# Tugas 2: Praktikum dan Latihan Mandiri 02

**Revani – 0110224111** [1]

[1] Teknik Informatika, STT Terpadu Nurul Fikri, Depok

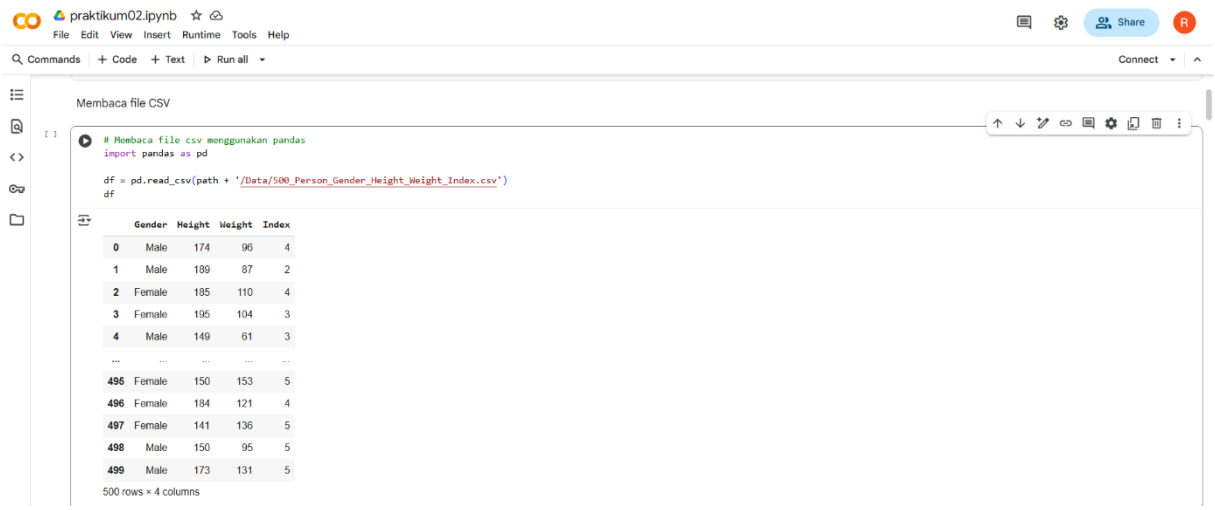
*E-mail: 0110224111@student.nurulfikri.ac.id

**Abstract.** Praktikum ini membahas penerapan statistik deskriptif dan probabilitas dasar dalam analisis data menggunakan Google Colab. Mempelajari cara membaca dataset, menghitung ukuran pemusatan dan penyebaran data, serta membuat visualisasi menggunakan Pandas, Matplotlib, dan Seaborn. Pada tugas mandiri, dilakukan pembagian dataset *day.csv* menjadi data training, validation, dan testing dengan proporsi 80%, 10%, dan 20% untuk memahami proses persiapan data pada *machine learning*. Kegiatan ini membantu mahasiswa memahami pentingnya analisis statistik serta pembagian dataset sebagai dasar dalam pengembangan model *machine learning*.

## 1. Praktikum Pekan 2

Pada praktikum pekan ke-2 ini, langkah awal yang dilakukan adalah menghubungkan Google Colab dengan Google Drive agar dapat mengakses file dataset.


Setelah itu, digunakan library Pandas untuk membaca data dari file CSV dan menyimpannya dalam bentuk *DataFrame.*




Selanjutnya dilakukan analisis statistik deskriptif, dimulai dari menampilkan informasi umum dataset menggunakan df.info()

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

Kemudian menghitung nilai mean, median, dan modus untuk memahami ukuran pemusatan data.

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

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

```python
# Menghitung median semua kolom numerik
df['Height'].median()
```

```
170.5
```

```python
# Mencari modus (hati-hati karena bisa lebih dari satu)
df['Height'].mode()
```

| | Height |
|---|---|
| 0 | 188 |

dtype: int64

Dilanjutkan dengan menghitung variansi dan standar deviasi untuk mengetahui seberapa besar penyebaran data, serta kuartil dan IQR untuk melihat distribusi nilai.

```python
# Menghitung variansi & standard deviasi
df.var(numeric_only=True)
```

| | 0 |
|---|---|
| Height | 268.149162 |
| Weight | 1048.633267 |
| Index | 1.836168 |

dtype: float64

```python
# Menghitung standard deviasi
df.std(numeric_only=True)
```

| | 0 |
|---|---|
| Height | 16.375261 |
| Weight | 32.382607 |
| Index | 1.355053 |

dtype: float64

```python
# 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
```

Setelah itu, digunakan df.describe() untuk menampilkan ringkasan statistik otomatis dari semua kolom numerik.

```python
# 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 |

Praktikum juga mencakup perhitungan korelasi antar variabel dengan df.corr() guna mengetahui hubungan linear antar data numerik.

```python
# 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
```

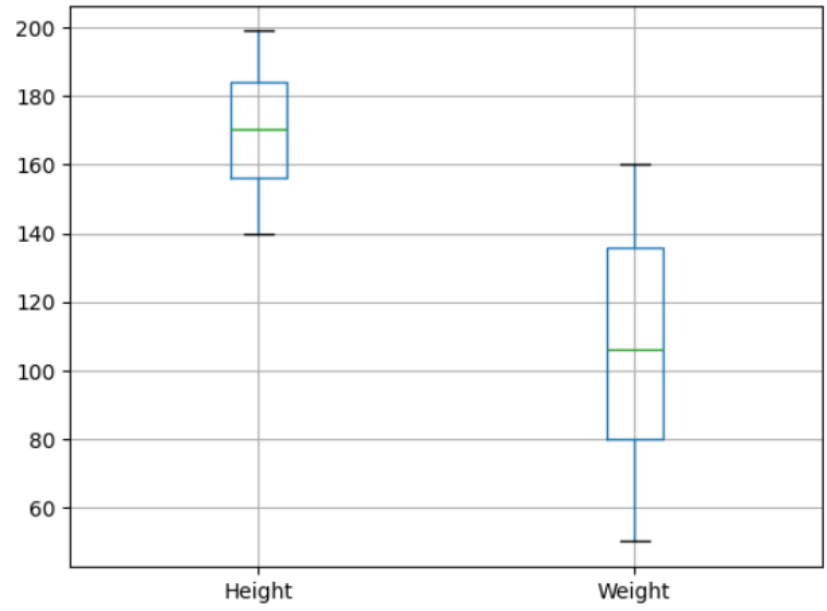
Bagian akhir praktikum berfokus pada visualisasi data, meliputi pembuatan boxplot untuk melihat sebaran dan pencilan data, histogram untuk menampilkan distribusi frekuensi, dan scatter plot untuk menggambarkan hubungan antar dua variabel. Melalui tahapan ini, mahasiswa memahami dasar analisis data secara menyeluruh mulai dari pembacaan, pengolahan, perhitungan statistik, hingga visualisasi hasil analisis.

Visualisasi Data

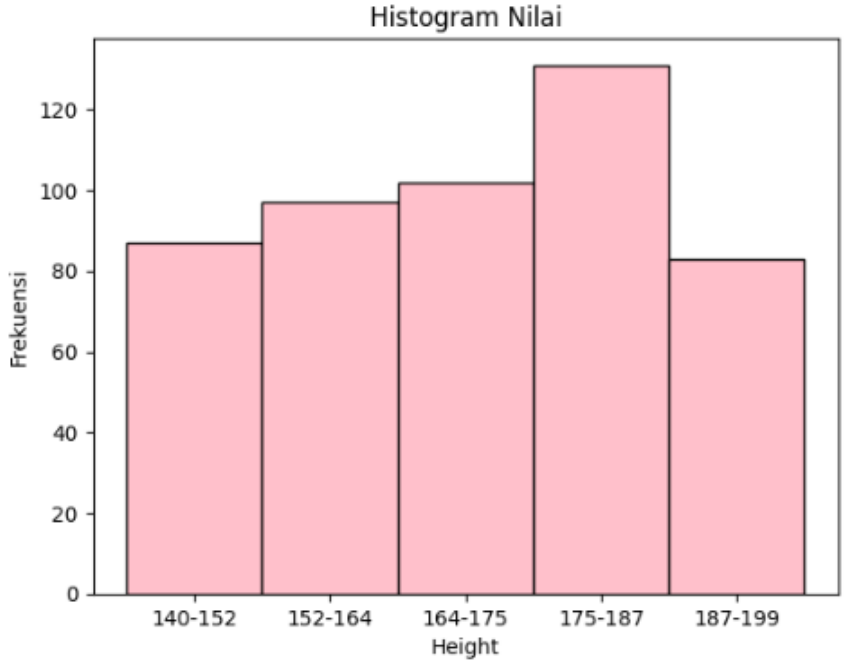
```
[ ]
```

```python
# Boxplot
import pandas as pd
import numpy as np

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

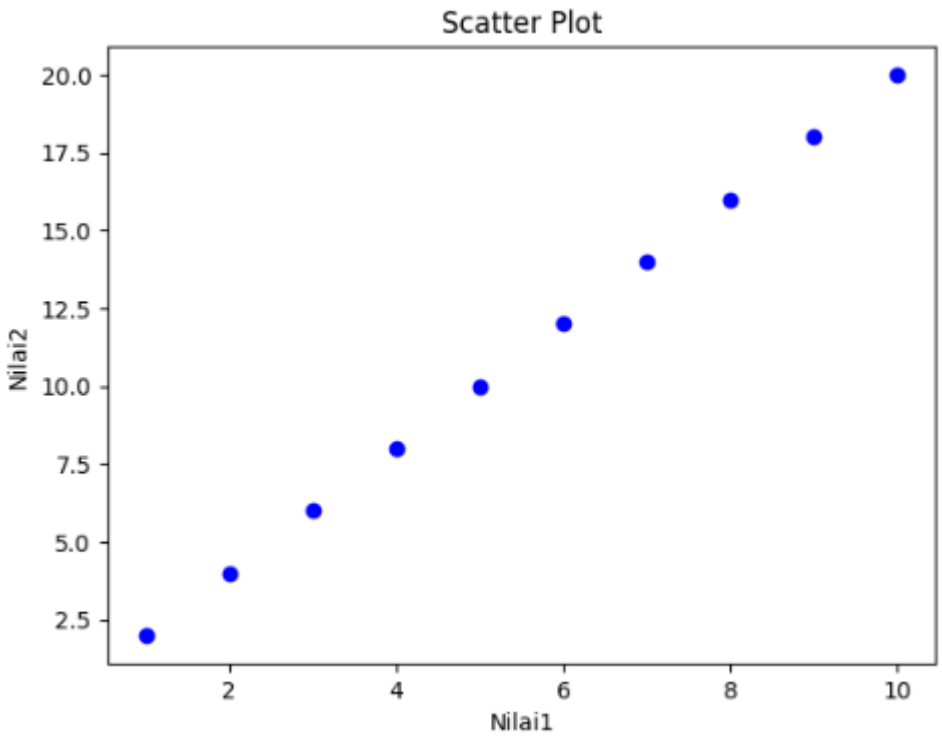
```
<Axes: >
```



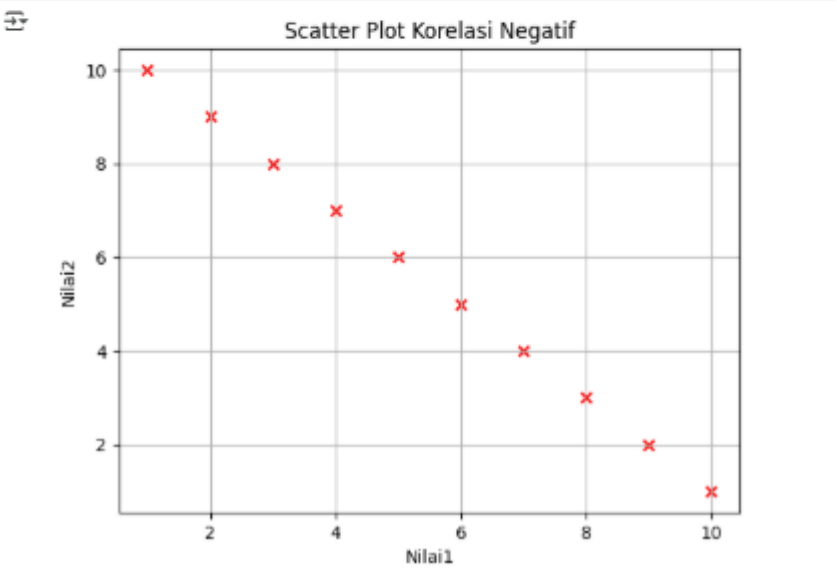
```python
# Histogram
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Ambil data Height
data_height = df["Height"]

# Buat histogram
n, bins, patches = plt.hist(data_height, bins=5, color='pink', edgecolor='black')

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

# Tampilkan rentang frekuensi di sumbu x
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)])

# Tampilkan histogram
plt.show()
```



```python
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')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

#tampilkan scatter plot
plt.show()
```



```python
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 1.** Pada bagian ini ditunjukkan praktikum pekan 2.

## Kesimpulan dan Hasil Implementasi

Pada praktikum pekan 2 ini bisa disimpulkan bahwa analisis statistik deskriptif dan probabilitas sangat penting sebagai langkah awal sebelum membuat model *machine learning*. Melalui praktikum ini, kita belajar memahami karakter data, melihat pola, dan mengetahui hubungan antarvariabel agar proses pemodelan nantinya lebih akurat. Implementasinya terlihat dari penggunaan Python di Google Colab untuk membaca data, menghitung nilai-nilai statistik, serta membuat visualisasi agar hasil analisis lebih mudah dipahami. Dengan memanfaatkan library seperti Pandas, Matplotlib, dan Seaborn, kita bisa mengolah data dengan cepat tanpa harus menghitung secara manual. Praktikum ini juga membantu dalam memahami bagaimana konsep statistik diterapkan langsung dalam pengolahan data di dunia nyata.

## 2. Latihan Praktikum Mandiri

Pada tugas praktikum mandiri diminta untuk membagi dataset *day.csv* menjadi tiga bagian, yaitu *data training* (80%)**,** *data validation* (10% dari *data training*)**,** dan *data testing* (20%). Tujuannya adalah untuk mempersiapkan data agar bisa digunakan dalam proses pelatihan, pengujian, dan validasi model *machine learning* dengan proporsi yang seimbang. Berikut ini adalah hasil latihan yang saya kerjakan pada latihan praktikum mandiri 02.

TUGAS PRAKTIKUM MANDIRI

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

# 1. Baca dataset
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Praktikum/Praktikum2/Data/day.csv')

print("Jumlah total data:", len(df))

# 2. Bagi dataset menjadi Training (80%) dan Testing (20%)
train_set, test_set = train_test_split(df, test_size=0.2, random_state=42)

# 3. Dari Training, ambil Validation (10% dari training)
train_set, val_set = train_test_split(train_set, test_size=0.1, random_state=42)

# 4. Tampilkan jumlah data masing-masing
print("\nJumlah data Training:", len(train_set))
print("Jumlah data Validation:", len(val_set))
print("Jumlah data Testing:", len(test_set))

# 5. Tampilkan 5 baris pertama masing-masing
print("\n--- Data Training ---")
print(train_set.head())

print("\n--- Data Validation ---")
print(val_set.head())

print("\n--- Data Testing ---")
print(test_set.head())
```

```
Jumlah total data: 731

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

--- Data Training ---
     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 Testing ---
     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
```

**Gambar 2.** Bagian ini merupakan hasil dari tugas latihan praktikum mandiri 2.

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

Bagian ini digunakan untuk memanggil dua library utama yaitu pandas yang digunakan untuk membaca, mengolah, dan menampilkan data dalam bentuk tabel (*DataFrame*). Serta *train_test_split* dari *scikit-learn* yang berfungsi untuk membagi dataset secara acak menjadi beberapa bagian (*training*, *validation*, *testing*).

```
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/Praktikum/Praktikum2/Data/day.csv')
print("Jumlah total data:", len(df))
```

Kode ini membaca file dataset *day.csv* yang tersimpan di Google Drive menggunakan fungsi *pd.read_csv()*. *len(df)* menghitung jumlah seluruh baris data yang ada. Output ini memberi tahu berapa banyak data yang akan dibagi ke dalam tiga bagian nanti.

```
train_set, test_set = train_test_split(df, test_size=0.2,
random_state=42)
```

Fungsi *train_test_split()* digunakan untuk membagi dataset menjadi *training set* (80%) dan *testing set* (20%)**.** Dengan parameter test_size=0.2 artinya 20% data digunakan untuk testing dan parameter random_state=42 menjaga hasil pembagian tetap sama setiap kali program dijalankan agar tidak acak berubah-ubah.

```
train_set, val_set = train_test_split(train_set, test_size=0.1,
random_state=42)
```

Bagian ini mengambil 10% dari data training untuk dijadikan validation set yang artinya 90% dari 80% total data tetap menjadi training, dan 10% dari 80% (atau sekitar 8% total data) menjadi validation. Validation set berguna untuk mengevaluasi model selama proses pelatihan, tanpa menyentuh data testing.

```
print("\nJumlah data Training:", len(train_set))
print("Jumlah data Validation:", len(val_set))
print("Jumlah data Testing:", len(test_set))
```

Pada bagian ini mencetak jumlah data dari masing-masing set untuk memastikan proporsinya sesuai dengan yang diminta. Output-nya menunjukkan berapa baris data yang termasuk ke dalam Training**,** Validation, dan Testing.

```
print("\n--- Data Training ---")
print(train_set.head())
print("\n--- Data Validation ---")
print(val_set.head())
print("\n--- Data Testing ---")
print(test_set.head())
```

Fungsi head() menampilkan 5 baris pertama dari setiap bagian dataset. Tujuannya adalah untuk memastikan bahwa setiap subset (Training, Validation, Testing) benar-benar berisi data yang acak dari dataset utama.

**Kesimpulan dan Hasil Implementasi**

Dari latihan mandiri ini dapat disimpulkan bahwa pembagian dataset menjadi *training, validation, dan testing* sangat penting dalam proses *machine learning*. Dengan membagi data seperti ini model dapat belajar pola data dari *training set*, diuji sementara di *validation set* untuk melihat performa selama pelatihan, dan akhirnya dievaluasi secara objektif menggunakan *testing set*. Pembagian yang baik membantu model lebih akurat dan tidak overfitting.

Implementasi latihan ini juga digunakan dalam tahap persiapan data sebelum membangun model *machine learning*. Data yang sudah dibagi bisa langsung dipakai untuk melatih model, menguji hasil model dan memvalidasi model. Dengan cara ini, dapat dipahami bagaimana proses pengelolaan data dilakukan sebelum model diterapkan pada kasus nyata.

**Link GitHub (Praktikum dan Latihan Mandiri)**

https://github.com/revani18/ML_2025_Revani_3AI01/tree/9d5f19a1ad4496e38668bb8061caf51e58c7943d/Praktikum02