Tugas Laporan Hasil Praktikum02 dan Praktikum Mandiri

Machine Learning

Syaiful Ilham - 0110224084

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

0110224084@student.nurulfikri.ac.id

1. Load Data dari Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

- `drive.mount()` → menghubungkan Google Drive ke Colab.

Dataset berisi **500 baris & 4 kolom**: *Gender, Height, Weight, Index*.

hasil :

```
from google.colab import drive
drive.mount('/content/drive')
```

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

2. Membaca data dari google drive di folder data yang pada praktikum 02

```
import pandas as pd

df = pd.read_csv("/content/drive/MyDrive/Praktikum02/data/500_Person_Gender_Height_Weight_Index.csv")
df
```

pd.read_csv() → membaca file CSV sebagai **DataFrame**.

df → menampilkan seluruh isi dari dataframe

Dataset berisi **500 baris & 4 kolom**: *Gender, Height, Weight, Index*.

3 . Informasi Dataset

```
df.info()
```

- Menampilkan tipe data setiap kolom & jumlah data.
- Mengecek apakah ada **missing values**.

hasil :

```
<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
```

Semua data terisi (tidak ada nilai kosong).

---

4. Statistik Dasar

```
df['Height'].mean()
```

```
np.float64(169.944)
```

```
df['Height'].median()
```

```
170.5
```

```
df['Height'].mode()
```

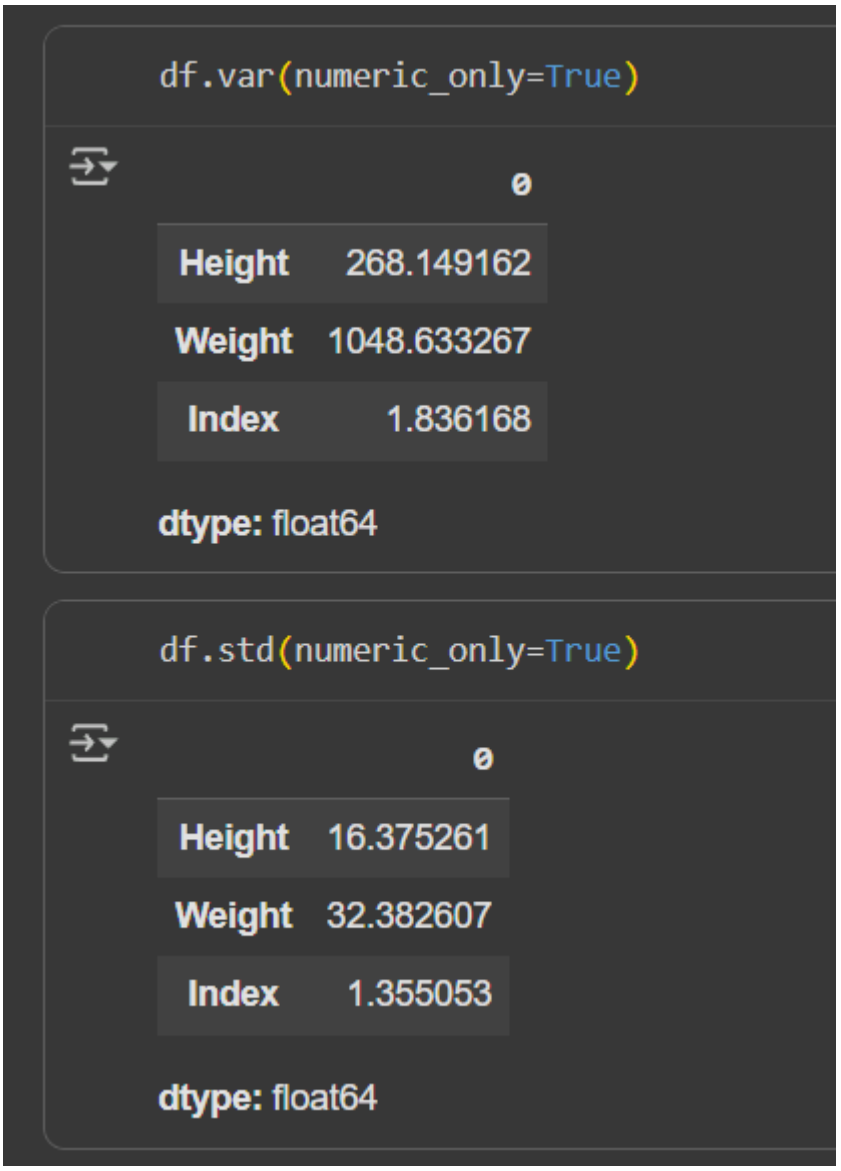
	Height
0	188

dtype: int64

- `mean()` → rata-rata.
- `median()` → nilai tengah.
- `mode()` → nilai yang paling sering muncul.

Mean = 169.94, Median = 170.5, Mode = 188.

5. Variansi & Standar Deviasi



- `var()` → sebaran data.
- `std()` → jarak rata-rata data dari mean.

hasil :

Std Dev Height ≈ 16.37 → data cukup bervariasi.

---

6. Quartile & IQR

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

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

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

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

- Q1 = kuartil bawah (25% data).
- Q3 = kuartil atas (75% data).
- IQR = Q3 - Q1 (rentang tengah data).

hasil :

Q1 = 156, Q3 = 184, IQR = 28.

---

7. Statistik Lengkap

```
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

hasil :

- Menampilkan **count, mean, std, min, Q1, median, Q3, max** untuk semua kolom numerik.

---

8. Korelasi antar Kolom

```
correlation_matrix = df.corr(numeric_only=True)
print("Correlation Matrix:")
print(correlation_matrix)
```

- `corr()` → menghitung hubungan antar variabel numerik.

Weight & Index = **positif kuat (0.80)**

Height & Index = **negatif lemah (-0.42)**

hasil :

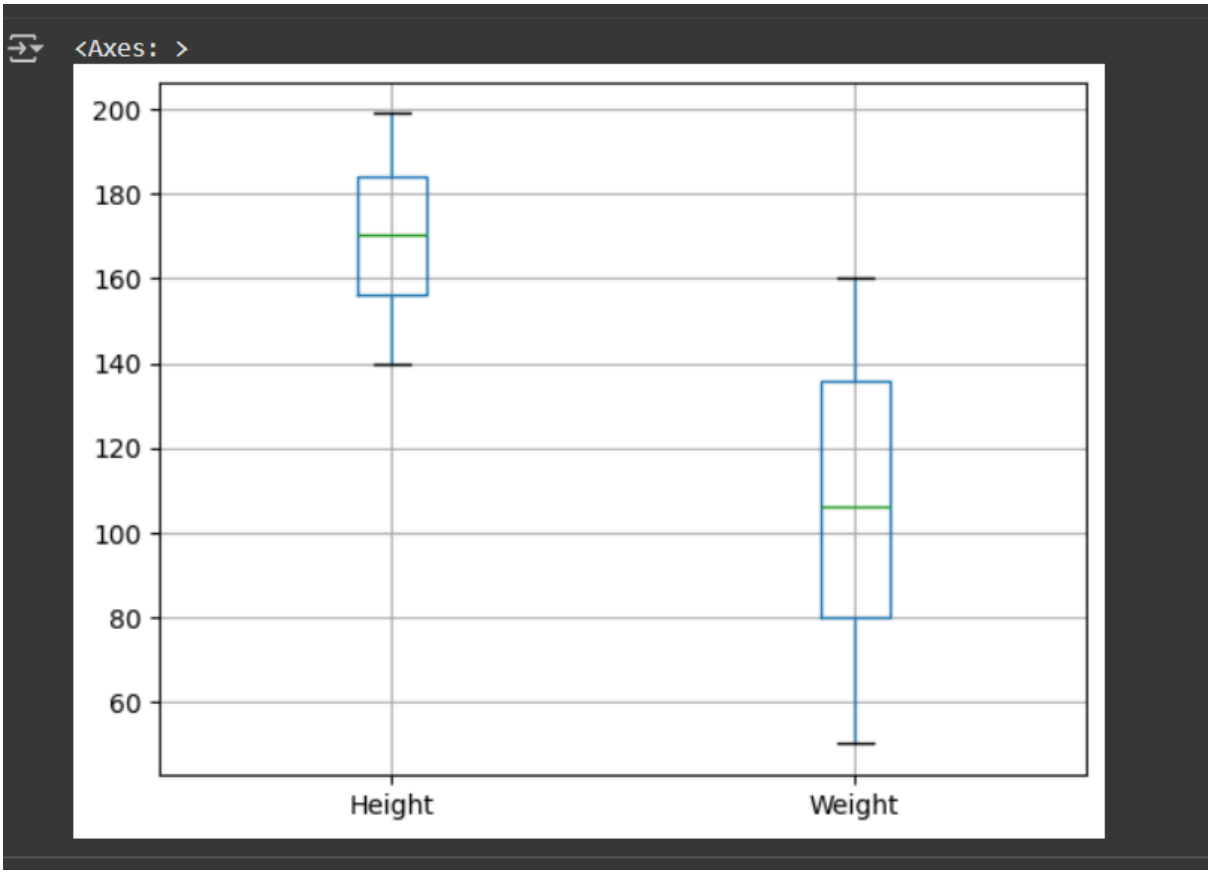
```
Correlation Matrix:
          Height    Weight     Index
Height  1.000000  0.000446 -0.422223
Weight  0.000446  1.000000  0.804569
Index  -0.422223  0.804569  1.000000
```

---

9. Boxplot (Deteksi Outlier)

```
import pandas as pd
import numpy as np

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

hasil :

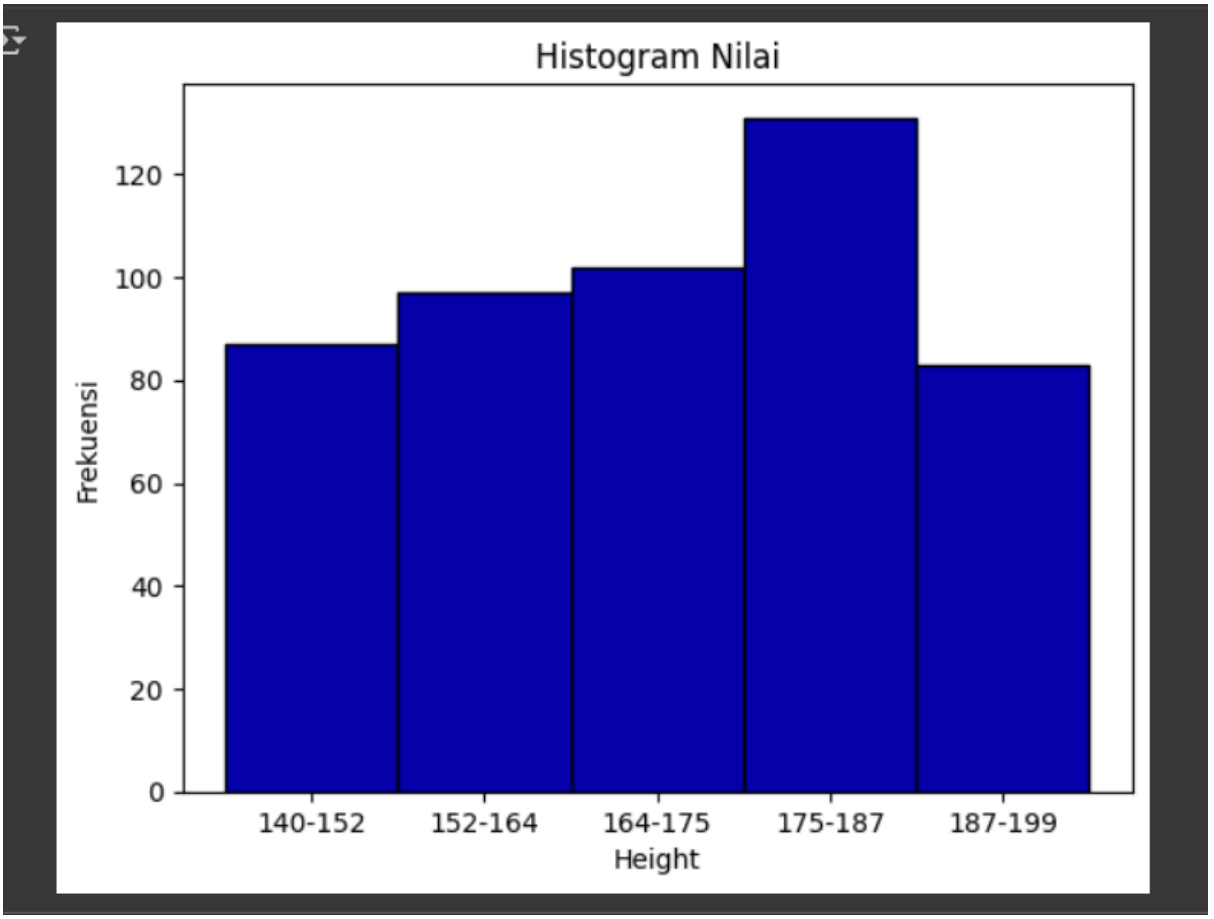


- Membuat boxplot untuk melihat sebaran data & outlier.

10. Histogram Tinggi Badan

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

# Load the dataframe (assuming the same path as in cell m4prfQ-7X--o)
df = pd.read_csv("/content/drive/MyDrive/Praktikum02/data/500_Person_Gender_Height_Weight_Index.csv")

data_height = df["Height"]

n,bins,patches = plt.hist(data_height,bins= 5,color='#0504aa',edgecolor='black')

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


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)])

plt.show()
```

hasil :



- Histogram menunjukkan distribusi tinggi badan.

- `bins=5` → data dibagi ke dalam 5 interval.

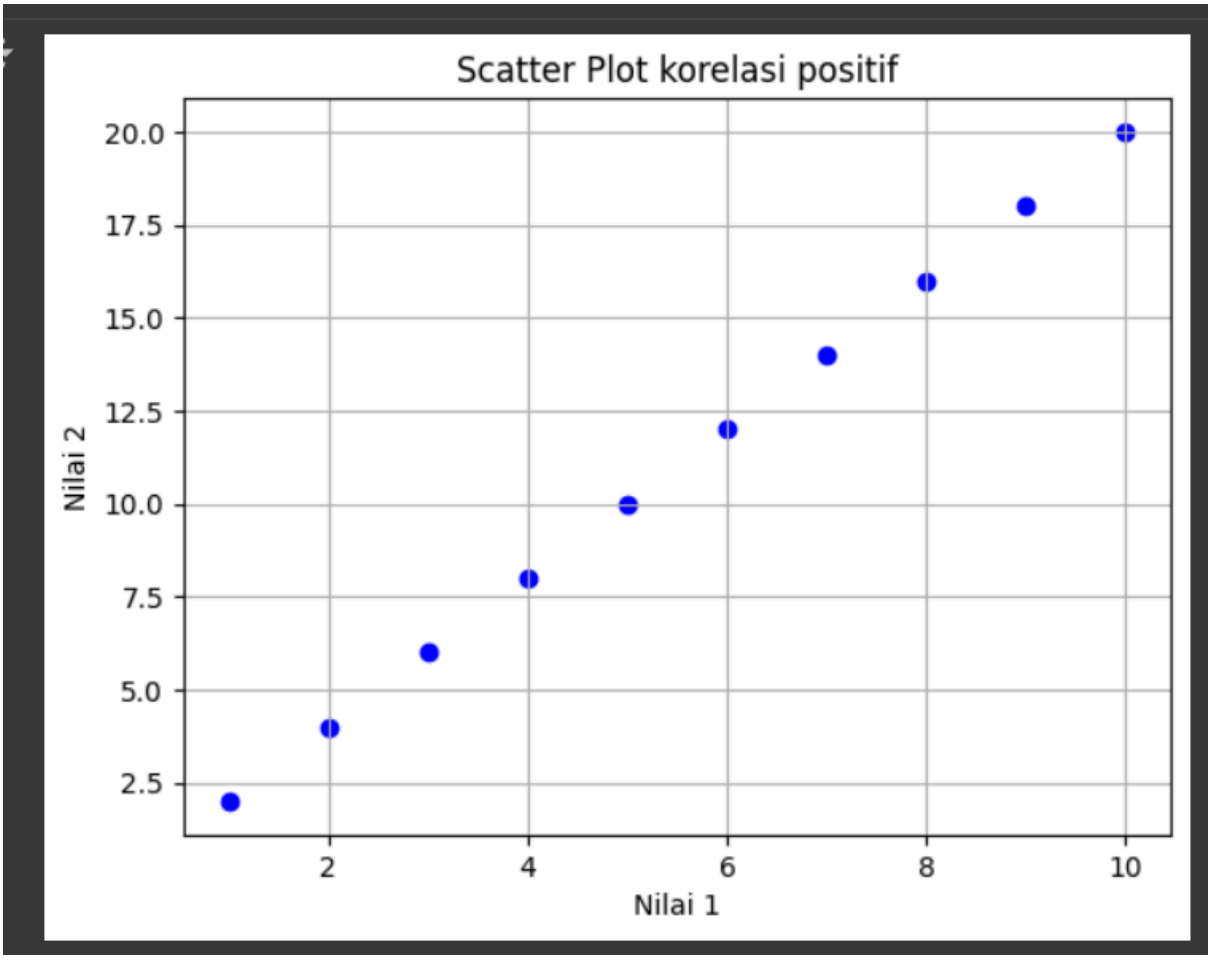
---

11. Scatter Plot (Korelasi Positif & Negatif)

a) Korelasi Positif

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

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)

plt.scatter(df2['nilai1'], df2['nilai2'], color='blue', marker='o')

plt.title('Scatter Plot korelasi positif')
plt.xlabel('Nilai 1')
plt.ylabel('Nilai 2')

plt.grid(True)

plt.show()
```

hasil :

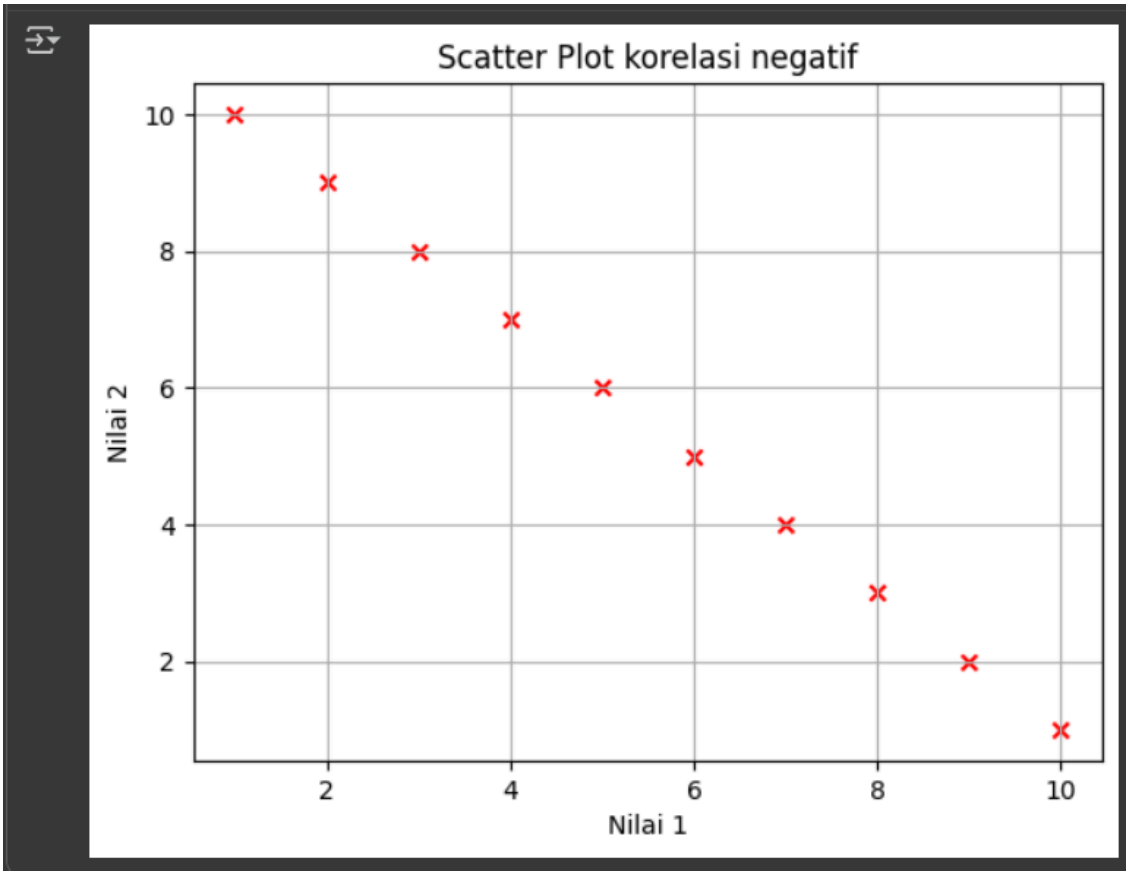


Saat **nilai1 naik, nilai2 juga naik** → korelasi positif.

b) Korelasi Negatif

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

data = {
    'nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'nilai2': [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
}

df2 = pd.DataFrame(data)

plt.scatter(df2['nilai1'], df2['nilai2'], color='red', marker='x')

plt.title('Scatter Plot korelasi negatif')
plt.xlabel('Nilai 1')
plt.ylabel('Nilai 2')

plt.grid(True)

plt.show()
```

hasil :



Saat **nilai1 naik, nilai2 turun** → korelasi negatif.

Ringkasan

- Dataset: **500 data, 4 kolom**.
- Statistik dasar: mean, median, mode, var, std, IQR.
- Korelasi: Weight & Index kuat positif, Height & Index negatif lemah.
- Visualisasi: **Boxplot, Histogram, Scatter Plot** memudahkan interpretasi data.

12. Praktikum Mandiri

```
import pandas as pd
from sklearn.model_selection import train_test_split

df = pd.read_csv("/content/drive/MyDrive/Praktikum02/data/day.csv")
print("Jumlah total data: ", len(df))

train,test = train_test_split(df, test_size=0.2, random_state=42)

train,val = train_test_split(train, test_size=0.1, random_state=42)

print("\nJumlah data train      : ",len(train))
print("Jumlah data validation : ",len(val))
print("Jumlah data test       : ",len(test))

print("\nData train:")
print(train.head())

print("\nData validation:")
print(val.head())

print("\nData test:")
print(test.head())
```

hasil :

```
Jumlah total data:  731

Jumlah data train      :  525
Jumlah data validation :  59
Jumlah data test       :  147

Data train:
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
657      658  2012-10-19       4   1    10        0        5           1
163      164  2011-06-13       2   0     6        0        1           1
305      306  2011-11-02       4   0    11        0        3           1
111      112  2011-04-22       2   0     4        0        5           1
538      539  2012-06-22       3   1     6        0        5           1

     weathersit      temp     atemp       hum  windspeed  casual  registered
657           2  0.563333  0.537896  0.815000   0.134954     753        4671
163           1  0.635000  0.601654  0.494583   0.305350     863        4157
305           1  0.377500  0.390133  0.718750   0.082092     370        3816
111           2  0.336667  0.321954  0.729583   0.219521     177        1506
538           1  0.777500  0.724121  0.573750   0.182842     964        4859

      cnt
657  5424
163  5020
305  4186
111  1683
538  5823
```

```
Data validation:
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
325      326  2011-11-22       4   0    11        0        2           1
410      411  2012-02-15       1   1     2        0        3           1
92        93  2011-04-03       2   0     4        0        0           0
47        48  2011-02-17       1   0     2        0        4           1
508      509  2012-05-23       2   1     5        0        3           1

     weathersit      temp     atemp       hum  windspeed  casual  registered
325           3  0.416667  0.421696  0.962500   0.118792      69        1538
410           1  0.348333  0.351629  0.531250   0.181600     141        4028
92            1  0.378333  0.378767  0.480000   0.182213    1651        1598
47            1  0.435833  0.428658  0.505000   0.230104     259        2216
508           2  0.621667  0.584612  0.774583   0.102000     766        4494

      cnt
325  1607
410  4169
92   3249
47   2475
508  5260

Data test:
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
703      704  2012-12-04       4   1    12        0        2           1
33        34  2011-02-03       1   0     2        0        4           1
300      301  2011-10-28       4   0    10        0        5           1
456      457  2012-04-01       2   1     4        0        0           0
633      634  2012-09-25       4   1     9        0        2           1

     weathersit      temp     atemp       hum  windspeed  casual  registered
703           1  0.475833  0.469054  0.733750   0.174129     551        6055
33            1  0.186957  0.177878  0.437826   0.277752      61        1489
300           2  0.330833  0.318812  0.585833   0.229479     456        3291
456           2  0.425833  0.417287  0.676250   0.172267    2347        3694
633           1  0.550000  0.544179  0.570000   0.236321     845        6693
```

```
      cnt
703  6606
33   1550
300  3747
456  6041
633  7538
```

`import pandas as pd` → Mengimpor library **pandas** untuk membaca dan mengolah data dalam bentuk tabel (DataFrame).

`from sklearn.model_selection import train_test_split` → Mengimpor fungsi **train_test_split** dari library **scikit-learn**, digunakan untuk **membagi dataset menjadi beberapa bagian** (train, validation, dan test).

`pd.read_csv()` → Membaca file `day.csv` dari lokasi di Google Drive dan menyimpannya ke variabel `df`.

`len(df)` → Menghitung jumlah total baris data dalam DataFrame.

`print()` → Menampilkan jumlah total data ke layar.

- Membagi dataset `df` menjadi **data training (train)** dan **data testing (test)**.
- `test_size=0.2` → 20% dari total data akan digunakan untuk **testing**, sisanya (80%) untuk **training**.
- `random_state=42` → Angka acak tetap agar hasil pembagian data selalu sama setiap kali dijalankan (reproducible).

Contoh pembagian:

- Total data = 731
- Data training = 80% → 584 baris
- Data testing = 20% → 147 baris
- Dari data **training (train)** tadi, dibagi lagi menjadi:
    - **Data train utama** (90%)
    - **Data validation (val)** (10%)

Validation digunakan untuk **mengukur performa model sebelum diuji ke data test.**

Jika `train` berisi 584 baris:

- Train akhir ≈ 525 baris
- Validation ≈ 59 baris

Kesimpulan:

Kode ini berfungsi untuk:

> **Membaca dataset `day.csv`, kemudian membagi data menjadi tiga bagian: train, validation, dan test, serta menampilkan jumlah dan contoh datanya.**

Link github praktikum = https://github.com/SyaifulIlham/Praktikum-machine_learning02.git
link google collab = praktikum02.ipynb