

# Tugas 2: Laporan Latihan Tugas 2

Noer Muhammad Ayub - 0110222142

<sup>1</sup> Teknik Informatika, STT Terpadu Nurul Fikri, Depok

\*E-mail: [noer22142ti@student.nurulfikri.ac.id](mailto:noer22142ti@student.nurulfikri.ac.id)

## 1. Menghubungkan dengan Google Drive

```
#menghubungkan colab dengan google drive
from google.colab import drive
drive.mount('/content/gdrive')

Drive already mounted at /content/gdrive; to attempt to forcibly remount,

path = "/content/gdrive/MyDrive/praktikum/praktikum2/data"
```

Sel ini berfungsi untuk menghubungkan lingkungan Google Colab dengan akun Google Drive-mu. Setelah kode ini dijalankan, akan muncul tautan otorisasi. Kamu harus mengklik tautan tersebut, memilih akun Google, dan memberikan izin agar Colab bisa mengakses file-file yang tersimpan di Google Drive-mu. Proses ini hanya perlu dilakukan satu kali per sesi.

### 1.1 Membaca File CSV

```
#membaca file csv menggunakan pandas
import pandas as pd
df = pd.read_csv (path + '/500_Person_Gender_Height_Weight_Index.csv')
df.head()
```

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3

Sel ini menggunakan library Pandas untuk membaca file data. Variabel path menyimpan lokasi folder di Google Drive tempat file 500\_Person\_Gender\_Height\_Weight\_Index.csv berada. Fungsi `pd.read_csv()` kemudian membaca file tersebut dan menyimpannya ke dalam sebuah DataFrame bernama `df`.

Menjalankan `df.head()` di akhir sel akan menampilkan lima baris pertama dan terakhir dari data. Akan menampilkan 500 baris dan 4 kolom dari dataset tersebut, yaitu 'Gender', 'Height', 'Weight', dan 'Index'. Ini memberikan gambaran awal tentang struktur data.

## 2. Statistik Deskriptif dan Probabilitas Dasar

### 2.1 Melihat Informasi Umum Data

```
[11]
✓ Os df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Gender  500 non-null    object
1   Height  500 non-null    int64
2   Weight  500 non-null    int64
3   Index   500 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

Metode `.info()` memberikan ringkasan singkat tentang DataFrame. Ini sangat penting untuk langkah awal analisis data karena menampilkan informasi berikut:

- Jumlah baris 500 entri dan 4 kolom, nama-nama kolom, Jumlah data non-null (data yang tidak kosong) per kolom, tipe data (Dtype) dari setiap kolom, seperti `int64` (integer) dan `object` (biasanya string).

### 2.2 Menghitung Nilai-Nilai Sentral (Mean, Median, Modus):

```
df['Height'].mean()
np.float64(169.944)

df['Height'].median()
170.5

df['Height'].mode()
Height
0    188
dtype: int64
```

Kode ini secara spesifik menghitung nilai-nilai sentral:

- `mean()`: Rata-rata dari setiap kolom numerik.
- `.median()`: Nilai tengah dari setiap kolom numerik.
- `.mode().iloc[0]`: Modus atau nilai yang paling sering muncul.

## 2.3 Menghitung Nilai-Nilai Sentral (Mean, Median, Modus):

```
#Menghitung variasi dan Standard Deviasi
df.var(numeric_only=True)
```

	0
Height	268.149162
Weight	1048.633267
Index	1.836168

dtype: float64

```
#Menghitung standar deviasi
df.std(numeric_only=True)
```

	0
Height	16.375261
Weight	32.382607
Index	1.355053

dtype: float64

### Menghitung Variasi

- `df.var()` → menghitung **variansi** dari setiap kolom numerik.
- `numeric_only=True` → supaya hanya kolom berisi angka yang dihitung (mengabaikan teks seperti Gender).

**Variansi** adalah ukuran seberapa jauh data menyebar dari rata-ratanya.

## 2.4 Menghitung Kuartil

```
[14] q1 = df['Height'].quantile(0.25)
      print("Q1: ", q1)
```

Q1: 156.0

```
[15] q3 = df['Height'].quantile(0.75)
      print("Q3: ", q3)
```

Q3: 184.0

```
[16] iqr = q3 - q1
      print("IQR : ", iqr)
```

IQR : 28.0

`df['Height'].quantile(0.25)`

dan `df['Height'].quantile(0.75)` menghitung kuartil pertama (Q1) dan kuartil ketiga (Q3). IQR dihitung dengan mengurangkan Q3 dengan Q1.

### Cari Q1 (Kuartil 1)

- Ambil kuartil 1 (25% data terbawah).
- Hasil: **156.0** → artinya 25% orang punya tinggi  $\leq 156$  cm.

### Cari Q3 (Kuartil 3)

- Ambil kuartil 3 (75% data terbawah).
- Hasil: **184.0** → artinya 75% orang punya tinggi  $\leq 184$  cm.

### Hitung IQR (Interquartile Range)

- $IQR = Q3 - Q1$
- Hasil: **28.0**
- Artinya: rentang tengah **50% data tinggi badan** ada di antara 156 cm sampai 184 cm, dengan jarak 28 cm.

## 2.5 Menghitung Statistik Deskriptif Otomatis:

```
#untuk membuat statistika deskripsi pada type data int
df.describe()
```

	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

Metod.describe() secara otomatis menghitung statistik deskriptif dasar untuk semua kolom numerik. Ini memberikan gambaran cepat tentang distribusi data.

Hasilnya adalah sebuah tabel yang berisi metrik-metrik berikut untuk setiap kolom numerik:

- count: Jumlah data non-null.
- mean: Rata-rata.
- std: Standar deviasi.
- min: Nilai minimum.
- 25%, 50%, 75%: Kuartil pertama, median, dan kuartil ketiga.
- max: Nilai maksimum.

## 2.6 Menghitung Korelasi

```
correlation_matrix = df.corr(numeric_only=True)
print('Matrix Korelasi:')
print(correlation_matrix)
```

	Height	Weight	Index
Height	1.000000	0.000446	-0.422223
Weight	0.000446	1.000000	0.804569
Index	-0.422223	0.804569	1.000000

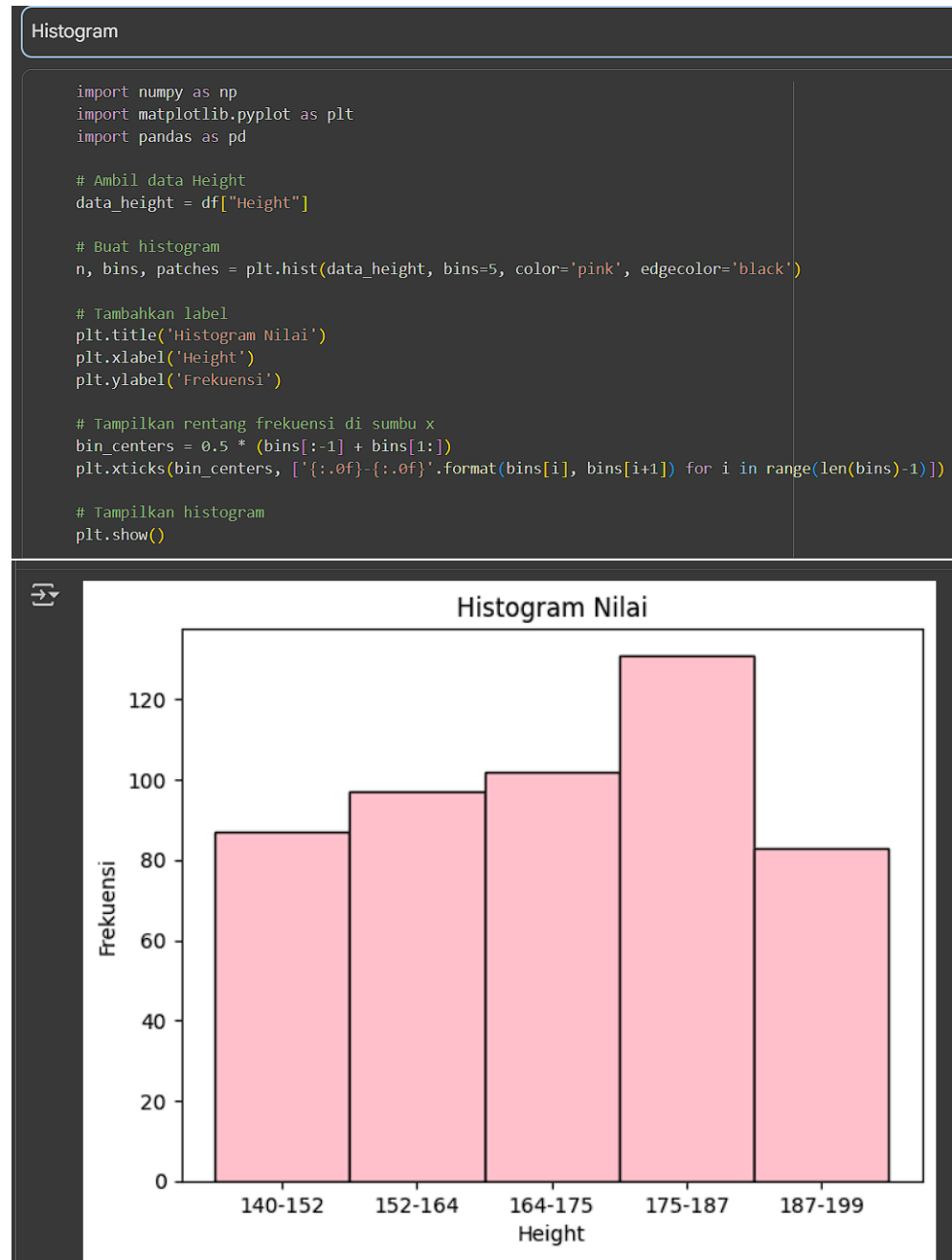
**df.corr()** → fungsi pandas untuk menghitung korelasi antar kolom numerik pada DataFrame.

**numeric\_only=True** → supaya hanya kolom tipe numerik (int/float) yang dihitung. Kolom non-numerik (string, tanggal, dsb.) otomatis diabaikan.

Hasilnya adalah sebuah matriks korelasi dalam bentuk DataFrame, dengan nilai antara -1 sampai +1.

**print('Matrix Korelasi:')** : hanya menampilkan teks biasa, yaitu judul supaya output lebih jelas.  
**print(correlation\_matrix)** : menampilkan isi dari variabel correlation\_matrix, yaitu hasil perhitungan korelasi antar kolom numerik pada DataFrame.

### 3. Visualisasi Data



**Penjelasan per bagian:**

**1. Import library**

- numpy, matplotlib.pyplot, pandas untuk olah data dan visualisasi.

**2. Ambil kolom Height**

- data\_height = df["Height"]
- hanya mengambil data tinggi badan dari DataFrame df.

### 3. Buat histogram

`n, bins, patches = plt.hist(data_height, bins=5, color='pink', edgecolor='black')`

- `bins=5` → membagi data menjadi 5 interval (kelompok).
- `color='pink'` → warna batang histogram.
- `edgecolor='black'` → garis tepi hitam.
- `n` → jumlah frekuensi di tiap bin.
- `bins` → batas interval tiap bin.
- `patches` → objek batang histogram.

### 4. Tambahkan judul dan label

- `plt.title()`, `plt.xlabel()`, `plt.ylabel()` → memberi informasi sumbu.

### 5. Custom label sumbu x

- `bin_centers = 0.5 * (bins[:-1] + bins[1:])`
- `plt.xticks(bin_centers, ['{:0f}-{:0f}'.format(bins[i], bins[i+1]) for i in range(len(bins)-1)])`
- membuat label sumbu-x jadi rentang (misalnya 140–150, 150–160, dst), bukan hanya angka batas.

### 6. Tampilkan plot

- `plt.show()` : menampilkan histogram.

## 3. Scatter Plot (Hubungan Antar Variabel):

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

# Buat DataFrame contoh
data = {
    'Nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Nilai2': [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
}

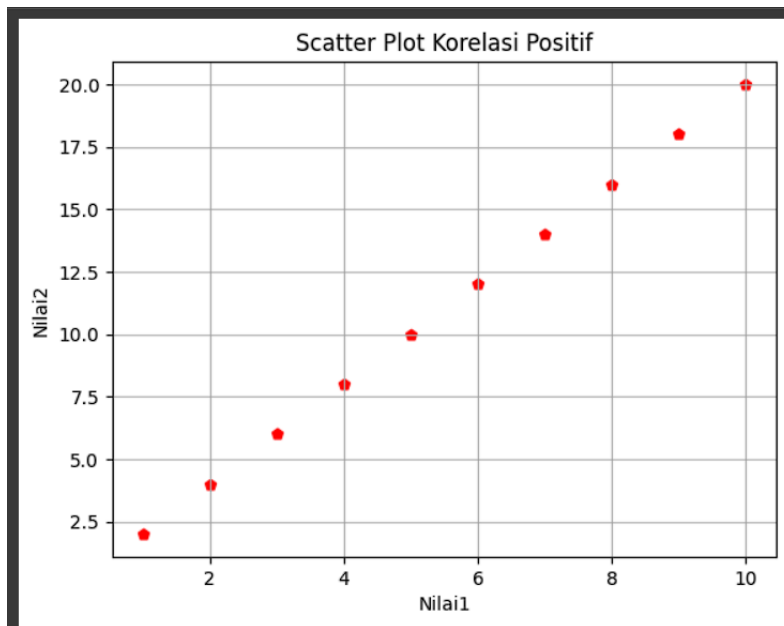
df2 = pd.DataFrame(data)

# Buat scatter plot
plt.scatter(df2['Nilai1'], df2['Nilai2'], color='red', marker='p')

# Tambahkan label
plt.title('Scatter Plot Korelasi Positif')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()
```



### Penjelasan Perbagian:

#### 1. Membuat DataFrame

- `data = {...}`
- `df2 = pd.DataFrame(data)`
- `Nilai1 = 1 s/d 10`
- `Nilai2 = kelipatan 2 dari Nilai1` Jadi datanya punya hubungan **linear sempurna**:  
$$\text{Nilai2} = 2 \times \text{Nilai1}.$$

#### 2. Scatter plot

```
plt.scatter(df2['Nilai1'], df2['Nilai2'], color='red', marker='p')
```

- `x = Nilai1`
- `y = Nilai2`
- `color='red'` → titik berwarna merah
- `marker='p'` → bentuk titiknya persegi lima (pentagon)

#### 3. Label dan judul

- `plt.title('Scatter Plot Korelasi Positif')` → memberi judul.
- `plt.xlabel('Nilai1'), plt.ylabel('Nilai2')` → memberi label sumbu.

#### 4. Grid

```
plt.grid(True)
```

- menambahkan garis bantu supaya lebih mudah membaca posisi titik.

#### 5. Tampilkan plot

```
plt.show()
```

```

import pandas as pd
import matplotlib.pyplot as plt

# Buat DataFrame contoh
data = {
    'Nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Nilai2': [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
}

df3 = pd.DataFrame(data)

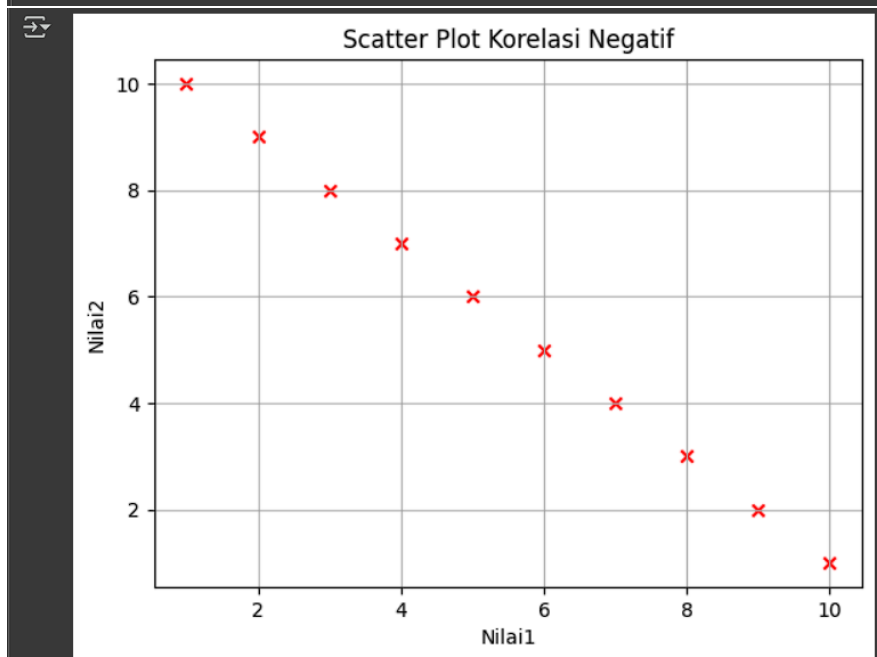
# Buat scatter plot
plt.scatter(df3['Nilai1'], df3['Nilai2'], color='red', marker='x')

# Tambahkan label
plt.title('Scatter Plot Korelasi Negatif')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()

```



### Penjelasan perbagian

#### 1. Membuat DataFrame

- Nilai1 naik dari 1 → 10
- Nilai2 turun dari 10 → 1 Jadi keduanya punya hubungan berbanding terbalik sempurna.

#### 2. Scatter Plot

`plt.scatter(df3['Nilai1'], df3['Nilai2'], color='red', marker='x')`

- x = Nilai1



- $y = \text{Nilai2}$
- Titik berwarna merah (`color='red'`)
- Marker berbentuk tanda silang (`marker='x'`)

### 3. **Judul & Label**

- `plt.title('Scatter Plot Korelasi Negatif')` : membuat Judul *Scatter Plot Korelasi Negatif*
- `plt.xlabel('Nilai1')` : Memberi label pada sumbu-x (horizontal) dengan nama "Nilai1"
- `plt.ylabel('Nilai2')` : Memberi label pada sumbu-y (vertikal) dengan nama "Nilai2".

### 4. **Grid**

`plt.grid(True)` :

Membantu melihat posisi titik dengan lebih jelas.

### 5. **Show**

`plt.show()` :

perintah untuk menampilkan grafik/plot yang sudah dibuat dengan matplotlib.