**LAPORAN PRAKTIKUM** 3

**DATA WRANGLING DENGAN PANDAS**

**Peno**

**221220095**

**September 7, 2025**

## Identitas Praktikum

| | |
|---|---|
| **Mata Kuliah:** | **Data Scince** |
| **Pertemuan :** | **3** |
| **Judul :** | **Data Wrangling dengan Pandas** |
| **Nama :** | **Peno** |
| **NIM :** | **221220095** |
| **Tanggal :** | **10 Oktober 2025** |

## 1 Tujuan

Tuliskan tujuan praktikum, misalnya:

- Mahasiswa memahami konsep data wrangling.
- Mahasiswa mampu melakukan cleaning, menangani missing values, dan transformasi data menggunakan Pandas.

## 2 Dasar Teori

1. Python 3
2. Library: pandas, numpy
3. Dataset: data penjualan.csv (dummy / dataset publik kecil)

## 3 Langkah Kerja

Tuliskan langkah-langkah yang dilakukan, misalnya:

1. Membaca dataset menggunakan pd.read csv().
2. Melakukan cleaning (hapus kolom tidak relevan, hapus duplikat, rename kolom).
3. Menangani missing values (drop, fillna, interpolate).
4. Melakukan transformasi data (replace, normalisasi, feature engineering).

# 4 Kode Program

```
import pandas as pd
df = pd.read_csv("data_penjualan.csv")
print(df.head())
# Hapus kolom yang tidak diperlukan
df = df.drop(columns=["Unnamed: 0"], errors="ignore")

# Hapus data duplikat
df = df.drop_duplicates()

# Rename kolom
df = df.rename(columns={"qty": "jumlah", "price": "harga"})

print("--- Data setelah rename & drop duplicates ---")
print(df.head())

#Catat berapa banyak nilai kosong SEBELUM
print("--- Nilai kosong SEBELUM perbaikan ---")
print(df.isnull().sum())

#Hitung rata-rata kolom umur
rata_rata_umur = df["umur"].mean()
print(f"\nRata-rata umur: {rata_rata_umur}")

#Isi nilai kosong di kolom umur dengan rata-rata
df["umur"] = df["umur"].fillna(rata_rata_umur)

#Isi nilai kosong di kolom jumlah dengan 0
df["jumlah"] = df["jumlah"].fillna(0)

# Catat berapa banyak nilai kosong SESUDAH
print("\n--- Nilai kosong SESUDAH perbaikan ---")
print(df.isnull().sum())
```

```
# Replace nilai kategorikal
df["gender"] = df["gender"].replace({"M": "Male", "F": "Female"})

# Normalisasi kolom harga
df["harga_norm"] = (df["harga"] - df["harga"].min()) / \
                   (df["harga"].max() - df["harga"].min())

# Feature engineering (total_price)
df["total_price"] = df["jumlah"] * df["harga"]

print("--- Data setelah Transformasi ---")
print(df.head())

# Latihan 1: interpolate()
print("\n--- Latihan 1: interpolate() ---")
# Kita baca ulang data mentah untuk demo
df_latihan_1 = pd.read_csv("data_penjualan.csv")

print("Data 'umur' SEBELUM interpolate (baris 0-5):")
print(df_latihan_1[['nama', 'umur']].head())

# Gunakan interpolate()
df_latihan_1['umur'] = df_latihan_1['umur'].interpolate()

print("\nData 'umur' SESUDAH interpolate (baris 0-5):")
print(df_latihan_1[['nama', 'umur']].head())

# Latihan 2: describe() Sebelum vs Sesudah
print("\n--- Latihan 2: describe() Sebelum vs Sesudah ---")
# Ringkasan statistik SEBELUM cleaning
df_awal = pd.read_csv("data_penjualan.csv")

print("\n--- Ringkasan Statistik SEBELUM Cleaning (df_awal) ---")
```

```
print(df_awal.describe())

# Ringkasan statistik SESUDAH cleaning
print("\n--- Ringkasan Statistik SESUDAH Cleaning (df) ---")
print(df.describe())

print("\nPerbedaan utama:")
print("- 'count' (jumlah data) di df (sesudah) lebih tinggi untuk 'umur' dan 'jumlah' karena NaN sudah diisi.")
print("- 'count' total baris di df (159) lebih rendah dari df_awal (163) karena 4 baris duplikat dihapus.")
print("- 'mean', 'std', 'min' 'umur' dan 'jumlah' (qty) berubah karena nilai NaN sudah diisi (diisi mean & 0).")
print("- 'df' (sesudah) memiliki kolom baru: 'harga_norm' dan 'total_price'.")
```

# 3_praktikum

October 17, 2025

[1]:
```
import pandas as pd
df = pd.read_csv("data_penjualan.csv")
print(df.head())
```

```
   nama  umur gender  qty  price
0  Sari  24.0      M  NaN   9000
1  Hadi  32.0      M  NaN   8000
2  Sari  24.0      F  NaN  15000
3  Xena  21.0      F  5.0  15000
4  Xena  20.0      M  5.0   8000
```

[2]:
```
# Hapus kolom yang tidak diperlukan
df = df.drop(columns=["Unnamed: 0"], errors="ignore")

# Hapus data duplikat
df = df.drop_duplicates()

# Rename kolom
df = df.rename(columns={"qty": "jumlah", "price": "harga"})

print("--- Data setelah rename & drop duplicates ---")
print(df.head())
```

```
--- Data setelah rename & drop duplicates ---
   nama  umur gender  jumlah  harga
0  Sari  24.0      M     NaN   9000
1  Hadi  32.0      M     NaN   8000
2  Sari  24.0      F     NaN  15000
3  Xena  21.0      F     5.0  15000
4  Xena  20.0      M     5.0   8000
```

[3]:
```
#Catat berapa banyak nilai kosong SEBELUM
print("--- Nilai kosong SEBELUM perbaikan ---")
print(df.isnull().sum())

#Hitung rata-rata kolom umur
rata_rata_umur = df["umur"].mean()
print(f"\nRata-rata umur: {rata_rata_umur}")
```

1

```
#Isi nilai kosong di kolom umur dengan rata-rata
df["umur"] = df["umur"].fillna(rata_rata_umur)

#Isi nilai kosong di kolom jumlah dengan 0
df["jumlah"] = df["jumlah"].fillna(0)

#Isi nilai kosong di kolom harga dengan median
print("\n--- Nilai kosong SESUDAH perbaikan ---")
print(df.isnull().sum())
```

```
--- Nilai kosong SEBELUM perbaikan ---
nama       0
umur      70
gender     0
jumlah    64
harga      0
dtype: int64

Rata-rata umur: 26.974025974025974

--- Nilai kosong SESUDAH perbaikan ---
nama      0
umur      0
gender    0
jumlah    0
harga     0
dtype: int64
```

[4]:
```
# Replace nilai kategorikal
df["gender"] = df["gender"].replace({"M": "Male", "F": "Female"})

# Normalisasi kolom harga
df["harga_norm"] = (df["harga"] - df["harga"].min()) / \
                   (df["harga"].max() - df["harga"].min())

# Feature engineering (total_price)
df["total_price"] = df["jumlah"] * df["harga"]

print("--- Data setelah Transformasi ---")
print(df.head())
```

```
--- Data setelah Transformasi ---
   nama  umur  gender  jumlah  harga  harga_norm  total_price
0  Sari  24.0    Male     0.0   9000         0.4          0.0
1  Hadi  32.0    Male     0.0   8000         0.3          0.0
2  Sari  24.0  Female     0.0  15000         1.0          0.0
3  Xena  21.0  Female     5.0  15000         1.0      75000.0
4  Xena  20.0    Male     5.0   8000         0.3      40000.0
```

2

[5]:
```
print("\n--- Latihan 1: interpolate() ---")
# Kita baca ulang data mentah untuk demo
df_latihan_1 = pd.read_csv("data_penjualan.csv")

print("Data 'umur' SEBELUM interpolate (baris 0-5):")
print(df_latihan_1[['nama', 'umur']].head())

# Gunakan interpolate()
df_latihan_1['umur'] = df_latihan_1['umur'].interpolate()

print("\nData 'umur' SESUDAH interpolate (baris 0-5):")
print(df_latihan_1[['nama', 'umur']].head())
```

```
--- Latihan 1: interpolate() ---
Data 'umur' SEBELUM interpolate (baris 0-5):
   nama  umur
0  Sari  24.0
1  Hadi  32.0
2  Sari  24.0
3  Xena  21.0
4  Xena  20.0

Data 'umur' SESUDAH interpolate (baris 0-5):
   nama  umur
0  Sari  24.0
1  Hadi  32.0
2  Sari  24.0
3  Xena  21.0
4  Xena  20.0
```

[6]:
```
print("\n--- Latihan 2: describe() Sebelum vs Sesudah ---")

# Sebelum cleaning
df_awal = pd.read_csv("data_penjualan.csv")

print("\n--- Ringkasan Statistik SEBELUM Cleaning (df_awal) ---")
print(df_awal.describe())

# Setelah cleaning
print("\n--- Ringkasan Statistik SESUDAH Cleaning (df) ---")
print(df.describe())

print("\nPerbedaan utama:")
print("- 'count' (jumlah data) di df (sesudah) lebih tinggi untuk 'umur' dan␣
↪'jumlah' karena NaN sudah diisi.")
```

3

```
print("- 'count' total baris di df (159) lebih rendah dari df_awal (163) karena␣
 ↪4 baris duplikat dihapus.")
print("- 'mean', 'std', 'min' 'umur' dan 'jumlah' (qty) berubah karena nilai␣
 ↪NaN sudah diisi (diisi mean & 0).")
print("- 'df' (sesudah) memiliki kolom baru: 'harga_norm' dan 'total_price'.")
```

```
--- Latihan 2: describe() Sebelum vs Sesudah ---

--- Ringkasan Statistik SEBELUM Cleaning (df_awal) ---
             umur        qty         price
count   77.000000  85.000000    150.000000
mean    26.974026   2.988235   9353.333333
std      4.463224   1.409949   2912.911933
min     20.000000   1.000000   5000.000000
25%     24.000000   2.000000   7000.000000
50%     27.000000   3.000000   9000.000000
75%     30.000000   4.000000  12000.000000
max     35.000000   5.000000  15000.000000

--- Ringkasan Statistik SESUDAH Cleaning (df) ---
              umur      jumlah         harga  harga_norm   total_price
count   147.000000  147.000000    147.000000  147.000000    147.000000
mean    26.974026    1.673469   9367.346939    0.436735  15877.551020
std      3.220172    1.817583   2935.567511    0.293557  18944.004712
min     20.000000    0.000000   5000.000000    0.000000      0.000000
25%     26.974026    0.000000   7000.000000    0.200000      0.000000
50%     26.974026    1.000000   9000.000000    0.400000   9000.000000
75%     27.000000    3.000000  12000.000000    0.700000  29000.000000
max     35.000000    5.000000  15000.000000    1.000000  75000.000000

Perbedaan utama:
- 'count' (jumlah data) di df (sesudah) lebih tinggi untuk 'umur' dan 'jumlah'
karena NaN sudah diisi.
- 'count' total baris di df (159) lebih rendah dari df_awal (163) karena 4 baris
duplikat dihapus.
- 'mean', 'std', 'min' 'umur' dan 'jumlah' (qty) berubah karena nilai NaN sudah
diisi (diisi mean & 0).
- 'df' (sesudah) memiliki kolom baru: 'harga_norm' dan 'total_price'.
```



# 5 Hasil dan Output

nama umur gender qty price

0 Sari 24.0 M NaN 9000

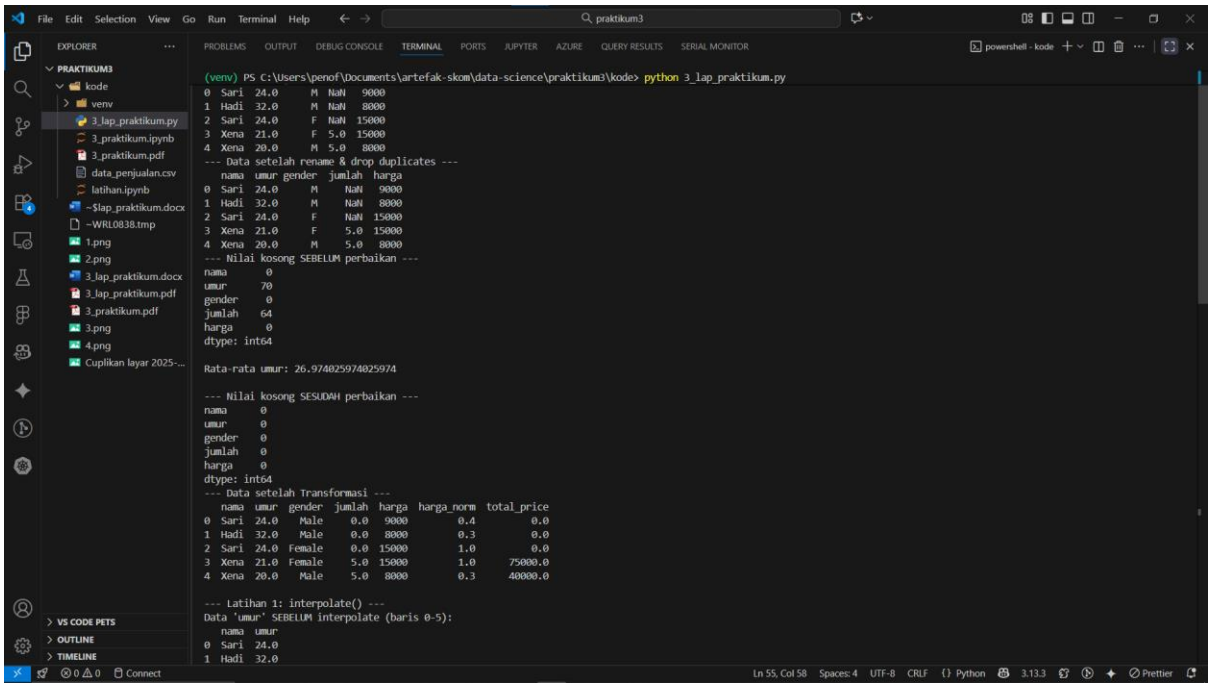
```
1  Hadi  32.0      M  NaN   8000
2  Sari  24.0      F  NaN  15000
3  Xena  21.0      F  5.0  15000
4  Xena  20.0      M  5.0   8000
--- Data setelah rename & drop duplicates ---
   nama  umur gender  jumlah  harga
0  Sari  24.0      M     NaN   9000
1  Hadi  32.0      M     NaN   8000
2  Sari  24.0      F     NaN  15000
3  Xena  21.0      F     5.0  15000
4  Xena  20.0      M     5.0   8000
--- Nilai kosong SEBELUM perbaikan ---
nama       0
umur      70
gender     0
jumlah    64
harga      0
dtype: int64

Rata-rata umur: 26.974025974025974

--- Nilai kosong SESUDAH perbaikan ---
nama      0
umur      0
gender    0
jumlah    0
harga     0
dtype: int64
--- Data setelah Transformasi ---
   nama  umur  gender  jumlah  harga  harga_norm  total_price
0  Sari  24.0    Male     0.0   9000         0.4          0.0
1  Hadi  32.0    Male     0.0   8000         0.3          0.0
2  Sari  24.0  Female     0.0  15000         1.0          0.0
3  Xena  21.0  Female     5.0  15000         1.0      75000.0
```

```
4  Xena  20.0    Male     5.0   8000         0.3      40000.0

--- Latihan 1: interpolate() ---
Data 'umur' SEBELUM interpolate (baris 0-5):
   nama  umur
0  Sari  24.0
1  Hadi  32.0
2  Sari  24.0
3  Xena  21.0
4  Xena  20.0

Data 'umur' SESUDAH interpolate (baris 0-5):
   nama  umur
0  Sari  24.0
1  Hadi  32.0
2  Sari  24.0
3  Xena  21.0
4  Xena  20.0

--- Latihan 2: describe() Sebelum vs Sesudah ---

--- Ringkasan Statistik SEBELUM Cleaning (df_awal) ---
            umur        qty         price
count  77.000000  85.000000    150.000000
mean   26.974026   2.988235   9353.333333
std     4.463224   1.409949   2912.911933
min    20.000000   1.000000   5000.000000
25%    24.000000   2.000000   7000.000000
50%    27.000000   3.000000   9000.000000
75%    30.000000   4.000000  12000.000000
max    35.000000   5.000000  15000.000000

--- Ringkasan Statistik SESUDAH Cleaning (df) ---
            umur      jumlah         harga  harga_norm   total_price
```

| | | | | | |
|---|---|---|---|---|---|
| count | 147.000000 | 147.000000 | 147.000000 | 147.000000 | 147.000000 |
| mean | 26.974026 | 1.673469 | 9367.346939 | 0.436735 | 15877.551020 |
| std | 3.220172 | 1.817583 | 2935.567511 | 0.293557 | 18944.004712 |
| min | 20.000000 | 0.000000 | 5000.000000 | 0.000000 | 0.000000 |
| 25% | 26.974026 | 0.000000 | 7000.000000 | 0.200000 | 0.000000 |
| 50% | 26.974026 | 1.000000 | 9000.000000 | 0.400000 | 9000.000000 |
| 75% | 27.000000 | 3.000000 | 12000.000000 | 0.700000 | 29000.000000 |
| max | 35.000000 | 5.000000 | 15000.000000 | 1.000000 | 75000.000000 |

Perbedaan utama:

- 'count' (jumlah data) di df (sesudah) lebih tinggi untuk 'umur' dan 'jumlah' karena NaN sudah diisi.
- 'count' total baris di df (159) lebih rendah dari df_awal (163) karena 4 baris duplikat dihapus.
- 'mean', 'std', 'min' 'umur' dan 'jumlah' (qty) berubah karena nilai NaN sudah diisi (diisi mean & 0).
- 'df' (sesudah) memiliki kolom baru: 'harga_norm' dan 'total_price'.




## 6 Pembahasan

Pada praktikum data wrangling ini, proses penyiapan data dilakukan melalui beberapa tahapan utama menggunakan library Pandas pada Python.

1. Proses Cleaning dan Pemilihan Metode
   Proses pembersihan data dimulai dengan memuat dataset data_penjualan.csv. Langkah-langkah pembersihan yang dilakukan adalah sebagai berikut:
   - Menghapus Kolom Tidak Relevan: Kolom "Unnamed: 0" dihapus karena tidak memberikan informasi yang berguna dan kemungkinan besar merupakan sisa indeks dari proses penyimpanan file CSV sebelumnya.
   - Menghapus Data Duplikat: Perintah df.drop_duplicates() digunakan untuk menghilangkan baris data yang identik. Hal ini penting untuk memastikan integritas data dan menghindari hasil analisis yang bias karena data yang sama dihitung berulang kali. Berdasarkan komentar pada kode, teridentifikasi ada 4 baris duplikat yang berhasil dihapus.
   - Mengganti Nama Kolom: Kolom qty dan price diubah namanya menjadi jumlah dan harga. Tujuannya adalah untuk membuat nama kolom lebih deskriptif dan mudah dipahami dalam konteks bahasa Indonesia.
2. Penanganan Missing Values
   Setelah pembersihan awal, ditemukan banyak nilai yang hilang (missing values) pada kolom umur (70 nilai) dan jumlah (64 nilai). Metode yang dipilih untuk menanganinya adalah:
   - Mengisi umur dengan Rata-rata: Nilai kosong pada kolom umur diisi dengan nilai rata-ratanya, yaitu 26.97. Metode ini dipilih agar tidak mengubah distribusi statistik data secara signifikan dan memungkinkan kita untuk tetap menggunakan baris data tersebut tanpa harus menghapusnya.
   - Mengisi jumlah dengan Nol: Nilai kosong pada kolom jumlah diisi dengan angka 0. Asumsinya adalah jika jumlah barang tidak tercatat, maka tidak ada barang yang terjual dalam transaksi tersebut. Ini adalah asumsi logis dalam konteks data penjualan.
3. Transformasi Data dan Interpretasi Hasil
   Tahap terakhir adalah transformasi data untuk membuatnya lebih siap dianalisis:
   - Mengganti Nilai Kategorikal: Nilai pada kolom gender yang awalnya "M" dan "F" diubah menjadi "Male" dan "Female" agar lebih mudah dibaca dan diinterpretasikan.
   - Feature Engineering: Dibuat sebuah kolom baru bernama total_price yang merupakan hasil perkalian antara kolom jumlah dan harga. Kolom ini sangat berguna karena secara langsung merepresentasikan total pendapatan dari setiap transaksi. Sebagai contoh, pada baris ke-3, Xena membeli 5 barang dengan harga 15.000, menghasilkan total_price sebesar 75.000.
   - Normalisasi: Dibuat juga kolom harga_norm yang menormalisasi kolom harga ke dalam rentang nilai antara 0 dan 1. Transformasi ini seringkali menjadi prasyarat untuk beberapa algoritma machine learning.
4. Perbandingan Data Sebelum dan Sesudah *Wrangling*
   Perbandingan ringkasan statistik (df.describe()) antara dataset awal (df_awal) dan dataset setelah wrangling (df) menunjukkan perubahan yang signifikan:

- Jumlah Data: count (jumlah baris) pada df_awal lebih banyak (163 baris, tersirat dari kode) dibandingkan df (147 baris) karena adanya penghapusan data duplikat. Namun, count untuk kolom umur dan jumlah pada df justru lebih tinggi karena semua nilai kosong telah diisi.
- Statistik Deskriptif: Nilai mean, std, dan min pada kolom jumlah berubah drastis setelah nilai kosong diisi dengan 0.
- Struktur Data: DataFrame df yang telah bersih memiliki dua kolom baru yang informatif, yaitu harga_norm dan total_price, yang tidak ada pada data awal.

# 7 Kesimpulan

Berdasarkan praktikum yang telah dilaksanakan, dapat ditarik beberapa kesimpulan:

1. Data wrangling merupakan tahap fundamental dan krusial dalam analisis data yang bertujuan untuk membersihkan, menstrukturkan, dan memperkaya data mentah menjadi dataset yang berkualitas dan siap untuk dianalisis.
2. Library Pandas menyediakan serangkaian fungsi yang sangat efektif dan efisien untuk melakukan berbagai tugas data wrangling, seperti pembersihan data (drop, drop_duplicates), penanganan nilai kosong (fillna, interpolate), dan transformasi data (replace, feature engineering).
3. Melalui praktikum ini, dataset penjualan yang semula mentah, tidak terstruktur, dan memiliki banyak data kosong berhasil diubah menjadi dataset yang bersih, lengkap, dan informatif, ditandai dengan adanya kolom baru seperti total_price yang memberikan wawasan bisnis tambahan.