

Nama : Sulthan Nabil Al Hakim

TI : 02

NIM : 0110224234

## 1. Menghubungkan Google Colab dengan Google Drive

```
[14] ✓ 1d
# 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, call drive.mount("/content/gdrive", force\_remount=True).

Penjelasan :

google.colab.drive digunakan untuk mengakses file di Google Drive.

## 2. Menentukan Path Dataset

```
[15] ✓
#memanggil data set lewat drive
path = "/content/gdrive/MyDrive/day.csv/day.csv1"
```

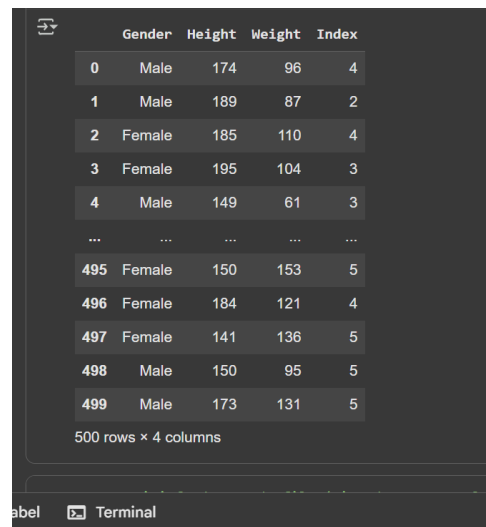
Penjelasan :

- Ini adalah **lokasi folder** tempat file dataset kamu berada di dalam Google Drive.
- dataset berada di /content/gdrive/MyDrive/day.csv, jadi pastikan path benar agar file bisa dibaca.

### 3. Membaca File CSV

```
[10] # membaca file csv menggunakan pandas
✓ 0d import pandas as pd

df=pd.read_csv(path+ '/data/day.csv')
df
```



	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
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

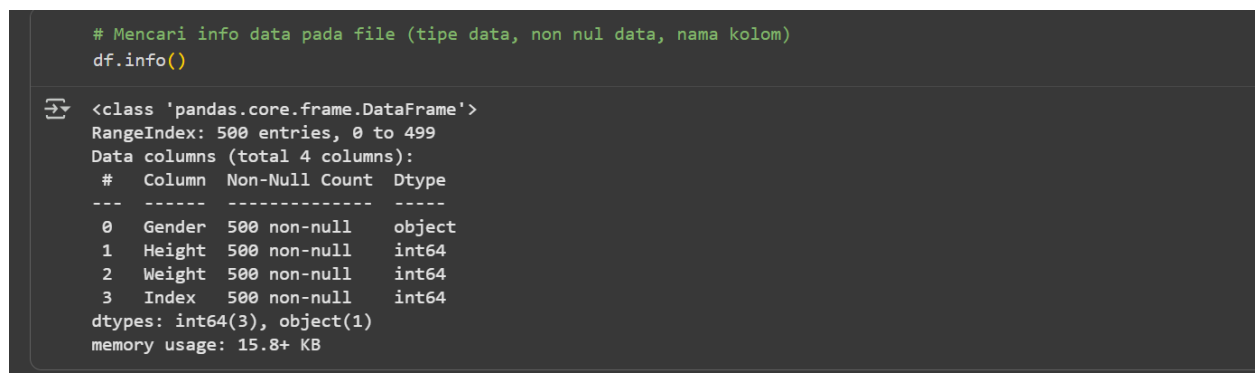
500 rows x 4 columns

Penjelasan :

- pandas digunakan untuk manipulasi data berbentuk tabel.
- `pd.read_csv()` membaca file .csv menjadi DataFrame (df).
- Menuliskan df di akhir sel akan menampilkan isi tabel (seperti Excel).

### 4. Melihat Informasi Dataset

```
# Mencari info data pada file (tipe data, non nul data, nama kolom)
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
```

Penjelasan :

Menampilkan:

- Jumlah baris dan kolom
- Nama kolom
- Jumlah data tidak kosong (non-null)
- Tipe data tiap kolom

## 5. Menghitung Statistik Dasar Kolom “Height”

```
1 #menghitung mean semua kolom numerik
  df['Height'].mean()
np.float64(169.944)

#menghitung median semua kolom numerik
df['Height'].median()
170.5

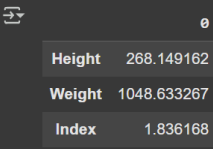
#menghitung modus (hati" karena bisa lebih dari satu)
df['Height'].mode()
Height
0    188
```

Penjelasan :

- **Mean** = nilai rata-rata tinggi badan.
- **Median** = nilai tengah setelah data diurutkan.
- **Mode** = nilai yang paling sering muncul.

## 6. Variansi dan Standar Deviasi

```
# menghitung variansi & standar deviasi
df.var(numeric_only=True)
```



	0
Height	268.149162
Weight	1048.633267
Index	1.836168

dtype: float64

Penjelasan :

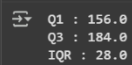
- **Variansi** mengukur seberapa besar sebaran data.
- **Standar deviasi** adalah akar dari variansi → makin besar nilainya, makin menyebar datanya.

## 7. Kuartil dan IQR

```
[ ] # Hitung kuartil pertama (Q1)
q1 = df['Height'].quantile(0.25)
print('Q1 :', q1)

# Hitung kuartil ketiga (Q3)
q3 = df['Height'].quantile(0.75)
print('Q3 :', q3)

# Hitung IQR (interquartile range)
iqr = q3 - q1
print('IQR :', iqr)
```



Q1 : 156.0  
Q3 : 184.0  
IQR : 28.0

Penjelasan :

- Q1 = kuartil bawah (25% data terendah)
- Q3 = kuartil atas (25% data tertinggi)
- IQR = Q3 - Q1 → digunakan untuk mendeteksi **outlier**

## 8. Statistik Deskriptif Lengkap

```
[ ] # 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

Penjelasan :

Count = jumlah data

Mean = rata-rata

Std = standar deviasi

Min = nilai terkecil

25%, 50%, 75% = kuartil

Max = nilai terbesar

## 9. Matriks Kolerasi

```
# Menghitung matriks kolerasi untuk semua kolom numerik
correlation_matrix = df.corr(numeric_only = True)

#Menampilkan matriks kolerasi
print('Matriks Korelasi:')
print(correlation_matrix)
```

↔ Matriks Korelasi:

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

Penjelasan :

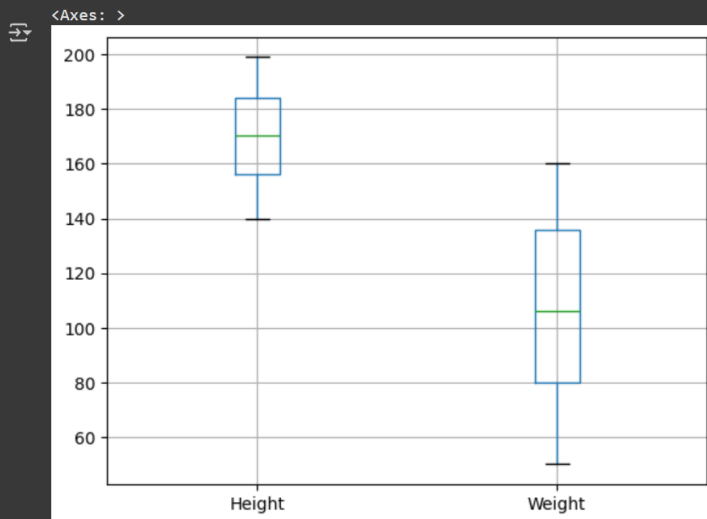
Menunjukkan **hubungan antar kolom numerik**.

- Nilai mendekati **1** → korelasi positif kuat (semakin tinggi Height, semakin tinggi Weight)
- Nilai mendekati **-1** → korelasi negatif kuat

## 10. Boxplot (Visualisasi Outlier)

```
import pandas as pd
import numpy as np

df.boxplot(column=['Height', 'Weight'])
```



Penjelasan :

Menampilkan distribusi data dan **outlier** dari kolom Height & Weight.

## 11. Histogram

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

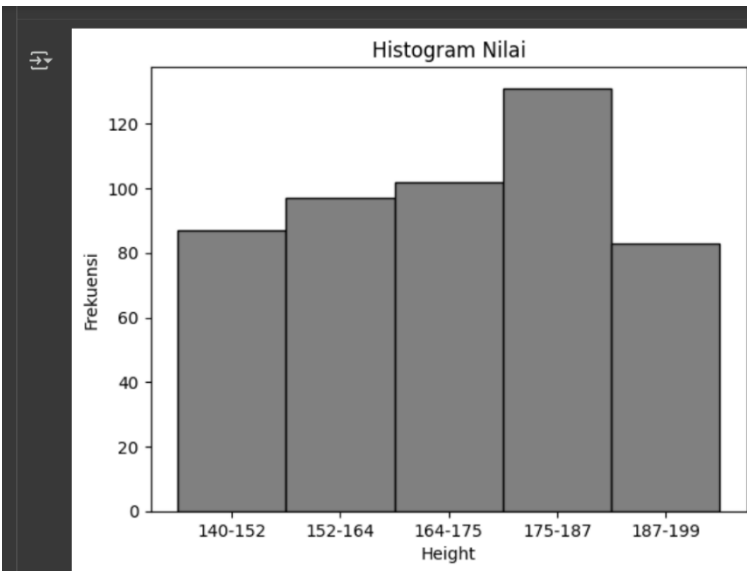
# Ambil data Height
data_height = df["Height"]

# Buat histogram
n, bins, patches = plt.hist(data_height, bins=5, color='grey', edgecolor='black')

# Tambahkan label
plt.title('Histogram Nilai')
plt.xlabel('Height')
plt.ylabel('Frekuensi')

# Tampilkan rentang frekuensi di 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)])

# Tampilkan histogram
plt.show()
```



Penjelasan :

Menunjukkan distribusi nilai tinggi badan (Height) ke dalam 5 kelompok (bins).



## 12. Scatter Plot Korelasi Positif

```
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]
}

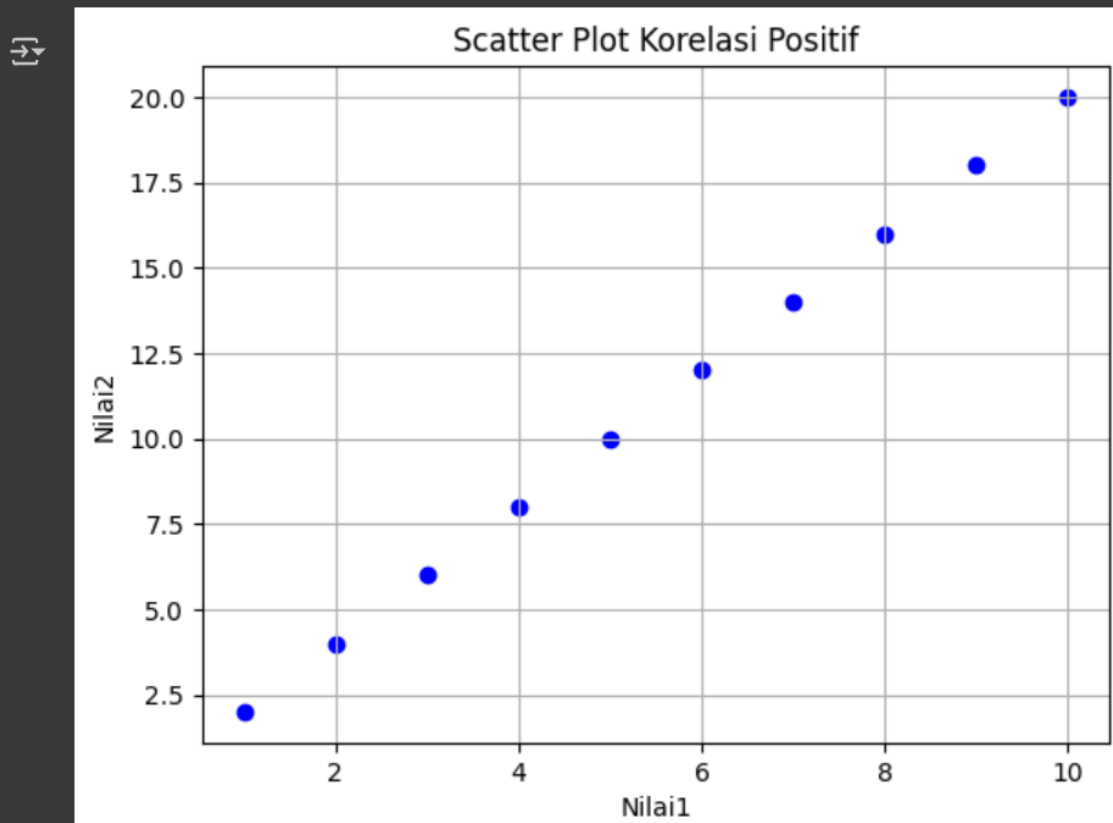
df2 = pd.DataFrame(data)

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

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

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()
```



Penjelasan :

Menunjukkan hubungan positif — ketika Nilai1 naik, Nilai2 juga naik.

### 13. Scatter Plot Korelasi Negatif

```
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]
}

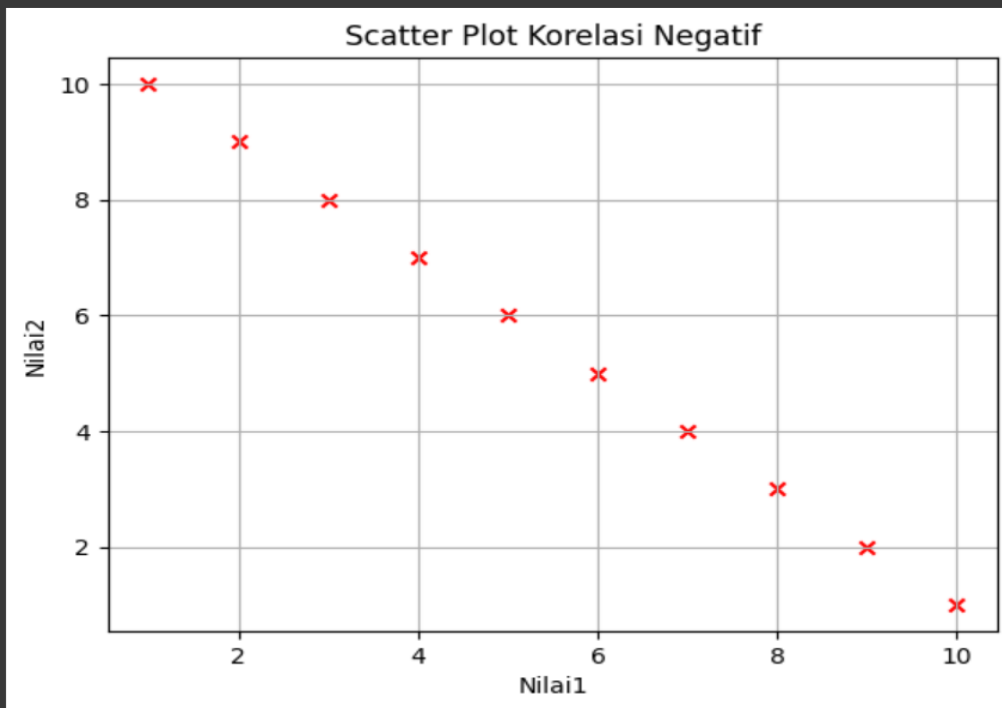
df3 = pd.DataFrame(data)

# 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 :

Menunjukkan hubungan negatif — ketika Nilai1 naik, Nilai2 turun.

#### 14. Pembagian Dataset

Penjelasan :

Membaca data sepeda harian (`df = pd.read_csv(path+ '/data/day.csv')`)

Melihat `df.info()`

Menghitung **mean, median, mode** untuk kolom `casual`

Menghitung **variansi, standar deviasi, Q1, Q3, IQR**

Menampilkan **statistik deskriptif**

Membuat **matriks korelasi**

Visualisasi:

- **Boxplot** kolom `casual` & `mnth`
- **Histogram** data `casual`
- **Scatter plot positif dan negatif** (contoh korelasi antar nilai)