

Pengembangan Sistem Prediksi Harga dan Rekomendasi Mobil Bekas Berbasis Machine Learning

**Jerry Anggara^{1*}, Fauzi Rahmat Ramadhan², Achmad Fikri Syofian³,
Apriansa Arwandi Panjaitan⁴, Raka Fazah Fithra⁵**

^{1,2,3,4,5}Informatika, Ilmu Komputer, Universitas Singaperbangsa Karawang, Jawa Barat, Indonesia

e-mail: 2110631170072@student.unsika.ac.id^{1*}, 2110631170066@student.unsika.ac.id²,

2110631170043@student.unsika.ac.id³, 2110631170053@student.unsika.ac.id⁴,

2110631170094@student.unsika.ac.id⁵

Informasi Artikel

Article History:

Received : 31 Januari 2025

Revised : 20 Maret 2025

Accepted : 21 April 2025

Published : 25 April 2025

***Korespondensi:**

2110631170072@student.unsika.ac.id

Keywords:

Used Car Price Prediction, Machine Learning, Web-Based Application, Rule-Based Recommendation System.

Hak Cipta ©2025 pada Penulis.

Dipublikasikan oleh Universitas

Dinamika



Artikel ini *open access* di bawah lisensi [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/).

 10.37802/joti.v7i1.987

Journal of Technology and Informatics (JoTI)

P-ISSN 2721-4842

E-ISSN 2686-6102

[https://e-](https://e-journals.dinamika.ac.id/index.php/joti)

[journals.dinamika.ac.id/index.php/joti](https://e-journals.dinamika.ac.id/index.php/joti)

Abstract:

The used car sales business in Indonesia has been experiencing rapid growth, driven by increasing market demand. However, determining the price of used cars remains a challenge due to various influencing factors such as the year of production, mileage, and vehicle specifications. This research develops a web-based used car price prediction system using the XGBoost Regressor algorithm. The data used undergoes preprocessing and hyperparameter tuning to produce a high-performance model (R^2 : 97.79% on training data and 89.90% on testing data, MSE: 2.3129, RMSE: 1.5208). Additionally, the system provides a car recommendation feature using a Rule-Based Method, allowing users to filter vehicles based on specific criteria. The results demonstrate that this system effectively assists both buyers and sellers in making more informed, efficient, and transparent decisions in used car transactions.

PENDAHULUAN

Era digital yang berkembang pesat, data berperan penting dalam pengambilan keputusan di berbagai industri [1], termasuk industri otomotif. Di Indonesia, bisnis mobil bekas mengalami pertumbuhan yang signifikan, tetapi proses penentuan harga jualnya masih menjadi tantangan utama. Harga jual dipengaruhi oleh berbagai faktor, seperti usia kendaraan, jarak tempuh, spesifikasi mesin, merek, model, dan fitur tambahan. Namun, metode

konvensional yang sering digunakan oleh penjual dan pembeli, seperti perbandingan harga manual dan perkiraan subjektif, cenderung tidak akurat dan tidak konsisten.

Ketidakpastian dalam menentukan harga yang kompetitif dapat menimbulkan dua permasalahan utama. Pertama, penjual berisiko menetapkan harga yang terlalu tinggi, sehingga memperlambat proses penjualan akibat rendahnya minat pembeli. Kedua, pembeli berpotensi membayar harga yang lebih tinggi dari nilai pasar sebenarnya, yang dapat menyebabkan ketidakseimbangan dalam transaksi. Selain itu, fluktuasi harga yang dipengaruhi oleh permintaan dan tren pasar semakin menambah kompleksitas dalam proses penetapan harga. Oleh karena itu, diperlukan sistem prediksi harga mobil bekas yang tidak hanya akurat tetapi juga adaptif terhadap pola data yang kompleks [2].

Salah satu pendekatan yang terbukti efektif dalam prediksi harga berbasis data adalah penggunaan algoritma *machine learning*. XGBoost Regressor (XGBRegressor) merupakan algoritma yang unggul dalam menangani dataset besar dan kompleks dengan teknik *boosting*, yang mampu menggabungkan model prediktif lemah menjadi model yang lebih kuat. Dengan menerapkan XGBRegressor, sistem prediksi harga mobil dapat lebih andal, mengurangi subjektivitas dalam penentuan harga, serta memberikan rekomendasi harga yang lebih sesuai dengan kondisi pasar [2].

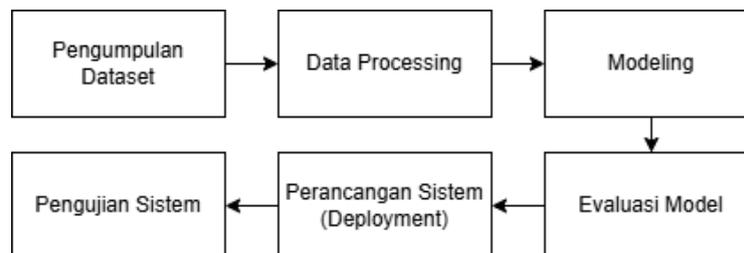
Namun, kebanyakan aplikasi yang ada hanya berfokus pada prediksi harga, tanpa memberikan dukungan tambahan bagi pengguna untuk mengambil keputusan pembelian. Oleh karena itu, proyek ini tidak hanya mengembangkan sistem prediksi harga mobil bekas menggunakan XGBRegressor, tetapi juga melengkapinya dengan fitur rekomendasi berbasis metode *rule-based*. Sistem ini dirancang untuk mengintegrasikan prediksi harga dengan preferensi pengguna, seperti merek, jenis mobil, usia kendaraan, dan spesifikasi mesin, sehingga dapat memberikan rekomendasi yang personal dan relevan. Dengan demikian, sistem ini diharapkan mampu membantu pengguna, baik penjual maupun pembeli, dalam membuat keputusan yang lebih tepat dan efisien di pasar mobil bekas.

Penelitian terdahulu yang menggunakan algoritma *Random Forest* menunjukkan performa yang cukup baik dalam memprediksi harga mobil bekas dengan tingkat akurasi R^2 sebesar 89% [3]. Namun, model ini masih memiliki keterbatasan dalam menangani variasi harga mobil yang lebih kompleks. Selain itu, penelitian yang menerapkan algoritma Regresi Linear menunjukkan akurasi yang lebih rendah, dengan nilai R^2 sebesar 76.22% dan RMSE sebesar 3813.65. Hal ini mengindikasikan bahwa model tersebut kurang efektif dalam menangkap pola data yang lebih kompleks dalam prediksi harga mobil bekas [4]. Selain aspek prediksi harga, beberapa penelitian sebelumnya hanya berfokus pada pengembangan model tanpa mempertimbangkan fitur rekomendasi kendaraan berbasis preferensi pengguna. Pada penelitian [2] sistem *machine learning* berbasis *website* yang dikembangkan hanya untuk prediksi harga tanpa fitur tambahan yang dapat membantu pengguna dalam memilih kendaraan berdasarkan kebutuhan spesifik mereka.

Oleh karena itu, penelitian ini tidak hanya mengembangkan model prediksi berbasis XGBoost yang unggul dalam hal akurasi, tetapi juga melengkapinya dengan sistem rekomendasi berbasis aturan (*rule-based method*). Sistem ini memungkinkan pengguna untuk memperoleh rekomendasi kendaraan yang sesuai dengan preferensi mereka. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang lebih signifikan dalam pengembangan sistem prediksi harga dan rekomendasi mobil bekas yang lebih akurat, efisien, dan relevan.

METODE

Metode penelitian yang digunakan dalam studi ini merupakan penelitian kuantitatif dengan pendekatan eksperimen. Penelitian ini melibatkan pengembangan dan evaluasi sistem prediksi harga dan rekomendasi mobil bekas berbasis machine learning. Tahapan penelitian dimulai dengan pengumpulan dataset dari kaggle, di mana data tersebut dilakukan data *processing* dan transformasi data agar sesuai dengan keperluan algoritma. Selanjutnya, model prediksi harga dikembangkan menggunakan beberapa algoritma regresi seperti *XGBoost Regressor*, *Linear Regression*, *Lasso Regression*, dan *Ridge Regression*, yang kemudian dibandingkan berdasarkan metrik evaluasi seperti R^2 , MSE, dan RMSE. Model machine learning dilakukan *deployment* dengan membuat *website* menggunakan framework flask. Terdapat fitur tambahan pada *website* berupa rekomendasi mobil berbasis aturan (*rule-based method*) yang dirancang untuk mengintegrasikan faktor harga dengan preferensi pengguna, seperti jenis mobil, merek tertentu, usia kendaraan, dan spesifikasi mesin. Hal ini bertujuan untuk memberikan rekomendasi pembelian yang lebih personal dan sesuai dengan kebutuhan pengguna. Hasil akhir sistem dievaluasi menggunakan pengujian *Black Box* untuk memastikan setiap fungsi bekerja sesuai dengan spesifikasi dan menghasilkan *output* yang sesuai dengan input yang diberikan.



Gambar 1. Tahapan Penelitian

Pengumpulan Dataset

Dataset yang digunakan bersumber dari *kaggle*, dataset ini dipilih karena memiliki fitur yang lengkap dan relevan dengan permasalahan penelitian, seperti harga mobil, spesifikasi kendaraan, serta faktor-faktor yang berpengaruh terhadap nilai jual mobil bekas. Dataset yang digunakan terdiri dari 6.019 baris dan 10 kolom, yang mencakup berbagai atribut penting terkait karakteristik mobil bekas. Adapun fitur-fitur dataset sebagai berikut:

Tabel 1. Fitur-fitur Dataset

Nama Fitur	Penjelasan
Name	Merek dan tipe mobil
Location	Tempat penjual mobil
Year	Tahun atau edisi model
KM_Driven	Total kilometer yang telah ditempuh
Fuel_Type	Jenis bahan bakar
Transmission	Jenis transmisi yang digunakan
Owner_Type	Tipe kepemilikan (Pemilik pertama, pemilik kedua, dan seterusnya)
Mileage	Rata-rata konsumsi bahan bakar
Engine	Volume perpindahan mesin mobil (satuan cc)

Power	Daya maksimum yang dihasilkan
Seats	Jumlah kursi yang tersedia dalam mobil
Price	Harga mobil

Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
Hyundai Creta 1.6 CRDI SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
Hyundai EON LPG Era Plus Option	Hyderabad	2012	75000	LPG	Manual	First	21.1 km/kg	814 CC	55.2 bhp	5.0	NaN	2.35
Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	3.50
Toyota Innova Crysta 2.8 GX AT BS	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmpl	2755 CC	171.5 bhp	8.0	21 Lakh	17.50
Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN	5.20
Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5.0	NaN	1.95
Maruti Ciaz Zeta	Kochi	2018	25692	Petrol	Manual	First	21.56 kmpl	1462 CC	103.25 bhp	5.0	10.65 Lakh	9.95

Gambar 2. Sampel Dataset

Data Processing

Data processing yang dilakukan terbagi menjadi beberapa tahapan yaitu:

1. Data Cleaning

Data cleaning dilakukan untuk memperbaiki kesalahan atau kekurangan dalam dataset. Proses ini mencakup penanganan missing values dengan metode seperti mengisi nilai median, penghapusan data duplikat, penanganan *outliers* menggunakan metode Interquartile Range (IQR), serta standarisasi format data untuk menjaga konsistensi.

2. Data Transformation

Data transformasi dilakukan untuk mengubah format data agar sesuai pada proses modeling *machine learning* [5]. Terdapat beberapa metode yang digunakan yaitu:

Scaling merupakan proses melakukan standarisasi pada data bertipe numerik. Proses ini mengubah data sehingga memiliki distribusi dengan rata-rata (μ) 0 dan standar deviasi (σ) 1, sehingga menghilangkan bias skala antar fitur. Adapun formula scaling sebagai berikut:

$$z = \frac{x - \mu}{\sigma}$$

Keterangan

z = Hasil standarisasi

x = Nilai fitur

μ = Nilai rata-rata fitur

σ = Standar deviasi dari fitur

One Hot Encoding merupakan mengubah data kategorik menjadi representasi numerik, yang merupakan bagian dari transformasi untuk membuat data kompatibel dengan algoritma machine learning.

3. Feature Engineering

Feature engineering bertujuan memilih dan memanipulasi fitur yang relevan untuk meningkatkan kinerja model. Proses ini meliputi penghapusan fitur-fitur yang kurang relevan seperti *Location*, *Owner_Type* dan lain-lain serta melakukan visualisasi korelasi menggunakan heatmap untuk mengidentifikasi hubungan antar fitur, membantu mendeteksi fitur yang redundan atau kurang relevan.

4. Data *Splitting*

Proses ini membagi dataset menjadi data training, data testing dan data validasi untuk menghindari overfitting dan untuk mengevaluasi kinerja model secara realistis. Pada penelitian ini dataset dibagi menjadi 2 yaitu data training dan data testing dengan proporsi 80% untuk data training dan 20% untuk data testing.

Modeling

Pemodelan dalam penelitian ini dilakukan menggunakan beberapa algoritma regresi, yaitu Regresi Linear, Regresi Ridge, Regresi Lasso, dan XGBoost Regressor. Pemilihan algoritma ini didasarkan pada kemampuan masing-masing dalam menangani data yang memiliki karakteristik berbeda serta memberikan hasil prediksi yang akurat.

1. Regresi linear adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel independen (fitur) dan variabel dependen (target) dengan cara menemukan garis terbaik yang meminimalkan kesalahan.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

Keterangan:

\hat{y} = Variabel dependen

β_0 = Intersep (bias)

$\beta_1, \beta_2, \dots, \beta_p$ = koefisien regresi

x_1, x_2, \dots, x_p = Variabel independen

2. Regresi Ridge, Algoritma ini merupakan varian dari regresi linear yang menambahkan penalti berupa norma L2 ke dalam fungsi loss. Penalti ini membantu mengurangi kompleksitas model dengan mengurangi magnitudo koefisien, sehingga model lebih stabil terhadap variabel yang saling berkorelasi tinggi.

$$J(\beta) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Keterangan:

λ = Parameter regularisasi yang mengontrol kekuatan penalti

$\sum_{j=1}^p \beta_j^2$ = Penalti L2 (jumlah kuadrat koefisien)

3. Regresi Lasso, Algoritma regresi dengan menggunakan norma L1 sebagai penalti dalam fungsi loss, yang tidak hanya mengurangi magnitudo koefisien tetapi juga dapat membuat beberapa koefisien menjadi nol. Dengan demikian, Lasso membantu mengidentifikasi fitur yang paling relevan, sekaligus menyederhanakan model.

$$J(\beta) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Keterangan:

$|\beta_j|$ = Penalti L1 (jumlah nilai absolut koefisien)

4. XGBoost Regressor merupakan algoritma yang lebih canggih berbasis boosting. XGBoost menggabungkan beberapa model regresi lemah (weak learners) untuk membentuk model yang kuat dan stabil. Algoritma ini memiliki kemampuan untuk menangani hubungan non-linear dan fitur interaksi, serta dilengkapi dengan mekanisme regulasi untuk mencegah overfitting. Dengan kemampuan optimisasi yang baik dan fleksibilitasnya, XGBoost sering digunakan untuk memodelkan data dengan karakteristik yang kompleks.

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}$$

Keterangan:

K = Jumlah pohon keputusan

$f_k(x_i)$ = Prediksi dari pohon k untuk data x_i

\mathcal{F} = Ruang pohon keputusan

Pelatihan model dilakukan dengan menerapkan hyperparameter tuning, yaitu proses untuk mencari kombinasi nilai terbaik dari hyperparameter pada model machine learning guna meningkatkan kinerja prediksi [6]. Tidak seperti parameter yang dipelajari selama pelatihan, hyperparameter ditentukan sebelum pelatihan dimulai dan berperan dalam mengatur cara model belajar dari data. Berikut adalah hyperparameter yang digunakan untuk setiap algoritma:

```
def tune_and_evaluate(model, param_grid):

    # Hyperparameter tuning
    grid_search = GridSearchCV(model, param_grid=param_grid, cv=5, scoring='r2')
    grid_search.fit(X_train, y_train)
    model = grid_search.best_estimator_
    print("Best Parameters: ", grid_search.best_params_)
    print("Mean of cross validation scores: ", grid_search.best_score_)
    print('-----')
```

Gambar 3. Pembuatan Fungsi Tuning

```
#Model Building
lin_reg = make_pipeline(
    preprocessor,
    LinearRegression()
)
#hyperparameter tuning
param_grid = {'columntransformer__pipeline-1__polynomialfeatures__degree':[2,3,4]}
lin_reg = tune_and_evaluate(lin_reg, param_grid)
```

Gambar 4. Hyperparameter Tuning Linear Regresi

```
#Model Building
ridge = make_pipeline(
    preprocessor,
    Ridge()
)
#hyperparameter tuning
param_grid = {
    'ridge__alpha':[1e-8,1e-9,1e-10],
    'columntransformer__pipeline-1__polynomialfeatures__degree':[2,3,4]
}
ridge = tune_and_evaluate(ridge, param_grid)
```

Gambar 5. Hyperparameter Tuning Linear Ridge

```
#Model Building
lasso = make_pipeline(
    preprocessor,
    Lasso()
)
#Hyperparameter Tuning
param_grid={
    "lasso__alpha": [1e-6,1e-7,1e-8],
    "columntransformer__pipeline-1__polynomialfeatures__degree": [2,3,4]
}
lasso = tune_and_evaluate(lasso, param_grid)
```

Gambar 6. Hyperparameter Tuning Linear Lasso

```

# Full pipeline for numerical and categorical features
preprocessor = make_column_transformer(
    (num_pipeline, num_features),
    (cat_pipeline, cat_features)
)

# Model Building: XGBoost Regression
xgb_reg = make_pipeline(
    preprocessor,
    XGBRegressor(random_state=42)
)

# Hyperparameter tuning
param_grid = {
    'xgbregressor__n_estimators': [100, 200, 300],
    'xgbregressor__max_depth': [3, 5, 7],
    'xgbregressor__learning_rate': [0.01, 0.1, 0.3],
    'xgbregressor__subsample': [0.8, 0.9, 1.0],
    'xgbregressor__colsample_bytree': [0.8, 0.9, 1.0],
}

xgb_reg = tune_and_evaluate(xgb_reg, param_grid)

```

Gambar 7. Hyperparameter Tuning XGBoost Regression

Evaluasi Modeling

Terdapat beberapa matriks evaluasi model sebagai berikut:

1. R2 (Coefficient of Determination) adalah metrik yang mengukur proporsi variabilitas dalam data target (y) yang dapat dijelaskan oleh model [7].
2. Mean Squared Error (MSE) adalah matriks evaluasi yang mengukur rata-rata kuadrat kesalahan antara nilai aktual (y) dan nilai prediksi (\hat{y}) [7].
3. Root Mean Squared Error (RMSE) adalah akar kuadrat dari MSE. Ini memberikan interpretasi error dalam skala yang sama dengan target (y), sehingga lebih mudah dipahami dibandingkan MSE [7].

Algoritma Rule-Based

Algoritma Rule-Based Method adalah pendekatan dalam kecerdasan buatan dan pemodelan data yang digunakan untuk pengambilan keputusan atau prediksi berdasarkan aturan logis yang telah ditetapkan [8]. Metode ini banyak diterapkan dalam sistem pendukung keputusan dan sistem pakar karena kemampuannya mengotomatisasi proses pengambilan keputusan. Pada sistem rekomendasi kendaraan bekas, algoritma ini digunakan untuk menyaring data berdasarkan kriteria tertentu yang telah ditetapkan pengguna. Pendekatan ini memungkinkan sistem untuk memberikan rekomendasi kendaraan bekas yang sesuai dengan preferensi pengguna. Implementasi kode algoritma Rule-Based Method pada penelitian ini sebagai berikut:

```

if priority_feature == 'Year' and priority_value is not None:
    filtered_df = filtered_df[filtered_df['Year'] >= priority_value]
elif priority_feature == 'Kilometers_Driven' and priority_value is not None:
    filtered_df = filtered_df[filtered_df['Kilometers_Driven'] <= priority_value]
elif priority_feature == 'Engine' and priority_value is not None:
    filtered_df = filtered_df[filtered_df['Engine'] >= priority_value]

```

Keterangan:

- a. Priority_feature dan priority_value adalah variabel input yang digunakan untuk menentukan kriteria pemfilteran.
- b. Filtered_df adalah DataFrame yang berisi data mobil yang akan difilter berdasarkan kriteria yang diberikan.

- c. Kondisi if-else-then digunakan untuk memfilter data berdasarkan kriteria yang ditetapkan, misalnya tahun pembuatan, jarak tempuh, dan kapasitas mesin.

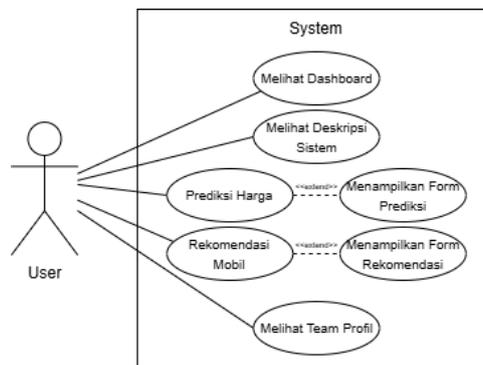
Perancangan Sistem (*Deployment*)

1. Analisis Kebutuhan Website

Analisis kebutuhan adalah proses untuk mengidentifikasi, memahami, dan mendokumentasikan kebutuhan sistem agar sesuai dengan tujuan pengguna atau pemangku kepentingan [9]. Proses ini bertujuan memastikan bahwa sistem yang akan dibangun dapat memenuhi kebutuhan dan harapan pengguna, baik dari sisi fungsionalitas maupun non-fungsionalitas. Dalam konteks pengembangan website, analisis kebutuhan mencakup pengumpulan informasi tentang fitur, interaksi pengguna, dan tujuan bisnis yang ingin dicapai oleh website tersebut.

Salah satu alat yang efektif untuk menganalisis kebutuhan adalah use case, yaitu metode yang menggambarkan interaksi antara pengguna (aktor) dengan sistem untuk mencapai tujuan tertentu. Use case membantu memvisualisasikan bagaimana pengguna berinteraksi dengan website.

Adapun use case yang dibuat untuk kebutuhan website seperti berikut:



Gambar 8. Use Case

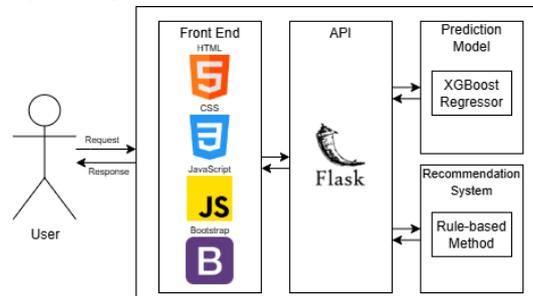
Pada *use case* tersebut, Pengguna dapat melihat dashboard untuk ringkasan informasi, mengakses deskripsi sistem, dan melihat profil tim pengembang. Untuk prediksi harga, pengguna mengisi formulir prediksi yang kemudian digunakan sistem untuk menampilkan hasil prediksi. Selain itu user dapat mengisi rekomendasi mobil kemudian akan digunakan sistem untuk Menampilkan tabel rekomendasi mobil.

2. Arsitektur Sistem

Arsitektur sistem ini menunjukkan alur interaksi antara pengguna, frontend, API, dan backend. Pengguna berinteraksi dengan sistem melalui antarmuka frontend yang dibangun menggunakan teknologi seperti HTML, CSS, JavaScript, dan Bootstrap. Frontend bertugas menangani permintaan pengguna dan menampilkan respons dalam bentuk antarmuka yang intuitif. Permintaan dari pengguna dikirimkan ke API yang dikembangkan menggunakan Flask, framework Python yang ringan. API ini bertindak sebagai penghubung antara frontend dan backend, menerima data dari frontend, memprosesnya, dan mengarahkan permintaan ke modul backend yang relevan.

Backend terdiri dari dua komponen utama yang masing-masing memiliki fungsi spesifik. *Prediction Model* menggunakan algoritme XGBoost Regressor untuk menghasilkan prediksi berdasarkan data yang diberikan oleh pengguna. Sementara itu, *Recommendation System* menggunakan metode berbasis aturan (*rule-based method*) untuk memberikan

rekomendasi yang relevan. Setelah backend memproses data, hasilnya dikirimkan kembali melalui API ke frontend, di mana informasi tersebut ditampilkan kepada pengguna. Arsitektur ini dirancang untuk memastikan interaksi yang efisien dan terstruktur, dengan pembagian tugas yang jelas antara setiap komponen.



Gambar 9. Arsitektur Website

Pengujian Sistem

Testing adalah proses penting dalam pengembangan perangkat lunak untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang diharapkan. Pada penelitian ini testing yang digunakan adalah blackbox testing. Blackbox yang berfokus pada pengujian aplikasi tanpa memperhatikan logika internal atau kode sumbernya. Blackbox testing mengevaluasi fungsionalitas aplikasi dari sudut pandang pengguna dengan memberikan berbagai input dan memeriksa apakah outputnya sesuai harapan [10].

HASIL DAN PEMBAHASAN

Model Machine Learning

Sistem ini dirancang untuk memprediksi harga jual mobil menggunakan berbagai algoritma regresi. Regresi Linear digunakan sebagai pendekatan awal untuk memahami hubungan fitur mobil dengan harga jual, sementara regresi Lasso dan Ridge mengatasi multikolinearitas serta memilih fitur yang paling berpengaruh. XGBoost Regressor diterapkan sebagai metode gradient boosting yang lebih canggih untuk meningkatkan akurasi prediksi.

Hiperparameter model XGBoost disesuaikan menggunakan GridSearchCV atau teknik tuning lainnya untuk mengoptimalkan jumlah pohon ($n_estimators$), kedalaman maksimum (max_depth), tingkat pembelajaran ($learning_rate$), serta rasio sampel dan kolom. Pengujian menunjukkan bahwa XGBoost Regressor memiliki akurasi lebih tinggi dan kesalahan prediksi lebih rendah dibandingkan Regresi Linear, Lasso, dan Ridge. Dengan kombinasi algoritma ini, sistem dapat menghasilkan prediksi harga jual mobil yang lebih akurat dan membantu konsumen serta penjual dalam pengambilan keputusan jual-beli kendaraan.

Tabel 2. Hasil Pelatihan Model

Algoritma	Akurasi Training	Akurasi Test	MSE	RMSE
Regresi Linear	85.89%	81.6%	4.2152	2.0531
Regresi Lasso	85.86%	81.73%	4.1846	2.0531
Regresi Ridge	85.89%	81.6%	4.2152	2.0456
XGBoost Regressor	97.79%	89.9%	2.3129	1.5208

Berdasarkan tabel 2, XGBoost Regressor menunjukkan performa terbaik dibandingkan metode regresi lainnya dengan akurasi 97.79% pada data latih dan 89.9% pada data uji. Model ini memiliki nilai Mean Squared Error (MSE) 2.3129 dan Root Mean Squared Error (RMSE)

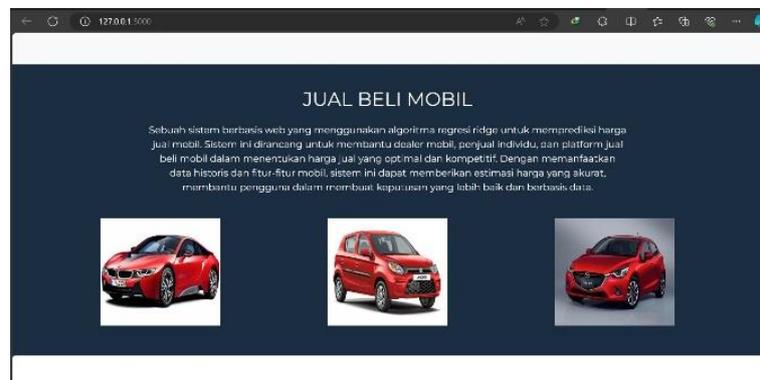
1.5208. Keunggulan XGBoost Regressor terletak pada kemampuannya menangani hubungan non-linier antar variabel serta mengoptimalkan bobot fitur melalui hyperparameter tuning. Sebaliknya, regresi linear, Lasso, dan Ridge memiliki akurasi lebih rendah (81.6% – 81.73%) dengan RMSE lebih tinggi. Keterbatasan utama model regresi ini adalah ketidakmampuannya menangkap pola kompleks dalam dataset harga mobil bekas. Meskipun Lasso dan Ridge mengurangi multikolinieritas, keduanya kurang optimal dalam mendeteksi variabel yang paling berpengaruh. Sehingga XGBoost Regressor menjadi model terbaik dalam memprediksi harga mobil bekas.

Tampilan Website

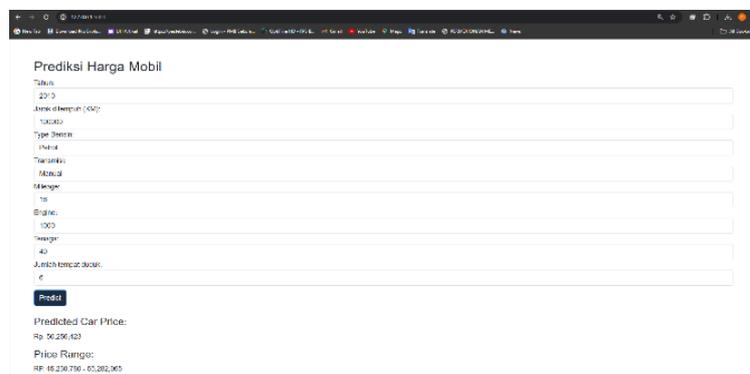
Adapun tampilan website yang dikembangkan sebagai berikut:



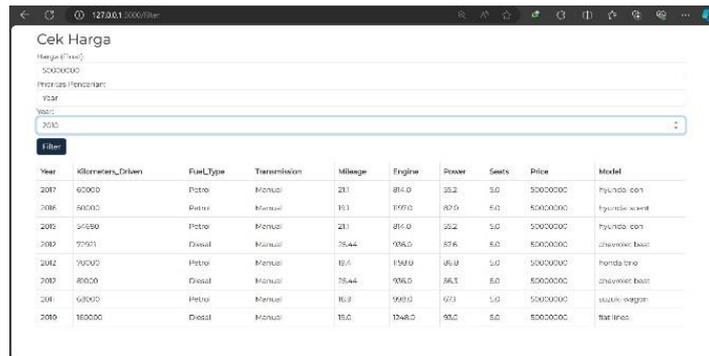
Gambar 10. Halaman Dashboard Sistem



Gambar 11. Halaman Deskripsi Sistem



Gambar 12. Halaman Prediksi Harga



Gambar 13. Halaman Rekomendasi Mobil



Gambar 14. Halaman Team Profil

Testing Black Box

Berikut adalah hasil skenario *test case* pada bagian prediksi harga mobil bekas.

Tabel 3. Skenario Test Case Bagian Prediksi Harga Mobil

Skenario Test	Output yang Diharapkan	Hasil Pengujian	Kesimpulan
Mengisikan tiap kolom form sesuai dengan instruksi. Setelah itu klik tombol "Predict".	Sistem akan membaca dan memproses hasil input kemudian menampilkan hasil prediksi harga mobil dan rentang harganya	Tampil hasil prediksi harga dan rentang harganya.	Berhasil
Salah satu kolom kosong atau tidak sesuai	Sistem akan mengalami error	Sistem akan mengalami error pada backend	Berhasil

Berikut adalah hasil skenario test case pada bagian pembelian mobil bekas berdasarkan harga.

Tabel 4. Skenario Test Case bagian prediksi harga mobil

Skenario Test	Output yang Diharapkan	Hasil Pengujian	Kesimpulan
Mengisikan tiap kolom form sesuai dengan instruksi. Pada kolom prioritas pencarian menggunakan tool combobox pilih salah satu saja.	Sistem akan membaca dan memproses hasil input kemudian menampilkan data lengkap mobil bekas sebanyak 10 data	10 data terdekat akan tampil berdasarkan harga mobil yang diinput beserta prioritas pencarian	Berhasil

Salah satu kolom kosong atau tidak sesuai	Sistem akan mengalami error	Sistem akan mengalami tampilan error	Sistem akan Berhasil
---	-----------------------------	--------------------------------------	----------------------

Pada pengujian sistem (*website*) menggunakan metode pengujian *black box* didapatkan bahwa semua fitur pada *website* berfungsi dengan baik dan menampilkan sesuai dengan yang diinginkan pengguna.

KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan sistem prediksi harga mobil bekas menggunakan XGBoost Regressor, yang menunjukkan akurasi 97.79% pada data latih dan 89.90% pada data uji. Sistem ini juga dilengkapi dengan fitur rekomendasi berbasis rule-based untuk membantu pengguna memperoleh rekomendasi kendaraan sesuai spesifikasi yang diprioritaskan. Sistem ini dapat diimplementasikan pada platform jual beli mobil bekas untuk meningkatkan akurasi estimasi harga, mendukung pengambilan keputusan, serta meningkatkan transparansi dalam transaksi.

Namun, penelitian ini memiliki keterbatasan, terutama pada cakupan dataset yang hanya mencakup spesifikasi kendaraan dan data penggunaan mobil tanpa mempertimbangkan kondisi fisik, riwayat perawatan, dan tren pasar. Oleh karena itu, penelitian selanjutnya disarankan untuk memperkaya dataset dengan faktor eksternal tersebut serta menerapkan teknik optimasi model lebih lanjut guna meningkatkan akurasi prediksi harga mobil bekas.

DAFTAR PUSTAKA

- [1] Y. M. Geasela, J. Isabel, S. Pereisia, A. F. N. Runkat, and F. Assahara, "Pengembangan Aplikasi Penyewaan Lapangan 'Connsfield' Berbasis Website," *Journal of Technology and Informatics (JoTI)*, Vol.4, No. 2, pp 69-76, 2023, doi: 10.37802/joti.v4i2.320
- [2] P. A. Azhar and M. A. Pratama, "Prediksi Harga Mobil Audi Bekas Menggunakan Model Regresi Linear Dengan Framework Streamlit," *Journal of Technology and Informatics (JoTI)*, Vol. 6, No. 1, pp 22-28, 2024. doi: 10.37802/joti.v6i1.763
- [3] B. Kriswantara and R. Sadikin, "Used Car Price Prediction with Random Forest Regressor Model," *Journal of Information Systems, Informatics and Computing*, Vol. 6, No. 1, pp 40-49, 2022. doi: 10.52362/jisicom.v6i1.752
- [4] E. Hasibuan and A. Karim, "Implementasi Machine Learning untuk Prediksi Harga Mobil Bekas dengan Algoritma Regresi Linear berbasis Web", *Jurnal Ilmiah KOMPUTASI*, Vol. 21, No 4, pp 595-602, 2022. doi: 10.32409/jikstik.21.4.3327
- [5] M. A. A. Syukur and M. Faisal, "Penerapan Model Regresi Linear Untuk Estimasi Mobil Bekas Menggunakan Bahasa Python," *EULER: Jurnal Ilmiah Matematika, Sains dan Teknologi*, vol. 11, no. 2, pp. 182-191, 2023. doi: 10.37905/euler.v11i2.20698.
- [6] A. Amalia, M. R. Radhi, D. R. H. Sitompul, S. H. Sinurat, and E. Indra, "Prediksi Harga Mobil Menggunakan Algoritma Regresi dengan Hyper-Parameter Tuning," *JUSIKOM PRIMA (Jurnal Sistem Informasi dan Ilmu Komputer Prima)*, Vol. 4 No. 2, pp 28-32, 2021.
- [7] P. Simamora, S. A. Pasaribu, dan V. Wijaya, "Peningkatan dan Optimalisasi Prediksi Harga Emas Menggunakan Metode Combine Machine Learning Random Forest dan Gradient Boosting," *Jurnal Mahkota Informatika*, vol. 1, no. 1, pp. 42-52, 2024. [Online]. Tersedia: <https://mojs.mtu.ac.id/index.php/mi>.
- [8] T. A. Pratami, Tursina, and R. Septiriana, "Rekomendasi Pemilihan Model Sepeda

- Menggunakan Rule Based System," *JITET (Jurnal Informatika and Teknik Elektro Terapan)*, vol. 12, no. 2, pp 1350-1361, 2024. doi: 10.23960/jitet.v12i2.4239
- [9] Y. M. Widyastuti, T. Oktiarso, and N. K. Putrianto, "Perencanaan dan analisis kebutuhan pengguna dalam pengembangan sistem informasi hubungan pelanggan (studi kasus di perusahaan bidang jasa finansial)," *KURAWAL Jurnal Teknologi, Informasi dan Industri*, vol. 7, no. 1, 2024.
- [10] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, "Pengujian Black Box dan White Box Sistem Informasi Parkir Berbasis Web", *Jurnal Testing dan Implementasi Sistem Informasi*, Vol. 1., No. 1, 2023, pp. 1-16.