

Tugas 2:

Laporan hasil praktikum dan tugas praktikum

Nama Mahasiswa: Muhamad Aditia

Program Studi: Teknik Informatika, STT Terpadu Nurul Fikri, Depok

E-mail: 0110224213@student.nurulfikri.ac.id

Link G : <https://github.com/muhammadaditia433/Machine-Leraning>

Abstract

Pada proses pembangunan model Machine Learning, pembagian dataset menjadi beberapa bagian memiliki peran penting dalam memastikan kualitas dan kemampuan generalisasi model. Praktikum mandiri ini bertujuan untuk memahami cara membagi dataset menjadi training set, validation set, dan testing set menggunakan pustaka scikit-learn. Dataset yang digunakan adalah day.csv yang berisi data harian (kemungkinan data sewa sepeda). Proses dilakukan dengan menggunakan fungsi `train_test_split` untuk menghasilkan proporsi data yang sesuai. Hasil praktikum menunjukkan bahwa dataset berhasil dibagi menjadi tiga bagian dengan jumlah data yang seimbang sesuai dengan persentase pembagian yang telah ditentukan.

Pendahuluan

Dalam pengembangan sistem berbasis *machine learning*, pembagian dataset merupakan langkah krusial untuk mendapatkan model yang andal dan tidak *overfitting*. Dataset yang digunakan biasanya dipisah menjadi tiga bagian, yaitu data pelatihan (*training*), data validasi (*validation*), dan data pengujian (*testing*).

Tujuan dari pembagian ini adalah agar model dapat belajar dari data pelatihan, disesuaikan dengan data validasi, dan kemudian diuji menggunakan data pengujian yang belum pernah dilihat sebelumnya.

Metodologi

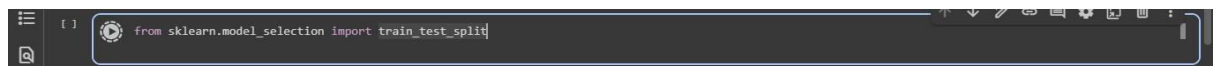
Proses eksperimen dilakukan menggunakan Python di lingkungan Google Colab.

Tahapan utama meliputi:

A. Tugas Praktikum Mandiri

1.1 Import Library

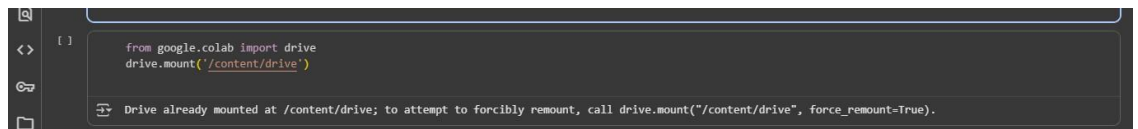
Tahap ini digunakan untuk memanggil pustaka atau library yang dibutuhkan dalam proses pengolahan data. Library seperti *pandas* digunakan untuk membaca dan mengelola data, sedangkan *sklearn.model_selection* digunakan untuk melakukan pembagian dataset menjadi data latih dan data uji.



Gambar 1. 1

1.2 Mount Google Drive

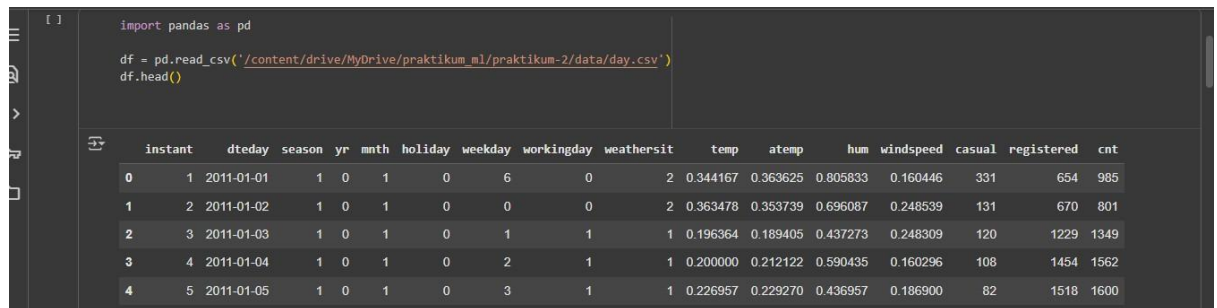
Langkah ini dilakukan jika bekerja di Google Colab. Fungsinya adalah untuk menghubungkan Google Drive agar file dataset yang tersimpan di sana bisa diakses dan digunakan langsung dalam program.



Gambar 1. 2

1.3 Membaca Dataset

Pada tahap ini, dataset dimuat ke dalam program agar dapat diolah. Dataset biasanya berupa file dengan format seperti CSV atau Excel. Setelah dibaca, data tersebut akan disimpan dalam variabel agar bisa digunakan untuk analisis lebih lanjut.



```
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/praktikum_ml/praktikum-2/data/day.csv')
df.head()
```

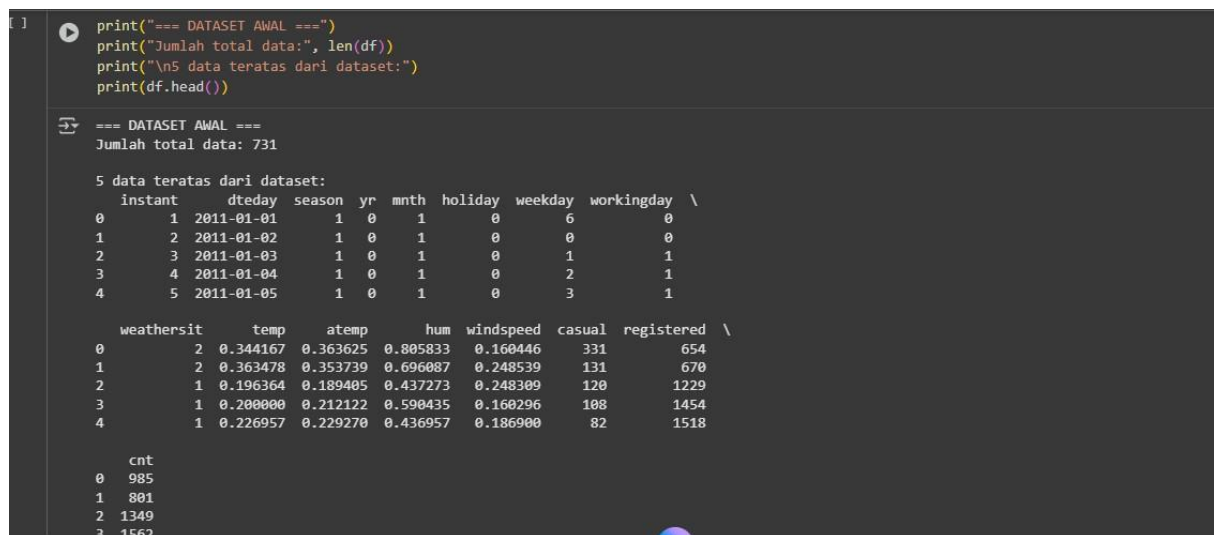
	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	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

Gambar 1. 3

1.4 Menampilkan Informasi Dataset

Tahap ini bertujuan untuk melihat isi dan struktur dari dataset. Informasi yang ditampilkan meliputi jumlah baris dan kolom, tipe data pada setiap kolom, serta ringkasan statistik data.

Hal ini membantu untuk memahami kondisi data sebelum dilakukan pemrosesan.



```
print("=== DATASET AWAL ===")
print("Jumlah total data:", len(df))
print("\n5 data teratas dari dataset:")
print(df.head())
```

```
=== DATASET AWAL ===
Jumlah total data: 731

5 data teratas dari dataset:
   instant  dteday  season  yr  mnth  holiday  weekday  workingday  \
0         1  2011-01-01     1   0     1         0         6           0
1         2  2011-01-02     1   0     1         0         0           0
2         3  2011-01-03     1   0     1         0         1           1
3         4  2011-01-04     1   0     1         0         2           1
4         5  2011-01-05     1   0     1         0         3           1

   weathersit    temp    atemp    hum  windspeed  casual  registered  \
0           2  0.344167  0.363625  0.805833  0.160446     331         654
1           2  0.363478  0.353739  0.696087  0.248539     131         670
2           1  0.196364  0.189405  0.437273  0.248309     120        1229
3           1  0.200000  0.212122  0.590435  0.160296     108        1454
4           1  0.226957  0.229270  0.436957  0.186900      82        1518

   cnt
0    985
1    801
2   1349
3   1562
```

Gambar 1. 4

1.5 Membagi Dataset

Dataset dibagi menjadi dua bagian, yaitu data latih dan data uji. Data latih digunakan untuk melatih model agar dapat mengenali pola dalam data, sedangkan data uji digunakan untuk mengukur seberapa baik model dapat memprediksi data baru.

```
train_data, test_data = train_test_split(df, test_size=0.2, random_state=42)

train_data, val_data = train_test_split(train_data, test_size=0.1, random_state=42)
```

Gambar 1. 5

1.6 Menampilkan Jumlah Data

Tahap ini digunakan untuk memastikan jumlah data pada masing-masing bagian (data latih dan data uji) sudah sesuai dengan proporsi yang diinginkan. Dengan demikian, kita dapat memastikan pembagian dataset telah dilakukan dengan benar.

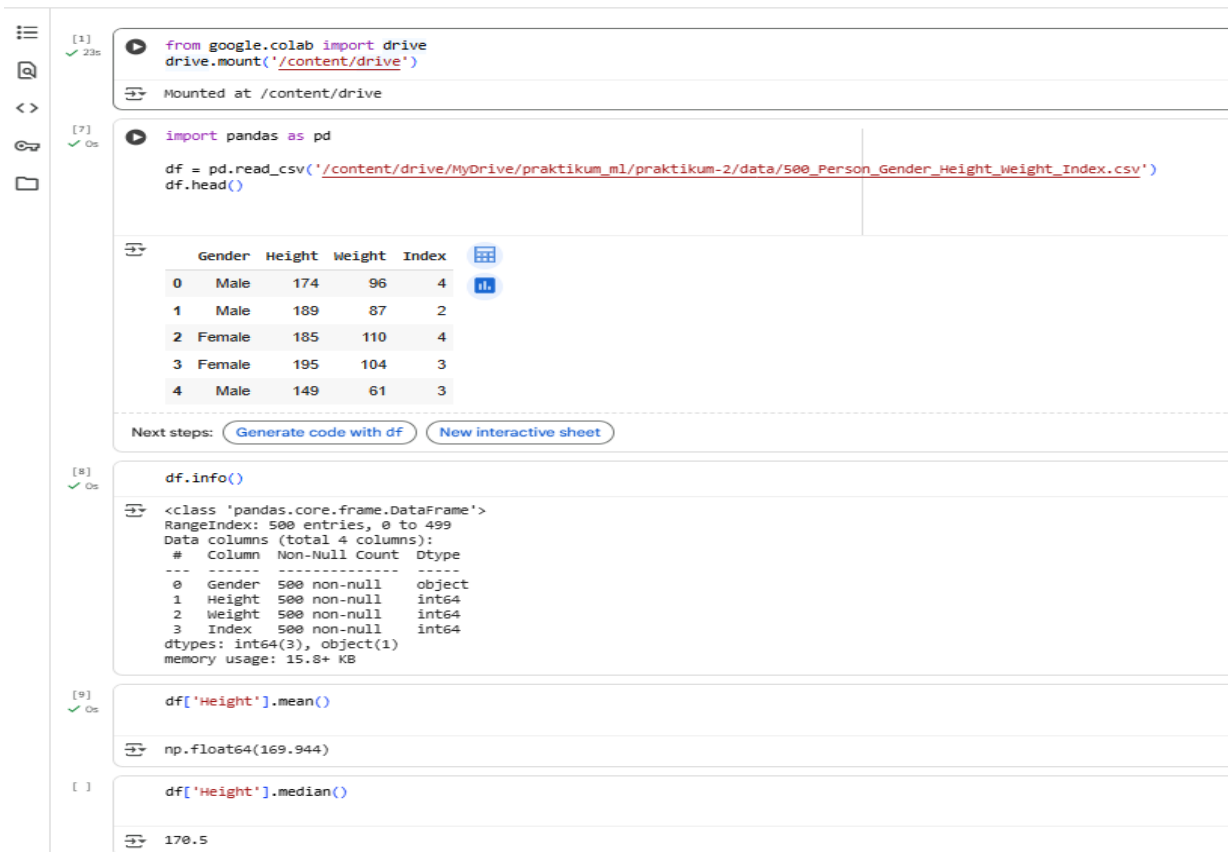
```
print("\n=== JUMLAH DATA ===")
print(f"Training: {len(train_data)}")
print(f"Validation: {len(val_data)}")
print(f"Testing: {len(test_data)}")

=== JUMLAH DATA ===
Training: 525
Validation: 59
Testing: 147
```

Gambar 1. 6

B. Praktikum Kelas

Kode 2.1



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

```
[1] ✓ 23s
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

```
[7] ✓ 0s
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/praktikum_ml/praktikum-2/data/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

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
[8] ✓ 0s
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
```

```
[9] ✓ 0s
df['Height'].mean()
```

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

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

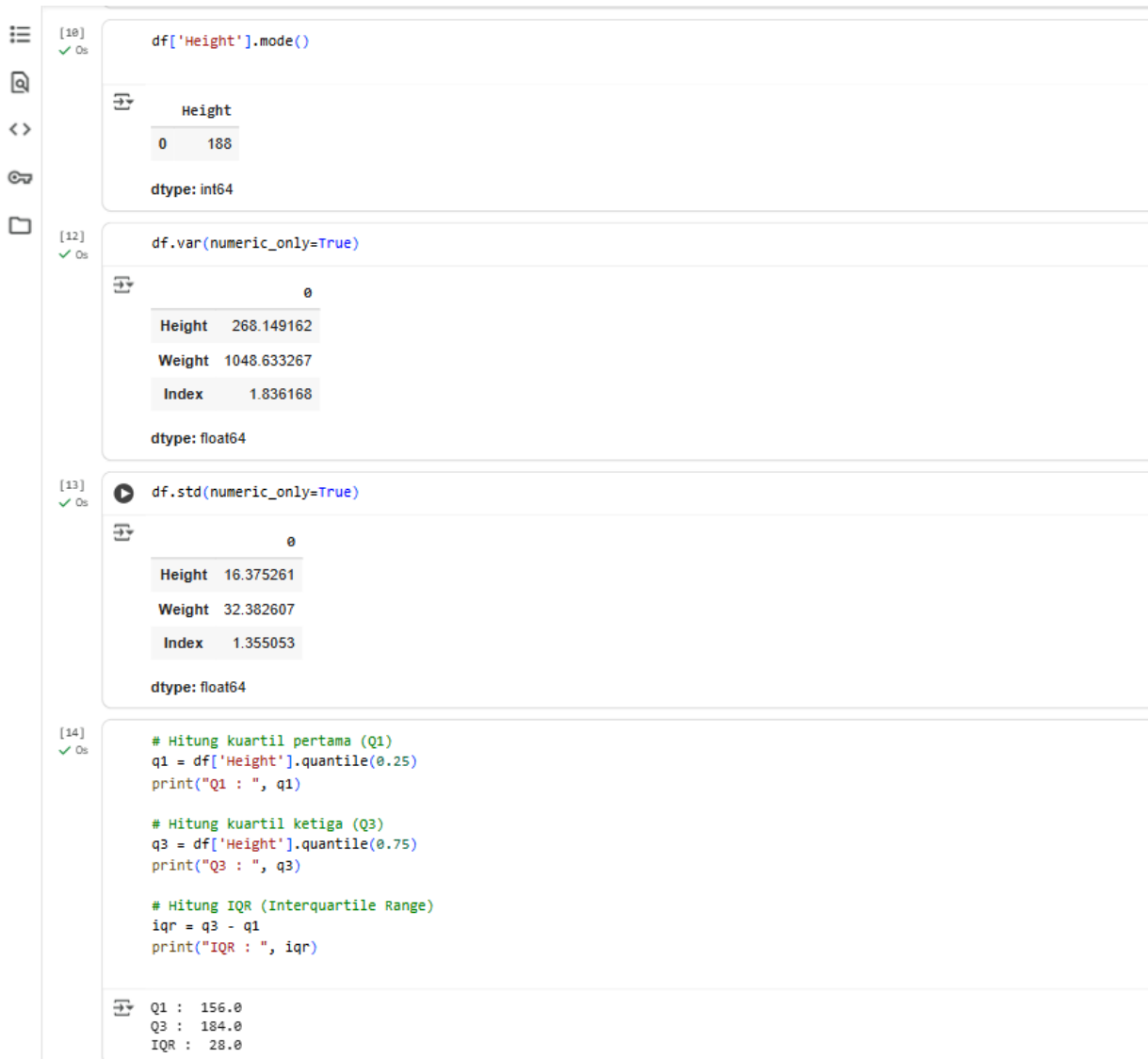
```
170.5
```

Gambar 2. 1

Kode di atas digunakan di Google Colab untuk menganalisis data tinggi dan berat badan. Pertama, Google Drive di-*mount* agar file CSV bisa diakses, lalu library Pandas dan NumPy digunakan untuk membaca dan mengolah data. Dataset dimuat dengan **pd.read_csv()** dan dicek strukturnya menggunakan

df.info(). Selanjutnya, dihitung rata-rata (**mean**) dan nilai tengah (**median**) dari kolom Height. Hasilnya menunjukkan rata-rata tinggi badan sekitar 169.94 cm dan median 170.5 cm, menandakan distribusi data cukup seimbang. Secara singkat, kode ini membaca data, menampilkan strukturnya, dan menghitung statistik dasar tinggi badan.

Kode 2.2



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

[12] df.var(numeric_only=True)
      0
Height  268.149162
Weight  1048.633267
Index   1.836168
dtype: float64

[13] df.std(numeric_only=True)
      0
Height  16.375261
Weight  32.382607
Index   1.355053
dtype: float64

[14] # 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)

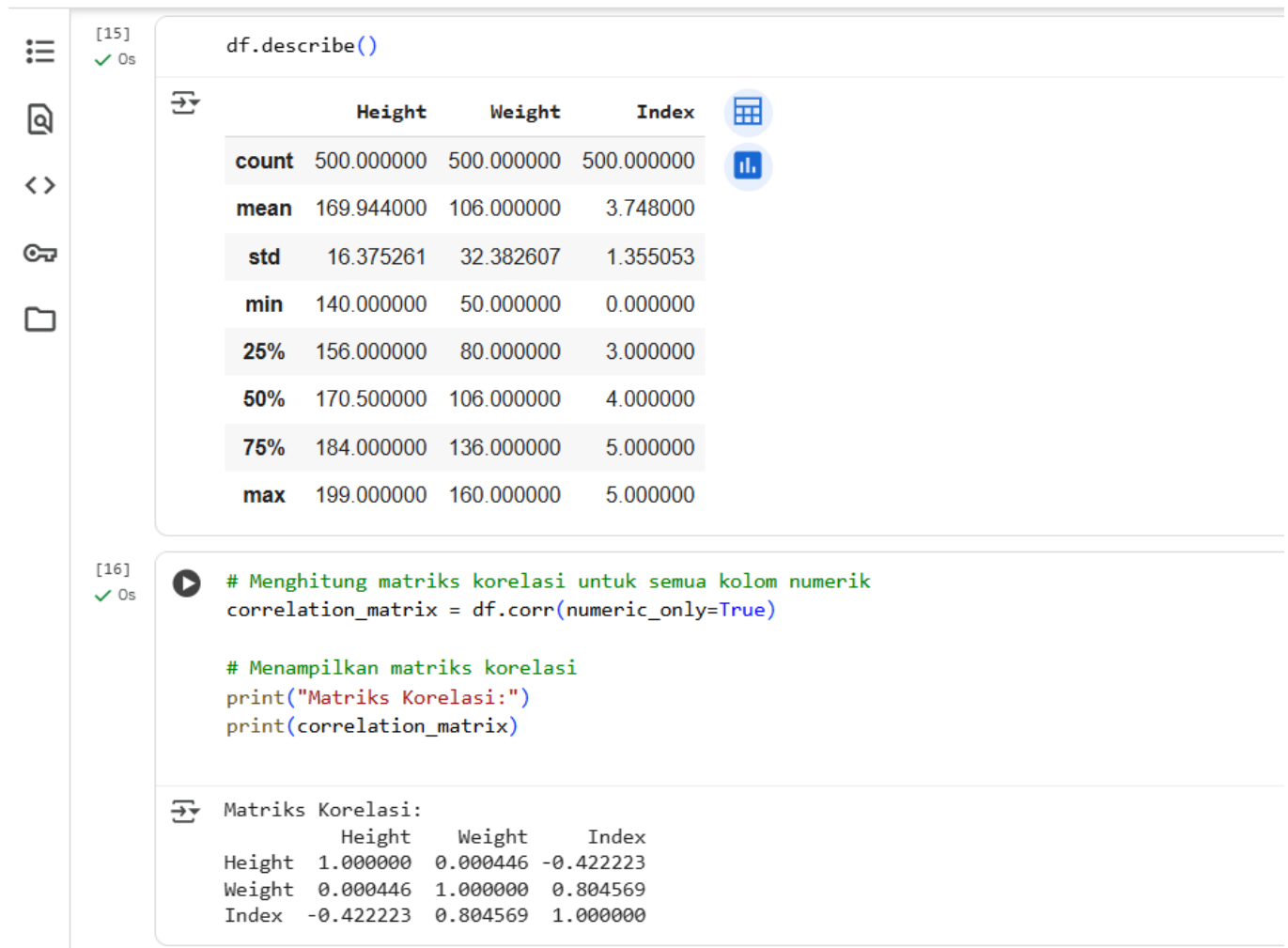
# Hitung IQR (Interquartile Range)
iqr = q3 - q1
print("IQR : ", iqr)

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

Gambar 2. 2

Kode di atas digunakan untuk menganalisis penyebaran data tinggi badan menggunakan statistik deskriptif. Pertama, fungsi `df['Height'].mode()` menghitung modus atau nilai tinggi yang paling sering muncul, yaitu 188 cm. Selanjutnya, `df.var()` digunakan untuk mencari variansi, sedangkan `df.std()` menghitung simpangan baku yang menunjukkan seberapa jauh data menyebar dari rata-rata; hasilnya menunjukkan tinggi badan memiliki simpangan sekitar 16.37 cm. Kemudian, kuartil pertama ($Q1 = 156$ cm) dan kuartil ketiga ($Q3 = 184$ cm) dihitung dengan fungsi `quantile()`, dan selisihnya disebut IQR (Interquartile Range) sebesar 28 cm, yang menunjukkan sebaran 50% data tengah. Secara keseluruhan, data tinggi badan dalam dataset cukup merata dengan sebagian besar nilai berada di rentang 156–184 cm.

Kode 2.3



```
[15]
✓ Os
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

```
[16]
✓ 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
```

Gambar 2. 3

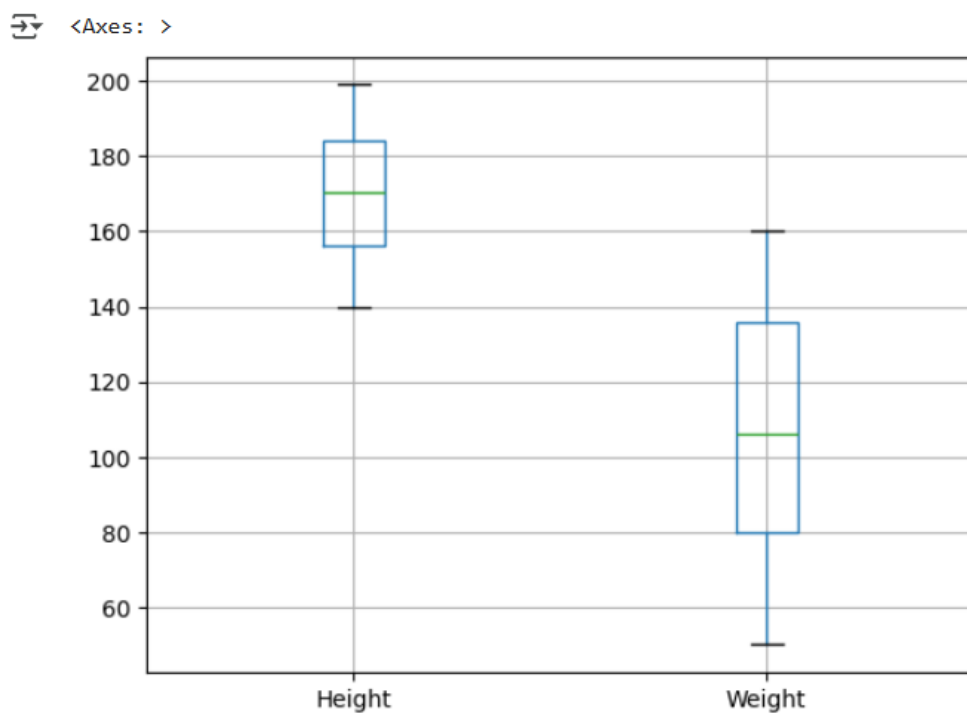
Kode di atas digunakan untuk menampilkan statistik deskriptif dan menghitung korelasi antar variabel numerik dalam dataset. Perintah `df.describe()` menampilkan ringkasan statistik seperti jumlah data, nilai rata-rata, standar deviasi, nilai minimum dan maksimum, serta kuartil untuk kolom **Height**, **Weight**, dan **Index**. Dari hasilnya, rata-rata tinggi badan adalah **169.94 cm**, berat badan **106 kg**, dan indeks rata-rata **3.75**, dengan rentang tinggi antara **140–199 cm**. Selanjutnya, perintah `df.corr(numeric_only=True)` digunakan untuk menghitung **matriks korelasi**, yang menunjukkan hubungan linear antar variabel. Hasilnya menunjukkan bahwa **Height** dan **Weight** memiliki korelasi hampir nol (0.0004), artinya keduanya tidak memiliki hubungan linear yang kuat, sedangkan **Weight** dan **Index** memiliki korelasi positif tinggi (**0.80**), menunjukkan bahwa semakin besar berat badan, semakin tinggi nilai indeks seseorang.

Kode 2.4

[17]
✓ 0s

```
import pandas as pd
import numpy as np

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



Gambar 2. 4

Kode tersebut menampilkan **boxplot** untuk kolom **Height** dan **Weight** guna melihat penyebaran data dan nilai median. Grafik menunjukkan tinggi badan memiliki median sekitar **170 cm** dengan rentang **140–200 cm**, sedangkan berat badan memiliki median **106 kg** dengan rentang **50–160 kg**. Visualisasi ini membantu memahami sebaran data serta mendeteksi adanya nilai ekstrem atau **outlier**.

Kode 2.5



Gambar 2. 5

Kode di atas menampilkan **histogram** data **Height** menggunakan **Matplotlib**. Data dibagi menjadi 5 kelompok dengan batang berwarna pink dan garis tepi hitam. Grafik menampilkan judul, label sumbu, serta rentang nilai di sumbu x, sehingga memperlihatkan **sebaran frekuensi tinggi badan** dalam dataset.

Kode 2.6

```
[19] ✓ Os
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]
}

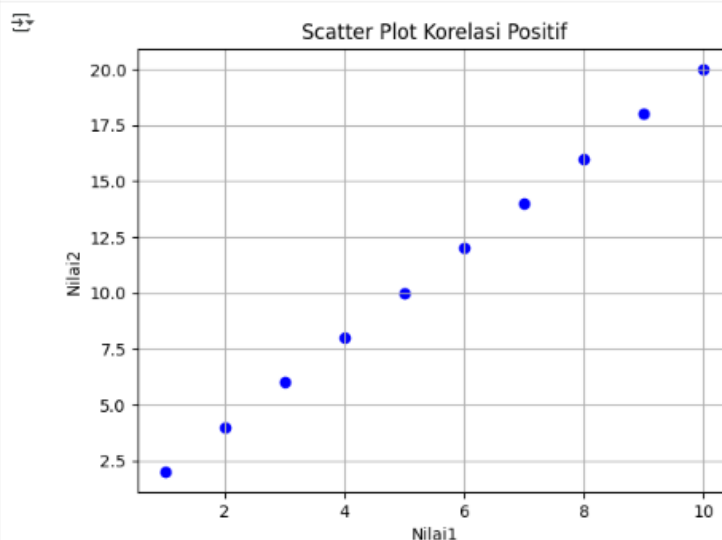
df2 = pd.DataFrame(data)

# 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()
```



Gambar 2. 6

Kode tersebut membuat **scatter plot** menggunakan **Matplotlib** untuk menampilkan hubungan positif antara **Nilai1** dan **Nilai2**. Data dimasukkan ke dalam DataFrame **df2**, lalu fungsi **plt.scatter()** menampilkan titik berwarna biru berbentuk lingkaran. Grafik diberi judul, label sumbu, dan grid agar mudah dibaca. Hasilnya menunjukkan **korelasi positif**, di mana nilai **Nilai2** meningkat seiring bertambahnya **Nilai1**, membentuk pola garis naik dari kiri ke kanan.

Kode 2.7

```
[20]
✓ Os
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]
}

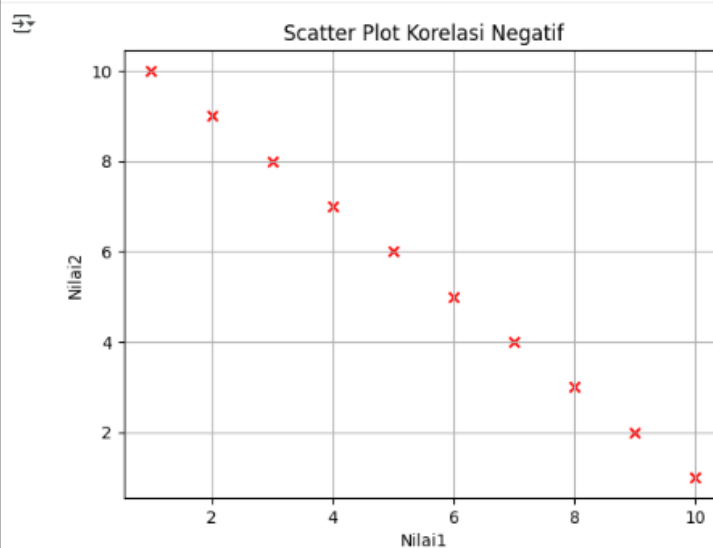
df3 = pd.DataFrame(data)

# 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()
```



Gambar 2. 7

Kode di atas membuat **scatter plot** menggunakan **Matplotlib** untuk menunjukkan hubungan negatif antara **Nilai1** dan **Nilai2**. Data dibuat dalam DataFrame **df3**, lalu **plt.scatter()** menampilkan titik berwarna merah berbentuk “x”. Grafik diberi judul, label sumbu, dan grid, menghasilkan pola menurun yang menunjukkan **korelasi negatif** antar variabel.

Kesimpulan

Berdasarkan hasil praktikum, dapat disimpulkan bahwa proses pembagian dataset merupakan langkah penting dalam persiapan data sebelum membangun model machine learning. Dengan melakukan pembagian dataset secara tepat, model dapat dilatih dengan data yang representatif dan diuji menggunakan data baru sehingga hasil prediksi menjadi lebih akurat dan tidak overfitting.

Proses ini juga membantu meningkatkan kemampuan model dalam melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya.