

# Laporan hasil praktikum dan tugas praktikum

## A. Tugas 2: Pembagian Dataset menjadi Training, Validation, dan Testing Set

Nama Mahasiswa: Al Hijir

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

E-mail: [0110224222@student.nurulfikri.ac.id](mailto:0110224222@student.nurulfikri.ac.id)

Link G : <https://github.com/Alhijir/Machine-Learning>

### **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.

## 1. 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.

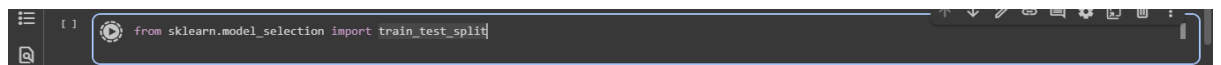
## 2. Metodologi

Proses eksperimen dilakukan menggunakan Python di lingkungan Google Colab.

Tahapan utama meliputi:

### 1. Import Library

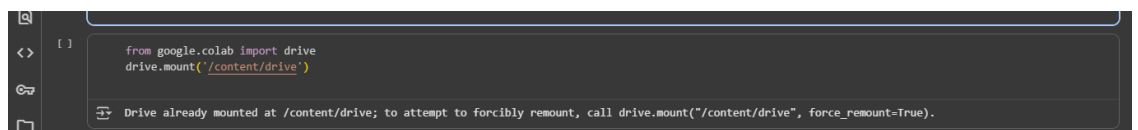
Tahap ini digunakan untuk memanggil pustaka atau library yang dibutuhkan dalam proses pengolahan data. Library seperti *pandas* digunakan untuk membaca dan mengelola data, sedangkan *sklearn.model\_selection* digunakan untuk melakukan pembagian dataset menjadi data latih dan data uji.



```
[ ] from sklearn.model_selection import train_test_split
```

### 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.



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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

### 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

#### 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
```

#### 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)
```

## 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
```

## 3.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.

## B. Tugas 2 Praktikum: Analisis Statistik Deskriptif Data Tinggi dan Berat Badan

### Abstrak

Praktikum ini bertujuan untuk memahami penerapan analisis statistik deskriptif menggunakan Python terhadap dataset tinggi dan berat badan. Analisis dilakukan untuk mengetahui gambaran umum data melalui perhitungan nilai mean, median, modus, variansi, standar deviasi, serta interquartile range (IQR). Selain itu, visualisasi data dilakukan menggunakan boxplot, histogram, dan scatter plot untuk melihat pola distribusi dan hubungan antarvariabel. Hasil praktikum menunjukkan bahwa data tinggi dan berat badan memiliki hubungan positif serta distribusi yang cukup seimbang tanpa banyak nilai ekstrem.

### 1. Pendahuluan

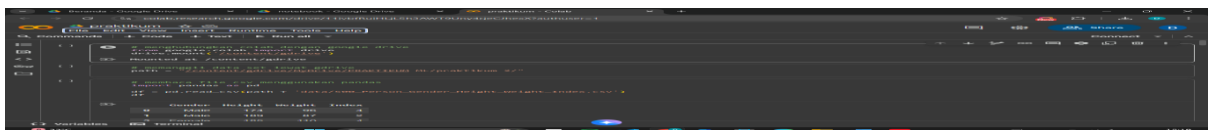
Analisis statistik deskriptif merupakan langkah awal dalam memahami suatu dataset sebelum dilakukan analisis lebih lanjut. Melalui statistik deskriptif, peneliti dapat mengetahui pola, sebaran, dan kecenderungan data. Dalam praktikum ini, digunakan dataset berisi data tinggi dan berat badan beberapa individu untuk dianalisis menggunakan Python. Hasil analisis akan membantu memahami karakteristik data serta hubungan antara dua variabel utama, yaitu tinggi badan dan berat badan.

### 2. Metodologi

Proses eksperimen dilakukan menggunakan **Python** di lingkungan **Google Colab**. Tahapan pelaksanaan praktikum meliputi beberapa langkah utama sebagai berikut:

#### 1. Import Library dan Mount Google Drive

Kode:



Langkah ini digunakan untuk menghubungkan Google Drive agar file dataset bisa diakses langsung dari penyimpanan drive ke Colab.

#### 2. Membaca Dataset

Kode:

```
File Edit View Insert Runtime Tools Help
ands | + Code + Text | ▶ Run all ▾

# membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + 'data/500_Person_Gender_Height_Weight_Index.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

Tahap ini berfungsi untuk membaca file CSV menggunakan **pandas** dan menyimpannya ke dalam variabel **df**. Dataset berisi kolom **Gender, Height, Weight, dan Index**.

### 3. Menampilkan Informasi Dataset

Kode:

```
ands | + Code + Text | ▶ Run all ▾

# Mencari info data pada file (tipe datanya, non null count 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
```

Menampilkan tipe data, jumlah baris, jumlah kolom, dan memastikan tidak ada nilai kosong (null) dalam dataset.

#### 4. Analisis Statistik Dasar

Kode:

```
File Edit View Insert Runtime Tools Help
nds | + Code + Text | ▶ Run all ▼

# Menghitung Variansi & Standar Deviasi
df.var(numeric_only=True)

0
Height    268.149162
Weight    1048.633267
Index      1.836168
dtype: float64

# Menghitung Standar Deviasi
df.std(numeric_only=True)

0
Height    16.375261
Weight    32.382607
Index      1.355053
```

Bagian ini menghitung **mean**, **median**, **modus**, **varian**, dan **standar deviasi** dari kolom numerik. Hasil ini membantu memahami sebaran data dan variasinya.

#### 5. Menghitung Kuartil dan IQR

Kode:

```
# 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
```

Langkah ini digunakan untuk mencari **Q1 (Kuartil 1)**, **Q3 (Kuartil 3)**, dan **IQR (Interquartile Range)**. Nilai ini berguna untuk mendeteksi potensi *outlier* pada data.

## 6. Statistik Deskriptif Lengkap

Kode:

```
# Untuk membuat statistika deskriptif 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

Menampilkan ringkasan statistik seperti jumlah data, nilai rata-rata, standar deviasi, nilai minimum, maksimum, serta kuartil.

## 7. Analisis Korelasi

Kode:

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

# Menampilkan matriks korelasi
print("Matriks Korelasi:")
print(correlation_matrix)
```

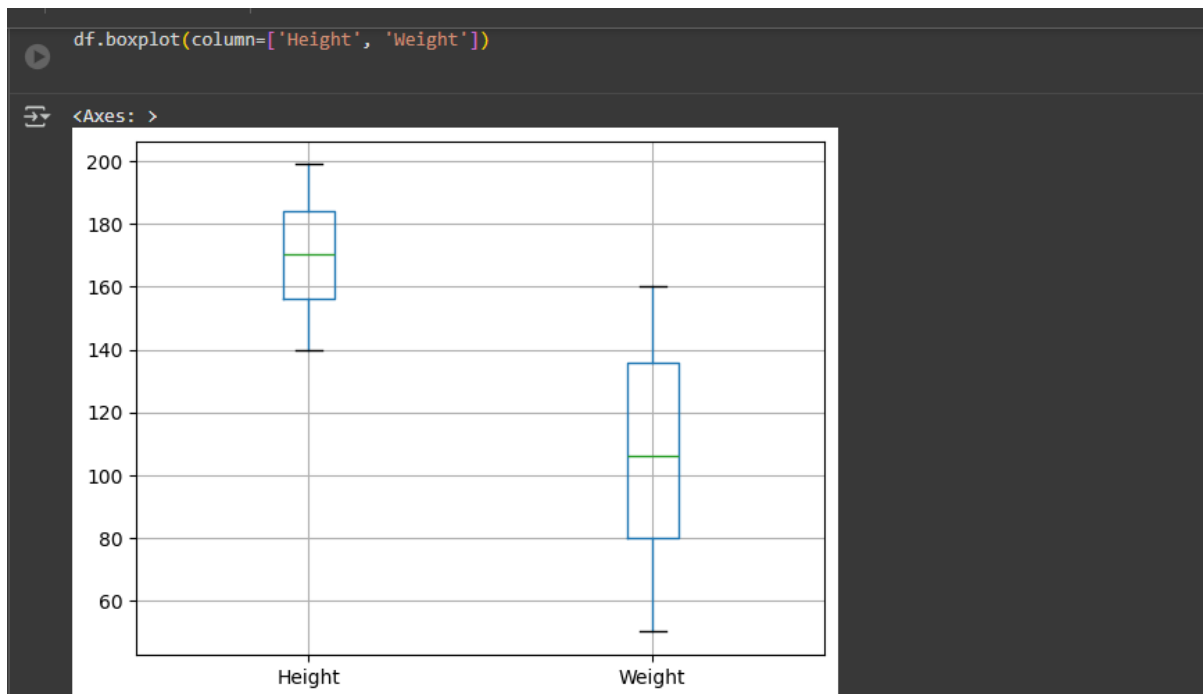
	Height	Weight	Index
Height	1.000000	0.000446	-0.422223
Weight	0.000446	1.000000	0.804569
Index	-0.422223	0.804569	1.000000

Digunakan untuk mengetahui hubungan antar variabel numerik dalam dataset, seperti korelasi antara **Height** dan **Weight**.



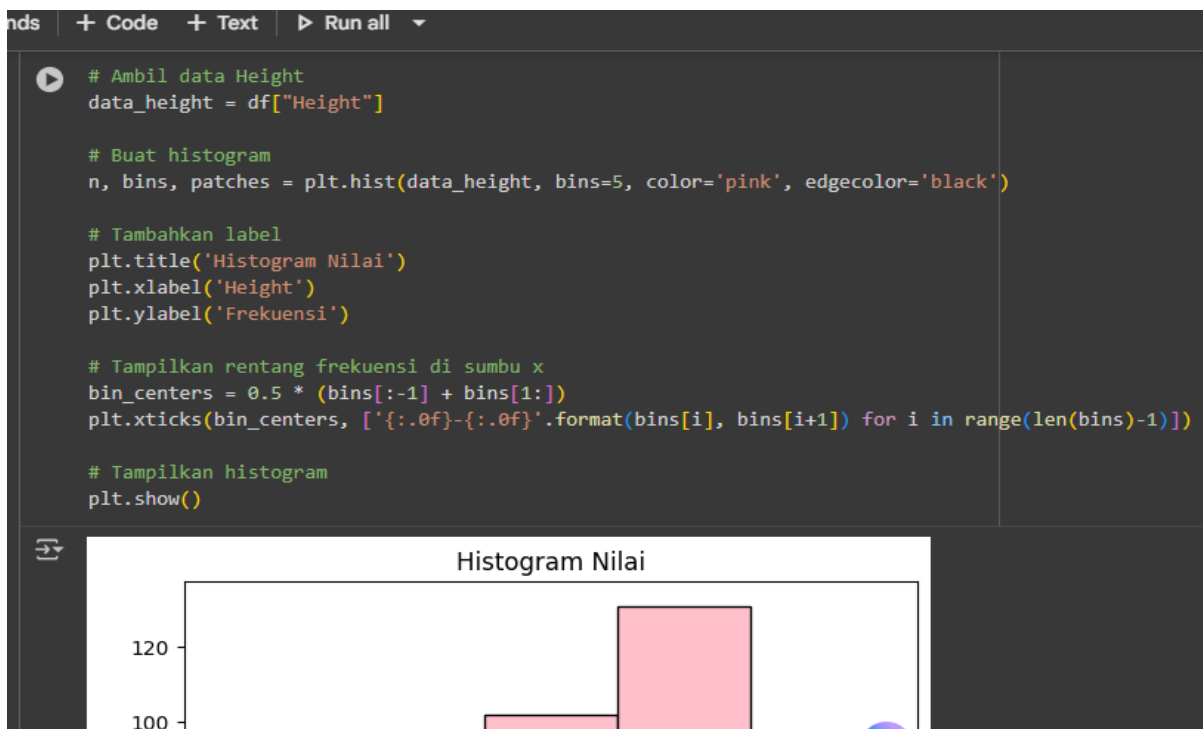
## 8. Visualisasi Data

Kode:



Membuat **boxplot** untuk melihat persebaran data dan mendeteksi *outlier*.

Kode tambahan:



Bagian ini membuat **histogram** untuk melihat sebaran nilai tinggi badan.

### 3. Kesimpulan

Berdasarkan hasil praktikum, dapat disimpulkan bahwa analisis statistik deskriptif menggunakan Python memberikan pemahaman yang jelas mengenai karakteristik data tinggi dan berat badan. Nilai mean, median, dan modus menunjukkan distribusi data yang cukup normal tanpa adanya penyimpangan ekstrem. Perhitungan variansi, standar deviasi, serta IQR menggambarkan bahwa data memiliki sebaran yang stabil dan tidak banyak mengandung outlier.

Selain itu, hasil korelasi memperlihatkan adanya hubungan positif antara tinggi dan berat badan, di mana semakin tinggi seseorang, cenderung berat badannya juga meningkat. Visualisasi berupa boxplot, histogram, dan scatter plot memperkuat hasil analisis dengan menampilkan pola distribusi dan hubungan antarvariabel secara lebih informatif.

Secara keseluruhan, praktikum ini menunjukkan pentingnya analisis statistik deskriptif dalam memahami struktur dan pola data sebelum melangkah ke tahap analisis yang lebih kompleks seperti pemodelan machine learning.

### Referensi

1. McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.
2. Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
3. Field, A. (2013). *Discovering Statistics Using IBM SPSS Statistics*. SAGE Publications.
4. VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

