Tugas Kelompok Data Mining

Kelas C

Kelompok : 4

Anggota kelompok:

1. Imam Mubarak.A (60900123064)
2. Habil Nasruddin (60900123072)

Step by Step Proses Pengolahan Data

Nama Dataset : penjualan_barang.csv

Sumber URL : https://www.kaggle.com/datasets/bejopamungkas/transaksi-pembelian-penjualan-sembako?select=penjualan+barang.csv

Pengolahan data ini dilakukan melalui empat tahap utama: Collecting, Integrasi, Analisis, dan Validasi. Setiap tahap menghasilkan file output tersendiri agar proses dapat ditelusuri ulang (data lineage) dan setiap perubahan dataset terdokumentasi dengan baik. Dataset yang digunakan adalah penjualan_barang.csv, berisi catatan transaksi penjualan produk.

1. Collecting Data (Pengumpulan Data)

D: > Semester 5 > Data Mining > Tahap_2_DataMining > Collecting_Data.py

```python
import pandas as pd

# === Tahap 1: Collecting Data ===
try:
    # Membaca dataset asli
    df = pd.read_csv("penjualan_barang.csv")

    # Simpan hasil tahap collecting
    df.to_csv("data_collecting.csv", index=False)

    print("✓ Tahap 1 (Collecting) selesai → data_collecting.csv berhasil dibuat")
except Exception as e:
    print("❌ Error di tahap Collecting:", e)
```

Tahap awal adalah mengumpulkan data mentah dari sumber yang tersedia.

- Sumber data: file penjualan_barang.csv.
- Dataset ini berisi informasi transaksi penjualan dengan atribut antara lain:
  - tanggal_transaksi → tanggal transaksi penjualan.
  - id_barang → kode unik barang.

- nama_barang → nama produk yang dijual.
- jumlah → jumlah unit barang yang dibeli.
- harga_satuan → harga satuan barang.
- total_harga → hasil perkalian jumlah × harga satuan.

Langkah yang dilakukan:

- Membaca file CSV menggunakan library pandas.
- Menyimpan ulang data mentah ke file data_collecting.csv tanpa perubahan.

Tujuan:

- Menyediakan salinan awal dataset sebelum dilakukan pembersihan.
- Jika ada kesalahan pada tahap berikutnya, data mentah ini bisa menjadi referensi ulang.

Output: data_collecting.csv.

2. Integrasi Data (Data Integration)

```python
import pandas as pd

# === Tahap 2: Integrasi Data ===
try:
    df = pd.read_csv("data_collecting.csv")

    # Hapus baris duplikat
    df_integration = df.drop_duplicates()

    # Normalisasi nama kolom
    df_integration.columns = [col.strip().lower().replace(" ", "_") for col in df_integration.columns]

    # Simpan hasil integrasi
    df_integration.to_csv("data_integration.csv", index=False)

    print("✓ Tahap 2 (Integrasi) selesai → data_integration.csv berhasil dibuat")
except Exception as e:
    print("❌ Error di tahap Integrasi:", e)
```

Setelah data dikumpulkan, tahap berikutnya adalah integrasi dan standarisasi data agar lebih konsisten.

Langkah yang dilakukan:

- Menghapus baris duplikat untuk menghindari penghitungan ganda.
- Menstandarkan nama kolom: semua huruf kecil dan spasi diganti dengan underscore (_).
- Mengecek konsistensi tipe data dasar (misalnya, kolom jumlah & harga tetap numerik).

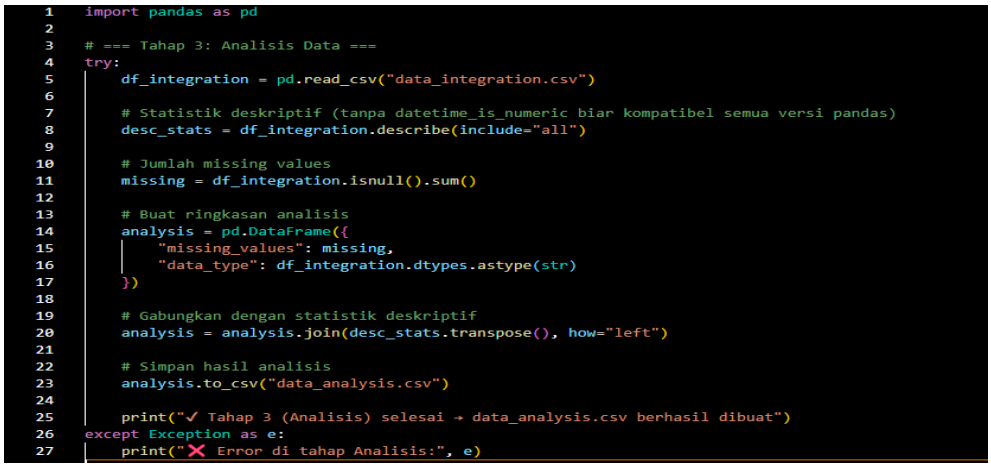
Tujuan:

- Memastikan dataset rapi, konsisten, dan siap dipakai untuk analisis.
- Tahap integrasi penting terutama jika data berasal dari berbagai sumber

(misalnya beberapa file penjualan dari cabang berbeda).

Output: data_integration.csv.

3. Analisis Data (Data Analysis)



```python
import pandas as pd

# === Tahap 3: Analisis Data ===
try:
    df_integration = pd.read_csv("data_integration.csv")

    # Statistik deskriptif (tanpa datetime_is_numeric biar kompatibel semua versi pandas)
    desc_stats = df_integration.describe(include="all")

    # Jumlah missing values
    missing = df_integration.isnull().sum()

    # Buat ringkasan analisis
    analysis = pd.DataFrame({
        "missing_values": missing,
        "data_type": df_integration.dtypes.astype(str)
    })

    # Gabungkan dengan statistik deskriptif
    analysis = analysis.join(desc_stats.transpose(), how="left")

    # Simpan hasil analisis
    analysis.to_csv("data_analysis.csv")

    print("✓ Tahap 3 (Analisis) selesai → data_analysis.csv berhasil dibuat")
except Exception as e:
    print("❌ Error di tahap Analisis:", e)
```

Pada tahap ini, dilakukan analisis eksplorasi awal untuk memahami isi dataset.

Langkah yang dilakukan:

- Menghitung statistik deskriptif: rata-rata, median, minimum, maksimum, dan standar deviasi pada kolom numerik.
- Mengidentifikasi jumlah missing values pada setiap kolom.
- Menyimpan tipe data (int, float, object) setiap kolom untuk memastikan kesesuaian.
- Menyimpan ringkasan analisis ke file CSV.

Contoh hasil analisis:

- Rata-rata kolom jumlah menunjukkan barang paling sering dibeli dalam jumlah kecil (1–5 unit).
- Kolom harga_satuan bervariasi, ada produk murah dan produk premium.
- Terdapat missing values pada total_harga karena beberapa baris belum terisi dengan benar.

Tujuan:

- Memberikan gambaran umum dataset.
- Menentukan masalah kualitas data yang harus ditangani di tahap validasi.

Output: data_analysis.csv.

4. Validasi Data (Data Validation)

```python
import pandas as pd

# === Tahap 4: Validasi Data ===
try:
    df_integration = pd.read_csv("data_integration.csv")

    # Hapus baris yang semua kolomnya kosong
    df_validation = df_integration.dropna(how="all")

    # Isi missing values: numerik → median, kategorikal → mode
    for col in df_validation.columns:
        if df_validation[col].dtype in ["int64", "float64"]:
            median_val = df_validation[col].median()
            df_validation[col] = df_validation[col].fillna(median_val)
        else:
            if not df_validation[col].mode().empty:
                mode_val = df_validation[col].mode()[0]
                df_validation[col] = df_validation[col].fillna(mode_val)

    # Simpan hasil validasi
    df_validation.to_csv("data_validation.csv", index=False)

    print("✓ Tahap 4 (Validasi) selesai → data_validation.csv berhasil dibuat")
except Exception as e:
    print("❌ Error di tahap Validasi:", e)
```

Tahap validasi bertujuan memastikan dataset final bersih dan bebas masalah.

Langkah yang dilakukan:

- Menghapus baris yang seluruh kolomnya kosong.
- Mengisi nilai kosong (imputasi):
    - Numerik → diganti dengan median (agar tidak bias terhadap nilai ekstrem).
    - Kategorikal → diganti dengan mode (nilai yang paling sering muncul).
- Memastikan tidak ada lagi nilai NaN setelah proses validasi.

Tujuan:

- Menghasilkan dataset final yang bisa langsung dipakai untuk analisis lanjutan, machine learning, atau visualisasi.
- Menjamin kualitas data agar hasil analisis tidak menyesatkan.

Output: data_validation.csv.

Kesimpulan

Proses pengolahan data dari penjualan_barang.csv telah mengikuti alur standar:

1. Dataset mentah disimpan ulang di data_collecting.csv.
2. Dataset dibersihkan dan disatukan di data_integration.csv.
3. Hasil analisis eksplorasi disimpan di data_analysis.csv.
4. Dataset final yang sudah tervalidasi disimpan di data_validation.csv.

Dengan menyimpan output di setiap tahap, proses ini memenuhi prinsip data lineage, sehingga setiap perubahan dapat dilacak dan dikembalikan ke versi sebelumnya jika diperlukan.

Tabel Ringkasan

Tahap	File Hasil	Perubahan Utama
Collecting	data_collecting.csv	Salinan dataset mentah dari penjualan_barang
Integrasi	data_integration.csv	Hapus duplikat, standarisasi nama kolom
Analisis	data_analysis.csv	Statistik deskriptif, missing values, tipe data
Validasi	data_validation.csv	Isi missing values (median/mode), data final