

**LAPORAN**  
**TUGAS REGRESI MACHINE LEARNING**

**DATASET: USED HANDHELD DEVICE DATA**



**Oleh:**

<b>Zahra Nabila</b>	<b>NIM. 2310817110013</b>
<b>Andra Braputa Akbar Saleh</b>	<b>NIM. 2310817210001</b>
<b>Alysa Armelia</b>	<b>NIM. 2310817120009</b>

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS LAMBUNG MANGKURAT**  
**BANJARMASIN**  
**FEBRUARI, 2025**

## **1. Pendahuluan**

Perangkat genggam seperti ponsel dan tablet telah menjadi kebutuhan penting dalam berbagai aktivitas di era digital. Perkembangan teknologi yang cepat membuat produsen terus merilis model baru setiap tahun sehingga siklus hidup produk menjadi lebih singkat (Özen et al., 2026). Kondisi ini mendorong pertumbuhan pasar perangkat bekas yang terus meningkat secara global, termasuk di Indonesia, karena banyak masyarakat memilih perangkat bekas untuk mendapatkan spesifikasi tinggi dengan harga lebih terjangkau (Halim et al., n.d.). Perubahan perilaku konsumen yang semakin mempertimbangkan nilai ekonomi dan fungsi teknologi juga ikut mendorong perkembangan perdagangan elektronik (Fuentes & Hedegård, 2025). Namun, penjual dan pembeli perangkat bekas masih sering mengalami kesulitan dalam menentukan harga yang objektif (Hazwan et al., 2024). Harga perangkat tidak hanya dipengaruhi oleh merek, tetapi juga oleh berbagai spesifikasi teknis seperti RAM, kamera, baterai, dan tahun rilis perangkat (Naaman et al., 2025). Ketidakteraturan standar penentuan harga sering menimbulkan ketidakpastian dan berpotensi merugikan salah satu pihak, karena penjual biasanya menentukan harga berdasarkan persepsi subjektif tanpa acuan data yang jelas (Hasnain et al., n.d.). Untuk mengatasi hal tersebut, diperlukan pendekatan yang lebih sistematis dalam memperkirakan harga perangkat (Wang, 2023). Sistem komputasi yang menganalisis berbagai fitur teknis dapat membantu menghasilkan estimasi harga yang lebih objektif dan transparan (Kumar et al., 2023). Dengan memanfaatkan metode berbasis data, hubungan antara spesifikasi teknis dan harga pasar dapat dianalisis lebih akurat sehingga sistem prediksi berbasis regresi diharapkan dapat membantu menciptakan pasar perangkat bekas yang lebih adil dan terpercaya (Kumar et al., 2023).

## **2. Penjelasan Dataset**

Dataset yang digunakan dalam proyek ini adalah "Used Handheld Device Data" yang diambil dari kaggle. Dataset ini terdiri dari 3.454 baris data yang mencakup spesifikasi teknis perangkat dari berbagai merek. Variabel target dalam penelitian ini adalah `normalized_used_price`, sementara variabel prediktor meliputi lebih dari 15 fitur seperti berat perangkat, ukuran layar, resolusi kamera, kapasitas memori internal, RAM, dan kapasitas baterai.

## **3. Alur Preprocessing (Pipeline)**

Untuk memastikan data siap diproses oleh model Regresi Linier, kami melakukan tahapan berikut:

### **1. Data Cleaning**

Tahap ini mencakup identifikasi dan penanganan data yang hilang (missing values). Pemindaian awal dilakukan menggunakan fungsi `isnull().sum()` untuk memetakan kekosongan data. Selanjutnya, kekosongan pada seluruh kolom numerik ditangani menggunakan metode Median Imputation. Pendekatan ini dipilih karena nilai median lebih tangguh terhadap keberadaan nilai ekstrem atau pencilan (outlier) pada data spesifikasi perangkat.

## 2. Analisis Eksploratif dan Transformasi Kategorikal (*Encoding*)

Tahap Exploratory Data Analysis (EDA) menghasilkan visualisasi distribusi harga dan matriks korelasi untuk memahami hubungan antarvariabel. Fungsi `get_dummies` mengonversi variabel kategorikal biner menjadi format dummy variables numerik. Selain itu, juga mengubah seluruh kolom tipe boolean menjadi format integer untuk memfasilitasi kebutuhan komputasi algoritma regresi.

## 3. Partisi Data dan Pencegahan *Data Leakage*

Pemisahan dataset dilakukan menggunakan rasio 80% data latih dan 20% data uji dengan parameter `random_state=42`. Tahapan split data dilakukan sebelum transformasi fitur lanjutan untuk meminimalkan risiko kebocoran data (data leakage) yang dapat menyebabkan bias pada akurasi model.

## 4. Target Encoding

Skema Target Encoding menghitung nilai rata-rata variabel target hanya berdasarkan kelompok merek (device brand) tersedia di dalam data training. Statistik yang dihasilkan dipetakan ke kolom merek baik pada data latih maupun data uji. Apabila terdapat merek baru pada data uji, sistem akan mengisi kekosongan tersebut dengan nilai rata-rata global (global mean) dari data training..

## 5. Standardisasi Fitur (Feature Scaling)

StandardScaler digunakan untuk menormalisasi fitur numerik agar memiliki rata-rata 0 dan standar deviasi 1 sehingga bobot antar fitur menjadi lebih seimbang. Proses ini dilakukan pada data training untuk mempelajari parameter distribusi, kemudian transformasi yang sama diterapkan pada data training dan testing. Standardisasi ini membantu mencegah fitur dengan skala besar, seperti kapasitas baterai, mendominasi fitur dengan skala lebih kecil seperti jumlah RAM.

## 4. Prompt Log

Pada penelitian ini, kode awal diadaptasi dari notebook Kaggle Predicting Used Phones | LASSO, LinearRegression oleh David Paul D. Pipeline asli masih bersifat baseline dan belum memenuhi standar eksperimen machine learning yang sistematis. Oleh karena itu, dilakukan pengembangan ulang menggunakan bantuan AI assistant melalui proses prompt engineering dan iterasi kode. Bagian ini mendokumentasikan seluruh perubahan yang dilakukan beserta prompt yang digunakan.

### 4.1. Perbaikan Pipeline Preprocessing

*Pipeline preprocessing* direstrukturisasi agar mengikuti praktik *machine learning* yang benar, terutama untuk mencegah terjadinya data leakage. Perubahan utama yang dilakukan meliputi tiga aspek berikut.

- a. Split data dilakukan sebelum proses target encoding.
- b. Scaling hanya diterapkan pada data training, bukan keseluruhan dataset.
- c. Penanganan missing value dibuat secara eksplisit dan terstruktur.

#### 4.1.1. Prompt yang Digunakan

"Perbaiki preprocessing agar tidak terjadi data leakage pada encoding dan scaling."

#### **4.1.2. Hasil**

Pipeline yang dihasilkan mengikuti urutan berikut:

*Data --> Split --> Target Encoding --> Scaling --> Training Model*

#### **4.2. Implementasi Target Encoding yang Aman**

##### **4.2.1. Permasalahan pada Kode Awal**

Kode awal melakukan encoding langsung pada seluruh dataset sebelum proses split. Hal ini menyebabkan informasi dari kolom target bocor ke dalam data test, sehingga hasil evaluasi menjadi tidak valid secara metodologis.

##### **4.2.2. Prompt yang Digunakan**

"Bagaimana cara melakukan encoding best Practice?"

##### **4.2.3. Solusi yang Diterapkan**

Target encoding dipindahkan ke tahap setelah proses split data, sehingga hanya statistik dari data training yang digunakan untuk melakukan encoding.

##### **4.2.4. Implementasi**

```
train_temp = X_train.copy()
train_temp["target"] = y_train
brand_mean = train_temp.groupby("device_brand") ["target"].mean()
```

#### **4.3. Standarisasi Evaluasi Model**

Evaluasi model sebelumnya dihitung secara manual dan tidak efisien.

##### **4.3.1. Solusi yang Diterapkan**

Dibuat sebuah fungsi evaluasi terpusat yang dapat digunakan secara seragam untuk semua model.

##### **4.3.2. Prompt yang Digunakan**

"Buatkan fungsi evaluasi model regresi dengan semua metric sekaligus."

##### **4.3.3. Implementasi**

```
def evaluate_model(y_true, y_pred, model_name):
```

##### **4.3.4. Metrik yang Digunakan**

- a. R2 Score
- b. Mean Absolute Error (MAE)
- c. Root Mean Squared Error (RMSE)
- d. Mean Absolute Percentage Error (MAPE)

#### **4.4. Transformasi Target (Log Transformation)**

Distribusi variabel harga bersifat skewed, sehingga model regresi linear mengalami kesulitan dalam mempelajari pola data. Transformasi logaritmik diterapkan untuk mengatasi permasalahan ini

##### **4.4.1. Prompt yang Digunakan**

"Bagaimana cara meningkatkan performa regresi jika target skewed?"

##### **4.4.2. Implementasi**

```
y_train_log = np.log1p(y_train)
pred = np.expm1(pred_log)
```

## 4.5. Feature Engineering: Polynomial Features

Model linear baseline dan Lasso tidak mampu menangkap hubungan non-linear antar fitur. Oleh karena itu, ditambahkan beberapa model alternatif untuk memperluas kapabilitas prediksi.

### 4.5.1. Prompt yang Digunakan

"Tambahkan polynomial regression sebagai improvement dari linear regression."

### 4.5.2. Model yang Ditambahkan

- a. Polynomial Linear Regression
- b. Ridge Regression
- c. Polynomial Ridge Regression

## 4.6. K-Fold Cross Validation

Evaluasi model hanya bergantung pada satu train-test split tunggal, sehingga hasil evaluasi rentan terhadap bias akibat distribusi data pada split tertentu.

### 4.6.1. Prompt yang Digunakan

"Tambahkan K-fold validation untuk evaluasi regresi multi-metric."

### 4.6.2. Implementasi

```
def kfold_evaluate(model, X, y, model_name):  
    kf = KFold(n_splits=5, shuffle=True, random_state=42)  
    scoring = {  
        "r2": "r2",  
        "mae": "neg_mean_absolute_error",  
        "rmse": "neg_root_mean_squared_error",  
        "mape": "neg_mean_absolute_percentage_error"  
    }  
    scores = cross_validate(  
        model,  
        X,  
        y,  
        cv=kf,  
        scoring=scoring,  
        return_train_score=False  
    )  
    print(f"\n===== K-FOLD RESULT : {model_name} =====")  
    print(f"Mean R2 : {scores['test_r2'].mean():.6f}")  
    print(f"Mean MAE : {-scores['test_mae'].mean():.6f}")  
    print(f"Mean RMSE : {-scores['test_rmse'].mean():.6f}")  
    print(f"Mean MAPE : {-scores['test_mape'].mean():.6f}")
```

### 4.6.3. Prompt yang Digunakan

"Buatkan fungsi evaluasi model regresi dengan semua metric sekaligus."

### 4.6.4. Implementasi

```
def evaluate_model(y_true, y_pred, model_name):  
    r2 = r2_score(y_true, y_pred)  
    mae = mean_absolute_error(y_true, y_pred)  
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))  
    mape = mean_absolute_percentage_error(y_true, y_pred)  
    print(f"\n===== {model_name} =====")  
    print(f"R2 : {r2:.6f}")  
    print(f"MAE : {mae:.6f}")
```

```

print(f"RMSE : {rmse:.6f}")
print(f"MAPE : {mape:.6f}")

```

#### 4.6.5. Metrik yang Digunakan

- a. R2 Score
- b. Mean Absolute Error (MAE)
- c. Root Mean Squared Error (RMSE)
- d. Mean Absolute Percentage Error (MAPE)

#### 4.7. Stress Testing Model

Bagian ini merupakan penambahan baru yang tidak terdapat pada kode asli dari Kaggle. Pengujian robustness model dilakukan melalui tiga skenario stres sebagai berikut

##### 4.7.1. Prompt yang Digunakan

"Tambahkan 3 Stress test ini

1. Outlier Stress Test: menguji performa model ketika terdapat nilai-nilai ekstrem pada data input.
2. Noise Injection: menambahkan noise acak ke dalam fitur untuk mensimulasikan ketidakakuratan data.
3. Distribution Shift Simulation: mensimulasikan perubahan distribusi data yang mungkin terjadi pada data dunia nyata."

##### 4.7.2. Implementasi

```

print("\n--- STRESS TEST 1 : EXTREME DISTRIBUTION SHIFT ---")
import numpy as np
import pandas as pd

# Ambil mean & std dari training (SEBELUM scaling)
mean_vals = X_train.mean()
std_vals = X_train.std()
# Buat skenario ekstrem
extreme_high = mean_vals + 3 * std_vals
extreme_low = mean_vals - 3 * std_vals
# Ubah jadi DataFrame agar feature names tetap ada
extreme_high_df = pd.DataFrame([extreme_high], columns=X_train.columns)
extreme_low_df = pd.DataFrame([extreme_low], columns=X_train.columns)
# Scaling (tetap pakai DataFrame)
extreme_high_scaled = scaler.transform(extreme_high_df)
extreme_low_scaled = scaler.transform(extreme_low_df)
# Ubah kembali jadi DataFrame setelah scaling
extreme_high_scaled_df = pd.DataFrame(extreme_high_scaled,
columns=X_train.columns)
extreme_low_scaled_df = pd.DataFrame(extreme_low_scaled,
columns=X_train.columns)
# Polynomial transform
extreme_high_poly = poly.transform(extreme_high_scaled_df)
extreme_low_poly = poly.transform(extreme_low_scaled_df)
# Prediksi
pred_high = poly_model.predict(extreme_high_poly)
pred_low = poly_model.predict(extreme_low_poly)
print("Prediksi Harga Spek Ekstrem Tinggi :", float(pred_high[0]))
print("Prediksi Harga Spek Ekstrem Rendah :", float(pred_low[0]))

```

## 5. Experiment Log

Log Experiment berguna untuk mengidentifikasi kombinasi optimal antara encoding, imputasi, dan arsitektur model yang menghasilkan  $R^2$  tertinggi dan metrik error terendah pada data uji maupun di bawah validasi K-Fold cross-validation.

### 5.1. Eksperimen Strategi Encoding (Train-Test Split)

Tabel berikut merangkum performa model di seluruh metode encoding yang dievaluasi pada data uji. Nilai  $R^2$  yang disorot hijau menunjukkan skor  $\geq 0.85$ .

Metode Encoding	Model	$R^2$	MAE	RMSE	MAPE
Target Encoding (Sebelum Split)	Baseline Linear Regression	0.836476	0.184911	0.230362	0.043576
	Baseline Decision Tree	0.703730	0.238735	0.310073	0.055480
	Ridge Regression (Improved)	0.818892	0.192447	0.242431	0.045266
	Lasso Regression (Improved)	0.818821	0.193580	0.242478	0.045615
Target Encoding (Sesudah Split)	Baseline Linear Regression	0.836402	0.184987	0.230414	0.043594
	Baseline Decision Tree	0.698919	0.240462	0.312580	0.055929
	Ridge Regression (Improved)	0.818836	0.192524	0.242468	0.045284
	Lasso Regression (Improved)	0.818778	0.193640	0.242508	0.045629
One Hot Encoding	Polynomial Linear	0.852135	0.174570	0.219055	0.040810
	Regression				
	Baseline Linear Regression	0.835729	0.184380	0.230888	0.043438
	Baseline Decision Tree	0.704020	0.238549	0.309921	0.055732
Label Encoding	Ridge Regression (Improved)	0.818510	0.192660	0.242687	0.045323
	Lasso Regression (Improved)	0.819229	0.193283	0.242205	0.045529
	Baseline Linear Regression	0.836522	0.184795	0.230329	0.043533
	Baseline Decision Tree	0.715119	0.232854	0.304054	0.054451
	Ridge Regression (Improved)	0.818502	0.192737	0.242692	0.045307
	Lasso Regression (Improved)	0.818532	0.193749	0.242672	0.045639
	Polynomial Linear	0.851843	0.174704	0.219271	0.040861
	Regression				

Polynomial Linear Regression secara konsisten menonjol sebagai model dengan performa terbaik yang mencapai skor  $R^2$  tertinggi di angka 0,852, sementara Baseline Linear Regression menunjukkan ketangguhan dengan mengungguli model regularisasi seperti Ridge dan Lasso melalui margin selisih yang tipis (0,836 vs 0,819). Sebaliknya, model Decision Tree memberikan performa yang jauh lebih rendah dengan rentang  $R^2$  antara 0,69 hingga 0,72, sebuah indikasi terjadinya gejala overfitting di mana model gagal menggeneralisasi data baru secara optimal.

### 5.2. Hasil K-Fold Cross-Validation

K-Fold cross-validation (5 fold) diterapkan untuk menilai kemampuan generalisasi model. Eksperimen membandingkan strategi imputasi (KNN vs Mean) dengan label encoding. Nilai Mean  $R^2$  yang disorot hijau menunjukkan skor  $\geq 0.84$ .

Eksperimen	Model	Mean $R^2$	Mean MAE	Mean RMSE	Mean MAPE
K-Fold: Label Encoding	Linear Regression	0.839825	0.183610	0.237087	0.044372
	Ridge Regression	0.821924	0.036392	0.049374	0.022484
	Lasso Regression	0.821249	0.036546	0.049479	0.022615
	Polynomial Linear	0.846154	0.179836	0.232155	0.043007
Regression					

<b>K-Fold: Imputasi</b>	Linear Regression	0.840414	0.183352	0.236638	0.044305
<b>KNN</b>	Polynomial Linear Regression	0.846139	0.179897	0.232174	0.043048
<b>K-Fold: Imputasi</b>	Linear Regression	0.840088	0.183548	0.236885	0.044354
<b>Mean</b>	Polynomial Linear Regression	0.845728	0.179989	0.232490	0.043079

Berdasarkan hasil pengujian, di mana skor Mean R<sup>2</sup> pada model Linear Regression meningkat dari 0.840088 menjadi 0.840414. Pada model Polynomial Linear Regression, dengan hasil yang hampir identik selisih metrik yakni kurang dari 0.001 di seluruh aspek evaluasi. Mean Imputation menjadi pilihan yang lebih disarankan karena efisiensi komputasi dan kesederhanaan model tanpa harus mengorbankan akurasi prediksi secara nyata.

### 5.3. Eksperimen Alpha Pencarian Hyperparameter (GridSearch)

Eksperimen ini bertujuan mencari nilai alpha optimal untuk Ridge dan Lasso Regression menggunakan GridSearchCV dengan metrik RMSE. Pencarian dilakukan pada rentang nilai alpha yang berbeda untuk masing-masing model.

#### 5.3.1. Konfigurasi GridSearch

Ridge: rentang alpha = [0.001, 0.01, 0.1, 1, 10, 50, 100]

Lasso: rentang alpha = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1, 1]

Metrik optimasi: RMSE (Root Mean Squared Error)

Strategi validasi: K-Fold Cross Validation

#### 5.3.2. Hasil GridSearch

Tabel berikut menunjukkan alpha terbaik yang ditemukan dan performa model pada data uji:

Model	Alpha Terbaik	R <sup>2</sup>	MAE	RMSE	MAPE
Ridge (GridSearch)	0.001	0.818818	0.192531	0.242481	0.045285
Lasso (GridSearch)	0.0001	0.819006	0.192571	0.242355	0.045305

#### 5.3.3. Observasi

Hasil eksperimen GridSearch menunjukkan bahwa nilai alpha terbaik untuk Ridge adalah 0.001, yang menandakan bahwa dataset ini tidak membutuhkan regularisasi yang kuat. Hal yang sama juga terjadi pada Lasso, di mana alpha terbaik adalah 0.0001, sehingga menunjukkan bahwa sebagian besar fitur dalam dataset sudah relevan. Dari sisi performa, Ridge GridSearch menghasilkan R<sup>2</sup> sebesar 0.8188 dan Lasso sebesar 0.8190, yang hampir sama dengan hasil sebelumnya. Hal ini menunjukkan bahwa proses pencarian hyperparameter dengan GridSearch tidak memberikan peningkatan yang signifikan, karena fitur dalam dataset sudah cukup informatif, tidak memiliki korelasi tinggi, dan tidak banyak mengandung noise.

## 6. Perbandingan Baseline vs Terbaru

Membandingkan performa model awal (*Baseline*) dengan model hasil pengembangan terbaru (*Improved Model*) bertujuan untuk memvalidasi sejauh mana teknik *feature engineering* dan perbaikan *pipeline* dapat memberikan dampak terhadap akurasi prediksi harga perangkat genggam (*handheld devices*) seperti smartphone dan tablet bekas.

## 6.1. Definisi Model

Penelitian ini membandingkan performa model baseline Linear Regression dan Lasso Regression ( $\alpha=0.1$ ) dengan model terbaru Polynomial Linear Regression dalam memprediksi harga perangkat bekas. Model baseline menggunakan konfigurasi standar yang berisiko mengalami data leakage. Sebagai perbaikan, model terbaru menerapkan Polynomial Features dan transformasi logaritmik (np.log1p) untuk menangkap hubungan non-linear antar spesifikasi. Selain itu, proses target encoding dan scaling dilakukan setelah pemisahan data (post-split) untuk menjaga integritas serta validitas evaluasi model.

## 6.2. Detail Konfigurasi Skenario

Berikut konfigurasi detail dari skenario yang menghasilkan performa terbaik di antara seluruh eksperimen yang telah didokumentasikan pada (Experiment Log):

Komponen	Konfigurasi
Encoding	Target Encoding sesudah Split (lebel encoding pada data training)
Imputasi	Mean Imputation
Scaler	StandardScaler
Target	Normalized_used_price (log-transformed)
Validasi	Train-Test Split (80:20)

## 6.3. Perbandingan Performa

Setelah menggunakan konfigurasi di tabel sebelumnya, diperoleh perbandingan metrik evaluasi pada data uji sebagai berikut:

Metrik Evaluasi	Linear Regression	(Polynomial LR)	Selisih Performa
R <sup>2</sup> Score	0.836402	0.852135	+1.57% (Peningkatan Akurasi)
MAE	0.184987	0.174570	5.63% (Lebih Akurat)
RMSE	0.230414	0.219055	-4.93% (Error Menurun)
MAPE	0.043594	0.040810	-6.38% (Error < 5%)

## 6.4. Analisis Keunggulan Model Terbaru

Model terbaru Polynomial Linear Regression menunjukkan performa terbaik dengan skor R<sup>2</sup> sebesar 0,852, yang menunjukkan bahwa hubungan antara spesifikasi seperti RAM dan baterai dengan harga memiliki pola non-linear yang tidak dapat ditangkap dengan baik oleh model linear standar. Sebaliknya, model baseline Lasso ( $\alpha=0,1$ ) menghasilkan skor terendah ( $R^2 = 0,777827$ ) karena regularisasi yang terlalu besar menyebabkan underfitting dan hilangnya informasi fitur penting. Masalah ini diperbaiki melalui hyperparameter tuning dengan menurunkan nilai  $\alpha$  menjadi 0,001. Model terbaru juga memiliki metodologi yang lebih baik karena menggunakan pipeline yang telah diperbaiki untuk mencegah data leakage, dengan melakukan data split sebelum target encoding dan scaling sehingga evaluasi lebih mencerminkan kemampuan generalisasi. Selain itu, berdasarkan

stress test, model terbaru menunjukkan stabilitas yang lebih baik dalam menghadapi data ekstrem atau noise, sehingga sistem menjadi lebih objektif dalam menentukan harga perangkat bekas.

## 7. Stress Test

### 7.1. Extreme Distribution Shift Test

Pengujian ini dilakukan untuk melihat apakah model masih dapat memberikan prediksi yang logis ketika menghadapi perangkat dengan spesifikasi yang sangat tinggi “Sultan” dan sangat rendah “Rongsok”. Hasilnya menunjukkan bahwa perangkat dengan spesifikasi ekstrem tinggi diprediksi memiliki harga sebesar 6.0079, sedangkan perangkat dengan spesifikasi ekstrem rendah diprediksi sebesar 1.0967. Hal ini menunjukkan bahwa model tetap menjaga hubungan yang konsisten antara spesifikasi dan harga, di mana semakin tinggi spesifikasi maka harga yang diprediksi juga semakin tinggi, meskipun data tersebut berada di luar distribusi utama data pelatihan.

### 7.2. Noise Injection Test

Pengujian ini bertujuan untuk menguji kestabilan model dengan menambahkan noise atau gangguan acak pada fitur teknis. Setelah dilakukan pengujian, model Polynomial Regression tetap menunjukkan performa yang baik dengan nilai  $R^2$  sebesar 0.852291. Nilai tersebut menunjukkan bahwa model masih mampu menjelaskan sebagian besar variasi data meskipun terdapat gangguan pada input, sehingga dapat disimpulkan bahwa model cukup robust dan tidak terlalu sensitif terhadap perubahan kecil pada data.

### 7.3. Outlier Sensitivity Test

Pengujian ini dilakukan untuk mengidentifikasi dan menghapus data yang memiliki nilai ekstrem atau outlier agar tidak memengaruhi performa model. Dengan menggunakan metode Interquartile Range (IQR), diperoleh batas bawah sebesar 2.9443 dan batas atas sebesar 5.8377. Berdasarkan batas tersebut, ditemukan 72 data yang termasuk outlier dan kemudian dihapus, sehingga jumlah dataset berkurang dari 2763 menjadi 2691 data. Proses ini membantu memastikan bahwa model dilatih dengan data yang lebih representatif dan tidak terdistorsi oleh nilai ekstrem.

### 7.4. Top 5 Error Analysis

Analisis ini dilakukan dengan mengevaluasi lima data yang memiliki selisih terbesar antara harga aktual dan harga prediksi. Kesalahan terbesar terdapat pada data dengan indeks 626, dengan absolute error sebesar 0.9093, di mana harga aktual sebesar 5.6985 diprediksi hanya sebesar 4.7892. Hal ini menunjukkan bahwa model masih memiliki keterbatasan dalam memprediksi harga yang sangat tinggi, kemungkinan karena adanya faktor lain di luar spesifikasi teknis seperti kelangkaan atau nilai koleksi yang tidak tercakup dalam dataset.

## 8. Link Youtube dan Github

Link Youtube : [https://youtu.be/S1soeb6ZFs0?si=OnRhXiTVwa7\\_gXUb](https://youtu.be/S1soeb6ZFs0?si=OnRhXiTVwa7_gXUb)

Link Github : <https://github.com/Andra-Braputra/Used-handheld-device-regression-analysis>

## 9. Referensi

- Fuentes, C., & Hedegård, L. (2025). Recirculating products: valuation practices and the socio-material making of second-hand products. *European Journal of Marketing*, 59(13), 391–418. <https://doi.org/10.1108/EJM-12-2022-0849>
- Halim, S., Gan, S.-S., & Oentoro, J. M. (n.d.). Identifying factors that influence customers' interest in buying refurbished smartphones: An Indonesian context. In *Asia-Pacific Journal of Science and Technology*. Retrieved <https://www.tci-thaijo.org/index.php/APST/index>
- Hasnain, M., Sajid, A., Awan, M. A., & Ayeb, A. (n.d.). Predicting The Price Of Used Electronic Devices Using Machine Learning Techniques. In *International Journal of Computing and Related Technologies* (Vol. 4, Number 1). Retrieved [www.whatmobile.com](http://www.whatmobile.com)
- Hazwan, M., Hazhar, M., Dzaihan, D., Dzailani, D., Hairulanuar, N. S., Anizan, N., & Yusoff, S. (2024). Determinants of Smartphone Prices using Backward Elimination Technique in Multiple Linear Regression. *Mathematical Sciences and Informatics Journal*, 5(1), 72–82. <https://doi.org/10.24191/mij.v5i1.1358>
- Kumar, M., Pilania, U., & Varshney, C. (2023). Predicting Mobile Phone Prices with Machine Learning. *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, 181–185. <https://doi.org/10.1109/AECE59614.2023.10428614>
- Naaman, D., Tahir Ahmed, B., & Sediq Kareem, O. (2025). Comparative Review of Machine Learning Models for Mobile Price Prediction Based on Specifications: A Systematic Literature Analysis. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 16(2), 96–121. <https://doi.org/10.31849/digitalzone.v16i2.27096>
- Özen, B. B., Alaeddinoğlu, M. F., & Aydın, T. (2026). Analysing the Refurbished Smart Phone Market with Machine Learning. *Trends in Business and Economics*, 40(1), 42–59. <https://doi.org/10.16951/trendbusecon.1607949>
- Wang, J. (2023). Mobile Phone Price Range Prediction Based on Machine Learning Algorithms. *Advances in Economics, Management and Political Sciences*, 46(1), 260–268. <https://doi.org/10.54254/2754-1169/46/20230348>