpretasi cluster tidak diberikan secara otomatis oleh K-Means. Interpretasi harus dibuat berdasarkan karakteristik data dan konteks permasalahan.

5.16 Menyusun Insight

Tahap terakhir adalah mengubah hasil clustering menjadi insight yang relevan.

Sebagai contoh:

Cluster dengan pendapatan tinggi dan skor pengeluaran tinggi dapat dianggap sebagai kelompok pelanggan bernilai tinggi. Kelompok ini dapat menjadi prioritas dalam program loyalitas dan layanan premium.

Cluster dengan pendapatan tinggi tetapi skor pengeluaran rendah memiliki kemampuan ekonomi yang tinggi, tetapi aktivitas belanjanya relatif rendah. Kelompok tersebut dapat menjadi sasaran strategi promosi untuk meningkatkan keterlibatan.

Cluster dengan pendapatan rendah dan skor pengeluaran tinggi menunjukkan pelanggan yang relatif aktif melakukan pembelian meskipun memiliki tingkat pendapatan lebih rendah. Insight harus dibuat berdasarkan hasil analisis, bukan hanya berdasarkan nomor cluster.

5.17 Menyimpan Hasil Clustering

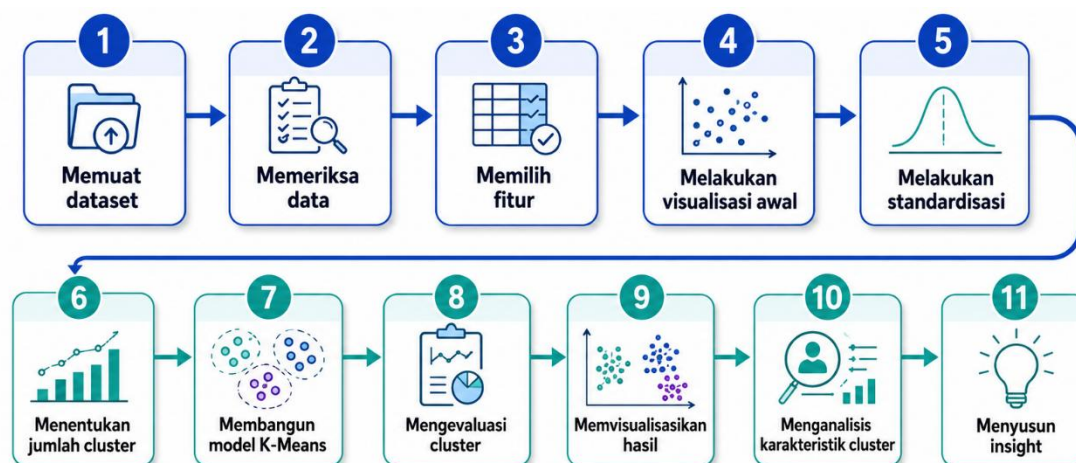
Dataset yang telah dilengkapi dengan label cluster dapat disimpan kembali:

```
df.to_csv(
    "hasil_clustering_kmeans.csv",
    index=False
)
```

File tersebut berisi dataset asli beserta label cluster yang dihasilkan.

5.18 Ringkasan Proses Praktikum

Setelah seluruh langkah selesai, mahasiswa telah menjalankan alur clustering secara lengkap, seperti digambarkan pada Gambar 7.



Gambar 7. Alur Praktikum Clustering K-Means

Mahasiswa perlu memahami bahwa tujuan utama clustering bukan sekadar menghasilkan nomor cluster, tetapi menemukan struktur yang bermakna dari data dan menginterpretasikannya menjadi informasi yang dapat mendukung pengambilan keputusan.

6. Latihan Praktikum

Latihan berikut bertujuan memperkuat pemahaman mahasiswa terhadap proses clustering menggunakan algoritma K-Means. Mahasiswa diminta tidak hanya menjalankan kode, tetapi juga mengamati perubahan hasil, membandingkan beberapa pilihan parameter, dan memberikan interpretasi terhadap cluster yang terbentuk.

Latihan 1 — Eksplorasi Dataset

Gunakan dataset `customer_clustering.csv` yang telah digunakan pada langkah praktikum.

Lakukan pemeriksaan berikut:

```
df.head()
df.info()
df.describe()
df.isnull().sum()
df.duplicated().sum()
```

Kemudian jawab pertanyaan berikut:

- 1) Berapa jumlah baris dan kolom pada dataset?
- 2) Atribut apa saja yang tersedia?
- 3) Atribut mana yang bersifat numerik?
- 4) Apakah terdapat nilai yang hilang?
- 5) Apakah terdapat data duplikat?
- 6) Atribut apa yang sebaiknya tidak digunakan sebagai fitur clustering? Jelaskan alasannya.

Latihan 2 — Membandingkan Pemilihan Fitur

Pada praktikum utama digunakan fitur:

```
X = df[["AnnualIncome", "SpendingScore"]]
```

Sekarang buat kombinasi fitur lain:

```
X2 = df[["Age", "AnnualIncome"]]
```

Lakukan visualisasi:

```
plt.figure(figsize=(8, 5))
plt.scatter(
    X2["Age"],
    X2["AnnualIncome"]
)
plt.xlabel("Age")
plt.ylabel("Annual Income")
plt.title("Distribusi Age dan Annual Income")
plt.show()
```

Bandingkan pola yang terlihat dengan visualisasi AnnualIncome dan SpendingScore.

Jawab:

1. Apakah pola kelompok terlihat berbeda?
2. Kombinasi fitur mana yang menurut Anda lebih informatif untuk segmentasi pelanggan?
3. Mengapa pemilihan fitur dapat memengaruhi hasil clustering?

Latihan 3 — Pengaruh Standardisasi

Gunakan fitur:

```
X = df[["AnnualIncome", "SpendingScore"]]
```

Buat dua model K-Means:

- Model A menggunakan data asli.
- Model B menggunakan data yang telah distandardisasi.

Model tanpa standardisasi:

```
kmeans_raw = KMeans(
    n_clusters=5,
    random_state=42,
    n_init=10
)

cluster_raw = kmeans_raw.fit_predict(X)
```

Model dengan standardisasi:

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans_scaled = KMeans(
    n_clusters=5,
    random_state=42,
    n_init=10
)

cluster_scaled = kmeans_scaled.fit_predict(X_scaled)
```

Bandingkan hasil kedua model.

Jawab:

1. Apakah label cluster yang dihasilkan sama?
2. Apakah posisi centroid berbeda?
3. Mengapa standardisasi penting pada algoritma berbasis jarak?

Latihan 4 — Menentukan Nilai K

Gunakan Elbow Method untuk mencoba nilai:

$$K = 1, 2, 3, \dots, 10$$

Gunakan kode:

```
inertia = []

for k in range(1, 11):
    model = KMeans(
        n_clusters=k,
        random_state=42,
        n_init=10
    )

    model.fit(X_scaled)
    inertia.append(model.inertia_)
```

Visualisasikan:

```
plt.plot(
    range(1, 11),
    inertia,
    marker="o"
)

plt.xlabel("Jumlah Cluster (K)")
plt.ylabel("Inertia")
plt.title("Elbow Method")
plt.xticks(range(1, 11))
plt.show()
```

Tentukan nilai K yang menurut Anda paling sesuai.

Jelaskan alasan pemilihan tersebut berdasarkan bentuk grafik.

Latihan 5 — Membandingkan Silhouette Score

Hitung Silhouette Score untuk beberapa nilai cluster:

```
for k in range(2, 8):
    model = KMeans(
        n_clusters=k,
        random_state=42,
        n_init=10
    )

    labels = model.fit_predict(X_scaled)

    score = silhouette_score(
        X_scaled,
        labels
    )

    print(
        "K =", k,
        "| Silhouette Score =", score
    )
```

Catat hasilnya dalam tabel:

K	Silhouette Score
2	...
3	...
4	...
5	...
6	...
7	...

Kemudian jawab:

1. Nilai K mana yang menghasilkan Silhouette Score tertinggi?
2. Apakah hasilnya sama dengan pilihan berdasarkan Elbow Method?
3. Jika berbeda, metode mana yang akan Anda gunakan? Jelaskan pertimbangannya.

Latihan 6 — Membandingkan Beberapa Nilai K

Bangun tiga model K-Means dengan:

- K = 3
- K = 4
- K = 5

Contoh:

```
for k in [3, 4, 5]:
    model = KMeans(
        n_clusters=k,
        random_state=42,
        n_init=10
    )

    labels = model.fit_predict(X_scaled)

    plt.figure(figsize=(8, 5))

    plt.scatter(
        X["AnnualIncome"],
        X["SpendingScore"],
        c=labels
    )

    plt.xlabel("Annual Income")
    plt.ylabel("Spending Score")
    plt.title(f"Hasil K-Means dengan K = {k}")

    plt.show()
```

Bandingkan ketiga visualisasi tersebut.

Perhatikan:

- keterpisahan antarcluster;
- ukuran cluster;
- kemungkinan tumpang tindih;
- kemudahan interpretasi hasil.

Latihan 7 — Analisis Karakteristik Cluster

Gunakan model yang menurut Anda menghasilkan clustering terbaik.

Tambahkan label cluster:

```
df["Cluster"] = labels
```

Hitung rata-rata karakteristik setiap cluster:

```
cluster_summary = df.groupby("Cluster") [
    ["Age", "AnnualIncome", "SpendingScore"]
].mean()

cluster_summary
```

Kemudian lengkapi tabel berikut berdasarkan hasil analisis Anda:

Cluster	Usia	Pendapatan	Spending Score	Karakteristik
0	...	...	...	...
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
4	...	...	...	...

Berikan nama atau deskripsi yang sesuai untuk setiap cluster berdasarkan karakteristiknya.

Latihan 8 — Menyusun Insight

Berdasarkan hasil clustering, susun minimal tiga insight yang dapat digunakan untuk mendukung pengambilan keputusan.

Gunakan struktur berikut:

Insight 1

Cluster yang diamati : ...

Karakteristik : ...

Interpretasi : ...

Rekomendasi : ...

Insight 2

Cluster yang diamati : ...

Karakteristik : ...

Interpretasi : ...

Rekomendasi : ...

Insight 3

Cluster yang diamati : ...

Karakteristik : ...

Interpretasi : ...

Rekomendasi : ...

Contoh:

Pelanggan pada cluster dengan pendapatan tinggi dan tingkat pengeluaran tinggi dapat dipertimbangkan sebagai pelanggan bernilai tinggi. Organisasi dapat mempertahankan kelompok ini melalui program loyalitas, layanan prioritas, atau penawaran eksklusif.

Mahasiswa harus memastikan bahwa rekomendasi yang dibuat benar-benar didasarkan pada karakteristik cluster yang ditemukan.

Latihan 9 — Eksperimen Mandiri

Lakukan satu eksperimen tambahan dengan salah satu pilihan berikut:

1. menggunakan tiga fitur sekaligus: Age, AnnualIncome, dan SpendingScore;
2. mencoba jumlah cluster yang berbeda;
3. membandingkan hasil clustering dengan dan tanpa standardisasi;
4. menggunakan dataset lain yang memiliki minimal dua fitur numerik;
5. membandingkan hasil visualisasi cluster menggunakan kombinasi fitur yang berbeda.

Dokumentasikan:

- 1) tujuan eksperimen;
- 2) fitur yang digunakan;
- 3) nilai K;
- 4) metode penentuan jumlah cluster;
- 5) hasil evaluasi;

- 6) visualisasi cluster;
- 7) interpretasi hasil;
- 8) kesimpulan eksperimen.

6.10 Kesimpulan Latihan

Melalui rangkaian latihan ini, mahasiswa diharapkan memahami bahwa hasil K-Means dapat berubah karena beberapa faktor, terutama:

pemilihan fitur → skala data → jumlah cluster → karakteristik distribusi data → metode evaluasi → interpretasi hasil.

Karena itu, clustering tidak sebaiknya dilakukan hanya dengan menjalankan satu perintah K-Means. Setiap keputusan dalam proses pemodelan perlu dianalisis dan dijelaskan berdasarkan karakteristik dataset serta tujuan analisis.

7. Tugas Praktikum

Tugas praktikum bertujuan mengukur kemampuan mahasiswa dalam menerapkan algoritma **K-Means Clustering** secara mandiri mulai dari penyiapan data hingga interpretasi hasil. Mahasiswa diharapkan mampu menjelaskan setiap keputusan analisis yang dilakukan, bukan hanya menghasilkan output program.

7.1 Deskripsi Tugas

Gunakan dataset pelanggan yang telah digunakan pada praktikum atau dataset lain yang memiliki minimal **dua fitur numerik** dan sesuai untuk analisis clustering.

Lakukan proses clustering menggunakan algoritma K-Means dengan alur berikut:



Gambar 8. Alur Proses Clustering K-Means

Setiap tahapan harus disertai kode program, output, dan penjelasan singkat mengenai hasil yang diperoleh.

7.2 Ketentuan Tugas

Mahasiswa diminta melakukan kegiatan berikut:

1. Menjelaskan dataset yang digunakan, termasuk sumber, jumlah data, dan atribut.
2. Melakukan pemeriksaan struktur dan kualitas data.
3. Menentukan minimal dua fitur yang digunakan untuk clustering.
4. Menjelaskan alasan pemilihan fitur.
5. Melakukan visualisasi awal terhadap distribusi data.
6. Melakukan standardisasi data apabila diperlukan.
7. Menggunakan **Elbow Method** untuk membantu menentukan jumlah cluster.
8. Menghitung **Silhouette Score** untuk minimal tiga kandidat nilai K.
9. Menentukan jumlah cluster yang digunakan dan memberikan alasan pemilihannya.
10. Membangun model K-Means menggunakan scikit-learn.
11. Menampilkan hasil clustering dalam bentuk visualisasi.
12. Menghitung dan menganalisis centroid setiap cluster.
13. Mengidentifikasi karakteristik masing-masing cluster.
14. Memberikan nama atau deskripsi yang sesuai untuk setiap cluster.
15. Menyusun minimal tiga insight berdasarkan hasil clustering.
16. Memberikan rekomendasi yang relevan berdasarkan pola yang ditemukan.

7.3 Eksperimen Perbandingan

Selain model utama, mahasiswa diminta melakukan **satu eksperimen perbandingan**.

Eksperimen dapat berupa salah satu pilihan berikut:

- 1) membandingkan hasil dengan dan tanpa standardisasi;
- 2) membandingkan dua kombinasi fitur;
- 3) membandingkan beberapa nilai K;
- 4) menggunakan dua pendekatan penentuan jumlah cluster;
- 5) menambahkan satu fitur numerik lain;
- 6) menggunakan dataset lain yang memiliki karakteristik berbeda.

Hasil eksperimen harus disajikan dalam bentuk tabel perbandingan.

Contoh:

Eksperimen	Fitur	K	Standardisasi	Silhouette Score	Catatan
Model A	AnnualIncome, SpendingScore	4	Ya	...	...
Model B	AnnualIncome, SpendingScore	5	Ya	...	...
Model C	Age, AnnualIncome	5	Ya	...	...

Berdasarkan tabel tersebut, mahasiswa menentukan model yang dianggap paling sesuai dan memberikan argumentasi berdasarkan hasil eksperimen.

7.4 Analisis Karakteristik Cluster

Buat tabel yang menjelaskan karakteristik cluster yang diperoleh.

Contoh format:

Cluster	Jumlah Anggota	Pendapatan	Spending Score	Karakteristik
0	...	...	...	...
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
4	...	...	...	...

Interpretasi harus didasarkan pada nilai statistik hasil clustering, bukan semata-mata berdasarkan nomor cluster.

Mahasiswa dapat menggunakan kode berikut:

```
cluster_summary = df.groupby("Cluster").agg({
    "AnnualIncome": ["mean", "min", "max"],
    "SpendingScore": ["mean", "min", "max"]
})

cluster_summary
```

Jumlah anggota setiap cluster dapat diperiksa menggunakan:

```
df["Cluster"].value_counts().sort_index()
```

7.5 Penyusunan Insight dan Rekomendasi

Setelah karakteristik cluster diketahui, mahasiswa diminta menyusun minimal **tiga insight**.

Gunakan format :

Cluster : ...

Karakteristik utama : ...

Insight : ...

Rekomendasi : ...

Contoh:

Cluster: pelanggan dengan pendapatan tinggi dan spending score tinggi.

Insight: kelompok ini memiliki kemampuan ekonomi dan aktivitas pengeluaran yang tinggi sehingga dapat dianggap sebagai pelanggan bernilai tinggi.

Rekomendasi: organisasi dapat mempertahankan kelompok tersebut melalui program loyalitas, layanan prioritas, atau penawaran khusus.

Rekomendasi harus relevan dengan karakteristik data dan tidak boleh dibuat tanpa didukung hasil analisis.

7.6 Pertanyaan Analisis

Jawab pertanyaan berikut berdasarkan eksperimen yang telah dilakukan:

1. Mengapa fitur tertentu dipilih untuk clustering?
2. Apakah standardisasi memberikan perubahan terhadap hasil clustering? Jelaskan.
3. Berapa jumlah cluster yang paling sesuai berdasarkan Elbow Method?
4. Berapa jumlah cluster yang menghasilkan Silhouette Score terbaik?
5. Apakah hasil Elbow Method dan Silhouette Score menghasilkan keputusan yang sama?
6. Bagaimana karakteristik masing-masing cluster?
7. Cluster mana yang paling menarik untuk dianalisis? Mengapa?
8. Apa insight utama yang diperoleh dari hasil clustering?
9. Bagaimana hasil clustering dapat digunakan dalam pengambilan keputusan?
10. Apa keterbatasan analisis clustering yang telah dilakukan?

7.7 Format Laporan

Laporan praktikum disusun dengan struktur berikut:

1. **Judul Praktikum**
2. **Tujuan**
3. **Deskripsi Dataset**
4. **Pemeriksaan Data**
5. **Pemilihan Fitur**
6. **Standardisasi Data**
7. **Penentuan Jumlah Cluster**
8. **Implementasi K-Means**
9. **Evaluasi Cluster**
10. **Visualisasi Hasil**
11. **Analisis Karakteristik Cluster**
12. **Eksperimen Perbandingan**
13. **Insight dan Rekomendasi**
14. **Kesimpulan**

Notebook harus berisi kombinasi antara **kode, output, visualisasi, dan penjelasan** sehingga proses analisis dapat ditelusuri dengan jelas.

7.8 Berkas yang Dikumpulkan

Mahasiswa mengumpulkan:

- a. file notebook .ipynb;
- b. dataset yang digunakan;
- c. file hasil clustering .csv;
- d. laporan praktikum dalam format PDF apabila diminta; dan
- e. visualisasi utama hasil clustering.

Penamaan file:

NIM_Nama_Modul8_KMeans.ipynb

Contoh:

23101001_Andi_Modul8_KMeans.ipynb

Pengumpulan dilakukan melalui **LMS Edlink** sesuai batas waktu yang ditentukan.

7.9 Kriteria Keberhasilan

Tugas dinyatakan berhasil apabila mahasiswa mampu menunjukkan bahwa:

- 1) dataset telah diperiksa dan disiapkan dengan benar;
- 2) fitur dipilih dengan alasan yang jelas;
- 3) jumlah cluster ditentukan berdasarkan hasil analisis;
- 4) K-Means berhasil diterapkan;
- 5) hasil clustering dievaluasi dan divisualisasikan;
- 6) karakteristik masing-masing cluster dapat dijelaskan;
- 7) insight disusun berdasarkan bukti dari data; dan
- 8) seluruh proses terdokumentasi secara sistematis dan dapat direproduksi.

8. Pertanyaan Evaluasi dan Refleksi Praktikum

Bagian ini digunakan untuk mengevaluasi pemahaman mahasiswa terhadap konsep dan penerapan clustering menggunakan algoritma K-Means. Pertanyaan tidak hanya menilai kemampuan menjalankan kode, tetapi juga pemahaman terhadap alasan pemilihan metode, interpretasi hasil, dan keterbatasan analisis.

8.1 Pertanyaan Konseptual

Jawab pertanyaan berikut secara singkat dan jelas.

- 1) Apa yang dimaksud dengan unsupervised learning?
- 2) Apa perbedaan utama antara clustering dan klasifikasi?
- 3) Apa tujuan utama dari proses clustering?
- 4) Apa yang dimaksud dengan centroid pada algoritma K-Means?
- 5) Mengapa K-Means termasuk algoritma berbasis jarak?
- 6) Apa fungsi nilai K pada K-Means?
- 7) Mengapa proses standardisasi sering dilakukan sebelum clustering?
- 8) Apa fungsi Elbow Method dalam K-Means?
- 9) Apa yang dimaksud dengan inertia?
- 10) Apa yang dimaksud dengan Silhouette Score?
- 11) Mengapa hasil clustering tetap perlu diinterpretasikan setelah model selesai dijalankan?
- 12) Apakah nomor cluster memiliki arti peringkat? Jelaskan.

8.2 Pertanyaan Analisis Hasil

Gunakan hasil praktikum yang telah diperoleh untuk menjawab pertanyaan berikut.

- 1) Berapa jumlah cluster yang dipilih pada model akhir?
- 2) Apa alasan pemilihan jumlah cluster tersebut?
- 3) Apakah hasil Elbow Method dan Silhouette Score memberikan rekomendasi yang sama?
- 4) Cluster mana yang memiliki jumlah anggota paling banyak?
- 5) Cluster mana yang memiliki jumlah anggota paling sedikit?
- 6) Bagaimana karakteristik centroid dari setiap cluster?
- 7) Cluster mana yang memiliki rata-rata AnnualIncome tertinggi?
- 8) Cluster mana yang memiliki rata-rata SpendingScore tertinggi?
- 9) Apakah terdapat cluster yang secara visual saling berdekatan?
- 10) Apakah ada cluster yang memiliki karakteristik sangat berbeda dibandingkan cluster lainnya?

8.3 Pertanyaan Interpretasi

Jawab pertanyaan berikut berdasarkan karakteristik cluster yang terbentuk.

- 1) Bagaimana Anda memberikan nama atau label deskriptif pada setiap cluster?
- 2) Cluster mana yang dapat dikategorikan sebagai pelanggan bernilai tinggi?
- 3) Cluster mana yang berpotensi menjadi target promosi?
- 4) Cluster mana yang mungkin membutuhkan strategi layanan yang berbeda?
- 5) Insight apa yang paling penting dari hasil clustering?
- 6) Bagaimana hasil clustering dapat mendukung proses pengambilan keputusan?
- 7) Apakah seluruh cluster memiliki makna yang relevan dalam konteks bisnis? Jelaskan.

8.4 Pertanyaan Refleksi Praktikum

Tuliskan refleksi singkat berdasarkan proses praktikum yang telah dilakukan.

- 1) Tahapan mana yang paling mudah dipahami?
- 2) Tahapan mana yang paling sulit dilakukan?
- 3) Apakah penentuan nilai K mudah dilakukan? Jelaskan.
- 4) Apa pengaruh pemilihan fitur terhadap hasil clustering?
- 5) Apa pengaruh standardisasi terhadap hasil model?
- 6) Apakah visualisasi membantu memahami cluster yang terbentuk?

- 7) Apa keterbatasan penggunaan K-Means yang Anda temukan selama praktikum?
- 8) Jika praktikum diulang menggunakan dataset berbeda, bagian apa yang perlu disesuaikan?
- 9) Apa pengetahuan baru yang diperoleh dari praktikum ini?
- 10) Bagaimana clustering dapat diterapkan pada permasalahan lain di bidang Sistem Informasi?

8.5 Kesimpulan Refleksi

Berdasarkan seluruh proses praktikum, mahasiswa diharapkan dapat menyimpulkan bahwa clustering bukan hanya proses membagi data menjadi beberapa kelompok, tetapi merupakan proses analitik yang memerlukan pemilihan fitur, penyiapan data, penentuan jumlah cluster, evaluasi model, dan interpretasi hasil secara hati-hati.

Mahasiswa juga diharapkan mampu memahami bahwa hasil K-Means harus dianalisis dalam konteks permasalahan. Cluster yang terbentuk baru memiliki nilai ketika karakteristiknya dapat dijelaskan dan digunakan untuk menghasilkan insight yang relevan bagi pengambilan keputusan.

9. Kesimpulan Praktikum

Pada praktikum ini, mahasiswa telah mempelajari proses clustering menggunakan algoritma K-Means sebagai salah satu metode unsupervised learning. Berbeda dengan klasifikasi, clustering tidak menggunakan label target, tetapi berusaha menemukan pola pengelompokan berdasarkan kemiripan karakteristik data.

Proses clustering dilakukan secara bertahap mulai dari memuat dataset, memeriksa kualitas data, memilih fitur, melakukan visualisasi awal, melakukan standardisasi, menentukan jumlah cluster, membangun model K-Means, mengevaluasi cluster, memvisualisasikan hasil, menganalisis karakteristik cluster, hingga menyusun insight.

Melalui praktikum ini dapat dipahami bahwa keberhasilan clustering tidak hanya ditentukan oleh algoritma yang digunakan. Pemilihan fitur, skala data, jumlah cluster, dan metode evaluasi juga berpengaruh terhadap kualitas hasil pengelompokan.

Elbow Method digunakan sebagai salah satu pendekatan untuk membantu menentukan jumlah cluster melalui perubahan nilai inertia, sedangkan Silhouette Score digunakan

untuk membantu mengevaluasi seberapa baik pemisahan antarcluster. Kedua pendekatan tersebut dapat digunakan secara bersama-sama sebagai dasar dalam menentukan jumlah cluster yang sesuai.

Hasil K-Means berupa label cluster dan centroid belum secara langsung memberikan makna. Oleh karena itu, setiap cluster perlu dianalisis berdasarkan karakteristik fitur yang digunakan. Proses interpretasi inilah yang memungkinkan hasil clustering dikembangkan menjadi insight yang relevan.

Dalam konteks Sistem Informasi, clustering dapat dimanfaatkan untuk berbagai kebutuhan seperti segmentasi pelanggan, pengelompokan produk, analisis perilaku pengguna, pengelompokan transaksi, dan identifikasi pola pada data organisasi. Hasil pengelompokan tersebut selanjutnya dapat digunakan sebagai salah satu bahan pendukung dalam proses pengambilan keputusan berbasis data.

Secara ringkas, kompetensi yang diperoleh melalui praktikum ini mencakup:

- a. memahami konsep unsupervised learning dan clustering;
- b. menyiapkan data untuk proses clustering;
- c. memilih fitur yang relevan;
- d. melakukan standardisasi data;
- e. menentukan jumlah cluster;
- f. menerapkan algoritma K-Means menggunakan scikit-learn;
- g. mengevaluasi hasil clustering;
- h. memvisualisasikan cluster dan centroid;
- i. menganalisis karakteristik setiap cluster; dan
- j. mengubah hasil clustering menjadi insight serta rekomendasi yang dapat dipertanggungjawabkan.

Dengan demikian, mahasiswa diharapkan tidak hanya mampu menjalankan algoritma K-Means secara teknis, tetapi juga mampu memahami proses analisis secara menyeluruh dan menginterpretasikan hasil clustering sesuai dengan konteks permasalahan yang dianalisis.

Referensi

1. Géron, A. (2023). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.
2. Han, J., Pei, J., & Tong, H. (2022). Data Mining: Concepts and Techniques (4th ed.). Morgan Kaufmann–Elsevier.
3. Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.
4. McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.
5. Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly Media.
6. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.

DAFTAR PUSTAKA

- Fawcett, F.P., Tom, 2013. *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. [online] O'Reilly Media.
- Géron, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Seco edn. O'Reilly Media.
- Google, 2026. *Google Colaboratory: Frequently Asked Questions*. [online]
- Han, J., Pei, J. and Tong, H., 2022. *Data Mining: Concepts and Techniques*. 4th ed edn. [online] Morgan Kaufmann Publishers..
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
- Matplotlib Development Team. *Matplotlib Documentation*. Dokumentasi resmi untuk visualisasi data dan hasil pemodelan.
- McKinney, W., 2017. *Python for Data Analysis*. 2nd edn. O'Reilly Media.
- NumPy Developers. *NumPy Documentation*. Dokumentasi resmi untuk operasi numerik dan array multidimensi.
- pandas Development Team. *pandas Documentation*. Dokumentasi resmi untuk manipulasi, pembersihan, transformasi, dan analisis data.
- Project Jupyter, 2026. *Project Jupyter Documentation*.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O'Reilly Media.
- Python Software Foundation. *Python Documentation*. Dokumentasi resmi bahasa pemrograman Python.
- scikit-learn Developers, 2026. *Getting Started with scikit-learn*. [online] scikit-learn.
- UNESCO, 2022. *Recommendation on the Ethics of Artificial Intelligence*.
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.