

Praktikum 2 dan Tugas Mandiri 2 :

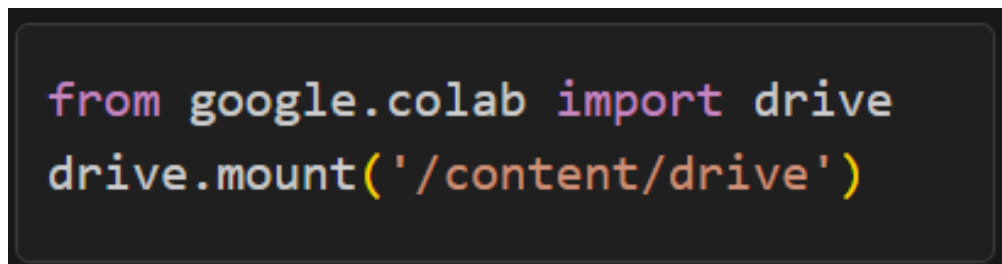
Raffa Yuda Pratama - 0110224081

Teknik Informatika, STT Terpadu Nurul Fikri, Depok
E-mail: 0110224081@student.nurulfikri.ac.id

Abstract. Laporan ini menyajikan implementasi dua tugas Machine Learning: analisis statistik deskriptif dengan visualisasi data dan pembagian dataset. Praktikum 2 menganalisis dataset 500_Person_Gender_Height_Weight_Index.csv melalui perhitungan statistik deskriptif dan visualisasi menggunakan boxplot, histogram, dan scatter plot. Tugas Mandiri 2 membagi dataset day.csv menjadi training (72%), validation (8%), dan testing (20%) menggunakan scikit-learn. Kedua implementasi berhasil dan memberikan foundation untuk pengembangan model machine learning.

1. Praktikum 2

1.1. Mount Google Drive

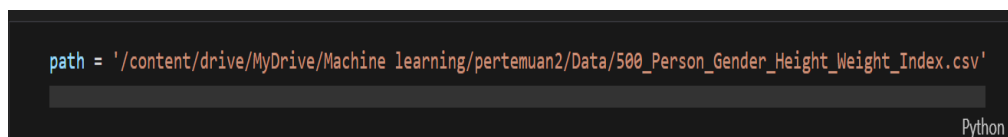


```
from google.colab import drive
drive.mount('/content/drive')
```

Gambar 1 Mount Google Drive

Cell ini berfungsi untuk menghubungkan Google Drive dengan environment Google Colab. Proses mounting diperlukan agar dapat mengakses file-file yang tersimpan di Google Drive. Fungsi `drive.mount()` akan meminta otorisasi untuk mengakses Google Drive dan membuat link antara direktori `/content/drive` di Colab dengan Google Drive pengguna. Setelah proses ini berhasil, semua file di Google Drive dapat diakses melalui path yang dimulai dengan `/content/drive/MyDrive/`.

1.2. Mendefinisikan Path Dataset



```
path = '/content/drive/MyDrive/Machine learning/pertemuan2/Data/500_Person_Gender_Height_Weight_Index.csv'
```

Python

Gambar 2 Set Path File

Cell ini mendefinisikan variabel `path` yang berisi lokasi lengkap file dataset yang akan dianalisis. Dataset yang digunakan adalah file CSV dengan nama **500_Person_Gender_Height_Weight_Index.csv** yang berisi data 500 orang dengan informasi gender, tinggi badan, berat badan, dan indeks. Penggunaan variabel `path` memudahkan dalam pengelolaan dan memungkinkan perubahan lokasi file tanpa mengubah kode di banyak tempat.

1.3. Import Library dan Load Dataset

```
import pandas as pd
df = pd.read_csv(path)
df.head()
```

Gambar 3 Load dan Preview Data

Cell ini melakukan tiga operasi penting: pertama, mengimpor library pandas yang merupakan library fundamental untuk manipulasi dan analisis data dalam Python. Kedua, membaca file CSV menggunakan fungsi **pd.read_csv()** dan menyimpannya dalam DataFrame yang diberi nama **df**. DataFrame adalah struktur data dua dimensi yang mirip dengan tabel spreadsheet. Ketiga, menggunakan method **df.head()** untuk menampilkan 5 baris pertama dari dataset sebagai preview awal untuk memahami struktur dan isi data.

1.4. Melihat Informasi Umum Data

```
df.info
```

Gambar 4 df.info()

Cell ini dimaksudkan untuk menampilkan informasi umum tentang dataset, namun terdapat kesalahan sintaks. Kode ini hanya mereferensi method **info** tanpa memanggilnya dengan tanda kurung. Method **df.info()** seharusnya digunakan untuk mendapatkan ringkasan informasi dataset seperti jumlah baris, jumlah kolom, nama kolom, tipe data setiap kolom, jumlah nilai non-null, dan penggunaan memori. Informasi ini sangat penting untuk memahami karakteristik dataset sebelum melakukan analisis lebih lanjut.

1.5. Menghitung Nilai Sentral (Mean, Median, Modus)

```
df['Height'].mean()  
df['Height'].median()  
df['Height'].mode()
```

Gambar 5 Menghitung Nilai Sentral

cell ini menghitung ukuran tendensi sentral untuk kolom 'Height' (tinggi badan). Mean atau rata-rata memberikan nilai pusat dari distribusi data dengan menjumlahkan semua nilai dan membaginya dengan jumlah observasi. Median adalah nilai tengah ketika data diurutkan dari terkecil ke terbesar, yang lebih robust terhadap outlier dibandingkan mean. Modus adalah nilai yang paling sering muncul dalam dataset. Ketiga ukuran ini memberikan perspektif berbeda tentang karakteristik sentral data tinggi badan dalam dataset.

1.6. Analisis Ukuran Persebaran Data

```
df.var(numeric_only=True)  
df.std(numeric_only=True)
```

Gambar 6 Analisis Ukuran Persebaran Data

Cell ini menghitung ukuran persebaran atau variabilitas data untuk semua kolom numerik dalam dataset. Variansi mengukur seberapa jauh data tersebar dari rata-rata dengan menghitung rata-rata kuadrat selisih setiap nilai dari mean. Standar deviasi adalah akar kuadrat dari variansi dan memiliki satuan yang sama dengan data asli, sehingga lebih mudah diinterpretasi. Parameter **numeric_only=True** memastikan bahwa perhitungan hanya dilakukan pada kolom yang berisi data numerik, mengabaikan kolom kategorikal seperti gender.

1.7. Analisis Kuartil dan IQR

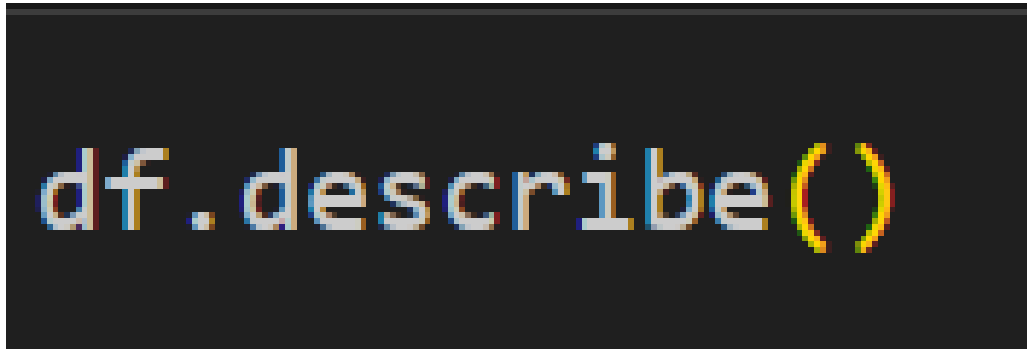
```
q1 = df['Height'].quantile(0.25) # Kuartil pertama
q3 = df['Height'].quantile(0.75) # Kuartil ketiga
iqr = q3 - q1                    # Interquartile Range

print(f'Q1: {q1}')
print(f'Q3: {q3}')
print(f'IQR: {iqr}')
```

Gambar 7 Analisis Kuartil dan IQR

Cell ini menghitung kuartil dan Interquartile Range (IQR) untuk kolom Height. Q1 (kuartil pertama) adalah nilai yang membagi 25% data terendah, sedangkan Q3 (kuartil ketiga) membagi 75% data terendah. IQR adalah selisih antara Q3 dan Q1 yang mengukur persebaran 50% data tengah. IQR berguna untuk mengidentifikasi outlier dan memahami variabilitas data tanpa terpengaruh nilai ekstrem.

1.8. Statistik Deskriptif Komprehensif



Gambar 8 method describe()

Method **describe()** menghasilkan ringkasan statistik deskriptif yang komprehensif untuk semua kolom numerik dalam dataset. Output mencakup count (jumlah observasi valid), mean (rata-rata), std (standar deviasi), min (nilai minimum), 25% (Q1), 50% (median/Q2), 75% (Q3), dan max (nilai maksimum). Informasi ini memberikan gambaran menyeluruh tentang distribusi data dan membantu mengidentifikasi anomali, outlier, atau karakteristik khusus dari setiap variabel numerik dalam dataset.

1.9. Analisis Korelasi Antar Variabel

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

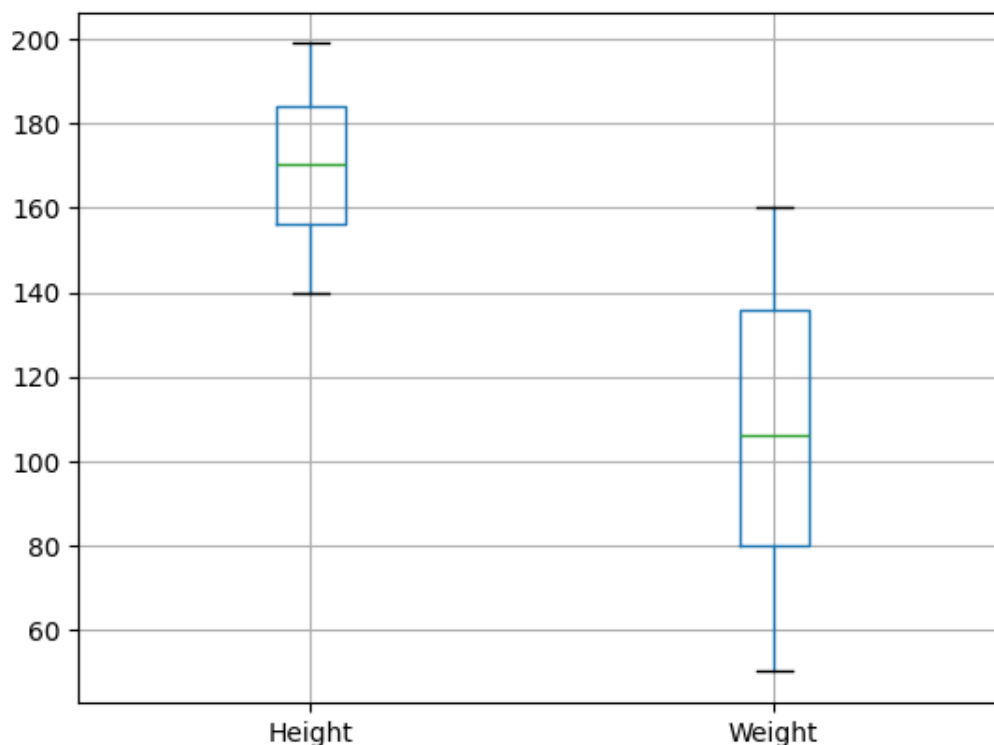
print("Matriks Korelasi : ")
print(correlation_matrix)
```

Cell ini menghitung dan menampilkan matriks korelasi Pearson untuk semua pasangan variabel numerik dalam dataset. Korelasi mengukur kekuatan dan arah hubungan linear antara dua variabel dengan nilai berkisar dari -1 hingga +1. Nilai mendekati +1 menunjukkan korelasi positif kuat, nilai mendekati -1 menunjukkan korelasi negatif kuat, dan nilai mendekati 0 menunjukkan tidak ada hubungan linear. Matriks korelasi sangat penting dalam analisis eksploratori untuk memahami hubungan antar variabel dan dapat membantu dalam pemilihan fitur untuk model machine learning.

1.10. Visualisasi Boxplot

```
import matplotlib.pyplot as plt
df.boxplot(column=['Height', 'Weight'])
```

Gambar 9 Kode Boxplot



Gambar 10 Visualisasi Boxplot

Cell ini membuat visualisasi boxplot untuk kolom Height dan Weight menggunakan library matplotlib. Boxplot adalah diagram yang menunjukkan distribusi data melalui kuartil dan membantu mengidentifikasi outlier secara visual. Box menunjukkan IQR (dari Q1 ke Q3), garis di tengah box menunjukkan median, whiskers menunjukkan rentang data normal, dan titik-titik di luar whiskers menunjukkan outlier. Visualisasi ini sangat efektif untuk membandingkan distribusi beberapa variabel secara bersamaan dan mengidentifikasi nilai-nilai yang tidak biasa.

1.11. Visualisasi Histogram

```
import matplotlib.pyplot as plt

data_height = df['Height']

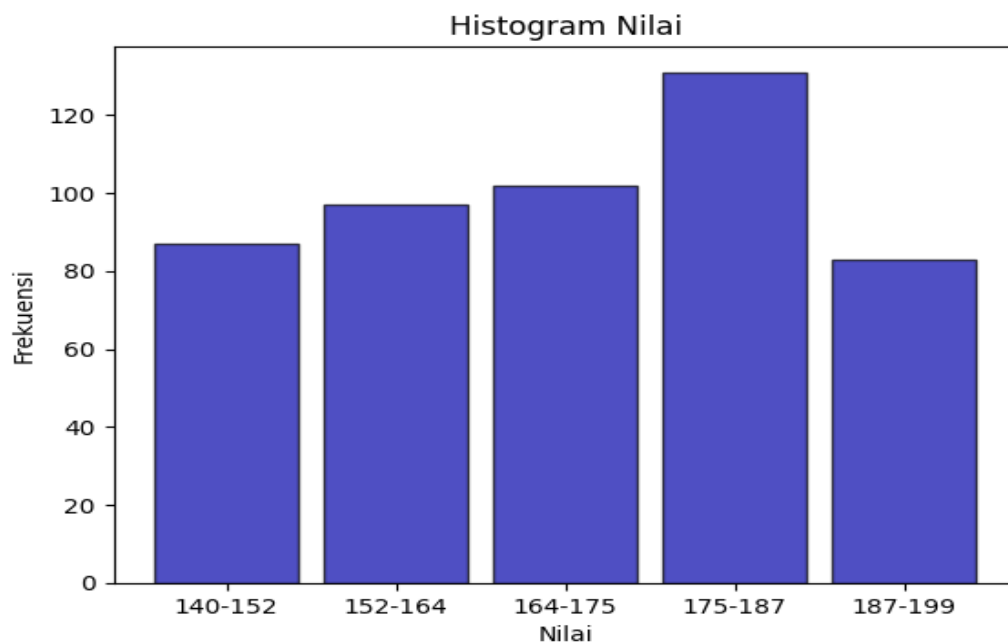
# buat histogram
n, bins, patches = plt.hist(
    x=data_height,
    bins=5,
    color='#0504aa',
    alpha=0.7,
    rwidth=0.85,
    edgecolor='black'
)

# Tambahkan Label
plt.title('Histogram Nilai')
plt.xlabel('Nilai')
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()
```

Gambar 11 Kode Histogram



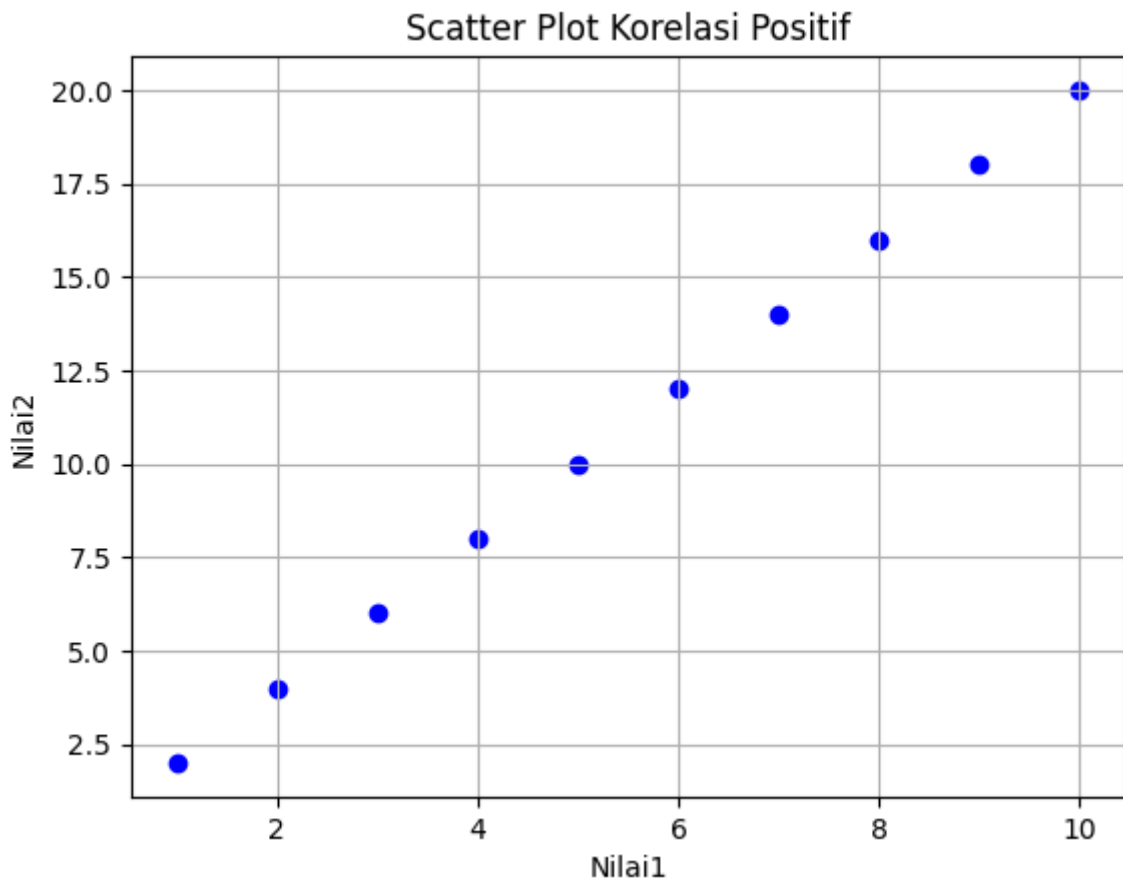
Gambar 12 Visualisasi Histogram

Cell ini membuat histogram untuk memvisualisasikan distribusi frekuensi data tinggi badan. Histogram dibagi menjadi 5 bin (interval) dengan warna biru gelap dan transparansi 70%. Fungsi histogram mengembalikan nilai frekuensi (n), batas-batas bin (bins), dan patches untuk styling. Kode ini juga menambahkan label pada sumbu x yang menunjukkan rentang nilai untuk setiap bin, sehingga memudahkan interpretasi.

1.12. Visualisasi Scatter Plot

```
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.xlabel('Nilai1')  
plt.ylabel('Nilai2')  
plt.title('Scatter Plot Korelasi Positif')  
  
# tambahkan grid  
plt.grid(True)  
  
# tampilkan scatter plot  
plt.show()
```

Gambar 13 Kode Scatter Plot



Gambar 14 Visualisasi Scatter Plot

Cell ini membuat scatter plot menggunakan data dummy untuk mendemonstrasikan korelasi positif sempurna. Data dummy sengaja dibuat dengan hubungan linear sempurna ($\text{Nilai2} = 2 \times \text{Nilai1}$) untuk menunjukkan contoh korelasi positif dengan nilai $r = 1$. Scatter plot menampilkan hubungan antara dua variabel kontinu dengan setiap titik mewakili satu observasi. Grid ditambahkan untuk memudahkan pembacaan koordinat titik. Dalam konteks analisis data yang sebenarnya, scatter plot biasanya dibuat menggunakan variabel dari dataset asli untuk mengeksplorasi hubungan antar variabel secara visual.

2. Tugas Mandiri 2

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

df = pd.read_csv("../Data/day.csv")

train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)

train_df, val_df = train_test_split(train_df, test_size=0.1, random_state=42)

print("Jumlah total data:", len(df))
print("Jumlah data Training:", len(train_df), "(80% dari total)")
print("Jumlah data Validation:", len(val_df), "(10% dari Training)")
print("Jumlah data Testing:", len(test_df), "(20% dari total)")

print("\n=== Data Training (5 baris teratas) ===")
print(train_df.head())

print("\n=== Data Validation (5 baris teratas) ===")
print(val_df.head())

print("\n=== Data Testing (5 baris teratas) ===")
print(test_df.head())
```

Gambar 15 Tugas Mandiri 2

2.1. Import Library yang diperlukan

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

Gambar 16 Import Library

Mengimpor library pandas untuk manipulasi data dan fungsi **train_test_split** dari scikit-learn untuk membagi dataset secara random dengan proporsi yang ditentukan. Library ini dipilih karena memberikan kontrol yang baik terhadap proses pembagian data dan memiliki fitur **random_state** untuk reproducibility.

2.2. Pembagian Dataset Pertama

```
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)
```

Gambar 17 Pembagian Dataset 1

Melakukan pembagian pertama dengan memisahkan 20% data untuk testing set dan 80% sisanya untuk kombinasi training dan validation. Parameter **test_size=0.2** menentukan proporsi data testing, sedangkan **random_state=42** memastikan hasil pembagian yang konsisten setiap kali program dijalankan. Nilai 42 dipilih sebagai seed untuk random number generator.

2.3. Pembagian Dataset Tahap Kedua

```
train_df, val_df = train_test_split(train_df, test_size=0.1, random_state=42)
```

Gambar 18 Pembagian Dataset 2

Membagi kembali data training (80% dari total) dengan mengambil 10% untuk validation set. Hasil pembagian ini menghasilkan training set sebesar 72% dari total dataset (90% dari 80%) dan validation set sebesar 8% dari total dataset (10% dari 80%). Strategi pembagian hierarkis ini memastikan validation set benar-benar merupakan subset dari data training original.

2.4. Menampilkan Informasi Statistik Dataset

```
print("Jumlah total data:", len(df))
print("Jumlah data Training:", len(train_df), "(80% dari total)")
print("Jumlah data Validation:", len(val_df), "(10% dari Training)")
print("Jumlah data Testing:", len(test_df), "(20% dari total)")
```

Gambar 19 Informasi Dataset

Menampilkan ringkasan jumlah data di setiap set beserta keterangan persentasenya. Output ini berfungsi sebagai verifikasi bahwa pembagian dataset telah dilakukan sesuai spesifikasi. Keterangan dalam kurung menjelaskan basis perhitungan persentase untuk memperjelas interpretasi.

2.5. Preview Data dari Setiap Set

```
print("\n=== Data Training (5 baris teratas) ===")
print(train_df.head())

print("\n=== Data Validation (5 baris teratas) ===")
print(val_df.head())

print("\n=== Data Testing (5 baris teratas) ===")
print(test_df.head())
```

Gambar 20 Preview Data

Menampilkan 5 baris pertama dari masing-masing set sebagai bukti visual bahwa pembagian telah berhasil dilakukan. Preview ini memungkinkan verifikasi struktur data, memastikan tidak ada data yang hilang atau rusak selama proses pembagian, dan memberikan gambaran tentang distribusi data dalam setiap set.

3. Kesimpulan

Kedua tugas ini membantu kita memahami dasar penting dalam menyiapkan data untuk machine learning. Praktikum 2 mengajarkan cara melihat dan memahami data (EDA), sedangkan Tugas Mandiri 2 menjelaskan cara terbaik membagi data agar hasil analisis lebih akurat. Jika digabung, keduanya menjadi bekal yang kuat untuk membuat model machine learning yang baik dan bisa diandalkan.

Link Github Praktikum :

<https://github.com/raffayuda/MachineLearning/blob/main/praktikum1/Notebook/praktikum01.ipynb>

Link Github Praktikum Mandiri :

https://github.com/raffayuda/Machine-Learning/blob/main/praktikum1/Notebook/praktikum01_mandiri.ipynb