PREDIKSI MOBIL BEKAS MENGGUNAKAN LINEAR PROGRESSION

Andika Hartanta
2310631170067
5A - Informatika

APA ITU REGRESI LINER

Regresi linear (Linear Regression) merupakan suatu teknik analisis statistik yang dipakai untuk mengamati hubungan antara variabel independen dan variabel dependen dalam bentuk garis yang lurus. Metode ini berguna untuk meramalkan nilai salah satu variabel dengan mempertimbangkan variabel lainnya, contohnya untuk menilai penjualan dari informasi iklan atau memperkirakan tinggi badan berdasarkan usia.

TAHAP PREPROCCESSING

PROFILLING DATA

- Profiling Data digunakan untuk memuat data, melihat total baris, kolom, dan nama kolomnya. Ini dilakukan agar kita bisa memeriksa data terlebih dahulu sebelum melanjutkan ke proses analisis.

- Di sini saya menggunakan dataset yang sudah ditentukan "Prediksi Harga Mobil Bekas".

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000.0	CNG	Manual	First	26.60	998.0	58.16	5.0	1.75
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000.0	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
2	2	Honda Jazz V	Chennai	2011	46000.0	Petrol	Manual	First	18.20	1199.0	88.70	5.0	4.50
3	3	Maruti Ertiga VDI	Chennai	2012	87000.0	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670.0	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6014	6014	Maruti Swift VDI	Delhi	2014	27365.0	Diesel	Manual	First	28.40	1248.0	74.00	5.0	4.75
6015	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000.0	Diesel	Manual	First	24.40	1120.0	71.00	5.0	4.00
6016	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000.0	Diesel	Manual	Second	14.00	2498.0	112.00	8.0	2.90
6017	6017	Maruti Wagon R VXI	Kolkata	2013	46000.0	Petrol	Manual	First	18.90	998.0	67.10	5.0	2.65
6018	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000.0	Diesel	Manual	First	25.44	936.0	57.60	5.0	2.50

6019 rows × 13 columns

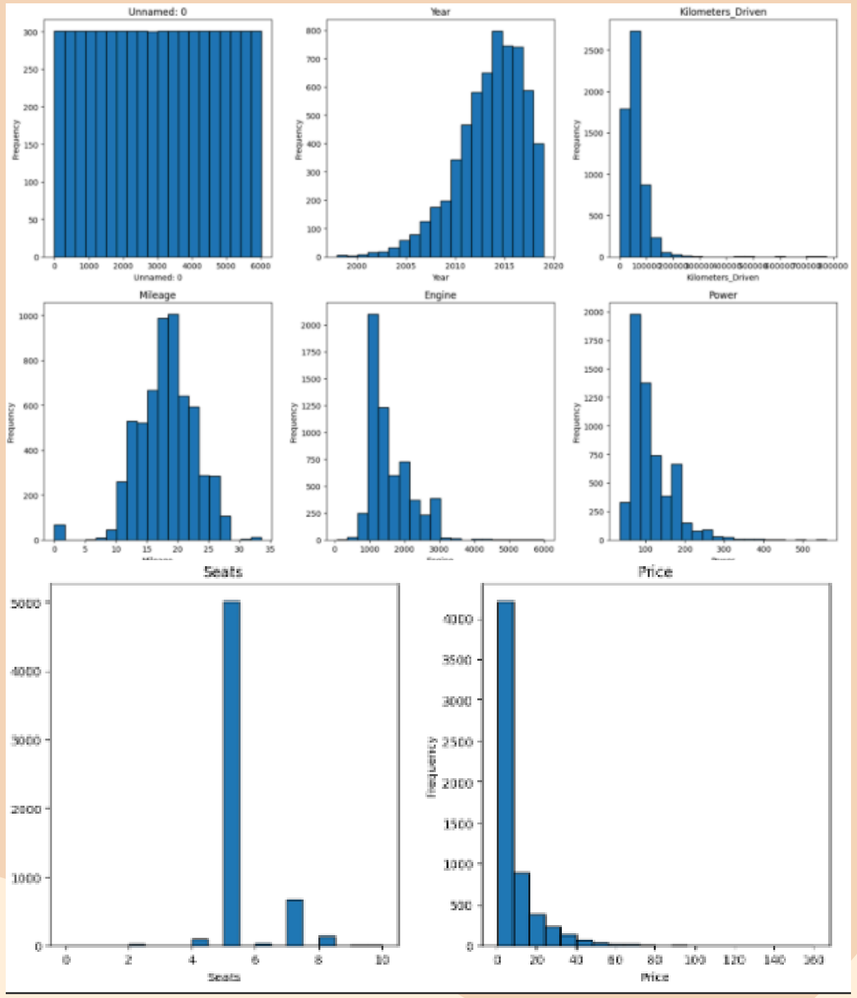
Sumber datasetnya:

- Kaggle - Used Cars Price Prediction
- Github - User Cars Price Prediction

EXPLORATORY DATA ANALYSIS (EDA)

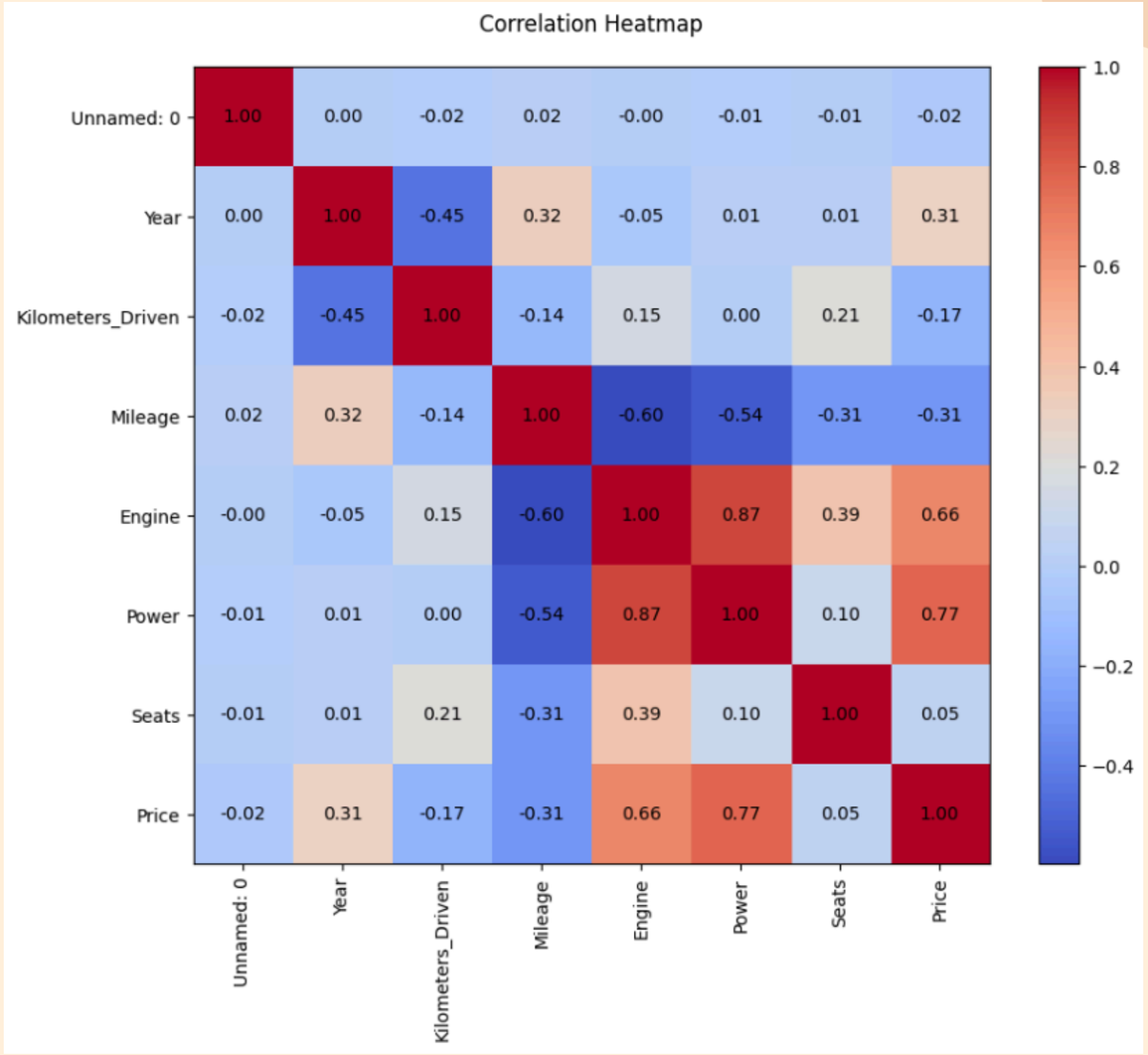
HISTOGRAM

Grafik ini menggambarkan pola sebaran dari setiap variabel dalam kumpulan data mobil. Dapat dilihat bahwa tahun 2015 memiliki jumlah mobil terbanyak. Kolom Kilometers_Driven, Engine, Power, dan Price menunjukkan distribusi yang condong ke kanan karena adanya beberapa nilai yang sangat tinggi. Sebaran Mileage terlihat hampir normal, sementara variabel Seats paling sering muncul pada angka 5, menunjukkan bahwa mobil dengan 5 kursi adalah yang paling umum.



CORRELATION HEATMAP

Heatmap korelasi menunjukkan hubungan antar variabel numerik. Engine, Power, dan Price memiliki korelasi yang tinggi. semakin besar mesin dan tenaga, maka harga akan semakin tinggi. Mileage berkorelasi negatif dengan ketiganya, artinya mobil bertenaga cenderung lebih boros. Year berkorelasi positif sedang dengan Price, menandakan mobil keluaran baru lebih mahal.



DATA CLEANING & TAHAP MODELLING

DATA CLEANING

Bagian Data Cleaning ini digunakan untuk membersihkan data agar hasil analisis atau model nantinya lebih akurat.
Langkah-langkah yang saya lakukan di tahap ini meliputi:

1. Remove Null Value

Saya cek dulu apakah ada data yang bernilai null atau kosong. Kalau ada, saya isi dengan nilai rata-rata dari kolom tersebut, supaya data tidak hilang terlalu banyak.

```
# Cek nilai Null pada dataset
df_clean.isnull().sum()
```

	0
Unnamed: 0	0
Name	0
Location	0
Year	0
Kilometers_Driven	300
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	2
Engine	36
Power	143
Seats	42
Price	0

dtype: int64

```
# Cek dataset setelah diisi nilainya
df_clean.isnull().sum()
```

	0
Unnamed: 0	0
Name	0
Location	0
Year	0
Kilometers_Driven	0
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	0
Engine	0
Power	0
Seats	0
Price	0

dtype: int64

```
# Isi nilai kosong di kolom numerik dengan nilai rata-rata kolomnya
cols_with_missing = ['Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats']

for col in cols_with_missing:
    df_clean[col] = df_clean[col].fillna(df_clean[col].mean())
```

DATA CLEANING

2. Remove Duplicate Value

Setelah itu, saya cek apakah ada data yang duplikat. Kalau ada, seharusnya dihapus, tapi di dataset saya tidak ditemukan data duplikat.

```
# Cek duplikasi dataset
df_clean.duplicated().sum()
```

```
np.int64(0)
```

3. Remove Column ID

Kolom ID seperti Unnamed: 0 saya hapus karena tidak memiliki pengaruh terhadap model atau analisis.

```
# Menghapus kolom ID atau kolom yang tidak memiliki nilai untuk model
df_clean.drop(['Unnamed: 0'], axis=1, inplace=True)

df_clean.columns
```

```
Index(['Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',
       'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats',
       'Price'],
      dtype='object')
```

4. Remove Outliers

Saya juga menghapus outlier atau nilai-nilai ekstrem menggunakan metode IQR (Interquartile Range).

Dari proses ini, sekitar 29% data terdeteksi sebagai outlier dan dihapus agar data menjadi lebih bersih dan stabil.

```
# Hapus outlier dari semua kolom numerik
def remove_outliers_iqr(df, columns):
  df_no_outlier = df.copy()

  Q1 = df_no_outlier[columns].quantile(0.25)
  Q3 = df_no_outlier[columns].quantile(0.75)
  IQR = Q3 - Q1
  lower = Q1 - 1.5 * IQR
  upper = Q3 + 1.5 * IQR

  outlier_filters = ~((df_no_outlier[columns] < lower) | (df_no_outlier[columns] > upper)).any(axis=1)
  df_no_outlier = df_no_outlier[outlier_filters]
  return df_no_outlier

df_no_outlier = remove_outliers_iqr(df_outlier, num_cols)

print("Jumlah data sebelum hapus outlier: ", len(df_outlier))
print("Jumlah data setelah hapus outlier: ", len(df_no_outlier))
print("Total data yang dihapus: ", len(df_outlier) - len(df_no_outlier))
print("Persentase data yang dihapus: ", round((len(df_outlier) - len(df_no_outlier)) / len(df_outlier) * 100, 2), "%")
```

```
Jumlah data sebelum hapus outlier:  6019
Jumlah data setelah hapus outlier:  4265
Total data yang dihapus:  1754
Persentase data yang dihapus:  29.14 %
```

DATA CLEANING

5. Variance Threshold (Filter Method)
Terakhir, saya gunakan metode Variance Threshold untuk melakukan feature selection, yaitu menyaring fitur-fitur dengan variasi rendah yang tidak berkontribusi besar pada model.
Hasilnya, semua fitur memiliki variansi di atas ambang batas, jadi tidak ada yang dihapus.

```
# Menggunakan list comprehension untuk mendapatkan kolom-kolom low variance
low_var_re = [column for column in df_numeric.columns
              if column not in df_numeric.columns[var_thr_re.get_support()]]

# Tampilkan menggunakan for loop
print("Low variance features:")
for features in low_var_re:
  print(features)
```

```
Low variance features:
```

```
# Cek dataset setelah seleksi fitur menggunakan Variance Threshold
df_clean.head()
```

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000.0	CNG	Manual	First	26.60	998.0	58.16	5.0	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000.0	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
2	Honda Jazz V	Chennai	2011	46000.0	Petrol	Manual	First	18.20	1199.0	88.70	5.0	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000.0	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670.0	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74

```
# Ambil hanya kolom numerik
df_numeric = df_clean.select_dtypes(include=['int64', 'float64'])

var_thr_re = VarianceThreshold(threshold=0.1)
var_thr_re.fit_transform(df_numeric)

# Cek hasilnya
print("Features with variance > 0.1:", var_thr_re.get_support(), '\n')

# Cek hasilnya dengan kolomnya
print("Columns with variance > 0.1:", df_numeric.columns[var_thr_re.get_support()])
```

```
Features with variance > 0.1: [ True  True  True  True  True  True  True]

Columns with variance > 0.1: Index(['Year', 'Kilometers_Driven', 'Mileage', 'Engine', 'Power', 'Seats',
       'Price'],
      dtype='object')
```

True = high variance

False = low variance

```
# Cek perbandingan jumlah keduanya
print("Total features:", len(var_thr_re.get_support()))
print("Features with variance > 0.1:", len(df_numeric.columns[var_thr_re.get_support()]))
```

```
Total features: 7
Features with variance > 0.1: 7
```

```
# Simpan dataset final
df_final = df_clean.copy()
df_final
```

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000.0	CNG	Manual	First	26.60	998.0	58.16	5.0	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000.0	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
2	Honda Jazz V	Chennai	2011	46000.0	Petrol	Manual	First	18.20	1199.0	88.70	5.0	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000.0	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670.0	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
...	...	...	...	...	...	...	...	...	...	...	...	...
6014	Maruti Swift VDI	Delhi	2014	27365.0	Diesel	Manual	First	28.40	1248.0	74.00	5.0	4.75
6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000.0	Diesel	Manual	First	24.40	1120.0	71.00	5.0	4.00
6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000.0	Diesel	Manual	Second	14.00	2498.0	112.00	8.0	2.90
6017	Maruti Wagon R VXI	Kolkata	2013	46000.0	Petrol	Manual	First	18.90	998.0	67.10	5.0	2.65
6018	Chevrolet Beat Diesel	Hyderabad	2011	47000.0	Diesel	Manual	First	25.44	936.0	57.60	5.0	2.50

6019 rows × 12 columns

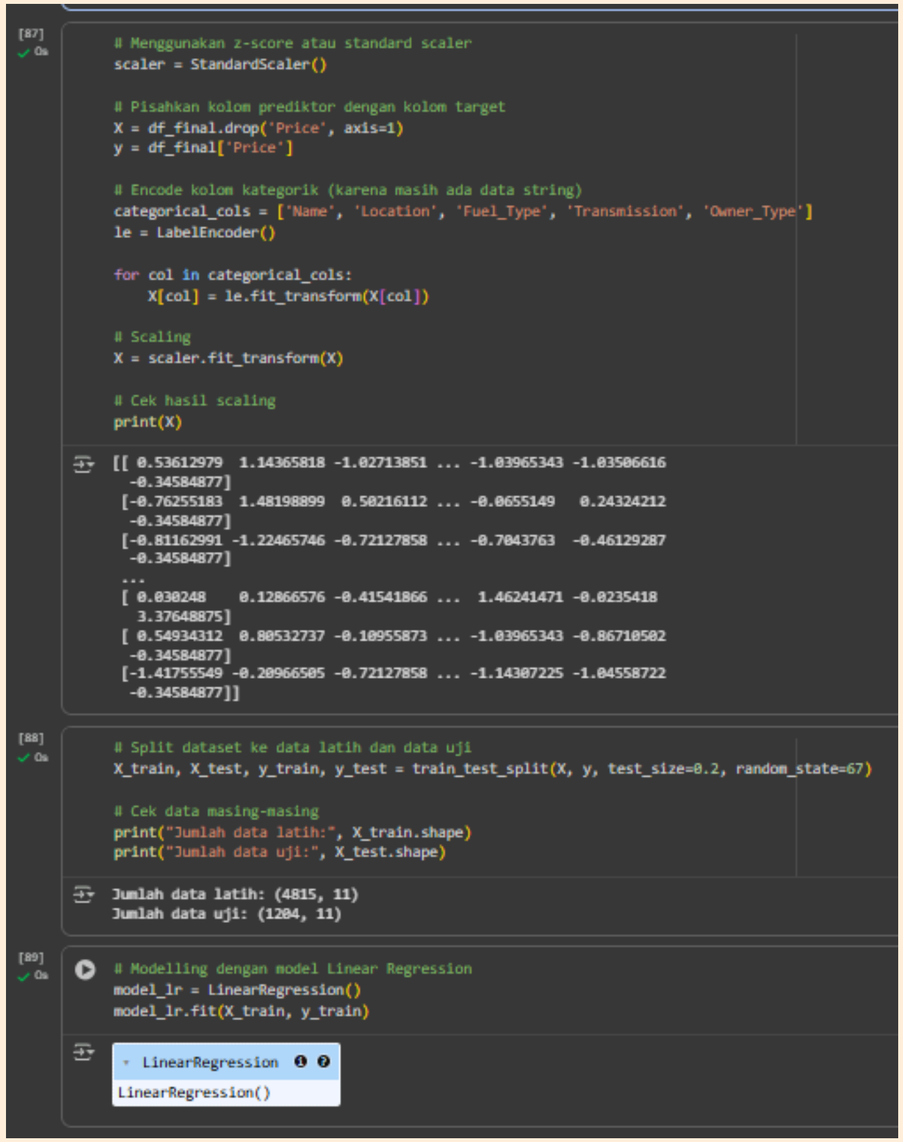
```
# Menampilkan nilai statistika deskriptif setiap variabel dataset
df_final.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Year	6019.0	2013.358199	3.269742	1998.00	2011.00	2014.00	2016.00	2019.00
Kilometers_Driven	6019.0	57545.592586	37029.520244	171.00	35000.00	55000.00	71801.50	775000.00
Mileage	6019.0	18.134961	4.581528	0.00	15.17	18.15	21.10	33.54
Engine	6019.0	1621.276450	599.553865	72.00	1198.00	1493.00	1969.00	5998.00
Power	6019.0	113.253050	53.231019	34.20	78.00	98.60	138.03	560.00
Seats	6019.0	5.278735	0.806012	0.00	5.00	5.00	5.00	10.00
Price	6019.0	9.479468	11.187917	0.44	3.50	5.64	9.95	160.00

+ Code + Text

MODELLING

Disini saya ubah data yang sebelumnya masih string menjadi angka Label Encoding. setelah itu semua data disamakan skalanya pakai Standard Scaler. Kalau udah disamain data yang tadi dibagi jadi data latih buat latih model dan data uji buat menguji hasilnya. Terakhir, model Linear Regression dipakai untuk mempelajari hubungan fitur mobil dan harganya.



[87]
0s

```
# Menggunakan z-score atau standard scaler
scaler = StandardScaler()

# Pisahkan kolom prediktor dengan kolom target
X = df_final.drop('Price', axis=1)
y = df_final['Price']

# Encode kolom kategorik (karena masih ada data string)
categorical_cols = ['Name', 'Location', 'Fuel_Type', 'Transmission', 'Owner_Type']
le = LabelEncoder()

for col in categorical_cols:
    X[col] = le.fit_transform(X[col])

# Scaling
X = scaler.fit_transform(X)

# Cek hasil scaling
print(X)
```

```
[[ 0.53612979  1.14365818 -1.02713851 ... -1.03965343 -1.03506616
  -0.34584877]
 [-0.76255183  1.48198899  0.50216112 ... -0.0655149   0.24324212
  -0.34584877]
 [-0.81162991 -1.22465746 -0.72127858 ... -0.7043763  -0.46129287
  -0.34584877]
 ...
 [ 0.030248    0.12866576 -0.41541866 ...  1.46241471 -0.0235418
   3.37648875]
 [ 0.54934312  0.80532737 -0.10955873 ... -1.03965343 -0.86710502
  -0.34584877]
 [-1.41755549 -0.20966505 -0.72127858 ... -1.14307225 -1.04558722
  -0.34584877]]
```

[88]
0s

```
# Split dataset ke data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=67)

# Cek data masing-masing
print("Jumlah data latih:", X_train.shape)
print("Jumlah data uji:", X_test.shape)
```

```
Jumlah data latih: (4815, 11)
Jumlah data uji: (1204, 11)
```

[89]
0s

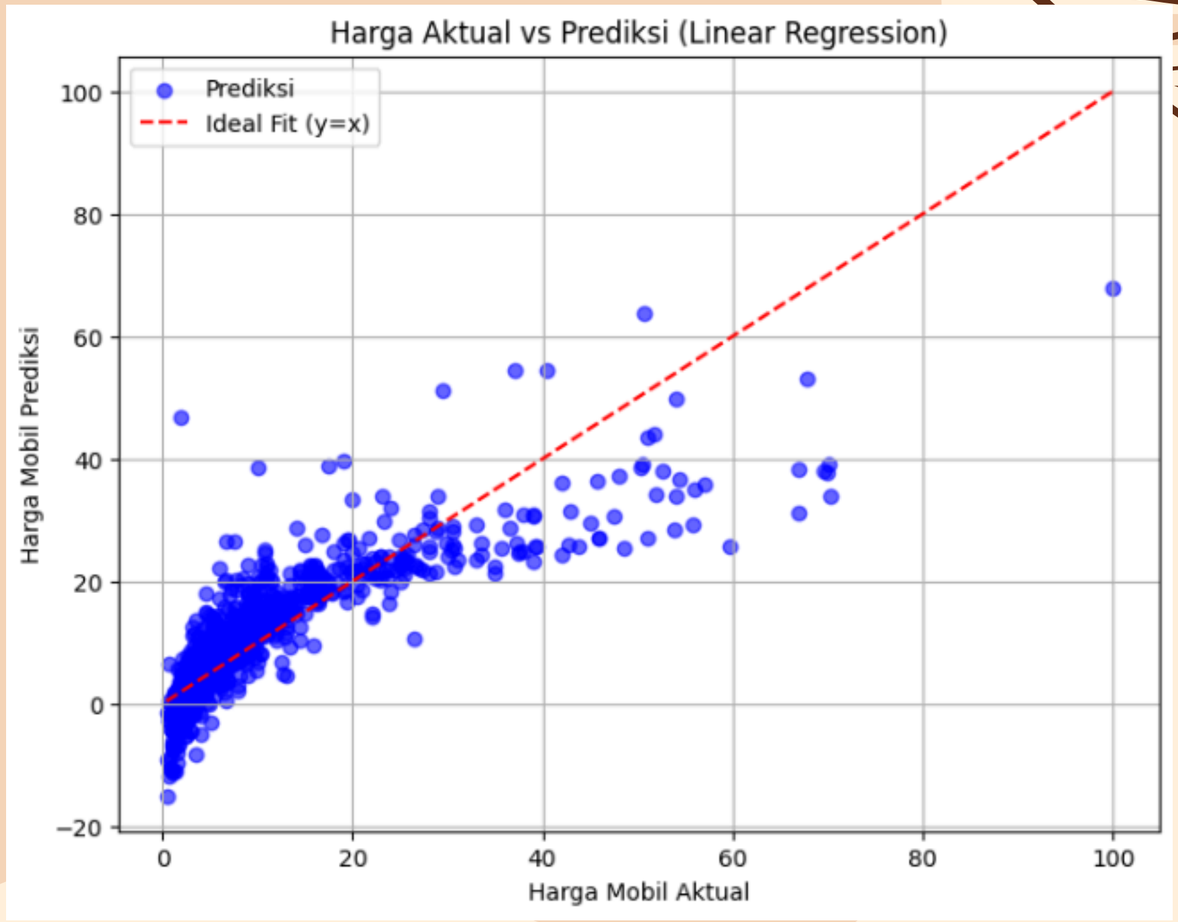
```
# Modelling dengan model Linear Regression
model_lr = LinearRegression()
model_lr.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

TAHAP EVALUASI MODEL

HASIL EVALUASI MODEL

Dari grafik ini menggambarkan perbandingan antara harga mobil yang sebenarnya dan prediksi yang dihasilkan oleh model Linear Regression. Dari penampilannya, sebagian besar titik berada tidak jauh dari garis yang seharusnya, sehingga dapat dikatakan bahwa model tersebut cukup tepat. Namun, terdapat beberapa titik yang tidak sesuai, kemungkinan disebabkan oleh fitur yang digunakan belum mampu mencakup semua aspek yang mempengaruhi harga mobil.



KESIMPULAN

Model Regresi Linear yang diterapkan dalam proyek ini dapat memahami hubungan antara berbagai karakteristik kendaraan, seperti merek, tahun produksi, jarak yang sudah ditempuh, dan kapasitas mesin terhadap harga jual mobil bekas. Setelah melalui langkah-langkah pra-pemrosesan, analisis data, serta penyesuaian skala fitur, model dilatih dan diuji. Hasil dari evaluasi menunjukkan bahwa model ini menunjukkan kinerja yang memuaskan, dengan sebagian besar nilai yang diprediksi mendekati harga nyata pada grafik perbandingan. Walaupun ada sedikit penyimpangan pada beberapa data yang ekstrem, secara keseluruhan model mampu memberikan estimasi harga yang cukup tepat.

LINK REPOSITORY GITHUB

https://github.com/AndikaHartanta/UTS_Data Mining_Andika-Hartanta_23067/tree/main

LINK LINKEDIN

https://www.linkedin.com/posts/andika-hartanta-157952346_used-cars-price-prediction-andika-hartanta-ugcPost-7381321096827965440-CXyt?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFajL1MBY94MEdcuX36-H_4Q2kyhscz8CgM

THANK YOU