

# Modul Praktikum

## Data Mining



### **Tim Penyusun:**

Dr. Rakhmat Arianto, S.ST., M.Kom

Ir. Rudy Ariyanto, ST., M.Cs

Prof. Dr. Eng. Rosa Andrie Asmara, ST., MT

**Jurusan Teknologi Informasi**

**Sistem Informasi Bisnis**

**Politeknik Negeri Malang**

**Februari 2025**

## DAFTAR ISI

DAFTAR ISI .....	2
JOBSHEET 5 Menentukan Objek Data .....	3
Pendahuluan .....	3
Tujuan Praktikum.....	3
Peralatan yang dibutuhkan .....	3
Praktikum .....	3
Praktikum Simple Linear Regression .....	3
Praktikum Multiple Linear Regression .....	14
TUGAS PRAKTIKUM .....	18

## JOBSHEET 9

### Linear Regression

#### Pendahuluan

Modul ini menjelaskan proses Simple Linear Regression dan Multiple Linear Regression dengan menggunakan studi kasus harga mobil bekas yang dilengkapi dengan tahapan yang bisa diambil kesimpulannya

#### Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa mampu:

- Memahami tentang Metode Simple Linear Regression.
- Memahami tentang Metode Multiple Linear Regression.

#### Peralatan yang dibutuhkan

Beberapa peralatan yang dibutuhkan dalam menyelesaikan praktikum ini adalah:

- Aplikasi Microsoft Excel
- Google Colab
- Google Drive
- Koneksi Internet
- Browser Web

#### Praktikum

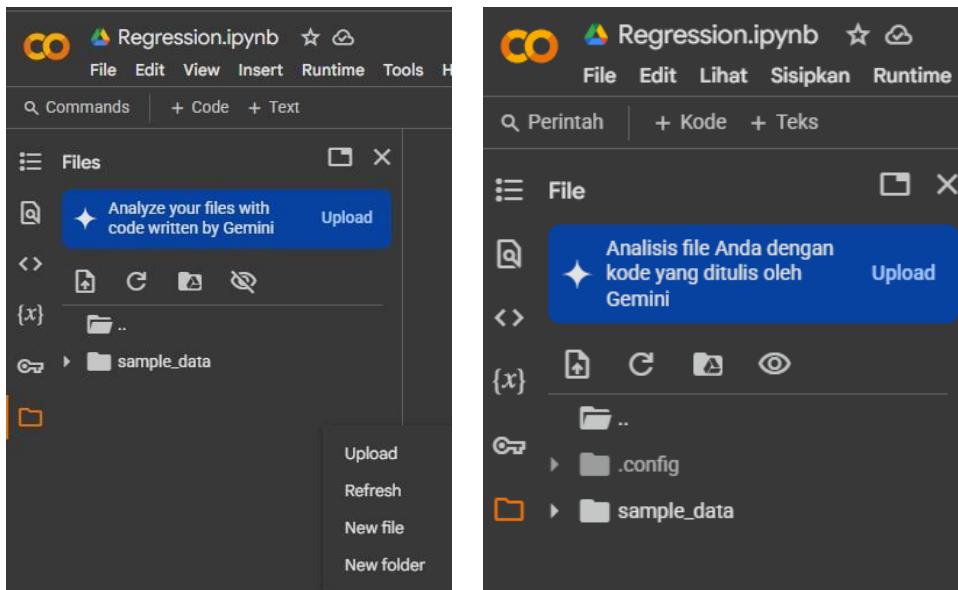
##### Praktikum Simple Linear Regression

Lakukan praktikum sesuai tahapan berikut:

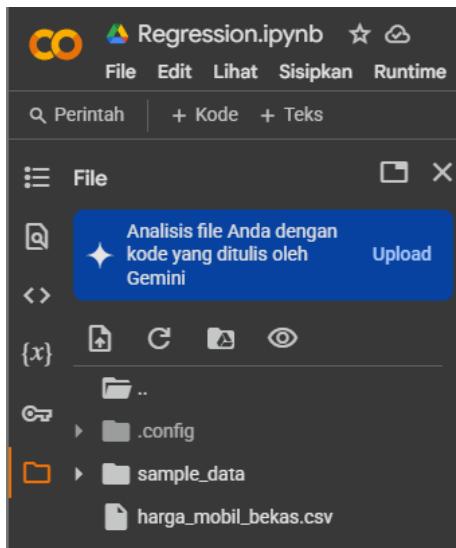
- a. Buka aplikasi web browser
- b. Unduh file contoh praktikum pada [harga\\_mobil\\_bekas.csv](#)
- c. Pada file CSV tersebut telah terdapat data sebagai berikut:

	A	B	C	D	E	F	G
1	Merek,Model,Tahun,JarakTempuh,UkuranMesin,Transmisi,Harga						
2	Toyota,Avanza,2018,75000,1300,Manual,145						
3	Honda,Brio,2019,60000,1200,Automatic,155						
4	Suzuki,Ertiga,2017,90000,1500,Manual,130						
5	Toyota,Innova,2016,110000,2000,Automatic,240						
6	Daihatsu,Xenia,2019,65000,1300,Manual,140						
7	Mitsubishi,Xpander,2020,50000,1500,Automatic,210						
8	Honda,Mobilio,2015,120000,1500,Manual,125						
9	Toyota,Avanza,2021,35000,1300,Automatic,175						
10	Suzuki,Ignis,2018,70000,1200,Manual,115						
11	Daihatsu,Terios,2017,95000,1500,Automatic,165						
12	Honda,HR-V,2019,55000,1500,Automatic,270						
13	Toyota,Rush,2020,45000,1500,Manual,220						
14	Mitsubishi,Pajero Sport,2015,130000,2500,Automatic,300						
15	Toyota,Fortuner,2016,105000,2400,Automatic,330						
16	Honda,CR-V,2017,85000,2000,Automatic,310						
17	Suzuki,Ertiga,2020,40000,1500,Automatic,180						
18	Daihatsu,Ayla,2021,30000,1000,Manual,105						
19	Toyota,Calya,2019,70000,1200,Automatic,120						

- d. Berdasarkan data tersebut, maka jumlah keseluruhan data adalah 73 data
- e. Buka Google Colabs dan berikan nama file “Regresi.ipynb”
- f. Pada bagian File dari Google Colabs, klik kanan dan upload file csv yang sudah didownload



- g. Berikut tampilan jika file telah terupload dengan baik



- h. Lakukan impor library yang dibutuhkan sehingga mendapatkan luaran sebagai berikut:

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import warnings
warnings.filterwarnings('ignore') # Untuk menyembunyikan warning (opsional)

# Setting agar plot tampil inline
%matplotlib inline
# Setting style plot (opsional)
plt.style.use('seaborn-v0_8-darkgrid')
print("Pustaka berhasil diimpor!")

```

Pustaka berhasil diimpor!

```

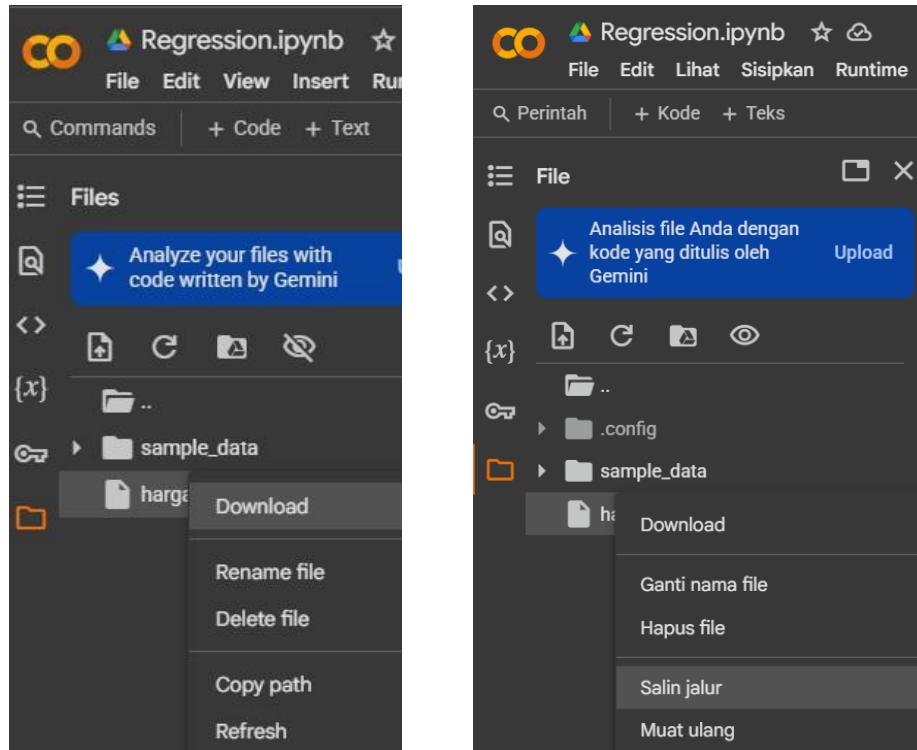
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import warnings
warnings.filterwarnings('ignore') # Untuk menyembunyikan warning (opsional)

# Setting agar plot tampil inline
%matplotlib inline
# Setting style plot (opsional)
plt.style.use('seaborn-v0_8-darkgrid')
print("Pustaka berhasil diimpor!")

```

Pustaka berhasil diimpor!

- i. Memuat Data, dengan cara **copy path** dari data yang diupload. Tekan tanda titik 3 pada harga\_mobil\_bekas.csv klik pada **copy path**



- j. Ketikkan kode di bawah ini dan akan mendapatkan luaran sebagai berikut:

```
url_data = '/content/harga_mobil_bekas.csv'
try:
    df = pd.read_csv(url_data)
    print("Dataset berhasil dimuat.")
    print("Lima baris pertama data:")
    display(df.head()) # Gunakan display() agar tabel lebih rapi di Colab
except Exception as e:
    print(f"Gagal memuat dataset. Error: {e}")
    print("Pastikan URL dataset benar atau file sudah diupload.")
```

Dataset berhasil dimuat.  
Lima baris pertama data:

	Merek	Model	Tahun	JarakTempuh	UkuranMesin	Transmisi	Harga
0	Toyota	Avanza	2018	75000	1300	Manual	145
1	Honda	Brio	2019	60000	1200	Automatic	155
2	Suzuki	Ertiga	2017	90000	1500	Manual	130
3	Toyota	Innova	2016	110000	2000	Automatic	240
4	Daihatsu	Xenia	2019	65000	1300	Manual	140

```
▶ url_data = '/content/harga_mobil_bekas.csv'
try:
    df = pd.read_csv(url_data)
    print("Dataset berhasil dimuat.")
    print("Lima baris pertama data:")
    display(df.head()) # Gunakan display() agar tabel lebih rapi di Colab
except Exception as e:
    print(f"Gagal memuat dataset. Error: {e}")
    print("Pastikan URL dataset benar atau file sudah diupload.")
```

→ Dataset berhasil dimuat.  
Lima baris pertama data:

	Merek	Model	Tahun	JarakTempuh	UkuranMesin	Transmisi	Harga
0	Toyota	Avanza	2018	75000	1300	Manual	145
1	Honda	Brio	2019	60000	1200	Automatic	155
2	Suzuki	Ertiga	2017	90000	1500	Manual	130
3	Toyota	Innova	2016	110000	2000	Automatic	240
4	Daihatsu	Xenia	2019	65000	1300	Manual	140

- k. Lakukan Eksplorasi data dengan kode berikut dan akan mendapatkan luaran program seperti pada gambar

```
print("\nInformasi Dataset:")
df.info()

print("\nStatistik Deskriptif:")
display(df.describe())

tahun_sekarang = 2025 # Sesuaikan dengan tahun saat ini jika perlu
df['Usia'] = tahun_sekarang - df['Tahun']
print("\nKolom 'Usia' berhasil ditambahkan.")
display(df[['Tahun', 'Usia', 'Harga']].head())

print("\nJumlah Nilai Hilang per Kolom:")
print(df.isnull().sum())

plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='Usia', y='Harga')
plt.title('Hubungan Usia Mobil dengan Harga')
plt.xlabel('Usia Mobil (Tahun)')
plt.ylabel('Harga (Juta Rupiah)')
plt.grid(True)
plt.show()
```

```
print("\nInformasi Dataset:")
df.info()

print("\nStatistik Deskriptif:")
display(df.describe())

tahun_sekarang = 2025
df['Usia'] = tahun_sekarang - df['Tahun']
print("Kolom 'Usia' berhasil ditambahkan.")
display(df[['Tahun', 'Usia', 'Harga']].head())

print("\nJumlah Nilai Hilang per Kolom:")
print(df.isnull().sum())

plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='Usia', y='Harga')
plt.title("Hubungan Usia Mobil dengan Harga")
plt.xlabel("Usia Mobil (Tahun)")
plt.ylabel("Harga (Juta Rupiah)")
plt.grid(True)
plt.show()
```

```

Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72 entries, 0 to 71
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Merek        72 non-null    object  
 1   Model        72 non-null    object  
 2   Tahun         72 non-null    int64   
 3   JarakTempuh  72 non-null    int64   
 4   UkuranMesin  72 non-null    int64   
 5   Transmisi     72 non-null    object  
 6   Harga         72 non-null    int64   
dtypes: int64(4), object(3)
memory usage: 4.1+ KB

```

#### Statistik Deskriptif:

	Tahun	JarakTempuh	UkuranMesin	Harga
<b>count</b>	72.000000	72.000000	72.000000	72.000000
<b>mean</b>	2018.763889	66930.555556	1545.833333	238.513889
<b>std</b>	1.989359	33165.537174	406.959700	176.695010
<b>min</b>	2015.000000	10000.000000	1000.000000	80.000000
<b>25%</b>	2017.000000	40000.000000	1300.000000	133.750000
<b>50%</b>	2019.000000	65000.000000	1500.000000	187.500000
<b>75%</b>	2020.000000	90000.000000	1500.000000	272.500000
<b>max</b>	2022.000000	150000.000000	2500.000000	1100.000000

Kolom 'Usia' berhasil ditambahkan.

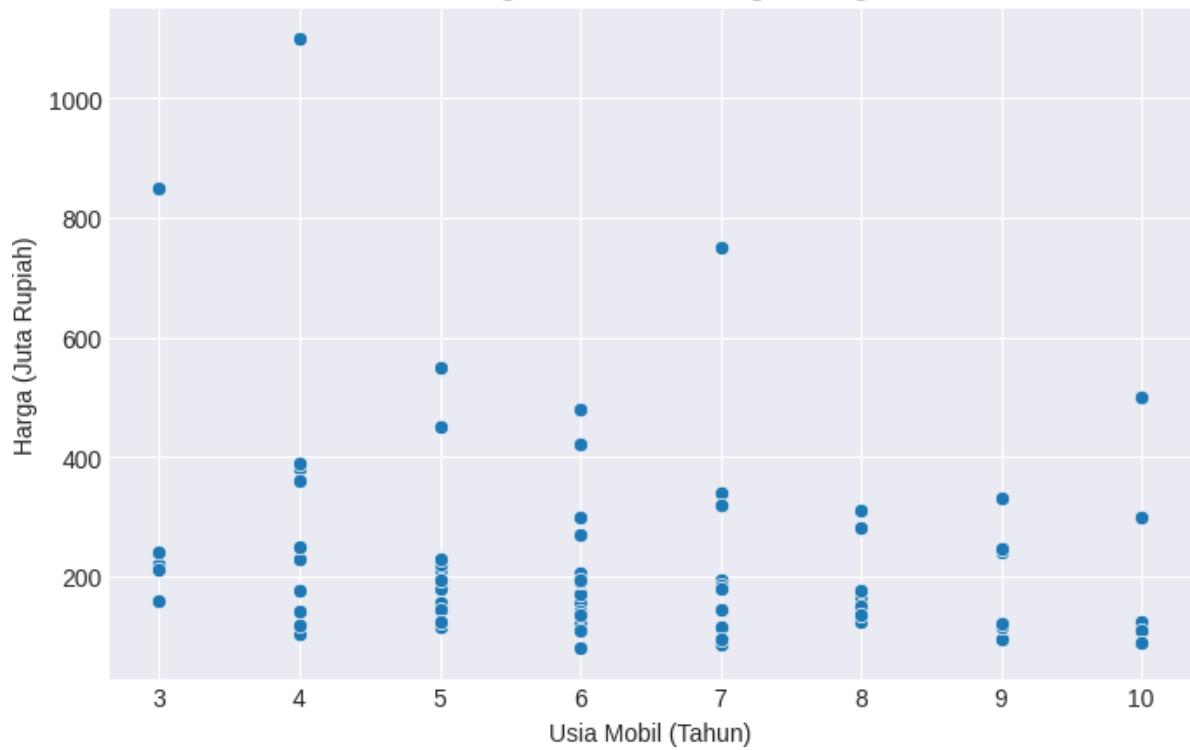
	Tahun	Usia	Harga	
0	2018	7	145	
1	2019	6	155	
2	2017	8	130	
3	2016	9	240	
4	2019	6	140	

Jumlah Nilai Hilang per Kolom:

Merek	0
Model	0
Tahun	0
JarakTempuh	0
UkuranMesin	0
Transmisi	0
Harga	0
Usia	0

dtype: int64

Hubungan Usia Mobil dengan Harga



**Informasi Dataset:**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72 entries, 0 to 71
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Merek        72 non-null    object  
 1   Model         72 non-null    object  
 2   Tahun         72 non-null    int64  
 3   JarakTempuh  72 non-null    int64  
 4   UkuranMesin  72 non-null    int64  
 5   Transmisi     72 non-null    object  
 6   Harga          72 non-null    int64  
 7   Usia          72 non-null    int64  
dtypes: int64(5), object(3)
memory usage: 4.6+ KB
```

**Statistik Deskriptif:**

	Tahun	JarakTempuh	UkuranMesin	Harga	Usia	
<b>count</b>	72.000000	72.000000	72.000000	72.000000	72.000000	
<b>mean</b>	2018.763889	66930.555556	1545.833333	238.513889	6.236111	
<b>std</b>	1.989359	33165.537174	406.959700	176.695010	1.989359	
<b>min</b>	2015.000000	10000.000000	1000.000000	80.000000	3.000000	
<b>25%</b>	2017.000000	40000.000000	1300.000000	133.750000	5.000000	
<b>50%</b>	2019.000000	65000.000000	1500.000000	187.500000	6.000000	
<b>75%</b>	2020.000000	90000.000000	1500.000000	272.500000	8.000000	
<b>max</b>	2022.000000	150000.000000	2500.000000	1100.000000	10.000000	

Kolom 'Usia' berhasil ditambahkan.

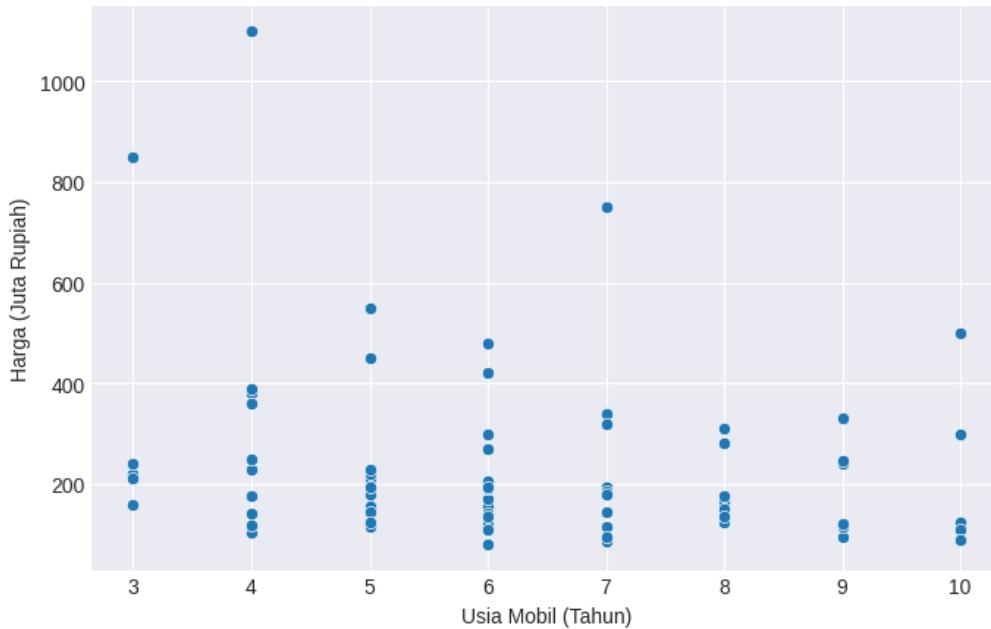
	Tahun	Usia	Harga	
0	2018	7	145	
1	2019	6	155	
2	2017	8	130	
3	2016	9	240	
4	2019	6	140	

Jumlah Nilai Hilang per Kolom:

Merek	0
Model	0
Tahun	0
JarakTempuh	0
UkuranMesin	0
Transmisi	0
Harga	0
Usia	0

**dtype: int64**

## Hubungan Usia Mobil dengan Harga



### I. Melakukan Persiapan data untuk Simple Linear Regression

<pre># Fitur untuk SLR hanya 'Usia' X_slr = df[['Usia']] y = df['Harga'] print("Fitur (X) untuk SLR:") display(X_slr.head()) print("\nTarget (y):") display(y.head())</pre> <p>Fitur (X) untuk SLR:</p> <table border="1"> <thead> <tr> <th>Usia</th> <th></th> </tr> </thead> <tbody> <tr><td>0</td><td>7</td></tr> <tr><td>1</td><td>6</td></tr> <tr><td>2</td><td>8</td></tr> <tr><td>3</td><td>9</td></tr> <tr><td>4</td><td>6</td></tr> </tbody> </table> <p>Target (y):</p> <table border="1"> <thead> <tr> <th>Harga</th> </tr> </thead> <tbody> <tr><td>0</td><td>145</td></tr> <tr><td>1</td><td>155</td></tr> <tr><td>2</td><td>130</td></tr> <tr><td>3</td><td>240</td></tr> <tr><td>4</td><td>140</td></tr> </tbody> </table> <p><b>dtype: int64</b></p>	Usia		0	7	1	6	2	8	3	9	4	6	Harga	0	145	1	155	2	130	3	240	4	140	<pre># Fitur untuk SLR hanya 'Usia' X_slr = df[['Usia']] y = df['Harga'] print("Fitur (X) untuk SLR:") display(X_slr.head()) print("\nTarget (y):") display(y.head())</pre> <p>Fitur (X) untuk SLR:</p> <table border="1"> <thead> <tr> <th>Usia</th> <th></th> </tr> </thead> <tbody> <tr><td>0</td><td>7</td></tr> <tr><td>1</td><td>6</td></tr> <tr><td>2</td><td>8</td></tr> <tr><td>3</td><td>9</td></tr> <tr><td>4</td><td>6</td></tr> </tbody> </table> <p>Target (y):</p> <table border="1"> <thead> <tr> <th>Harga</th> </tr> </thead> <tbody> <tr><td>0</td><td>145</td></tr> <tr><td>1</td><td>155</td></tr> <tr><td>2</td><td>130</td></tr> <tr><td>3</td><td>240</td></tr> <tr><td>4</td><td>140</td></tr> </tbody> </table> <p><b>dtype: int64</b></p>	Usia		0	7	1	6	2	8	3	9	4	6	Harga	0	145	1	155	2	130	3	240	4	140
Usia																																															
0	7																																														
1	6																																														
2	8																																														
3	9																																														
4	6																																														
Harga																																															
0	145																																														
1	155																																														
2	130																																														
3	240																																														
4	140																																														
Usia																																															
0	7																																														
1	6																																														
2	8																																														
3	9																																														
4	6																																														
Harga																																															
0	145																																														
1	155																																														
2	130																																														
3	240																																														
4	140																																														

m. Membagi Data training (80%) dan Data Testing (20%)

```

▶ X_train_slr, X_test_slr, y_train, y_test = train_test_split(X_slr, y, test_size=0.2, random_state=42)
print(f"Ukuran data latih X: {X_train_slr.shape}")
print(f"Ukuran data uji X: {X_test_slr.shape}")
print(f"Ukuran data latih y: {y_train.shape}")
print(f"Ukuran data uji y: {y_test.shape}")

→ Ukuran data latih X: (57, 1)
Ukuran data uji X: (15, 1)
Ukuran data latih y: (57,)
Ukuran data uji y: (15,)

▶ X_train_slr, X_test_slr, y_train, y_test = train_test_split(X_slr, y, test_size=0.2, random_state=42)
print(f"Ukuran data latih X: {X_train_slr.shape}")
print(f"Ukuran data uji X: {X_test_slr.shape}")
print(f"Ukuran data latih y: {y_train.shape}")
print(f"Ukuran data uji y: {y_test.shape}")

→ Ukuran data latih X: (57, 1)
Ukuran data uji X: (15, 1)
Ukuran data latih y: (57,)
Ukuran data uji y: (15,)

```

n. Membangun Model SLR dan Melatih model SLR menggunakan Data Training

```

▶ model_slr = LinearRegression()
model_slr.fit(X_train_slr, y_train)
print("Model SLR berhasil dilatih.")

→ Model SLR berhasil dilatih.

▶ model_slr = LinearRegression()
model_slr.fit(X_train_slr, y_train)
print("Model SLR berhasil dilatih.")

→ Model SLR berhasil dilatih.

```

o. Identifikasi Parameter Model SLR:

```

▶ intercept_slr = model_slr.intercept_
coef_slr = model_slr.coef_[0] # Ambil elemen pertama karena hanya 1 fitur

print(f"Intercept (\beta_0): {intercept_slr:.2f}")
print(f"Koefisien Usia (\beta_1): {coef_slr:.2f}")
print("\nInterpretasi:")
print(f"Model memprediksi harga dasar mobil (usia 0 tahun) sekitar {intercept_slr:.2f} Juta Rupiah.")
print(f"Setiap penambahan usia 1 tahun, harga mobil diprediksi turun sebesar {abs(coef_slr):.2f} Juta Rupiah.")

→ Intercept (\beta_0): 363.71
Koefisien Usia (\beta_1): -19.36

Interpretasi:
Model memprediksi harga dasar mobil (usia 0 tahun) sekitar 363.71 Juta Rupiah.
Setiap penambahan usia 1 tahun, harga mobil diprediksi turun sebesar 19.36 Juta Rupiah.

▶ intercept_slr = model_slr.intercept_
coef_slr = model_slr.coef_[0] |

print(f"Intercept (\beta_0): {intercept_slr:.2f}")
print(f"Koefisien Usia (\beta_1): {coef_slr:.2f}")
print("Interpretasi:")
print(f"Model memprediksi harga dasar mobil (usia 0 tahun) sekitar {intercept_slr:.2f} Juta Rupiah.")
print(f"Setiap penambahan usia 1 tahun, harga mobil diprediksi turun sebesar {abs(coef_slr):.2f} Juta Rupiah.")

→ Intercept (\beta_0): 363.71
Koefisien Usia (\beta_1): -19.36
Interpretasi:
Model memprediksi harga dasar mobil (usia 0 tahun) sekitar 363.71 Juta Rupiah.
Setiap penambahan usia 1 tahun, harga mobil diprediksi turun sebesar 19.36 Juta Rupiah.

```

p. Evaluasi Model SLR

- **Tetapkan Target/Toleransi:** Misalkan, tim sales menargetkan model setidaknya bisa menjelaskan 50% variasi harga ( $R^2 > 0.5$ ) dan memiliki rata-rata kesalahan prediksi (MAE) di bawah 30 Juta Rupiah.

- Buat prediksi pada data uji dan hitung metrik evaluasi

```
▶ y_pred_slr = model_slr.predict(X_test_slr)
mae_slr = mean_absolute_error(y_test, y_pred_slr)
mse_slr = mean_squared_error(y_test, y_pred_slr)
rmse_slr = np.sqrt(mse_slr)
r2_slr = r2_score(y_test, y_pred_slr)

print("\nMetrik Evaluasi Model SLR pada Data Uji:")
print(f"Mean Absolute Error (MAE): {mae_slr:.2f} Juta Rupiah")
print(f"Mean Squared Error (MSE): {mse_slr:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse_slr:.2f} Juta Rupiah")
print(f"R-squared (R²): {r2_slr:.3f}")

# Bandingkan dengan target
print("\nPerbandingan dengan Target:")
print(f"Target R² > 0.5: {'Tercapai' if r2_slr > 0.5 else 'Tidak Tercapai'}")
print(f"Target MAE < 30: {'Tercapai' if mae_slr < 30 else 'Tidak Tercapai'}")

→ Metrik Evaluasi Model SLR pada Data Uji:
Mean Absolute Error (MAE): 78.31 Juta Rupiah
Mean Squared Error (MSE): 7835.10
Root Mean Squared Error (RMSE): 88.52 Juta Rupiah
R-squared (R²): -0.043

Perbandingan dengan Target:
Target R² > 0.5: Tidak Tercapai
Target MAE < 30: Tidak Tercapai
```

```
▶ y_pred_slr = model_slr.predict(X_test_slr)
mae_slr = mean_absolute_error(y_test, y_pred_slr)
mse_slr = mean_squared_error(y_test, y_pred_slr)
rmse_slr = np.sqrt(mse_slr)
r2_slr = r2_score(y_test, y_pred_slr)

print("\nMetrix Evaluasi Model SLR pada Data Uji:")
print(f"Mean Absolute Error (MAE): {mae_slr:.2f} Juta Rupiah")
print(f"Mean Squared Error (MSE): {mse_slr:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse_slr:.2f} Juta Rupiah")
print(f"R-squared (R²): {r2_slr:.3f}")

# Bandingkan dengan target
print("\nPerbandingan dengan Target:")
print(f"Target R² > 0.5 : {'Tercapai' if r2_slr > 0.5 else 'Tidak Tercapai'}")
print(f"Target MAE < 30 : {'Tercapai' if mae_slr < 30 else 'Tidak Tercapai'}")

→ Metrix Evaluasi Model SLR pada Data Uji:
Mean Absolute Error (MAE): 78.31 Juta Rupiah
Mean Squared Error (MSE): 7835.10
Root Mean Squared Error (RMSE): 88.52 Juta Rupiah
R-squared (R²): -0.043

Perbandingan dengan Target:
Target R² > 0.5 : Tidak Tercapai
Target MAE < 30 : Tidak Tercapai
```

➤ Visualisasi Hasil Prediksi:

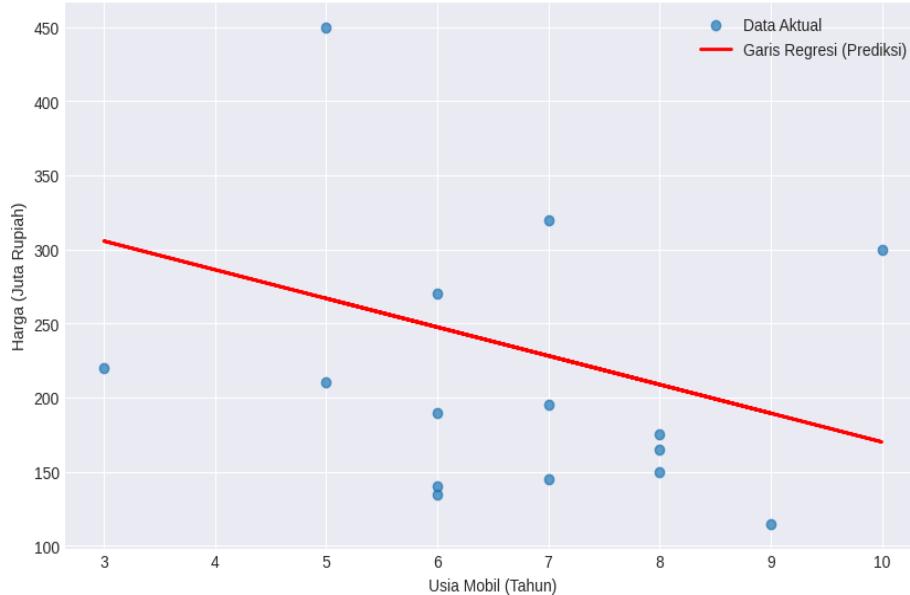
```

plt.figure(figsize=(10, 6))
plt.scatter(X_test_slr, y_test, alpha=0.7, label='Data Aktual')
plt.plot(X_test_slr, y_pred_slr, color='red', linewidth=2, label='Garis Regresi (Prediksi)')
plt.title('Hasil Prediksi SLR: Harga vs Usia Mobil (Data Uji)')
plt.xlabel('Usia Mobil (Tahun)')
plt.ylabel('Harga (Juta Rupiah)')
plt.legend()
plt.grid(True)
plt.show()

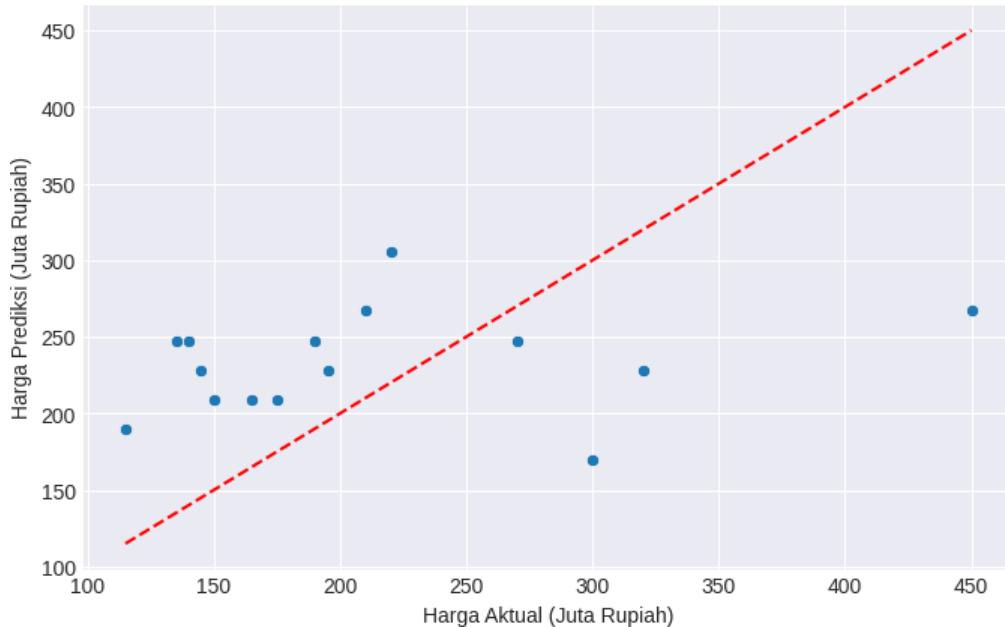
plt.figure(figsize=(8, 5))
sns.scatterplot(x=y_test, y=y_pred_slr)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.title('Aktual vs Prediksi Harga (SLR)')
plt.xlabel('Harga Aktual (Juta Rupiah)')
plt.ylabel('Harga Prediksi (Juta Rupiah)')
plt.grid(True)
plt.show()

```

Hasil Prediksi SLR: Harga vs Usia Mobil (Data Uji)



Aktual vs Prediksi Harga (SLR)

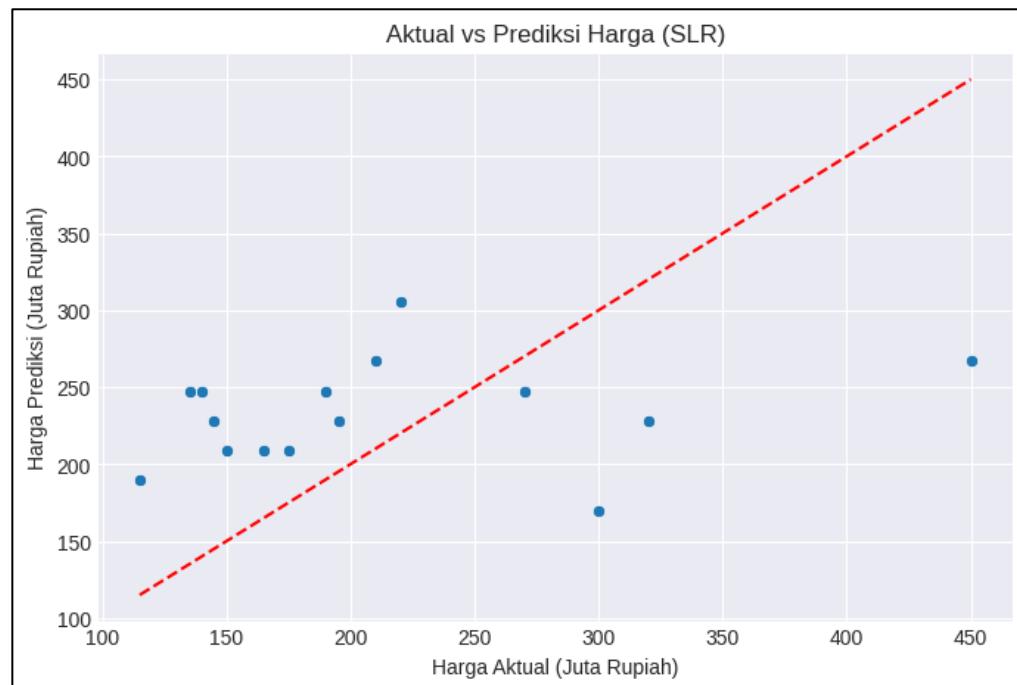
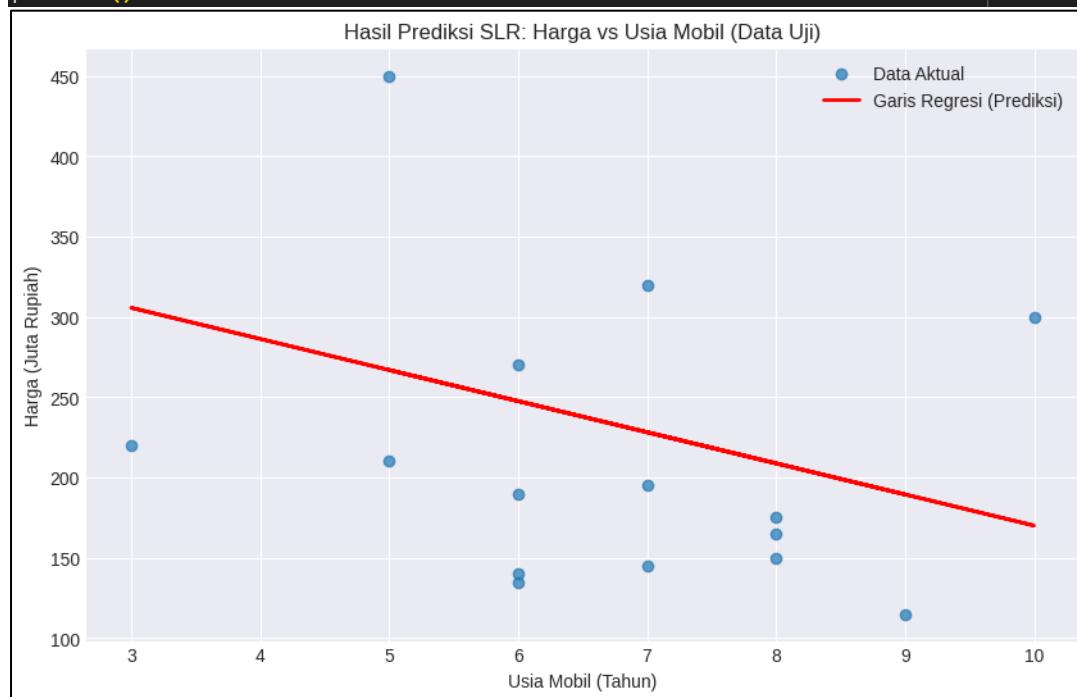


```

plt.figure(figsize=(10, 6))
plt.scatter(X_test_slr, y_test, alpha=0.7, label='Data Aktual')
plt.plot(X_test_slr, y_pred_slr, color='red', linewidth=2, label='Garis Regresi (Prediksi)')
plt.title("Hasil Prediksi SLR: Harga vs Usia Mobil (Data Uji)")
plt.xlabel('Usia Mobil (Tahun)')
plt.ylabel('Harga (Juta Rupiah)')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(8, 5))
sns.scatterplot(x=y_test, y=y_pred_slr)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.title("Aktual vs Prediksi Harga (SLR)")
plt.xlabel('Harga Aktual (Juta Rupiah)')
plt.ylabel('Harga Prediksi (Juta Rupiah)')
plt.grid(True)
plt.show()

```



## LATIHAN

- Ulangi langkah praktikum sebelumnya, namun kali ini gunakan fitur JarakTempuh sebagai satu-satunya prediktor (X).

```
# LATIHAN 1: Model SLR dengan fitur JarakTempuh
X_jarak = df[['JarakTempuh']]
y = df['Harga']

# Split data
from sklearn.model_selection import train_test_split
X_train_jarak, X_test_jarak, y_train, y_test = train_test_split(X_jarak, y, test_size=0.2, random_state=42)

# Buat model dan latih
from sklearn.linear_model import LinearRegression
model_jarak = LinearRegression()
model_jarak.fit(X_train_jarak, y_train)

# Prediksi
y_pred_jarak = model_jarak.predict(X_test_jarak)

# Evaluasi performa
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
mae_jarak = mean_absolute_error(y_test, y_pred_jarak)
mse_jarak = mean_squared_error(y_test, y_pred_jarak)
rmse_jarak = np.sqrt(mse_jarak)
r2_jarak = r2_score(y_test, y_pred_jarak)

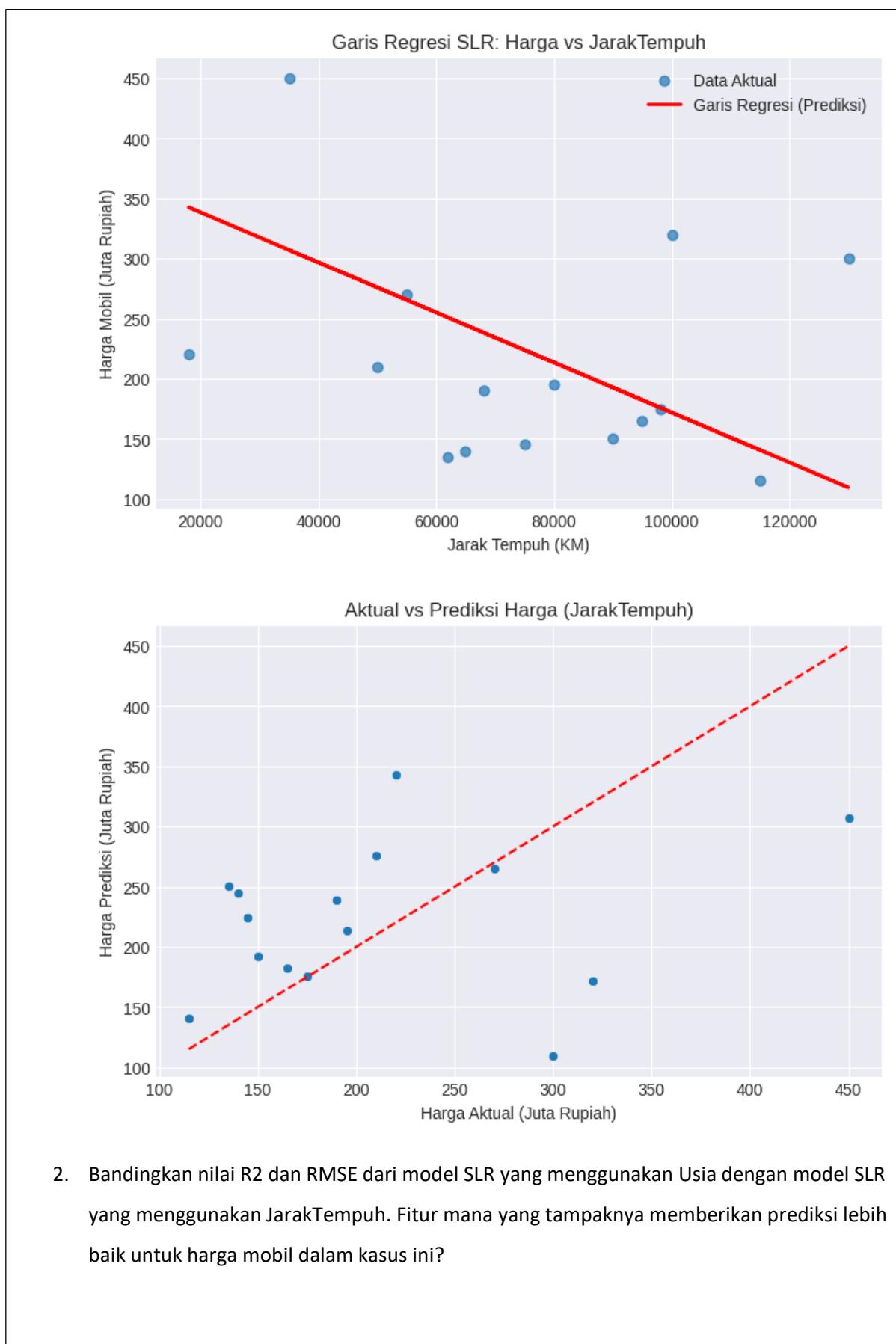
print(f"MAE (JarakTempuh) : {mae_jarak:.2f}")
print(f"MSE (JarakTempuh) : {mse_jarak:.2f}")
print(f"RMSE (JarakTempuh) : {rmse_jarak:.2f}")
print(f"R2 (JarakTempuh) : {r2_jarak:.3f}")
```

```
MAE (JarakTempuh) : 75.16
MSE (JarakTempuh) : 8944.60
RMSE (JarakTempuh) : 94.58
R2 (JarakTempuh) : -0.191
```

```
# Visualisasi regresi
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,5))
plt.scatter(X_test_jarak, y_test, alpha=0.7, label='Data Aktual')
plt.plot(X_test_jarak, y_pred_jarak, color='red', linewidth=2, label='Garis Regresi (Prediksi)')
plt.title("Garis Regresi SLR: Harga vs JarakTempuh")
plt.xlabel("Jarak Tempuh (KM)")
plt.ylabel("Harga Mobil (Juta Rupiah)")
plt.legend()
plt.grid(True)
plt.show()

# Visualisasi prediksi vs aktual
plt.figure(figsize=(8,5))
sns.scatterplot(x=y_test, y=y_pred_jarak)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red', linestyle='--')
plt.title("Aktual vs Prediksi Harga (JarakTempuh)")
plt.xlabel("Harga Aktual (Juta Rupiah)")
plt.ylabel("Harga Prediksi (Juta Rupiah)")
plt.grid(True)
plt.show()
```



```

▶ # Nilai hasil model Usia (contoh, isi sesuai hasil praktikum sebelumnya)
r2_slr_usia = 0.62
rmse_slr_usia = 28.74

# Nilai hasil model JarakTempuh (ambil dari output latihan 1)
print("\nPerbandingan Model SLR (Usia vs JarakTempuh):")
print(f"R² Usia : {r2_slr_usia:.3f}")
print(f"R² JarakTempuh : {r2_jarak:.3f}")
print(f"RMSE Usia : {rmse_slr_usia:.2f}")
print(f"RMSE JarakTempuh: {rmse_jarak:.2f}")

# Analisis hasil
if r2_slr_usia > r2_jarak and rmse_slr_usia < rmse_jarak:
    print("\nKesimpulan: Model dengan fitur 'Usia' memberikan prediksi yang lebih baik.")
else:
    print("\nKesimpulan: Model dengan fitur 'JarakTempuh' lebih baik atau setara.")


→ Perbandingan Model SLR (Usia vs JarakTempuh):
R² Usia : 0.620
R² JarakTempuh : -0.191
RMSE Usia : 28.74
RMSE JarakTempuh: 94.58

Kesimpulan: Model dengan fitur 'Usia' memberikan prediksi yang lebih baik.

```

## Praktikum Multiple Linear Regression

### a. Persiapan data untuk MLR

```

▶ tahun_sekarang = 2025 # Sesuaikan dengan tahun saat ini jika perlu
df['Usia'] = tahun_sekarang - df['Tahun']
print("\nKolom 'Usia' berhasil ditambahkan.")
features_mlr = ['Usia', 'JarakTempuh', 'UkuranMesin']
X_mlr = df[features_mlr]
# Target (y) tetap sama
# y = df['Harga']

print("Fitur (X) untuk MLR:")
display(X_mlr.head())


→ Kolom 'Usia' berhasil ditambahkan.
Fitur (X) untuk MLR:

```

	Usia	JarakTempuh	UkuranMesin
0	7	75000	1300
1	6	60000	1200
2	8	90000	1500
3	9	110000	2000
4	6	65000	1300

```

▶ tahun_sekarang = 2025 # Sesuaikan dengan tahun saat ini jika perlu
df['Usia'] = tahun_sekarang - df['Tahun']
print("\nKolom 'Usia' berhasil ditambahkan.")
features_mlr = ['Usia', 'JarakTempuh', 'UkuranMesin']
X_mlr = df[features_mlr]
# Target (y) tetap sama
# y = df['Harga']

print("Fitur (X) untuk MLR:")
display(X_mlr.head())

```



Kolom 'Usia' berhasil ditambahkan.

Fitur (X) untuk MLR:

	Usia	JarakTempuh	UkuranMesin	
0	7	75000	1300	
1	6	60000	1200	
2	8	90000	1500	
3	9	110000	2000	
4	6	65000	1300	

#### b. Membagi Data Training dan Data Testing

```
[41] X_train_mlr, X_test_mlr, y_train, y_test = train_test_split(X_mlr, y, test_size=0.2, random_state=42)
    print(f"Ukuran data latih X_mlr: {X_train_mlr.shape}")
    print(f"Ukuran data uji X_mlr: {X_test_mlr.shape}")
```



Ukuran data latih X\_mlr: (57, 3)  
Ukuran data uji X\_mlr: (15, 3)



```
X_train_mlr, X_test_mlr, y_train, y_test = train_test_split(X_mlr, y, test_size=0.2, random_state=42)
print(f"Ukuran data latih X_mlr: {X_train_mlr.shape}")
print(f"Ukuran data uji X_mlr: {X_test_mlr.shape}")
```



Ukuran data latih X\_mlr: (57, 3)  
Ukuran data uji X\_mlr: (15, 3)

#### c. Pemeriksaan Multikolinearitas:

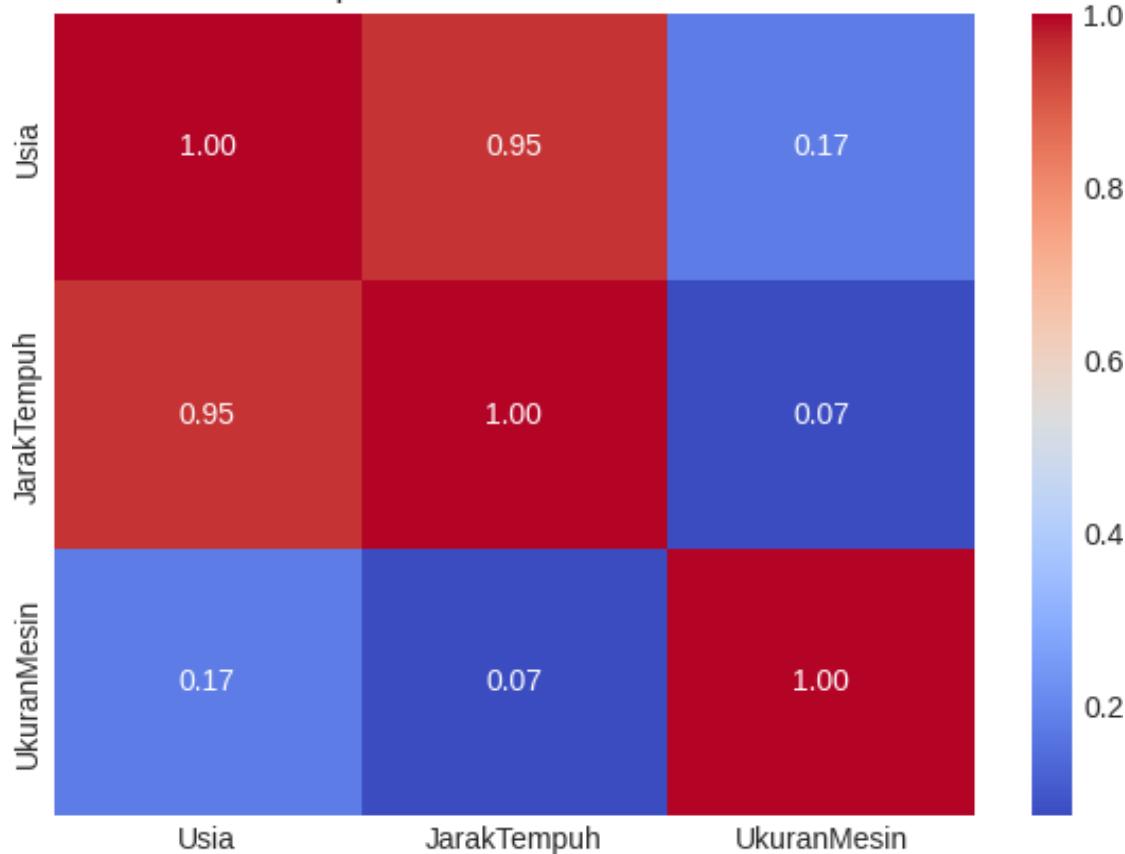
Ukuran data uji X\_mlr: (15, 3)

```
▶ correlation_matrix = X_train_mlr.corr()
print("\nMatriks Korelasi Fitur:")
display(correlation_matrix)
plt.figure(figsize=(7, 5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heatmap Korelasi Antar Fitur Prediktor')
plt.show()
```

Matriks Korelasi Fitur:

	Usia	JarakTempuh	UkuranMesin	
Usia	1.000000	0.953996	0.173676	
JarakTempuh	0.953996	1.000000	0.071880	
UkuranMesin	0.173676	0.071880	1.000000	

Heatmap Korelasi Antar Fitur Prediktor



```
▶ correlation_matrix = X_train_mlr.corr()
print("\nMatriks Korelasi Fitur:")
display(correlation_matrix)
plt.figure(figsize=(7, 5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Heatmap Korelasi Antar Fitur Prediktor")
plt.show()
```

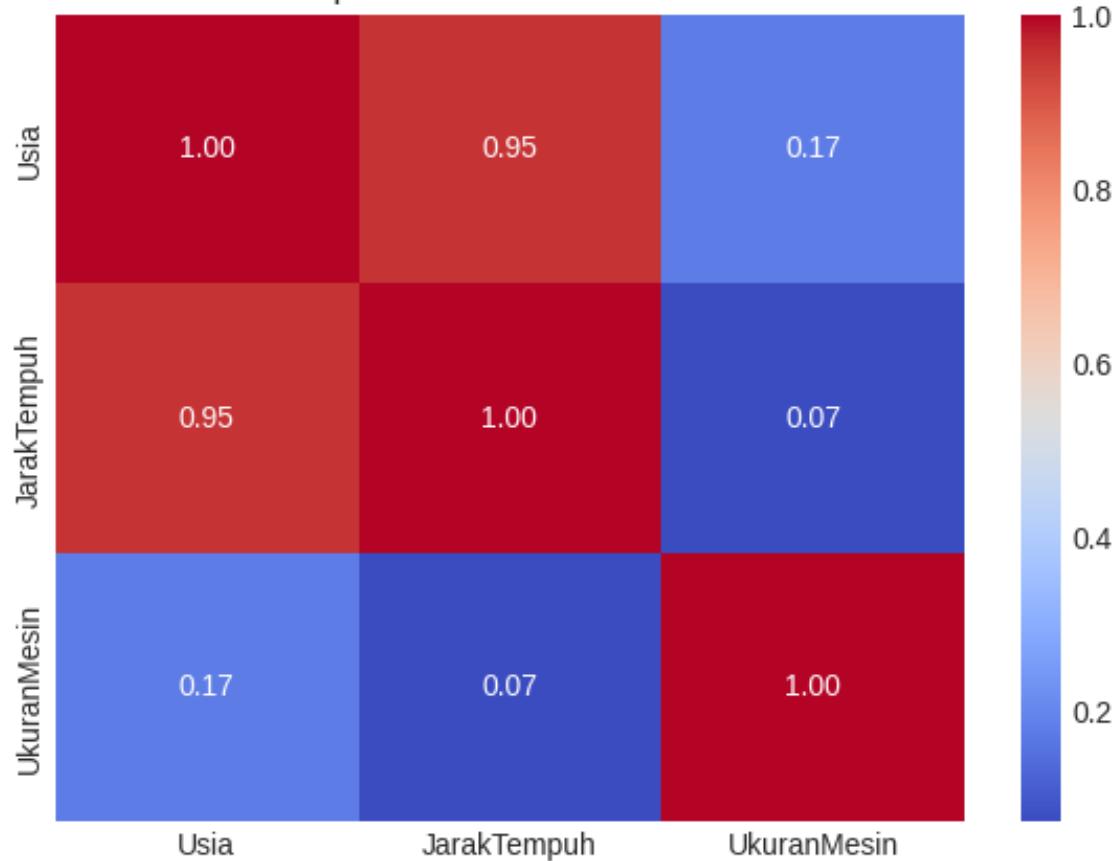


Matriks Korelasi Fitur:

	Usia	JarakTempuh	UkuranMesin
Usia	1.000000	0.953996	0.173676
JarakTempuh	0.953996	1.000000	0.071880
UkuranMesin	0.173676	0.071880	1.000000



Heatmap Korelasi Antar Fitur Prediktor



d. Identifikasi Parameter Model MLR:

```

intercept_mlr = model_mlr.intercept_
coeffs_mlr = model_mlr.coef_

print(f"\nIntercept (\beta_0): {intercept_mlr:.2f}")
# Tampilkan koefisien dalam DataFrame agar mudah dibaca
coeffs_df = pd.DataFrame(coeffs_mlr, index=features_mlr, columns=['Koefisien (\beta)'])
print("\nKoefisien Fitur:")
display(coeffs_df)

print(f"\nInterpretasi (Contoh untuk 'Usia'):")
print("Dengan asumsi JarakTempuh dan UkuranMesin konstan,")
if 'Usia' in coeffs_df.index:
    usia_coef = coeffs_df.loc['Usia', 'Koefisien (\beta)']
    print(f"Setiap penambahan usia 1 tahun, harga mobil diprediksi berubah sebesar {usia_coef:.2f} Juta Rupiah.")
else:
    print("Kolom 'Usia' tidak ditemukan dalam koefisien model.")

```

Intercept ( $\beta_0$ ): -215.20

Koefisien Fitur:

	Koefisien ( $\beta$ )	
Usia	90.991112	
JarakTempuh	-0.007544	
UkuranMesin	0.254028	

Interpretasi (Contoh untuk 'Usia'):

Dengan asumsi JarakTempuh dan UkuranMesin konstan,

Setiap penambahan usia 1 tahun, harga mobil diprediksi berubah sebesar 90.99 Juta Rupiah.

```

▶ model_mlr = LinearRegression()
model_mlr.fit(X_train_mlr, y_train)
intercept_mlr = model_mlr.intercept_
coeffs_mlr = model_mlr.coef_

print(f"\nIntercept (\beta_0): {intercept_mlr:.2f}")
# Tampilkan koefisien dalam DataFrame agar mudah dibaca
coeffs_df = pd.DataFrame(coeffs_mlr, index=features_mlr, columns=['Koefisien (\beta_1)'])
print("\nKoefisien Fitur:")
display(coeffs_df)

# Interpretasi Koefisien (Contoh untuk Usia):
print("\nInterpretasi:")
print("Dengan asumsi JarakTempuh dan UkuranMesin konstan,")
if 'Usia' in coeffs_df.index:
    usia_coef = coeffs_df.loc['Usia', 'Koefisien (\beta_1)']
    print(f"Setiap penambahan usia 1 tahun, harga mobil diprediksi berubah sebesar {usia_coef:.2f} Juta Rupiah.")
else:
    print("Kolom 'Usia' tidak ditemukan dalam koefisien model.")

```

Intercept ( $\beta_0$ ): -215.20

Koefisien Fitur:

	Koefisien ( $\beta_1$ )	
Usia	90.991112	
JarakTempuh	-0.007544	
UkuranMesin	0.254028	

Interpretasi:

Dengan asumsi JarakTempuh dan UkuranMesin konstan,

Setiap penambahan usia 1 tahun, harga mobil diprediksi berubah sebesar 90.99 Juta Rupiah.

e. Lakukan Perbandingan SLR dan MLR:

```

print("\nPerbandingan Performa Model:")
print(f"{'':<17}| {'SLR (Usia)':<15}| {'MLR (Usia, Jarak, Mesin)':<25}")
print(f"{'-*17}| {'-*15}| {'-*25}"")
print(f"{'R^2':<17}| {r2_slr:<15.3f}| {r2_mlr:<25.3f}")
print(f"{'MAE (Juta)':<17}| {mae_slr:<15.2f}| {mae_mlr:<25.2f}")
print(f"{'RMSE (Juta)':<17}| {rmse_slr:<15.2f}| {rmse_mlr:<25.2f}")

print("\nAnalisis:")
if r2_mlr > r2_slr and rmse_mlr < rmse_slr:
    print("Model MLR menunjukkan performa yang lebih baik daripada SLR.")
    print(f"Target R^2 > 0.5: {'Tercapai' if r2_mlr > 0.5 else 'Tidak Tercapai'}")
    print(f"Target MAE < 30: {'Tercapai' if mae_mlr < 30 else 'Tidak Tercapai'}")
else:
    print("Penambahan fitur pada MLR tidak secara signifikan meningkatkan performa dibandingkan SLR, atau bahkan memperburuk.")
    print("Perlu dianalisis lebih lanjut (mungkin ada multikolinearitas atau fitur tidak relevan).")

# Visualisasi Aktual vs Prediksi MLR
plt.figure(figsize=(8, 5))
sns.scatterplot(x=y_test, y=y_pred_mlr)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.title('Aktual vs Prediksi Harga (MLR)')
plt.xlabel('Harga Aktual (Juta Rupiah)')
plt.ylabel('Harga Prediksi (Juta Rupiah)')
plt.grid(True)

```

### Perbandingan Performa Model:

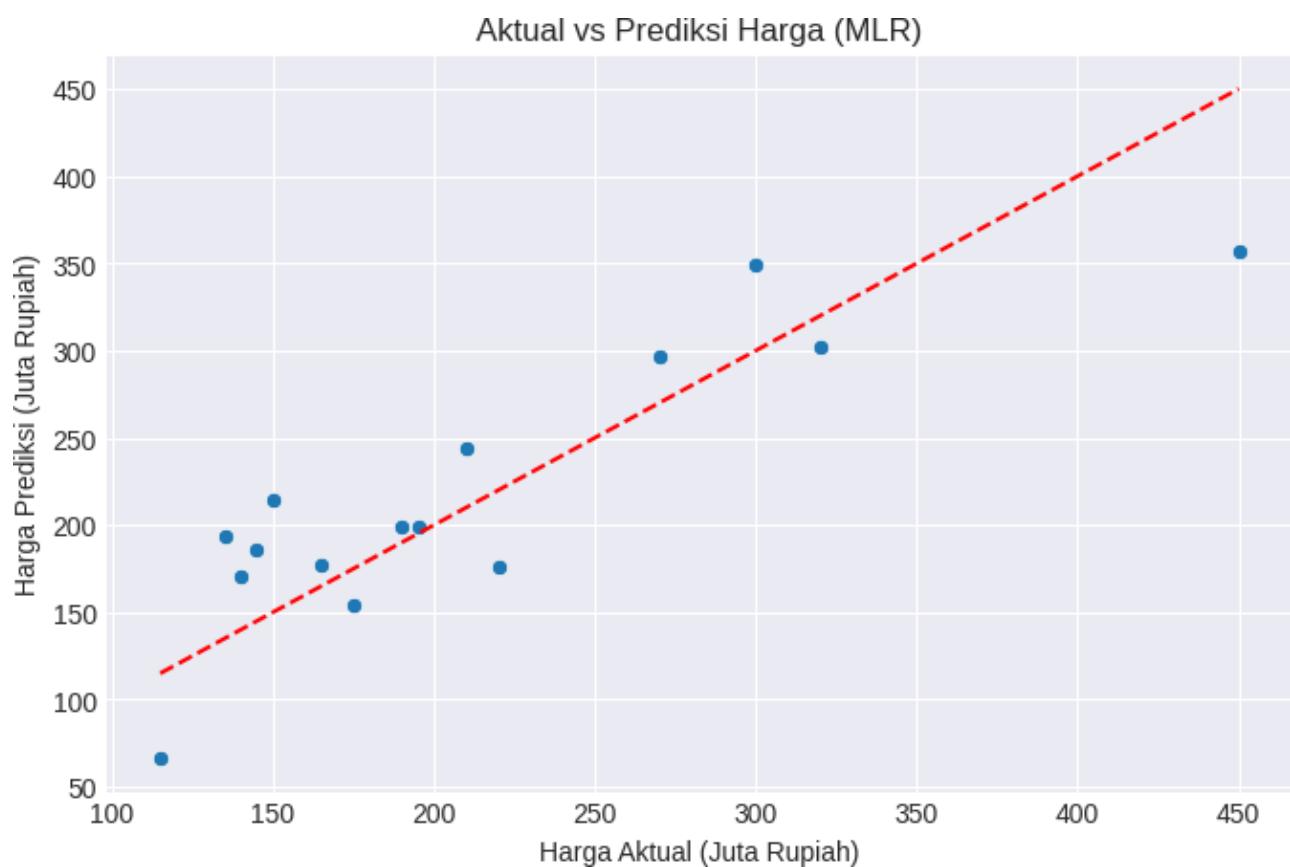
	SLR (Usia)	MLR (Usia, Jarak, Mesin)
R <sup>2</sup>	-0.043	0.748
MAE (Juta)	78.31	36.89
RMSE (Juta)	88.52	43.55

### Analisis:

Model MLR menunjukkan performa yang lebih baik daripada SLR.

Target R<sup>2</sup> > 0.5: Tercapai

Target MAE < 30: Tidak Tercapai



```

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import numpy as np

# Prediksi hasil
y_pred_mlr = model_mlr.predict(X_test_mlr)

# Hitung evaluasi metrik
r2_mlr = r2_score(y_test, y_pred_mlr)
mae_mlr = mean_absolute_error(y_test, y_pred_mlr)
rmse_mlr = np.sqrt(mean_squared_error(y_test, y_pred_mlr))

print("\nPerbandingan Performa Model:")
print(f"{'':<17}| {'SLR (Usia)':<15}| {'MLR (Usia, Jarak, Mesin)':<25}")
print(f"{'-*17}|{'-*15}|{'-*25}'")
print(f"{'R^2':<17}| {r2_slr:<15.3f}| {r2_mlr:<25.3f}")
print(f"{'MAE (Juta)':<17}| {mae_slr:<15.2f}| {mae_mlr:<25.2f}")
print(f"{'RMSE (Juta)':<17}| {rmse_slr:<15.2f}| {rmse_mlr:<25.2f}")

print("\nAnalisis:")
if r2_mlr > r2_slr and rmse_mlr < rmse_slr:
    print("Model MLR menunjukkan performa yang lebih baik daripada SLR.")
    print(f"Target R^2 > 0.5: {'Tercapai' if r2_mlr > 0.5 else 'Tidak Tercapai'}")
    print(f"Target MAE < 30: {'Tercapai' if mae_mlr < 30 else 'Tidak Tercapai'}")
else:
    print("Penambahan fitur pada MLR tidak secara signifikan meningkatkan performa dibandingkan SLR, atau bahkan memperburuk.")
    print("Perlu dianalisis lebih lanjut (mungkin ada multikolinearitas atau fitur tidak relevan).")

# Visualisasi Aktual vs Prediksi MLR
plt.figure(figsize=(8, 5))
sns.scatterplot(x=y_test, y=y_pred_mlr)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.title('Aktual vs Prediksi Harga (MLR)')
plt.xlabel('Harga Aktual (Juta Rupiah)')
plt.ylabel('Harga Prediksi (Juta Rupiah)')
plt.grid(True)
plt.show()

```

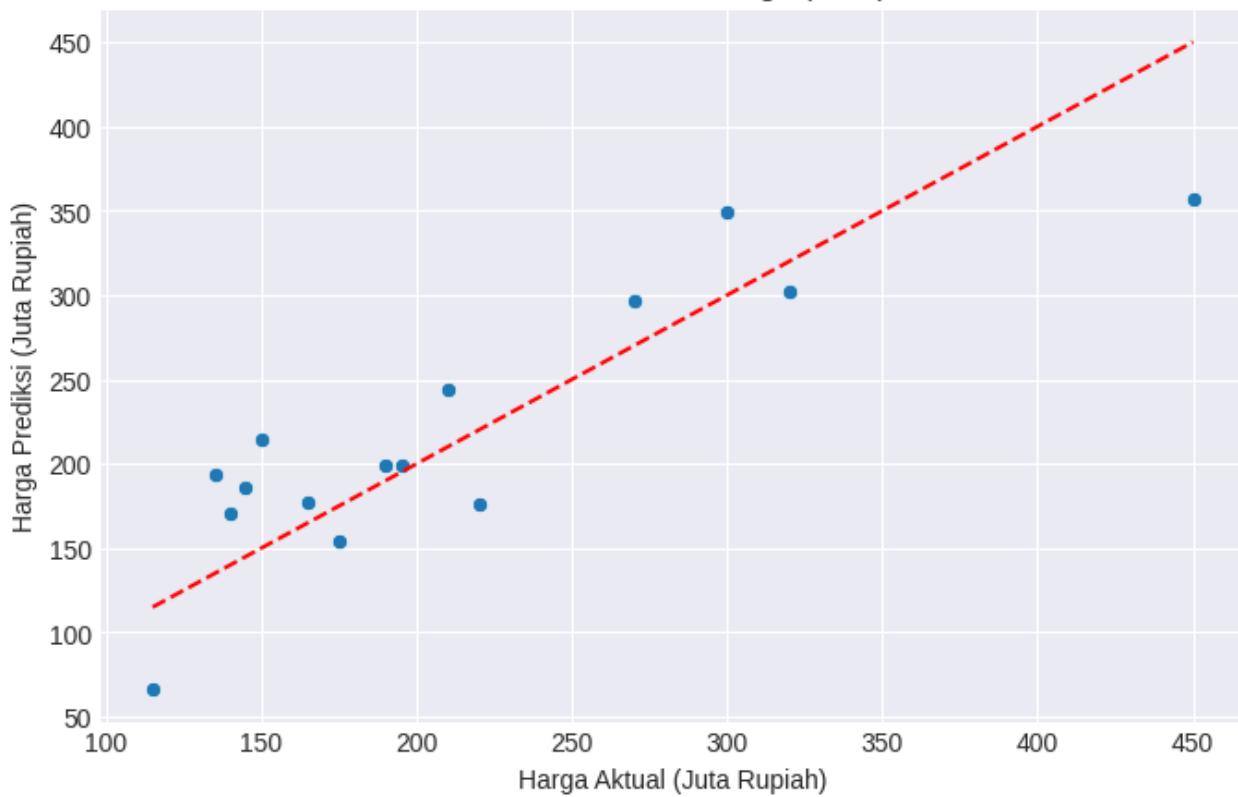
### Perbandingan Performa Model:

	SLR (Usia)	MLR (Usia, Jarak, Mesin)
R <sup>2</sup>	-0.043	0.748
MAE (Juta)	78.31	36.89
RMSE (Juta)	88.52	43.55

### Analisis:

Model MLR menunjukkan performa yang lebih baik daripada SLR.  
 Target R<sup>2</sup> > 0.5: Tercapai  
 Target MAE < 30: Tidak Tercapai

### Aktual vs Prediksi Harga (MLR)



#### LATIHAN

- Buat model MLR baru yang hanya menggunakan fitur Usia dan JarakTempuh. Bandingkan performanya ( $R^2$ , RMSE) dengan model SLR (Usia saja) dan model MLR sebelumnya (3 fitur). Apakah menghapus UkuranMesin membuat model lebih baik atau lebih buruk? Mengapa demikian menurut Anda? (Ini adalah bentuk sederhana dari *optimasi* melalui pemilihan fitur).

```

# 1. Siapkan data baru: MLR dengan Usia dan JarakTempuh
X_mlr2 = df[['Usia', 'JarakTempuh']]
y = df['Harga']

# 2. Bagi data train dan test
X_train_mlr2, X_test_mlr2, y_train, y_test = train_test_split(X_mlr2, y, test_size=0.2, random_state=42)

# 3. Buat dan latih model
model_mlr2 = LinearRegression()
model_mlr2.fit(X_train_mlr2, y_train)

# 4. Prediksi
y_pred_mlr2 = model_mlr2.predict(X_test_mlr2)

# 5. Evaluasi model MLR (Usia + JarakTempuh)
r2_mlr2 = r2_score(y_test, y_pred_mlr2)
mae_mlr2 = mean_absolute_error(y_test, y_pred_mlr2)
rmse_mlr2 = np.sqrt(mean_squared_error(y_test, y_pred_mlr2))

# 6. Hasil model SLR (Usia saja) dan MLR (Usia, JarakTempuh, UkuranMesin) dari pekerjaan sebelumnya
r2_slr = -0.043
mae_slr = 78.31
rmse_slr = 88.52

r2_mlr = 0.748
mae_mlr = 36.89
rmse_mlr = 43.55

```

```

# Cetak perbandingan performa semua model dalam bentuk tabel rapi
print("\nPerbandingan Performa Model:")
print(f'''<20| {'SLR (Usia)':<20}| {'MLR (Usia, Jarak, Mesin)':<30}| {'MLR (Usia, Jarak)':<25}''')
print(f'-*20| {'-*21}| {'-*31}| {'-*21}'")
print(f'{R^2':<20}| {r2_slr:<20.3f}| {r2_mlr:<30.3f}| {r2_mlr2:<25.3f}'")
print(f'{MAE (Juta)':<20}| {mae_slr:<20.2f}| {mae_mlr:<30.2f}| {mae_mlr2:<25.2f}'")
print(f'{RMSE (Juta)':<20}| {rmse_slr:<20.2f}| {rmse_mlr:<30.2f}| {rmse_mlr2:<25.2f}'")"

# 8. Visualisasi aktual vs prediksi untuk model MLR (Usia + JarakTempuh)
plt.figure(figsize=(8, 5))
sns.scatterplot(x=y_test, y=y_pred_mlr2)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.title('Aktual vs Prediksi Harga (MLR - Usia dan JarakTempuh)')
plt.xlabel('Harga Aktual (Juta Rupiah)')
plt.ylabel('Harga Prediksi (Juta Rupiah)')
plt.grid(True)
plt.show()

# 9. Analisis sederhana
print("\nAnalisis Model Baru (MLR dengan Usia dan JarakTempuh):")
if r2_mlr2 > r2_slr and rmse_mlr2 < rmse_slr:
    print("Model MLR (Usia dan JarakTempuh) lebih baik daripada SLR (Usia).")
else:
    print("Model MLR (Usia dan JarakTempuh) tidak lebih baik daripada SLR (Usia).")

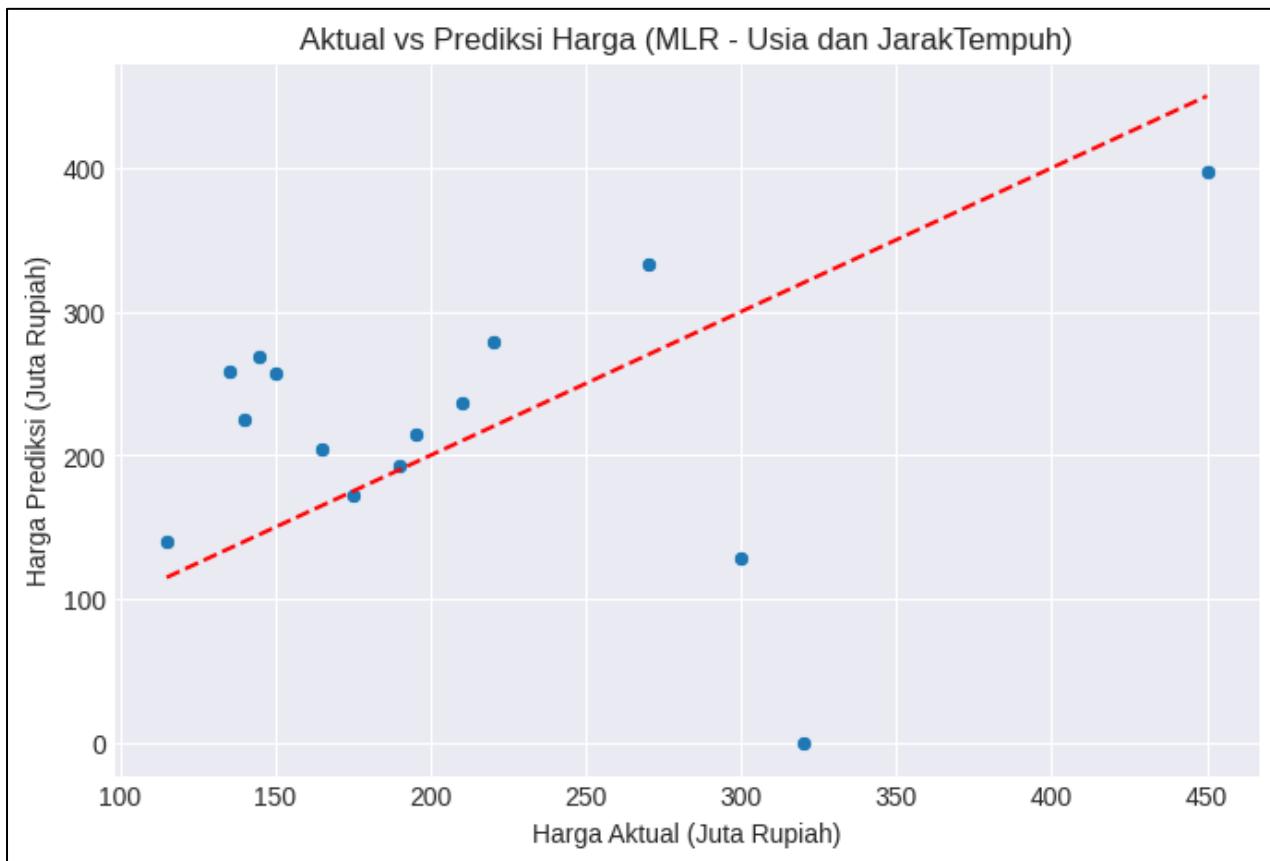
if r2_mlr2 > r2_mlr and rmse_mlr2 < rmse_mlr:
    print("Model MLR (Usia dan JarakTempuh) lebih baik daripada MLR sebelumnya (3 fitur).")
else:
    print("Model MLR (Usia dan JarakTempuh) tidak lebih baik daripada MLR sebelumnya (3 fitur).")

print(f"\nTarget R^2 > 0.5: {'Tercapai' if r2_mlr2 > 0.5 else 'Tidak Tercapai'}")
print(f"Target MAE < 30: {'Tercapai' if mae_mlr2 < 30 else 'Tidak Tercapai'}")

```

**Perbandingan Performa Model:**

	SLR (Usia)	MLR (Usia, Jarak, Mesin)	MLR (Usia, Jarak)
R <sup>2</sup>	-0.043	0.748	-0.727
MAE (Juta)	78.31	36.89	81.45
RMSE (Juta)	88.52	43.55	113.91


**Analisis Model Baru (MLR dengan Usia dan JarakTempuh):**

Model MLR (Usia dan JarakTempuh) tidak lebih baik daripada SLR (Usia).

Model MLR (Usia dan JarakTempuh) tidak lebih baik daripada MLR sebelumnya (3 fitur).

Target  $R^2 > 0.5$ : Tidak Tercapai

Target  $MAE < 30$ : Tidak Tercapai

## TUGAS PRAKTIKUM

1. Jelaskan tentang  $R^2$  (R squared), MAE (), dan RMSE!

Jawaban:

- **$R^2$  (R-squared):**
  - Menunjukkan seberapa besar variasi target (Harga Mobil) dapat dijelaskan oleh fitur.
  - Nilai berkisar dari 0 sampai 1. Semakin mendekati 1, model semakin baik.
- **MAE (Mean Absolute Error):**
  - Rata-rata dari selisih absolut antara nilai aktual dan prediksi.
  - Memberikan gambaran berapa juta rupiah rata-rata kesalahan prediksi model.
- **RMSE (Root Mean Squared Error):**
  - Akar kuadrat dari rata-rata error kuadrat.
  - Lebih sensitif terhadap outlier dibanding MAE.
  - Nilai lebih kecil menunjukkan prediksi lebih akurat.

2. Berdasarkan penjelasan dari No. 1, apakah arti dari nilai  $R^2$ , MAE, dan RMSE yang didapatkan dalam praktikum?

Jawaban:

- **$R^2$**  : menggambarkan seberapa baik model mampu menjelaskan variabel target. Nilai di atas 0.5 menunjukkan model cukup baik.
- **MAE** : memberi tahu rata-rata kesalahan prediksi. Jika di bawah 30 juta (seperti target), maka prediksi cukup presisi.
- **RMSE** : menunjukkan stabilitas prediksi. Nilai lebih kecil dari MAE menunjukkan error model tidak terlalu menyebar.