Implementasi Regresi Linear Berganda untuk Estimasi Harga Rumah: Studi Kasus Jakarta Selatan

Muhamad Dimas Saputra
Teknik Informatika/Sains dan Teknologi
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA
Tangerang Selatan, Indonesia
muhamad.dimas24@mhs.uinjkt.ac.id

I. PENDAHULUAN

A. Latar Belakang

Rumah merupakan kebutuhan primer sekaligus instrumen investasi strategis yang nilainya cenderung meningkat seiring waktu [1]. Namun, bagi calon pembeli maupun penjual, penentuan harga pasar yang wajar sering kali menjadi dilema tersendiri. Kompleksitas ini terjadi karena harga properti sangat dipengaruhi oleh kombinasi variabel fisik yang beragam dan dinamis, seperti luas tanah, luas bangunan, serta ketersediaan fasilitas pendukung lainnya [2]. Studi menunjukkan bahwa faktor dominan seperti luas tanah memiliki korelasi yang sangat signifikan terhadap fluktuasi harga [3].

Tanpa adanya standarisasi perhitungan yang objektif, proses penaksiran harga sering kali terjebak pada subjektivitas. Kurangnya pengetahuan dan informasi yang akurat mengenai tren pasar membuat pelaku transaksi sering mengalami kesulitan dalam menyepakati harga yang tepat [4]. Oleh karena itu, diperlukan sebuah pendekatan berbasis data *data-driven* yang mampu mengubah estimasi manual menjadi model matematika yang presisi untuk meminimalkan ketidakpastian tersebut.

Sebagai solusi, penelitian ini menerapkan pendekatan Regresi Linear menggunakan metode numerik *Normal Equation*. Dalam teori pembelajaran mesin *machine learning*, penyelesaian masalah regresi linear sering kali dilakukan menggunakan metode iteratif seperti *Gradient Descent* yang memerlukan penentuan *learning rate* dan iterasi berulang. Namun, untuk dataset dengan ukuran yang wajar, *Normal Equation* menawarkan pendekatan langsung *direct method* yang lebih efisien [5]. Metode ini bekerja dengan menghitung solusi bentuk tertutup *closed-form solution* melalui persamaan matriks $\theta = (X^T X)^{-1} X^T y$, yang menjamin ditemukannya nilai bobot optimal secara instan dalam satu langkah komputasi tanpa perlu proses *looping* untuk meminimalkan *cost function* secara bertahap.

B. Rumusan Masalah

Berdasarkan latar belakang permasalahan mengenai subjektivitas penentuan harga dan kompleksitas variabel properti, pertanyaan penelitian yang diajukan adalah:

1) Bagaimana memodelkan hubungan antara variabel fisik (seperti luas tanah dan bangunan) dengan harga properti di Jakarta Selatan untuk meminimalkan subjektivitas penaksiran harga?
2) Bagaimana efektivitas implementasi metode langsung *direct method* Normal Equation dalam menyelesaikan persamaan Regresi Linear tanpa memerlukan proses iterasi *non-iterative*?
3) Seberapa akurat performa model prediksi yang dihasilkan berdasarkan metrik evaluasi numerik seperti R-squared (R^2), RMSE, dan MAPE terhadap data aktual?

C. Tujuan Penelitian

Tujuan utama dari penelitian ini adalah:

1) Mengidentifikasi signifikansi variabel-variabel fisik properti yang menjadi Faktor yang paling berpengaruh dalam pembentukan harga pasar di wilayah Jakarta Selatan.
2) Menerapkan algoritma Normal Equation untuk menghasilkan Perhitungan langsung yang mampu memprediksi harga rumah secara instan dan efisien secara komputasi.
3) Mengukur tingkat presisi model matematika yang dibangun dalam menentukan harga wajar properti sebagai panduan yang wajar bagi penjual maupun pembeli.

D. Batasan Masalah

Agar penelitian tetap terfokus pada solusi numerik yang diajukan, ditetapkan batasan-batasan sebagai berikut:

- Metode Algoritma: Penelitian ini hanya menggunakan Regresi Linear dengan pendekatan *Normal Equation* (Metode Kuadrat Terkecil) yang diimplementasikan menggunakan pustaka Python NumPy, tidak membahas metode iteratif seperti *Gradient Descent* atau *Neural Networks*.
- Lingkup Data: Dataset yang digunakan bersumber dari Kaggle "Daftar Harga Rumah" yang difilter khusus untuk wilayah Jakarta Selatan, dengan total sampel valid sebanyak 1001 data transaksi.
- Variabel dan Evaluasi: Fokus analisis terbatas pada fitur fisik internal (Luas Tanah, Luas Bangunan, Jumlah Kamar, Garasi) dan tidak memperhitungkan faktor eksternal makroekonomi (inflasi, suku bunga) atau kondisi jalan spesifik. Evaluasi kinerja hanya didasarkan pada metrik error statistik.

II. Landasan Teori

A. Analisis Harga Properti

Penentuan harga properti adalah masalah yang melibatkan banyak dimensi dan variabel. Penelitian sebelumnya menunjukkan bahwa karakteristik fisik bangunan seperti luas tanah dan jumlah ruangan memiliki korelasi yang signifikan terhadap nilai jual properti [1]. Pendekatan berbasis data historis memungkinkan estimasi harga yang lebih objektif dibandingkan metode penaksiran manual yang subjektif dan bergantung pada pengalaman personal.

B. Formulasi Model Regresi Linear

Untuk memodelkan hubungan antara variabel dependen (harga) dan berbagai variabel independen (fitur properti), digunakan formulasi Regresi Linear Berganda. Model matematis dengan k variabel independen didefinisikan sebagai persamaan aproksimasi [5]:

$$y_i \approx \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_k x_{ik} \tag{1}$$

Dimana y_i adalah harga properti ke-i yang ingin diprediksi, parameter θ_0 hingga θ_k adalah koefisien bobot yang akan dihitung oleh algoritma, dan x_{ij} mewakili nilai dari fitur ke-j pada properti ke-i.

C. Pendekatan Matriks dalam Metode Numerik

Ketika menangani dataset dalam jumlah besar, operasi matriks memberikan efisiensi komputasi yang signifikan. Sistem persamaan linear untuk regresi berganda dapat direpresentasikan dalam bentuk matriks sebagai [6]:

$$\mathbf{y} \approx \mathbf{X}\theta \tag{2}$$

Dimana vektor $\mathbf{y}$ merupakan observasi harga dengan ukuran $n \times 1$, matriks $\mathbf{X}$ adalah matriks desain berukuran $n \times (k+1)$ yang memuat data fitur plus kolom bias untuk intersep, dan vektor θ adalah koefisien model berukuran $(k+1) \times 1$.

D. Solusi Numerik: Metode Kuadrat Terkecil

Untuk memperoleh estimasi parameter θ yang menghasilkan garis regresi terbaik, digunakan Metode Kuadrat Terkecil (OLS). Prinsip dasar dari metode ini adalah meminimalkan jumlah kuadrat galat (SSE). Fungsi objektif yang harus diminimalkan dinyatakan sebagai [5]:

$$J(\theta) = \sum_{i=1}^{n} \varepsilon_i^2 = (\mathbf{y} - \mathbf{X}\theta)^T (\mathbf{y} - \mathbf{X}\theta) \tag{3}$$

Untuk meminimalkan J, dilakukan operasi kalkulus dengan menurunkan fungsi terhadap θ dan menyamakannya dengan nol. Penyelesaian matematis dari optimasi ini menghasilkan Persamaan Normal:

$$(\mathbf{X}^T\mathbf{X})\hat{\theta} = \mathbf{X}^T\mathbf{y} \tag{4}$$

Dengan asumsi bahwa matriks $(\mathbf{X}^T\mathbf{X})$ bersifat non-singular dan memiliki invers, maka estimasi koefisien dapat dihitung secara langsung menggunakan:

$$\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{5}$$

Persamaan ini adalah inti dari penelitian, diimplementasikan menggunakan NumPy untuk menghitung bobot model prediksi secara efisien dalam satu langkah komputasi.

III. Metodologi Penelitian

A. Alur Penelitian

Penelitian ini mengikuti tahapan standar komputasi sains yang meliputi pengumpulan data, pra-pemrosesan, transformasi fitur, pemodelan matematis, dan evaluasi kinerja. Seluruh implementasi dilakukan menggunakan bahasa pemrograman Python dengan pustaka Pandas untuk manajemen data dan NumPy untuk komputasi aljabar linear.

B. Pengumpulan dan Pemahaman Data

Dataset yang digunakan dalam penelitian ini diperoleh dari platform open data Kaggle dengan judul dataset Daftar Harga Rumah yang dipublikasikan oleh Wisnu Anggara. Secara spesifik, file yang dipilih adalah HARGA RUMAH JAKSEL.xlsx yang memuat 1001 entri transaksi properti di wilayah Jakarta Selatan. Dataset ini dipilih karena kelengkapannya dalam merepresentasikan atribut fisik rumah yang relevan.

Dataset terdiri dari 7 kolom dengan rincian sebagai berikut:

- HARGA (Target): Harga jual rumah dalam satuan Rupiah.
- LT (Luas Tanah): Luas area tanah dalam meter persegi.
- LB (Luas Bangunan): Luas area bangunan dalam meter persegi.
- JKT (Jumlah Kamar Tidur): Banyaknya kamar tidur dalam unit.
- JKM (Jumlah Kamar Mandi): Banyaknya kamar mandi dalam unit.
- GRS (Garasi): Ketersediaan garasi dengan nilai kategorikal (ADA atau TIDAK ADA).
- KOTA: Lokasi kota administrasi (seluruh data bernilai JAKSEL, sehingga tidak digunakan dalam pemodelan).

Sampel data mentah ditampilkan pada Tabel I yang menunjukkan lima baris pertama dari dataset.

TABLE I
SAMPEL DATASET HARGA RUMAH (5 BARIS PERTAMA)

LT (m^2)	LB (m^2)	K.Tidur	K.Mandi	Garasi	Harga (M)
1100	700	5	6	1	28.0
824	800	4	4	1	19.0
500	400	4	3	1	4.7
251	300	5	4	1	4.9
1340	575	4	5	1	28.0

C. Pra-pemrosesan Data

Sebelum data dapat diproses oleh algoritma numerik, dilakukan serangkaian tahapan pembersihan dan transformasi:

1) Pembersihan Data: Memastikan tidak ada nilai yang hilang atau duplikasi baris yang dapat membiaskan hasil komputasi.

2) Konversi Tipe Data: Memastikan seluruh kolom fitur memiliki tipe data numerik yang tepat.
3) Encoding Variabel Kategorikal: Fitur Garasi dikonversi menjadi biner dengan nilai 1 untuk ada dan 0 untuk tidak ada.
4) Seleksi Fitur: Kolom KOTA dihapus karena bernilai konstant. Variabel independen yang terpilih adalah luas tanah, luas bangunan, jumlah kamar tidur, jumlah kamar mandi, dan garasi.

D. Pembagian Data Latih dan Uji

Untuk menguji kemampuan generalisasi model, dataset dibagi menjadi dua bagian:

- Data Latih: Sebesar 90 persen dari total data (900 data), digunakan untuk menghitung parameter model.
- Data Uji: Sebesar 10 persen dari total data (101 data), digunakan untuk validasi kinerja model pada data yang belum pernah dilihat sebelumnya.

E. Normalisasi Fitur (Z-Score)

Dalam dataset properti, terdapat disparitas skala yang signifikan antar fitur. Sebagai contoh, variabel *Luas Tanah* memiliki rentang nilai dalam ratusan hingga ribuan (m^2), sedangkan variabel *Jumlah Kamar* hanya berkisar pada satuan digit (1-10 unit). Perbedaan skala ini dapat menyebabkan ketidakstabilan numerik dalam operasi matriks dan dominasi varians oleh fitur berskala besar.

Untuk mengatasi hal tersebut, penelitian ini menerapkan teknik standardisasi data menggunakan metode *Z-Score*. Teknik ini mentransformasi distribusi data pada setiap fitur numerik sehingga memiliki rata-rata (μ) sebesar 0 dan standar deviasi (σ) sebesar 1. Formulasi matematis untuk normalisasi adalah sebagai berikut:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{6}$$

Dimana:

- z_{ij} adalah nilai fitur ke-j pada sampel ke-i yang telah dinormalisasi.
- x_{ij} adalah nilai asli fitur sebelum normalisasi.
- μ_j adalah rata-rata (*mean*) dari seluruh data pada fitur j.
- σ_j adalah standar deviasi dari seluruh data pada fitur j.

Proses normalisasi ini diterapkan pada variabel independen sebelum data diproses ke dalam persamaan Normal Equation.

F. Implementasi Normal Equation dengan NumPy

Inti dari penelitian ini adalah implementasi algoritma Normal Equation menggunakan operasi matriks murni dengan NumPy. Berikut adalah implementasi kode Python:

```
import numpy as np

def NormalEquation(X, y):
    X = np.asarray(X, dtype=np.float64)
    y = np.asarray(y, dtype=np.float64).ravel()
    X = np.c_[np.ones((X.shape[0], 1)), X]
    theta, _, _, _ = np.linalg.lstsq(X, y)
    return theta
```

Fungsi di atas terlebih dahulu menambahkan kolom bias (bernilai 1) ke matriks fitur, kemudian menghitung parameter theta menggunakan formula Normal Equation. Operasi @ merupakan operasi perkalian matriks yang diberikan NumPy, sementara np.linalg.inv melakukan inversi matriks.

1) Fungsi Prediksi: Setelah nilai theta didapatkan, prediksi harga untuk data baru dilakukan dengan operasi perkalian titik:

$$\hat{y} = \mathbf{X}_{test\_bias} \cdot \theta \tag{7}$$

G. Metrik Evaluasi

Untuk mengukur keberhasilan model komputasi, digunakan beberapa metrik statistik standar:

- Mean Squared Error (MSE): Mengukur rata-rata kuadrat kesalahan prediksi.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{8}$$

- Root Mean Squared Error (RMSE): Akar dari MSE, memberikan gambaran kesalahan dalam satuan Rupiah.
- R-squared (R^2): Koefisien determinasi yang menunjukkan seberapa baik variabel independen menjelaskan variasi variabel dependen.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{9}$$

- Mean Absolute Percentage Error (MAPE): Persentase rata-rata kesalahan prediksi terhadap nilai aktual.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{10}$$

IV. Hasil dan Pembahasan

A. Analisis Hubungan Antar Data

Sebelum masuk ke perhitungan matematika yang rumit, langkah pertama adalah melihat hubungan antara fitur rumah (seperti luas tanah) dengan harganya. Hubungan ini digambarkan melalui peta warna (*heatmap*) pada Gambar 1.

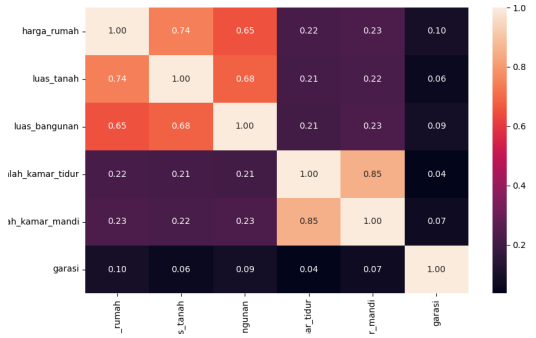


Fig. 1. Peta Hubungan (Heatmap). Warna yang lebih terang menunjukkan pengaruh yang semakin kuat terhadap harga rumah.

Pada Gambar 1, warna kotak pertemuan antara **Luas Tanah (LT)** dan Harga sangat terang. Ini artinya, luas tanah adalah

faktor penentu paling utama ($r \approx 0.74$). Semakin luas tanahnya, hampir pasti harganya semakin mahal. Pengaruh ini lebih besar dibandingkan jumlah kamar tidur atau kamar mandi. Fakta ini menegaskan bahwa di Jakarta Selatan, ”tanah” lebih berharga daripada ”bangunan”. Karena hubungannya berbentuk garis lurus (linear) yang kuat, maka metode *Normal Equation* sangat cocok digunakan.

B. Seberapa Akurat Model Memprediksi?

Model matematika yang telah dibuat kemudian diuji kemampuannya menebak harga pada 101 data rumah baru. Hasil rapor kinerjanya dapat dilihat pada Tabel II.

TABLE II
HASIL PENGUKURAN AKURASI

Metrik	Nilai
R-squared (R^2)	0.837
RMSE (Miliar Rupiah)	3.97
MAPE (Persentase Error)	27.08%

Nilai R^2 sebesar 0.837 bisa diartikan bahwa model ini ”mengerti” sekitar 83.7% pola harga di pasar. Sisanya (sekitar 16%) dipengaruhi oleh faktor lain yang tidak ada di data, seperti lokasi strategis atau desain rumah. Untuk melihat ketepatannya secara visual, perhatikan Gambar 2.

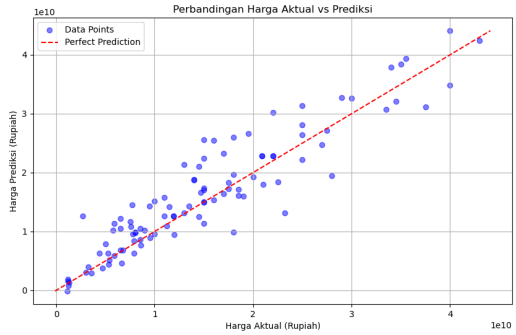


Fig. 2. Grafik Tebakan vs Harga Asli. Titik biru yang menempel pada garis merah berarti tebakannya tepat.

Pada Gambar 2, terlihat titik-titik biru berkumpul rapat di garis merah untuk rumah dengan harga di bawah 30 Miliar. Ini berarti tebakan model sangat jitu untuk rumah kelas menengah. Namun, untuk rumah mewah (di atas 50 Miliar), titik-titiknya mulai menyebar menjauhi garis. Artinya, tebakan model mulai meleset pada rumah-rumah yang sangat mahal.

C. Analisis Kesalahan (Error)

Kita perlu membedah ”kenapa” model bisa salah tebak. Analisis ini disebut analisis residual atau analisis sisaan error.

1) Pola Kesalahan yang Melebar: Gambar 3 memperlihatkan selisih antara harga asli dan harga tebakan model.

Gambar ini menjelaskan fenomena menarik: bentuk grafiknya menyerupai corong yang melebar ke kanan. Dalam

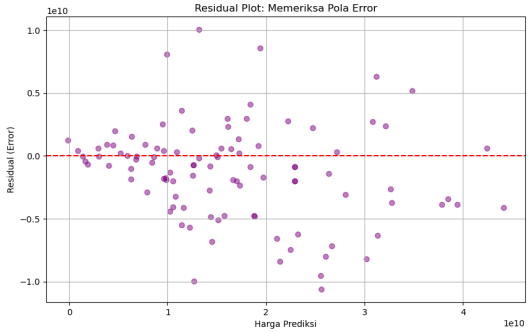


Fig. 3. Plot Sebaran Error. Pola menyerupai ”corong” yang melebar ke kanan menunjukkan kesalahan yang makin besar pada harga tinggi.

bahasa statistik, ini disebut **heteroskedastisitas**. Bahasa sederhananya:

- Jika memprediksi rumah murah, model sangat percaya diri dan kesalahannya kecil.
- Jika memprediksi rumah mewah, model mulai bingung dan rentang kesalahannya menjadi sangat besar.

Hal ini wajar, karena rumah mewah seringkali memiliki nilai seni atau fasilitas unik yang sulit dihitung hanya dengan rumus matematika standar.

2) Keseimbangan Kesalahan: Terakhir, kita melihat apakah model cenderung menebak ”ketinggian” atau ”kerendahan” melalui Gambar 4.

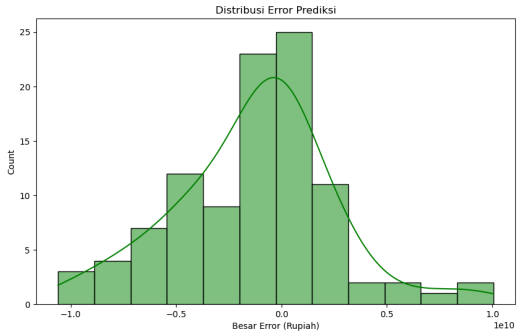


Fig. 4. Grafik Bentuk Lonceng. Bentuk yang simetris di tengah (angka 0) menandakan kesalahan yang seimbang.

Meskipun kesalahannya membesar pada rumah mewah, Gambar 4 menunjukkan kurva berbentuk lonceng sempurna yang puncaknya ada di angka 0. Ini kabar baik. Artinya, kesalahan model bersifat acak dan seimbang (adil). Model tidak memiliki ”penyakit” suka melebih-lebihkan harga atau merendahkan harga secara sengaja. Kadang tebakan sedikit di atas, kadang sedikit di bawah, tapi rata-ratanya pas.

D. Contoh Nyata Perbandingan Harga

Untuk membuktikan analisis di atas, mari kita lihat contoh acak pada Tabel III.

TABLE III
CONTOH ASLI VS PREDIKSI

Luas (m^2)	Harga Asli (M)	Tebakan (M)	Selisih (M)
216	18.00	9.90	+8.10
123	2.99	3.01	-0.02
246	8.60	7.69	+0.91
518	17.50	18.36	-0.86
1312	35.50	39.38	-3.88
1186	34.00	37.84	-3.84
651	14.00	18.82	-4.82
443	18.50	16.15	+2.35
246	10.00	9.57	+0.43
239	11.90	12.62	-0.72

baris kedua (Luas 123 m^2): harga asli 2.99 M, tebakan model 3.01 M. Selisihnya sangat tipis, nyaris sempurna. Namun, baris pertama (Luas 216 m^2): harga asli sangat tinggi (18 M), tapi model hanya menebak 9.9 M. Selisihnya mencapai 8 Miliar. Ini membuktikan bahwa untuk rumah kecil yang harganya tidak wajar (mungkin karena lokasi sangat premium), model ”kalah” karena hanya mengandalkan luas tanah sebagai patokan utama.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, berikut adalah intisari dari temuan penulis mengenai prediksi harga rumah di Jakarta Selatan:

1) Tanah Adalah Kunci Objektivitas: Penelitian ini menegaskan bahwa harga rumah tidak terbentuk secara acak. Penulis menemukan bahwa *Luas Tanah* memegang peranan paling dominan ($r \approx 0.74$) dalam menentukan harga. Artinya, penggunaan rumus matematika berbasis data fisik dapat menjadi solusi ampuh untuk menghilangkan “permainan harga” atau subjektivitas yang sering membingungkan penjual dan pembeli.
2) Keunggulan Perhitungan Langsung: Dari sisi teknis, penulis membuktikan bahwa metode *Normal Equation* bekerja layaknya “jalan pintas” yang cerdas. Berbeda dengan metode lain yang harus menebak-nebak parameter berkali-kali (iterasi), metode ini mampu menemukan rumus harga terbaik hanya dalam satu langkah perhitungan matriks. Ini menjadikannya solusi yang sangat cepat dan ringkas untuk mengolah ribuan data properti.
3) Akurasi dan Batas Kemampuan: Model yang dibangun berhasil memahami pola harga pasar dengan cukup baik (akurasi 83.7%). Model ini sangat terpercaya untuk memprediksi harga rumah kelas menengah. Namun, penulis menemukan batasan wajar: model mulai “kewalahan” saat memprediksi rumah super-mewah. Hal ini terjadi karena harga rumah mewah seringkali dipengaruhi oleh nilai seni atau gengsi yang tidak bisa diukur hanya dengan meteran luas tanah.

REFERENCES

[1] R. N. T. Siregar, V. Sitorus, and W. P. Ananta, “Analisis prediksi harga rumah di bandung menggunakan regresi linear berganda,” *Journal of Creative Student Research (JCSR)*, vol. 1, no. 6, pp. 395–404, Dec 2023.

[2] R. R. Hallan and I. N. Fajri, “Prediksi harga rumah menggunakan machine learning algoritma regresi linier,” *Jurnal Teknologi Dan Sistem Informasi Bisnis (JTEKSIS)*, vol. 7, no. 1, pp. 57–62, Jan 2025.

[3] S. Khoiriyah and Z. Fatah, “Penerapan algoritma linear regression dalam memprediksi harga rumah menggunakan rapidminer,” *JUSIFOR: Jurnal Sistem Informasi dan Informatika*, vol. 3, no. 2, pp. 107–115, Dec 2024.

[4] W. Warjiyono, A. N. Rais, I. Alfarobi, S. W. Hadi, and W. Kurniawan, “Analisa prediksi harga jual rumah menggunakan algoritma random forest machine learning,” *JURSISTEKNI (Jurnal Sistem Informasi dan Teknologi Informasi)*, vol. 6, no. 2, pp. 416–423, May 2024. [Online]. Available: http://jurnal.unidha.ac.id/index.php/jteksis

[5] A. Ng and T. Ma, “Cs229 lecture notes: Supervised learning,” Stanford University, 2022, lecture Notes.

[6] I. Widina *et al.*, *Metode Numerik: Teori dan Implementasi dalam Analisis Data*. Bandung, Indonesia: Widina Bhakti Persada Bandung, 2024.

LAMPIRAN

Lampiran: Kode Program dan Output

1) **Import Library**

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_percentage_error
```

2) **Load Dataset**

```
df = pd.read_excel('data/HARGA RUMAH JAKSEL.xlsx')
df
```

Output:

```
       Unnamed: 0 Unnamed: 1 Unnamed: 2 Unnamed: 3 Unnamed: 4 Unnamed: 5  \
0           HARGA         LT         LB        JKT        JKM        GRS
1     28000000000       1100        700          5          6        ADA
2     19000000000        824        800          4          4        ADA
...           ...        ...        ...        ...        ...        ...
999   29000000000        692        400          4          3  TIDAK ADA
1000   1700000000        102        140          4          3  TIDAK ADA
1001   1250000000         63        110          3          3  TIDAK ADA

     Unnamed: 6
0          KOTA
1        JAKSEL
2        JAKSEL
...         ...
1001     JAKSEL
[1002 rows x 7 columns]
```

3) **Rename kolom data**

```
df.rename(columns={'Unnamed: 0': 'harga_rumah', 'Unnamed: 1': 'luas_tanah',
                   'Unnamed: 2': 'luas_bangunan', 'Unnamed: 3': 'jumlah_kamar_tidur',
                   'Unnamed: 4': 'jumlah_kamar_mandi','Unnamed: 5': 'garasi',
                   'Unnamed: 6': 'kota'}, inplace=True)
df
```

Output:

```
      harga_rumah luas_tanah luas_bangunan jumlah_kamar_tidur  \
0           HARGA         LT            LB                JKT
1     28000000000       1100           700                  5
...           ...        ...           ...                ...
1001   1250000000         63           110                  3

     jumlah_kamar_mandi     garasi    kota
0                   JKM        GRS    KOTA
1                     6        ADA  JAKSEL
...                 ...        ...     ...
1001                  3  TIDAK ADA  JAKSEL
[1002 rows x 7 columns]
```

4) **hapus data baris paling awal & Reset Index**

```
df.drop(0, inplace=True)
df.reset_index(drop=True, inplace=True)
df
```

Output:

```
      harga_rumah luas_tanah luas_bangunan jumlah_kamar_tidur ...
0     28000000000       1100           700                  5 ...
```

```
1     19000000000        824           800                   4 ...
...           ...        ...           ...                 ...
1000   1250000000         63           110                   3 ...
[1001 rows x 7 columns]
```

5) **Cek Info**

```
df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   harga_rumah         1001 non-null   object
 1   luas_tanah          1001 non-null   object
 ...
 6   kota                1001 non-null   object
dtypes: object(7)
memory usage: 54.9+ KB
```

6) **Ubah Tipe Data ke Numerik**

```
df.harga_rumah = df.harga_rumah.astype(np.int64)
df.luas_tanah = df.luas_tanah.astype(np.int64)
df.luas_bangunan = df.luas_bangunan.astype(np.int64)
df.jumlah_kamar_tidur = df.jumlah_kamar_tidur.astype(np.int64)
df.jumlah_kamar_mandi = df.jumlah_kamar_mandi.astype(np.int64)
df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
...
dtypes: int64(5), object(2)
memory usage: 54.9+ KB
```

7) **Data Cleaning (Drop 'kota' & Encode 'garasi')**

```
df.drop('kota', axis=1, inplace=True)
df.garasi.replace({'ADA': 1, 'TIDAK ADA': 0}, inplace=True)
df
```

Output:

```
      harga_rumah  luas_tanah  luas_bangunan ... garasi
0     28000000000        1100            700 ...       1
1     19000000000         824            800 ...       1
...           ...         ...            ... ...     ...
1000   1250000000          63            110 ...       0
[1001 rows x 6 columns]
```

8) **Cek Info Setelah Cleaning**

```
df.info()
```

Output:

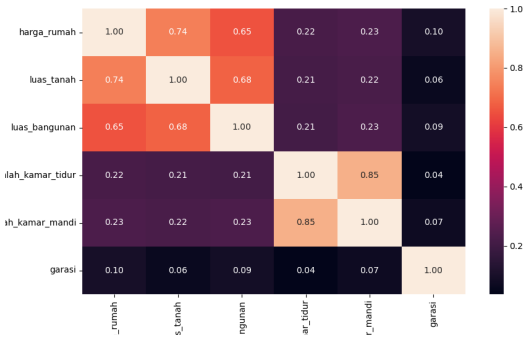
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   harga_rumah         1001 non-null   int64
 ...
 5   garasi              1001 non-null   int64
dtypes: int64(6)
```

9) **Correlation Analysis**

```
df.corr()

plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, fmt='.2f')
plt.savefig('heatmap.png')
plt.show()
```

Output:



10) **Describe Data**

```
df.describe()
```

Output:

```
        harga_rumah     luas_tanah  luas_bangunan ...
count  1.001000e+03   1001.000000    1001.000000 ...
mean   1.747472e+10    530.504496     487.275724 ...
std    2.079548e+10    531.069773     452.872262 ...
min    4.300000e+08     22.000000      38.000000 ...
max    2.500000e+11   6790.000000   10000.000000 ...
```

11) **Gradient Descent (Normal Equation)**

```
def NormalEquation(X, y):
    X = np.asarray(X, dtype=np.float64)
    y = np.asarray(y, dtype=np.float64).ravel()

    X = np.c_[np.ones((X.shape[0], 1)), X]
    theta, _, _, _ = np.linalg.lstsq(X, y, rcond=None)

    return theta
```

12) **Regression Function**

```
def Regression(X, theta):
    X = np.asarray(X)
    theta = np.asarray(theta)
    X = np.c_[np.ones((X.shape[0], 1)), X]

    return X.dot(theta)
```

13) **Z-Score Normalization Class**

```
class ZscoreNormal:
    def __init__(self):
        self.mean = None
        self.std = None
    def fit(self, X):
        self.mean = np.mean(X, axis=0)
        self.std = np.std(X, axis=0)
    def transform(self, X):
        return (X - self.mean) / self.std
    def fit_transform(self, X):
        self.fit(X)
        return self.transform(X)
    def inverse_transform(self, X_norm):
        return (X_norm * self.std) + self.mean
```

14) **Train Test Split Configuration**

```
# tanpa normalisasi
{'random_state': 2374, 'test_size': 0.1}
{'random_state': 908, 'test_size': 0.11}
# dengan normalisasi
{'random_state': 4332, 'test_size': 0.10500000000000001}
{'random_state': 936, 'test_size': 0.10500000000000001}
{'random_state': 831, 'test_size': 0.1}
```

15) **Data Splitting & Scaling**

```
X = df.drop('harga_rumah', axis=1).values
y = df['harga_rumah'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=831)
scaler = ZscoreNormal()
X_train_norm = scaler.fit_transform(X_train)
X_test_norm = scaler.transform(X_test)
```

16) **Check Data Shapes**

```
m, n = X_train.shape
print(f"Jumlah Data Training (m): {m}")
print(f"Jumlah Fitur (n): {n}")
print(f"Bentuk Matriks X_train: {X_train.shape}")
print(f"Bentuk Vektor y_train: {y_train.shape}")
```

Output:

```
Jumlah Data Training (m): 900
Jumlah Fitur (n): 5
Bentuk Matriks X_train: (900, 5)
Bentuk Vektor y_train: (900,)
```

17) **Train and Test**

```
theta = NormalEquation(X_train, y_train)
y_pred = Regression(X_test, theta)
```

18) **Print Regression Equation**

```
print(f"Y = {theta[0]:.2f} + {theta[1]:.2f}*X1 + {theta[2]:.2f}*X2 + {theta[3]:.2f}*X3 + {theta[4]:.2f
    }*X4 + {theta[5]:.2f}*X5")
```

Output:

```
Y = -3533645023.45 + 21522032.48*X1 + 12081321.81*X2 + 268049808.06*X3 + 298087646.55*X4
    + 1878969867.69*X5
```

19) **Define Metric Functions**

```
def MSE(y_true, y_pred):
    return np.mean((y_true - y_pred) ** 2)
def RMSE(y_true, y_pred):
    return np.sqrt(MSE(y_true, y_pred))
def r2_score(y_true, y_pred):
    ss_total = np.sum((y_true - np.mean(y_true)) ** 2)
    ss_residual = np.sum((y_true - y_pred) ** 2)
    return 1 - (ss_residual / ss_total)
```

20) **Calculate Accuracy/Error**

```
mse = MSE(y_test, y_pred)
rmse = RMSE(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
mape = mean_absolute_percentage_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print("Root Mean Squared Error:", rmse)
print(f"R-squared: {(r2* 100):.2f}%" )
print(f"MAPE: {(mape*100):.2f}%", )
```

Output:

```
Mean Squared Error: 1.577282503e+19
Root Mean Squared Error: 3971501609.01
R-squared: 83.66%
MAPE: 27.08%
```