MODUL PEMBELAJARAN PYTHON

MODUL 10
DATA CLEANING & EXPLORATORY DATA ANALYSIS

Teknik Pembersihan Data, Analisis Eksploratif, dan Interpretasi Temuan

Informasi Modul	Keterangan
Mata pembelajaran	Pemrograman Python / Data Analysis
Level	Pemula – Menengah
Prasyarat	Python dasar, Pandas, DataFrame, dan operasi dasar NumPy
Perangkat	Python 3, VS Code/Jupyter Notebook, Pandas, NumPy, Matplotlib, Seaborn
Estimasi waktu	3–4 pertemuan (dapat disesuaikan)
Produk akhir	Dataset bersih, notebook EDA, visualisasi, dan laporan temuan

Bahan ajar praktik – disusun untuk kegiatan pembelajaran

Petunjuk Penggunaan Modul

Modul ini menggunakan pendekatan praktik berbasis data. Peserta didik dianjurkan menjalankan kode secara bertahap, mencatat keputusan pembersihan data, dan membedakan fakta yang terlihat pada data dari interpretasi atau dugaan.

- Gunakan Python 3 dan VS Code atau Jupyter Notebook.
- Buat folder proyek, misalnya data_cleaning_eda.
- Siapkan salinan dataset mentah; jangan menimpa file sumber.
- Jalankan setiap contoh kode dan periksa outputnya.
- Catat jumlah baris sebelum dan sesudah pembersihan serta alasan setiap keputusan.
- Gunakan data simulasi atau data yang telah dianonimkan untuk latihan.

1. Capaian dan Tujuan Pembelajaran

Capaian pembelajaran: peserta didik mampu menilai kualitas dataset, menerapkan teknik pembersihan yang sesuai konteks, melakukan analisis eksploratif, membuat visualisasi, dan menyampaikan temuan berdasarkan bukti.

- Menjelaskan tujuan, manfaat, dan alur kerja Data Cleaning serta EDA.
- Mengidentifikasi masalah kualitas data: missing values, duplikasi, inkonsistensi, tipe data, dan nilai tidak valid.
- Melakukan pembersihan data dengan Pandas dan NumPy.
- Memahami statistik deskriptif, distribusi, frekuensi, dan hubungan antarvariabel.
- Mengidentifikasi indikasi outlier dan memahami keterbatasan metode deteksinya.
- Membuat visualisasi univariat dan bivariat menggunakan Matplotlib dan Seaborn.
- Merumuskan pertanyaan analisis, menyusun temuan, dan membuat rekomendasi yang tidak melampaui bukti.

2. Materi Teori

2.1 Pengertian Data Cleaning

Data Cleaning adalah proses mendeteksi, mengevaluasi, dan menangani masalah pada data agar data lebih akurat, konsisten, lengkap, dan layak digunakan. Data yang belum dibersihkan dapat menghasilkan ringkasan yang menyesatkan, visualisasi yang keliru, atau model prediksi yang buruk.

Pembersihan bukan sekadar menghapus baris yang kosong. Setiap keputusan perlu mempertimbangkan makna variabel, proses pengumpulan data, tujuan analisis, dan risiko kehilangan informasi.

Dimensi kualitas	Pertanyaan pemeriksaan	Contoh masalah
Completeness	Apakah nilai penting tersedia?	Kolom usia kosong.
Validity	Apakah nilai sesuai aturan/rentang?	Nilai ujian 145 pada skala 0–100.
Consistency	Apakah format dan kategori seragam?	'Laki-laki', 'L', dan 'Pria'.
Uniqueness	Apakah catatan unik sesuai aturan?	ID transaksi tercatat dua kali.
Accuracy	Apakah nilai mewakili kondisi sebenarnya?	Tanggal lahir salah input.
Timeliness	Apakah data cukup mutakhir?	Data stok lama dipakai untuk keputusan hari ini.

2.2 Alur Kerja Data Cleaning dan EDA

Tahap	Aktivitas	Hasil yang diharapkan

1. Memahami tujuan	Menentukan pertanyaan analisis dan unit observasi.	Ruang lingkup analisis jelas.
2. Memuat data	Membaca CSV/Excel dan memeriksa sumber.	Dataset awal tersedia.
3. Profiling	Memeriksa shape, tipe, distribusi, missing, duplikasi.	Daftar masalah data.
4. Cleaning	Memperbaiki format, nilai kosong, duplikasi, dan nilai tidak valid.	Dataset bersih dan terdokumentasi.
5. Validasi	Menguji aturan kualitas setelah cleaning.	Dataset siap dianalisis.
6. EDA	Menganalisis distribusi, kelompok, hubungan, dan anomali.	Temuan awal dan visualisasi.
7. Komunikasi	Menulis interpretasi, batasan, dan rekomendasi.	Laporan analisis.

2.3 Menyiapkan Library

```
python -m pip install pandas numpy matplotlib seaborn openpyxl

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

print("Pandas:", pd.__version__)
print("NumPy:", np.__version__)
```

Pandas digunakan untuk manipulasi tabel, NumPy untuk operasi numerik, Matplotlib sebagai dasar visualisasi, dan Seaborn untuk grafik statistik yang lebih ringkas.

2.4 Data Profiling: Memahami Kondisi Dataset

Profiling adalah pemeriksaan awal untuk mengenali struktur dan kualitas data sebelum melakukan perubahan.

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

print("Ukuran:", df.shape)
print(df.head())
print(df.sample(min(5, len(df)), random_state=42))
df.info()
print(df.describe(include="all").T)
print("Missing value:\n", df.isna().sum())
print("Duplikasi:", df.duplicated().sum())
print("Nama kolom:", df.columns.tolist())
```

Periksa pula nilai unik pada kolom kategorikal, rentang nilai numerik, format tanggal, dan apakah satu baris benar-benar mewakili satu entitas atau kejadian.

2.5 Menangani Missing Values

Missing value adalah nilai yang tidak tersedia. Penyebabnya dapat berupa pertanyaan yang tidak dijawab, kesalahan input, data tidak tercatat, atau variabel yang memang tidak berlaku untuk suatu observasi.

Strategi	Kapan dipertimbangkan	Risiko/catatan
Menghapus baris	Jumlah kecil dan kehilangan baris tidak menimbulkan bias berarti.	Sampel berkurang; pola missing mungkin tidak acak.
Menghapus kolom	Kolom tidak relevan atau terlalu banyak kosong dan tidak dibutuhkan.	Informasi penting bisa hilang.
Imputasi mean	Data numerik relatif simetris dan konteks mendukung.	Sensitif terhadap outlier; menurunkan variasi.
Imputasi median	Data numerik miring atau memiliki outlier.	Dapat menyembunyikan variasi kelompok.

Imputasi modus/kategori khusus	Data kategorikal, dengan alasan yang sesuai.	Kategori paling sering belum tentu tepat.
Forward/backward fill	Data berurutan waktu dan nilai sebelumnya/berikutnya relevan.	Tidak cocok untuk semua jenis data.

```
# Jumlah dan persentase missing per kolom
missing = pd.DataFrame({
    "jumlah": df.isna().sum(),
    "persen": (df.isna().mean() * 100).round(2)
})
print(missing)

# Mengisi numerik dengan median
df["usia"] = df["usia"].fillna(df["usia"].median())

# Mengisi kategori dengan label eksplisit
df["kota"] = df["kota"].fillna("Tidak diketahui")

# Menghapus baris jika kolom identitas wajib kosong
df = df.dropna(subset=["id_responden"])
```

Penting: bedakan nilai kosong dari nol. Nilai 0 dapat merupakan data sah, misalnya jumlah transaksi nol. Jangan mengubah nol menjadi missing tanpa dasar.

2.6 Menangani Duplikasi

Duplikasi dapat terjadi karena data diimpor berulang, pengiriman formulir ganda, atau kesalahan integrasi. Tentukan dahulu definisi duplikat: apakah seluruh kolom harus sama atau cukup berdasarkan ID tertentu.

```
# Memeriksa duplikasi seluruh kolom
print(df.duplicated().sum())

# Memeriksa duplikasi berdasarkan ID
print(df[df.duplicated(subset=["id_responden"], keep=False)])

# Menghapus duplikasi berdasarkan ID
df = df.drop_duplicates(subset=["id_responden"], keep="first")
```

keep='first' mempertahankan catatan pertama. Dalam data transaksi, ID pelanggan dapat muncul berulang secara sah; karena itu jangan menghapus duplikasi hanya berdasarkan ID pelanggan.

2.7 Memperbaiki Tipe Data dan Format

```
# Konversi numerik yang gagal akan menjadi NaN
df["nilai"] = pd.to_numeric(df["nilai"], errors="coerce")

# Konversi tanggal
df["tanggal"] = pd.to_datetime(df["tanggal"], errors="coerce", dayfirst=True)

# Membersihkan spasi dan konsistensi teks
df["nama"] = df["nama"].astype("string").str.strip()
df["kategori"] = df["kategori"].astype("string").str.strip().str.title()

# Mengganti label kategori yang tidak seragam
df["jenis_kelamin"] = df["jenis_kelamin"].replace({
    "L": "Laki-laki",
    "Pria": "Laki-laki",
    "P": "Perempuan",
    "Wanita": "Perempuan"
})
```

Periksa hasil konversi setelahnya. errors='coerce' mengubah nilai yang tidak dapat dikonversi menjadi missing value; nilai tersebut perlu ditinjau, bukan dibiarkan tanpa pemeriksaan.

2.8 Validasi Rentang dan Aturan Bisnis

Validasi memeriksa apakah nilai memenuhi aturan logis atau aturan bisnis. Contohnya nilai ujian 0–100, jumlah barang tidak negatif, atau tanggal transaksi tidak berada di luar periode yang masuk akal.

```
# Menandai nilai ujian di luar rentang
mask_invalid = ~df["nilai"].between(0, 100)
data_invalid = df.loc[mask_invalid]
print(data_invalid)

# Memeriksa tanggal yang gagal dibaca
tanggal_invalid = df[df["tanggal"].isna()]

# Aturan bisnis: kuantitas dan harga tidak negatif
invalid_transaksi = df[
    (df["kuantitas"] < 0) | (df["harga"] < 0)
]
```

Jangan otomatis menghapus nilai yang tidak biasa. Tandai, telusuri sumbernya, lalu putuskan apakah nilai harus dikoreksi, dikeluarkan, atau dipertahankan dengan catatan.

2.9 Outlier: Konsep dan Penanganan

Outlier adalah observasi yang jauh berbeda dari pola sebagian besar data. Outlier dapat berasal dari kesalahan input, kejadian langka yang sah, atau variasi alami. Outlier tidak selalu salah.

Metode IQR menggunakan rentang antarkuartil. Batas bawah = Q1 − 1,5 × IQR dan batas atas = Q3 + 1,5 × IQR. Nilai di luar batas ditandai sebagai kandidat outlier, bukan otomatis dihapus.

```
Q1 = df["nilai"].quantile(0.25)
Q3 = df["nilai"].quantile(0.75)
IQR = Q3 - Q1

batas_bawah = Q1 - 1.5 * IQR
batas_atas = Q3 + 1.5 * IQR

kandidat_outlier = df[
    (df["nilai"] < batas_bawah) |
    (df["nilai"] > batas_atas)
]
print(kandidat_outlier)
```

Alternatif lain adalah z-score, yang mengukur jarak nilai dari rata-rata dalam satuan simpangan baku. Z-score lebih sesuai pada distribusi yang mendekati normal dan dapat sensitif terhadap distribusi miring.

```
mean = df["nilai"].mean()
std = df["nilai"].std()

if std > 0:
    df["z_score"] = (df["nilai"] - mean) / std
    kandidat = df[df["z_score"].abs() > 3]
```

Pilihan penanganan: mempertahankan dengan penjelasan, memperbaiki kesalahan sumber, melakukan transformasi, winsorizing berdasarkan aturan yang disepakati, atau mengecualikan observasi dengan alasan kuat. Dokumentasikan dampaknya.

2.10 Pengantar Exploratory Data Analysis (EDA)

EDA adalah proses eksplorasi untuk memahami karakteristik data, menemukan pola, memeriksa asumsi, mengidentifikasi anomali, dan menghasilkan pertanyaan lanjutan. EDA bersifat eksploratif; temuan hubungan tidak otomatis membuktikan sebab-akibat.

Jenis analisis	Fokus	Contoh pertanyaan
Univariat	Satu variabel.	Bagaimana distribusi nilai siswa?

Bivariat	Dua variabel.	Apakah waktu belajar berkaitan dengan nilai?
Multivariat	Tiga variabel atau lebih.	Apakah pola hubungan berbeda antar kelas?
Temporal	Perubahan menurut waktu.	Bagaimana tren penjualan per bulan?
Segmentasi	Perbandingan kelompok.	Kelompok mana memiliki rata-rata tertinggi?

2.11 Statistik Deskriptif

Statistik deskriptif merangkum data melalui ukuran pemusatan, penyebaran, dan posisi.

Ukuran	Makna	Catatan interpretasi
Mean	Rata-rata aritmetika.	Dapat dipengaruhi nilai ekstrem.
Median	Nilai tengah setelah data diurutkan.	Lebih tahan terhadap outlier.
Mode	Nilai yang paling sering muncul.	Bisa lebih dari satu modus.
Minimum/Maximum	Nilai terkecil/terbesar.	Periksa apakah nilainya masuk akal.
Range	Maximum − Minimum.	Menggambarkan rentang total.
Variance/Std	Ukuran penyebaran.	Std menggunakan satuan yang sama dengan data.
Quartile	Pembagian data menjadi empat bagian.	Membantu memahami sebaran dan IQR.

```
kolom_numerik = df.select_dtypes(include="number")
print(kolom_numerik.describe().T)

print("Mean:", df["nilai"].mean())
print("Median:", df["nilai"].median())
print("Modus:", df["nilai"].mode().tolist())
print("Standar deviasi:", df["nilai"].std())
```

2.12 Analisis Distribusi dan Frekuensi

```
# Frekuensi kategori
print(df["kelas"].value_counts())
print(df["kelas"].value_counts(normalize=True).mul(100).round(1))

# Histogram
sns.histplot(data=df, x="nilai", bins=10, kde=True)
plt.title("Distribusi Nilai")
plt.xlabel("Nilai")
plt.ylabel("Frekuensi")
plt.tight_layout()
plt.show()

# Boxplot untuk melihat sebaran dan kandidat outlier
sns.boxplot(data=df, x="kelas", y="nilai")
plt.title("Sebaran Nilai per Kelas")
plt.tight_layout()
plt.show()
```

Histogram memperlihatkan bentuk distribusi; boxplot merangkum median, kuartil, dan kandidat nilai ekstrem. Bentuk grafik perlu dibaca bersama ukuran sampel dan konteks pengumpulan data.

2.13 Analisis Hubungan Antarvariabel

```
# Scatter plot untuk dua variabel numerik
sns.scatterplot(data=df, x="jam_belajar", y="nilai", hue="kelas")
plt.title("Jam Belajar dan Nilai")
plt.tight_layout()
plt.show()
```

```
# Korelasi numerik
corr = df.select_dtypes(include="number").corr(method="pearson")
print(corr)

sns.heatmap(corr, annot=True, cmap="coolwarm", vmin=-1, vmax=1)
plt.title("Matriks Korelasi")
plt.tight_layout()
plt.show()
```

Korelasi Pearson mengukur hubungan linear antarvariabel numerik dengan rentang −1 sampai +1. Korelasi positif menunjukkan kecenderungan bergerak searah; korelasi negatif menunjukkan kecenderungan berlawanan arah. Korelasi tidak membuktikan kausalitas dan dapat dipengaruhi outlier atau variabel perancu.

Untuk hubungan monoton yang tidak harus linear atau data ordinal, pertimbangkan korelasi Spearman. Periksa diagram pencar sebelum menafsirkan satu angka korelasi.

2.14 Analisis Kelompok dan Pivot Table

```
# Ringkasan nilai per kelas
ringkasan = df.groupby("kelas")["nilai"].agg(
    jumlah="count",
    rata_rata="mean",
    median="median",
    minimum="min",
    maksimum="max"
).round(2)
print(ringkasan)

# Pivot table
pivot = pd.pivot_table(
    df,
    index="kelas",
    columns="jenis_kelamin",
    values="nilai",
    aggfunc="mean"
)
print(pivot)
```

Perbandingan kelompok harus memperhatikan jumlah observasi, variasi dalam kelompok, dan kemungkinan perbedaan komposisi. Rata-rata yang berbeda belum tentu menunjukkan perbedaan yang signifikan atau hubungan sebab-akibat.

3. Contoh Kode Terpadu: Cleaning dan EDA

Contoh berikut membuat dataset simulasi agar kode dapat dijalankan tanpa mengunduh file eksternal. Dataset sengaja memuat nilai kosong, kategori tidak konsisten, dan satu baris duplikat.

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

data = {
    "id": [1, 2, 3, 4, 5, 6, 6, 7, 8, 9],
    "kelas": ["XI RPL", "XI RPL", "XI DKV", "XI DKV", "XI RPL",
              "XI DKV", "XI DKV", "XI RPL", "XI DKV", "XI RPL"],
    "jam_belajar": [2, 3, 1, 4, np.nan, 2, 2, 5, 1, 3],
    "nilai": [78, 85, 70, 92, 88, 75, 75, 95, np.nan, 82],
    "kehadiran": [90, 95, 80, 98, 92, 85, 85, 100, 75, 90]
}
```

```
df = pd.DataFrame(data)

# A. Profiling
print("Shape awal:", df.shape)
print(df.isna().sum())
print("Duplikasi:", df.duplicated().sum())

# B. Cleaning: hapus duplikasi seluruh baris
df = df.drop_duplicates()

# C. Isi missing value numerik menggunakan median
for col in ["jam_belajar", "nilai"]:
    df[col] = df[col].fillna(df[col].median())

# D. Validasi rentang nilai
df = df[df["nilai"].between(0, 100)]
df = df[df["kehadiran"].between(0, 100)]

# E. EDA ringkas
print(df.describe().round(2))
print(df.groupby("kelas")["nilai"].agg(["count", "mean", "median"]))

# F. Visualisasi
fig, axes = plt.subplots(1, 2, figsize=(11, 4))
sns.histplot(data=df, x="nilai", bins=8, kde=True, ax=axes[0])
axes[0].set_title("Distribusi Nilai")
sns.boxplot(data=df, x="kelas", y="nilai", ax=axes[1])
axes[1].set_title("Nilai per Kelas")
plt.tight_layout()
plt.show()

# G. Simpan hasil
df.to_csv("dataset_bersih.csv", index=False)
```

Perhatikan bahwa contoh ini menghapus duplikasi seluruh baris. Jika aturan duplikasi proyek menggunakan kolom ID, ganti dengan drop_duplicates(subset=['id']). Keputusan tersebut harus mengikuti definisi satu observasi dalam dataset.

4. Latihan Praktik

Latihan 1 – Data Profiling

Gunakan dataset CSV yang berisi data siswa, transaksi, atau data publik sederhana.

- Tampilkan jumlah baris dan kolom, nama kolom, serta tipe data.
- Hitung missing value dan persentasenya untuk setiap kolom.
- Identifikasi kolom numerik dan kategorikal.
- Temukan kolom yang memiliki nilai unik tidak wajar atau format tidak konsisten.
- Buat daftar minimal lima masalah kualitas data yang ditemukan.

Latihan 2 – Missing Values dan Duplikasi

- Buat dataset kecil yang memuat nilai kosong dan baris duplikat.
- Bandingkan hasil dropna() dengan imputasi median untuk kolom numerik.
- Hitung jumlah baris sebelum dan sesudah pembersihan.
- Jelaskan kapan menghapus baris lebih tepat daripada mengisi nilai kosong.
- Uji duplikasi berdasarkan seluruh baris dan berdasarkan kolom ID.

Latihan 3 – Standardisasi dan Validasi

- Bersihkan spasi pada kolom nama dan kategori.
- Samakan label kategori yang memiliki variasi penulisan.
- Ubah kolom tanggal menjadi datetime dan identifikasi tanggal gagal konversi.
- Tandai nilai numerik di luar rentang yang ditentukan.
- Buat tabel ringkasan jumlah data valid dan tidak valid.

Latihan 4 – Outlier dan Statistik Deskriptif

- Hitung mean, median, modus, standar deviasi, kuartil, dan IQR.
- Identifikasi kandidat outlier dengan metode IQR.
- Bandingkan mean dan median sebelum dan sesudah kandidat outlier ditinjau.
- Jelaskan mengapa kandidat outlier tidak boleh otomatis dihapus.

Latihan 5 – EDA dan Visualisasi

- Buat histogram dan boxplot untuk variabel numerik.
- Buat bar chart untuk frekuensi kategori.
- Buat scatter plot untuk dua variabel numerik.
- Buat heatmap korelasi dan interpretasikan dua hubungan yang terlihat.
- Tulis tiga temuan berbasis data dan satu keterbatasan analisis.

5. Mini Project – Student Performance Data Exploration

5.1 Deskripsi Proyek

Peserta didik bertindak sebagai analis data akademik yang menerima dataset performa siswa. Dataset mungkin memiliki data kosong, duplikasi, penulisan kategori tidak seragam, serta nilai yang perlu divalidasi. Tugasnya adalah membersihkan dataset dan melakukan EDA untuk memahami pola performa siswa.

5.2 Tujuan Proyek

- Membuat profil kualitas data sebelum pembersihan.
- Menerapkan aturan cleaning yang transparan dan dapat diulang.
- Mengeksplorasi distribusi nilai, kehadiran, dan waktu belajar.
- Membandingkan ringkasan performa antar kelas.
- Menyajikan temuan, keterbatasan, dan rekomendasi berbasis data.

5.3 Spesifikasi Dataset

Kolom	Tipe	Deskripsi / aturan validasi
id_siswa	String	Kode unik untuk identifikasi catatan.
nama	String	Nama samaran atau data anonim.
kelas	Kategori	Contoh XI RPL, XI DKV, XI TKJ.
jenis_kelamin	Kategori	Label kategori konsisten sesuai dataset.
jam_belajar	Numerik	Estimasi jam belajar per hari/minggu; definisi harus ditentukan.
kehadiran	Numerik	Persentase kehadiran 0–100.
nilai_tugas	Numerik	Nilai tugas 0–100.
nilai_uts	Numerik	Nilai UTS 0–100.
nilai_uas	Numerik	Nilai UAS 0–100.

Gunakan sekurang-kurangnya 50 baris data simulasi atau data anonim. Definisikan satu baris sebagai satu siswa pada satu periode penilaian. Jika siswa dapat muncul di beberapa periode, gunakan kunci gabungan seperti id_siswa dan periode.

5.4 Pertanyaan Analisis

- Bagaimana kualitas dataset sebelum dan sesudah cleaning?
- Variabel mana yang paling banyak memiliki missing value?
- Bagaimana distribusi nilai akhir dan kehadiran siswa?
- Apakah terdapat kandidat outlier pada nilai atau jam belajar?
- Bagaimana rata-rata nilai akhir antar kelas?
- Apakah jam belajar dan kehadiran menunjukkan hubungan dengan nilai akhir?
- Temuan apa yang dapat menjadi dasar rekomendasi pembelajaran?

5.5 Tahapan Pengerjaan

1. Buat folder proyek dan simpan dataset mentah sebagai student_performance_raw.csv.
2. Baca dataset dan buat laporan profiling awal: shape, info, describe, missing values, dan duplikasi.
3. Tentukan aturan validasi untuk ID, nilai, kehadiran, dan jam belajar.
4. Standarkan tipe data, spasi, dan kategori.
5. Tangani missing values dengan strategi yang dijelaskan dalam laporan.
6. Periksa duplikasi menggunakan definisi satu observasi yang telah ditentukan.
7. Hitung nilai akhir berbobot: tugas 30%, UTS 30%, dan UAS 40%.
8. Buat visualisasi distribusi nilai, boxplot per kelas, serta hubungan jam belajar dan nilai.
9. Susun ringkasan statistik dan perbandingan antar kelas.
10. Tulis temuan, keterbatasan, dan rekomendasi; simpan dataset bersih dan notebook.

5.6 Kode Awal Mini Project

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

# 1. Load data
df = pd.read_csv("student_performance_raw.csv")
print("Ukuran awal:", df.shape)
print(df.info())
print(df.isna().sum())

# 2. Standardisasi teks
for col in ["nama", "kelas", "jenis_kelamin"]:
    df[col] = df[col].astype("string").str.strip()

df["kelas"] = df["kelas"].str.upper()

# 3. Konversi kolom numerik
numeric_cols = [
    "jam_belajar", "kehadiran",
    "nilai_tugas", "nilai_uts", "nilai_uas"
]
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors="coerce")

# 4. Periksa dan tangani duplikasi berdasarkan kunci
# Asumsi: satu siswa hanya memiliki satu baris pada periode ini.
df = df.drop_duplicates(subset=["id_siswa"], keep="first")

# 5. Validasi rentang nilai dan kehadiran
```

```
for col in ["kehadiran", "nilai_tugas", "nilai_uts", "nilai_uas"]:
    df.loc[~df[col].between(0, 100), col] = np.nan

# Jam belajar harus non-negatif
df.loc[df["jam_belajar"] < 0, "jam_belajar"] = np.nan

# 6. Imputasi numerik dengan median
for col in numeric_cols:
    if df[col].notna().any():
        df[col] = df[col].fillna(df[col].median())

# 7. Hitung nilai akhir
df["nilai_akhir"] = (
    df["nilai_tugas"] * 0.30 +
    df["nilai_uts"] * 0.30 +
    df["nilai_uas"] * 0.40
).round(2)

# 8. Profiling setelah cleaning
print("Ukuran akhir:", df.shape)
print(df.isna().sum())
print(df.describe().round(2))

# 9. Ringkasan per kelas
summary = df.groupby("kelas").agg(
    jumlah_siswa=("id_siswa", "count"),
    rata_rata=("nilai_akhir", "mean"),
    median=("nilai_akhir", "median"),
    rata_kehadiran=("kehadiran", "mean")
).round(2)
print(summary)

# 10. Visualisasi
sns.histplot(data=df, x="nilai_akhir", bins=10, kde=True)
plt.title("Distribusi Nilai Akhir")
plt.tight_layout()
plt.show()

sns.boxplot(data=df, x="kelas", y="nilai_akhir")
plt.title("Perbandingan Nilai Akhir per Kelas")
plt.tight_layout()
plt.show()

sns.scatterplot(data=df, x="jam_belajar", y="nilai_akhir", hue="kelas")
plt.title("Jam Belajar dan Nilai Akhir")
plt.tight_layout()
plt.show()

# 11. Simpan keluaran
df.to_csv("student_performance_clean.csv", index=False)
summary.to_csv("student_performance_eda_summary.csv")
```

Catatan penting untuk implementasi: kode awal ini merupakan kerangka pembelajaran. Jika kolom nilai wajib masih kosong setelah proses imputasi, jangan menghitung nilai akhir secara diam-diam. Periksa kembali kolom tersebut dan putuskan apakah observasi perlu dikecualikan atau strategi imputasi lain diperlukan. Untuk data multi-periode, jangan deduplikasi hanya berdasarkan id_siswa.

5.7 Luaran Proyek

- Notebook Jupyter (.ipynb) atau script Python (.py).
- Dataset bersih student_performance_clean.csv.
- Tabel ringkasan EDA per kelas.

- Minimal tiga visualisasi yang diberi judul, label, dan interpretasi.
- Laporan 3–5 halaman berisi tujuan, profil data, aturan cleaning, hasil EDA, keterbatasan, dan rekomendasi.
- Presentasi singkat temuan utama.

6. Kuis Evaluasi

1. Tujuan utama Data Cleaning adalah …
    - A. Memperbanyak jumlah kolom
    - B. Memperbaiki kualitas dan konsistensi data
    - C. Menghapus seluruh nilai ekstrem
    - D. Membuat grafik
2. Fungsi Pandas untuk menghitung missing value per kolom adalah …
    - A. isna().sum()
    - B. describe().sum()
    - C. dropna().count()
    - D. info().mean()
3. Median sering dipilih untuk imputasi numerik ketika …
    - A. Data selalu simetris
    - B. Data memiliki kemencengan atau outlier
    - C. Semua nilai sama
    - D. Kolom bertipe teks
4. Metode IQR menggunakan …
    - A. Mean dan modus
    - B. Q1, Q3, dan rentang antarkuartil
    - C. Nilai maksimum saja
    - D. Korelasi
5. EDA digunakan untuk …
    - A. Mengganti seluruh data mentah
    - B. Mengeksplorasi pola, distribusi, dan hubungan data
    - C. Menjamin hubungan sebab-akibat
    - D. Menghapus kebutuhan validasi
6. Grafik yang umum digunakan untuk melihat distribusi satu variabel numerik adalah …
    - A. Histogram
    - B. Pie chart untuk setiap kasus
    - C. Peta
    - D. Tabel relasi
7. Korelasi tinggi antara dua variabel berarti …
    - A. Salah satu pasti menyebabkan yang lain
    - B. Keduanya selalu memiliki nilai sama
    - C. Ada hubungan statistik, tetapi tidak otomatis kausal
    - D. Tidak ada hubungan
8. Kandidat outlier sebaiknya …
    - A. Selalu dihapus
    - B. Selalu diganti mean

- C. Ditinjau berdasarkan konteks dan sumber data
- D. Diabaikan

9. Fungsi drop_duplicates(subset=['id']) menghapus …

- A. Semua baris yang memiliki nilai kosong
- B. Baris duplikat berdasarkan kolom id
- C. Kolom id
- D. Semua nilai unik

10. Laporan EDA yang baik seharusnya …

- A. Hanya berisi grafik tanpa penjelasan
- B. Menyertakan temuan, konteks, dan keterbatasan
- C. Mengklaim sebab-akibat dari korelasi
- D. Menyembunyikan keputusan cleaning

7. Kunci Jawaban Kuis

Nomor	Jawaban	Penjelasan
1	B	Cleaning bertujuan meningkatkan kualitas data.
2	A	isna() menandai nilai kosong dan sum() menghitungnya.
3	B	Median relatif tahan terhadap nilai ekstrem.
4	B	IQR adalah Q3 dikurangi Q1.
5	B	EDA mengeksplorasi struktur, pola, dan hubungan.
6	A	Histogram memperlihatkan frekuensi dalam interval numerik.
7	C	Korelasi tidak membuktikan sebab-akibat.
8	C	Outlier perlu ditinjau dengan konteks.
9	B	subset menentukan kolom acuan duplikasi.
10	B	Temuan harus disertai konteks dan batasan.

8. Rubrik Penilaian Mini Project

Aspek	Bobot	Indikator
Profiling dan pemahaman data	15%	Mampu menjelaskan struktur, variabel, dan masalah kualitas data.
Strategi Data Cleaning	25%	Keputusan missing value, duplikasi, tipe data, dan validasi tepat serta terdokumentasi.
Analisis statistik dan EDA	20%	Statistik deskriptif dan analisis kelompok dihitung dengan benar.
Visualisasi dan interpretasi	20%	Grafik sesuai pertanyaan dan interpretasi tidak berlebihan.
Kode dan reproduksibilitas	10%	Kode terstruktur, dapat dijalankan ulang, dan menghasilkan keluaran konsisten.
Laporan dan presentasi	10%	Temuan, batasan, dan rekomendasi disampaikan jelas.

Skala penilaian: 4 = sangat baik, 3 = baik, 2 = cukup, 1 = perlu perbaikan. Nilai aspek dihitung berdasarkan skor yang diperoleh dibanding skor maksimal, kemudian dikalikan bobot.

9. Checklist Kompetensi

- Saya dapat menjelaskan dimensi kualitas data.
- Saya dapat melakukan profiling dataset menggunakan Pandas.
- Saya dapat menghitung dan menangani missing values.
- Saya dapat memeriksa duplikasi berdasarkan aturan observasi.
- Saya dapat memperbaiki tipe data dan standardisasi kategori.
- Saya dapat melakukan validasi rentang dan aturan bisnis.
- Saya dapat mengidentifikasi kandidat outlier menggunakan IQR.
- Saya dapat menghitung statistik deskriptif dan frekuensi.
- Saya dapat membuat histogram, boxplot, scatter plot, dan heatmap.
- Saya dapat menafsirkan korelasi secara hati-hati.
- Saya dapat menulis laporan EDA dengan temuan dan keterbatasan.

10. Glosarium

Istilah	Definisi
Data Cleaning	Proses mendeteksi dan menangani masalah kualitas data.
EDA	Eksplorasi data untuk memahami pola, distribusi, dan hubungan.
Missing value	Nilai yang tidak tersedia pada suatu variabel.
Imputation	Pengisian nilai kosong menggunakan metode tertentu.
Duplicate	Catatan yang berulang menurut definisi identitas observasi.
Outlier	Observasi yang jauh berbeda dari pola mayoritas data.
IQR	Interquartile Range, selisih kuartil ketiga dan pertama.
Univariat	Analisis terhadap satu variabel.
Bivariat	Analisis hubungan dua variabel.
Korelasi	Ukuran hubungan statistik antarvariabel.
Data profiling	Pemeriksaan awal struktur, kelengkapan, dan karakteristik data.

11. Referensi Belajar

- Pandas Documentation – User Guide: https://pandas.pydata.org/docs/user_guide/
- Pandas Missing Data: https://pandas.pydata.org/docs/user_guide/missing_data.html
- NumPy Documentation: https://numpy.org/doc/
- Matplotlib Tutorials: https://matplotlib.org/stable/tutorials/
- Seaborn Tutorial: https://seaborn.pydata.org/tutorial.html
- Python Documentation: https://docs.python.org/3/

Penutup

Data Cleaning dan EDA merupakan fondasi penting dalam analisis data. Dataset yang bersih tidak berarti dataset tanpa variasi atau tanpa outlier, melainkan dataset yang telah diperiksa, dipahami, dan diproses dengan keputusan yang transparan. EDA membantu analis mengajukan pertanyaan yang lebih baik dan menyampaikan temuan secara bertanggung jawab.

Setelah modul ini, peserta didik dapat melanjutkan ke Modul 11 – Data Visualization with Matplotlib & Seaborn secara lebih mendalam, kemudian pengantar Statistical Analysis atau Machine Learning.