

Tugas 2: Machine Learning Statistik Deskriptif dan Probabilitas

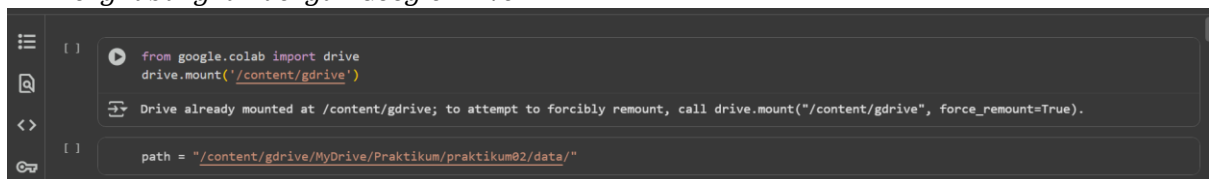
Ahmad Azka Ridha - 0110222062 ¹

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: ahma22062ti@student.nurulfikri.ac.id

1. Praktikum

1.1 menghubungkan dengan Google Drive



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

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

[ ] path = "/content/gdrive/MyDrive/Praktikum/praktikum02/data/"
```

Perintah ini memasang Google Drive ke Colab di `/content/gdrive` menggunakan `drive.mount()`, dengan opsi `force_remount=True` untuk remount jika sudah terpasang, dan mendefinisikan `path` ke folder data spesifik. 2. Contoh Lain Heading - **replace or delete**

1.2 Membaca File CSV

```
[ ] import pandas as pd

[ ] df = pd.read_csv(path + '/500_Person_Gender_Height_Weight_Index (1).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

Perintah ini menggunakan library Pandas untuk membaca file CSV bernama "500_Person_Gender_Height_Weight_Index (1).csv" dari direktori yang ditentukan oleh variabel `path`. Data dimuat ke dalam DataFrame `df` yang berisi 500 baris dan 4 kolom: `Gender`, `Height` (tinggi badan), `Weight` (berat badan), dan `Index`. Data ini mencakup informasi individu seperti jenis kelamin, tinggi, berat, dan indeks terkait.

1.3 Melihat Informasi Umum Data

```
[ ] 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
```

Perintah `df.info()` menampilkan ringkasan informasi DataFrame `df`, termasuk 500 entri dengan indeks 0 hingga 499. Terdapat 4 kolom: `Gender` (objek), `Height` (int64), `Weight` (int64), dan `Index` (int64), semuanya tanpa nilai null. Total penggunaan memori adalah 15.8 KB.

1.4 Menghitung Nilai-Nilai Sentral

```
[ ] df['Height'].mean()
np.float64(169.944)

[ ] df['Height'].median()
170.5

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

1.5 Menghitung Ukuran Persebaran

```
[ ] df.var(numeric_only=True)
```

	0
Height	268.149162
Weight	1048.633267
Index	1.836168

dtype: float64

```
[ ] df.std(numeric_only=True)
```

	0
Height	16.375261
Weight	32.382607
Index	1.355053

dtype: float64

Perintah `df.var(numeric_only=True)` menghitung varians untuk kolom numerik dalam DataFrame `df`, yaitu `Height` (268.149162), `Weight` (1048.633267), dan `Index` (1.836168), dengan tipe data float64. Perintah `df.std(numeric_only=True)` menghitung standar deviasi untuk kolom yang sama, yaitu `Height` (16.375261), `Weight` (32.382607), dan `Index` (1.355053), juga dengan tipe data float64.

1.6 Menghitung Kuartil

```
[ ] q1 = df['Height'].quantile(0.25)
q3 = df['Height'].quantile(0.75)
iqr = q3 - q1
print(q1, q3, iqr)
```

156.0 184.0 28.0

Perintah ini menghitung kuartil dan rentang antar kuartil (IQR) untuk kolom `Height` dalam DataFrame `df`. `q1` adalah kuartil pertama (25%, bernilai 156.0), `q3` adalah kuartil ketiga (75%, bernilai 184.0), dan `iqr` adalah IQR ($q3 - q1$, bernilai 28.0), yang kemudian ditampilkan.

1.7 Menghitung Statistik Deskriptif Otomatis

```
[ ] 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

Perintah `df.describe()` memberikan ringkasan statistik untuk kolom numerik dalam DataFrame `df`. Untuk `Height`: rata-rata 169.94, standar deviasi 16.38, rentang 140 hingga 199, dengan kuartil 25% (156), 50% (170), dan 75% (184). Untuk `Weight`: rata-rata 106.0, standar deviasi 32.38, rentang 50 hingga 160, dengan kuartil 25% (80), 50% (106), dan 75% (136). Untuk `Index`: rata-rata 3.74, standar deviasi 1.36, rentang 0 hingga 5, dengan kuartil 25% (3), 50% (4), dan 75% (5). Semua kolom memiliki 500 entri tanpa nilai null.

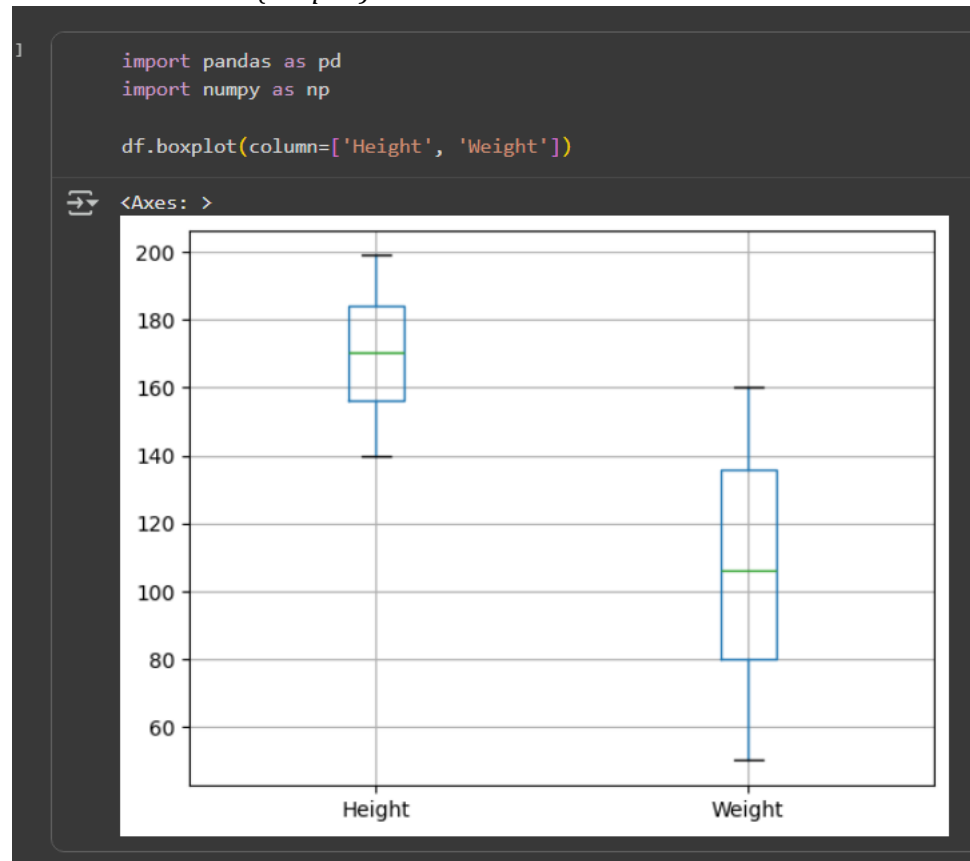
1.8 Menghitung Korelasi

```
[ ] correlation_matrix = df.corr(numeric_only=True)

print("Matrix Korelasi:")
print(correlation_matrix)
```

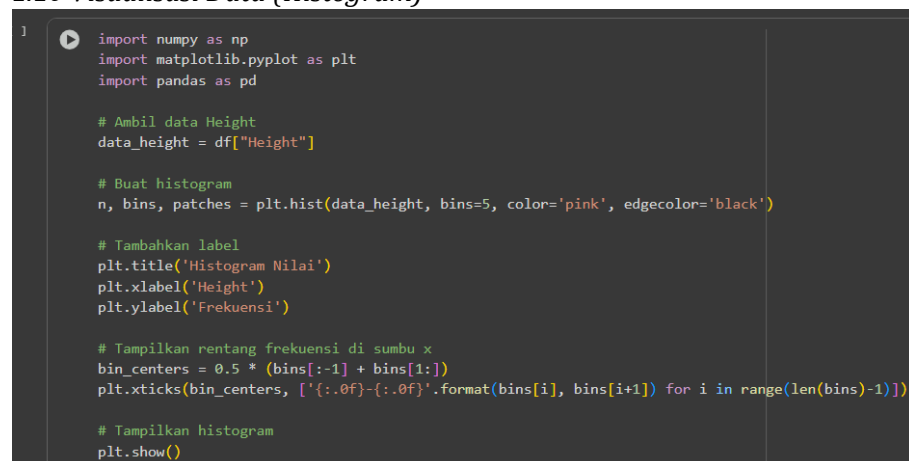
	Height	Weight	Index
Height	1.000000	0.000446	-0.422223
Weight	0.000446	1.000000	0.804569
Index	-0.422223	0.804569	1.000000

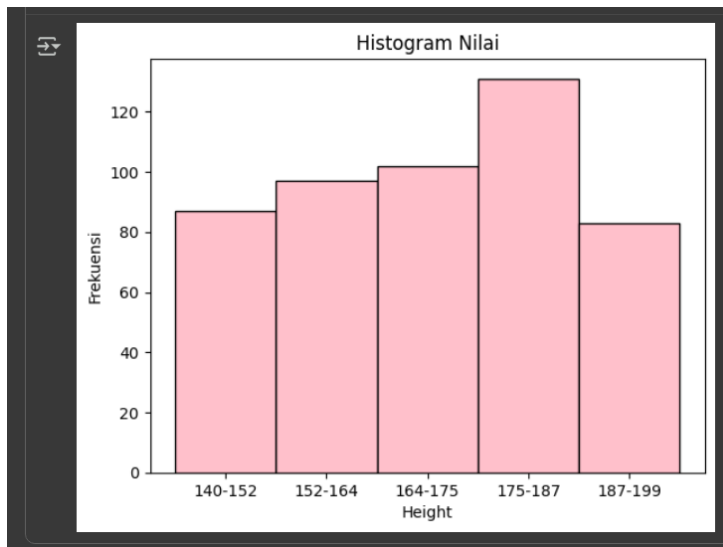
1.9 Visualisasi Data (Boxplot)



Perintah ini menggunakan `df.boxplot(column=['Height', 'Weight'])` untuk membuat diagram kotak (boxplot) yang menampilkan distribusi data tinggi badan ('Height') dan berat badan ('Weight') dari DataFrame `df`. Diagram menunjukkan median, kuartil, dan rentang data, dengan outlier ditandai sebagai titik di luar whisker.

1.10 Visualisasi Data (Histogram)





1.11 Visualisasi Data (Scatter Plot)

```
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]
}

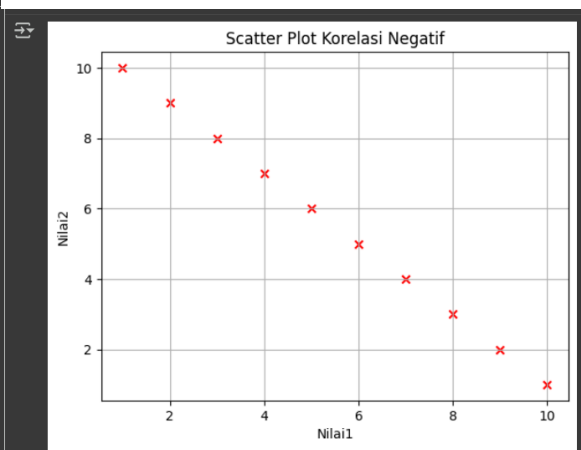
df3 = pd.DataFrame(data)

# 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()
```



```
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]
}

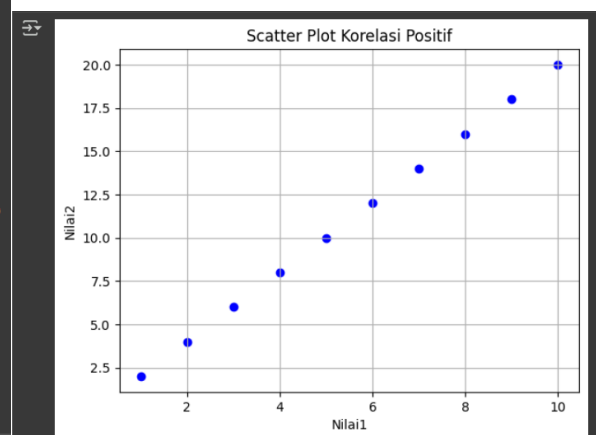
df2 = pd.DataFrame(data)

# 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()
```



2. Tugas Praktikum Mandiri

2.1 Mount Google Drive

```
[1]
✓ 45s # Step 1: Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

2.2 Import Library

```
[2]
✓ 3s # Step 2: Import library
import pandas as pd
from sklearn.model_selection import train_test_split
```

2.3 Baca Dataset dari Google Drive

```
[3]
✓ 0s # Step 3: Baca dataset dari Google Drive
# Ganti path sesuai lokasi file Anda di Google Drive
df = pd.read_csv('/content/drive/MyDrive/Praktikum Mandiri ML/Praktikum 2/Data/day.csv')

print("Total Data:", len(df))
print("\n5 Data Teratas:")
print(df.head())

Total Data: 731

5 Data Teratas:
   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
4   1600
```


2.4 Split Dataset

```
[4]
✓ Os # Step 4: Split dataset
      # 80% Training, 20% Testing
      train, test = train_test_split(df, test_size=0.2, random_state=42)

[5]
✓ Os # Dari training, ambil 10% lagi untuk validation
      train, val = train_test_split(train, test_size=0.1, random_state=42)
```

2.5 Tampilkan Jumlah dan 5 Baris Teratas

```
[7]
✓ Os # Step 5: Tampilkan jumlah dan 5 baris teratas
print("\n--- Data Training ---")
print("Jumlah data:", len(train))
print(train.head())
```

--- Data Training ---
Jumlah data: 525

	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

[10]
✓ 0s

```
print("\n--- Data Validation ---")  
print("Jumlah data:", len(val))  
print(val.head())
```



--- Data Validation ---

Jumlah data: 59

	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

[9]
✓ 0s

```
print("\n--- Data Testing ---")  
print("Jumlah data:", len(test))  
print(test.head())
```



--- Data Testing ---

Jumlah data: 147

	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