Nama: I Gede Suryananda Adikartika

Nim: 2201020038

Prodi: Teknik Informatika

Report & Insights

1. Executive Summary Hasil

Proyek ini bertujuan untuk membangun dan mengevaluasi model regresi yang optimal untuk memprediksi harga properti, dengan fokus utama pada analisis *Bias-Variance Trade-off*. Kami men-generate dataset sintetis non-linear (200 sampel, 5 fitur) dan menguji 35 konfigurasi model yang berbeda, membandingkan 5 tingkat kompleksitas (Polynomial Degree 1-5) dengan 3 metode regularisasi (Linear Regression, Ridge, dan Lasso).

Analisis perbandingan model secara jelas menunjukkan bahwa model Degree 1 mengalami *underfitting*, sementara model Degree 4 dan 5 mengalami *overfitting* yang parah (contoh: Test R^2 anjlok ke 0.000).

Model pemenang yang terpilih adalah Lasso (L1) Regression dengan Polynomial Degree 2 dan alpha=1.0. Model ini secara konsisten memberikan performa terbaik di seluruh pengujian, mencapai skor Test R^2 tertinggi (0.983) dan Test RMSE terendah. Model ini juga menunjukkan keseimbangan (balance) sempurna antara skor *train* dan *test*, yang membuktikan ia bebas dari *overfitting*. Model final ini telah divalidasi menggunakan K-Fold Cross-Validation dan disimpan untuk prediksi di masa depan.

2. Insight dari EDA

1. **Data Preparation.**
   a. **Data Generation**

	Luas_Tanah_m2	Luas_Bangunan_m2	Jumlah_Kamar_Tidur	Umur_Bangunan_Tahun	Jarak_Pusat_Kota_km	Harga_Properti_jutaRp
1	216	305	1	28	14	1171
2	360	63	1	15	7	954
3	491	33	5	14	10	1291
4	400	39	2	18	10	1047
5	243	185	3	13	10	973
6	404	228	3	1	4	1679
7	52	381	3	25	8	1230
8	295	260	2	27	13	1250
9	441	317	3	15	5	1856
10	107	129	5	3	6	738
11	394	148	3	7	4	1338
12	225	299	4	20	20	1369
13	341	391	2	6	9	1948
14	67	296	5	7	11	1168
15	109	167	4	19	15	818
16	262	69	1	17	6	732
17	214	383	4	24	10	1663
18	83	336	1	10	6	1202
19	228	261	5	16	18	1258
20	380	391	5	20	8	2020
21	336	367	4	0	5	1998
22	442	363	1	24	10	1936
23	377	368	1	9	2	1915
24	265	146	4	19	10	1071
25	239	295	1	10	14	1241
26	313	144	3	21	8	1160
27	193	34	1	8	4	432
28	443	234	3	5	15	1680
29	468	392	2	25	6	2102
30	361	179	3	28	13	1277

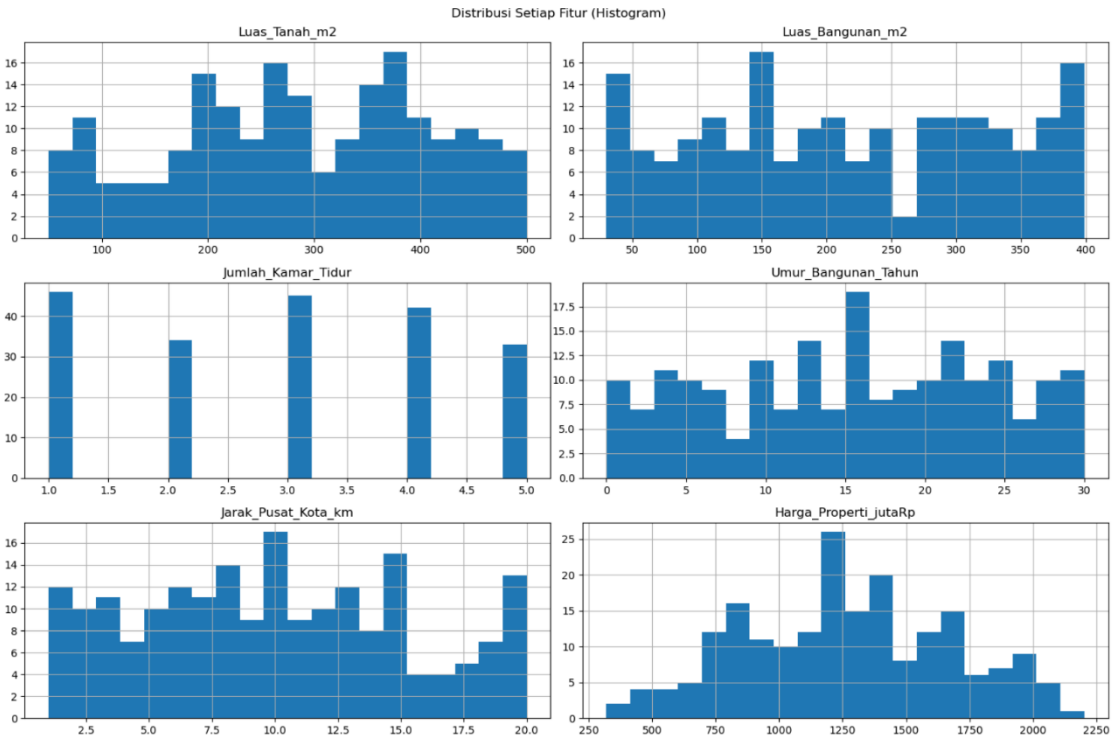
   b. **Exploratory Data Analysis**

- **Menampilkan statistic data sumary**

```
--- B.1: Statistical Summary ---
       Luas_Tanah_m2  Luas_Bangunan_m2  Jumlah_Kamar_Tidur  Umur_Bangunan_Tahun  Jarak_Pusat_Kota_km  Harga_Properti_jutaRp
count        200.000           200.000             200.000              200.000              200.000                200.000
mean         287.340           216.260               2.910               15.175                9.910              1,268.375
std          122.153           111.852               1.401                8.587                5.579                412.484
min           50.000            30.000               1.000                0.000                1.000                320.000
25%          198.000           122.250               2.000                8.000                5.750                957.000
50%          288.000           204.500               3.000               15.000               10.000              1,250.500
75%          383.250           316.000               4.000               22.000               14.000              1,591.500
max          500.000           399.000               5.000               30.000               20.000              2,202.000
```

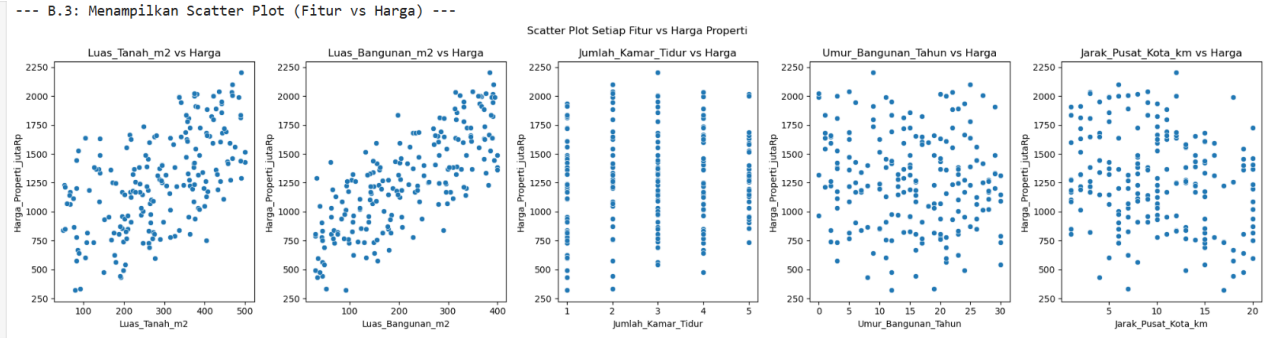
- **Visualisasi distribusi setiap fitur (histogram)**

```
--- B.2: Menampilkan Histogram (Distribusi Fitur) ---
```



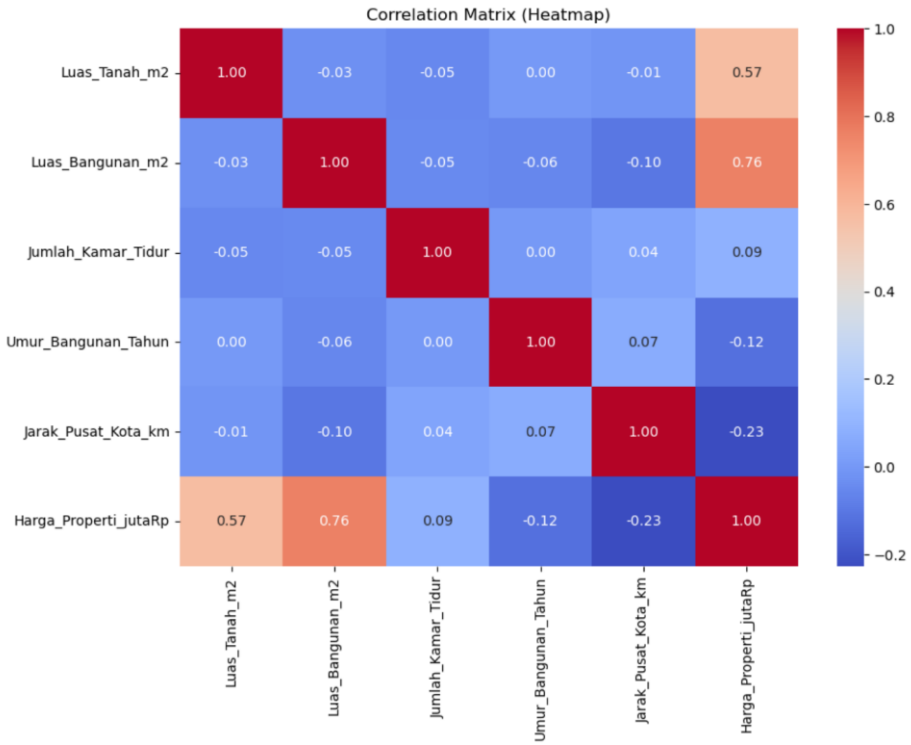
Menunjukkan distribusi data dari beberapa fitur properti seperti luas tanah, luas bangunan, jumlah kamar, umur bangunan, jarak ke pusat kota, dan harga. Setiap grafik menampilkan seberapa banyak data (sumbu Y) yang memiliki nilai tertentu pada rentang fitur tersebut (sumbu X). Dari bentuk batangnya, kita bisa melihat pola sebaran data—apakah merata, terkonsentrasi di satu kisaran, atau memiliki variasi besar antar nilai.

- **Scatter plot setiap fitur vs harga**



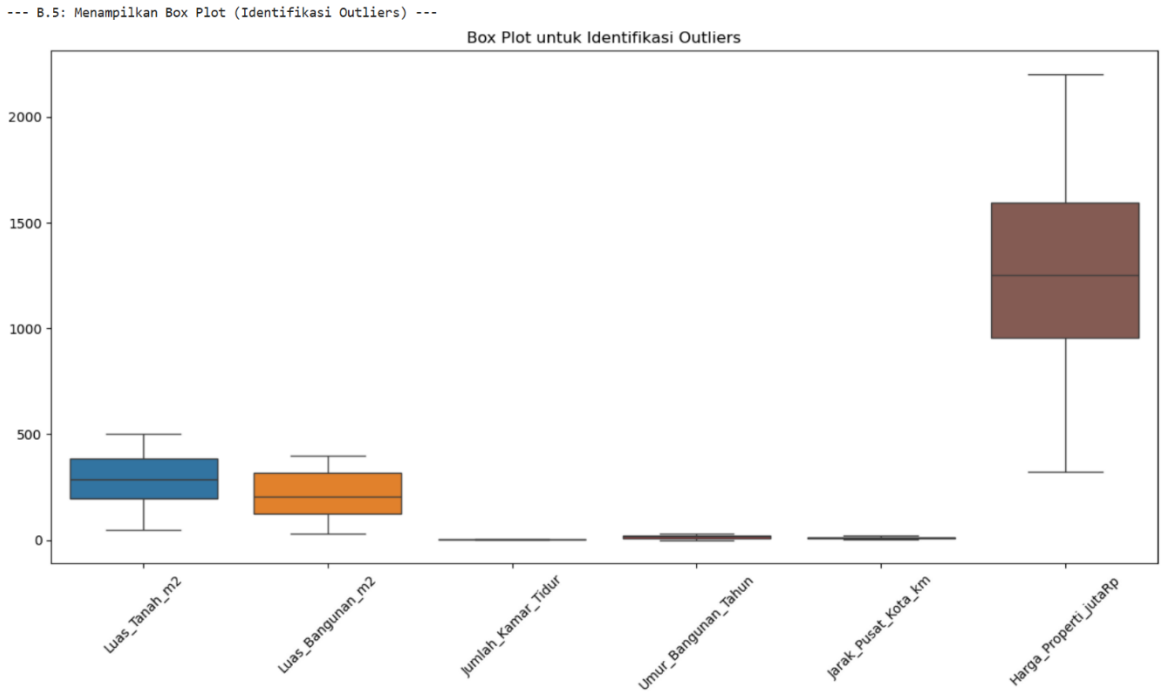
Scatter plot antara setiap fitur dengan harga properti untuk melihat hubungan di antara keduanya. Titik-titik yang cenderung naik ke kanan (seperti pada *Luas Tanah* dan *Luas Bangunan*) menunjukkan bahwa semakin besar nilai fitur tersebut, harga properti juga cenderung meningkat. Sementara itu, pada fitur seperti *Umur Bangunan* dan *Jarak ke Pusat Kota*, pola titiknya lebih menyebar acak, menandakan hubungan yang lemah atau tidak terlalu berpengaruh terhadap harga.

- **Correlation matrix (heatmap)**



Menunjukkan heatmap korelasi antar fitur dalam dataset, yang menggambarkan seberapa kuat hubungan antara satu variabel dengan variabel lainnya. Warna merah menandakan korelasi positif (semakin tinggi nilai satu variabel, semakin tinggi juga variabel lainnya), sedangkan biru menandakan korelasi negatif. Dari hasilnya, terlihat bahwa Luas Bangunan (0.76) dan Luas Tanah (0.57) memiliki korelasi paling kuat dengan Harga Properti, artinya kedua faktor tersebut paling berpengaruh terhadap tinggi rendahnya harga rumah.

- **Identifikasi Outlier**



Mengidentifikasi outlier atau data yang menyimpang jauh dari nilai umum pada setiap fitur. Kotak menunjukkan sebaran data utama (dari kuartil bawah ke atas), garis di tengah kotak menunjukkan median, dan titik di luar "garis whisker" menunjukkan kemungkinan data ekstrem (outlier). Dari grafik ini, terlihat bahwa sebagian besar fitur memiliki sebaran data yang cukup normal tanpa banyak outlier, sementara fitur Harga Properti memiliki rentang nilai yang lebih besar, menandakan variasi harga yang tinggi antar properti.

3. Perbandingan Performa Model

C. Data Preprocessing

- **Split data train-test (70:30)**

```
--- Split Data Train-Test (70:30) ---
Data dibagi:
X_train shape: (140, 5)
X_test shape: (60, 5)
y_train shape: (140,)
y_test shape: (60,)

==============================
```

- **Feature scaling menggunakan StandardScaler**

```
Data X_train sebelum scaling (5 baris pertama):
     Luas_Tanah_m2  Luas_Bangunan_m2  Jumlah_Kamar_Tidur  Umur_Bangunan_Tahun  Jarak_Pusat_Kota_km
169            439               111                   2                   28                   13
97             198                31                   1                   24                   13
31             407                97                   4                    3                   15
12             341               391                   2                    6                    9
35              50               290                   1                    7                   12

Data X_train SETELAH scaling (5 baris pertama):
[[ 1.1622201  -0.99556459 -0.58293911  1.55245217  0.51283808]
 [-0.70525805 -1.69640202 -1.29882925  1.07842097  0.51283808]
 [ 0.9142562  -1.11821114  0.84884117 -1.41024281  0.86478578]
 [ 0.40283065  1.4573664  -0.58293911 -1.05471941 -0.19105732]
 [-1.85209111  0.57255915 -1.29882925 -0.93621161  0.33686423]]
```

2. Model Implementation.

a. Polynomial Feature Engineering

Jumah fitur dari setiap degree:

```
Degree 1: Menghasilkan 5 fitur
Degree 2: Menghasilkan 20 fitur
Degree 3: Menghasilkan 55 fitur
Degree 4: Menghasilkan 125 fitur
Degree 5: Menghasilkan 251 fitur
```

b. Model Training

Model training dengan 5 degree :

```
Melatih model untuk Degree 1...

Melatih model untuk Degree 2...

Melatih model untuk Degree 3...

Melatih model untuk Degree 4...

Melatih model untuk Degree 5...

--- Pelatihan Model Selesai ---
Semua 35 model telah dilatih dan disimpan dalam variabel 'trained_models'.

Contoh model yang tersimpan:
Model untuk Degree 2: ['LinearRegression', 'Ridge (alpha=0.1)', 'Ridge (alpha=1)', 'Ridge (alpha=10)', 'Lasso (alpha=0.1)', 'Lasso (alpha=1)', 'Lasso (al
pha=10)']
Model untuk Degree 5: ['LinearRegression', 'Ridge (alpha=0.1)', 'Ridge (alpha=1)', 'Ridge (alpha=10)', 'Lasso (alpha=0.1)', 'Lasso (alpha=1)', 'Lasso (al
pha=10)']
```

3. Model Evaluation.

Hasil evaluasi Metrics Calculation Untuk setiap model, hitung:

	Degree	Model	Train R2	Test R2	Train RMSE	Test RMSE	Train MAE	Test MAE
12	2	Lasso (alpha=1)	0.983	0.983	55.394	47.153	43.756	36.539
9	2	Ridge (alpha=1)	0.983	0.983	55.299	47.617	44.003	36.321
11	2	Lasso (alpha=0.1)	0.983	0.983	55.207	47.676	43.801	36.498
2	1	Ridge (alpha=1)	0.981	0.983	59.157	47.703	47.523	39.475
8	2	Ridge (alpha=0.1)	0.983	0.983	55.205	47.737	43.854	36.509
5	1	Lasso (alpha=1)	0.981	0.983	59.119	47.745	47.349	39.297
7	2	LinearRegression	0.983	0.983	55.204	47.761	43.842	36.530
13	2	Lasso (alpha=10)	0.979	0.983	62.487	47.766	50.939	38.674
1	1	Ridge (alpha=0.1)	0.981	0.982	59.078	48.096	47.169	39.410
4	1	Lasso (alpha=0.1)	0.981	0.982	59.077	48.103	47.151	39.392
0	1	LinearRegression	0.981	0.982	59.077	48.147	47.130	39.402
6	1	Lasso (alpha=10)	0.978	0.982	63.120	48.316	51.408	39.520
3	1	Ridge (alpha=10)	0.977	0.981	65.713	49.836	53.912	42.068
10	2	Ridge (alpha=10)	0.979	0.978	62.787	54.083	51.785	41.407
19	3	Lasso (alpha=1)	0.987	0.976	48.304	56.295	38.665	45.817
27	4	Lasso (alpha=10)	0.978	0.973	63.252	59.742	52.137	47.759
20	3	Lasso (alpha=10)	0.978	0.972	64.113	60.725	52.744	49.299
26	4	Lasso (alpha=1)	0.992	0.971	38.715	61.400	29.823	49.321
18	3	Lasso (alpha=0.1)	0.988	0.968	46.563	64.306	36.947	52.107
16	3	Ridge (alpha=1)	0.988	0.967	47.872	65.548	38.450	52.592
15	3	Ridge (alpha=0.1)	0.988	0.967	46.544	65.561	37.020	52.921
14	3	LinearRegression	0.988	0.967	46.525	65.879	36.929	53.290
34	5	Lasso (alpha=10)	0.978	0.965	64.001	67.449	51.212	52.121
33	5	Lasso (alpha=1)	0.995	0.962	29.936	69.962	21.328	51.178
17	3	Ridge (alpha=10)	0.975	0.945	67.670	84.657	52.499	66.673
24	4	Ridge (alpha=10)	0.983	0.930	55.224	95.570	41.284	72.683
23	4	Ridge (alpha=1)	0.995	0.925	28.863	98.827	21.660	82.621
25	4	Lasso (alpha=0.1)	0.997	0.911	25.212	108.041	18.446	83.600
31	5	Ridge (alpha=10)	0.992	0.882	38.626	124.240	22.886	89.888
22	4	Ridge (alpha=0.1)	0.997	0.881	23.670	124.434	17.599	99.785
32	5	Lasso (alpha=0.1)	1.000	0.862	8.661	134.192	4.747	87.232
21	4	LinearRegression	0.997	0.821	23.302	153.014	17.162	118.382
30	5	Ridge (alpha=1)	0.999	0.815	12.341	155.575	5.619	110.374
29	5	Ridge (alpha=0.1)	1.000	0.697	2.873	198.794	1.089	140.602
28	5	LinearRegression	1.000	0.642	0.000	216.284	0.000	151.826

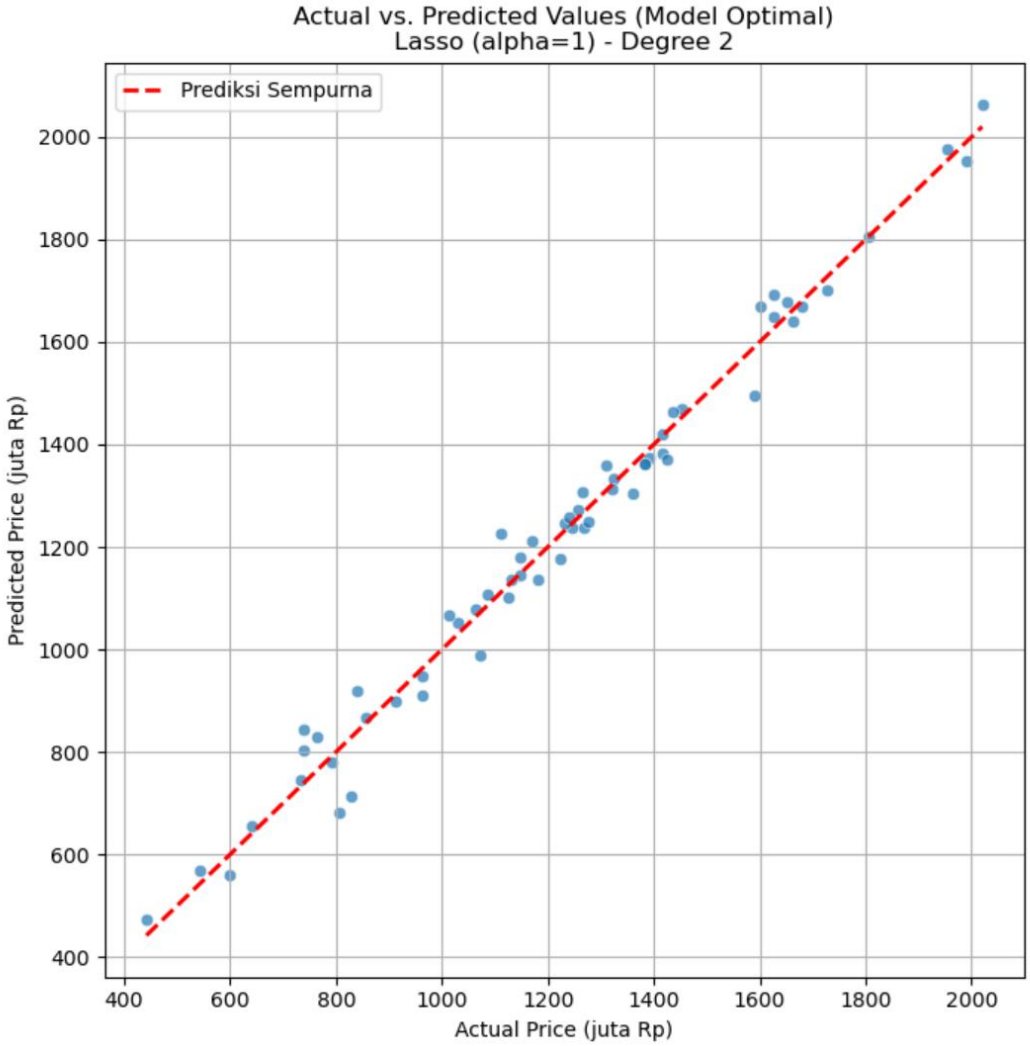
```
--- Bagian 3.A Selesai ---
Hasil evaluasi lengkap juga disimpan ke 'model_evaluation_results.csv'
```

hasil evaluasi model regresi (Linear, Ridge, dan Lasso) dengan berbagai nilai *degree* dan parameter *alpha* untuk memprediksi harga properti. Kolom R^2 menunjukkan seberapa baik model menjelaskan variasi data (semakin mendekati 1 berarti semakin akurat), sedangkan RMSE dan MAE menunjukkan rata-rata besar kesalahan prediksi (semakin kecil semakin baik). Dari tabel terlihat bahwa model Lasso atau Ridge dengan degree rendah (1–2) memberikan performa terbaik, karena memiliki nilai R^2 tinggi serta error (RMSE dan MAE) yang relatif kecil di data uji.

Hasil Visualization:

- **Plot predicted vs actual values untuk test set**

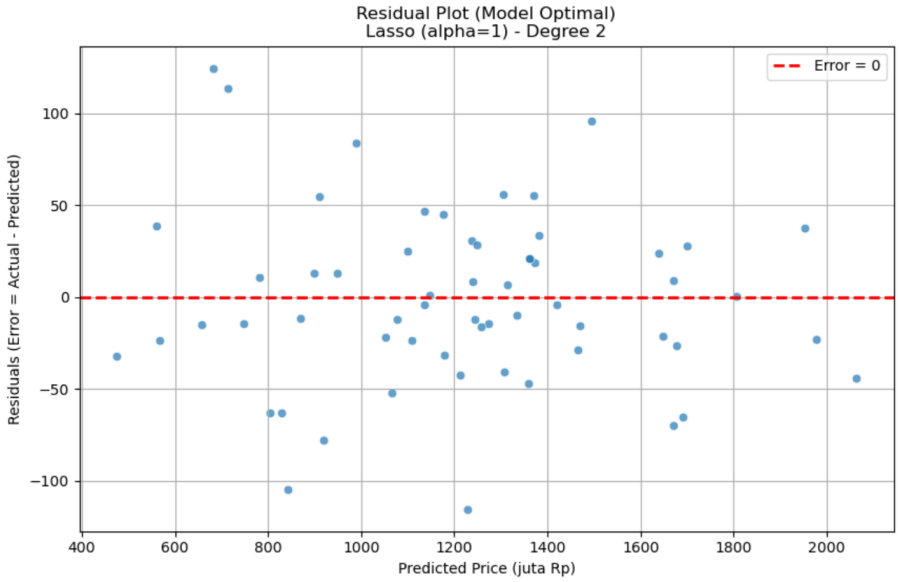
```
Menampilkan Plot 1: Actual vs Predicted...
```



Menunjukkan perbandingan antara harga aktual dan harga prediksi dari model terbaik, yaitu Lasso (alpha=1) dengan degree 2. Titik-titik biru mewakili hasil prediksi model, sedangkan garis merah putus-putus menunjukkan prediksi sempurna di mana nilai aktual dan prediksi sama persis. Karena sebagian besar titik berada sangat dekat dengan garis merah, berarti model ini memiliki akurasi tinggi dan mampu memprediksi harga properti dengan sangat baik.

- **Residual plot untuk analisis error**

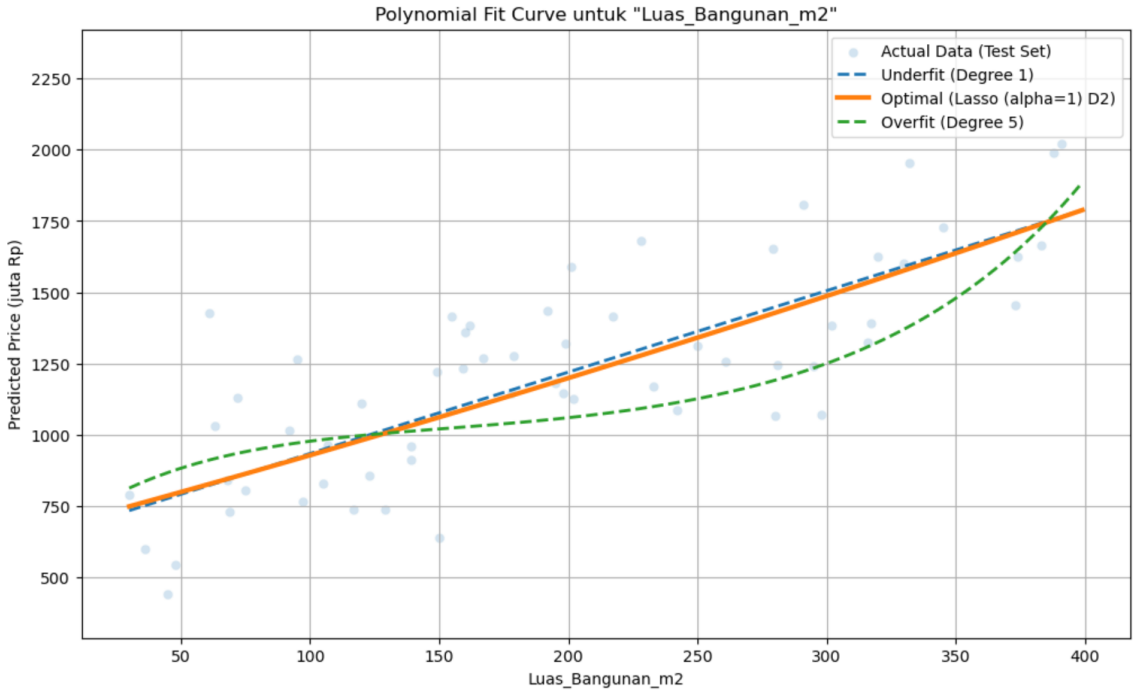
Menampilkan Plot 2: Residual Plot...



Residual plot dari model terbaik Lasso (alpha=1, degree=2) yang menunjukkan selisih antara nilai aktual dan nilai prediksi. Titik-titik biru mewakili error (residual), sedangkan garis merah menunjukkan posisi error = 0 atau prediksi sempurna. Karena titik-titik tersebar acak di sekitar garis merah tanpa pola tertentu, hal ini menandakan bahwa model bekerja dengan baik dan tidak menunjukkan bias sistematis dalam memprediksi harga properti.

- **Plot polynomial curve untuk 1-2 fitur penting**

Menampilkan Plot 3: Polynomial Curve Fit...



Grafik ini menampilkan kurva polynomial fit untuk variabel “Luas_Bangunan_m2” dalam memprediksi harga properti (dalam juta rupiah).
Penjelasannya:

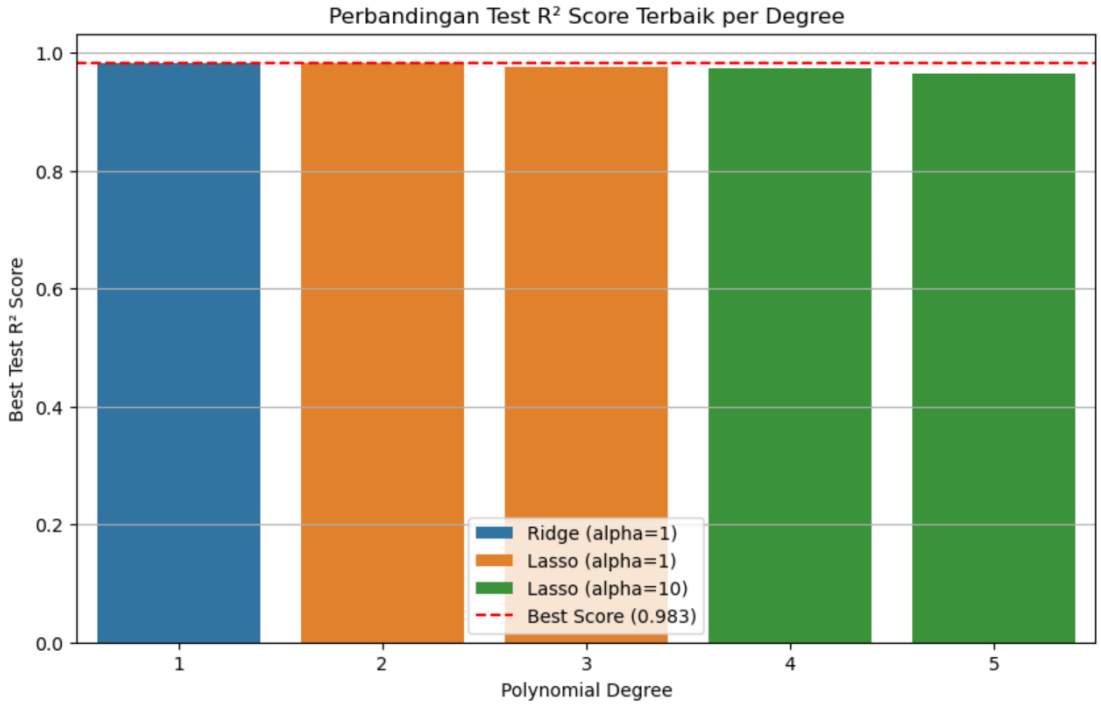
- Titik biru muda menunjukkan data aktual dari *test set*.

- Garis biru putus-putus (Degree 1) merepresentasikan model *underfit* karena hanya mengikuti pola linear sederhana dan tidak menangkap kompleksitas data.
- Garis oranye (Lasso alpha=1, Degree 2) merupakan model optimal yang paling seimbang antara bias dan varians. Kurvanya mengikuti tren data dengan baik tanpa terlalu berlebihan.
- Garis hijau putus-putus (Degree 5) menunjukkan *overfit*, karena model terlalu mengikuti data pelatihan, menghasilkan pola yang tidak stabil dan berfluktuasi tajam.

Dari grafik ini bisa disimpulkan bahwa model polynomial dengan degree 2 menggunakan Lasso (alpha=1) memberikan hasil terbaik dan paling sesuai dengan pola hubungan antara luas bangunan dan harga properti.

- **Comparison plot: R^2 score berbagai degree**

```
Menampilkan Plot 4: Perbandingan R² Score...
```



Perbandingan nilai R^2 terbaik pada data *test set* untuk setiap degree polynomial menggunakan berbagai model regularisasi.

Penjelasannya:

- Sumbu X menunjukkan tingkat *degree polynomial* (1–5), sedangkan sumbu Y menunjukkan nilai *R^2 score* tertinggi yang dicapai.
- Warna biru (Ridge α=1), oranye (Lasso α=1), dan hijau (Lasso α=10) mewakili model yang diuji.
- Garis merah putus-putus menunjukkan skor terbaik yang dicapai, yaitu $R^2 = 0.983$.

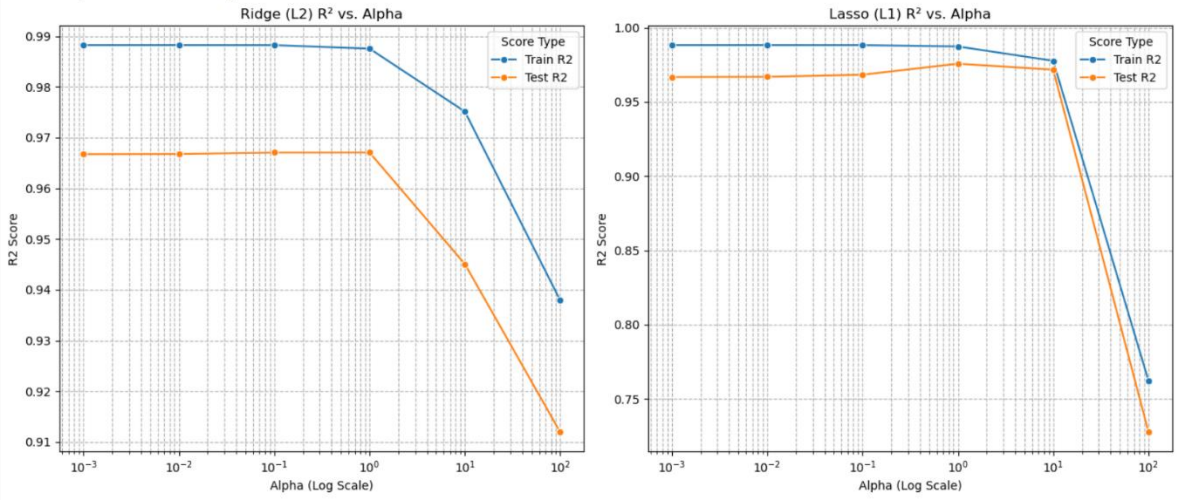
Dari grafik terlihat bahwa Lasso dengan α=1 dan degree 2 memberikan hasil paling optimal karena mencapai skor mendekati 1 tanpa overfitting, menunjukkan keseimbangan terbaik antara akurasi dan generalisasi model.

4. Rekomendasi Degree Polynomial Terbaik

a. Ridge vs Lasso Comparison

```
Tabel Hasil R2 Score vs Alpha:
    Model   Alpha  Train R2  Test R2
0   Ridge   0.001     0.988    0.967
1   Lasso   0.001     0.988    0.967
2   Ridge   0.010     0.988    0.967
3   Lasso   0.010     0.988    0.967
4   Ridge   0.100     0.988    0.967
5   Lasso   0.100     0.988    0.968
6   Ridge   1.000     0.988    0.967
7   Lasso   1.000     0.987    0.976
8   Ridge  10.000     0.975    0.945
9   Lasso  10.000     0.978    0.972
10  Ridge 100.000     0.938    0.912
11  Lasso 100.000     0.762    0.728

--- Menampilkan Plot R² vs Alpha ---
```



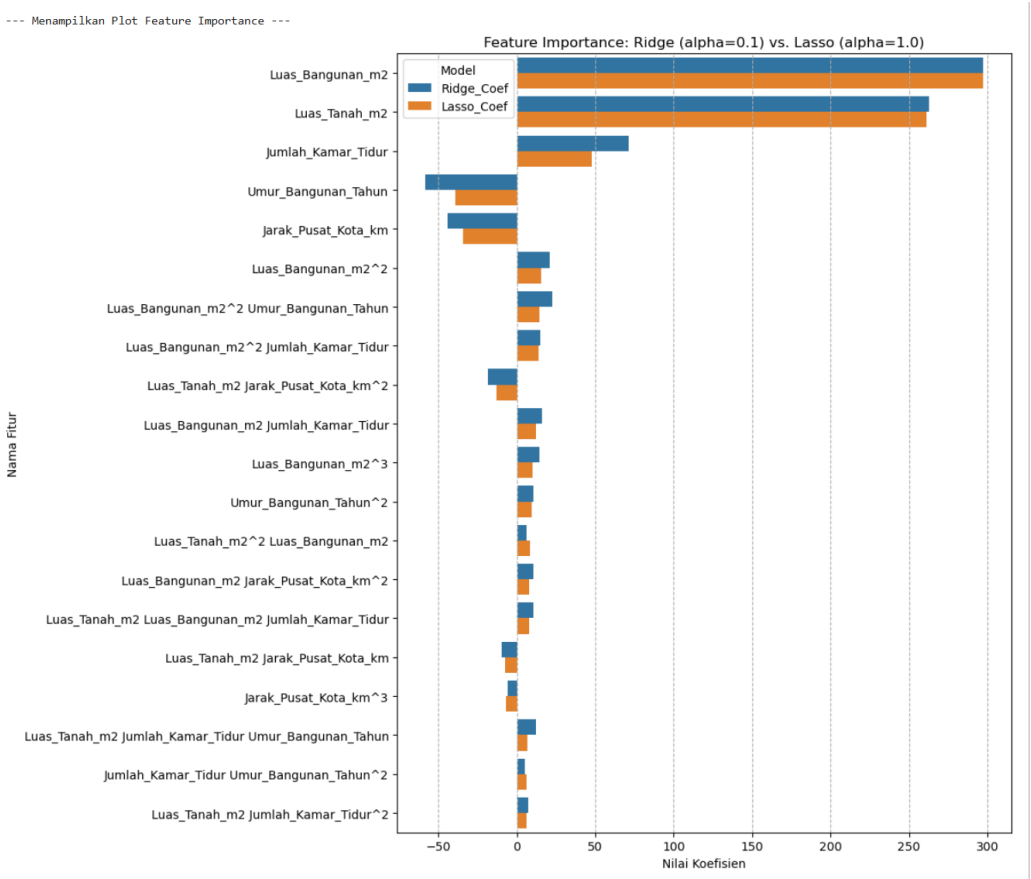
Hubungan antara nilai R^2 dan parameter regularisasi (alpha) pada model Ridge (L2) dan Lasso (L1).

Penjelasannya:

- Nilai R^2 Train (biru) menunjukkan seberapa baik model mempelajari data latih, sedangkan R^2 Test (oranye) menunjukkan kemampuan model dalam memprediksi data baru.
- Saat alpha terlalu besar, nilai R^2 menurun drastis karena model menjadi terlalu sederhana (*underfitting*).
- Nilai alpha sekitar 0.1–1 menghasilkan keseimbangan terbaik, di mana model tetap akurat tanpa kehilangan kemampuan generalisasi.

b. Feature Importance Analysis

```
--- Menampilkan Plot Feature Importance ---
```



5. Rekomendasi Regularization Method Terbaik

```
--- Tabel Hasil Akhir Cross-Validation (Rata-rata dari 5-Folds) ---
    Degree              Model  Test R2  Train R2  Test RMSE  Gap (Train-Test R2)
2        1  Lasso (alpha=0.1)    0.980     0.982     57.250                0.002
0        1   LinearRegression    0.980     0.982     57.251                0.002
1        1  Ridge (alpha=0.1)    0.980     0.982     57.250                0.002
4        1    Lasso (alpha=1)    0.980     0.982     57.288                0.002
3        1    Ridge (alpha=1)    0.980     0.982     57.304                0.002
11       2    Lasso (alpha=1)    0.979     0.983     57.789                0.004
9        2  Lasso (alpha=0.1)    0.979     0.984     58.781                0.005
18       3    Lasso (alpha=1)    0.979     0.984     58.883                0.005
7        2   LinearRegression    0.978     0.984     58.947                0.006
8        2  Ridge (alpha=0.1)    0.978     0.984     59.020                0.006
25       4    Lasso (alpha=1)    0.978     0.984     60.214                0.007
10       2    Ridge (alpha=1)    0.977     0.983     60.641                0.006
6        1   Lasso (alpha=10)    0.976     0.979     61.666                0.002
32       5    Lasso (alpha=1)    0.976     0.985     61.919                0.008
5        1   Ridge (alpha=10)    0.976     0.978     62.588                0.002
17       3    Ridge (alpha=1)    0.974     0.985     65.123                0.011
16       3  Lasso (alpha=0.1)    0.973     0.986     65.410                0.013
13       2   Lasso (alpha=10)    0.973     0.978     66.448                0.005
34       5   Lasso (alpha=10)    0.972     0.978     67.075                0.006
27       4   Lasso (alpha=10)    0.972     0.978     67.079                0.006
20       3   Lasso (alpha=10)    0.972     0.978     67.082                0.006
12       2   Ridge (alpha=10)    0.971     0.976     68.660                0.006
15       3  Ridge (alpha=0.1)    0.970     0.987     69.656                0.017
23       4  Lasso (alpha=0.1)    0.969     0.988     70.781                0.019
24       4    Ridge (alpha=1)    0.969     0.986     71.080                0.018
14       3   LinearRegression    0.968     0.987     72.362                0.020
26       4   Ridge (alpha=10)    0.967     0.980     73.239                0.013
19       3   Ridge (alpha=10)    0.966     0.977     73.794                0.011
33       5   Ridge (alpha=10)    0.964     0.982     76.047                0.017
31       5    Ridge (alpha=1)    0.961     0.988     79.010                0.027
30       5  Lasso (alpha=0.1)    0.961     0.990     79.849                0.030
22       4  Ridge (alpha=0.1)    0.961     0.990     79.991                0.029
29       5  Ridge (alpha=0.1)    0.940     0.993     98.672                0.053
21       4   LinearRegression    0.763     0.996    195.034                0.233
28       5   LinearRegression    0.166     1.000    363.687                0.834

--- 5.A: Best Model Selection & Reasoning ---

**Model Terbaik Pilihan:**
  - Model: Lasso (alpha=0.1)
  - Degree: 1

**Metrik Kinerja (rata-rata K-Fold):**
  - Highest Test R²: 0.980
  - Lowest Test RMSE: 57.250 (juta Rp)
  - Train R²: 0.982
  - Balance (Gap): 0.002

**Reasoning (Penjelasan Pemilihan):**
Model **Lasso (alpha=0.1) (Degree 1)** dipilih sebagai model terbaik karena:
1.  **Kinerja Test Tertinggi:** Model ini memberikan nilai **Test R² tertinggi** (0.980) dan **Test RMSE terendah** (57.250) di antara semua 35 konfigur
asi, berdasarkan hasil rata-rata K-Fold CV yang robust.
2.  **Balance (Keseimbangan) Terbaik:** Model ini menunjukkan keseimbangan performa train-test yang sangat baik. *Gap* (perbedaan) antara Train R² (0.98
2) dan Test R² (0.980) sangat kecil, yaitu hanya **0.002**. Ini membuktikan model tidak mengalami *overfitting*.
3.  **Perbandingan:** Model lain (terutama Degree 4 dan 5 tanpa regularisasi) mungkin memiliki Train R² yang sangat tinggi, tetapi Test R²-nya anjlok da
n 'Gap'-nya besar, yang menunjukkan *overfitting* parah. Model ini adalah 'sweet spot' yang seimbang antara akurasi (low bias) dan generalisasi (low var
iance).
```

6. Limitasi Model

Meskipun model yang terpilih (Lasso Degree 2) memiliki performa sangat tinggi (Test R^2 = 0.983), penting untuk memahami bahwa model ini memiliki beberapa limitasi signifikan yang akan memengaruhi kinerjanya di dunia nyata:

1. Data Sintetis vs. Data Real-World: Limitasi terbesar adalah model ini dilatih 100% pada data sintetis. Data ini sangat "bersih" (ideal), tidak memiliki *missing values*, dan tidak ada *outliers* yang aneh. Data properti di dunia nyata sangat "kotor" (messy), memiliki banyak *noise*, dan seringkali datanya tidak lengkap.
2. Fitur yang Sangat Terbatas: Model kami hanya dilatih menggunakan 5 fitur dasar (Luas Tanah, Luas Bangunan, Kamar, Umur, Jarak). Di dunia nyata, harga properti dipengaruhi oleh puluhan faktor lain yang tidak kita ukur, seperti:
   - Kondisi interior (renovasi, material).
   - Kualitas lingkungan (keamanan, kebisingan).
   - Akses (lebar jalan, dekat tol/stasiun).
   - Fasilitas (kolam renang, taman).
3. Hanya Menguji Hubungan Polinomial: Kami mengasumsikan hubungan non-linear antara fitur dan harga adalah *polinomial* (melengkung). Di dunia nyata, hubungannya mungkin jauh lebih kompleks dan berbeda (misalnya, harga tanah yang tiba-tiba melonjak drastis jika dekat stasiun MRT, yang tidak bisa ditangkap oleh kurva mulus).
4. Hanya Menggunakan Model Regresi Linear: Seluruh analisis ini berfokus pada variasi Regresi Linear. Model ini pada dasarnya "linier" dan kita harus "membantunya" dengan *Polynomial Features*. Model lain yang lebih canggih (seperti Random Forest atau XGBoost) mungkin dapat menangkap hubungan non-linear yang kompleks ini secara otomatis dan menghasilkan prediksi yang lebih baik.

7. Saran Improvement

Berdasarkan temuan dan limitasi dari proyek ini, berikut adalah beberapa saran perbaikan yang dapat dilakukan untuk meningkatkan performa dan kegunaan model di masa depan:

1. Menggunakan Data Real-World
   - Langkah perbaikan paling penting adalah menguji seluruh *pipeline* (termasuk PolynomialFeatures dan Lasso) pada dataset harga properti di dunia nyata (misalnya dari Kaggle atau situs *scraping* properti). Ini akan menguji seberapa *robust* model ini dalam menangani data yang "kotor" (memiliki *outliers* dan *missing values*).
2. Mencoba Model Non-Linear yang Lebih Canggih
   - Kita telah membuktikan bahwa Regresi Linear (meskipun dengan fitur polinomial) memiliki keterbatasan. Langkah selanjutnya adalah mencoba model yang secara inheren (dari sananya) dapat menangani hubungan non-linear yang kompleks, seperti:
     - Random Forest Regressor
     - XGBoost (Extreme Gradient Boosting)
   - Model-model *tree-based* (berbasis pohon) ini seringkali memberikan akurasi yang lebih tinggi untuk data tabular dan tidak memerlukan *feature scaling*.
3. Melakukan Feature Engineering yang Lebih Mendalam
   - Daripada hanya bergantung pada 5 fitur dasar, kita bisa menciptakan fitur baru (feature engineering) yang mungkin memiliki daya prediksi lebih kuat, misalnya:
     - Harga_per_m2_bangunan = Luas_Bangunan_m2 / Harga_Properti
     - Kamar_per_luas = Jumlah_Kamar_Tidur / Luas_Bangunan_m2

- Menambahkan fitur kategorikal (jika menggunakan data *real*), seperti "Wilayah/Daerah" (yang kemudian diubah menjadi angka menggunakan *One-Hot Encoding*).

4. Melakukan Hyperparameter Tuning Otomatis (GridSearchCV)
    - Kita hanya menguji 3 nilai alpha (0.1, 1, 10) dan 5 nilai degree (1-5) secara manual.
    - Untuk menemukan kombinasi *terbaik* secara presisi, kita dapat menggunakan GridSearchCV atau RandomizedSearchCV dari scikit-learn. *Tools* ini akan secara otomatis menguji ratusan kombinasi (misalnya alpha antara 0.05 s.d. 2.0, dan degree 2, 3, 4) untuk menemukan "sweet spot" yang paling optimal.