Prediksi Harga Mobil Bekas Menggunakan Regresi Linear

Wildan Mufti Brawijaya - 2310631170121

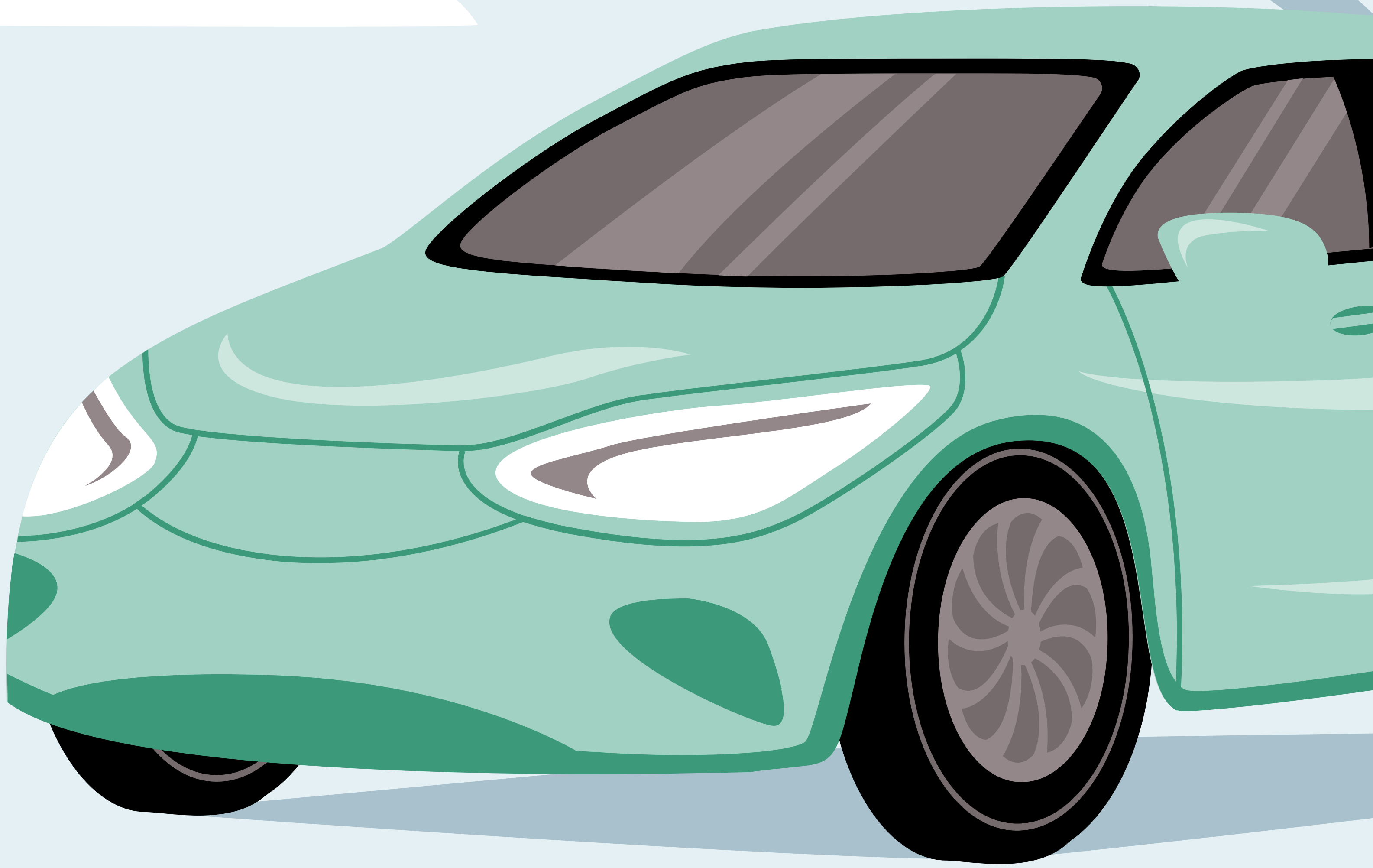


TABLE OF CONTENTS

01 Pendahuluan

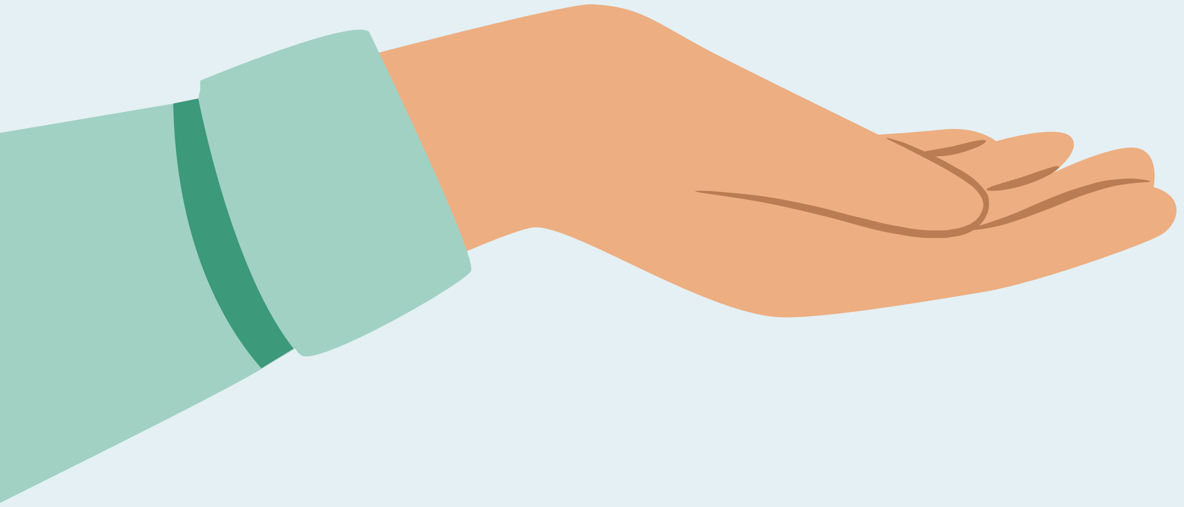
02 Data Profiling

03 Pre - processing Data

04 Modelling + Evaluasi

05 Kesimpulan

01 Pendahuluan



Pendahuluan

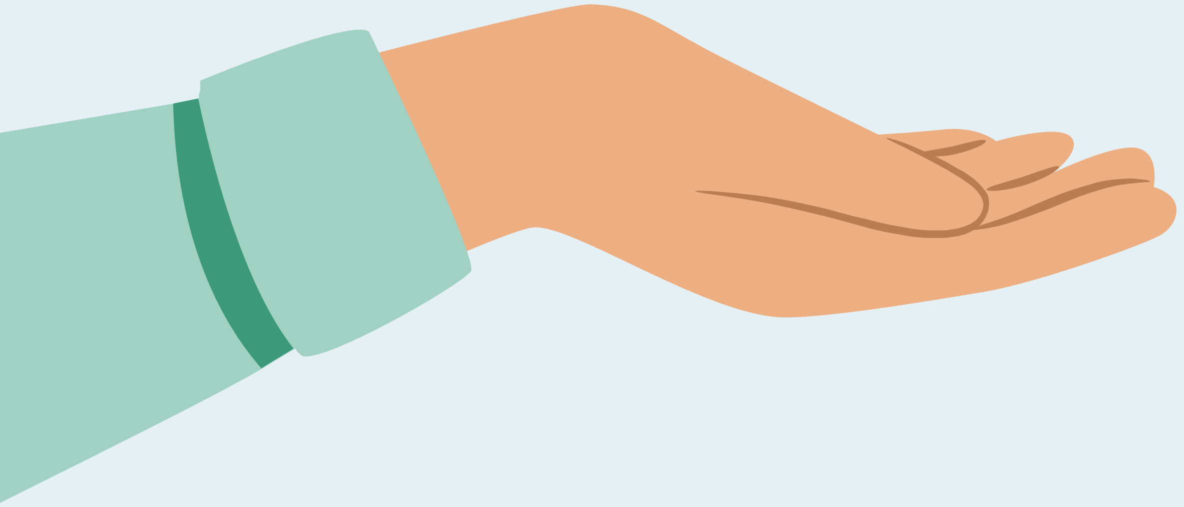
Tujuan

- Membangun model machine learning untuk memprediksi variabel Price (harga) berdasarkan fitur-fitur yang paling relevan dari mobil bekas.

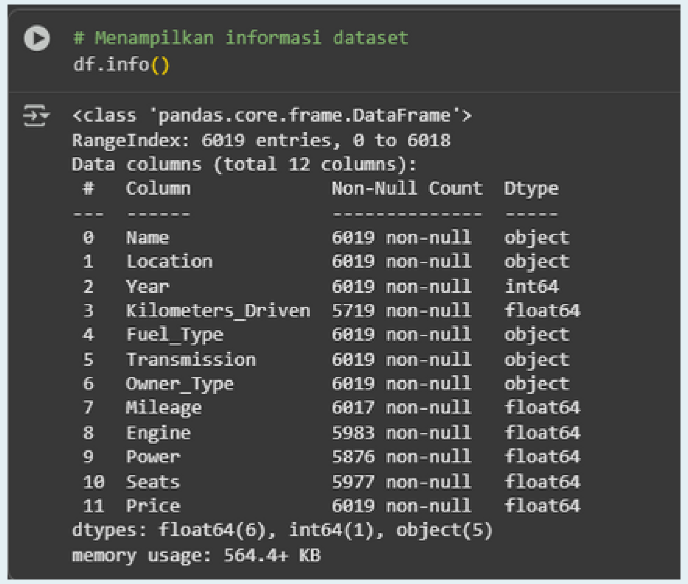
Metode

- Feature Selection : Forward Selection
- Regresi Linear

02
Data Profiling



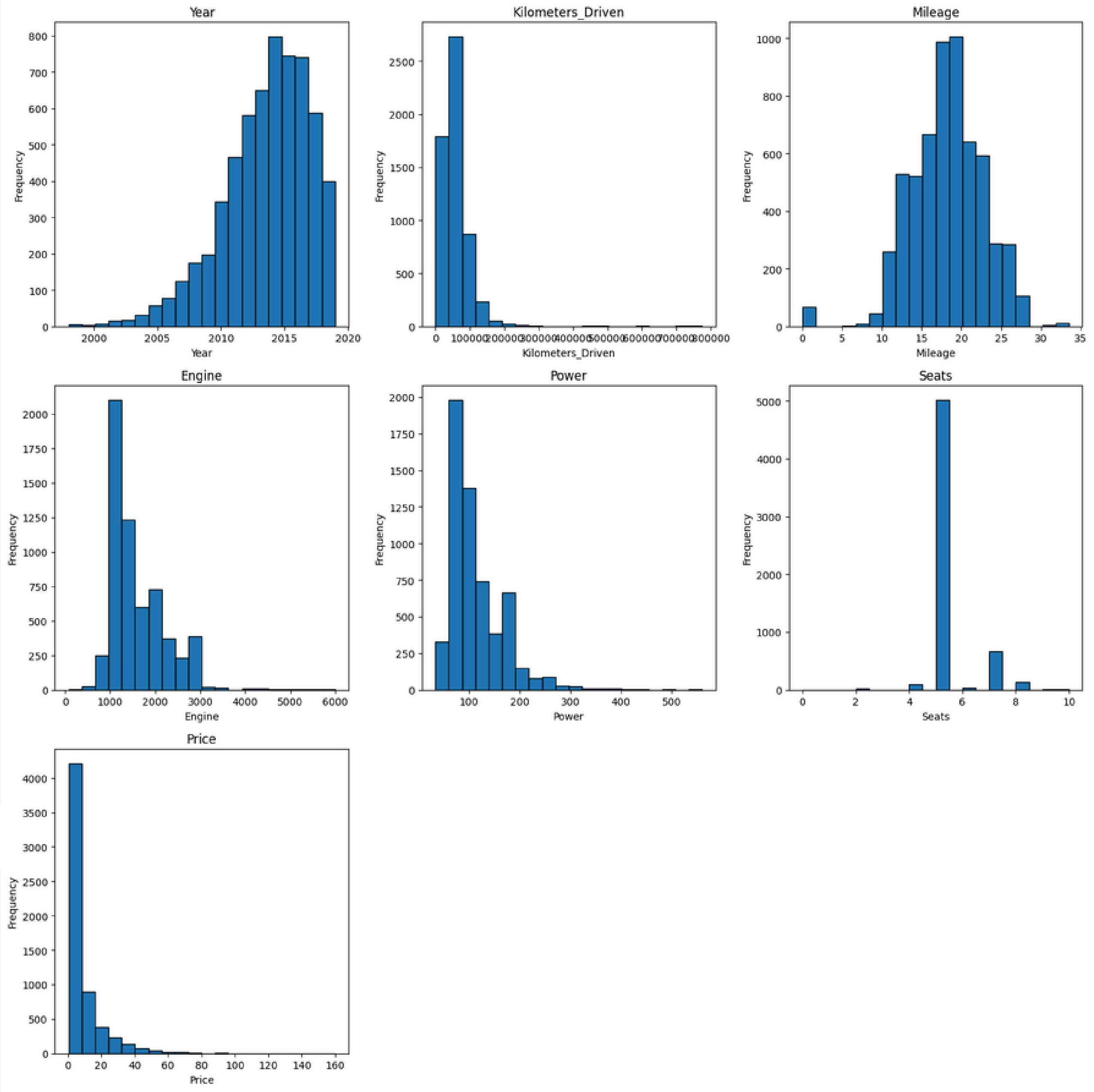
Data Profiling & EDA



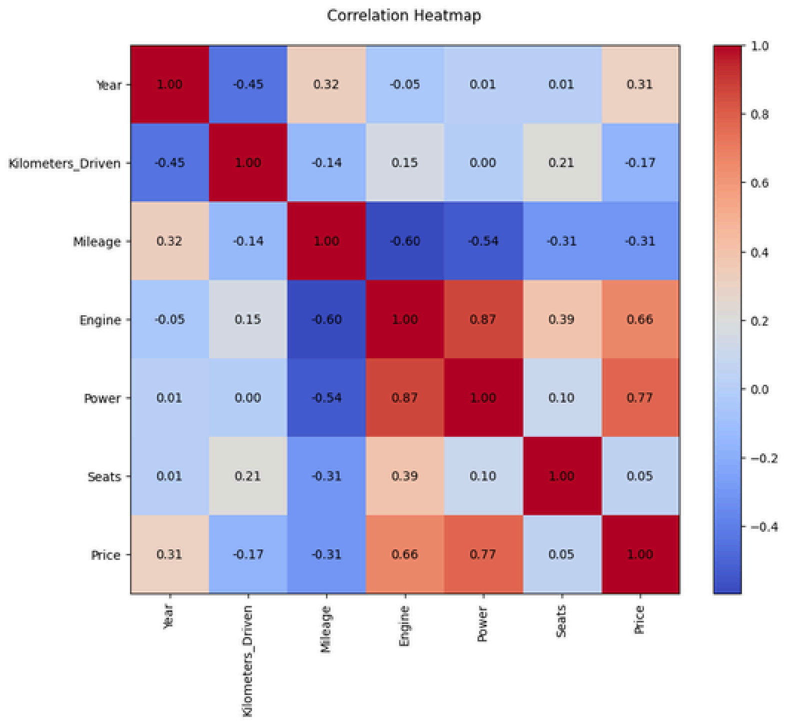
```
# Menampilkan informasi dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6019 entries, 0 to 6018
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               6019 non-null   object 
 1   Location           6019 non-null   object 
 2   Year               6019 non-null   int64  
 3   Kilometers_Driven  5719 non-null   float64
 4   Fuel_Type          6019 non-null   object 
 5   Transmission       6019 non-null   object 
 6   Owner_Type         6019 non-null   object 
 7   Mileage            6017 non-null   float64
 8   Engine             5983 non-null   float64
 9   Power              5876 non-null   float64
 10  Seats              5977 non-null   float64
 11  Price              6019 non-null   float64
dtypes: float64(6), int64(1), object(5)
memory usage: 564.4+ KB
```

Dataset memiliki 6019 baris dengan 12 kolom



Grafik Batang Kategorikal

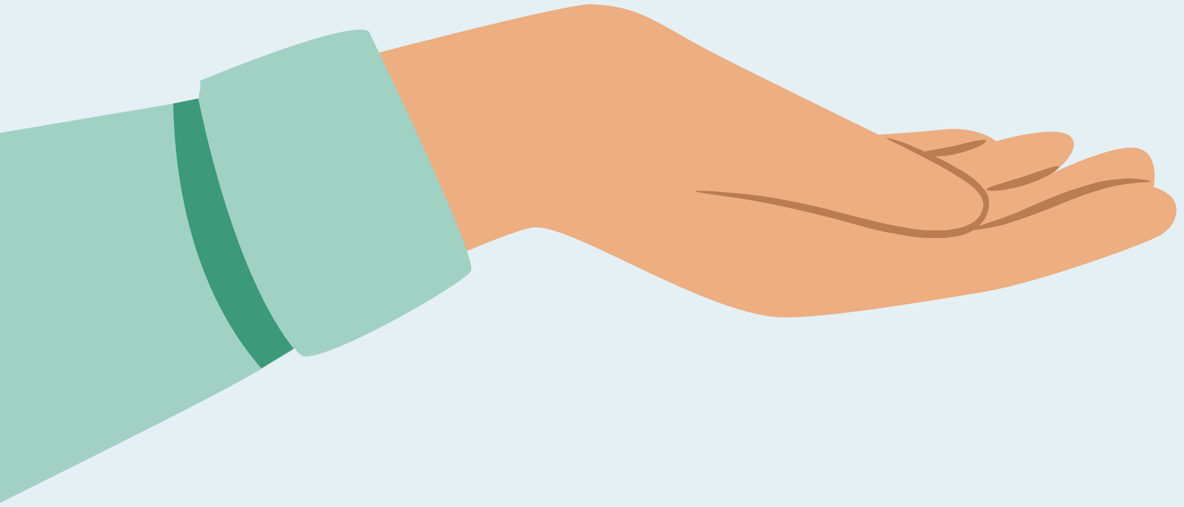


	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Price
Year	1.00	-0.45	0.32	-0.05	0.01	0.01	0.31
Kilometers_Driven	-0.45	1.00	-0.14	0.15	0.00	0.21	-0.17
Mileage	0.32	-0.14	1.00	-0.60	-0.54	-0.31	-0.31
Engine	-0.05	0.15	-0.60	1.00	0.87	0.39	0.66
Power	0.01	0.00	-0.54	0.87	1.00	0.10	0.77
Seats	0.01	0.21	-0.31	0.39	0.10	1.00	0.05
Price	0.31	-0.17	-0.31	0.66	0.77	0.05	1.00

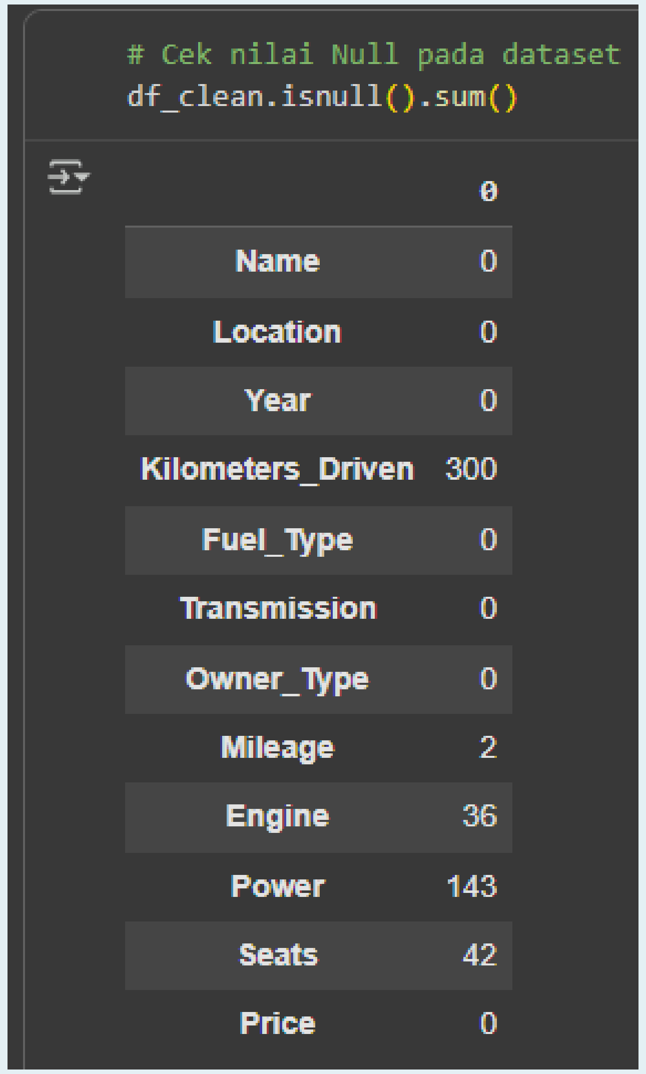
Heatmap Korelasi

03

Pre - Processing

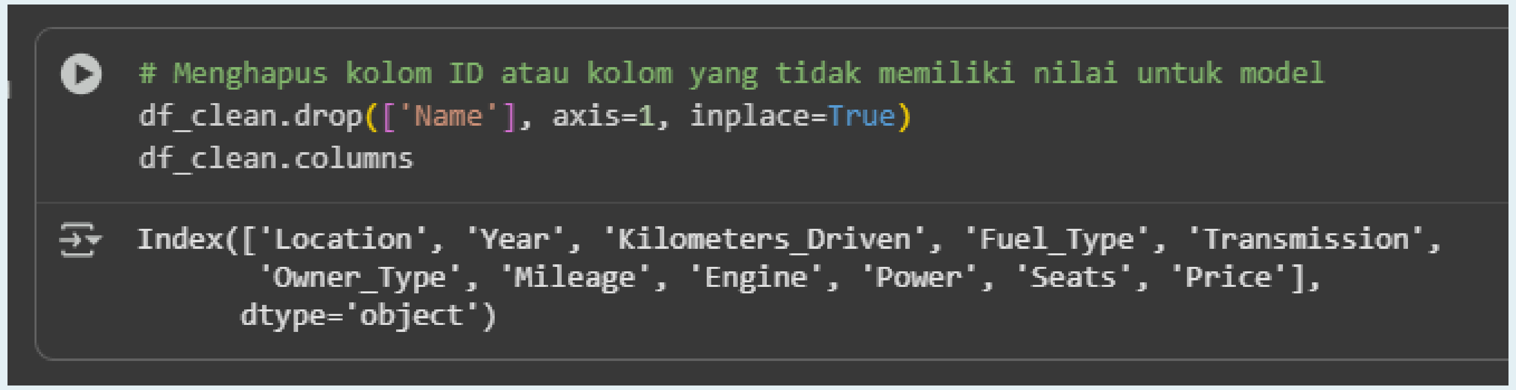


Pre - Processing



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

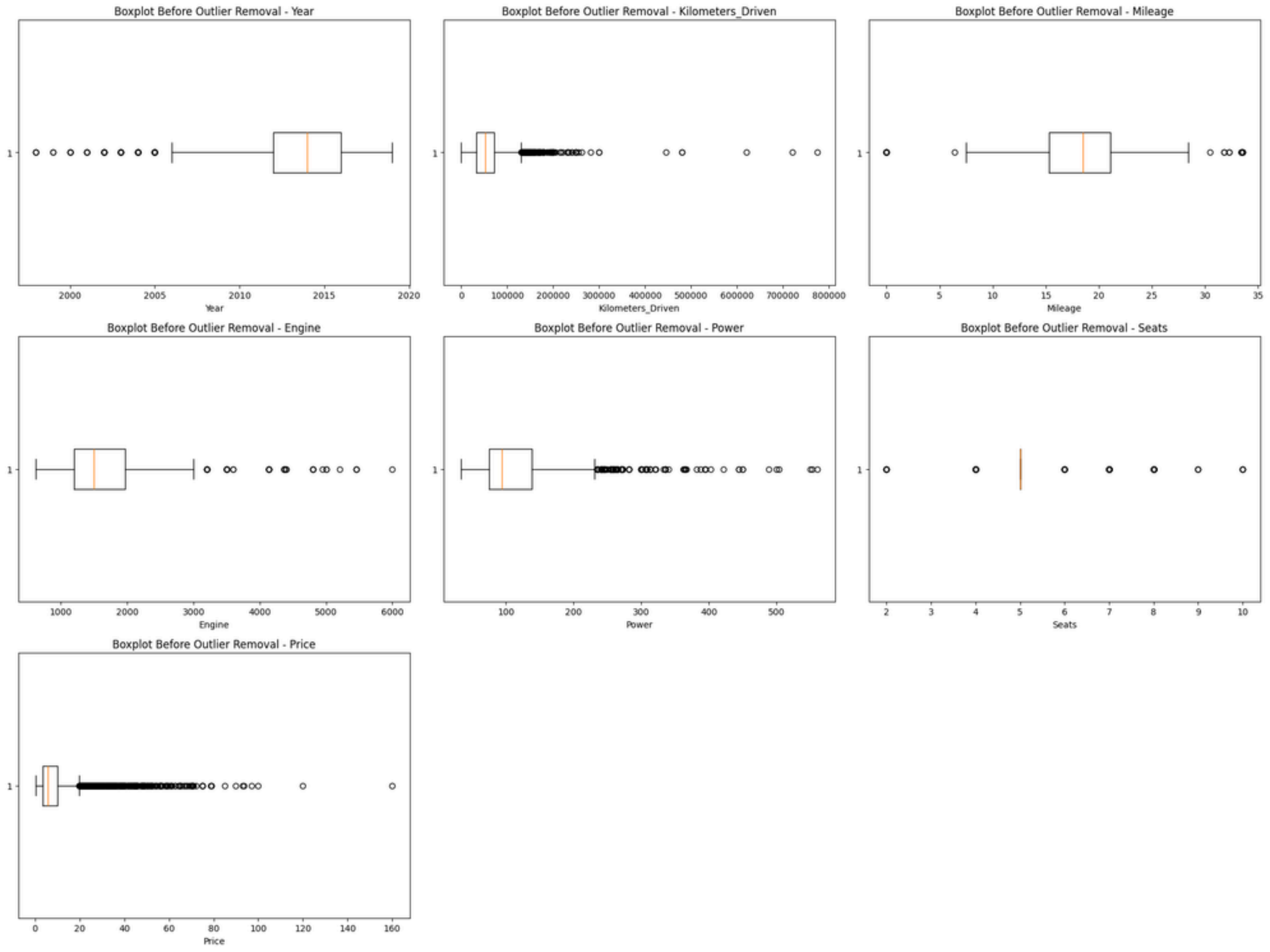
	0
Name	0
Location	0
Year	0
Kilometers_Driven	300
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	2
Engine	36
Power	143
Seats	42
Price	0

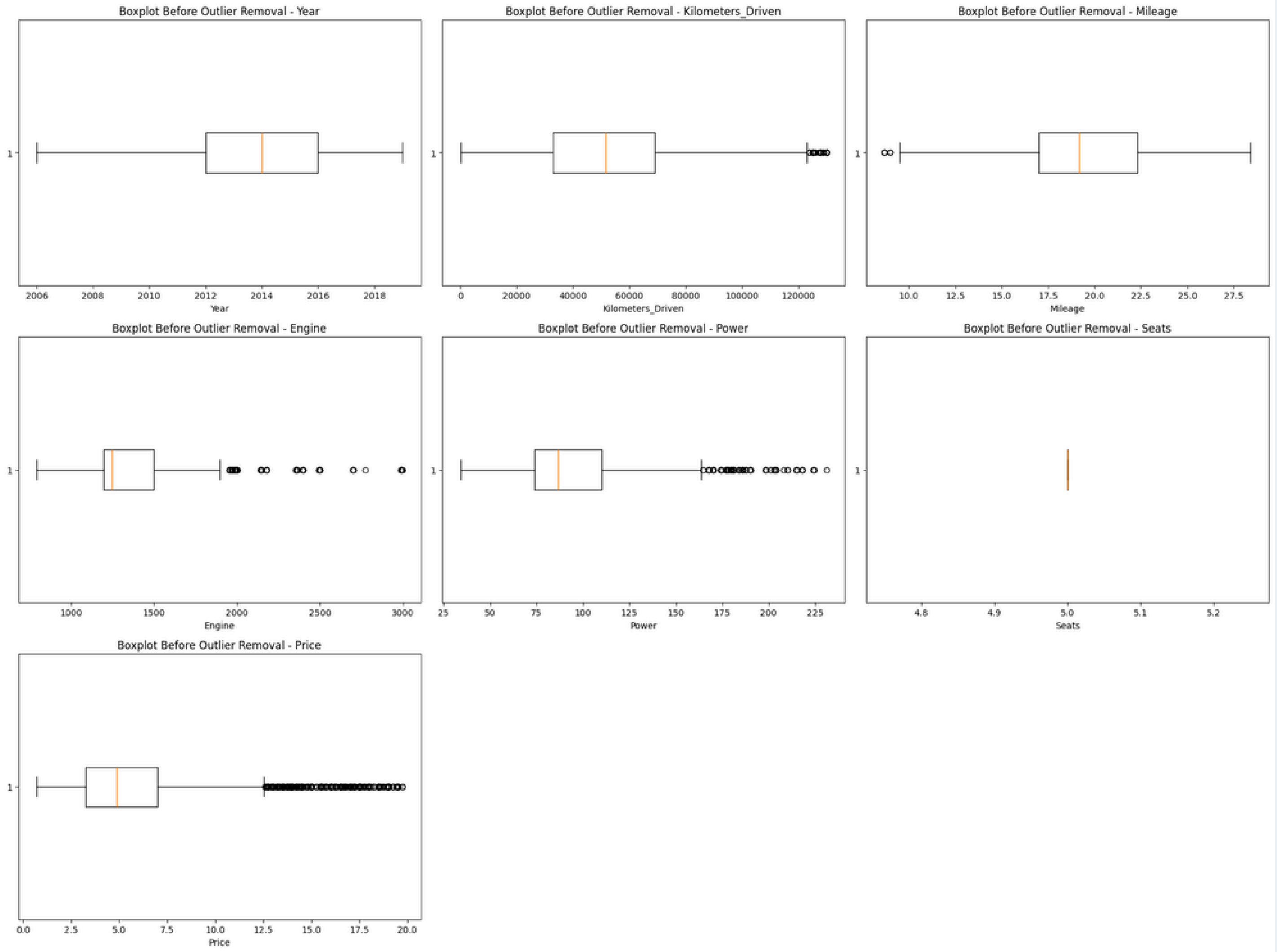


```python
# Menghapus kolom ID atau kolom yang tidak memiliki nilai untuk model
df_clean.drop(['Name'], axis=1, inplace=True)
df_clean.columns
```

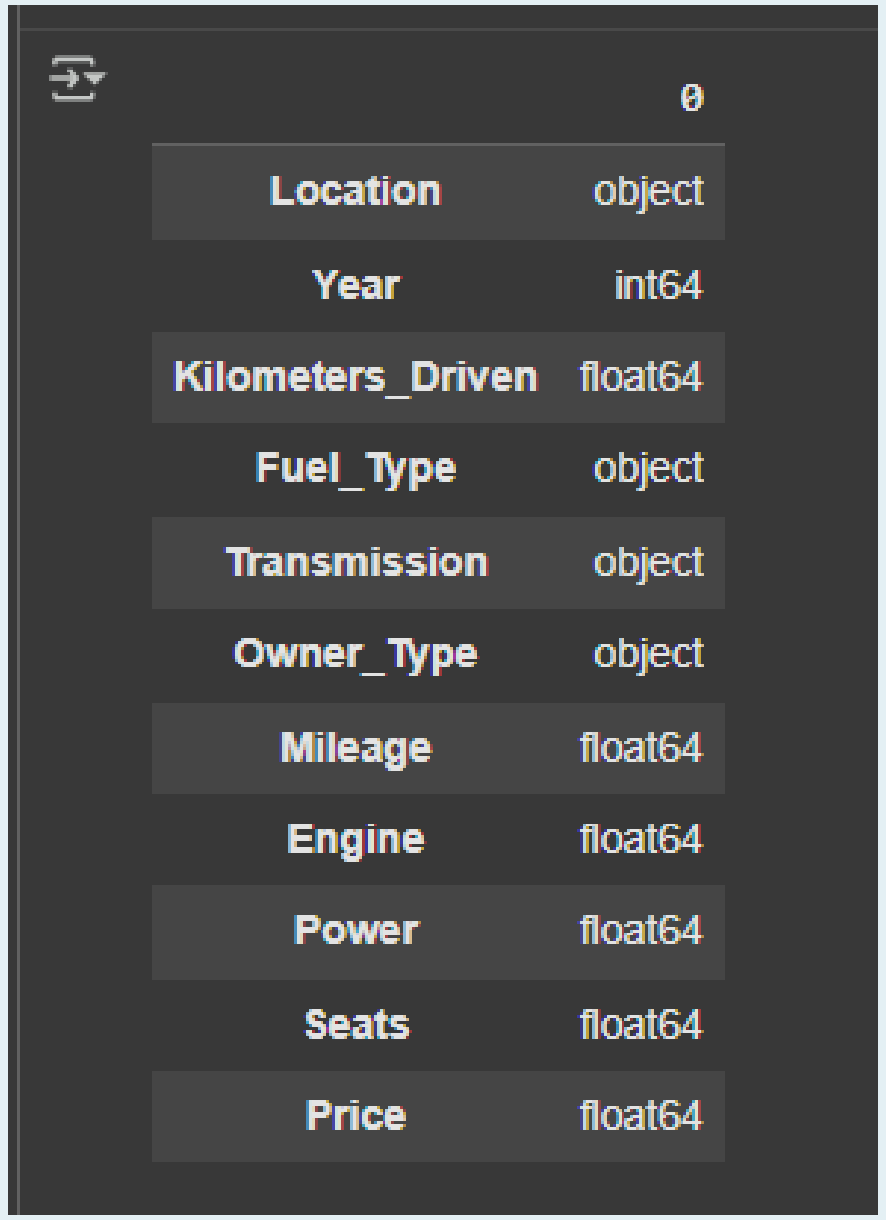
```
Index(['Location', 'Year', 'Kilometers_Driven', 'Fuel_Type', 'Transmission',
       'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats', 'Price'],
      dtype='object')
```

Pre - Processing

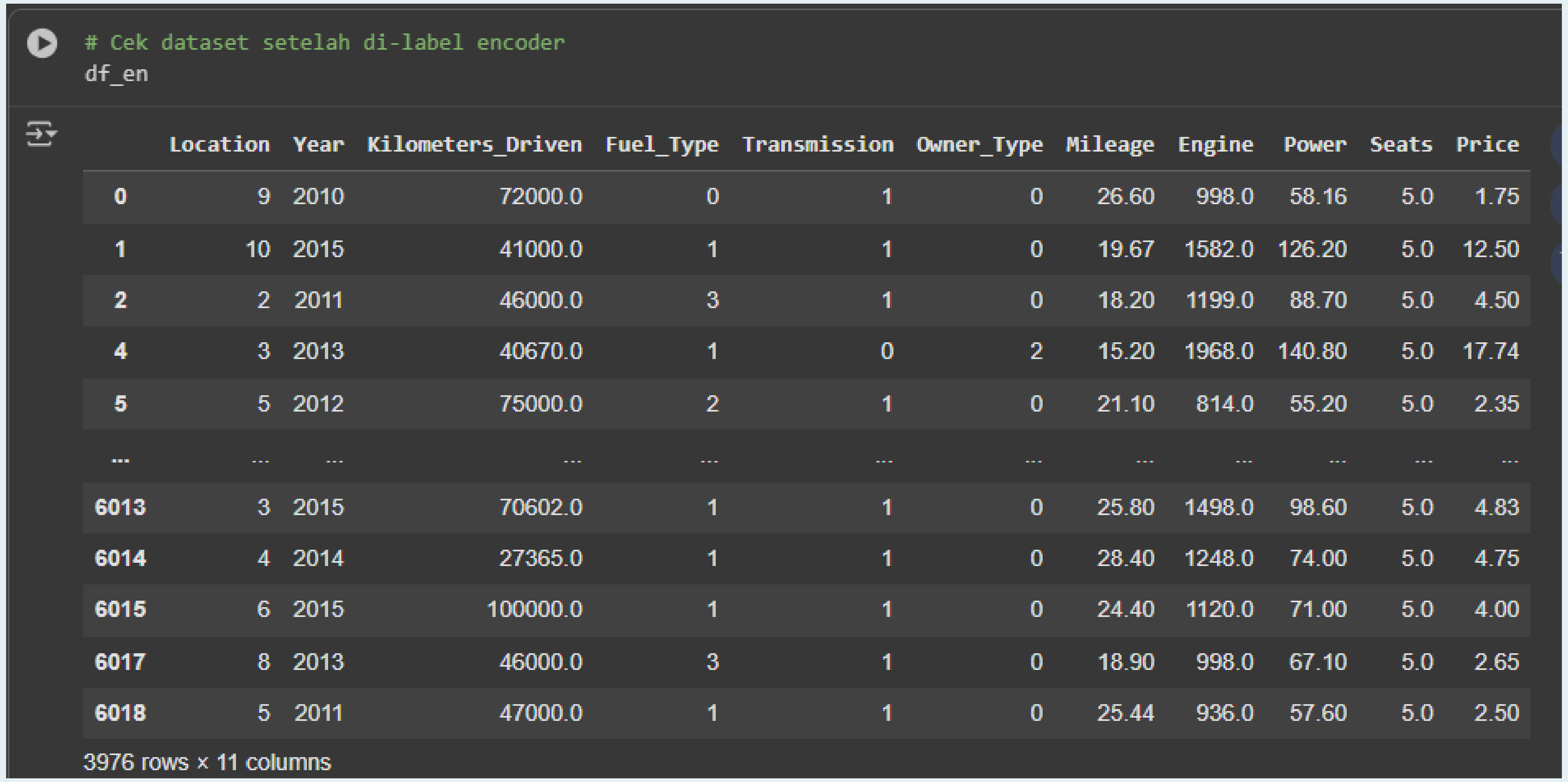




Pre - Processing



	0
Location	object
Year	int64
Kilometers_Driven	float64
Fuel_Type	object
Transmission	object
Owner_Type	object
Mileage	float64
Engine	float64
Power	float64
Seats	float64
Price	float64

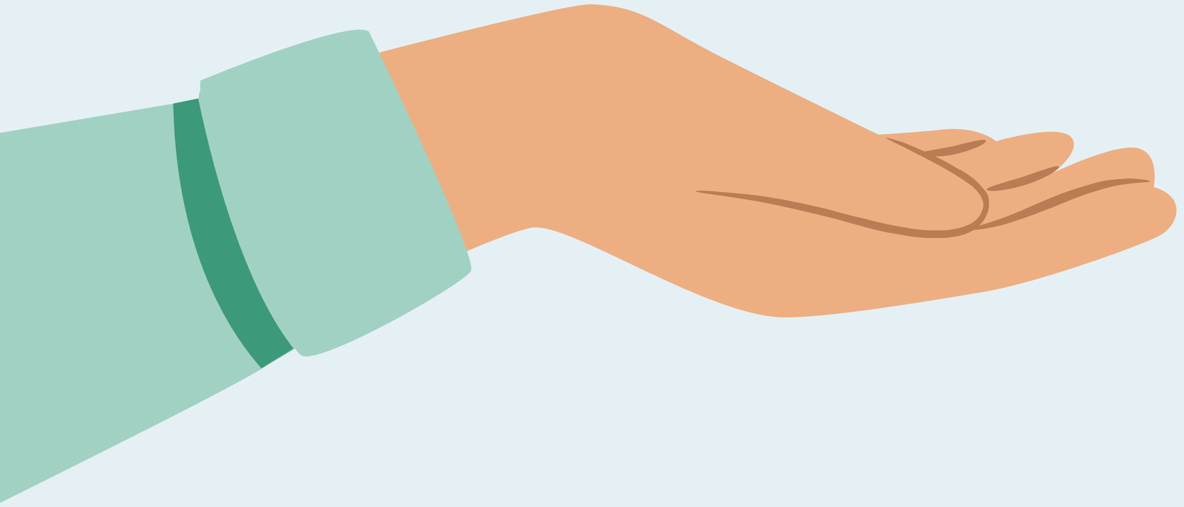


```
# Cek dataset setelah di-label encoder
df_en
```

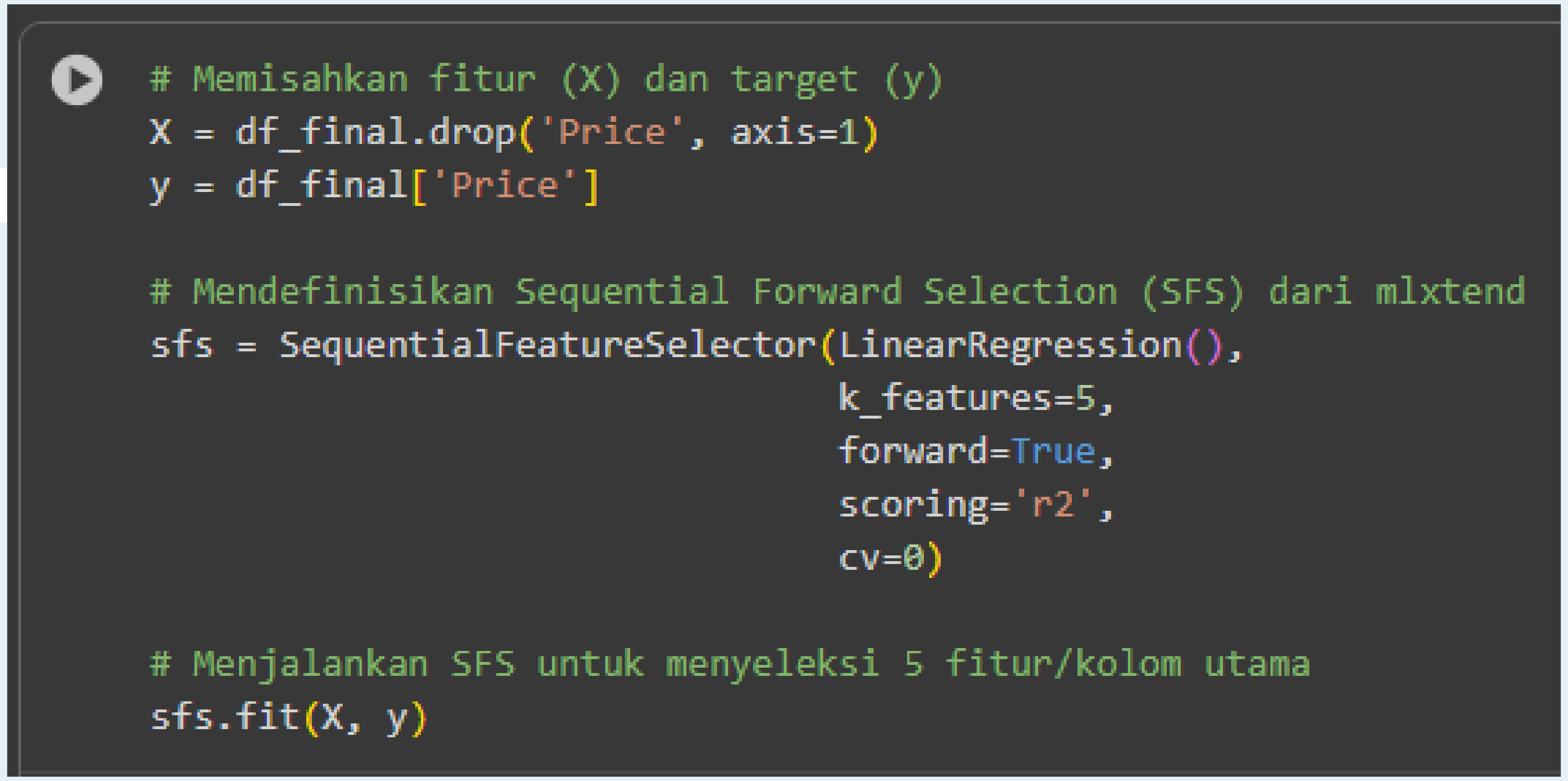
	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	9	2010	72000.0	0	1	0	26.60	998.0	58.16	5.0	1.75
1	10	2015	41000.0	1	1	0	19.67	1582.0	126.20	5.0	12.50
2	2	2011	46000.0	3	1	0	18.20	1199.0	88.70	5.0	4.50
4	3	2013	40670.0	1	0	2	15.20	1968.0	140.80	5.0	17.74
5	5	2012	75000.0	2	1	0	21.10	814.0	55.20	5.0	2.35
...	...	...	...	...	...	...	...	...	...	...	...
6013	3	2015	70602.0	1	1	0	25.80	1498.0	98.60	5.0	4.83
6014	4	2014	27365.0	1	1	0	28.40	1248.0	74.00	5.0	4.75
6015	6	2015	100000.0	1	1	0	24.40	1120.0	71.00	5.0	4.00
6017	8	2013	46000.0	3	1	0	18.90	998.0	67.10	5.0	2.65
6018	5	2011	47000.0	1	1	0	25.44	936.0	57.60	5.0	2.50

3976 rows × 11 columns

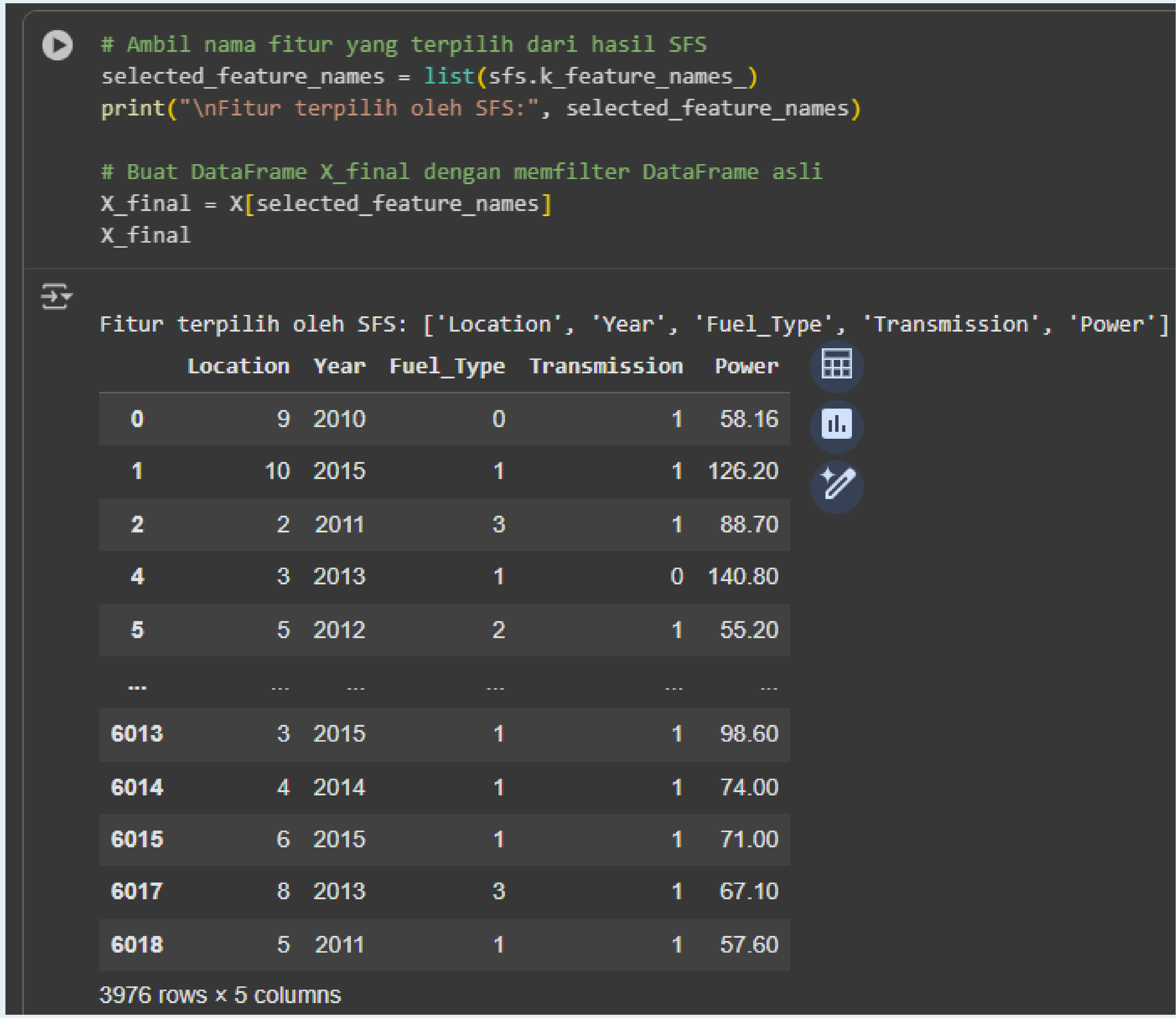
04
Modelling + Evaluasi



Modelling + Evaluasi



```python
# Memisahkan fitur (X) dan target (y)
X = df_final.drop('Price', axis=1)
y = df_final['Price']

# Mendefinisikan Sequential Forward Selection (SFS) dari mlxtend
sfs = SequentialFeatureSelector(LinearRegression(),
                                k_features=5,
                                forward=True,
                                scoring='r2',
                                cv=0)

# Menjalankan SFS untuk menyeleksi 5 fitur/kolom utama
sfs.fit(X, y)
```



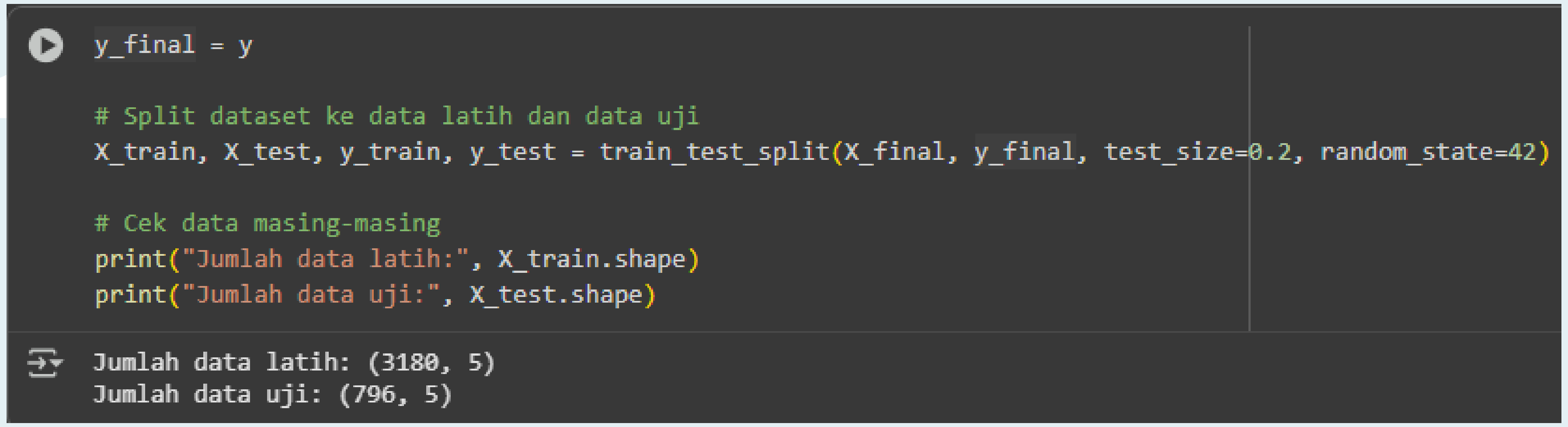
```python
# Ambil nama fitur yang terpilih dari hasil SFS
selected_feature_names = list(sfs.k_feature_names_)
print("\nFitur terpilih oleh SFS:", selected_feature_names)

# Buat DataFrame X_final dengan memfilter DataFrame asli
X_final = X[selected_feature_names]
X_final
```

```
Fitur terpilih oleh SFS: ['Location', 'Year', 'Fuel_Type', 'Transmission', 'Power']
```

	Location	Year	Fuel_Type	Transmission	Power
0	9	2010	0	1	58.16
1	10	2015	1	1	126.20
2	2	2011	3	1	88.70
4	3	2013	1	0	140.80
5	5	2012	2	1	55.20
...	...	...	...	...	...
6013	3	2015	1	1	98.60
6014	4	2014	1	1	74.00
6015	6	2015	1	1	71.00
6017	8	2013	3	1	67.10
6018	5	2011	1	1	57.60

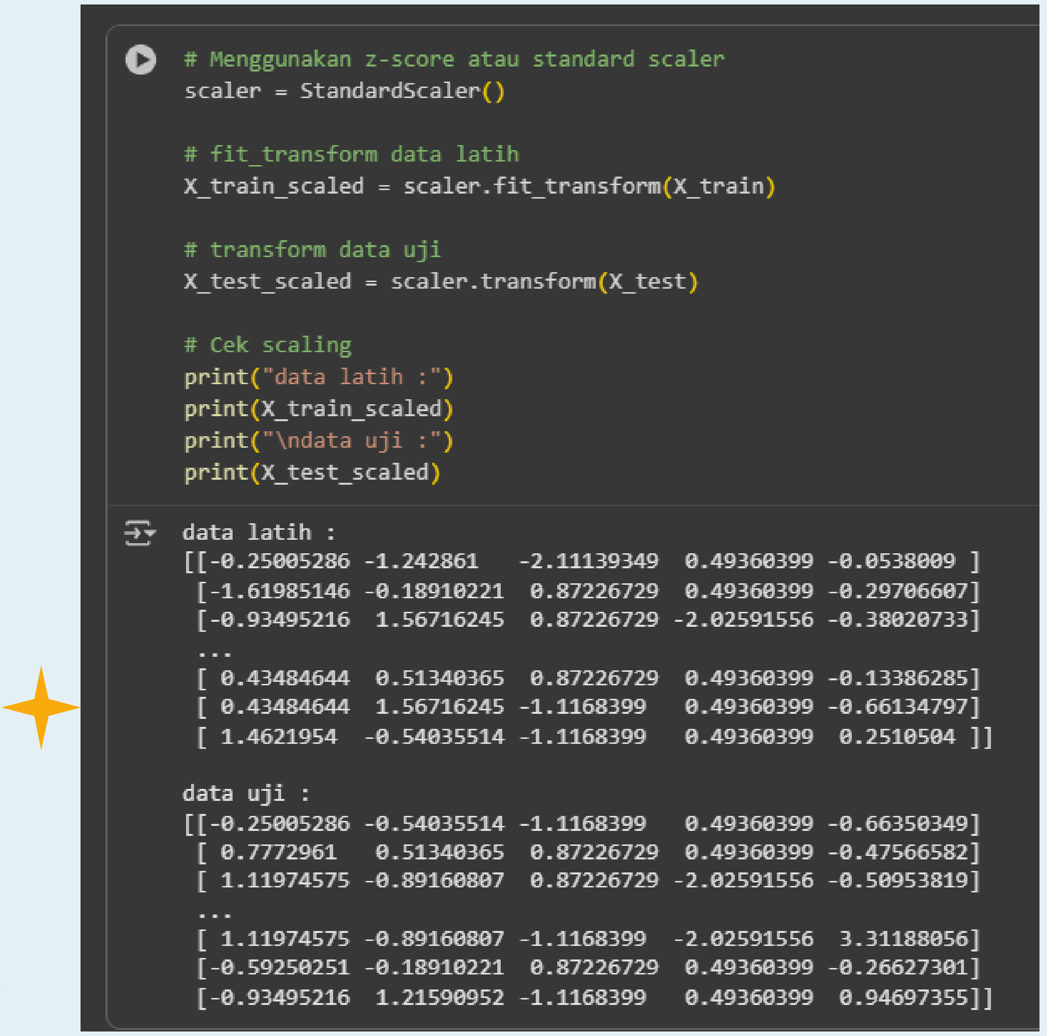
3976 rows × 5 columns

Modelling + Evaluasi



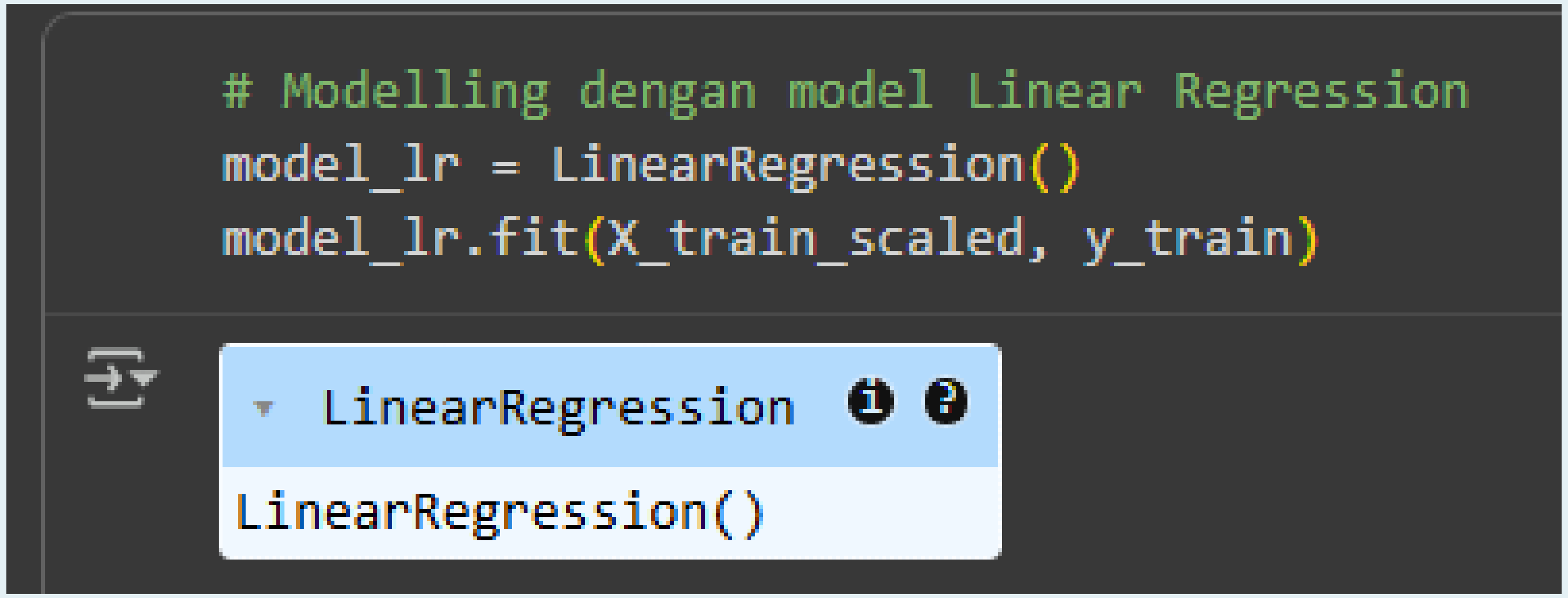
```
y_final = y

# Split dataset ke data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X_final, y_final, test_size=0.2, random_state=42)

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

```
Jumlah data latih: (3180, 5)
Jumlah data uji: (796, 5)
```



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

# fit_transform data latih
X_train_scaled = scaler.fit_transform(X_train)

# transform data uji
X_test_scaled = scaler.transform(X_test)

# Cek scaling
print("data latih :")
print(X_train_scaled)
print("\ndata uji :")
print(X_test_scaled)
```

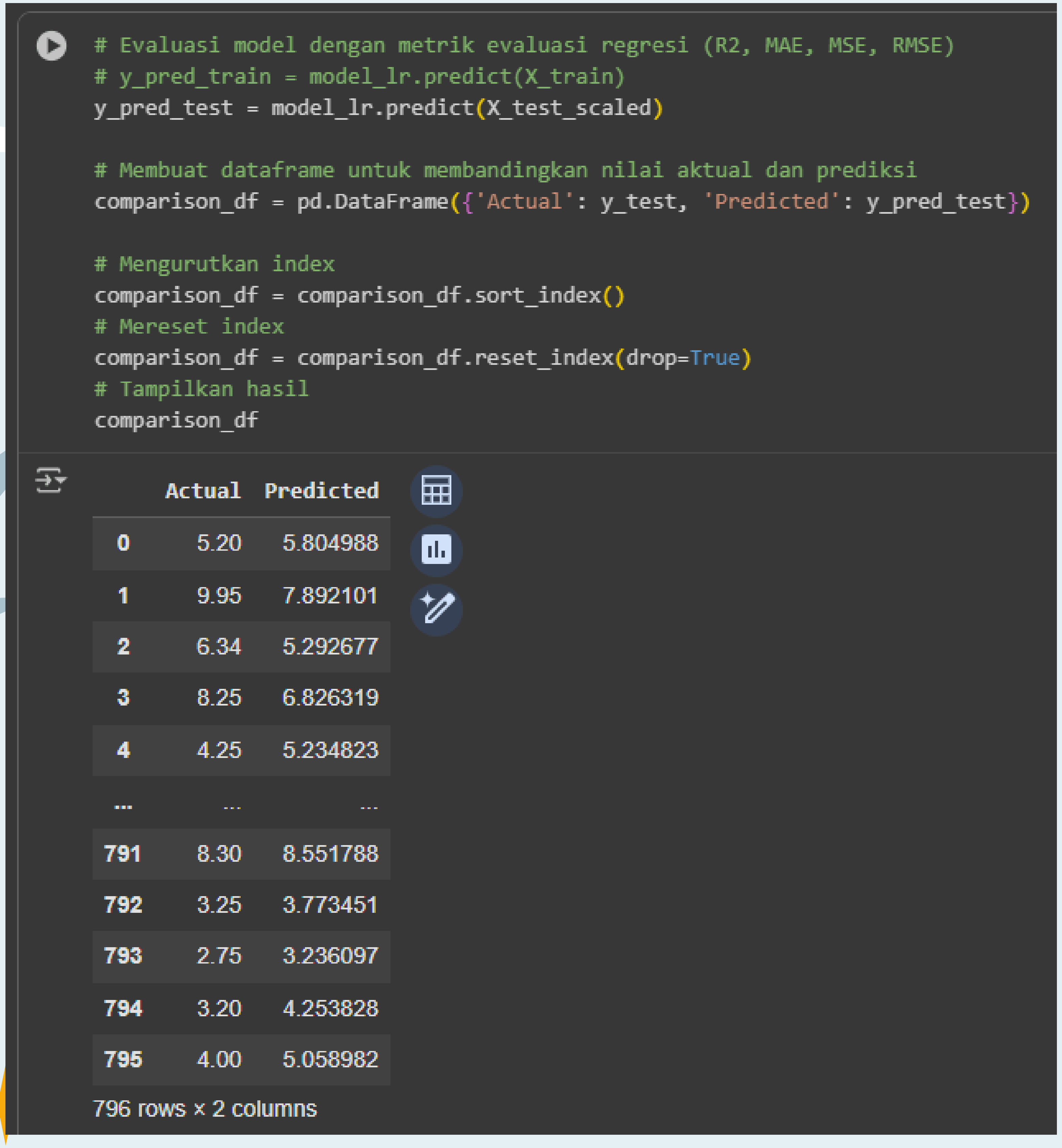
```
data latih :
[[-0.25005286 -1.242861   -2.11139349  0.49360399 -0.0538009 ]
 [-1.61985146 -0.18910221  0.87226729  0.49360399 -0.29706607]
 [-0.93495216  1.56716245  0.87226729 -2.02591556 -0.38020733]
 ...
 [ 0.43484644  0.51340365  0.87226729  0.49360399 -0.13386285]
 [ 0.43484644  1.56716245 -1.1168399   0.49360399 -0.66134797]
 [ 1.4621954  -0.54035514 -1.1168399   0.49360399  0.2510504 ]]

data uji :
[[-0.25005286 -0.54035514 -1.1168399   0.49360399 -0.66350349]
 [ 0.7772961   0.51340365  0.87226729  0.49360399 -0.47566582]
 [ 1.11974575 -0.89160807  0.87226729 -2.02591556 -0.50953819]
 ...
 [ 1.11974575 -0.89160807 -1.1168399  -2.02591556  3.31188056]
 [-0.59250251 -0.18910221  0.87226729  0.49360399 -0.26627301]
 [-0.93495216  1.21590952 -1.1168399   0.49360399  0.94697355]]
```



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

```
LinearRegression
LinearRegression()
```

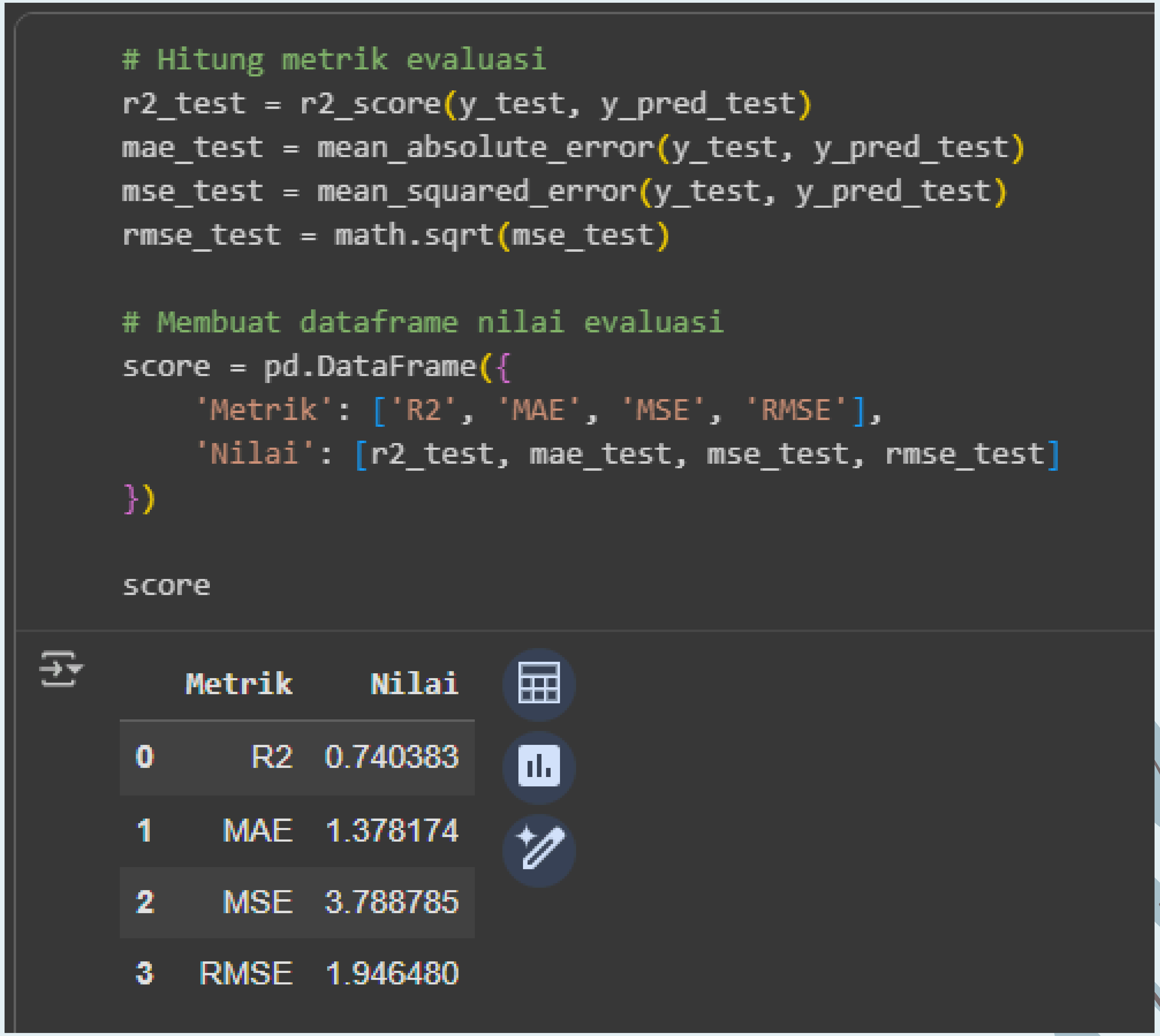
Modelling + Evaluasi



```python
# Evaluasi model dengan metrik evaluasi regresi (R2, MAE, MSE, RMSE)
# y_pred_train = model_lr.predict(X_train)
y_pred_test = model_lr.predict(X_test_scaled)

# Membuat dataframe untuk membandingkan nilai aktual dan prediksi
comparison_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred_test})

# Mengurutkan index
comparison_df = comparison_df.sort_index()
# Mereset index
comparison_df = comparison_df.reset_index(drop=True)
# Tampilkan hasil
comparison_df
```

	Actual	Predicted
0	5.20	5.804988
1	9.95	7.892101
2	6.34	5.292677
3	8.25	6.826319
4	4.25	5.234823
...	...	...
791	8.30	8.551788
792	3.25	3.773451
793	2.75	3.236097
794	3.20	4.253828
795	4.00	5.058982

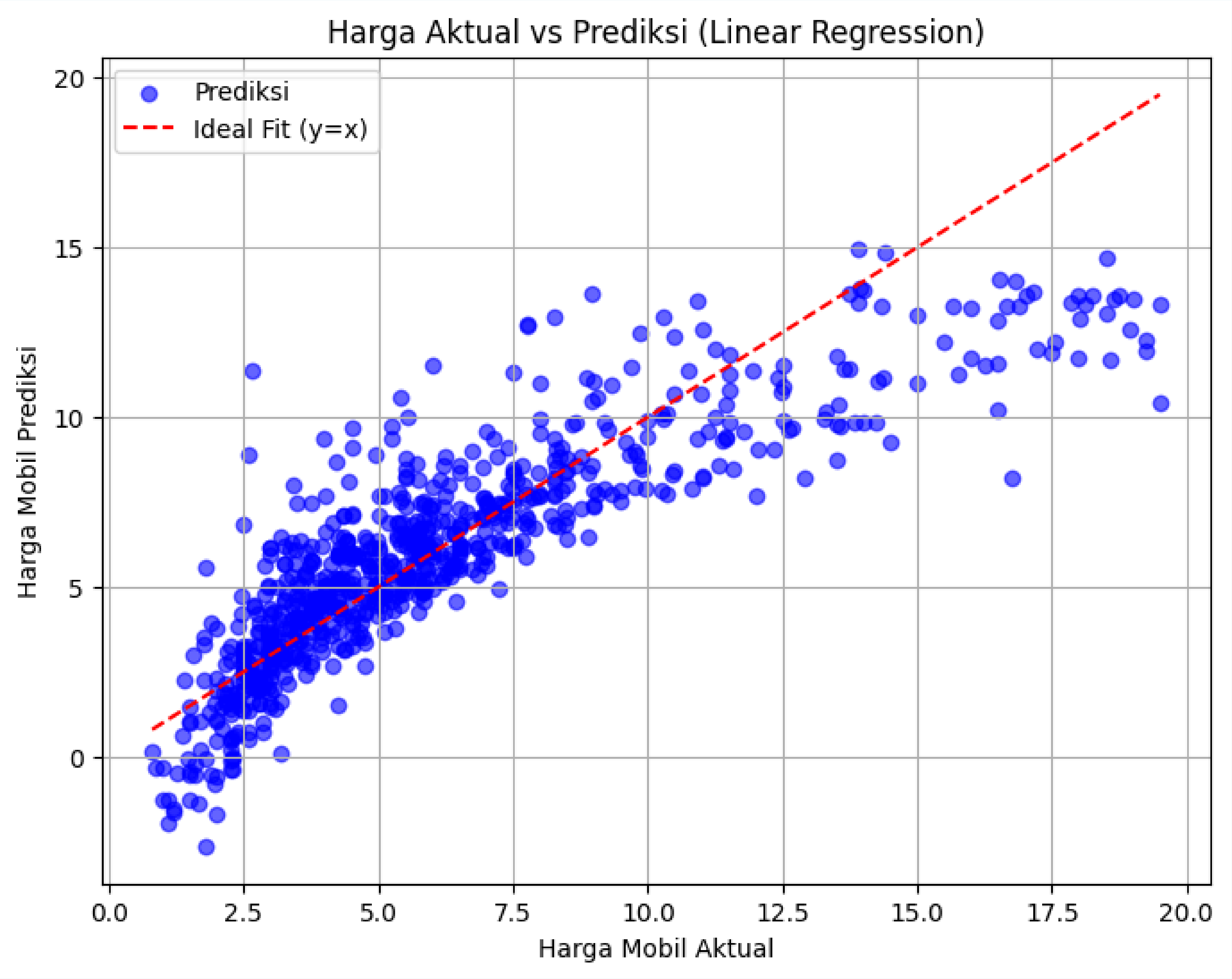
796 rows × 2 columns



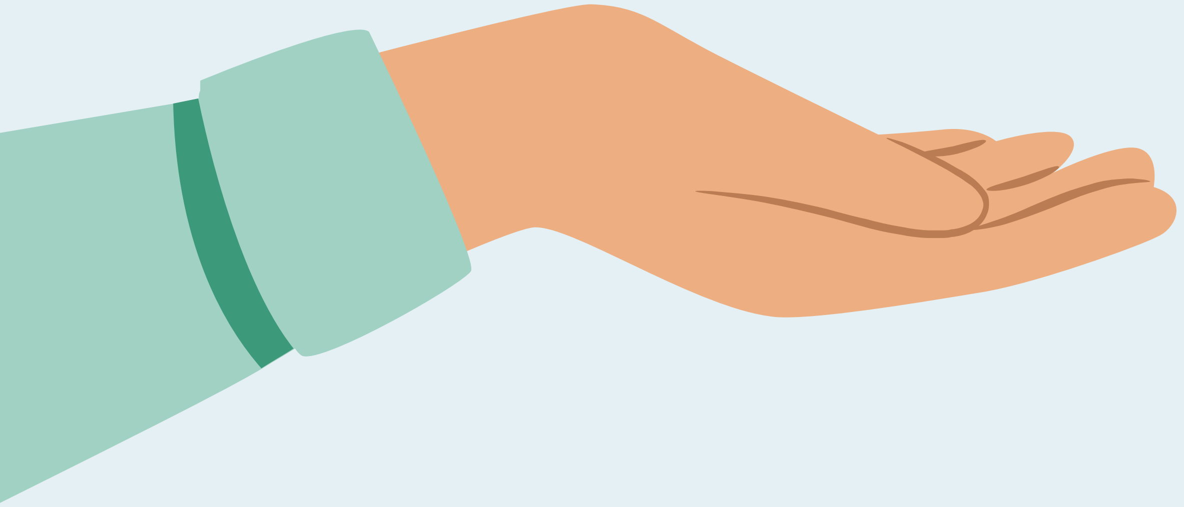
```python
# Hitung metrik evaluasi
r2_test = r2_score(y_test, y_pred_test)
mae_test = mean_absolute_error(y_test, y_pred_test)
mse_test = mean_squared_error(y_test, y_pred_test)
rmse_test = math.sqrt(mse_test)

# Membuat dataframe nilai evaluasi
score = pd.DataFrame({
    'Metrik': ['R2', 'MAE', 'MSE', 'RMSE'],
    'Nilai': [r2_test, mae_test, mse_test, rmse_test]
})

score
```

	Metrik	Nilai
0	R2	0.740383
1	MAE	1.378174
2	MSE	3.788785
3	RMSE	1.946480

Modelling + Evaluasi



05
Kesimpulan



Kesimpulan

- Model Regresi Linear dengan 5 fitur terpilih berhasil dibuat dan mampu memprediksi harga mobil bekas dengan tingkat akurasi yang baik (R^2 = 74%).

- Performa model dapat ditingkatkan dengan mencoba algoritma lain atau menggunakan teknik feature engineering yang lebih kompleks.

THANKS!