

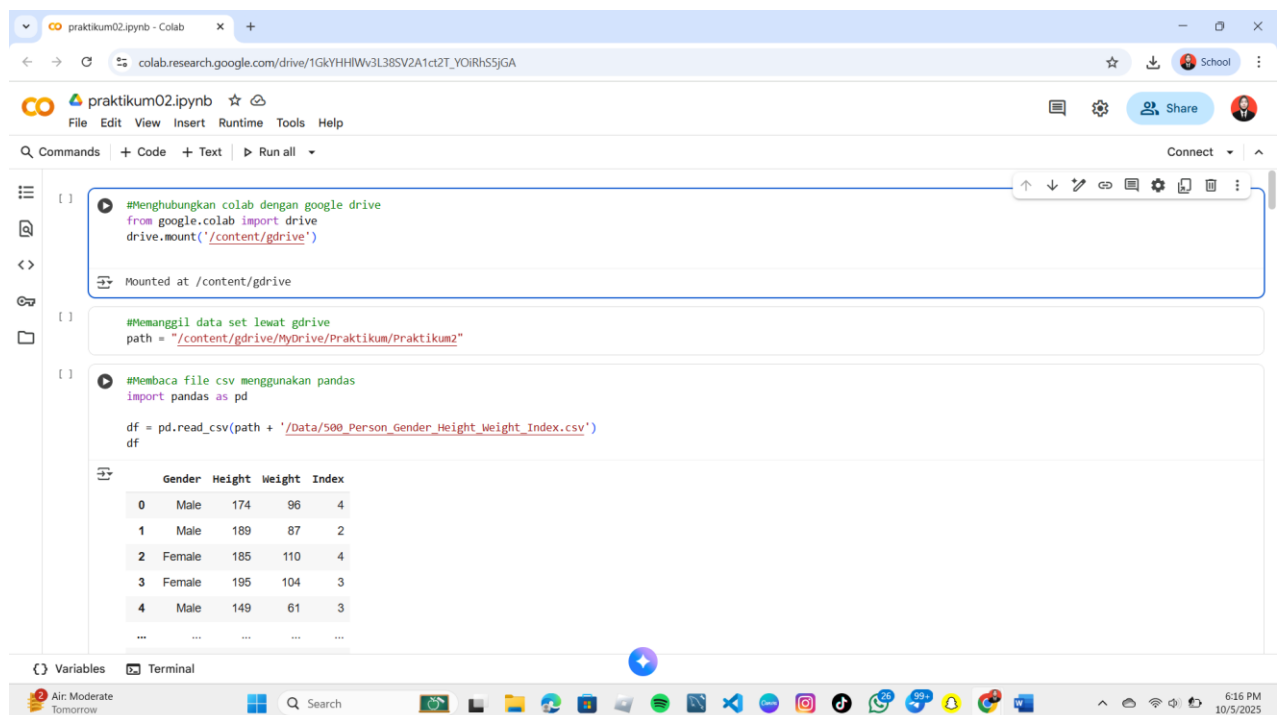
# Tugas 2: Praktikum 2 dan Latihan 2 – Machine Learning

Yurida Yahsyia 1 - 0110224100 <sup>1</sup>\*

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

\*E-mail: [0110224100@student.nurulfikri.ac.id](mailto:0110224100@student.nurulfikri.ac.id) – email mahasiswa 1

## 1. Praktikum mandiri 2



The screenshot shows a Google Colab notebook with the following code and output:

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

#Memanggil data set lewat gdrive
path = "/content/gdrive/MyDrive/Praktikum/Praktikum2"

#Membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + '/Data/500_Person_Gender_Height_Weight_Index.csv')
df
```

The output of the last cell is a table with 5 columns: Gender, Height, Weight, Index, and an unlabeled column. The first 5 rows are shown:

	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	

### 1. from google.colab import drive

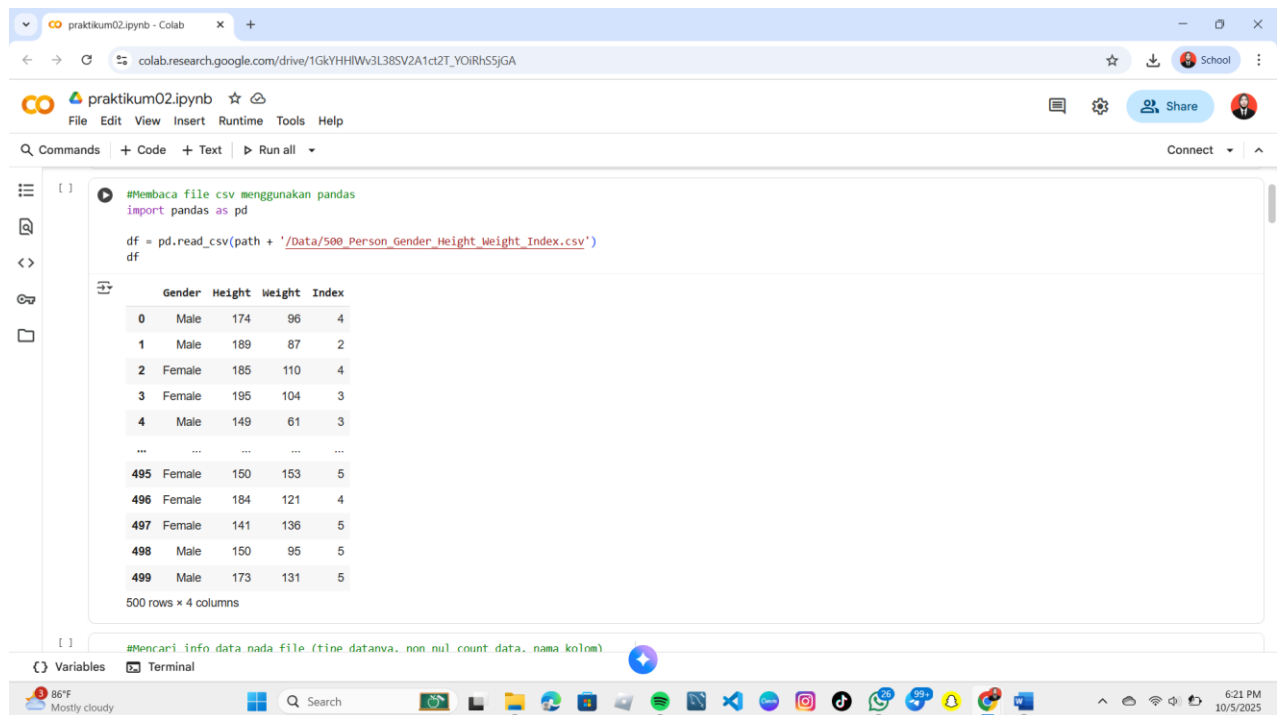
Memanggil modul bawaan Colab.

### 2. drive.mount('/content/gdrive')

Membuat “jembatan” supaya Colab bisa membaca file yang ada di Drive, dan semua file bisa diakses lewat folder /content/gdrive/MyDrive/.

### 3. path = "/content/gdrive/MyDrive/Praktikum/Praktikum2"

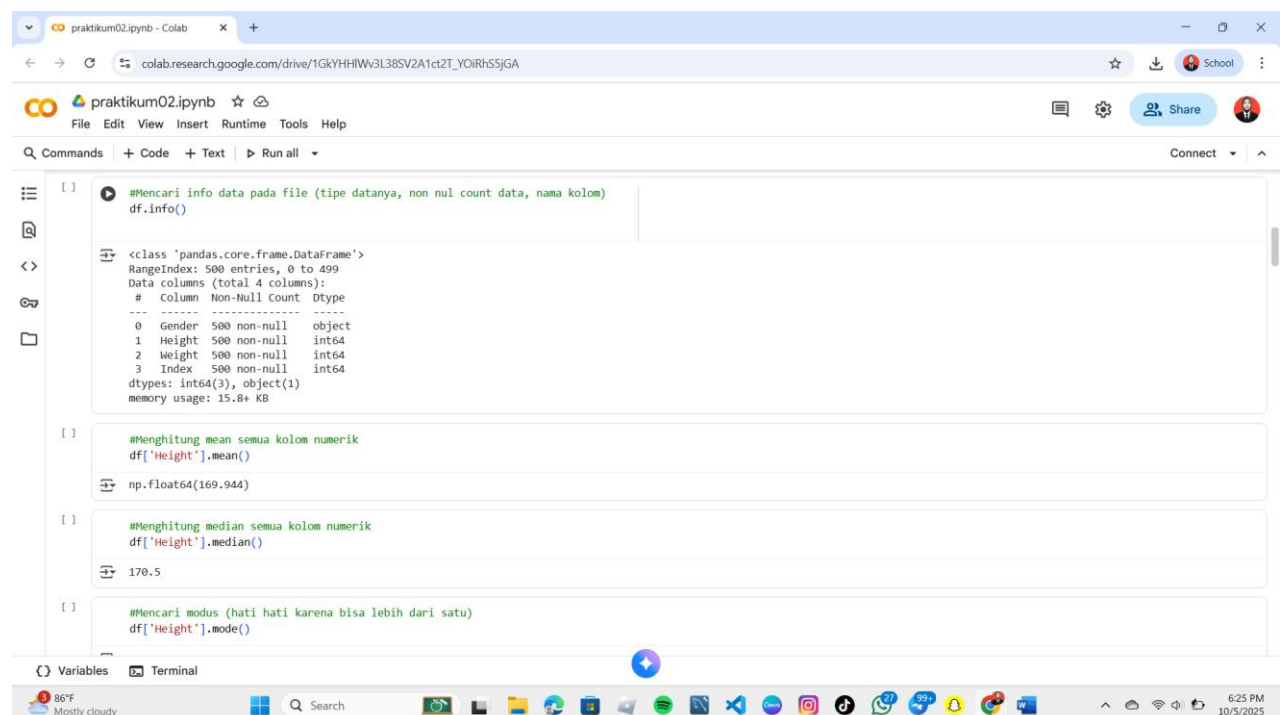
path = alamat menuju folder Praktikum2 di dalam Google Drive.



1. **import pandas as pd**  
memanggil library Pandas.

2. **df=pd.read\_csv(path + '/Data/500\_Person\_Gender\_Height\_Weight\_Index.csv')**

**Membaca file CSV (Comma Separated Values)**, df = variabel yang menyimpan dataset dalam bentuk tabel (DataFrame), df di baris terakhir = supaya Colab menampilkan isi dataset (beberapa baris pertama).



## 1. df.info()

Menampilkan ringkasan tentang dataset: berguna untuk cek struktur data sebelum dianalisis.

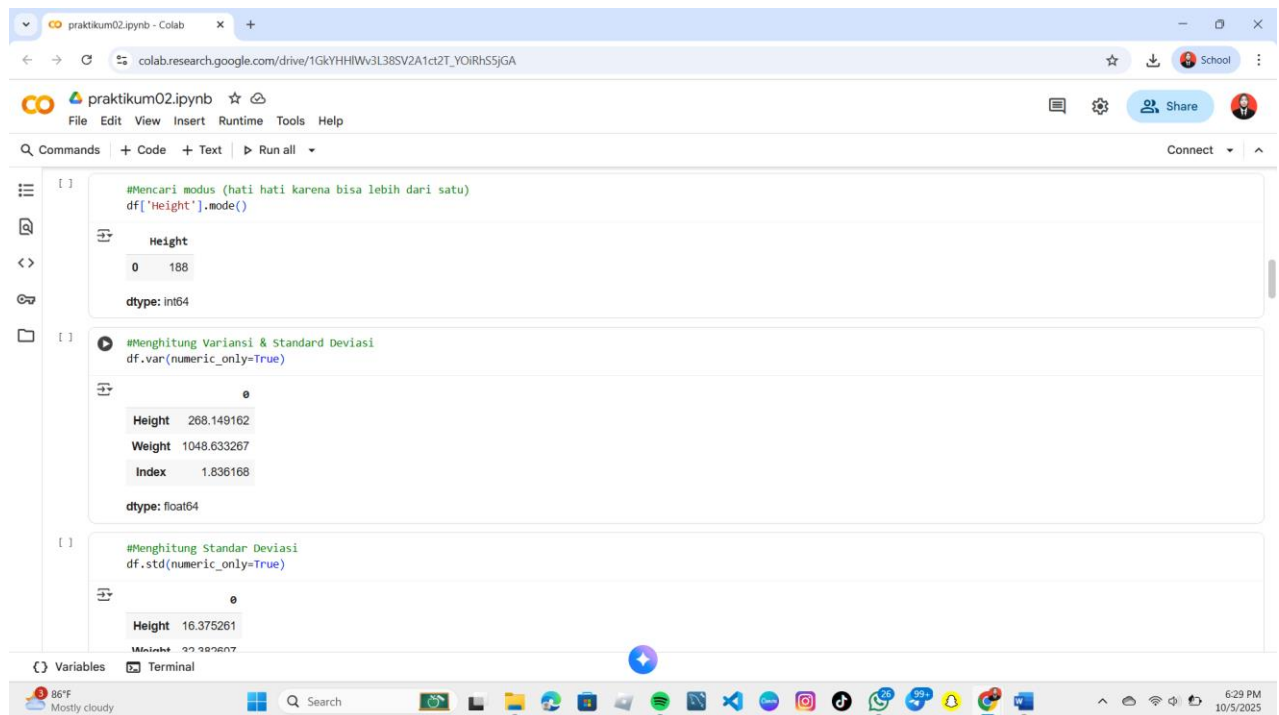
- Jumlah baris (500 entries)
- Nama kolom (Gender, Height, Weight, Index)
- Jumlah data yang **tidak kosong** (non-null) → semuanya 500, artinya **tidak ada missing value**

## 2. df['Height'].mean()

Mengambil kolom Height (tinggi badan) lalu dihitung rata-ratanya.

## 3. df['Height'].median()

Nilai tengah setelah semua data diurutkan.



```
[ ]  
#Mencari modus (hati hati karena bisa lebih dari satu)  
df['Height'].mode()  
  
Height  
0    188  
  
dtype: int64  
  
[ ]  
#Menghitung Variansi & Standard Deviasi  
df.var(numeric_only=True)  
  
Height    268.149162  
Weight    1048.633267  
Index      1.836168  
  
dtype: float64  
  
[ ]  
#Menghitung Standar Deviasi  
df.std(numeric_only=True)  
  
Height    16.375261  
Weight     32.382607  
  
dtype: float64
```

## 1. df['Height'].mode()

Nilai yang paling sering muncul.

## 2. df.var(numeric\_only=True)

Variansi = ukuran seberapa jauh data menyebar dari rata-rata. Misalnya variansi tinggi (Height) = 268 : menunjukkan bahwa tinggi badan tidak semua sama, ada penyebaran/keragaman. Semakin besar variansi = data makin tersebar.

```
[ ]  
#Menghitung Standar Deviasi  
df.std(numeric_only=True)  
  
0  
Height    16.375261  
Weight    32.382607  
Index      1.355053  
  
dtype: float64  
  
[ ]  
#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
```

### 1. `df.std(numeric_only=True)`

Dipakai untuk **mengukur seberapa menyebar data dari rata-rata**. Kalau standar deviasi kecil = data rapat dekat rata-rata. Kalau besar = data lebih beragam.

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

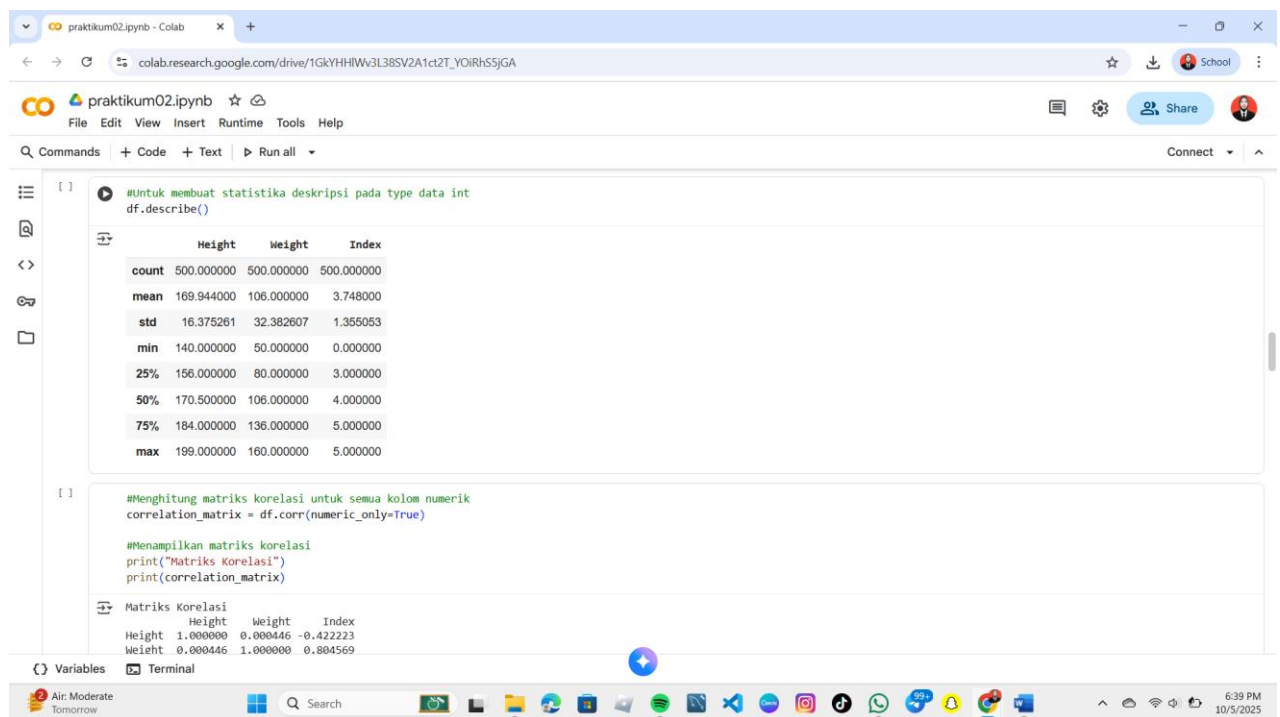
Q1 adalah nilai di bawah 25% data berada. Artinya, 25% orang dalam dataset punya tinggi  $\leq 156$  cm.

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

Q3 adalah nilai di bawah 75% data berada. Artinya, 75% orang dalam dataset punya tinggi  $\leq 184$  cm.

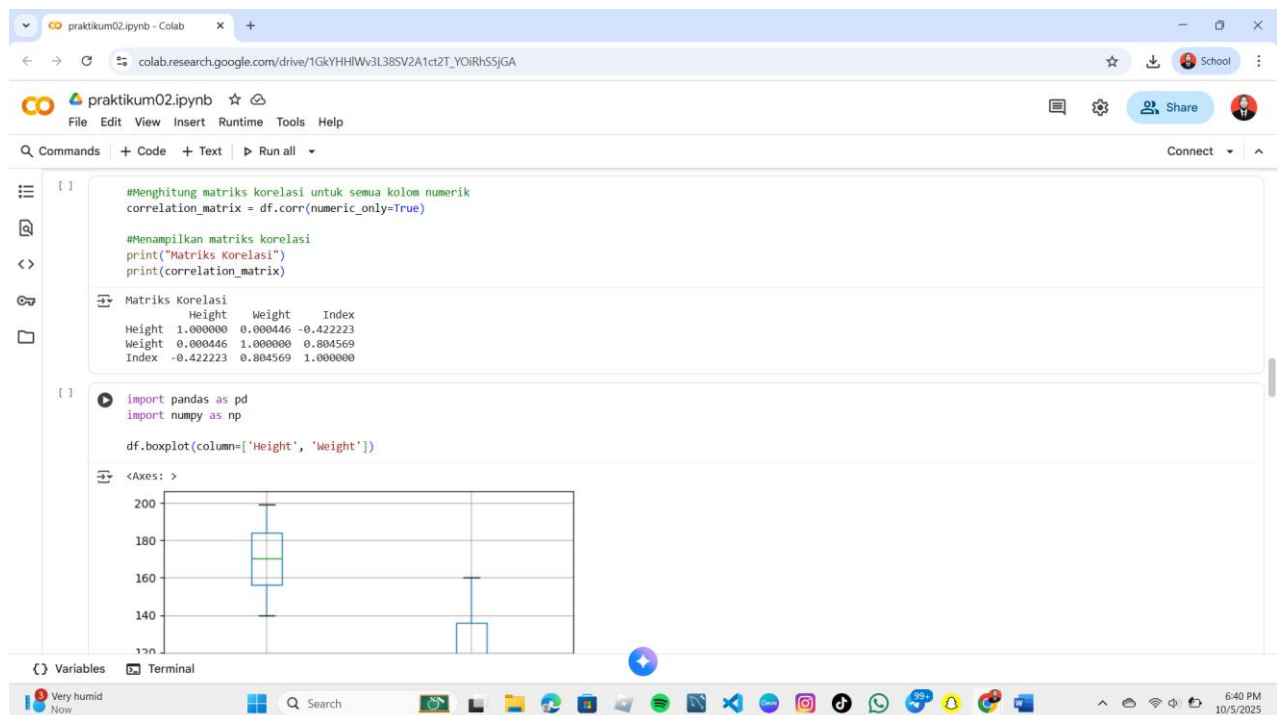
### 4. `iqr = q3 - q1` `print("IQR : ", iqr)`

$IQR = Q3 - Q1$  = rentang tengah dari 50% data. Hasilnya = 28 cm berarti 50% orang punya tinggi antara 156 cm dan 184 cm.



## 1. df.describe()

Otomatis menghitung ringkasan statistik untuk semua kolom numerik.



## 1. df.corr(numeric\_only=True)

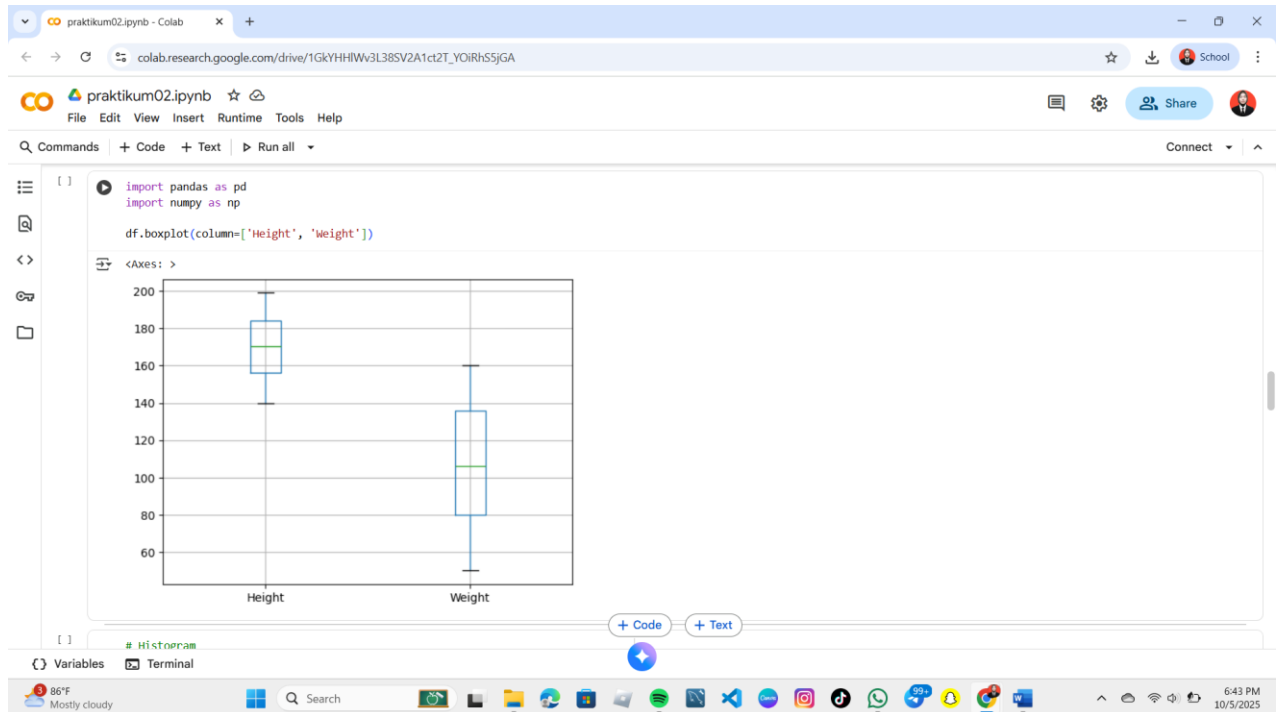
menghitung korelasi antar kolom yang berisi angka (numerik).

## 2. correlation\_matrix

menyimpan hasil korelasi.

## 3. print(correlation\_matrix)

menampilkan tabel korelasi.



### 1. import pandas as pd

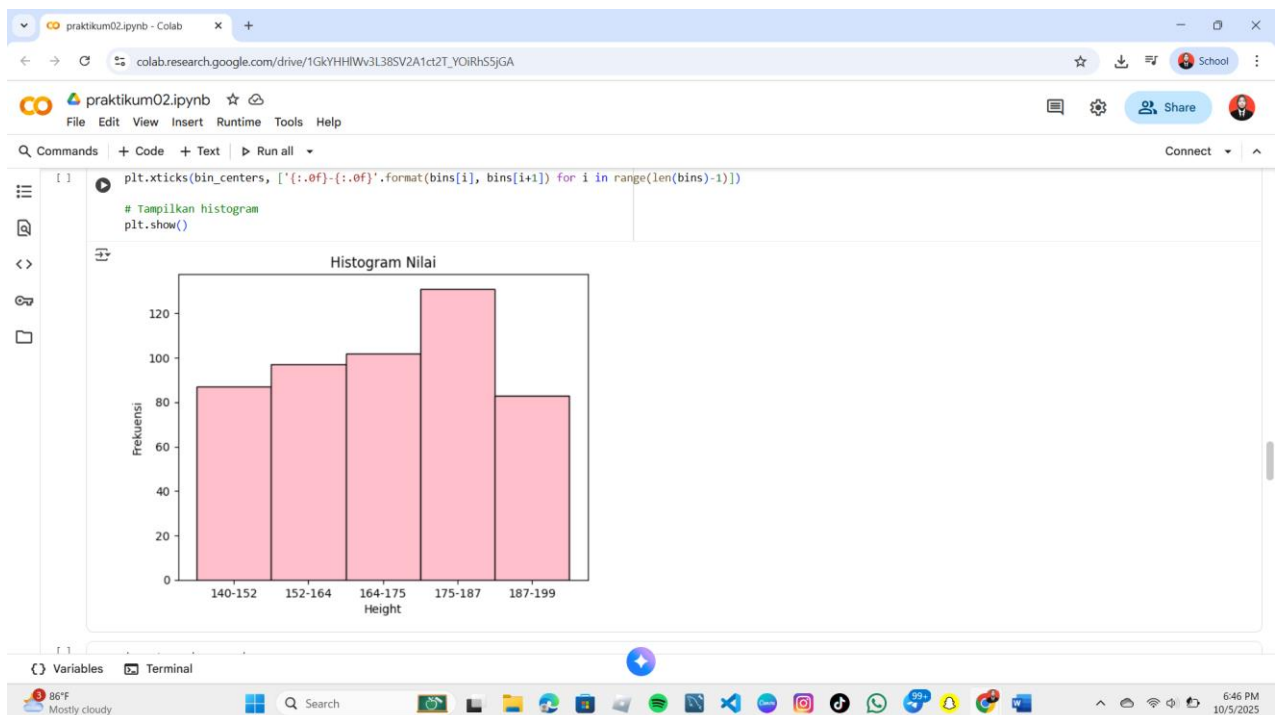
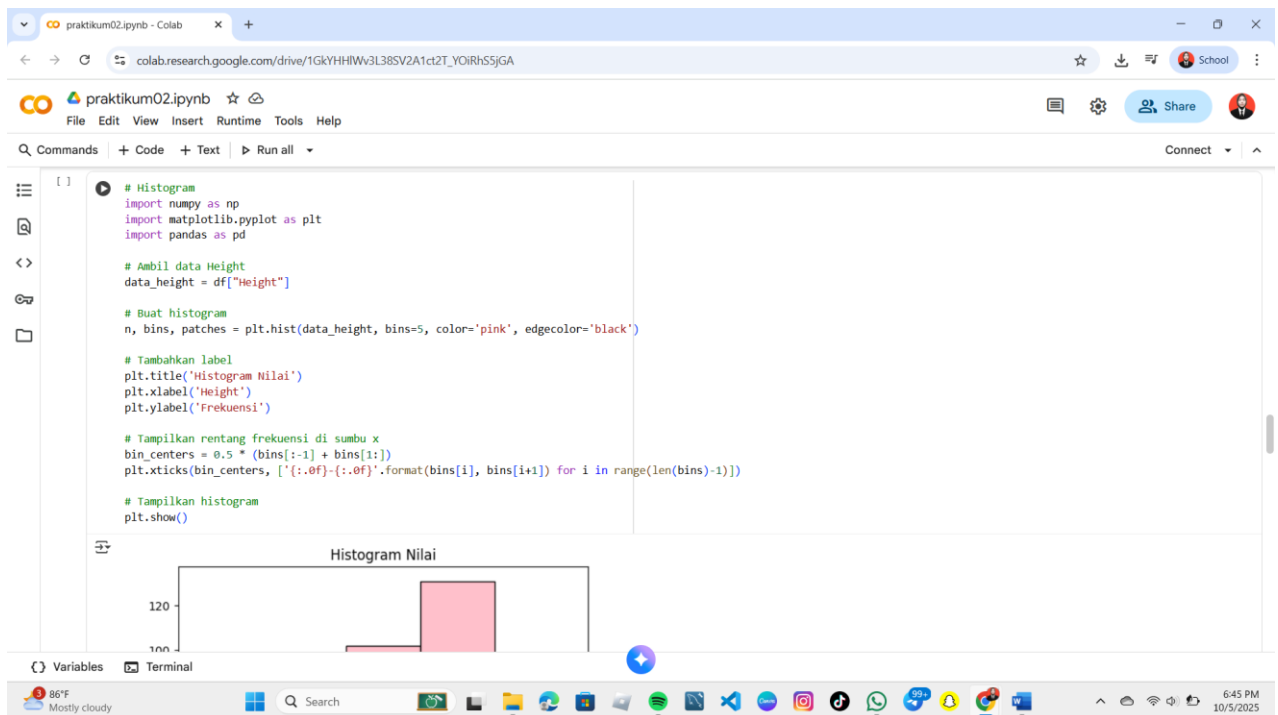
Mengimpor pandas dengan alias pd.

### 2. import numpy as np

Mengimpor numpy dengan alias np.

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

Membuat boxplot dari DataFrame df. Hanya kolom Height (tinggi) dan Weight (berat) yang divisualisasikan.



### 1. import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

Mengimpor library yang diperlukan. numpy = untuk perhitungan matematis. matplotlib.pyplot = untuk membuat grafik visualisasi. pandas = untuk mengolah data tabular (seperti DataFrame).

### 2. data\_height = df["Height"]

Mengambil kolom Height dari DataFrame df.

### 3. n, bins, patches = plt.hist(data\_height, bins=5, color='pink', edgecolor='black')

Membuat **histogram** dari data tinggi badan.

- bins=5 : data dibagi menjadi 5 kelompok/rentang (interval).
- color='pink' : warna batang histogram pink.
- edgecolor='black' : garis tepi batang berwarna hitam.
- n = jumlah data di tiap batang, bins = batas tiap interval, patches = objek batang histogram.

#### 4. plt.title('Histogram Nilai')

plt.xlabel('Height')

plt.ylabel('Frekuensi')

Menambahkan judul dan label sumbu:

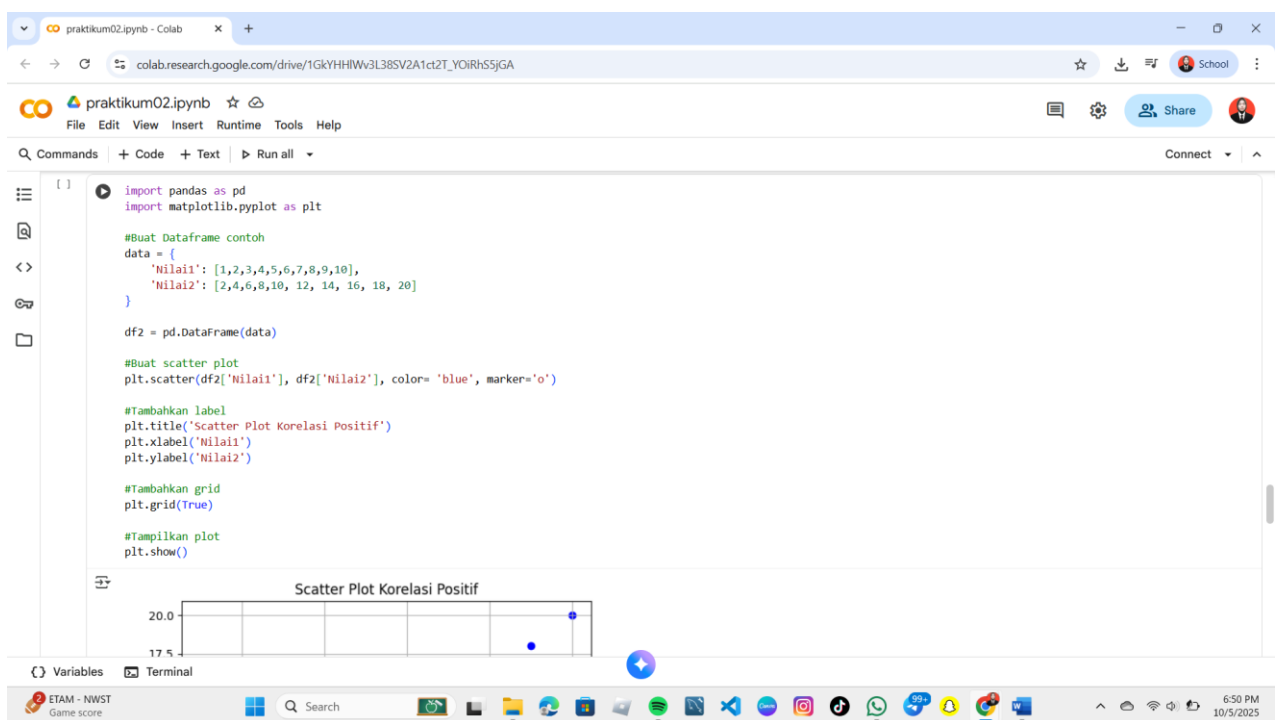
#### 5. bin\_centers = 0.5 \* (bins[:-1] + bins[1:])

plt.xticks(bin\_centers, ['{:0f}-{:0f}'.format(bins[i], bins[i+1]) for i in range(len(bins)-1)])

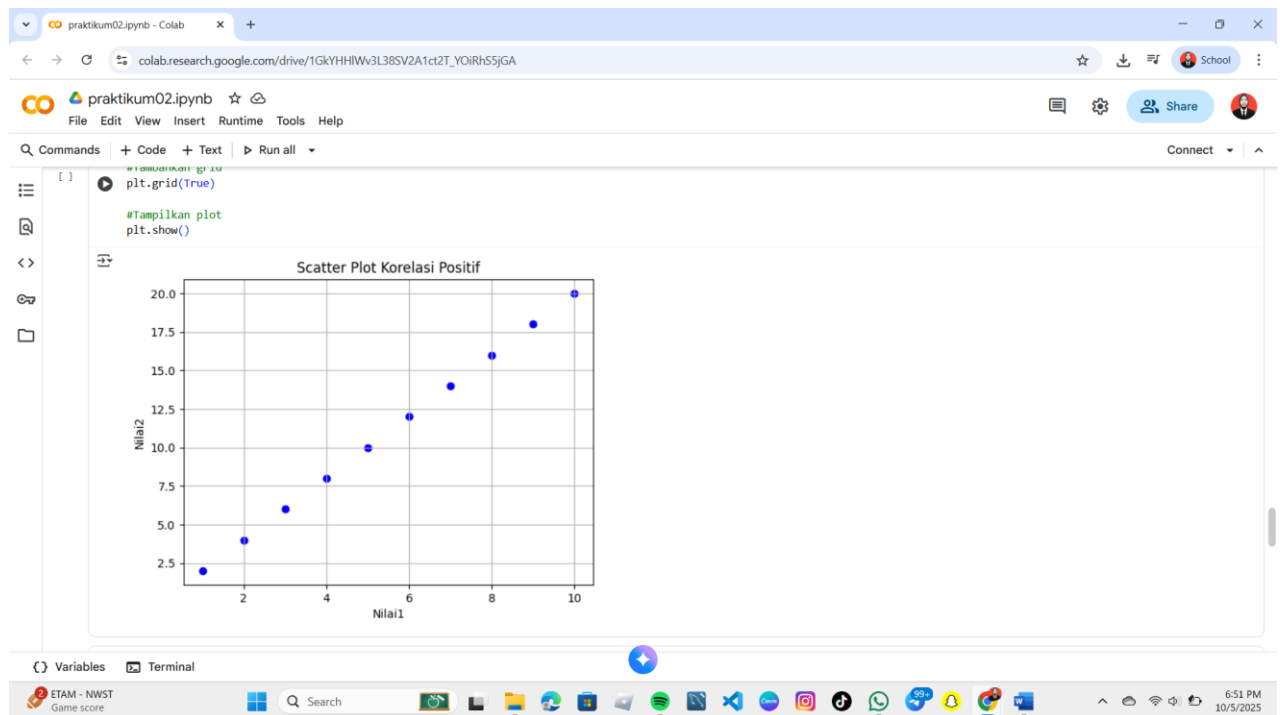
- bins[:-1] = ambil semua batas bawah, bins[1:] = ambil semua batas atas.
- 0.5 \* (bins[:-1] + bins[1:]) : menghitung posisi tengah tiap batang (supaya labelnya rapi).
- plt.xticks() : mengganti angka default di sumbu X dengan label rentang.

#### 6. plt.show()

Menampilkan histogram di layar.







**1. import pandas as pd**

**import matplotlib.pyplot as plt**

Mengimpor library

**2. data = {**

**'Nilai1': [1,2,3,4,5,6,7,8,9,10],**

**'Nilai2': [2,4,6,8,10, 12, 14, 16, 18, 20]**

**}**

Membuat data dalam bentuk dictionary Python.

**3. df2 = pd.DataFrame(data)**

Mengubah dictionary menjadi DataFrame pandas bernama df2

**4. plt.scatter(df2['Nilai1'], df2['Nilai2'], color='blue', marker='o')**

Membuat scatter plot (diagram sebar).

**5. plt.title('Scatter Plot Korelasi Positif')**

**plt.xlabel('Nilai1')**

**plt.ylabel('Nilai2')**

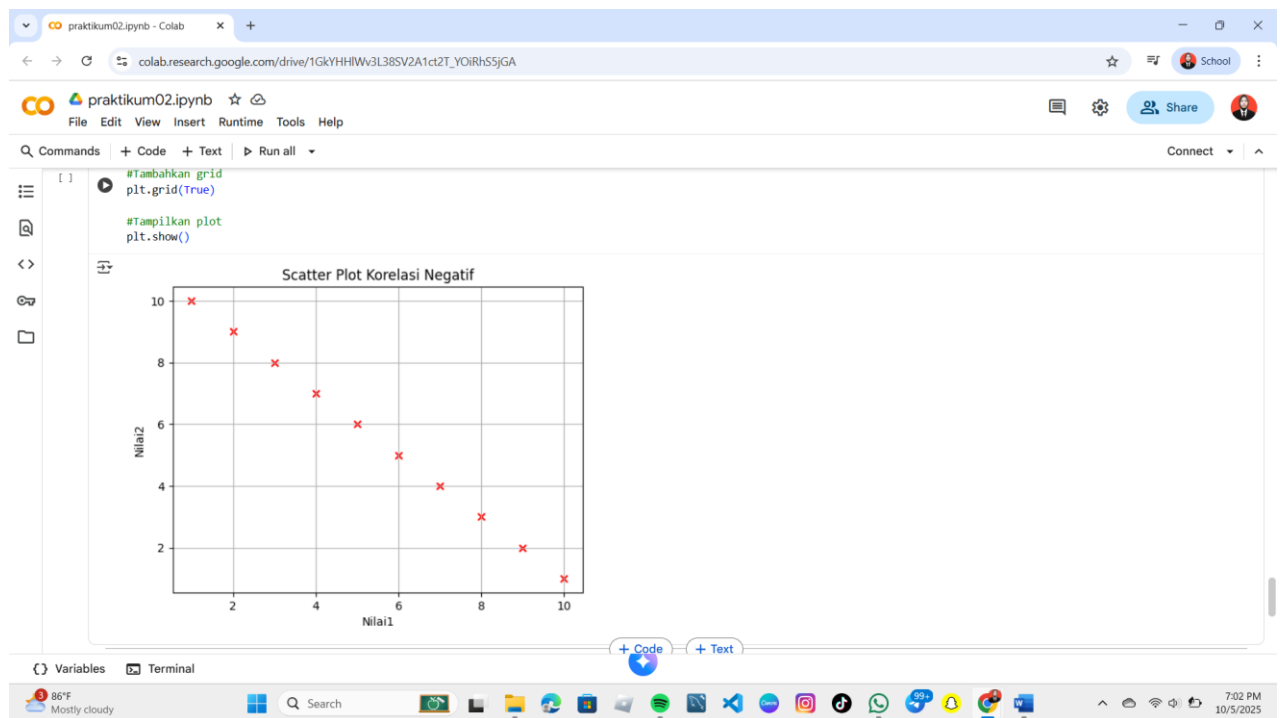
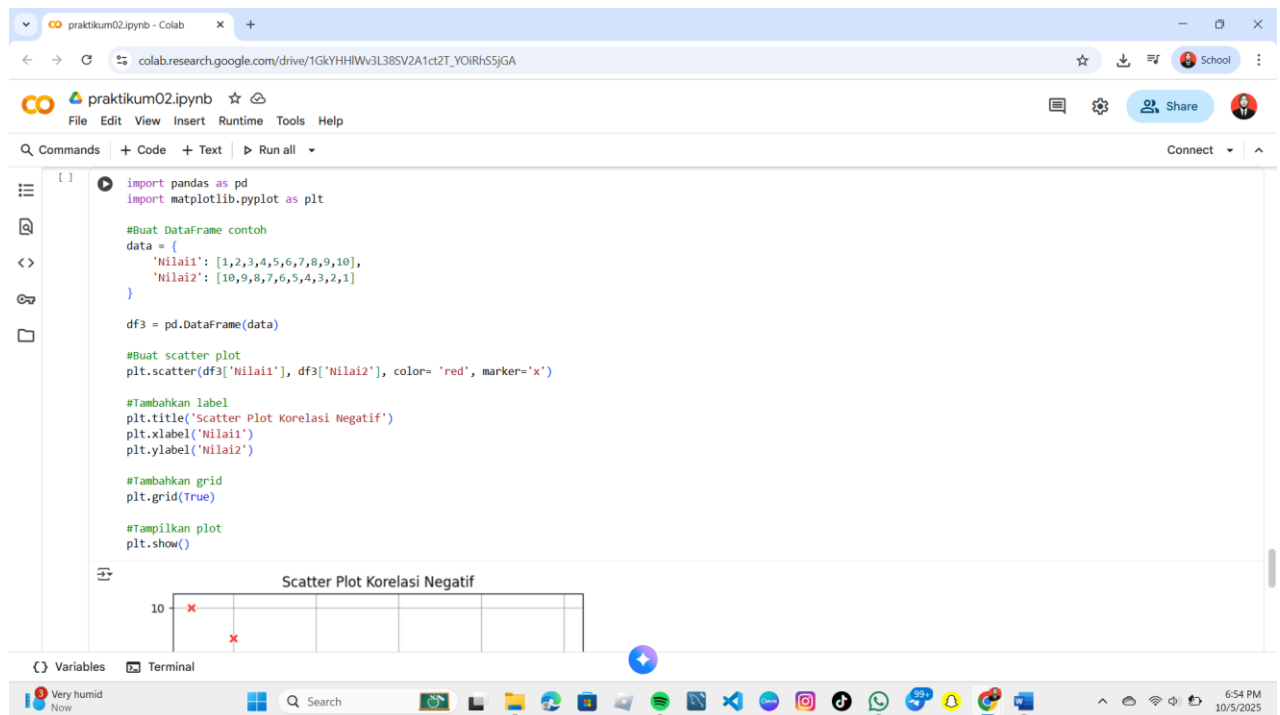
Memberi keterangan pada grafik

**6. plt.grid(True)**

Menampilkan **garis bantu (grid)** di grafik supaya lebih mudah membaca posisi titik.

**7. plt.show()**

Menampilkan scatter plot ke layar.



1. **import pandas as pd**  
**import matplotlib.pyplot as plt**  
Mengimpor library

2. **data = {**  
    **'Nilai1': [1,2,3,4,5,6,7,8,9,10],**  
    **'Nilai2': [10,9,8,7,6,5,4,3,2,1]**  
**}**

Membuat data kamus dengan dua kolom (Nilai1 naik, Nilai2 turun) yang menunjukkan korelasi

negatif.

### 3. `df3 = pd.DataFrame(data)`

Mengubah data kamus menjadi DataFrame (tabel data) Pandas.

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

Membuat Scatter Plot menggunakan Nilai1 sebagai sumbu X dan Nilai2 sebagai sumbu Y dengan marker silang merah.

### 5. `plt.title('Scatter Plot Korelasi Positif')`

`plt.xlabel('Nilai1')`

`plt.ylabel('Nilai2')`

Memberi keterangan pada grafik

### 6. `plt.grid(True)`

Menampilkan **garis bantu (grid)** di grafik supaya lebih mudah membaca posisi titik.

### 7. `plt.show()`

Menampilkan scatter plot ke layar.

## 2. Latihan Mandiri 2

The screenshot shows a Google Colab notebook interface. The code in the notebook is as follows:

```
[8] #Menghubungkan colab dengan google drive
from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

[2] #Memanggil data set lewat gdrive
path = "/content/gdrive/MyDrive/Praktikum/Praktikum2"

[6] # Lokasi dataset (di dalam folder Data)
import os
data_path = os.path.join(path, "Data", "day.csv")

[10] # Load dataset
import pandas as pd
df = pd.read_csv(data_path)
print("Jumlah total data:", len(df))
df.head()
```

The output of the code shows the dataset's head:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1									446	331	654 985
1	2	2011-01-02	1	0	1									139	131	670 801
2	3	2011-01-03	1	0	1									120	1229	1349

### 1. `from google.colab import drive`

Memanggil modul bawaan Colab.

### 2. `drive.mount('/content/gdrive')`

Membuat “jembatan” supaya Colab bisa membaca file yang ada di Drive, dan semua file bisa diakses lewat folder `/content/gdrive/MyDrive/`.

**3. path = "/content/gdrive/MyDrive/Praktikum/Praktikum2"**

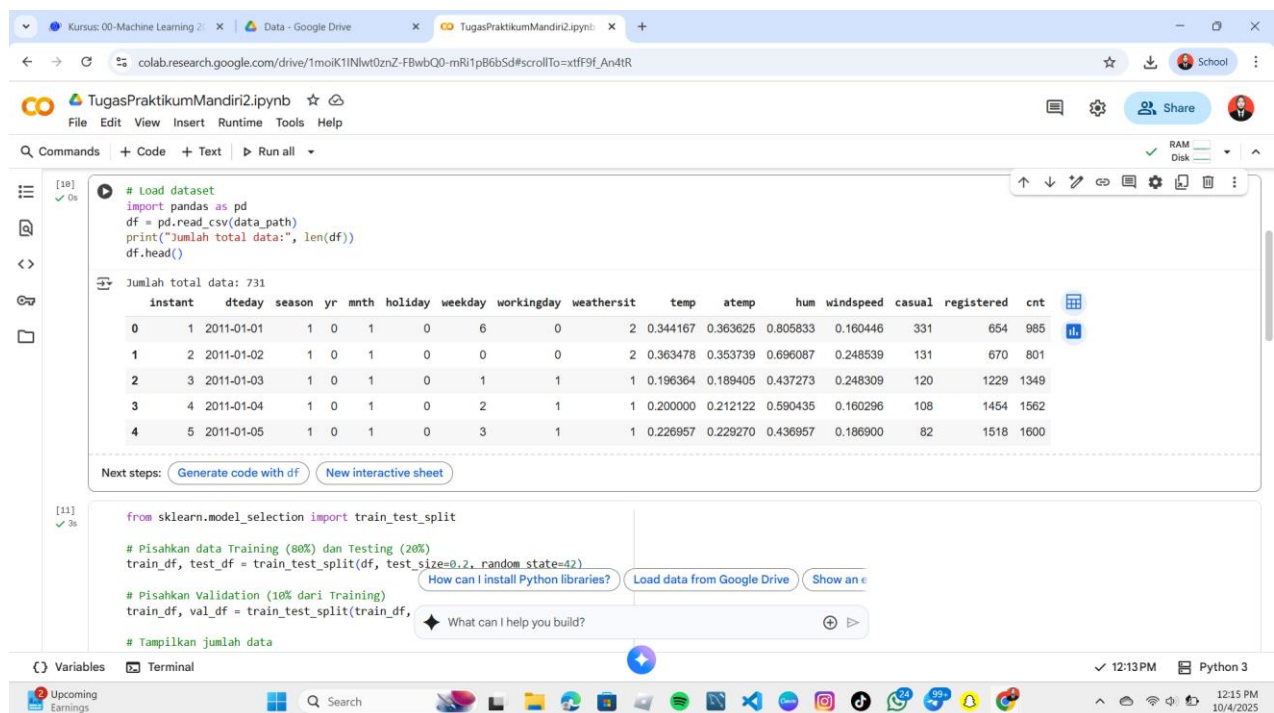
path = alamat menuju folder Praktikum2 di dalam Google Drive.

**4. import os**

Mengimpor pustaka **os** (Operating System), yang berguna untuk berinteraksi dengan sistem berkas.

**5. data\_path = os.path.join(path, "Data", "day.csv")**

Membuat variabel **data\_path** yang berisi jalur lengkap ke file day.csv. Fungsi **os.path.join()** digunakan untuk menggabungkan path secara aman, memastikan format jalur benar terlepas dari sistem operasinya.



**1. import pandas as pd**

Mengimpor pustaka Pandas Kembali

**2. df = pd.read\_csv(data\_path)**

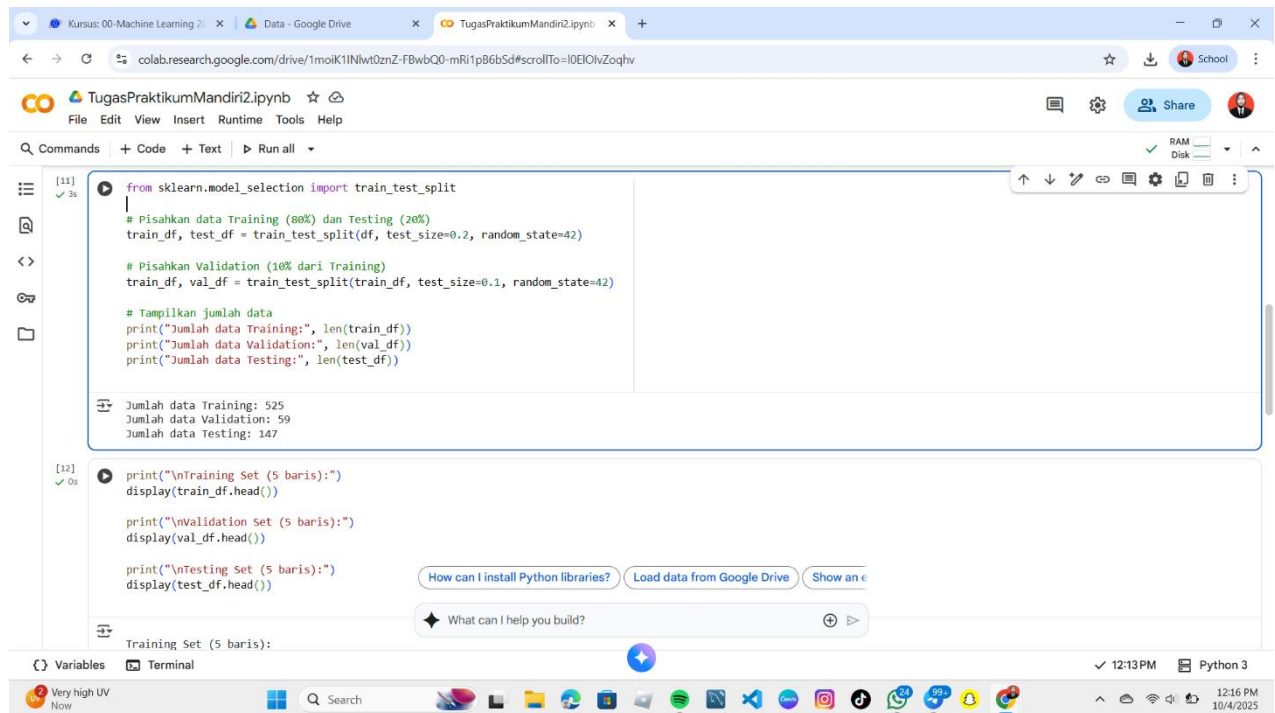
Fungsi inti untuk memuat data. Baris ini membaca file CSV yang ditunjuk oleh **data\_path** dan menyimpannya sebagai objek **DataFrame** Pandas bernama **df**.

**3. print("Jumlah total data:", len(df))**

Mencetak jumlah total baris atau entri dalam **DataFrame df** menggunakan fungsi **len()**.

**4. df.head()**

Menampilkan **5 baris pertama** dari **DataFrame df**.

The screenshot shows a Google Colab notebook interface. The browser tabs at the top include 'Kursus: 00-Machine Learning 2', 'Data - Google Drive', and 'TugasPraktikumMandiri2.ipynb'. The notebook's URL is 'colab.research.google.com/drive/1moiK1INlw0znZ-FBwbQ0-mRi1pB6bSd#scrollTo=I0EIOlvZoqhv'. The notebook title is 'TugasPraktikumMandiri2.ipynb'. The code editor shows two cells. Cell [111] contains code to split a DataFrame 'df' into training, validation, and testing sets using 'train\_test\_split' from 'sklearn.model\_selection'. It prints the counts for each set. Cell [112] prints the first five rows of each set. The output for cell [111] shows: 'Jumlah data Training: 525', 'Jumlah data Validation: 59', and 'Jumlah data Testing: 147'. The output for cell [112] shows the first five rows of the training set. The bottom of the notebook shows a 'Variables' panel with 'Training Set (5 baris):' and a 'Terminal' panel. The system tray at the bottom shows the time as 12:13 PM and the date as 10/4/2025.

## 1. `from sklearn.model_selection import train_test_split`

Mengimpor fungsi `train_test_split` dari modul `model_selection` di pustaka **Scikit-learn**.

## 2. `train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)`

Membagi DataFrame awal (`df`) menjadi dua subset. Hasilnya disimpan di variabel `train_df` (untuk training dan validation) dan `test_df` (untuk testing).

## 3. `train_df, val_df = train_test_split(train_df, test_size=0.1, random_state=42)`

Baris ini mengambil `train_df` dari yg sebelumnya dan membaginya lagi menjadi dua: `train_df` (data training akhir) dan `val_df` (validation).

## 4. `print("Jumlah data Training:", len(train_df))`

`print("Jumlah data Validation:", len(val_df))`

`print("Jumlah data Testing:", len(test_df))`

Baris-baris ini menggunakan fungsi `len()` untuk menghitung dan mencetak jumlah baris (jumlah data) di masing-masing subset yang baru dibuat: **Training** (`train_df`), **Validation** (`val_df`), dan **Testing** (`test_df`).

The screenshot shows a Google Colab notebook titled 'TugasPraktikumMandiri2.ipynb'. The code cell contains the following Python code:

```
print("\nTraining Set (5 baris):")
display(train_df.head())

print("\nValidation Set (5 baris):")
display(val_df.head())

print("\nTesting Set (5 baris):")
display(test_df.head())
```

The output of the code is displayed below the code cell. It shows the first 5 rows of the training set, the first 3 rows of the validation set, and the first 1 row of the testing set. Each output is a table with 17 columns: instant, dteday, season, yr, mnth, holiday, weekday, workingday, weathersit, temp, atemp, hum, windspeed, casual, registered, and cnt.

Training Set (5 baris):

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt	
657	658	2012-10-19	4	1	10	0	5	1	2	0.563333	0.537896	0.815000	0.134954	753	4671	5424
163	164	2011-06-13	2	0	6	0	1	1	1	0.635000	0.601654	0.494583	0.305350	863	4157	5020
305	306	2011-11-02	4	0	11	0	3	1	1	0.377500	0.390133	0.718750	0.082092	370	3816	4186
111	112	2011-04-22	2	0	4	0	5	1	2	0.336667	0.321954	0.729583	0.219521	177	1506	1683
538	539	2012-06-22	3	1	6	0	5	1	1	0.777500	0.724121	0.573750	0.182842	964	4859	5823

Validation Set (5 baris):

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt	
325	326	2011-11-22	4	0	11	0	2	1	1	0.348333	0.354670	0.531250	0.181600	69	1538	1607
410	411	2012-02-15	1	1	2	0	2	1	1	0.348333	0.354670	0.531250	0.181600	141	4028	4169
92	93	2011-04-03	2	0	4	0	2	1	1	0.348333	0.354670	0.531250	0.181600	12213	1651	1598

Testing Set (5 baris):

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt	
12213	12214	2012-10-19	4	1	10	0	5	1	2	0.563333	0.537896	0.815000	0.134954	753	4671	5424

1. `print("\nTraining Set (5 baris):")`  
`display(train_df.head())`

`print("\nValidation Set (5 baris):")`  
`display(val_df.head())`

`print("\nTesting Set (5 baris):")`  
`display(test_df.head())`

Mencetak *header* untuk setiap *subset* data.

- `display(train_df.head())`: Menggunakan fungsi **head()** untuk menampilkan **5 baris pertama** dari DataFrame **train\_df** (Training Set).
- `display(val_df.head())`: Menampilkan **5 baris pertama** dari DataFrame **val\_df** (Validation Set).
- `display(test_df.head())`: Menampilkan **5 baris pertama** dari DataFrame **test\_df** (Testing Set).

Validation Set (5 baris):

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
325	326	2011-11-22	4	0	11	0	2	1	3	0.416667	0.421696	0.962500	0.118792	69	1538	1607
410	411	2012-02-15	1	1	2	0	3	1	1	0.348333	0.351629	0.531250	0.181600	141	4028	4169
92	93	2011-04-03	2	0	4	0	0	0	1	0.378333	0.378767	0.480000	0.182213	1651	1598	3249
47	48	2011-02-17	1	0	2	0	4	1	1	0.435833	0.428658	0.505000	0.230104	259	2216	2475
508	509	2012-05-23	2	1	5	0	3	1	2	0.621667	0.584612	0.774583	0.102000	766	4494	5260

Testing Set (5 baris):

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
703	704	2012-12-04	4	1	12	0	2	1	1	0.475833	0.469054	0.733750	0.174129	551	6055	6606
33	34	2011-02-03	1	0	2	0	4	1	1	0.186957	0.177878	0.437826	0.277752	61	1489	1550
300	301	2011-10-28	4	0	10	0	5	1	2	0.330833	0.318812	0.585833	0.229479	456	3291	3747
456	457	2012-04-01	2	1	4	0	0	0	2	0.425833	0.417287	0.676250	0.172267	2347	3694	6041
633	634	2012-09-25	4	1	9	0	2	1	1	0.550000	0.544179	0.570000	0.236321	845	6693	7538

**Kesimpulan hasil implementasi algoritma:** Melalui praktikum ini, saya berhasil menyelesaikan tahapan persiapan data dalam Machine Learning, saya berhasil memuat dataset besar dari Google Drive menggunakan Pandas, mengidentifikasi hubungan antar variabel dengan visualisasi scatter plot yang menunjukkan korelasi negatif, Saya mengimplementasikan pembagian data yang akurat menjadi Training, Validation, dan Testing Set. Ini adalah dasar yang lumayan karena memastikan bahwa dataset siap digunakan untuk melatih, menyetel, dan menguji model ML secara objektif dan terstruktur.

**LINK GITHUB UPLOAD TUGAS :** <https://github.com/Yurida26/Machine-Learning/tree/c49d4c21796ac74a8612f07f959eb4cf5f829ac5/Praktikum2>

