Prediksi Harga Mobil pada Sebuah Dealer menggunakan Pemodelan Regresi *Random Forest, GradientBoost* dan *XGBoost*

Achmad Ajie Priyajie[1, 2], Adi Purnama [3], Alwan Maulana Firdaus[3]
[1]Jurusan Teknik Informatika, UIN Sunan Gunung Djati Bandung
[2]Jurusan Teknik Informatika, UIN Sunan Gunung Djati Bandung
[3]Jurusan Teknik Informatika, UIN Sunan Gunung Djati Bandung

Article Info

Article history:

Received month July 7th , 2025
Revised month July 8th , 2025
Accepted month July 9th, 2025

Keywords:

Prediksi Harga Mobil
Random Forest
GradientBoost
XGBoost
Regresi
Machine Learning

ABSTRACT

Dalam industri otomotif modern, prediksi harga jual mobil menjadi tantangan penting dalam merumuskan strategi penetapan harga yang kompetitif. Penelitian ini membandingkan kinerja tiga algoritma regresi berbasis *machine learning* seperti *Random Forest*, *GradientBoost* , dan *XGBoost* dalam memprediksi harga mobil berdasarkan atribut kendaraan seperti usia, jarak tempuh, tipe bahan bakar, dan kapasitas mesin. Dataset yang digunakan terdiri dari 15.411 data penjualan mobil dari situs *Kaggle*. Proses penelitian meliputi pembersihan data, encoding variabel kategorial, scaling, pembagian data latih dan uji, serta pelatihan dan penyetelan model menggunakan teknik *cross-validation* dan *RandomizedSearchCV*. Evaluasi model dilakukan menggunakan MAE, MSE, RMSE, dan R^2, disertai analisis *learning curve* dan *Q-Q plot* untuk mengamati kinerja dan distribusi residual. Hasil penelitian menunjukkan bahwa *Random Forest* memberikan keseimbangan terbaik antara akurasi prediksi dan stabilitas model, meskipun *XGBoost* mencatat skor R^2 tertinggi. Studi ini menunjukkan bahwa pemilihan algoritma yang tepat sangat penting untuk membangun sistem prediksi harga yang andal dan efisien dalam konteks data non-linear.

Corresponding Author:

1. Achmad Ajie Priyajie, 2. Adi Purnama, 3. Alwan Maulana Firdaus
Jurusan Teknik Informatika, UIN Sunan Gunung Djati Bandung
Email: achmadajie74@gmail.com , adipuramaa8@gmail.com , alwanmaulana24@gmail.com

1. PENDAHULUAN

. Dalam era digital yang semakin maju ini, industri otomotif menghadapi tantangan besar dalam menentukan strategi penetapan harga yang kompetitif dan sesuai dengan kondisi pasar. Salah satu aspek terpenting dalam proses ini adalah kemampuan untuk secara akurat menentukan harga suatu kendaraan (mobil) berdasarkan karakteristik pasar dan kondisinya. Penentuan harga yang tepat tidak hanya memengaruhi keputusan pembelian konsumen, tetapi juga strategi bisnis para produsen dan distributor. Oleh karena itu, pengembangan sistem prediksi berbasis data menjadi sangat penting bagi industri ini.

Data historis yang tersedia, seperti informasi mengenai tahun pembuatan, jangka waktu penggunaan, kapasitas mesin, dan fitur kendaraan, memberikan peluang besar untuk melakukan analisis prediktif. Namun, kompleksitas hubungan antarvariabel serta jenis kendaraan menuntut penggunaan metode yang mampu menangani non-linearitas dan interaksi fitur yang rumit. Dalam konteks ini, algoritma *machine learning* berbasis regresi, seperti *Random Forest, Gradient Boosting*, dan *XGBoost*, merupakan alat yang memiliki performa prediksi terbaik saat menangani data dengan pola yang kompleks dan terdistribusi secara non-linear.

Tujuan dari penelitian ini adalah membandingkan kinerja tiga algoritma regresi dalam menentukan harga mobil berdasarkan atribut kendaraan yang relevan. Model *Random Forest* dipilih karena kemampuannya memprediksi menggunakan berbagai teknik pohon keputusan yang stabil terhadap *overfitting*, sementara *GradientBoost* dan *XGBoost* dipilih karena kemampuannya mengoptimalkan masalah secara metodis menggunakan teknik *adaptive boosting*.

2. METODE

A. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data tentang penjualan mobil yang didapat dari sumber berikut : https://www.kaggle.com/datasets/manishkr1754/cardekho-used-car-data . Dataset terdiri dari 14 Label dan 15411 data point. Adapun atribut dari data yang digunakan adalah sebagai berikut:

Tabel 1 Atribut Data

Atribut Data	Keterangan
car_name	nama lengkap mobil (gabungan merek dan model)
brand	merek mobil
model	model mobil
vehicle_age	usia mobil dalam tahun
km_driven	total jarak tempuh dalam kilometer
seller_type	jenis penjual
fuel_type	jenis bahan bakar
transmission_type	jenis transmisi mobil
mileage	konsumsi bahan bakar
engine	kapasitas mesin mobil dalam cc
max_power	daya maksimum mesin dalam tenaga kuda
seats	jumlah kursi
selling_price	harga jual mobil dalam rupee

B. . Data Understanding

Pada tahap ini, dilakukan pemeriksaan awal untuk memahami struktur dan karakteristik dari dataset.

1. Struktur dan Isi Dataset

   Dataset yang digunakan diperoleh dari file CSV bernama "cardekho_imputated.csv" yang berisi data harga jual berbagai jenis mobil berdasarkan spesifikasi dan perfoma. Data diimpor menggunakan library Python seperti *pandas*, *numpy*, dan *matplotlib* [9] , [10] .
2. Distribusi Data

   Untuk memahami pola penyebaran data pada masing-masing atribut mobil, digunakan visualisasi dalam bentuk histogram [10] , [11]. Histogram dipilih karena dapat memperlihatkan sebaran frekuensi nilai dan membantu mendeteksi potensi keberadaan outlier maupun pola distribusi tertentu..
3. Visualisasi Hubungan Antar Variabel

   Tahapan ini dilakukan dengan memvisualisasikan hubungan antar variabel menggunakan lineplot. Visualisasi ini bertujuan untuk mengamati kecenderungan perubahan antar variabel dan memberikan indikasi awal mengenai kemungkinan hubungan linier atau pola yang berulang [2].

C. Data Preparation

Tahap ini bertujuan untuk mengubah data mentah menjadi bentuk yang dapat digunakan secara efektif dalam proses pelatihan model prediksi. Tahapan dalam data preparation ini mencakup sebagai berikut:

1. Seleksi Variabel

   Pada tahap seleksi variabel, dilakukan dengan memilih 11 variabel dari dataset, yang terdiri dari 10 variabel independen dan 1 variabel dependen, yaitu "*selling_price*". Pemilihan variabel tersebut dilakukan berdasarkan relevansi terhadap spesifikasi dan perfoma utama yang menjadi fokus utama.
2. Pembersihan Data

   Dilakukan pemeriksaan nilai duplikat untuk dihapus agar menghindari bias pada data. Deteksi nilai hilang (*missing values*) juga dilakukan, dan seluruh baris yang mengandung nilai kosong dihapus untuk menjaga konsistensi [12] , [13].
3. *Train-Test Split*

   Data yang telah dibersihkan, dibagi menjadi 2 bagian menggunakan metode *train-test split*, dengan proposi 80% untuk pelatihan (training) dan 20% untuk pengujuan (testing) [14] .
4. *Feature Scaling dan Encoder*

   Mengingat variabel memiliki rentang nilai yang berbeda signifikan, transformasi ini dilakukan untuk meningkatkan peforma algoritma *machine learning* yang sensitif terhadap skala [15] , [11], [16]. Proses *fitting* dilakukan pada data latih dan hasilnya digunakan untuk mentransformasikan data latih dan data uji secara konsisten. Selain itu, pada penelitian kali ini kami juga melakukan F*eature Encoder* menggunakan *OneHotEncoder* dan *LabelEncoder* [17] , [6] , [18], [19]. Tujuannya adalah agar tipe data yang tidak cocok untuk regresi seperti kategorial, diskrit dan *object* menjadi cocok untuk regresi.

D. Training Model

Tahapan *data modeling* bertujuan untuk membangun dan melatih model regresi yang dapat memprediksi variabel target berdasarkan sejumlah variabel independen. Dalam penelitian ini, digunakan tiga algoritma regresi sebagai *baseline* model, yaitu, *Random Forest Regressor* [20], [6] , [21], [22] *GradientBoost Regressor* [23], [24], [11] dan *XGBoost Regressor* [25], [26], [8], [10]. Ketiga model dilatih menggunakan data latih (*X_train* dan *y_train*) yang telah melalui proses *feature scaling* dan *Encoder*. Kemudian dilakukan penyetelan (*tuning*) parameter lebih lanjut.

E. Evaluasi Model

Untuk proses evaluasi model, digunakan fungsi khusus yang menghitung empat metrik evaluasi regresi, yaitu *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *Coefficient of Determination* (R^2 Score).

Model yang telah dilatih kemudian digunakan untuk melakukan prediksi terhadap data latih dan data uji (*X_test*). Untuk memahami kinerja model terhadap jumlah data pelatihan yang berbeda, dilakukan visualisasi *learning curve* untuk setiap model menggunakan *cross-validation* sebanyak 5 kali lipat. Grafik *learning curve*

ini memperlihatkan perbandingan antara training error dan validation error terhadap jumlah data pelatihan, dengan metrik evaluasi berupa MSE [26] , [27].
Analisis *residual* dilakukan untuk menguji asumsi distribusi normal terhadap selisih antara nilai aktual dan nilai prediksi (*residual*). Distribusi *residual* dari masing-masing model dianalisis menggunakan *Q-Q Plot* (*Quantile-Quantile Plot*) untuk mengevaluasi sejauh mana residual mengikuti distribusi normal [28] , [29].

F. Retraining Model

Model akan di-tuning menggunakan teknik *RandomizedSearchCV* dengan *cross-validation* sebanyak 3 kali lipat. Proses ini dilakukan untuk mencari kombinasi parameter terbaik yang menghasilkan kinerja optimal [11] , [30] , [31].

Setelah itu, dilakukan retraining model dengan parameter hasil *tuning* tersebut. Model hasil *retraining* dievaluasi kembali dengan metrik yang sama (MAE, MSE, RMSE, dan R^2) baik pada data latih maupun data uji. Evaluasi ini bertujuan untuk mengetahui peningkatan performa model setelah dilakukan *tuning*, dan juga memastikan bahwa model tidak mengalami overfitting [22] , [13] , [21].

Setelah model dilatih dan disempurnakan melalui proses retraining dengan parameter terbaik, tahap selanjutnya adalah melakukan evaluasi eksternal menggunakan data baru yang belum pernah dilihat oleh model sebelumnya. Evaluasi ini bertujuan untuk menguji kemampuan generalisasi model dalam memprediksi keluaran berdasarkan input yang benar-benar baru dan tidak termasuk dalam data latih maupun uji sebelumnya.

3. HASIL PENELITIAN

A. Data Import and Cleaning

Pada penelitian kali ini, kami melakukan import library seperti *pandas, numpy, matplotlib dan seaborn*. Kemudian kami melakukan impor file pada *Google Colaboratory* yaitu untuk dataset yang akan digunakan. File dataset yang sudah diimport akan dibaca oleh *Google Colaboratoy* dengan *pd.read_csv.* Setelah itu kami melakukan penggalian informasi mengenai dataset tersebut. Kami terlebih dahulu untuk memilih variabel-variabel yang akan digunakan seperti pada Gambar 2. Masih menggunakan *library pandas*, kami tinggal menulis nama kolom-kolom yang akan dieliminasi ke dalam sebuah *array*. Hasil *array* tersebut diproses untuk membentuk data baru.

	car_name	brand	model	vehicle_age	km_driven	seller_type	fuel_type	transmission_type	mileage	engine	max_power	seats	selling_price
0	Maruti Alto	Maruti	Alto	9	120000	Individual	Petrol	Manual	19.70	796	46.30	5	120000
1	Hyundai Grand	Hyundai	Grand	5	20000	Individual	Petrol	Manual	18.90	1197	82.00	5	550000
2	Hyundai i20	Hyundai	i20	11	60000	Individual	Petrol	Manual	17.00	1197	80.00	5	215000
3	Maruti Alto	Maruti	Alto	9	37000	Individual	Petrol	Manual	20.92	998	67.10	5	226000
4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer	Diesel	Manual	22.77	1498	98.59	5	570000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
19537	Hyundai i10	Hyundai	i10	9	10723	Dealer	Petrol	Manual	19.81	1086	68.05	5	250000
19540	Maruti Ertiga	Maruti	Ertiga	2	18000	Dealer	Petrol	Manual	17.50	1373	91.10	7	925000
19541	Skoda Rapid	Skoda	Rapid	6	67000	Dealer	Diesel	Manual	21.14	1498	103.52	5	425000

Gambar 1 Format Data Awal

	model	vehicle_age	km_driven	seller_type	fuel_type	transmission_type	mileage	engine	max_power	seats	selling_price
0	Alto	9	120000	Individual	Petrol	Manual	19.70	796	46.30	5	120000
1	Grand	5	20000	Individual	Petrol	Manual	18.90	1197	82.00	5	550000
2	i20	11	60000	Individual	Petrol	Manual	17.00	1197	80.00	5	215000
3	Alto	9	37000	Individual	Petrol	Manual	20.92	998	67.10	5	226000
4	Ecosport	6	30000	Dealer	Diesel	Manual	22.77	1498	98.59	5	570000

Gambar 2 Variabel yang Terpilih

Kemudian setelah memilih variabel terpilih, Kami mengkoversi kolom '*model*' ke dalam array agar bisa diproses *Encoder* kemudian. Selanjutnya kami melakukan penggalian informasi mengenai fitur-fitur pada data. Berdasarkan penggalian informasi tersebut, kami mendapati empat jenis fitur yang ada pada dataset yaitu : *numeric*, kategori, diskrit dan kontinu.

```
## Mengetahui fitur-fitur yang ada
num_features = [feature for feature in df1.columns if df1[feature].dtype != 'O']
print('Num of Numerical Features :', len(num_features))
cat_features = [feature for feature in df1.columns if df1[feature].dtype == 'O']
print('Num of Categorical Features :', len(cat_features))
discrete_features=[feature for feature in num_features if len(df1[feature].unique())<=25]
print('Num of Discrete Features :',len(discrete_features))
continuous_features=[feature for feature in num_features if feature not in discrete_features]
print('Num of Continuous Features :',len(continuous_features))
```

```
Num of Numerical Features : 7
Num of Categorical Features : 4
Num of Discrete Features : 2
Num of Continuous Features : 5
```

Gambar 3 Informasi Fitur

Kemudian, kami melakukan pembersihan data dengan mengecek nilai yang duplikat. Hasil pengecekan tersebut yaitu didapat terdapat enam data point yang memiliki nilai-niali sama dengan data point lainnya. Adapun yang kami lakukan yaitu menghapus salah satu baris dan mempertahankan lainnya.

```
##Menghapus duplikat value namun mempertahankan yang pertama
df2 = df1.drop_duplicates()
df2
```

	model	vehicle_age	km_driven	seller_type	fuel_type	transmission_type	mileage	engine	max_power	seats	selling_price
0	Alto	9	120000	Individual	Petrol	Manual	19.70	796	46.30	5	120000
1	Grand	5	20000	Individual	Petrol	Manual	18.90	1197	82.00	5	550000
2	i20	11	60000	Individual	Petrol	Manual	17.00	1197	80.00	5	215000
3	Alto	9	37000	Individual	Petrol	Manual	20.92	998	67.10	5	226000
4	Ecosport	6	30000	Dealer	Diesel	Manual	22.77	1498	98.59	5	570000
...	...	...	...	...	...	...	...	...	...	...	...
19537	i10	9	10723	Dealer	Petrol	Manual	19.81	1086	68.05	5	250000
19540	Ertiga	2	18000	Dealer	Petrol	Manual	17.50	1373	91.10	7	925000
19541	Rapid	6	67000	Dealer	Diesel	Manual	21.14	1498	103.52	5	425000

Gambar 4 Hasil Pembersihan Data yang Duplikat.

Selain pengecekan nilai duplikat, kami juga melakukan pengecekan terhadap *null value*. Adapun pada penelitian kali ini yaitu kami memilikinya pada beberapa baris. Untuk mengatasinya, maka kami memutuskan untuk membuang baris yang mengandung *null value* tersebut.

```
## Mengecek missing value
missing_per_col = df2.isna().sum()
print(missing_per_col)
```

```
model                0
vehicle_age          0
km_driven            0
seller_type          0
fuel_type            0
transmission_type    0
mileage              0
engine               0
max_power            0
seats                0
selling_price        0
dtype: int64
```

Gambar 5 Null Value

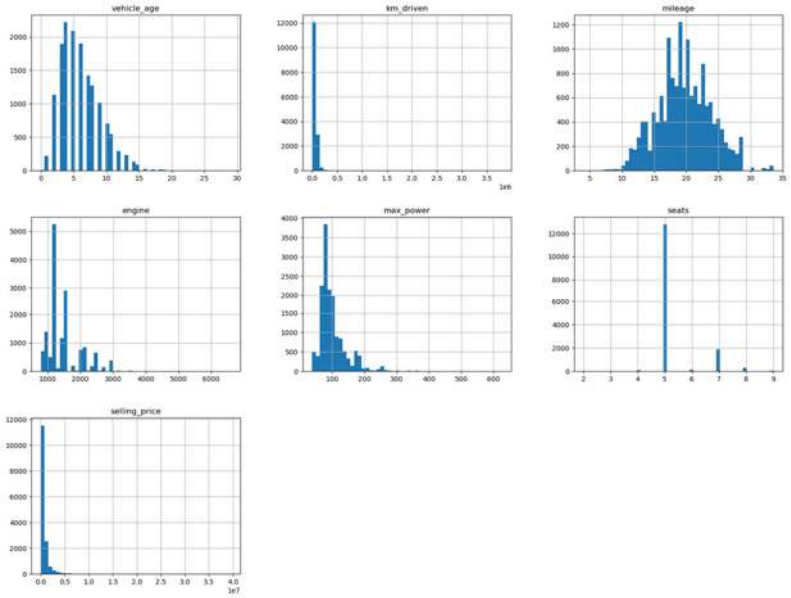
Untuk mendapatkan info dan nilai-nilai penting bagi evaluasi nantinya, maka kami melakukan penggalian info untuk nilai – nilai penting bagi evaluasi.Info-info penting tersebut meliputi jumlah data setiap variabel, rata-rata setiap variabel, simpangan baku, nilai terkecil, nilai terbesar, kuartil I, kuartil II dan kuartil III . Untuk lebih jelasnya peran info-info tersebut silahkan tunggu hingga hasil evaluasi nanti

	vehicle_age	km_driven	mileage	engine	max_power	seats	selling_price
count	15244.000000	1.524400e+04	15244.000000	15244.000000	15244.000000	15244.000000	1.524400e+04
mean	6.041131	5.563958e+04	19.697333	1486.171543	100.607652	5.326161	7.747014e+05
std	3.016228	5.176630e+04	4.169307	520.419390	42.915687	0.808760	8.946761e+05
min	0.000000	1.000000e+02	4.000000	793.000000	38.400000	0.000000	4.000000e+04
25%	4.000000	3.000000e+04	17.000000	1197.000000	74.000000	5.000000	3.850000e+05
50%	6.000000	5.000000e+04	19.670000	1248.000000	88.500000	5.000000	5.590000e+05
75%	8.000000	7.000000e+04	22.700000	1582.000000	117.300000	5.000000	8.250000e+05
max	29.000000	3.800000e+06	33.540000	6592.000000	626.000000	9.000000	3.950000e+07

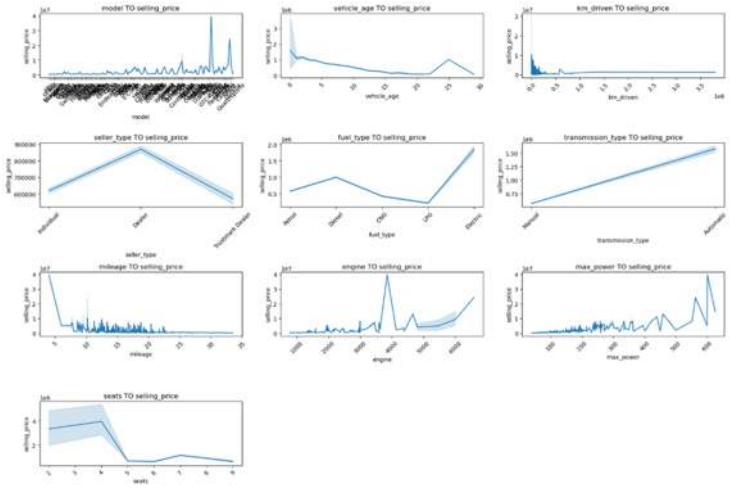
Gambar 6 Info - Info Penting

B. Visualisasi Grafik

Langkah selanjutnya yaitu kami mulai melakukan visualisasi grafik untuk mengetahui pola pada data. Kami mulai melakukan visualisasi grafik untuk diagram batang di tiap variabel. Tujuannya adalah untuk mengetahui rentang nilai yang ada di setiap variabel. Dengan begitu, bisa diketahui apakah nilai-nilai tiap variabel terskalakan dengan baik atau memiliki ketimpangan.



Gambar 7 Histogram Batang



Gambar 8 Histogram Lineplot

Setelah itu, kemudian kami melakukan *feature selection*, yaitu dengan pertama-tama menentukan variabel X dan Y. Untuk variabel Y kami memilih kolom Lain-lain dan sisanya sebagai X. Kemudian, kami melakukan *train test split* dengan rasio pembagian 80% untuk data latih dan 20% untuk data tes dengan *random state* = 42.

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
X['model']=le.fit_transform(X['model'])
```

```
X.head()
```

	model	vehicle_age	km_driven	seller_type	fuel_type	transmission_type	mileage	engine	max_power	seats
0	7	9	120000	Individual	Petrol	Manual	19.70	796	46.30	5
1	54	5	20000	Individual	Petrol	Manual	18.90	1197	82.00	5
2	118	11	60000	Individual	Petrol	Manual	17.00	1197	80.00	5
3	7	9	37000	Individual	Petrol	Manual	20.92	998	67.10	5
4	38	6	30000	Dealer	Diesel	Manual	22.77	1498	98.59	5

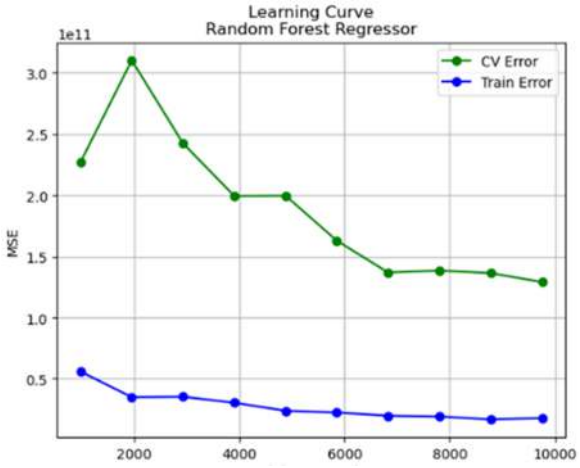
Gambar 10 Transform Kolom Model bagian 2

Selanjutnya yaitu kami melakukan *Scaling*. Tujuannya adalah untuk menormalisasi, mengurangi dominasi fitur ekstrem dan meningkatkan kesetaraan antar fitur dalam pemodelan. Selain itu, kami juga melakukan konversi pada beberapa variabel yaitu *'model', 'seller_type', 'fuel_type','transmission_type'*. Variabel-variabel tersebut memiliki format data yang tidak cocok untuk pemodelan regresi, sehingga kami memutuskan untuk melakukan konversi agar variabel cocok dengan model regresi. Kami memanfaatkan pustaka *Python* yaiitu *LabelEncoder* dan *OneHotEncoder*.

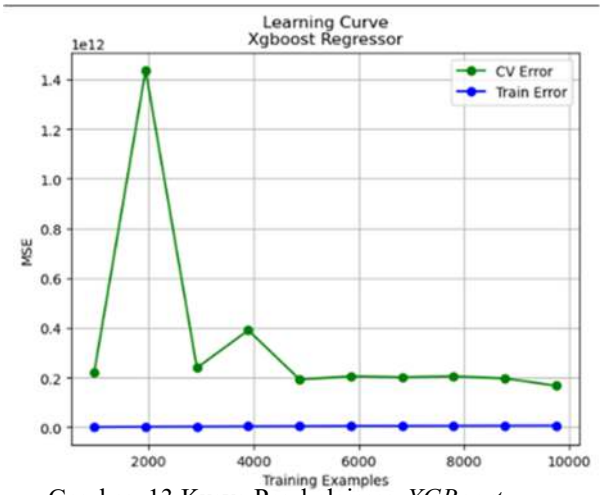
C. Training Model

Untuk pemodelan regresi yang kami gunakan, kami memilih tiga jenis pemodelan : *Random Forest, GradientBoost* dan *XGBoost*. Model regresi dibuat menggunakan data latih. Tujuannya adalah agar model dapat memprediksi nilai variabel Y berdasarkan variabel X. Model regresi yang dibangun selanjutnya akan dievaluasi terlebih dahulu melalui tiga indikator.

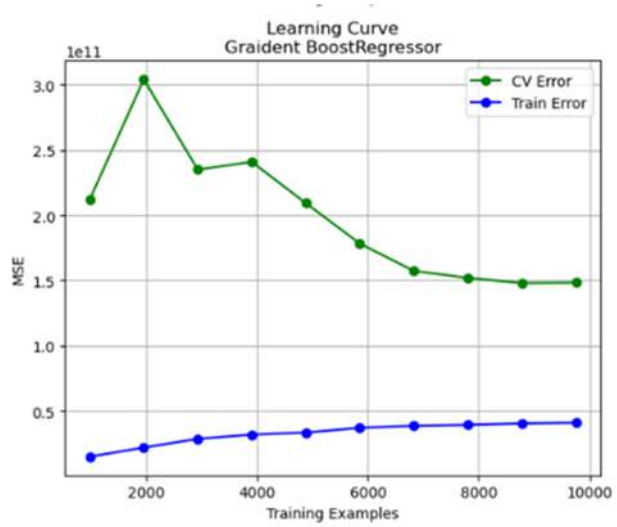
Pertama yaitu evaluasi standar regresi yang meliputi *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), *Root Mean Square Error* (RMSE) dan *Coefficient Determination* (R2 *score*). Hasil evaluasi standar meliputi yang menggunakan data latih dan satu lagi untuk data tes. Untuk hasil standar evaluasi model dapat dlihat pada Tabel 2. Kedua yaitu, meninjau kurva pembelajaran untuk setiap model. Model regresi yang baik memiliki nilai *training error* yang menurun dan tidak terlalu dekat dengan nilai *validation error*. Jika nilai *training error* naik maka model kurang baik. Sedangkan jika nilai *training error* sama dengan *validation error*, maka model *overfitting*. Ketiga yaitu meninjau distribusi normal melalui *Q-Q Plot*. Tujuannya adalah untuk mengetahui sebaran data dari pemodelan yang dibuat. Semakin sebaran data mendekati garis normal, maka perfoma model semakin baik. Namun jika sebaran data terlalu menempel dengan garis normal, maka model sama dengan *overfitting*.



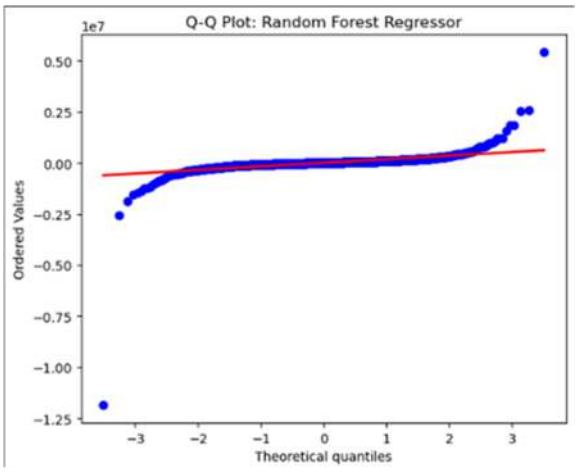
Gambar 12 Kurva Pembelajaran *Random Forest*



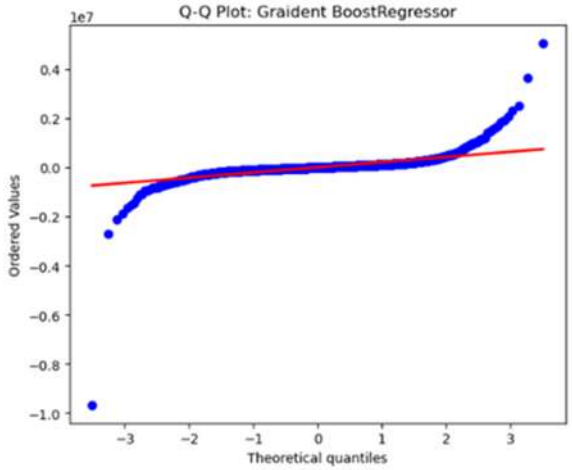
Gambar 13 Kurva Pembelajaran *XGBoost*



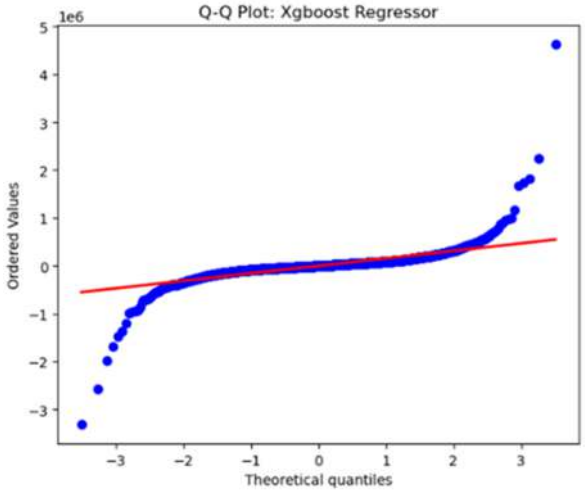
Gambar 14 Kurva Pembelajaran *GradientBoost*



Gambar 15 *Q-Q Plot Random Forest*



Gambar 16 *Q-Q Plot GradientBoost*



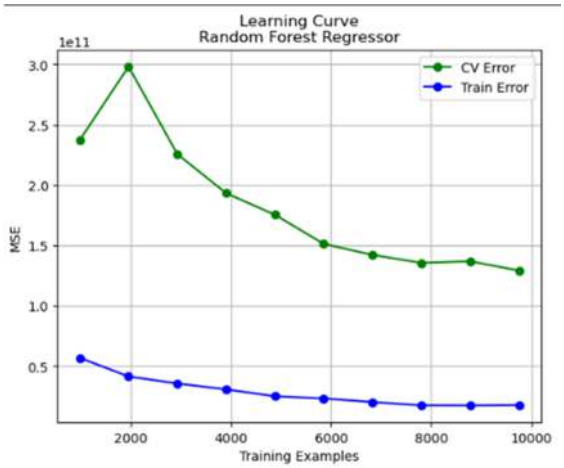
Gambar 17 *Q-Q Plot XGBoost*

Tabel 2 Standar Evaluasi Regresi

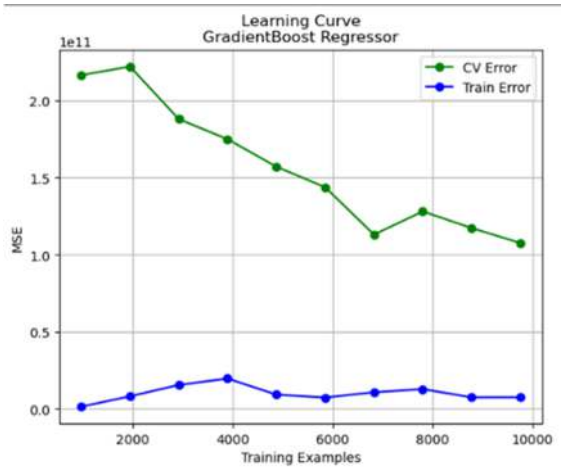
Model	**R2 *Score***		**MAE**		**MSE**		**RMSE**	
	Train	Test	Train	Test	Train	Test	Train	Test
Random Forest	0.9778	0.8578	40032.87	102562.72	18870469884.67	86440170247.75	137369.82	294007.09
Gradient Boost	0.9482	0.8063	112714.42	128372.59	43956408515.00	11777463366.12	209657.83	343183.09
XGBoost	0.9903	0.9300	63046.82	97172.21	8267483136.00	42542518272.00	90925.70	206258.37

D. Retraining Model

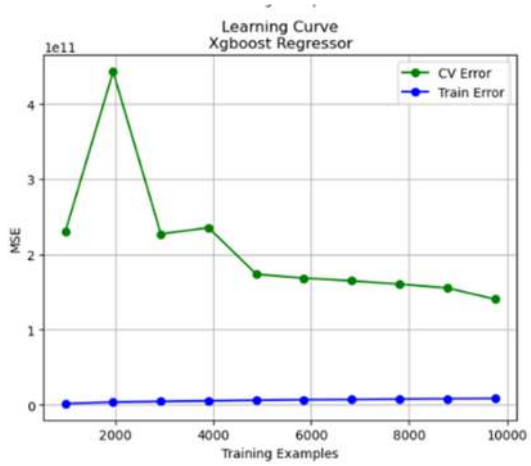
Setelah sebelumnya model regresi dilatih, untuk memaksimalkan perfomanya, maka kami melakukan *hyperparameter tuning*. Itu merupakan suatu teknik dalam membangun model *machine learning* menggunakan paramater terbaik berdasarkan data yang digunakan. Untuk mencari parameter terbaik, maka dapat dicari menggunakan teknik *cross-validation*. Dengan teknik tersebut model akan melakukan uji parameter dimana dengan data latih yang dibagi menjadi beberapa bagian lagi. Adapun pada penelitian kali ini, *cross validation*. Setelah parameter terbaik diperoleh, maka dilakukanlah *retraining model*. Ini dimana model regresi akan dilatih kembali seperti pada training model namun kali ini akan menggunakan parameter terbaik yang sudah diperoleh. Untuk seluruh indikator evaluasi sama seperti sebelumnya.



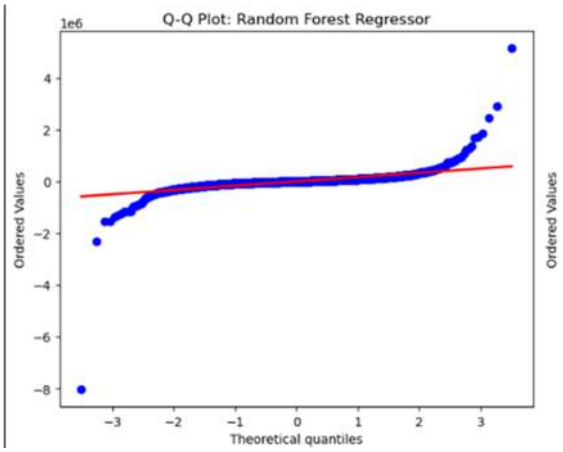
Gambar 18 Kurva Pembelajaran *Random Forest - Retraining -*



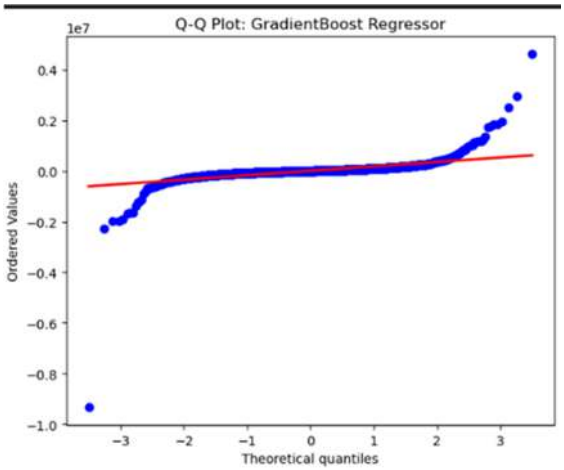
Gambar 19 Kurva Pembelajaran *GradientBoost - Retraining -*



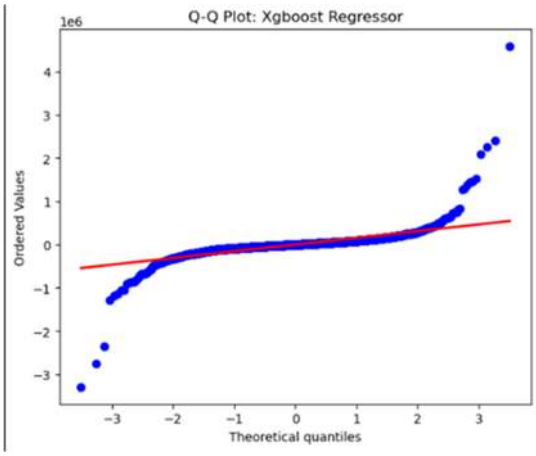
Gambar 20 Kurva Pembelajaran *XGBoost - Retraining -*



Gambar 21 *Q-Q Plot Random Forest - Retraining -*



Gambar 22 *Q-Q Plot GradientBoost - Retraining -*



Gambar 23 *Q-Q Plot XGBoost - Retraining -*

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
Fitting 3 folds for each of 100 candidates, totalling 300 fits
Fitting 3 folds for each of 100 candidates, totalling 300 fits
---------- Best Params for RF ------------------
{'n_estimators': 100, 'min_samples_split': 2, 'max_features': 7, 'max_depth': 15}
---------- Best Params for GradientBoost ------------------
{'n_estimators': 100, 'min_samples_split': 8, 'max_depth': 10, 'loss': 'huber', 'criterion': 'squared_error'}
---------- Best Params for XGboost ------------------
{'n_estimators': 300, 'max_depth': 8, 'learning_rate': 0.1, 'colsample_bytree': 0.5}
```

Gambar 24 Parameter Terbaik

Tabel 3 Evaluasi Standar Regresi - *Retraining* -

Model	**R2 *Score***		**MAE**		**MSE**		**RMSE**	
	Train	Test	Train	Test	Train	Test	Train	Test
Random Forest	0.9808	0.8986	544559.03	248269.58	16308292080.32	61637787046.41	122703.92	248269.58
Gradient Boost	0.9913	0.8910	47618.09	100973.26	7354816487.56	66260.173261.40	85760.22	257410.51
XGBoost	0.9893	0.9275	65631.27	95393.44	9046411264.00	44053094400.00	95112.62	209888.29

4. CONCLUSION

Berdasarkan proses-proses diatas, maka kami memutuskan untuk menggunakan *Random Forest Regression* sebagai model regresi kami pada studi kali ini. Adapun alasannya adalah karena *Random Forest* memiliki nilai-nilai evaluasi yang lebih baik daripada model-model lainnya walaupun bukan yang terbaik. Selain itu, dari *learning curve Random Forest Regression* lebih aman daripada *overfitting* ataupun *underfitting* . Kemudian, dari sebaran data *Random Forest* lebih dekat dengan garis normal (distribusi data nya baik). Sekilas jika melihat evaluasi, semua model memiliki perfoma yang baik. Namun itu tidaklah cukup. Untuk menguji hasil pemodelan regresi, maka kita harus melakukan dua visualisasi penting. Pertama, yaitu *learning curve* untuk mengetahui perfoma model terhadap jumlah data yang semakin banyak. Kedua, yaitu *Q-Q Plot* untuk mengetahui sebaran data dengan garis prediksi normal.

DAFTAR PUSTAKA

[1] P. Zhang, Y. Jia, and Y. Shang, “Research and application of XGBoost in imbalanced data,” *Int J Distrib Sens Netw*, vol. 18, no. 6, Jun. 2022, doi: 10.1177/15501329221106935.

[2] Q. Zhang, X. Han, X. Fang, M. Liu, K. Ge, and H. Jiang, “Optical frequency multiplication using residual network with random forest regression,” *Heliyon*, vol. 10, no. 10, May 2024, doi:

10.1016/j.heliyon.2024.e30958.

[3] J. Shah, “Gradient Boosting COMPREHENSIVE ASSIGNMENT GRADIENT BOOSTING.” [Online]. Available: https://www.researchgate.net/publication/354401342

[4] S. Ramya, S. Srinath, and P. Tuppad, “Enhanced Wastewater Parameter Prediction using Comprehensive Regression Framework: Leveraging Feature Engineering and Hyperparameter Optimization with Machine Learning,” in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 822–830. doi: 10.1016/j.procs.2025.03.263.

[5] S. Emami and G. Martínez-Muñoz, “Condensed-gradient boosting,” *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 1, pp. 687–701, Jan. 2025, doi: 10.1007/s13042-024-02279-0.

[6] K. Purnomo, R. A. Wijaya, M. Fajar, and P. A. Suri, “Comparative Analysis of BiLSTM Deep Learning Model and Random Forest Regressor Performance on Indonesian Nickel Mining Company Stock Prices,” in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 778–786. doi: 10.1016/j.procs.2024.10.304.

[7] N. Simarmata *et al.*, “Comparison of random forest, gradient tree boosting, and classification and regression trees for mangrove cover change monitoring using Landsat imagery,” *Egyptian Journal of Remote Sensing and Space Science*, vol. 28, no. 1, pp. 138–150, Mar. 2025, doi: 10.1016/j.ejrs.2025.02.002.

[8] J. Avanija, G. Sunitha, K. Reddy Madhavi, P. Kora, R. Hitesh, and S. Vittal E A Associate, “Prediction of House Price Using XGBoost Regression Algorithm,” 2021. [Online]. Available: https://www.researchgate.net/publication/350810698

[9] N. Gunasekara, B. Pfahringer, H. Gomes, and A. Bifet, “Gradient boosted trees for evolving data streams,” *Mach Learn*, vol. 113, no. 5, pp. 3325–3352, May 2024, doi: 10.1007/s10994-024-06517-y.

[10] S. Fatima, A. Hussain, S. Bin Amir, S. Haseeb Ahmed, and S. Muhammad Huzaifa Aslam, “XGBoost and Random Forest Algorithms: An In-Depth Analysis.”

[11] B. Miftahurrohmah, H. Kuswanto, D. S. Pambudi, F. Fauzi, and F. Atmaja, “Assessment of the Support Vector Regression and Random Forest Algorithms in the Bias Correction Process on Temperatures,” in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 637–644. doi: 10.1016/j.procs.2024.03.049.

[12] R. Huang, C. McMahan, B. Herrin, A. McLain, B. Cai, and S. Self, “Gradient boosting: A computationally efficient alternative to Markov chain Monte Carlo sampling for fitting large Bayesian spatio-temporal binomial regression models,” *Infect Dis Model*, vol. 10, no. 1, pp. 189–200, Mar. 2025, doi: 10.1016/j.idm.2024.09.008.

[13] L. W. Rizkallah, “Enhancing the performance of gradient boosting trees on regression problems,” *J Big Data*, vol. 12, no. 1, Dec. 2025, doi: 10.1186/s40537-025-01071-3.

[14] H. A. Elsayed *et al.*, “Design and Performance Prediction of a Multilayer Metamaterial Absorber for Broadband Solar-Thermal Energy Conversion Using Random Forest Regression,” *Case Studies in Thermal Engineering*, p. 106615, Jun. 2025, doi: 10.1016/j.csite.2025.106615.

[15] S. Chowdhury, A. K. Saha, and D. K. Das, “Hydroelectric Power Potentiality Analysis for the Future Aspect of Trends with R2 Score Estimation by XGBoost and Random Forest Regressor Time Series Models,” in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 450–456. doi: 10.1016/j.procs.2025.01.004.

[16] S. Jafari, J. H. Yang, and Y. C. Byun, “Optimized XGBoost modeling for accurate battery capacity degradation prediction,” *Results in Engineering*, vol. 24, Dec. 2024, doi: 10.1016/j.rineng.2024.102786.

[17] R. R. Aisy, L. Zulfa, Y. Rahim, and M. Ahsan, “Residual XGBoost regression—Based individual moving range control chart for Gross Domestic Product growth monitoring,” *PLoS One*, vol. 20, no. 5 May, May 2025, doi: 10.1371/journal.pone.0321660.

[18] F. Özen, “Random forest regression for prediction of Covid-19 daily cases and deaths in Turkey,” *Heliyon*, vol. 10, no. 4, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25746.

[19] M. Balzer, E. Bergherr, S. Hutter, and T. Hepp, “Gradient boosting for Dirichlet regression models,”

AStA Advances in Statistical Analysis, 2025, doi: 10.1007/s10182-025-00526-5.

[20] Y. R. Shahare, M. P. Singh, S. P. Singh, P. Singh, and M. Diwakar, "ASUR: Agriculture Soil Fertility Assessment Using Random Forest Classifier and Regressor," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 1732–1741. doi: 10.1016/j.procs.2024.04.164.

[21] J. C. M. Sánchez, H. G. A. Mesa, A. T. Espinosa, S. R. Castilla, and F. G. Lamont, "Improving wheat yield prediction through variable selection using Support Vector Regression, Random Forest, and Extreme Gradient Boosting," Mar. 01, 2025, *Elsevier B.V.* doi: 10.1016/j.atech.2025.100791.

[22] M. Wahba, R. Essam, M. El-Rawy, N. Al-Arifi, F. Abdalla, and W. M. Elsadek, "Forecasting of flash flood susceptibility mapping using random forest regression model and geographic information systems," *Heliyon*, vol. 10, no. 13, Jul. 2024, doi: 10.1016/j.heliyon.2024.e33982.

[23] R. S. Zemel and T. Pitassi, "A Gradient-Based Boosting Algorithm for Regression Problems."

[24] J. Velthoen, C. Dombry, J. J. Cai, and S. Engelke, "Gradient boosting for extreme quantile regression," *Extremes (Boston)*, vol. 26, no. 4, pp. 639–667, Dec. 2023, doi: 10.1007/s10687-023-00473-x.

[25] H. T. Wen, H. Y. Wu, and K. C. Liao, "Using XGBoost Regression to Analyze the Importance of Input Features Applied to an Artificial Intelligence Model for the Biomass Gasification System," *Inventions*, vol. 7, no. 4, Dec. 2022, doi: 10.3390/inventions7040126.

[26] S. Hakkal and A. A. Lahcen, "XGBoost To Enhance Learner Performance Prediction," *Computers and Education: Artificial Intelligence*, vol. 7, Dec. 2024, doi: 10.1016/j.caeai.2024.100254.

[27] V. Verma, "Exploring Key XGBoost Hyperparameters: A Study on Optimal Search Spaces and Practical Recommendations for Regression and Classification," *International Journal of All Research Education and Scientific Methods*, vol. 12, no. 10, pp. 3259–3266, 2024, doi: 10.56025/ijaresm.2024.1210243259.

[28] P. Patil, D. Pawar, S. Shete, S. Tote, and H. Rathod, "XGBoost Algorithm and Its Comparative Analysis," 2022. [Online]. Available: www.ijnrd.org

[29] Z. Arif Ali, Z. H. Abduljabbar, H. A. Tahir, A. Bibo Sallow, and S. M. Almufti, "eXtreme Gradient Boosting Algorithm with Machine Learning: a Review," *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 320–334, May 2023, doi: 10.25007/ajnu.v12n2a1612.

[30] M. D. Guillen, J. Aparicio, and M. Esteve, "Gradient tree boosting and the estimation of production frontiers," *Expert Syst Appl*, vol. 214, Mar. 2023, doi: 10.1016/j.eswa.2022.119134.

[31] A. Panda, R. Datar, S. Deshpande, and G. Bacher, "Enhancing pH prediction accuracy in Al2O3 gated ISFET using XGBoost regressor and stacking ensemble learning," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-04530-2.