

Laporan Praktikum dan Tugas Mandiri Machine Learning



Muhammad Riandana Pranatha - 0110224076

Teknik Informatika, STT Terpadu Nurul Fikri. Depok

0110224076@student.nurulfikri.ac.id

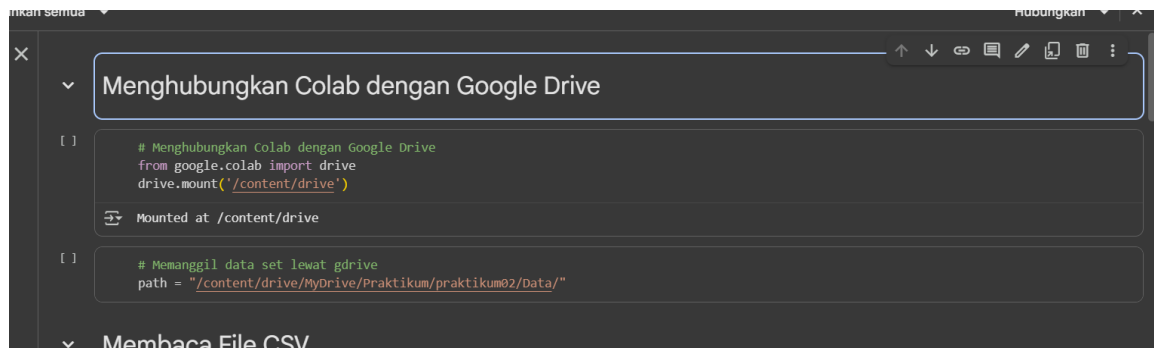
Abstrak

Pada praktikum ini digunakan Google Colab sebagai media utama untuk melakukan eksplorasi data dan penerapan dasar-dasar analisis statistik menggunakan Python. Google Colab dipilih karena berbasis cloud, sehingga tidak perlu instalasi khusus, dan sudah mendukung berbagai library populer seperti Pandas, Matplotlib, serta Scikit-learn. Dataset yang dipakai adalah 500_Person_Gender_Height_Weight_Index.csv dan day.csv, yang kemudian dianalisis melalui perhitungan ukuran pemusatan, penyebaran data, serta dibuat visualisasi untuk memahami pola distribusi dan hubungan antar variabel. Selain itu, dilakukan juga pembagian dataset ke dalam training, validation, dan testing untuk simulasi tahapan awal pembelajaran mesin.

1. Eksperimen Dasar

1.1 Menghubungkan Google Colab dengan Google Drive

Langkah pertama adalah menyambungkan Colab ke akun Google Drive agar file dataset dapat diakses. Setelah menjalankan perintah otorisasi, pengguna akan diminta login dan memberikan izin. Proses ini hanya perlu dilakukan sekali pada setiap sesi Colab.



```
[ ] # Menghubungkan Colab dengan Google Drive
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] # Memanggil data set lewat gdrive
path = "/content/drive/MyDrive/Praktikum/praktikum02/Data/"

Membaca File CSV
```

1.2 Membaca Dataset CSV

Dengan menggunakan library Pandas, dataset dalam format .csv dimuat ke dalam DataFrame. Hal ini memudahkan manipulasi serta analisis data. Perintah head() dipakai untuk menampilkan beberapa baris pertama sehingga struktur data bisa langsung terlihat.

```

Membaca File CSV

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

df = pd.read_csv(path + '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
499 Male 173 131 5
500 rows x 4 columns

```

1.3 Melihat Informasi Dataset

Perintah `info()` menampilkan ringkasan struktur data, seperti jumlah baris, nama kolom, tipe data, serta banyaknya nilai non-null. Dari sini kita bisa mengetahui apakah ada data kosong dan bagaimana karakteristik tipe datanya.

```

[ ] # 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

```

1.4 Menghitung Mean, Median, dan Modus

Statistik deskriptif dasar dihitung untuk memahami distribusi data. Nilai rata-rata (mean) menunjukkan kecenderungan umum, median menggambarkan nilai tengah dari data yang sudah diurutkan, sedangkan modus menampilkan nilai yang paling sering muncul.

```
[ ] # Menghitung mean semua kolom numerik
df['Height'].mean()
np.float64(169.944)

[ ] # Menghitung median semua kolom numerik
df['Height'].median()
170.5

[ ] # Menghitung modus semua kolom numerik (hati-hati karena bisa lebih dari satu)
df['Height'].mode()
Height
0    188
dtype: int64
```

1.5 Variansi dan Standar Deviasi

Ukuran penyebaran data dianalisis menggunakan variansi (var()) dan standar deviasi (std()). Variansi menggambarkan seberapa jauh data menyebar dari rata-rata, sementara standar deviasi lebih mudah diinterpretasikan karena satuannya sama dengan data asli.

```
dtype: int64

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

1.6 Menghitung Kuartil

Fungsi quantile() digunakan untuk mendapatkan kuartil pertama (Q1) dan kuartil ketiga (Q3). Selisih antara keduanya menghasilkan IQR (Interquartile Range), yang sering dipakai untuk mendeteksi outlier atau nilai ekstrem dalam data.

```
[ ] # 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

1.7 Statistik Deskriptif Otomatis

Perintah `describe()` memberikan ringkasan cepat berisi nilai count, mean, std, min, max, serta kuartil. Hal ini sangat membantu untuk melihat gambaran keseluruhan dataset secara instan.

```
[ ] # Untuk membuat statistika deskripsi pada type data int
    df.describe()
```

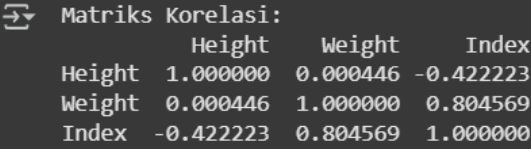
	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

1.8 Menghitung Korelasi

Korelasi Pearson digunakan untuk melihat hubungan linear antar variabel. Nilainya berkisar -1 hingga 1, dengan tanda positif menunjukkan hubungan searah, dan negatif berarti berlawanan arah.

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

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



The output shows a correlation matrix for three variables: Height, Weight, and Index. The matrix is symmetric, with diagonal elements all equal to 1.000000. The correlation between Height and Weight is 0.000446, and between Height and Index is -0.422223. The correlation between Weight and Index is 0.804569.

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

2. Visualisasi Data

2.1 Boxplot

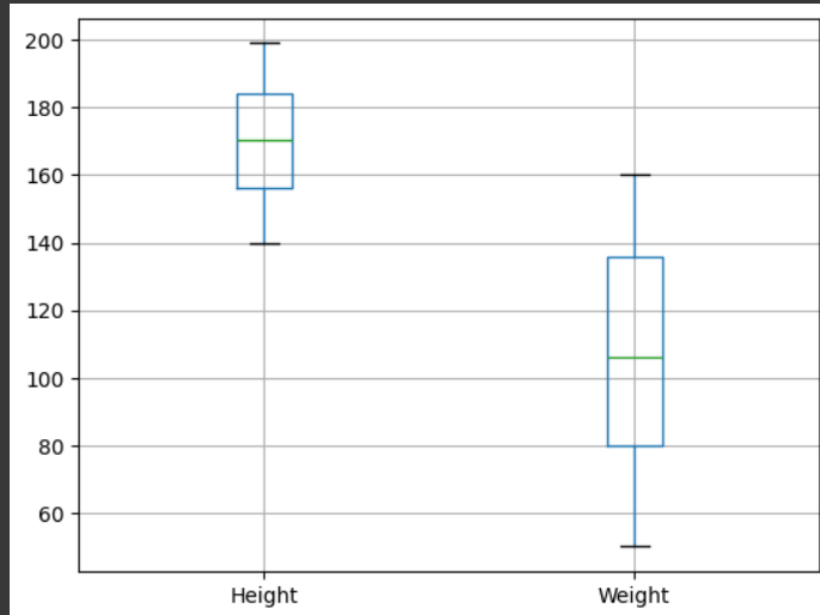
Boxplot digunakan untuk menampilkan distribusi data beserta outlier. Median ditunjukkan oleh garis dalam kotak, sedangkan whiskers menggambarkan rentang nilai normal.

[]

```
import pandas as pd
import numpy as np

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

<Axes: >



2.2 Histogram

Histogram menggambarkan distribusi frekuensi data, dalam hal ini tinggi badan. Data dibagi menjadi beberapa interval (bins), lalu dihitung jumlah data pada tiap interval.

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

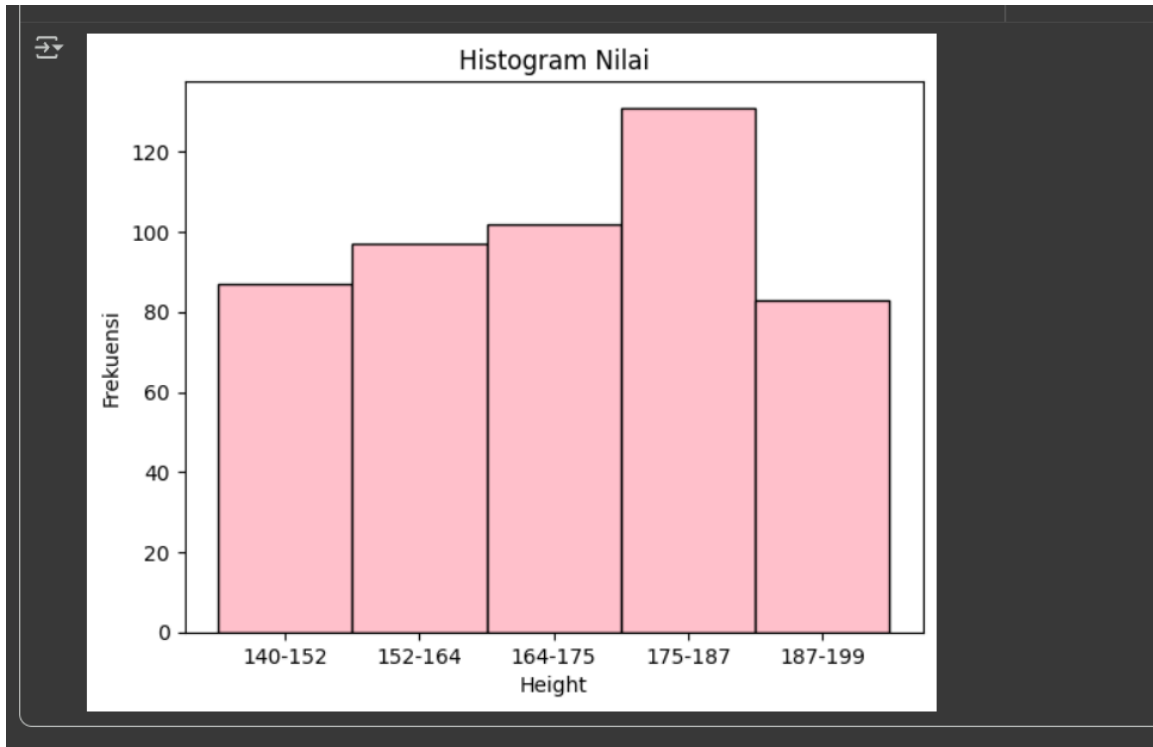
# Ambil data Height
data_height = df["Height"]

# 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()
```



2.3 Scatter Plot – Korelasi Positif

Scatter plot menunjukkan hubungan antara dua variabel numerik. Pada kasus korelasi positif, titik-titik cenderung membentuk pola naik.

[Tambahkan screenshot kode dan output di sini]

2.4 Scatter Plot – Korelasi Negatif

Pada korelasi negatif, peningkatan satu variabel diikuti penurunan variabel lain. Titik-titik akan membentuk pola menurun.


```

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)

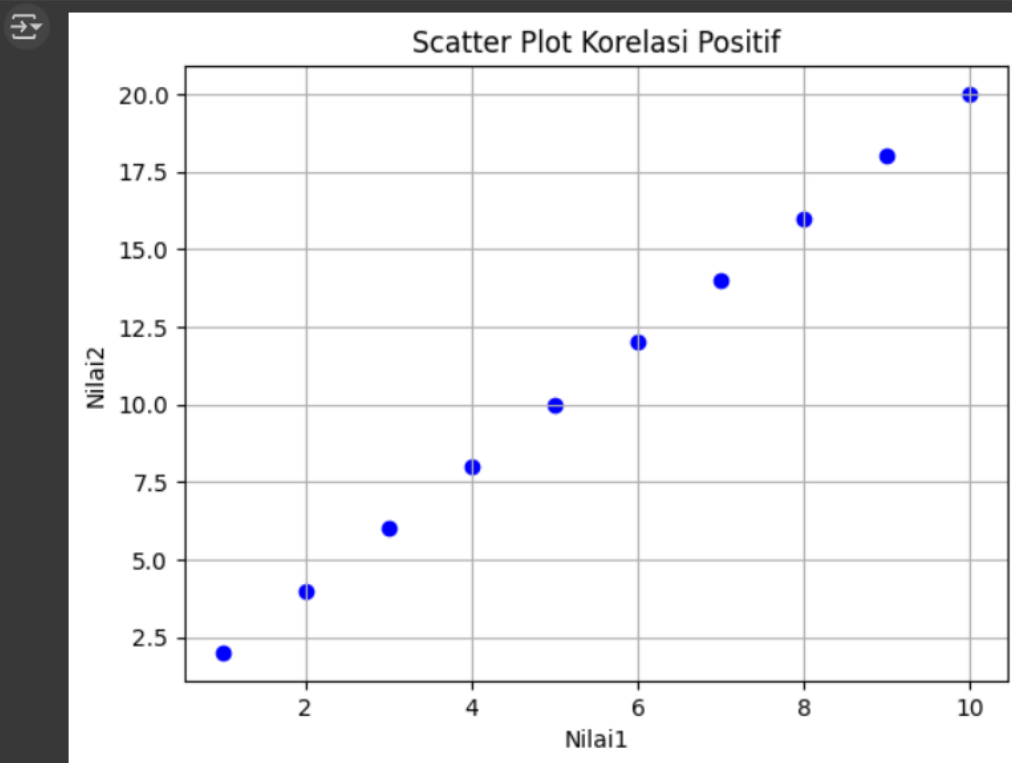
# Buat scatter plot
plt.scatter(df2['Nilai1'], df2['Nilai2'], color='blue', marker='o')

# Tambahkan label
plt.title('Scatter Plot Korelasi Positif')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()

```



3. Tugas Mandiri

3.1 Instruksi

Dataset day.csv dibagi menjadi tiga bagian:

- Data Training (80%)
- Data Validation (10% dari training)
- Data Testing (20%)

3.2 Langkah Penyelesaian

- Mengimpor library Pandas dan Scikit-learn.
- Memuat dataset day.csv ke dalam DataFrame.
- Membagi dataset menjadi training (80%) dan testing (20%) dengan `train_test_split`.
- Dari data training, diambil kembali 10% sebagai validation set.
- Menampilkan jumlah baris serta lima data pertama dari masing-masing set untuk memastikan pembagian sudah benar.

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

# 1. Baca dataset day.csv
df_day = pd.read_csv("/content/drive/MyDrive/Praktikum/praktikum02/Data/day.csv")

# 2. Split data: Training (80%) & Testing (20%)
train_set, test_set = train_test_split(df_day, test_size=0.2, random_state=42)

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

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

# 5. Tampilkan 5 baris pertama dari masing-masing set
print("\nData Training:\n", train_set.head())
print("\nData Validation:\n", val_set.head())
print("\nData Testing:\n", test_set.head())
```

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

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

Kesimpulan

1. Google Colab adalah sarana yang praktis untuk melakukan eksperimen Machine Learning, terutama bagi pemula.
2. Analisis statistik deskriptif membantu memahami sifat data sebelum pemodelan.
3. Visualisasi mempermudah interpretasi distribusi dan hubungan antar variabel.
4. Pembagian dataset menjadi training, validation, dan testing penting untuk evaluasi model.
5. Praktikum ini memberi gambaran awal proses analisis data sebelum masuk ke tahap pemodelan lebih lanjut.

Berikut Link Github dan Google Colab:

Github : <https://github.com/Sakiaihara12/Machine-Learning.git>

Colab : https://colab.research.google.com/drive/1txredeNL_JzFg5DXITe-p5zbyFP8kthx?usp=sharing