

LINEAR DAN POLYNOMIAL REGRESSION

CAR PRICE PREDICTION

KELOMPOK 2

Alvia Zuhra	(2208107010003)
Al-Mahfuzh Fadhlur Rohman	(2208107010016)
Nurul Uzratun Nashriyyah	(2208107010030)
Ganang Setyo Hadi	(2208107010052)
Azimah Al-Huda	(2208107010069)



PEMAHAMAN DATASET

- o Menjelaskan sumber data dan variabel yang digunakan.
- o Menampilkan statistik deskriptif dan visualisasi awal data.

PEMAHAMAN DATASET

DESKRIPSI DATASET

Dataset ini berasal dari platform Kaggle dengan judul "Cars4u" yang disediakan oleh Sukhmani Bedi. Dataset ini berisi informasi tentang 7.253 mobil bekas yang dijual di India, mencakup 14 variabel yang menjelaskan berbagai karakteristik mobil.

VARIABEL YANG DIGUNAKAN

No.	Nama Variabel	Tipe Data	Deskripsi
1	S.No.	Integer	Nomor urut data sebagai pengidentifikasi unik setiap entri
2	Name	String	Nama mobil yang mencakup merek dan model
3	Location	String	Lokasi di mana mobil dijual atau tersedia untuk dibeli
4	Year	Integer	Tahun pembuatan mobil
5	Kilometers_Driven	Integer	Total kilometer yang telah ditempuh mobil (dalam KM)
6	Fuel_Type	String	Jenis bahan bakar (Petrol, Diesel, Electric, CNG, LPG)
7	Transmission	String	Jenis transmisi (Otomatis/Manual)
8	Owner_Type	String	Tipe kepemilikan mobil
9	Mileage	String	Efisiensi bahan bakar dalam kmpl atau km/kg
10	Engine	String	Volume mesin dalam satuan CC
11	Power	String	Tenaga maksimum mesin dalam satuan bhp
12	Seats	Float	Jumlah kursi dalam mobil
13	New_Price	String	Harga mobil baru dengan model yang sama (dalam Lakh Rupee)
14	Price	Float	Harga mobil bekas (dalam Lakh Rupee, 1 Lakh = 100.000)

PEMAHAMAN DATASET

STATISTIK DESKRIPTIF VISUALISASI AWAL

1. Melihat 5 baris pertama dataset

S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.00000	NaN 1.75000
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.00000	NaN 12.50000
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.00000	8.61 Lakh 4.50000
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.00000	NaN 6.00000
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.00000	NaN 17.74000

3. Memperbaiki Struktur Data

- Mengonversi kolom Mileage, Engine, dan Power menjadi numerik.
- Membersihkan dan mengonversi kolom New_Price yang bertipe object.
- Memastikan kolom Seats tidak mengandung nilai kosong dan sudah sesuai dalam format float.

2. Melihat Informasi Dasar Dataset

Informasi Dasar Dataset:
Jumlah Baris: 7253
Jumlah Kolom: 14
Ukuran Memory: 0.77 MB

Tipe Data Masing-masing Kolom:
S.No. int64
Name object
Location object
Year int64
Kilometers_Driven int64
Fuel_Type object
Transmission object
Owner_Type object
Mileage object
Engine object
Power object
Seats float64
New_Price object
Price float64

dtype: object

PEMAHAMAN DATASET

STATISTIK DESKRIPTIF VISUALISASI AWAL

4. Statistik Deskriptif untuk Kolom Numerik

Statistik Deskriptif untuk Kolom Numerik:										
	count	mean	std	min	25%	50%	75%	max	range	
New_Price	1005.00000	2278884.57711	2777164.54159	391000.00000	788000.00000	1156000.00000	2614000.00000	3750000.00000	37109000.00000	
Kilometers_Driven	7252.00000	58700.26269	84433.48037	171.00000	34000.00000	53429.00000	73000.00000	6500000.00000	6499829.00000	
S.No.	7252.00000	3625.88693	2094.02732	0.00000	1812.75000	3625.50000	5439.25000	7252.00000	7252.00000	
Engine	7206.00000	1616.78782	595.04824	624.00000	1198.00000	1493.00000	1968.00000	5998.00000	5374.00000	
Power	7077.00000	112.77535	53.49053	34.20000	75.00000	94.00000	138.10000	616.00000	581.80000	
Price	6018.00000	9.47888	11.18875	0.44000	3.50000	5.64000	9.95000	160.00000	159.56000	
Mileage	7170.00000	18.34653	4.15791	6.40000	15.30000	18.20000	21.10000	33.54000	27.14000	
Year	7252.00000	2013.36500	3.25450	1996.00000	2011.00000	2014.00000	2016.00000	2019.00000	23.00000	
Seats	7199.00000	5.27976	0.81171	0.00000	5.00000	5.00000	5.00000	10.00000	10.00000	

Jumlah Nilai Unik pada Kolom Numerik:
S.No.: 7252 nilai unik
Year: 23 nilai unik
Kilometers_Driven: 3660 nilai unik
Mileage: 437 nilai unik
Engine: 149 nilai unik
Power: 382 nilai unik
Seats: 9 nilai unik
New_Price: 625 nilai unik
Price: 1373 nilai unik

6. Analisis Missing Value

Analisis Missing Values:		
	Jumlah Missing	Persentase (%)
New_Price	6247	86.14175
Price	1234	17.01600
Power	175	2.41313
Mileage	82	1.13072
Seats	53	0.73083
Engine	46	0.63431

5. Statistik Deskriptif untuk Kolom Kategorikal

Statistik Deskriptif untuk Kolom Kategorikal:			
Name: 2040 nilai unik			
	Name	Jumlah	Persentase
0	Mahindra XUV500 W8 2WD	55	0.75841
1	Maruti Swift VDI	49	0.67568
2	Maruti Swift Dzire VDI	42	0.57915
3	Honda City 1.5 S MT	39	0.53778
4	Maruti Swift VDI BSIV	37	0.51020
5	Maruti Ritz VDi	35	0.48263
6	Toyota Fortuner 3.0 Diesel	35	0.48263
7	Honda City 1.5 V MT	32	0.44126
8	Honda Amaze S i-Dtech	32	0.44126
9	Honda Brio S MT	32	0.44126

Location: 11 nilai unik			
	Location	Jumlah	Persentase
0	Mumbai	949	13.08605
1	Hyderabad	876	12.07943
2	Coimbatore	772	10.64534
3	Kochi	772	10.64534
4	Pune	765	10.54881
5	Delhi	660	9.10094
6	Kolkata	654	9.01820
7	Chennai	590	8.13569
8	Jaipur	499	6.88086
9	Bangalore	440	6.06729
10	Ahmedabad	275	3.79206

Fuel_Type: 5 nilai unik			
	Fuel_Type	Jumlah	Persentase
0	Diesel	3852	53.11638
1	Petrol	3325	45.84942
2	CNG	62	0.85494
3	LPG	12	0.16547
4	Electric	1	0.01379

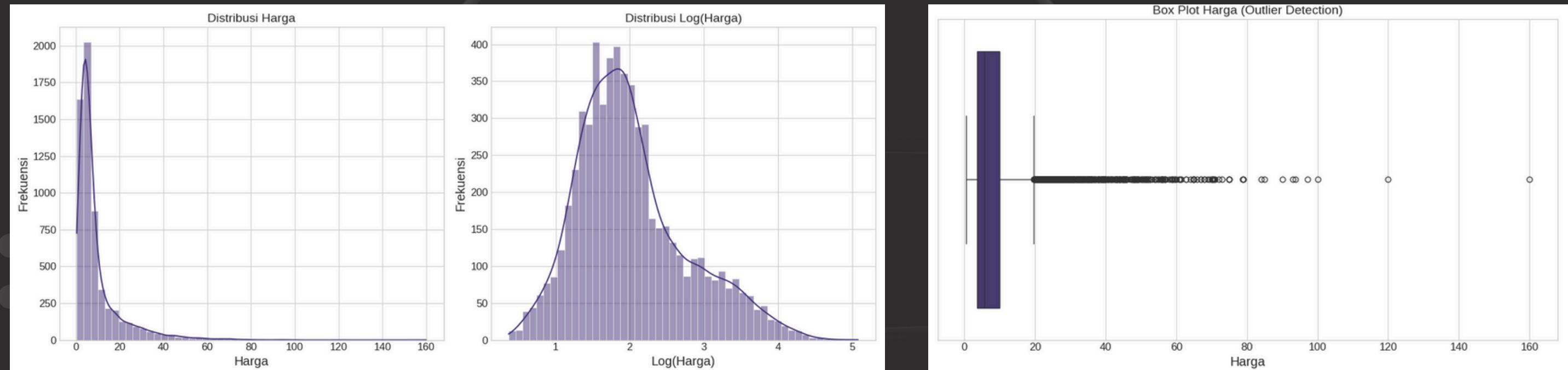
Transmission: 2 nilai unik			
	Transmission	Jumlah	Persentase
0	Manual	5204	71.75951
1	Automatic	2048	28.24049

Owner_Type: 4 nilai unik			
	Owner_Type	Jumlah	Persentase
0	First	5951	82.06012
1	Second	1152	15.88527
2	Third	137	1.88913
3	Fourth & Above	12	0.16547

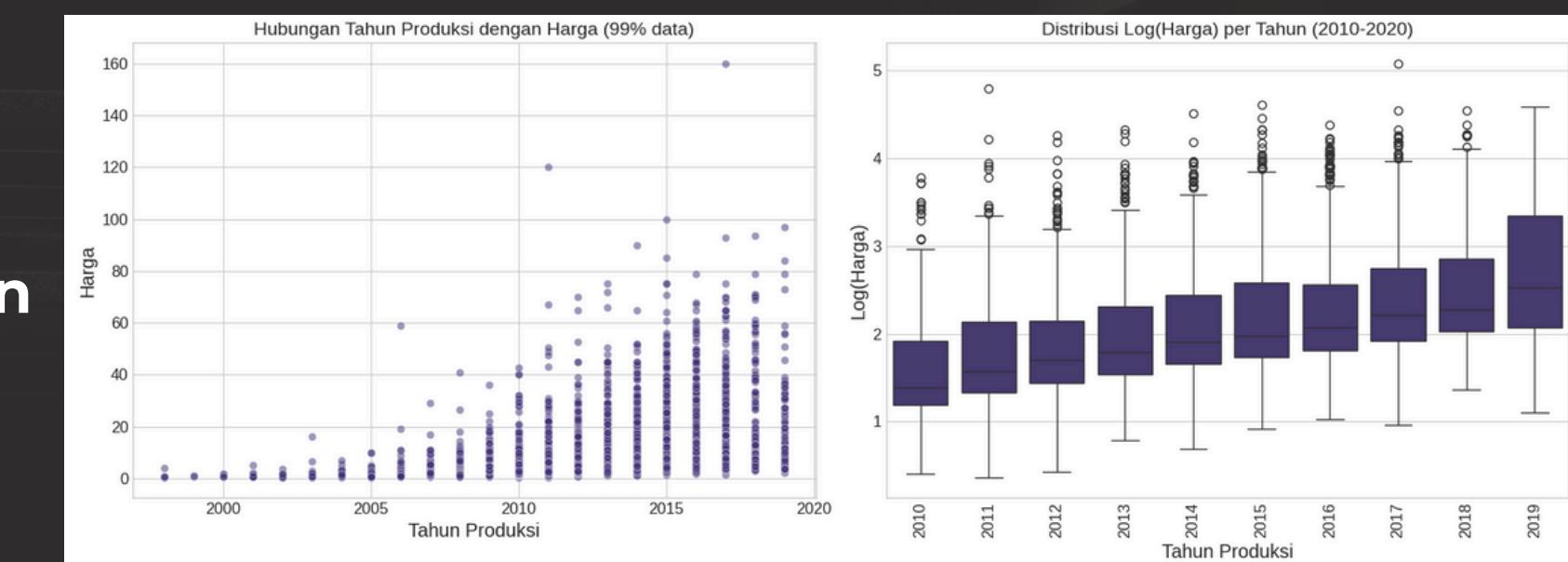
PEMAHAMAN DATASET

STATISTIK DESKRIPTIF VISUALISASI AWAL

7. Analisis Distribusi Harga



8. Harga Produksi Berdasarkan Tahun



Preprocessing Data

- o Menangani missing values (Mileage, Engine, Power, Seats, new_price).
- o Mengubah data string menjadi numerik
- o Normalisasi data dan melakukan encoding data kategorikal

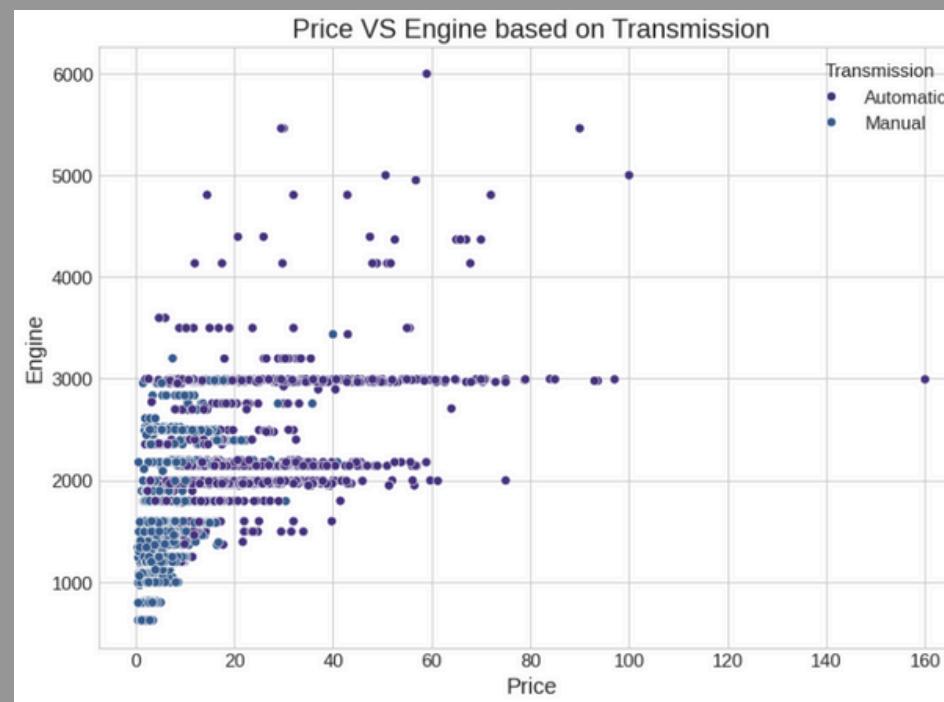
Eksplorasi Data

- o Dataset : 7.253 data mobil bekas India, terdiri dari 14 kolom
- o Fokus variabel : Tahun, Jarak Tempuh, Jenis Bahan Bakar, Transmisi, Tipe Pemilik, Konsumsi BBM, Mesin, Tenaga, Jumlah Kursi, Harga Baru

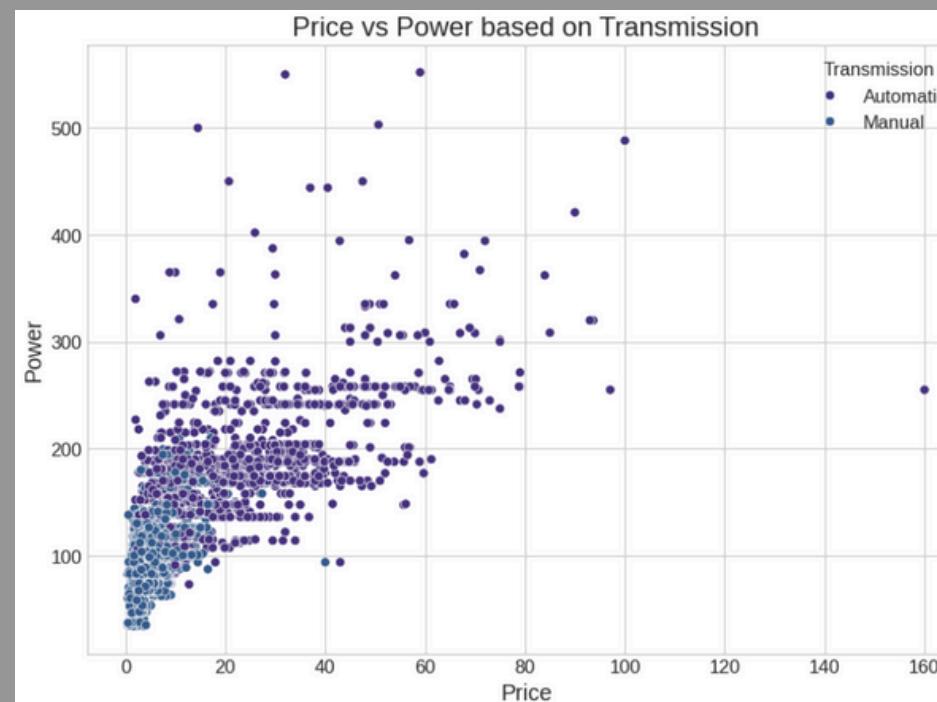
EKSPLORASI DATA DAN PRA- PEMROSESAN

Analisis Korelasi

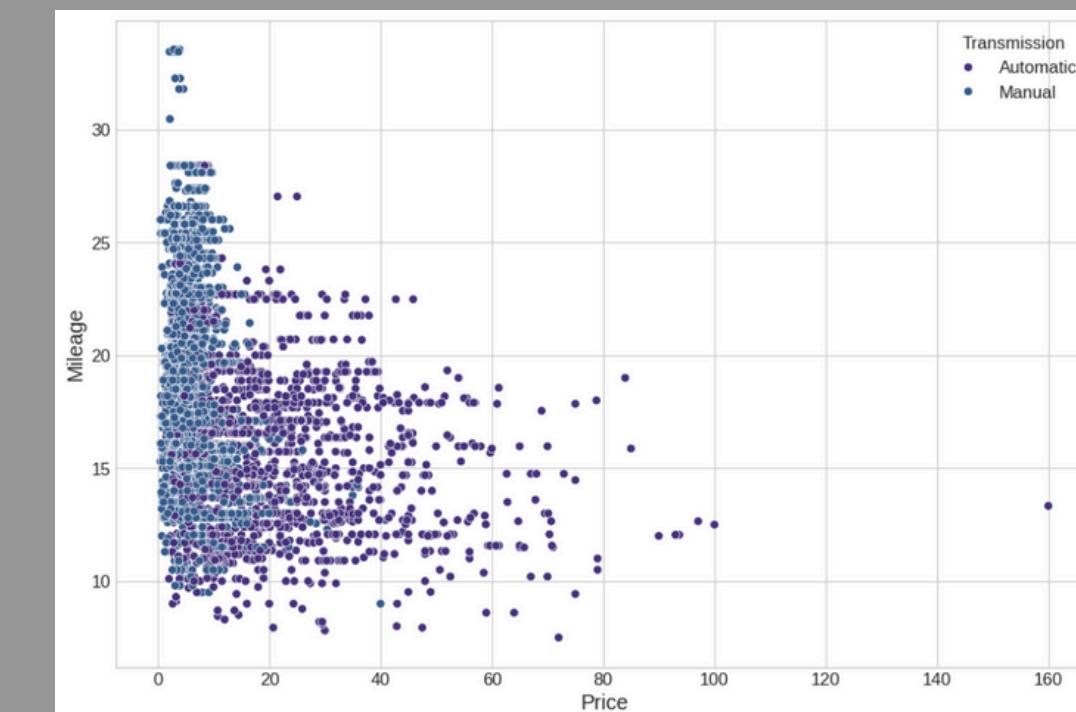
Harga vs Mesin



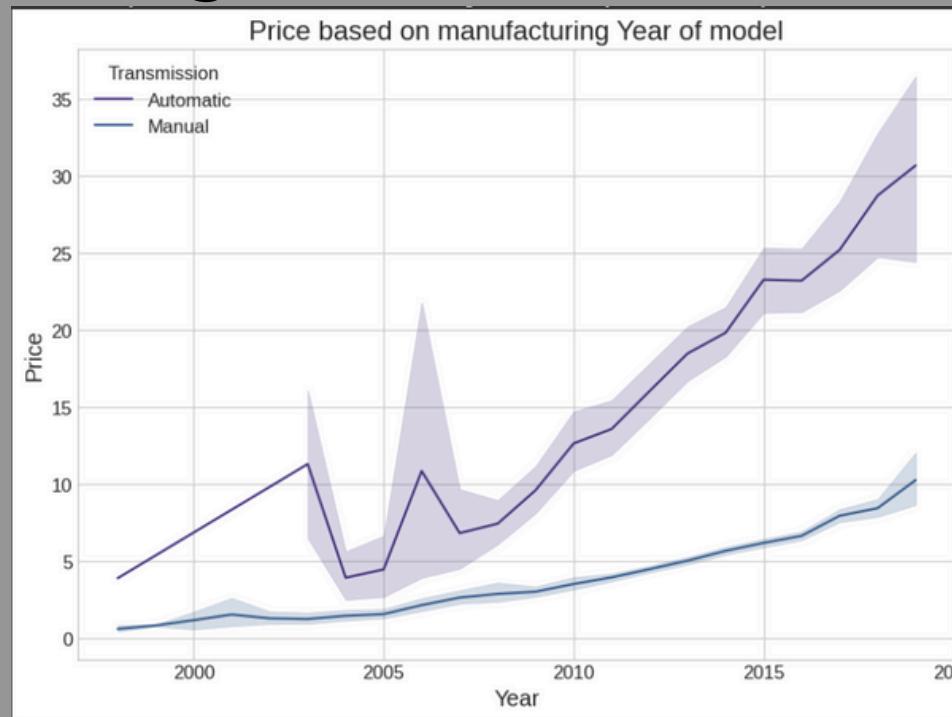
Harga vs Power



Harga vs Jarak Tempuh Kendaraan



Harga vs Tahun Produksi



Fitur Engine, Power, dan Year memiliki korelasi kuat dengan Price. Ketiganya dipilih sebagai fitur utama untuk prediksi, karena mobil dengan mesin besar, tenaga tinggi, dan tahun lebih baru cenderung lebih mahal.

IMPLEMENTASI MODEL

1. Memisahkan Data dan Target

```
1 X = df.drop(["Price", "Price_log"], axis=1)
2 y = df[["Price_log", "Price"]]
```

2. Mengubah Data Kategorikal menjadi Numerik

```
[ ] 1 def encode_cat_vars(x):
2     x = pd.get_dummies(
3         x,
4         columns=x.select_dtypes(include=["object", "category"]).columns.tolist(),
5         drop_first=True,
6     )
7     return x

[ ] #Dummy variable creation is done before splitting the data , so all the different categories are covered
2 #create dummy variable
3 X = encode_cat_vars(X)
4 X.head()
```

3. Membagi Data untuk Pengujian

```
[ ] 1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
2 X_train.reset_index()
3 print("X_train:", X_train.shape)
4 print("X_test:", X_test.shape)
5 print("y_train:", y_train.shape)
6 print("y_test:", y_test.shape)

→ X_train: (4123, 26)
X_test: (1768, 26)
y_train: (4123, 2)
y_test: (1768, 2)
```

IMPLEMENTASI MODEL

Membangun Model Prediksi Harga

1. Membuat Model Regresi

```
1 X_train = sm.add_constant(X_train)
2 X_test = sm.add_constant(X_test)
3
4 def build_ols_model(train):
5     # Convert input data explicitly to numeric
6     train_numeric = train.astype(float)
7
8     # Ensure no infinity or NaN values
9     train_numeric = train_numeric.replace([np.inf, -np.inf], np.nan)
10    train_numeric = train_numeric.dropna(axis=1)
11
12    # Print info to debug
13    print("Training data shape after cleaning:", train_numeric.shape)
14
15    # Use numpy arrays explicitly
16    y_array = np.asarray(y_train["Price_log"], dtype=np.float64)
17    X_array = np.asarray(train_numeric, dtype=np.float64)
18
19    # Create model
20    olsmodel = sm.OLS(y_array, X_array)
21    return olsmodel.fit()
```

Setelah pembersihan data, model regresi linier menghasilkan R-squared 89,4%, menunjukkan hubungan yang kuat antara fitur dengan harga mobil. Sebagian besar fitur signifikan ($p\text{-value} < 0.05$), dan model secara keseluruhan terbukti valid berdasarkan nilai F-statistic. Ini menandakan model cukup baik untuk memprediksi harga mobil bekas.

2. Membuat Model Regresi Polinomial

```
Final training data shape: (4123, 405)
OLS Regression Results
=====
Dep. Variable:                      y      R-squared:         0.929
Model:                          OLS      Adj. R-squared:    0.924
Method:                         Least Squares      F-statistic:     212.4
Date:                Mon, 07 Apr 2025      Prob (F-statistic):   0.00
Time:                    21:12:50      Log-Likelihood:  169.13
No. Observations:          4123      AIC:             139.7
Df Residuals:                  3884      BIC:             1651.
Df Model:                       238
Covariance Type:            nonrobust
=====
            coef    std err        t      P>|t|      [0.025      0.975]
-----
const    0.3166     1.392     0.227     0.820     -2.412      3.046
x1      6.334e-05  9.43e-05    0.672     0.502     -0.000      0.000
x2      0.0076     0.031     0.248     0.804     -0.052      0.068
x3      0.0004     0.000     0.831     0.406     -0.001      0.001
x4     -0.0024     0.006    -0.424     0.671     -0.014      0.009
x5      0.1238     0.146     0.848     0.396     -0.162      0.410
x6     -0.0331     0.032    -1.041     0.298     -0.095      0.029
x7      0.1983     0.298     0.666     0.505     -0.385      0.782
x8      0.0890     0.279     0.319     0.749     -0.457      0.636
x9      0.3377     0.318     1.061     0.289     -0.287      0.962
```

Dengan penerapan polynomial regression, performa model meningkat dengan R-squared 92,9%, namun jumlah fitur bertambah signifikan (hingga 405 fitur). Model tetap kuat (Adjusted R-squared 92,4%), tetapi muncul indikasi overfitting karena beberapa fitur baru tidak signifikan. Diperlukan regularisasi atau seleksi fitur untuk mengoptimalkan model.

EVALUASI MODEL

- o Menggunakan metrik evaluasi seperti Mean Squared Error (MSE), R² Score, dan Mean Absolute Error (MAE).
- o Membandingkan kinerja regresi linear dan polinomial.

EVALUASI MODEL

```
 1 import math
 2
 3 # RMSE
 4 def rmse(predictions, targets):
 5     return np.sqrt((targets - predictions) ** 2).mean()
 6
 7
 8 # MAPE
 9 def mape(predictions, targets):
10     return np.mean(np.abs(targets - predictions)) / targets * 100
11
12
13 # MAE
14 def mae(predictions, targets):
15     return np.mean(np.abs(targets - predictions))
16
17
18 # Model Performance on test and train data
19 def model_pref(olsmodel, x_train, x_test):
20
21     # Insample Prediction
22     y_pred_train_pricelog = olsmodel.predict(x_train)
23     y_pred_train_Price = y_pred_train_pricelog.apply(math.exp)
24     y_train_Price = y_train["Price"]
25
26     # Prediction on test data
27     y_pred_test_pricelog = olsmodel.predict(x_test)
28     y_pred_test_Price = y_pred_test_pricelog.apply(math.exp)
29     y_test_Price = y_test["Price"]
30
31     print(
32         pd.DataFrame(
33             {
34                 "Data": ["Train", "Test"],
35                 "RMSE": [
36                     rmse(y_pred_train_Price, y_train_Price),
37                     rmse(y_pred_test_Price, y_test_Price),
38                 ],
39                 "MAE": [
40                     mae(y_pred_train_Price, y_train_Price),
41                     mae(y_pred_test_Price, y_test_Price),
42                 ],
43                 "MAPE": [
44                     mape(y_pred_train_Price, y_train_Price),
45                     mape(y_pred_test_Price, y_test_Price),
46                 ],
47             }
48         )
49     )
50
51
52 # Checking model performance
53 model_pref(olsmodel1, X_train, X_test) # High Overfitting.
```

	Data	RMSE	MAE	MAPE
0	Train	6.50314	2.27893	23.30111
1	Test	7.70234	2.56031	23.66396

Model Regresi Polinomial menunjukkan akurasi sangat tinggi di data latih (R-squared 98,6%) namun mengalami overfitting parah saat diuji di data baru. Sebaliknya, Regresi Linear, meski memiliki akurasi lebih rendah (R-squared 89,4%), menunjukkan performa yang stabil dan andal baik di data latih maupun data uji, sehingga lebih cocok untuk generalisasi.

EVALUASI MODEL

Perbandingan Model

Aspek	Regresi Linear	Regresi Polinomial
Akurasi (R^2)	0,894 (cukup tinggi)	0,986 (sangat tinggi)
Generalitas ke Data Baru	Baik – performa train dan test serupa	Buruk – overfitting berat pada data test
RMSE	Train: 6.50 Test: 7.70	Train: 3.67 Test: ~
MAE	Train: 2.28 Test: 2.56	Train: 1.68 Test: ~1.24e+107
MAPE	Train: 23.30% Test: 23.66%	Train: 18.12% Test: ~1.90e+107%
Overfitting	Tidak – perbedaan train-test kecil	Sangat parah – test error ekstrem
Kemampuan Tangkap Pola Non-Linear	Terbatas – hanya linear	Kuat – tangkap hubungan kompleks
Catatan Khusus	Multikolinearitas terdeteksi (perlu cek VIF)	Perlu regularisasi & normalisasi
Kesimpulan	Stabil dan andal untuk prediksi	Hanya cocok untuk data latih, tidak untuk generalisasi

Perbandingan Regresi Linear dan Regresi Polinomial menunjukkan bahwa Regresi Linear lebih stabil dan andal untuk prediksi, dengan R^2 sebesar 0,894 dan error yang wajar di data train maupun test. Sementara itu, Regresi Polinomial mencapai R^2 sangat tinggi (0,986) di data latih, namun mengalami overfitting parah di data uji. Oleh karena itu, Regresi Linear lebih cocok untuk generalisasi, sedangkan Regresi Polinomial perlu perbaikan sebelum digunakan lebih luas.

- o Menginterpretasikan koefisien regresi.
- o Menyajikan grafik regression line dan polynomial fit.
- o Menyimpulkan apakah model yang dibuat cukup baik dalam memprediksi target.

ANALISIS HASIL



ANALISIS HASIL

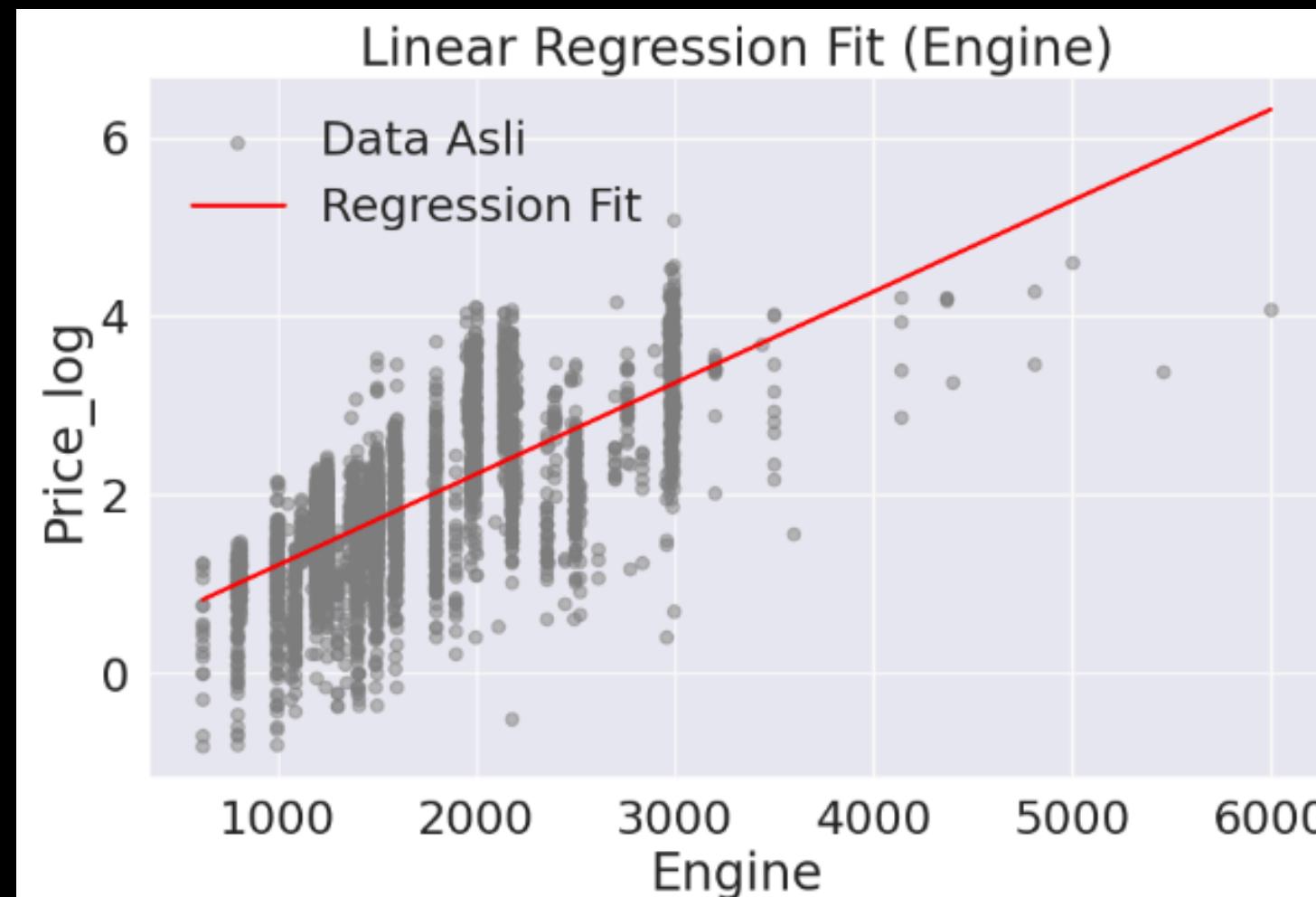
Interpretasi Koefisien Regresi

No.	Fitur	Koefisien	Signifikan ($p < 0.05$)?	Interpretasi
1	const	3.3549	<input checked="" type="checkbox"/> Ya	Nilai prediksi saat semua variabel independen bernilai nol.
2	Kilometers_Driven	-2.74e-07	<input checked="" type="checkbox"/> Tidak	Tidak signifikan, sedikit pengaruh terhadap harga.
3	Mileage	-0.0214	<input checked="" type="checkbox"/> Ya	Setiap kenaikan 1 km/l, harga turun ~2.14% (mungkin multikolinearitas).
4	Engine	0.0081	<input checked="" type="checkbox"/> Ya	Setiap kenaikan 1 cc mesin, harga naik ~0.81%.
5	Power	0.0503	<input checked="" type="checkbox"/> Ya	Setiap kenaikan 1 bhp, harga naik ~5.03%.
6	Seats	-0.1108	<input checked="" type="checkbox"/> Ya	Setiap tambahan 1 kursi, harga turun ~11.08%.
7	Age	-0.0621	<input checked="" type="checkbox"/> Ya	Mobil lebih tua 1 tahun → harga turun ~6.21%.
8	Kilometers_Driven_log	0.1477	<input checked="" type="checkbox"/> Ya	Log jarak tempuh lebih tinggi → harga naik ~14.77%.
9	Location_Bangalore	0.0105	<input checked="" type="checkbox"/> Tidak	Tidak signifikan terhadap harga.
10	Location_Chennai	0.0794	<input checked="" type="checkbox"/> Ya	Mobil di Chennai cenderung lebih mahal ~7.94%.
11	Location_Coimbatore	-0.0579	<input checked="" type="checkbox"/> Ya	Mobil di Coimbatore cenderung lebih murah ~5.79%.
12	Location_Delhi	0.0956	<input checked="" type="checkbox"/> Ya	Mobil di Delhi cenderung lebih mahal ~9.56%.
13	Location_Hyderabad	-0.0779	<input checked="" type="checkbox"/> Ya	Mobil di Hyderabad cenderung lebih murah ~7.79%.
14	Location_Jaipur	-0.0338	<input checked="" type="checkbox"/> Tidak	Tidak signifikan.
15	Location_Kochi	-0.2493	<input checked="" type="checkbox"/> Ya	Mobil di Kochi jauh lebih murah ~24.93%.
16	Location_Kolkata	-0.0775	<input checked="" type="checkbox"/> Ya	Mobil di Kolkata lebih murah ~7.75%.
17	Location_Mumbai	-0.0566	<input checked="" type="checkbox"/> Ya	Mobil di Mumbai lebih murah ~5.66%.
18	Location_Pune	0.1908	<input checked="" type="checkbox"/> Ya	Mobil di Pune lebih mahal ~19.08%.
19	Fuel_Type_Diesel	1.2610	<input checked="" type="checkbox"/> Ya	Mobil diesel cenderung lebih mahal ~126.1%.
20	Fuel_Type_Electric	-0.0514	<input checked="" type="checkbox"/> Tidak	Tidak signifikan.
21	Fuel_Type_LPG	-0.1349	<input checked="" type="checkbox"/> Ya	Mobil LPG lebih murah ~13.49%.
22	Fuel_Type_Petrol	-0.2285	<input checked="" type="checkbox"/> Ya	Mobil bensin lebih murah ~22.85%.
23	Transmission_Manual	0.2185	<input checked="" type="checkbox"/> Tidak	Belum signifikan meskipun koefisien positif.
24	Owner_Type_Fourth & Above	-0.0943	<input checked="" type="checkbox"/> Ya	Pemilik ke-4+ → harga turun ~9.43%.
25	Owner_Type_Second	-0.1444	<input checked="" type="checkbox"/> Ya	Pemilik kedua → harga turun ~14.44%.
26	Owner_Type_Third	-0.2178	<input checked="" type="checkbox"/> Ya	Pemilik ketiga → harga turun ~21.78%.
27	Brand_Class_Low	-0.2178	<input checked="" type="checkbox"/> Ya	Brand kelas bawah → harga lebih rendah ~21.78%.

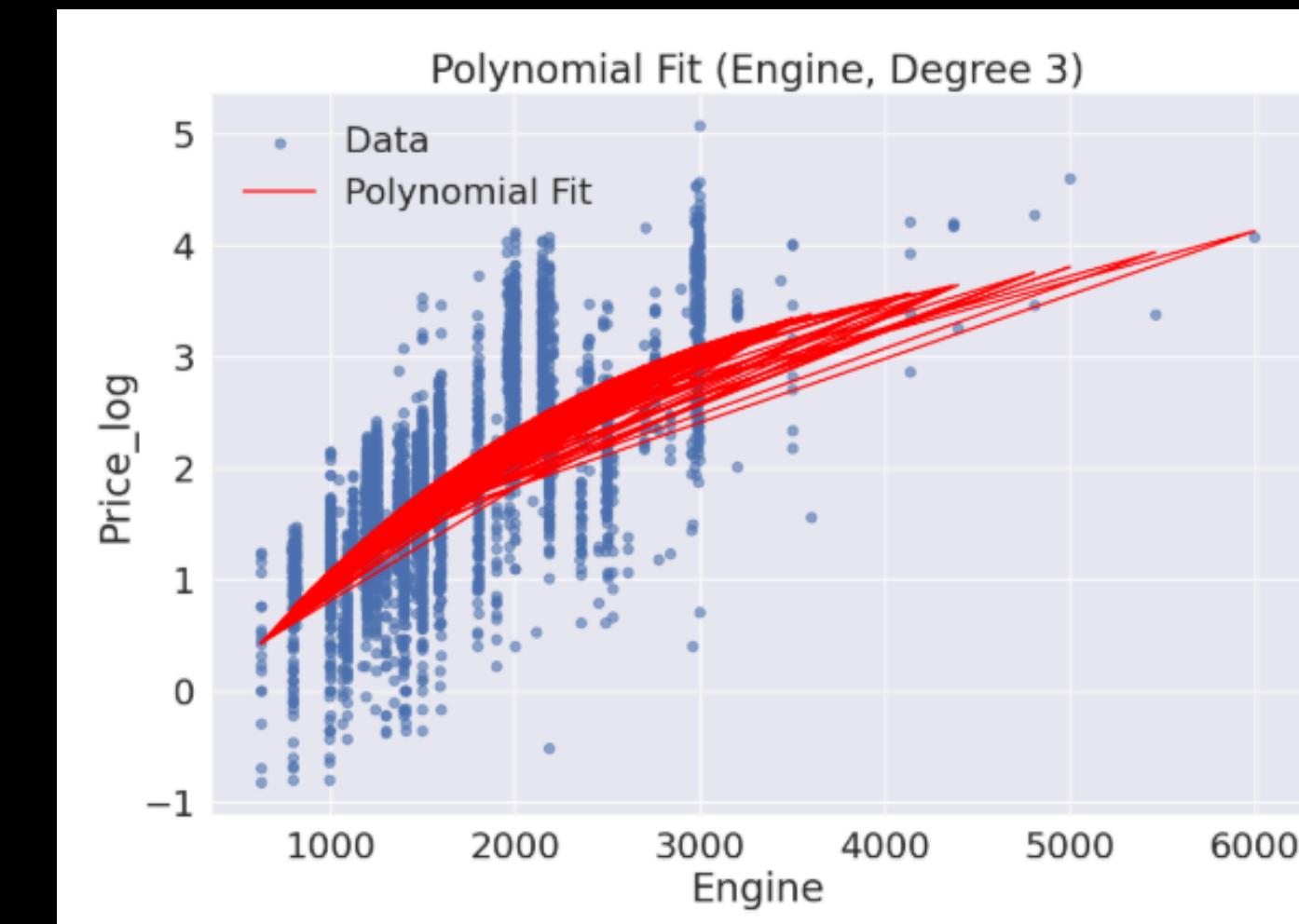
Hasil regresi menunjukkan bahwa Power, Engine, dan Kilometers_Driven_log berpengaruh positif terhadap harga mobil, sementara Seats, Mileage, dan Age berpengaruh negatif. Lokasi juga memengaruhi harga: mobil di Chennai lebih mahal, sedangkan di Coimbatore lebih murah. Variabel Kilometers_Driven dan Location_Bangalore tidak berpengaruh signifikan.

ANALISIS HASIL

Grafik Regression Line dan Polynomial Fit



Gambar ini menunjukkan hubungan positif antara kapasitas mesin (Engine) dan harga mobil bekas (Price_log). Semakin besar kapasitas mesin, cenderung harga mobil semakin tinggi, terlihat dari garis regresi merah yang menanjak. Meskipun data tersebar, tren umum berhasil ditangkap dengan baik oleh model regresi.



Gambar ini menunjukkan hubungan kapasitas mesin (Engine) dengan harga mobil bekas (Price_log) menggunakan regresi polinomial orde 3. Model ini lebih fleksibel dalam menangkap pola non-linear dibanding regresi linear, namun terlihat indikasi overfitting di area tengah karena kurva terlalu mengikuti data.

KESIMPULAN

- Regresi linear terbukti sebagai solusi terbaik karena performanya stabil di data pelatihan dan pengujian.
- Nilai R-squared sebesar 0.894 menunjukkan kemampuan model dalam menjelaskan variasi data dengan baik.
- Nilai RMSE, MAE, dan MAPE rendah dan seimbang, menandakan error yang konsisten dan tidak berlebihan.
- Model linear mampu menggeneralisasi pola data secara konsisten, tidak hanya menghafal data pelatihan.

- Pemeriksaan multikolinearitas (VIF) menunjukkan fitur aman, sehingga interpretasi tiap variabel dapat dipercaya.
- Model regresi polinomial mengalami overfitting, dengan akurasi tinggi di training ($R^2 = 0.986$) namun buruk di pengujian.
- Error ekstrem pada regresi polinomial membuktikan bahwa model tidak mampu menggeneralisasi data baru.
- Regresi linear direkomendasikan karena lebih sederhana, mudah dipahami, dan prediksinya lebih andal.

THANK YOU

BY: KELOMPOK 2

