

# Tugas 2: Praktikum Mandiri 2 Machine Learning

**Hisyam Wildan Alfath - 0110222206**

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

E-mail: [hisy22206ti@student.nurulfikri.ac.id](mailto:hisy22206ti@student.nurulfikri.ac.id)

**Abstract.** Praktikum ini melatih kemampuan membaca, mengolah, dan membagi dataset menggunakan pandas dan scikit-learn di lingkungan Python. Mulai dari membaca file CSV yang berisi data kompleks terkait atribut cuaca dan pengguna, kemudian membagi data menjadi subset training, validation, dan testing untuk persiapan pemodelan machine learning. Visualisasi dan eksplorasi data menggunakan statistik deskriptif serta grafik juga dilakukan untuk memahami karakteristik data. Praktikum mandiri memperdalam pengalaman dengan dataset nyata yang lebih besar dan beragam, serta mengaplikasikan teknik pembagian data secara mandiri agar diperoleh model yang siap diuji dan dioptimalkan. Keseluruhan proses ini menyiapkan dasar kuat dalam alur kerja analisis data dan machine learning yang efektif dan terstruktur.

## 1. Praktikum 02

- Menghubungkan Colab dengan Google Drive

```
[1] ✓ 20s # Menghubungkan colab dengan google drive
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Kode ini memanggil modul drive dari google.colab untuk menghubungkan (mount) Google Drive ke lingkungan Google Colab, dan outputnya adalah muncul pesan “Mounted at /content/drive” yang menandakan folder Google Drive sekarang sudah bisa diakses pada path tersebut di Colab.

- Mendefinisikan Path Data

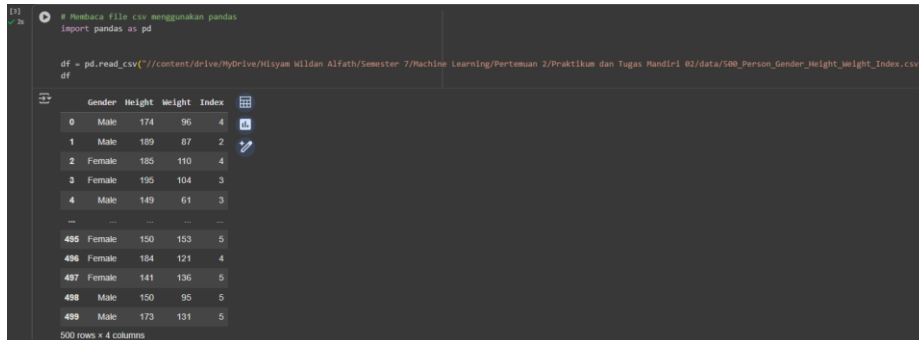
```
[2] ✓ 0s # Memanggil data set lewat gdrive
path = "/content/drive/MyDrive/Hisyam Wildan Alfath/Semester 7/Machine Learning/Pertemuan 2/Praktikum dan Tugas Mandiri 02"
```

Baris ini mendefinisikan variabel bernama path yang berisi lokasi folder spesifik di Google Drive untuk menyimpan dataset atau dokumen tugas, dan tidak menghasilkan output karena hanya deklarasi variabel; variabel ini nantinya bisa digunakan untuk mengambil atau memproses file di lokasi tersebut.

- Membaca file csv menggunakan pandas

```
[3] # Membaca file csv menggunakan pandas
import pandas as pd

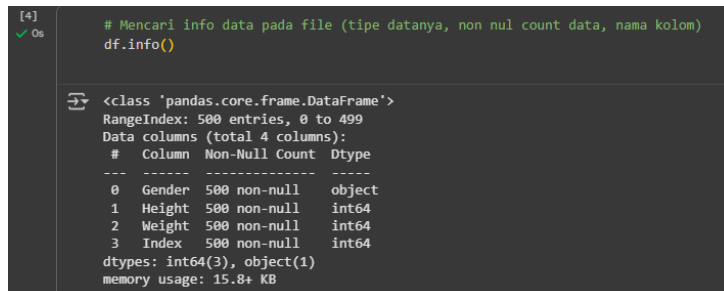
df = pd.read_csv("content/drive/MyDrive/Hisyan Wlidan Alfath/Semester 7/Machine Learning/Pertemuan 2/Praktikum dan Tugas Mandiri 02/data/500_Person_Gender_Height_Weight_Index.csv")
df
```



Kode ini mengimpor library pandas sebagai pd, lalu menggunakan fungsi pd.read\_csv untuk membaca file CSV dari Google Drive berdasarkan path yang diberikan. Outputnya adalah DataFrame yang menampilkan data dari file CSV tersebut, berupa tabel dengan kolom Gender, Height, Weight, dan Index serta berisi 500 baris data.

- Mendapatkan info data pada file CSV menggunakan pandas

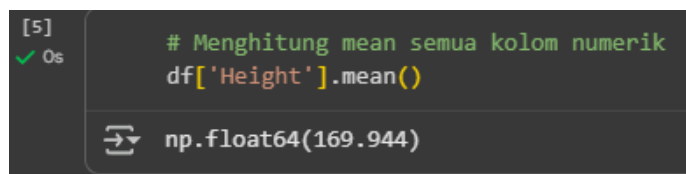
```
[4] # Mencari info data pada file (tipe datanya, non nul count data, nama kolom)
df.info()
```



Kode ini menjalankan fungsi df.info() untuk menampilkan ringkasan informasi DataFrame, termasuk tipe data, jumlah nilai non-null di setiap kolom, nama kolom, dan penggunaan memori. Outputnya menunjukkan DataFrame memiliki 4 kolom (Gender, Height, Weight, Index), masing-masing dengan 500 data non-null dan tipe data yang sesuai.

- Menghitung mean kolom numerik

```
[5] # Menghitung mean semua kolom numerik
df['Height'].mean()
```



Kode ini menggunakan df['Height'].mean() untuk menghitung rata-rata (mean) dari seluruh nilai pada kolom Height di DataFrame. Outputnya adalah nilai mean tinggi, yaitu 169.944, dengan tipe data np.float64.

- Menghitung median kolom numerik

```
[6] ✓ Os # Menghitung median semua kolom numerik
      df['Height'].median()
⇒ 170.5
```

Kode ini menggunakan `df['Height'].median()` untuk mencari nilai tengah (median) dari semua data pada kolom Height di DataFrame. Outputnya adalah median tinggi, yaitu 170.5.

- Mencari modus kolom numerik

```
[7] ✓ Os # Mencari modus (hati-hati karena bisa lebih dari satu)
      df['Height'].mode()
⇒
      Height
0      188
dtype: int64
```

Kode ini menggunakan `df['Height'].mode()` untuk mencari nilai yang paling sering muncul (modus) pada kolom Height. Outputnya menunjukkan bahwa nilai modus pada kolom Height adalah 188, dan tipe datanya `int64`.

- Menghitung variansi kolom numerik

```
[8] ✓ Os # Menghitung Variansi & Standard Deviasi
      df.var(numeric_only=True)
⇒
      0
Height  268.149162
Weight  1048.633267
Index   1.836168
dtype: float64
```

Kode ini menggunakan `df.var(numeric_only=True)` untuk menghitung nilai variansi dari setiap kolom numerik pada DataFrame. Outputnya menunjukkan variansi untuk kolom Height adalah 268.149162, Weight adalah 1048.633267, dan Index adalah 1.836168, semuanya bertipe `float64`.

- Menghitung standar deviasi kolom numerik

```
[9]
✓ Os # Menghitung Standar Deviasi
      df.std(numeric_only=True)
```



Output:

	0
Height	16.375261
Weight	32.382607
Index	1.355053

dtype: float64

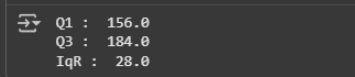
Kode ini menggunakan `df.std(numeric_only=True)` untuk menghitung nilai standar deviasi dari setiap kolom numerik pada DataFrame. Outputnya menunjukkan standar deviasi untuk kolom Height adalah 16.375261, Weight adalah 32.382607, dan Index adalah 1.355053, semuanya bertipe float64.

- Menghitung kuartil dan IQR kolom numerik

```
[10]
✓ Os # 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)
```



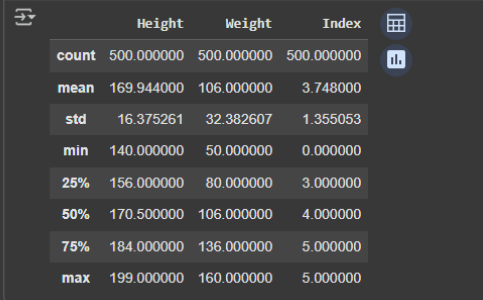
Output:

```
Q1 : 156.0
Q3 : 184.0
IQR : 28.0
```

Kode ini menggunakan `df['Height'].quantile(0.25)` untuk mencari kuartil pertama (Q1) dan `df['Height'].quantile(0.75)` untuk kuartil ketiga (Q3) dari kolom Height, lalu menghitung IQR (Interquartile Range) sebagai selisih  $Q3 - Q1$ . Outputnya menunjukkan nilai Q1 sebesar 156.0, Q3 sebesar 184.0, dan IQR sebesar 28.0.

- Statistik deskriptif kolom numerik

```
[11]
✓ Os # 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

Kode ini menggunakan `df.describe()` untuk menghasilkan ringkasan statistik deskriptif pada semua kolom numerik DataFrame, seperti count, mean, std (standar deviasi), nilai minimum, persentil (25%, 50%, 75%), dan nilai maksimum untuk masing-masing kolom Height, Weight, dan Index. Outputnya berupa tabel ringkasan statistik dari ketiga kolom numerik tersebut.

- Matriks korelasi kolom numerik

```
[12]
✓ Os
# Menghitung matriks korelasi untuk semua kolom numerik
correlation_matrix = df.corr(numeric_only=True)

# Menampilkan matriks korelasi
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

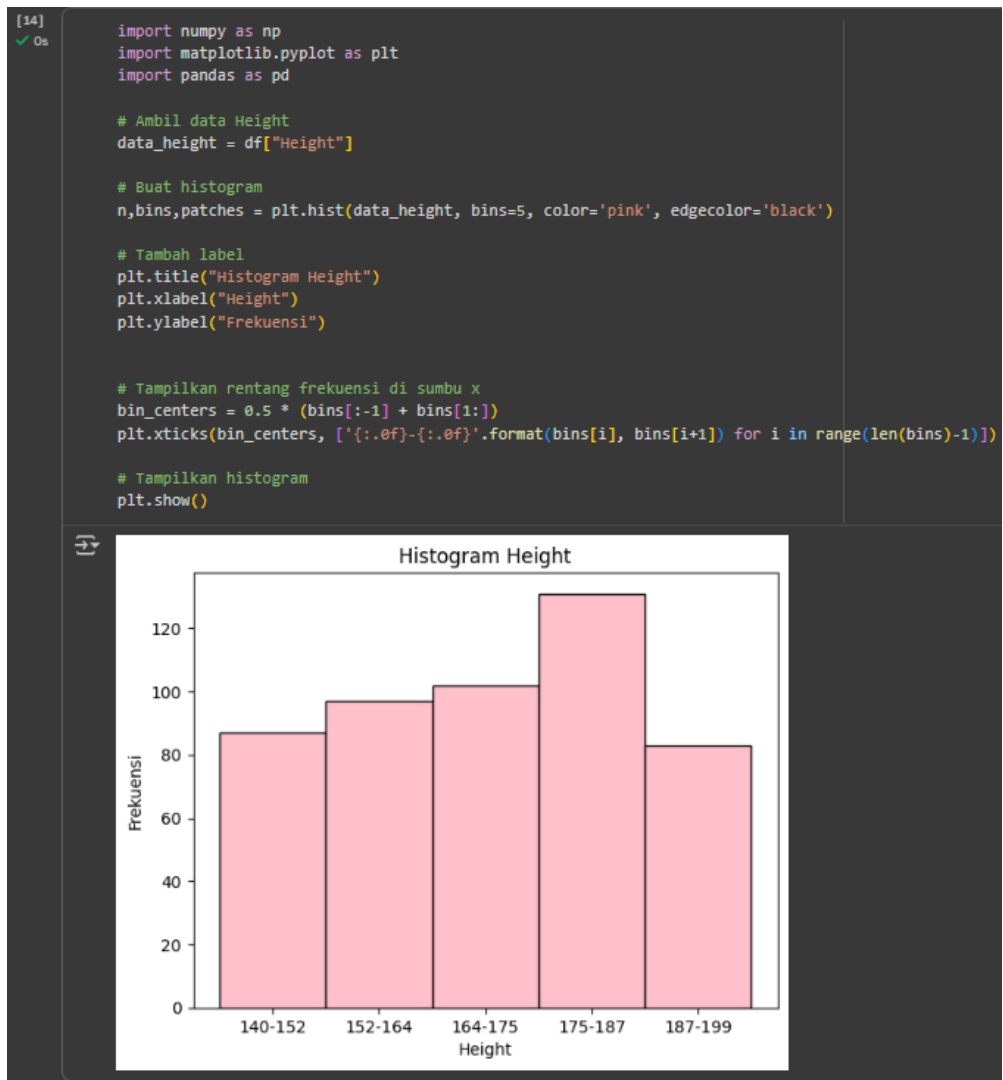
Kode ini menggunakan `df.corr(numeric_only=True)` untuk menghitung matriks korelasi antar semua kolom numerik, dan hasilnya disimpan pada variabel `correlation_matrix`. Output yang ditampilkan berupa tabel matriks korelasi yang menunjukkan hubungan antar kolom Height, Weight, dan Index, di mana masing-masing nilai menunjukkan derajat hubungan antar kolom tersebut.

- Visualisasi boxplot kolom Height dan Weight



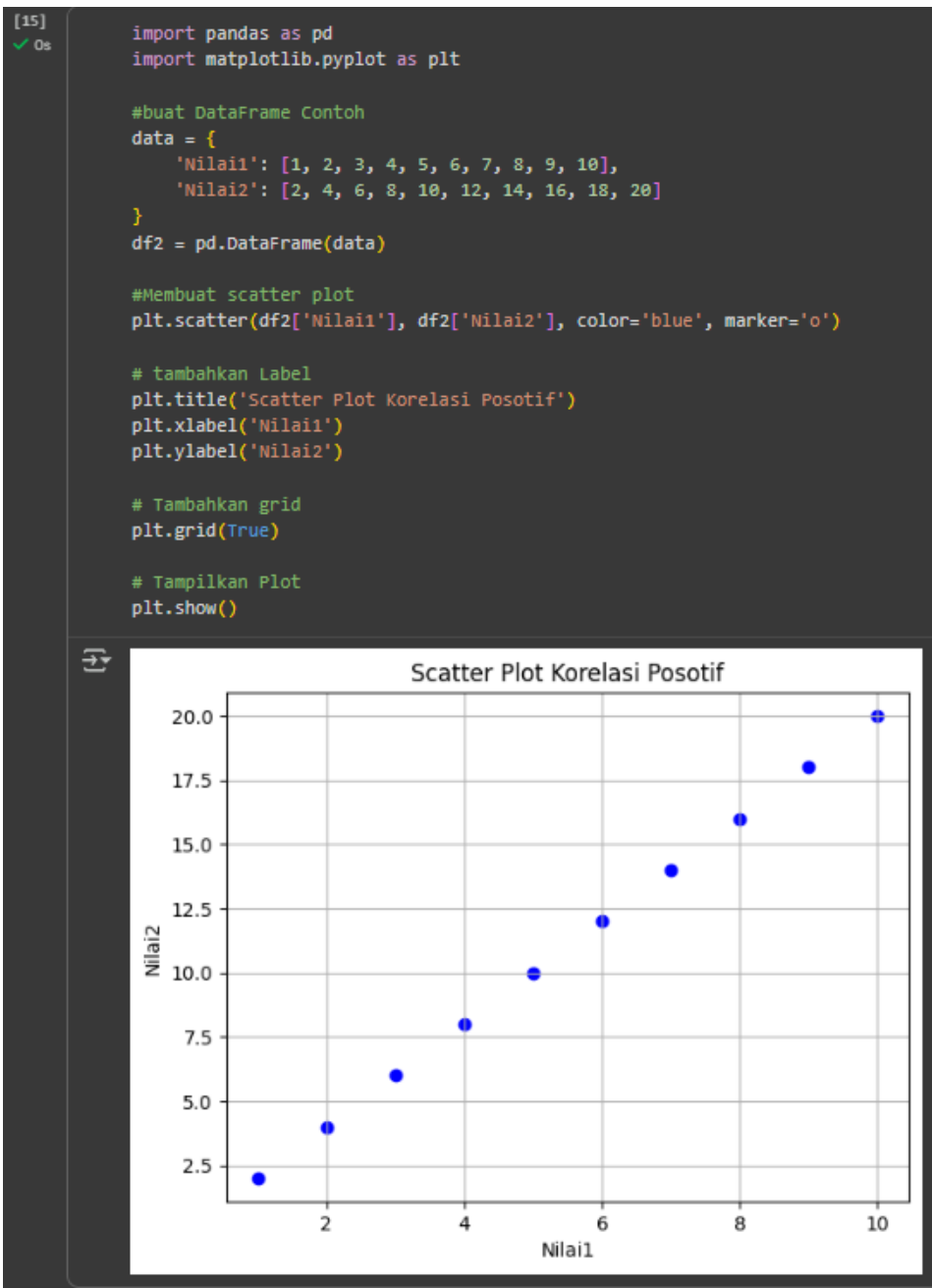
Kode ini menggunakan `df.boxplot(column=['Height', 'Weight'])` untuk membuat visualisasi boxplot dua kolom, yaitu Height dan Weight, sehingga memudahkan analisis distribusi, letak median, serta deteksi outlier pada kedua data. Outputnya berupa grafik boxplot berdampingan untuk kolom Height dan Weight.

- Visualisasi histogram kolom Height



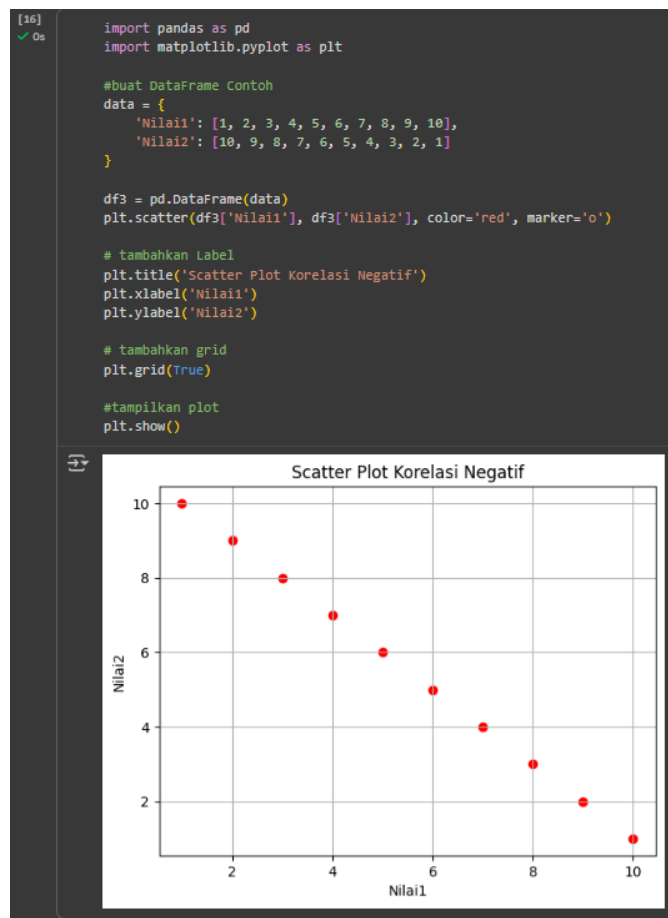
Kode ini membuat histogram untuk kolom Height menggunakan matplotlib dengan pembagian dalam 5 interval (bins), warna pink, dan label pada sumbu x serta y. Outputnya adalah grafik histogram yang memperlihatkan distribusi frekuensi nilai Height pada lima rentang interval berbeda.

- Scatter plot untuk korelasi positif



Kode ini membuat DataFrame sederhana dua kolom (Nilai1 dan Nilai2) lalu memvisualisasikan datanya dalam bentuk scatter plot menggunakan matplotlib, di mana setiap titik berwarna biru dan diberi marker bulat. Outputnya adalah grafik scatter plot yang memperlihatkan hubungan positif antara Nilai1 dan Nilai2, dilengkapi judul, label sumbu, dan grid pada plot.

- Scatter plot untuk korelasi negatif



Kode ini membuat DataFrame sederhana dua kolom (Nilai1 dan Nilai2), lalu divisualisasikan dalam scatter plot menggunakan matplotlib dengan warna merah dan marker bulat. Outputnya adalah grafik scatter plot yang menunjukkan hubungan negatif antara Nilai1 dan Nilai2, lengkap dengan judul, label pada sumbu, dan grid pada plot.

Kesimpulan praktikum ini adalah bahwa library Python seperti pandas, numpy, dan matplotlib sangat powerful dan efisien untuk membangun alur kerja analisis data mulai dari membaca data, eksplorasi data deskriptif, hingga visualisasi statistik dan hubungan antar variabel. Dengan alat-alat ini, data dapat diolah secara cepat dan visualisasi seperti boxplot, histogram, scatter plot sangat membantu dalam memahami distribusi data dan korelasi antar variabel. Praktikum ini memperlihatkan pentingnya langkah awal eksplorasi data sebagai fondasi analisis lebih lanjut di bidang machine learning atau ilmu data secara umum. Visualisasi yang tepat akan mempercepat insight dan pengambilan keputusan berdasarkan data.



## 2. Praktikum Mandiri 02

- Membaca file CSV dengan pandas

```
[17] # Membaca file csv menggunakan pandas
import pandas as pd

df2 = pd.read_csv("../content/drive/MyDrive/Hisyam Wildan Alfath/Semester 7/Machine Learning/Pertemuan 2/Praktikum dan Tugas Mandiri 02/data/day.csv")
df2
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

731 rows x 16 columns

Kode ini menggunakan pandas untuk membaca file CSV dari Google Drive dan menampilkannya sebagai DataFrame dengan 731 baris dan 16 kolom, berisi data harian yang komprehensif termasuk tanggal, musim, temperatur, kelembaban, dan jumlah pengguna casual serta terdaftar. Outputnya adalah tabel DataFrame yang merepresentasikan data lengkap tersebut.

- Membagi dataset menjadi data training, validation, dan testing

```
[18] import pandas as pd
from sklearn.model_selection import train_test_split

pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)

path = '/content/drive/MyDrive/Hisyam Wildan Alfath/Semester 7/Machine Learning/Pertemuan 2/Praktikum dan Tugas Mandiri 02//data/day.csv'

df2 = pd.read_csv(path)
df_train, df_test = train_test_split(df2, test_size=0.2, random_state=42)
df_train, df_val = train_test_split(df_train, test_size=0.1, random_state=42)

styles = [
    {'selector': 'th:not(:first-child)', # Hanya header kolom (kecuali index)
     'props': [('background-color', '#51c47d'), ('color', 'white'), ('border', '1px solid black')]}
]

print(f"Data Training: Jumlah data {len(df_train)}")
display(df_train.head().style.set_table_styles(styles).set_table_attributes('style="border:1px solid black;"'))

print(f"\nData Validation: Jumlah data {len(df_val)}")
display(df_val.head().style.set_table_styles(styles).set_table_attributes('style="border:1px solid black;"'))

print(f"\nData Testing: Jumlah data {len(df_test)}")
display(df_test.head().style.set_table_styles(styles).set_table_attributes('style="border:1px solid black;"'))
```

Kode ini mengimpor library pandas dan fungsi `train_test_split` dari `sklearn` untuk membagi dataset menjadi subset train, validation, dan test. Pengaturan display di pandas dibuat agar semua kolom terlihat secara penuh. File CSV dibaca dari Google Drive dan disimpan ke `df2`. Dataset diformat menjadi tiga bagian dengan ukuran 80% untuk training dan 20% testing, lalu dari data training dibuat lagi subset validation sebesar 10%. Selanjutnya, dibuat gaya tabel agar hasil tampilan DataFrame lebih rapi.

Terakhir, dicetak jumlah data dan ditampilkan lima baris awal dari masing-masing subset data tersebut dengan format yang sudah ditata agar mudah dibaca secara terpisah.

- Menampilkan data training, validation, dan testing hasil pembagian dataset

Data Training: Jumlah data 525																
	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
Data Validation: Jumlah data 59																
	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
Data Testing: Jumlah data 147																
	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

Kode ini mencetak jumlah data dan menampilkan lima baris awal dari masing-masing subset dataset training, validation, dan testing dengan format tabel yang sudah diberi gaya agar mudah dibaca dan dipahami. Outputnya menampilkan data dengan kolom lengkap seperti tanggal, musim, suhu, kelembaban, dan jumlah pengguna sehingga pengguna dapat memantau hasil pembagian data secara jelas dan terpisah.

Kesimpulan setelah mengerjakan praktikum mandiri adalah bahwa proses membaca dan membagi dataset menggunakan pandas dan scikit-learn penting dilakukan untuk mempersiapkan data dalam machine learning. Pembagian menjadi data training, validation, dan testing memungkinkan pelatihan model yang lebih handal dan evaluasi yang valid. Melalui praktikum ini, pemahaman mengenai alur persiapan data yang benar serta kemampuan menggunakan teknik pembagian data secara tepat dapat diperkuat sebagai dasar penting dalam proyek machine learning selanjutnya.

Link Praktikum dan Tugas Mandiri 02:

[https://github.com/HisyamWildan/TI03\\_HisyamW.A\\_0110222206/blob/main/Pertemuan%202/Praktikum%20dan%20Tugas%20Mandiri%2002/notebooks/praktikum02.ipynb](https://github.com/HisyamWildan/TI03_HisyamW.A_0110222206/blob/main/Pertemuan%202/Praktikum%20dan%20Tugas%20Mandiri%2002/notebooks/praktikum02.ipynb)