Praktikum 2 – Machine Learning
Statistik Deskriptif dan Probabilitas

Prepared By:
Dr. Sirojul Munir S.Si,M.Kom
Diah Ayu Puspasari

Tujuan :

1. Mengenal lingkungan pengembangan Google Colab untuk praktikum Machine Learning
2. Memahami konsep dasar statistik deskriptif dan probabilitas dalam konteks analisis data.
3. Mampu melakukan perhitungan statistik dasar menggunakan library Pandas di Google Colab.
4. Mampu memvisualisasikan distribusi dan hubungan antar variabel menggunakan Matplotlib dan Seaborn.

Dateline : 1 Pekan

Gitlab/Github :

Branch Repository : [PRODI ROMBEL] _[NAMASINGKAT]_[NIM] (contoh: ti01_budi_0110112001)

Aturan Pengerjaan:

1. Gunakan text editor yang nyaman bagi anda
2. Diperkenankan mengerjakan langsung bagi yang sudah memahami dan menguasai materi
3. Dilarang melakukan tindakan plagiarism (asisten lab akan mengecek hasil pekerjaan)
    a. 1x nilai praktikum terkait bernilai 0
    b. 2x nilai matakuliah pemrograman web E
    c. 3x mahasiswa akan di sidang komite etik kampus

2.1 Lingkungan Pengembangan ML - Google Colab

Pada praktikum kali ini akan diperkenalkan lingkungan pengembangan pemodelan machine learning menggunakan platform aplikasi Google Colab. Platform Google Colab memungkinkan Anda menulis dan mengeksekusi Python di browser, dengan tidak memerlukan konfigurasi, akses tanpa biaya ke GPU dan dapat berbagi dengan mudah.

1. Direktori Program

Lakukan langkah-langkah berikut pada direktori untuk membuat struktur direktori program python machine learning.

1. Gunakan akun Google masing-masing.
2. Buat struktur folder praktikum_ml/praktikum02 di Google Drive dengan struktur direktori berikut ini:

praktikum

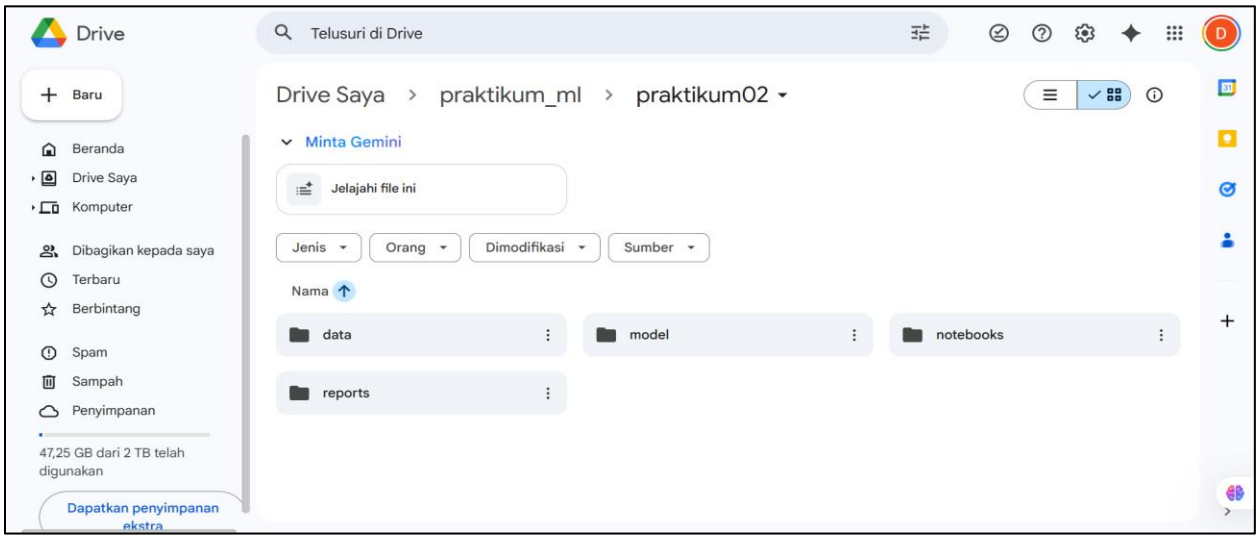
├— praktikum02/

├— data/

├— notebooks/

├— models/

└— reports/



3. Berikut penjelasan isi folder program machine learning.
   a. Semua coding dikerjakan di dalam folder notebooks/ dan buat file kode program python dengan nama file praktikum02.ipynb
   b. Dataset mentah diletakkan pada folder data/
   c. Model hasil training disimpan di folder models/ (format. pkl atau .joblib).
   d. Laporan/visualisasi disimpan di folder reports/.
4. Upload hasil akhir (notebook dan laporan) ke GitHub sesuai format branch repository.

2. Tutorial Penggunaan Google Colab

1. Membuat Notebook di Colab

- Buka: https://colab.research.google.com
- Pilih New Notebook In Drive → simpan di praktikum_ml/praktikum02/notebooks/.

2. Menjalankan Cell

- Gunakan Code Cell untuk Python, Text Cell (Markdown) untuk catatan.
- Shortcut: Shift+Enter untuk run cell.

3. Menghubungkan dengan Google Drive

Sel ini berfungsi untuk menghubungkan lingkungan Google Colab dengan akun Google Drive-mu. Setelah kode ini dijalankan, akan muncul tautan otorisasi. Kamu harus mengklik tautan tersebut, memilih akun Google, dan memberikan izin agar Colab bisa mengakses file-file yang tersimpan di Google Drive-mu. Proses ini hanya perlu dilakukan satu kali per sesi.

```
# menghubungkan colab dengan google drive
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
# memanggil data set lewat gdrive
path = "/content/gdrive/MyDrive/praktikum_ml/praktikum02/"
```

4. Membaca File CSV:

Sel ini menggunakan library **Pandas** untuk membaca file data. Variabel path menyimpan lokasi folder di Google Drive tempat file 500_Person_Gender_Height_Weight_Index.csv berada. Fungsi pd.read_csv() kemudian membaca file tersebut dan menyimpannya ke dalam sebuah **DataFrame** bernama df.

```
# 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
499	Male	173	131	5

500 rows × 4 columns

Menjalankan df.head() di akhir sel akan menampilkan lima baris pertama dan terakhir dari data. Akan menampilkan 500 baris dan 4 kolom dari dataset tersebut, yaitu 'Gender', 'Height', 'Weight', dan 'Index'. Ini memberikan gambaran awal tentang struktur data.

2.2 Statistik Deskriptif dan Probabilitas Dasar

Sebelum membangun model *machine learning*, langkah pertama yang krusial adalah memahami data. Materi praktikum ini akan memperkenalkan dua fondasi utama dalam analisis data: Statistik Deskriptif dan Probabilitas Dasar.

Analisis Statistik Deskriptif

1. **Melihat Informasi Umum Data:**

```
# 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
```

Metode .info() memberikan ringkasan singkat tentang DataFrame. Ini sangat penting untuk langkah awal analisis data karena menampilkan informasi berikut:

- Jumlah baris **500 entri** dan **4 kolom**
- Nama-nama kolom.
- Jumlah data non-null (data yang tidak kosong) per kolom.
- Tipe data (Dtype) dari setiap kolom, seperti int64 (integer) dan object (biasanya string).

2. **Menghitung Nilai-Nilai Sentral (Mean, Median, Modus):**

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

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

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

```
170.5
```

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

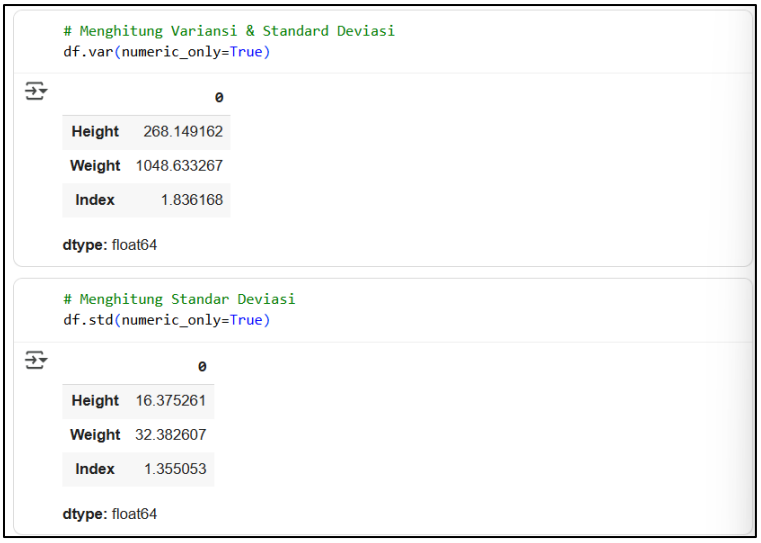
	Height
0	188

dtype: int64

Kode ini secara spesifik menghitung nilai-nilai sentral:

- .mean(): Rata-rata dari setiap kolom numerik.
- .median(): Nilai tengah dari setiap kolom numerik.
- .mode().iloc[0]: Modus atau nilai yang paling sering muncul.

3. **Menghitung Ukuran Persebaran (Variansi & Standar Deviasi):**



```
# Menghitung Variansi & Standard 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)
```

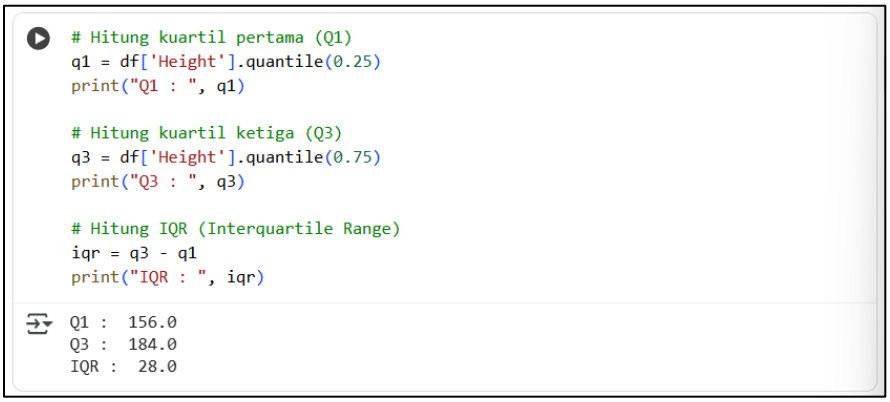
	0
Height	16.375261
Weight	32.382607
Index	1.355053

dtype: float64

Sel ini menghitung metrik untuk melihat seberapa besar persebaran atau variasi data:

- .var(): Variansi, yang mengukur seberapa jauh setiap titik data dari rata-rata.
- .std(): Standar deviasi, akar kuadrat dari variansi, yang memberikan ukuran persebaran dalam satuan yang sama dengan data aslinya.

4. **Menghitung Kuartil**



```
# 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
```

df['Height'].quantile(0.25) dan df['Height'].quantile(0.75) menghitung kuartil pertama (Q1) dan kuartil ketiga (Q3). IQR dihitung dengan mengurangkan Q3 dengan Q1.

5. **Menghitung Statistik Deskriptif Otomatis:**

```
# 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

Metode .describe() secara otomatis menghitung statistik deskriptif dasar untuk semua kolom numerik. Ini memberikan gambaran cepat tentang distribusi data.

Hasilnya adalah sebuah tabel yang berisi metrik-metrik berikut untuk setiap kolom numerik:

- count: Jumlah data non-null.
- mean: Rata-rata.
- std: Standar deviasi.
- min: Nilai minimum.
- 25%, 50%, 75%: Kuartil pertama, median, dan kuartil ketiga.
- max: Nilai maksimum.

6. **Menghitung Korelasi:**

```
# 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
```

Kode di bawah ini akan menghitung koefisien korelasi Pearson antara semua kolom numerik dalam DataFrame df. Koefisien ini menunjukkan seberapa kuat hubungan linear antar dua variabel. Nilai korelasi berkisar dari -1 hingga 1.

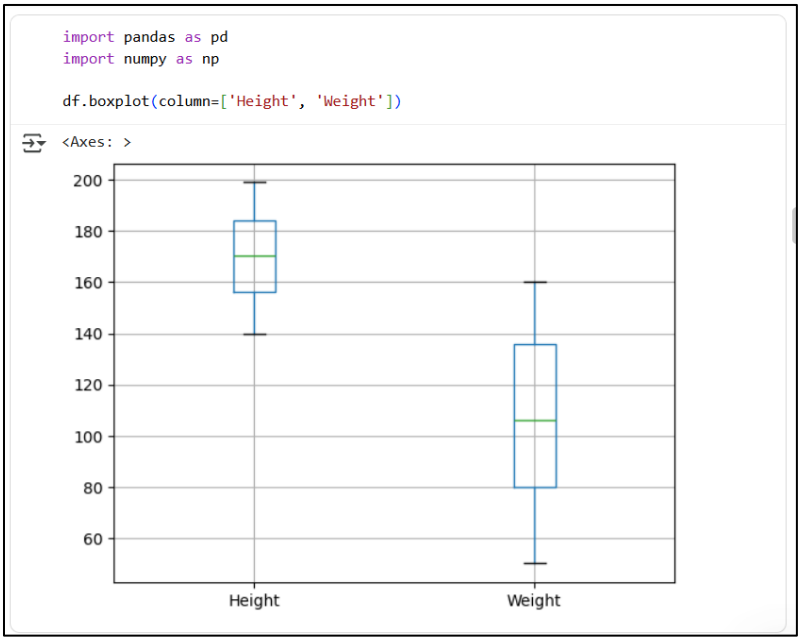
- Nilai positif (mendekati 1) menunjukkan korelasi positif, di mana satu variabel naik saat variabel lain juga naik.
- Nilai negatif (mendekati -1) menunjukkan korelasi negatif, di mana satu variabel naik saat variabel lain turun.
- Nilai nol (mendekati 0) menunjukkan tidak ada hubungan linear.

Visualisasi Data

1. **Boxplot:**

```
import pandas as pd
import numpy as np

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

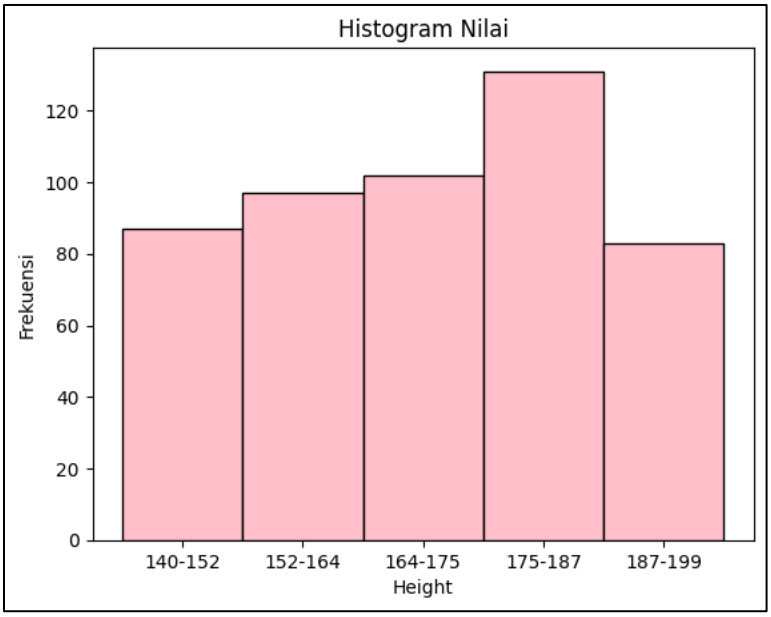
```
<Axes: >
```



Sel ini menggunakan fungsi df.boxplot() untuk membuat visualisasi boxplot. Gambar yang dihasilkan menampilkan boxplot untuk kolom 'Height' dan 'Weight'. Garis tengah di dalam kotak menunjukkan median, dan kotak itu sendiri menunjukkan Interquartile Range (IQR). Garis-garis horizontal di ujung (whiskers) menunjukkan batas data normal.plt.suptitle(): Memberikan judul utama untuk keseluruhan plot.

2. **Histogram**

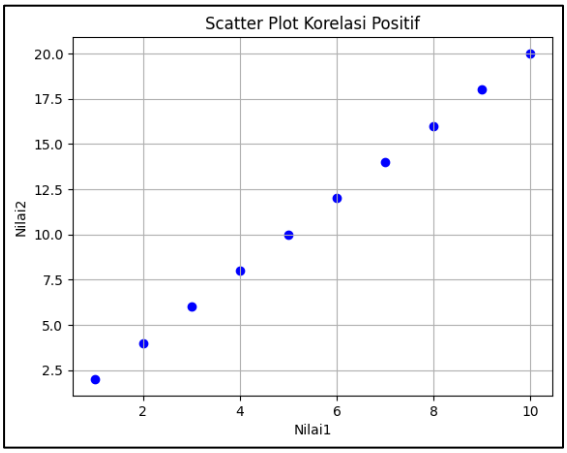
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# 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

# Tampilkan histogram
plt.show()
```



Sel ini membuat histogram untuk kolom 'Height'. Histogram adalah grafik yang menunjukkan distribusi frekuensi data. Outputnya adalah histogram dengan lima batang (bins=5). Kamu bisa melihat frekuensi atau jumlah data yang jatuh dalam setiap rentang tinggi badan.

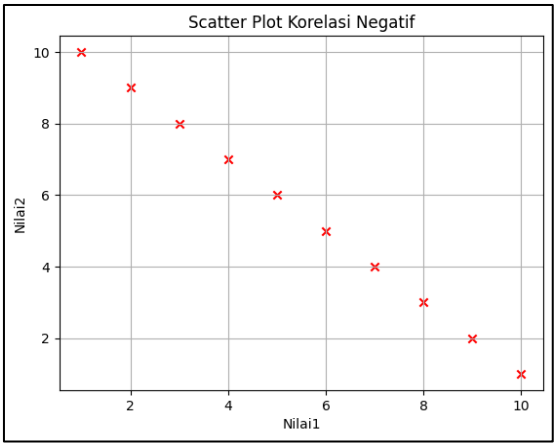
3. Scatter Plot (Hubungan Antar Variabel):

```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 Korelasi Positif')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()
```



Sel ini membuat ***scatter plot*** atau diagram pencar untuk memvisualisasikan hubungan antara dua variabel numerik. Berisi dua blok kode untuk membuat dua *scatter plot* terpisah. Kode pertama menggunakan data Nilai1 dan Nilai2 yang memiliki **korelasi positif**, sedangkan kode kedua memiliki **korelasi negatif**.

```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()
```



Hasil:

- **Plot Korelasi Positif:** Titik-titik data membentuk pola yang naik ke atas. Ini menunjukkan bahwa ketika Nilai1 meningkat, Nilai2 juga cenderung meningkat.
- **Plot Korelasi Negatif:** Titik-titik data membentuk pola yang menurun. Ini menunjukkan bahwa ketika Nilai1 meningkat, Nilai2 justru cenderung menurun.

Tugas Praktikum Mandiri

1. Praktikan semua kode program di Modul ini dan kumpulkan sebagai praktikum Pekan 2.
2. Buat program untuk membagi dataset day.csv menjadi tiga bagian, yaitu:

   (a) Data Training: 80% dari total dataset

   (b) Data Validation: 10% dari data training

   (c) Data Testing: 20% dari total dataset

   Tampilkan jumlah data dan 5 baris data teratas untuk setiap set (Training, Validation, dan Testing) sebagai bukti pengerjaan.