E-ISSN : 2988-1986
https://ejournal.warunayama.org/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 8 No 7 Tahun 2025

PERBANDINGAN METODE *RANDOM FOREST* DAN *XGBOOST* DALAM PREDIKSI HARGA JUAL RUMAH DI WILAYAH JABODETABEK

Harun Arrasyid[1], Foni Agus Setiawan[2], Freza Riana[3]
[1,2,3]Teknik Informatika, Universitas Ibn Khaldun Bogor
Jalan Sholeh Iskandar, Kota Bogor, Indonesia
[1]harunarrasyid0600@gmail.com

Abstrak

Wilayah Jabodetabek, yang mencakup Jakarta, Bogor, Depok, Tangerang, dan Bekasi, merupakan kawasan metropolitan terbesar di Indonesia dengan pertumbuhan populasi dan infrastruktur yang pesat, menyebabkan permintaan akan properti, khususnya rumah, terus meningkat. Penelitian ini membandingkan dua algoritma yang berbasis *ensemble learning*, *Random Forest* dan *XGBoost*, untuk memprediksi harga jual rumah di wilayah Jabodetabek. Tujuan penelitian adalah membandingkan hasil prediksi berdasarkan nilai *error* antara kedua metode tersebut. *Dataset* yang digunakan berasal dari situs *Kaggle* yang berfokus pada data penjualan rumah di wilayah Jabodetabek. Variabel yang digunakan dalam penelitian ini adalah variabel yang memiliki korelasi *pearson* lebih dari sama dengan 0.5 terhadap variabel target atau harga rumah dari semua variabel yang ada. Kinerja kedua model dievaluasi berdasarkan *error metrics* RMSE, MAE, dan MAPE. Hasil penelitian menunjukkan bahwa *Random Forest* menghasilkan prediksi harga jual rumah yang lebih baik dengan nilai RMSE 0.33, MAE 0.20, dan MAPE 0.91%. Sementara itu, *XGBoost*, setelah *tuning* parameter, menunjukkan penurunan nilai *error* dengan RMSE 0.30, MAE 0.18, dan MAPE 0.84%. *Random Forest* lebih mudah diterapkan tanpa perlu *tuning parameter* yang kompleks, sementara *XGBoost* membutuhkan perhatian lebih pada *tuning* parameter untuk mengoptimalkan performa terbaiknya.

Kata kunci: Prediksi Harga Jual Rumah, Jabodetabek, Korelasi *Pearson*, *Random Forest*, *XGBoost*, *Error Metrics*.

Article History:
Received: June 2025
Reviewed: June 2025
Published: June 2025

Plagirism Checker No 234
Prefix DOI:
10.8734/Kohesi.v1i2.365
Copyright: Author
Publish by: Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

1. PENDAHULUAN

Jabodetabek, yang mencakup Jakarta, Bogor, Depok, Tangerang, dan Bekasi, merupakan kawasan metropolitan terbesar di Indonesia yang mengalami pertumbuhan pesat dan infrastruktur yang berkembang [1]. Populasi Jabodetabek diperkirakan mencapai 30 juta pada tahun 2020 [2], menjadikan kawasan ini merupakan daerah dengan tingkat kepadatan penduduk tertinggi di Indonesia dan ketiga di dunia [3]. Pasar properti di wilayah Jabodetabek selalu menarik perhatian investor dan pembeli properti karena pertumbuhan jumlah penduduk yang terus meningkat, yang berdampak pada peningkatan permintaan akan rumah [4].

Rumah sebagai struktur tempat tinggal wajib memenuhi standar kenyamanan, keamanan, dan kesehatan. Tujuan utamanya adalah untuk mendukung penghuninya agar dapat beraktivitas dengan produktif dan menjadikan rumah sebagai lingkungan tinggal yang sehat dan aman bagi para penghuninya [5]. Calon pembeli rumah biasanya memiliki beberapa pertimbangan sebelum memutuskan untuk membeli. Pertimbangan tersebut meliputi kondisi bangunan, luas tanah, luas bangunan, dan fasilitas [6].

Menyadari pentingnya kriteria-kriteria ini dalam keputusan pembelian rumah, pengembang properti berusaha untuk mengembangkan sistem prediksi harga jual rumah yang dapat memberikan perkiraan yang akurat dan optimal. *Developer* perlu menggunakan model algoritma yang sesuai dengan karakteristik *dataset* untuk mendapatkan nilai *error* atau galat yang cukup baik. Faktor-faktor yang dapat mempengaruhi nilai galat dalam melakukan prediksi dapat berupa kualitas data yang digunakan, metode yang digunakan, proses seleksi data dan terjadinya *overfitting* dan *underfitting* [7]. Oleh karena itu dibutuhkan penggunaan metode yang dapat menangani bagaimana karakteristik *dataset*.

Penelitian perbandingan algoritma yang diimplementasikan pada prediksi harga jual rumah ini, menggunakan algoritma *Random Forest* dan *XGBoost*. Hal ini didasari oleh penelitian tentang '*A comparative performance assessment of ensemble learning for credit scoring*' [8].

Random Forest dan *XGBoost* telah diakui sebagai sebuah estimator canggih dengan kinerja yang sangat tinggi baik dalam klasifikasi maupun regresi. Mampu mencegah *overfitting*, hasil prediksi yang relatif tinggi terhadap data yang hilang dan data yang tidak seimbang merupakan kemampuan kedua algoritma ini.

Random Forest merupakan metode klasifikasi dan prediksi yang dikembangkan dari metode CART (*Classification and Regression Trees*). Metode ini menggunakan teknik *bootstrap aggregating* (*bagging*) yang berbasis *ensemble learning* dan *random feature selection* untuk meningkatkan akurasi dan stabilitas prediksi [9]. *eXtreme Gradient Boosting* (*XGBoost*) merupakan algoritma pengembangan dari *gradient tree boosting* yang berbasis algoritma *ensemble learning*, secara efektif bisa menanggulangi kasus pembelajaran mesin yang berskala besar [10].

Kedua metode merupakan termasuk dalam *ensemble learning* yang membedakan antara kedua metode adalah pendekatan algoritma. *Random Forest* menggunakan *bagging* dan *XGBoost* menggunakan *boosting*. Cara kedua model juga memiliki perbedaan dalam penanganan *overfitting* [11]. Penelitian lainnya tentang analisis perbandingan metode untuk prediksi harga rumah menunjukan *Random Forest* dan *XGBoost* sangat baik dalam melakukan prediksi dengan nilai akurasi tertinggi mencapai 80% [12].

Penelitian "Perbandingan Metode *Random Forest* dan *XGBoost* dalam Prediksi Harga Jual Rumah di Wilayah Jabodetabek" ini menggunakan *dataset* yang bersumber dari situs *Kaggle*. *Kaggle* merupakan situs berbagi ide, mendapatkan inspirasi, bersaing dengan data *scientist* lain, mendapatkan informasi baru termasuk *dataset* dan teknik *coding*, dan melihat berbagai aplikasi data *science* di dunia nyata [13]. *Dataset* yang digunakan dalam penelitian ini terfokus pada data penjualan rumah di wilayah Jabodetabek.

2. TINJAUAN PUSTAKA

2.1. Rumah

Rumah tidak hanya sekadar bangunan, tetapi juga memiliki makna dan nilai yang lebih dari sekadar tempat berlindung. Tempat ini menjadi tempat di mana kehidupan sehari-hari dijalani, interaksi sosial dengan orang-orang terkasih dilakukan, dan kenangan yang tak terlupakan dibangun.

Kenyamanan dan keamanan dirasakan di rumah. Di dalam rumah, seseorang bisa menjadi diri sendiri tanpa perlu khawatir dihakimi atau ditolak. Kebahagiaan dan kesedihan dibagikan dengan orang-orang terkasih di rumah [14].

2.2. Korelasi *Pearson*

Analisis korelasi sering digunakan dalam mengukur kekuatan hubungan antara dua variabel. Dua variabel dikatakan berkorelasi jika perubahan dalam salah satu variabel biasanya disertai dengan perubahan dalam variabel lainnya dalam pola yang linier, baik positif (arah yang sama) maupun negatif (arah yang berlawanan). Kekuatan hubungan diukur dengan koefisien korelasi, yang menunjukkan seberapa erat hubungan antara variabel tersebut [15].

Tabel 1. Kriteria Korelasi *Pearson*

No	Nilai r	Interpretasi
1	0	Tidak Berkorelasi
2	≤ 0.20	Sangat Lemah
3	0.21 - 0.40	Lemah
4	0.41 - 0.70	Cukup Erat
5	0.71 – 0.90	Erat
6	0.91 – 0.99	Sangat Erat
7	1	Sempurna

Dalam statistik, korelasi *pearson* banyak digunakan untuk mengukur tingkat hubungan antara dua variabel yang berkaitan secara linier. Dalam pasar properti, misalnya, jika kita ingin mengetahui seberapa terkait dua komoditas tertentu satu sama lain, korelasi *Pearson* digunakan untuk melakukannya. Koefisien korelasi *Pearson* dihitung menggunakan rumus berikut [15] :

$$r = \frac{\sum_{i=1}^{n}(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \underline{x})^2 \sum_{i=1}^{n}(y_i - \underline{y})^2}} \quad (1)$$

- Keterangan:
  1. *n* adalah jumlah pasangan data yang diamati.
  2. *r* adalah koefisien korelasi *Pearson*
  3. x_i adalah nilai dari variabel pertama pada pasangan data ke-i.
  4. y_i adalah nilai dari variabel kedua pada pasangan data ke-i.
  5. $\hat{x}$ adalah rata-rata dari semua nilai variabel pertama.
  6. $\hat{y}$adalah rata-rata dari semua nilai variabel kedua.

2.3. Metode *Z-Score*

Metode *Z-Score* merupakan salah satu metode yang digunakan untuk menentukan batas *outlier* berdasarkan standar deviasi antara nilai observasi dan rata-rata [1]. Setelah batas *outlier* ditentukan, nilai yang melebihi batas tersebut akan dihapus secara permanen dari data. ±3 adalah nilai ambang batas yang paling sering digunakan untuk deteksi pencilan. Hal ini mengindikasikan data dianggap sebagai pencilan jika nilai *Z-score* nya lebih tinggi dari ±3. Dalam metode ini, nilai *Z-score* dibangun dengan rumus [16]:

$$Z = \frac{(X - \mu)}{\sigma} \quad (2)$$

- Keterangan :
  1. Z = *Z-score*
  2. X = Nilai yang diamati
  3. μ = Rata-rata
  4. σ = Standar Deviasi

2.4. *Ensemble Learning*

Pembelajaran ansambel adalah teknik pembelajaran algoritma yang dibangun dari berbagai model pengklasifikasian atau pemrediksi, ini digunakan untuk mengklasifikasi data baru berdasarkan bobot prediksi yang telah dibuat sebelumnya.

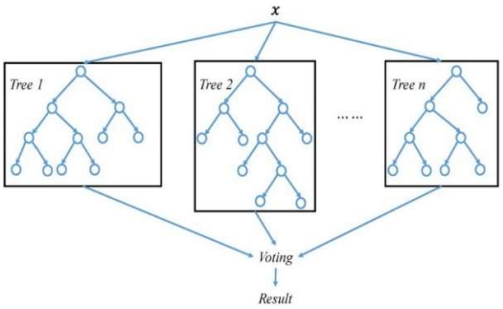
Ensemble learning umumnya dapat dibagi menjadi homogen dan heterogen. Yang pertama menggabungkan model dari jenis yang sama, sedangkan yang terakhir menggabungkan model dari jenis yang berbeda. *Boosting* yang diwakili oleh *XGBoost* dan *Bagging* yang diwakili

oleh *Random Forest* termasuk dalam integrasi homogen dan *Stacking* termasuk dalam integrasi heterogen dalam *ensemble learning* [8].

2.5. *Random Forest*

Random Forest merupakan metode klasifikasi dan prediksi yang dikembangkan dari metode CART (*Classification and Regression Trees*). Metode ini menggunakan teknik *bootstrap aggregating* (*bagging*) dan *random feature selection* untuk meningkatkan akurasi dan stabilitas prediksi [9].

Metode ini digunakan untuk membangun pohon keputusan yang terdiri dari *root node*, *internal node*, dan *leaf node* dengan mengambil atribut dan data secara acak sesuai ketentuan yang diberlakukan. *Root node* merupakan simpul yang terletak paling atas, atau biasa disebut sebagai akar dari pohon keputusan. *Internal node* adalah simpul percabangan, dimana *node* ini mempunyai *output* minimal dua dan hanya ada satu input. *Leaf node* atau *terminal node* merupakan simpul terakhir yang hanya memiliki satu input dan tidak mempunyai *output*. Pohon keputusan dimulai dengan cara menghitung nilai *entropy* sebagai penentu tingkat ketidakmurnian atribut dan nilai *information gain*. Arsitektur *Random Forest* dapat dilihat pada Gambar 1 [17]:



Gambar 1. Arsitektur *Random Forest*

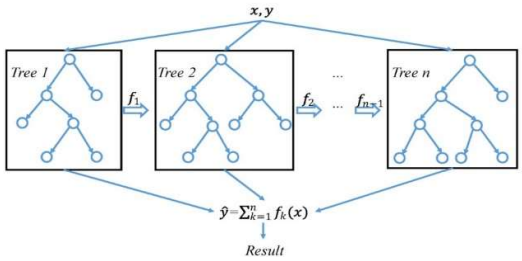
Random Forest dalam prosesnya memerlukan beberapa parameter sebagai acuan diantaranya adalah sebagai berikut [18]:

a. *N_estimators* : Jumlah pohon dalam hutan.
b. *Max_depth* : Kedalaman maksimum pohon; jika tidak ditentukan, pemekaran *node* dilakukan hingga daun murni atau berisi sampel kurang dari jumlah minimum.
c. *Min_samples_split* : Jumlah minimum sampel untuk membagi sebuah *node*.
d. *Min_samples_leaf* : Jumlah minimum sampel dalam setiap daun.
e. *Max_features* : Jumlah maksimum fitur yang digunakan untuk membangun setiap pohon.

2.6. *eXtreme Gradient Boosting*

eXtreme Gradient Boosting (*XGBoost*) merupakan adalah algoritma pengembangan dari *gradient tree boosting* yang berbasis algoritma *ensemble*, secara efektif bisa menanggulangi kasus pembelajaran mesin yang berskala besar [10]. *XGBoost* membangun model baru untuk memprediksi *error* dari model sebelumnya digunakan dalam metode *boosting*. *Gradient descent* untuk memperkecil *error* saat membuat model baru, algoritma tersebut dinamakan *gradient boosting* [18].

Arsitektur *XGBoost* dapat dilihat pada Gambar 2. [17]



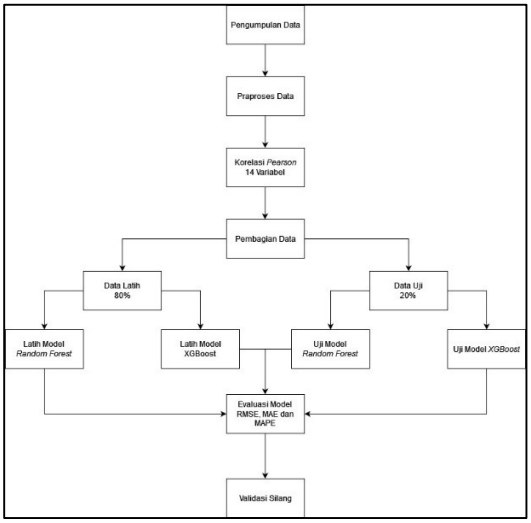
Gambar 2. Arsitektur *XGBoost*

XGBoost dalam prosesnya memerlukan beberapa parameter sebagai acuan diantaranya adalah sebagai berikut [18]:

a. *N_estimators* : Jumlah pohon dalam hutan.
b. *Colsample_bytree* : Proporsi kolom yang digunakan per pohon (*default* 1; *range*: 0–1).
c. Eta (*learning_rate*) : *Learning rate* untuk mencegah *overfitting* (*range*: 0–1).
d. *Max_depth* : Kedalaman *maksimum* pohon (*default* 6; *range*: 0 hingga tak terhingga).
e. *Subsample* : Proporsi baris data yang digunakan per pohon (*default* 1; *range*: 0–1).
f. *Alpha* : Regulasi L1 untuk mengurangi *overfitting* (*range*: 0–10).
g. *Lambda* : Regulasi L2 untuk mengurangi *overfitting* (*range*: 0–10).

3. METODE PENELITIAN

Tahapan Penelitian ini dapat dilihat pada Gambar 3.



Gambar 3. Metodologi Penelitian

3.1 Pengumpulan Data

Tahap awal penelitian ini pengumpulan data daftar harga rumah di wilayah Jabodetabek bersumber dari situs *Kaggle* [30]. Data yang diperoleh memiliki *shape* (3553,14). Data ini meliputi daftar rumah yang dijual di wilayah Jabodetabek yang bersumber dari *marketplace rumah123.com* dan memiliki update terakhir pada tahun 2022. Data penjualan rumah di wilayah Jabodetabek dapat dilihat pada Gambar 4.

harga	kota	kamar_tidur	mar_man	uas_tanah	s_bangun	mpat_par	kelistrikan	tidur_per	mandi_pe	lantai	ondisi_prop	garasi	perabotan
2.99E+09	Bekasi	4	4	239	272	0	4400 mah	0	1	2	bagus	0	unfurnished
1.27E+09	Bekasi	3	2	55	69	1	2200 mah	0	0	2	bagus	0	
1.95E+09	Bekasi	3	3	119	131	1	2200 mah	1	1	2	bagus	1	unfurnished
3.3E+09	Bekasi	3	3	180	174	0	3500 mah	1	1	2	bagus sek	2	unfurnished
4.5E+09	Bekasi	4	3	328	196	2	3500 mah	1	1	2	bagus	1	unfurnished
2.7E+09	Bekasi	3	3	136	200	2	3500 mah	1	1	2	bagus	1	semi furnished
2.35E+09	Bekasi	2	2	144	144	1	4400 mah	0	0	2		1	
4.5E+09	Bekasi	4	4	216	250	2	3500 mah	1	1	2		1	
2.9E+09	Bekasi		3	200	152	2	4400 mah	3	1	2		0	semi furnished
2.7E+09	Bekasi	3	3	136	200	1	3500 mah	1	1	2		1	semi furnished
2.55E+09	Bekasi	3	2	144	186	1	2200 mah	1	0	2		0	
9.52E+08	Bekasi	2	2	55	50	1	2200 mah	0	0	2	baru	0	unfurnished
1.8E+09	Bekasi	2	1	119	82	1	2200 mah	0	0	2	bagus sek	1	semi furnished
2.35E+09	Bekasi	3	3	144	149	1	2200 mah	1	1	2	bagus	1	unfurnished
2.5E+09	Bekasi	3	3	144	200	1	2200 mah	0	0	2	sudah ren	1	unfurnished
2.7E+09	Bekasi	3	3	193	136	1	3500 mah	1	1	3	bagus	1	unfurnished
5.36E+09	Bekasi	5	4	192	370	2	3500 mah	1	1	2	baru	0	unfurnished
9.52E+08	Bekasi	2	2	55	50	1	2200 mah	1	1	2	baru	0	unfurnished
2.25E+09	Bekasi	4	3	119	150	0	2200 mah	0	0	2	bagus	0	unfurnished
2.25E+09	Bekasi	3	2	119	150	1	2200 mah	1	1	1	bagus	1	semi furnished
2.45E+09	Bekasi	3	3	144	150	1	2200 mah	1	1	1	bagus	0	unfurnished
1.25E+08	Bekasi	3	1	110	110	0	1300 mah	0	0	2	baru	0	unfurnished

Gambar 4. Tampilan *Dataset* pada *Excel*

3.2 Praproses Data

Setelah data diperoleh, langkah selanjutnya sebelum data tersebut digunakan untuk melatih model adalah melakukan tahap praproses data. Praproses data ini penting untuk memastikan data yang digunakan dalam pelatihan model berkualitas dan siap untuk analisis lebih lanjut. Berikut adalah langkah-langkah yang dilakukan dalam tahap praproses data:

3.2.1 Mengatasi nilai hilang atau *Null* dan *NaN*

Mengatasi data *Null* dan *NaN* (*Not A Number*) merupakan langkah penting dalam analisis data dan pengolahan data. Berikut adalah beberapa cara untuk mengatasi nilai hilang atau *Null* dan *NaN* yang dilakukan pada penelitian ini.

1) *Drop cells with less NaN value*

   Apabila data tersebut memiliki sedikit jumlah *Null* dan *NaN* maka dilakukan *drop cells with less NaN value*.

2) Mengganti Nilai Hilang

   Cara kedua adalah dengan mengganti nilai hilang dengan nilai yang di peroleh dari analisis data. Mengganti nilai hilang dengan nilai yang didapat dari analisis data adalah mengganti nilai hilang dengan rata-rata atau nilai modus yang dimiliki oleh setiap variabel masing-masing.

3) Mengisi nilai hilang dengan *Interpolate*

   Cara ketiga yaitu mengisi data *missing values* dari nilai sebelum dan sesudahnya. Pendekatan ini membantu dalam mengisi kekosongan data dengan berdasarkan konteks sekitarnya

3.2.2 Pengkodean Variabel Data Kategorial

Pengkodean Data Kategorial dilakukan untuk mengubah variabel kategorial menjadi bentuk yang dapat dipahami oleh model pembelajaran mesin. Tahap ini penting dilakukan agar data kategorial dapat dipahami oleh model pembelajaran mesin.

3.2.3 Analisis Data *Outlier* dan *Interpolate*

Pada tahap ini dilakukan analisis data *outlier* untuk mengidentifikasi data-data yang menyimpang dari distribusi normal *dataset* dengan menggunakan metode *Z-score*. Setelah mengidentifikasi dan menghapus data *outlier* akan terdapat *cell* dengan *missing values*.

Untuk mengatasi hal tersebut maka dilakukan teknik *interpolate* yaitu mengisi data *missing values* dari nilai sebelum dan sesudahnya. Pendekatan ini membantu dalam mengisi kekosongan data dengan berdasarkan konteks sekitarnya.

3.2.4 Normalisasi Data

Normalisasi dilakukan agar menghasilkan prediksi yang lebih akurat dan efisien. Normalisasi data pada penelitian ini menggunakan teknik *min-max scaling* pada *dataset* yang telah disiapkan.

3.2.5 Distribusi Variabel Target

Tahap ini dilakukan pada variabel target yang di ubah menjadi nilai logaritmik dari nilai asli. Ini bertujuan untuk mengubah distribusi data agar lebih merata, mudah dibaca dan menghasilkan nilai prediksi yang lebih akurat.

3.3 Uji Korelasi *Pearson*

Uji korelasi *Pearson* merupakan proses menyeleksi atau memilih variabel yang memiliki korelasi lebih dari sama dengan 0.5 pada variabel target. Metode ini digunakan untuk mengukur tingkat hubungan linier antara setiap fitur dengan target. Melalui analisis korelasi *Pearson*, dapat diidentifikasi fitur-fitur yang memiliki korelasi yang kuat atau signifikan dengan variabel target.

3.4 Pembagian Data

Penelitian ini mengalokasikan data menjadi dua jenis, yaitu data *training* atau data latih, dan data *testing* atau data uji. Data tersebut dibagi menggunakan perbandingan 80% : 20%, dimana 80% dari total data digunakan sebagai data latih, sedangkan 20% sisanya digunakan sebagai data uji. Pembagian data dapat dilihat pada Tabel 2.

Tabel 1. Pembagian Data

Pembagian Data	Persentase Data
Latih	80%
Uji	20%
Total	100%

3.5 Pemodelan

Tahap ini dilakukan pembuatan kedua model *Random Forest* dan *XGBoost*. Pada tahap ini pembuatan model *Random Forest* menggunakan *n_estimators* 100, *random_state* 42, dan *max_depth* 100. Untuk pembuatan model *XGBoost* menggunakan *objective reg:squarederror*, *colsample_bytree* 0.3, *learning_rate* 0.1, *max_depth* 100, *alpha* 10 dan n_estimators 100.

3.6 Latih dan Uji model

Setelah model berhasil di buat, Model *Random Forest* dan *XGBoost* akan Latih dan Uji menggunakan data latih yang telah dipersiapkan yaitu sebanyak 80% dan data uji 20% dari keseluruhan data pada tahap pembagian data.

3.7 Evaluasi Model

Pada tahap evaluasi model, hasil dari kedua model yang telah di tes menggunakan data uji sebelumnya, yaitu model *Random Forest* dan *XGBoost* akan dibandingkan. Pada tahap evaluasi ini peneliti menggunakan galat *metrics* seperti RSME, MAE, dan MAPE.

3.8 Validasi *K-Fold*

Pada tahap ini dilakukan validasi terhadap model yang memiliki nilai galat terendah menggunakan *cross validation K-Fold* dengan membuat nilai lipatan K atau *n_splits* 10 dan *shuffle True*. Hal ini membantu untuk memastikan bahwa model tidak *overfitting* atau *underfitting* dan cukup baik dalam melakukan prediksi.

4. HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Data yang dikumpulkan merupakan data rumah dijual yang terdaftar di wilayah Jabodetabek dengan sumber informasi dari *marketplace rumah123.com* dan data yang sudah dikumpulkan dan dijadikan *dataset* oleh *developer* tersebut tersedia di situs *Kaggle*.

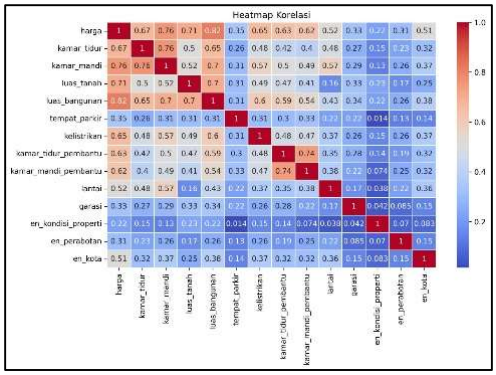
4.2 Praproses Data

Pada tahap praproses data atau tahap untuk mempersiapkan data sebelum data diproses. Pada proses ini data akan dianalisa untuk melihat adanya nilai hilang dan divisualisasikan agar mudah dipahami sehingga dapat melakukan antisipasi atau mengatasi data nilai hilang.

Praproses data ini penting untuk memastikan data yang digunakan dalam pelatihan model berkualitas dan siap untuk analisis lebih lanjut. Berikut adalah langkah-langkah yang dilakukan dalam tahap praproses data.

4.3 Uji Korelasi *Pearson*

Tahap selanjutnya dalam penelitian ini adalah melakukan uji korelasi *Pearson*. Hal ini dilakukan untuk menyeleksi variabel-variabel yang memiliki korelasi diatas 0.5 atau memiliki korelasi sedang pada variabel target. Tujuannya agar menaikkan tingkat akurasi dalam memprediksi data variabel target. *Heatmap* korelasi dapat dilihat pada Gambar 5.



Gambar 5. *Heatmap* korelasi

Pada Gambar 5, variabel yang memiliki korelasi dengan fitur target atau variabel target (harga). Selanjutnya pemilihan fitur-fitur yang memiliki korelasi diatas 0.5 atau sedang dengan variabel target. Hasil pemilihan atau penyeleksian fitur dengan korelasi diatas 0.5.

4.4 Pembagian Data

Tahap pembagian data penelitian ini mengalokasikan data menjadi dua jenis, yaitu data *train* atau data latih, dan data *testing* atau data uji. Data tersebut dibagi menggunakan perbandingan 80% : 20%, dimana 80% dari total data digunakan sebagai data latih, sedangkan 20% sisanya digunakan sebagai data uji. Spesifik pembagian data dapat dilihat pada Tabel 3.

Tabel 3. Spesifikasi pembagian data

Pembagian Data	Persentase Data	Jumlah Data
Latih	80%	2808
Uji	20%	702
Jumlah	100%	3510

4.5 Pemodelan

Tahap pemodelan adalah tahap pembuatan atau pembangunan model pembelajaran mesin. Penelitian ini menggunakan dua model pembelajaran mesin dengan teknik *ensemble learning*.

4.5.1 *Random Forest*

Pada percobaan ini, digunakan *hyperparameter* yang akan diuji adalah *n_estimators, max_depth, min_sample_split, min_sample_leaf* dan *max_features*. *max_depth* merupakan kedalaman maksimal pohon keputusan yang diperbolehkan. *n_estimators* jumlah *decision tree* yang dibuat. *min_sample_split* merupakan jumlah minimum sampel yang dibutuhkan untuk membagi suatu *node* dalam suatu pohon keputusan. *min_sample_leaf* merupakan jumlah minimum sampel yang dibutuhkan dalam suatu *leaf node* dalam pohon keputusan. *max_features* merupakan jumlah maksimum fitur yang digunakan dalam membangun setiap pohon keputusan. Kombinasi *hyperparameter* dapat dilihat pada Tabel 4. dan Tabel 5.

Tabel 4. Parameter *Random Forest* (*Default*)

Nama parameter	Nilai parameter
max_depth	*None*
n_estimators	100
min_sample_split	2
min_sample_leaf	1
max_features	*Auto*

Tabel 5. Parameter *Random Forest (Tuning)*

Nama parameter	Nilai parameter
max_depth	50
n_estimators	100
min_sample_split	2
min_sample_leaf	1
max_features	7

4.5.2 *eXtreme Gradient Boosting* (*XGBoost*)

Pada percobaan ini, digunakan *hyperparameter* yang akan diuji adalah *n_estimators*, *max_depth*, *colsample_bytree*, *learning_rate*, *subsample alpha* dan *lambda*. *max_depth* merupakan kedalaman maksimal *decision tree* yang diperbolehkan, *n_estimators* adalah jumlah pohon keputusan yang dibuat. *colsample_bytree* merupakan parameter untuk memilih banyak *sample* kolom yang akan digunakan. *alpha* dan *lambda* digunakan untuk mengontrol regularisasi yang bertujuan mencegah *overfitting*. *learning rate* berfungsi untuk mencegah model mengalami *overfitting* dengan *range* parameter dari 0 sampai 1.

subsample parameter untuk memilih banyak *sample* baris data yang akan digunakan, *default* 1 yang berarti keseluruhan baris data. *range* dari 0 sampai 1. Kombinasi *hyperparameter* dapat dilihat pada Tabel 6 dan Tabel 7.

Tabel 6. Parameter *XGBoost* (*Default*)

Nama parameter	Nilai parameter
max_depth	100
n_estimators	6
colsample_bytree	1
learning_rate	0.3
subsample	1
alpha	0
lambda	1

Tabel 7. Parameter *XGBoost* (*Tuning*)

Nama parameter	Nilai parameter
max_depth	100
n_estimators	50
colsample_bytree	0.7
learning_rate	0.1
subsample	0.6
alpha	0.01
lambda	0.1

4.6 Latih Model

Setelah model berhasil di buat dengan parameter masing-masing model yang telah ditentukan, Model *Random Forest* dan *XGBoost* dilakukan *train* dan *test* menggunakan data *train* yang telah dipersiapkan yaitu sebanyak 80% dan setelah dilatih kemudian melakukan validasi silang model menggunakan *k-fold* dengan membuat nilai lipatan K atau *n_splits* 10 dan *shuffle True*.

Hal ini membantu untuk memaksimalkan pemanfaatan data dan identifikasi variabilitas sehingga dapat membantu kinerja model apakah mengalami *overfitting* atau *underfitting*.

4.7 Validasi Silang *K-Fold*

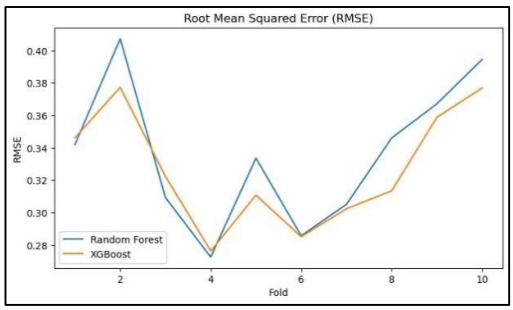
Dalam tahap ini dilakukan validasi silang menggunakan *K-Fold*, tahap ini adalah melakukan validasi terhadap model untuk memaksimalkan pemanfaatan data dan identifikasi variabilitas sehingga dapat membantu kinerja model apakah mengalami *overfitting* atau *underfitting*.

Kedua keadaan tersebut terjadi apabila model gagal dalam memahami dan membaca data sebelumnya sehingga terjadinya perubahan pada nilai evaluasi model. *K-Fold* menggunakan lipatan atau iterasi terhadap k atau berapa kali lipatan yang ditentukan dan akan mengambil rata-rata nilai dari evaluasi model. Pada penelitian dilakukan lipatan berjumlah 10.

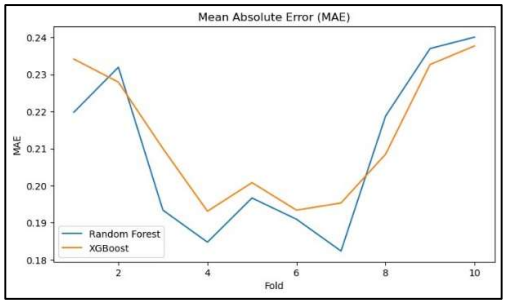
Dalam validasi silang menggunakan tiga evaluasi untuk mengetahui apakah model mengalami *overfitting* atau *underfitting*. Ketiga cara itu adalah RMSE, yang mengukur rata-rata selisih antara prediksi dan data aktual dengan memperhitungkan kuadratnya. MAPE, yang mengukur persentase rata-rata kesalahan prediksi dibandingkan dengan nilai aktual. Serta MAE, yang mengukur rata-rata dari selisih absolut antara prediksi dan data aktual

4.7.1 Perbandingan Model *Default*

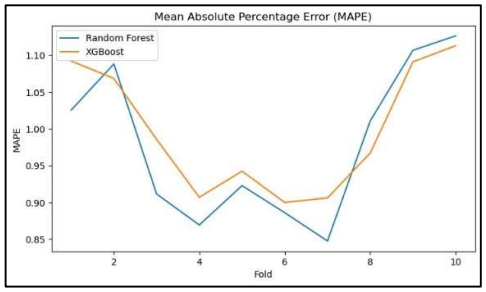
Visualisasi perbandingan hasil validasi silang per *fold* kedua model *set default* dapat di lihat pada Gambar 6, Gambar 7 dan Gambar 8.



Gambar 6. Perbandingan RMSE hasil validasi silang model *default*

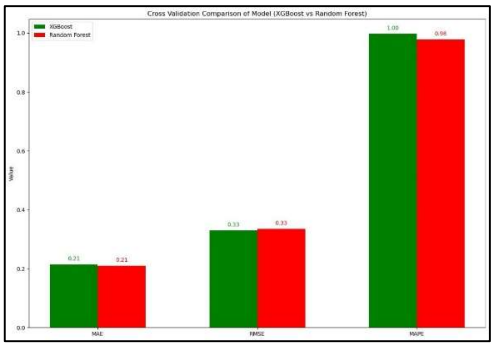


Gambar 7. Perbandingan MAE hasil validasi silang model *default*



Gambar 8. Perbandingan MAPE hasil validasi silang model *default*

Visualisasi perbandingan hasil rata-rata validasi silang kedua model *default* dapat dilihat pada Gambar 9.

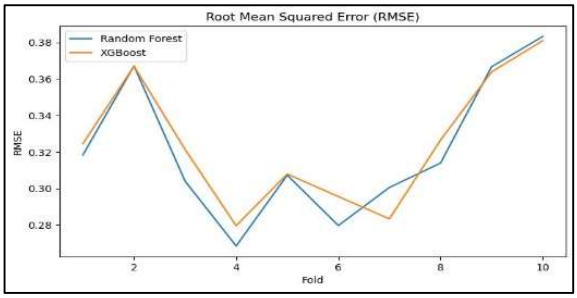


Gambar 9. Perbandingan rata-rata hasil validasi silang kedua model *default*

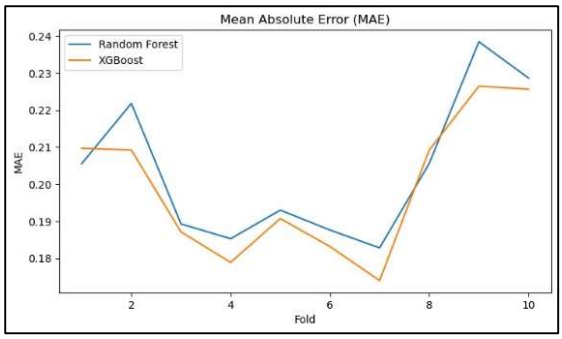
Dapat lihat dari Gambar 9, bahwa hasil dari validasi silang kedua model *default* adalah model *default Random Forest* memiliki nilai *error* pada MAPE sedikit lebih kecil atau dapat dikatakan lebih baik dibandingkan model *default XGBoost*, namun secara MAE dan RSME kedua model memiliki nilai yang sama. Pada *Random Forest* terdapat variasi yang lebih besar dalam RMSE, MAE, dan MAPE di antara *fold* (misalnya, RMSE tertinggi adalah 0.44 pada *fold* 7). Namun pada model *XGBoost* kinerja lebih konsisten dengan nilai RMSE yang tidak terlalu jauh dari rata-rata, meskipun terdapat juga variasi.

4.7.2 Perbandingan Model *Tuning Hyperparemeter*

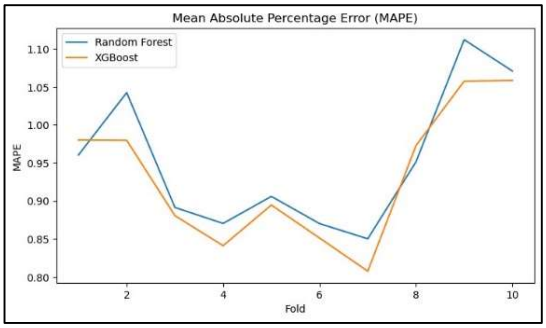
Visualisasi perbandingan hasil validasi silang per *fold* kedua model *set tuning hyperparameter* dapat dilihat pada Gambar 10, Gambar 11 dan Gambar 12.



Gambar 10. Perbandingan RMSE hasil validasi silang model *tuning hyperparameter*

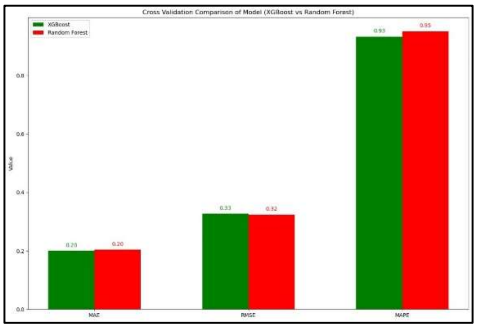


Gambar 11. Perbandingan MAE hasil validasi silang model *tuning hyperparameter*



Gambar 12. Perbandingan MAPE hasil validasi silang model *tuning hyperparameter*

Visualisasi perbandingan hasil rata-rata validasi silang kedua model *default* dapat dilihat pada Gambar 13.



Gambar 13. Perbandingan rata-rata hasil validasi silang kedua model *tuning hyperparameter*

Dapat lihat dari Gambar 13, bahwa hasil dari validasi silang kedua model *set tuning hyperparameter* adalah model *tuning hyperparameter XGBoost* memiliki nilai *error* pada RMSE lebih stabil, MAE *dan* MAPE lebih rendah, menunjukan ketahanan yang lebih baik terhadap variasi dalam data dan dapat dikatakan lebih baik dibandingkan model *tuning Random Forest*.

4.8 Uji Model

Setelah model berhasil dilakukan validasi silang, model *Random Forest* dan *XGBoost* dilakukan *train* dan *test* menggunakan data *train* yang telah dipersiapkan yaitu sebanyak 80% dan data *test* 20% dari keseluruhan data pada tahap pembagian data.

4.9 Evaluasi Model

Mengukur seberapa baik model pembelajaran mesin dalam penelitian ini, digunakan tiga cara untuk mengukur sejauh mana model berkinerja dengan baik. Ketiga cara ini adalah RMSE, MAE, dan MAPE. Sama seperti saat dilakukan validasi silang namun yang membedakan pada tahap ini ialah model diuji dengan menggunakan data *test* yang berjumlah 20% dari total keseluruhan data.

Pada tahap ini ditambahkan kecepatan model dalam proses *training* dan prediksi untuk melihat model yang lebih cepat dalam proses *training* dan prediksi.

4.9.1 Model *Default*

Model *default* adalah model yang merujuk pada pengaturan parameter yang digunakan ketika model tersebut diinisialisasi tanpa spesifikasi tambahan dari pengguna.

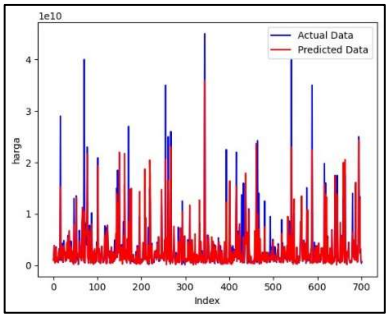
1) *Random Forest*

Pada *Random Forest* digunakan dengan parameter yang telah diatur menghasilkan evaluasi model yang dapat dilihat pada Tabel 8.

Tabel 8. Hasil Evaluasi model Random Forest

RMSE	MAE	MAPE (%)	*Training*(s)	*Prediction*(s)
0.33	0.19	0.91	1.6850	0.0389

Visualisasi hasil prediksi dan aktual model *Random Forest* dapat dilihat pada Gambar 14.



Gambar 14. Visualisasi data aktual dan prediksi model *Random Forest default*

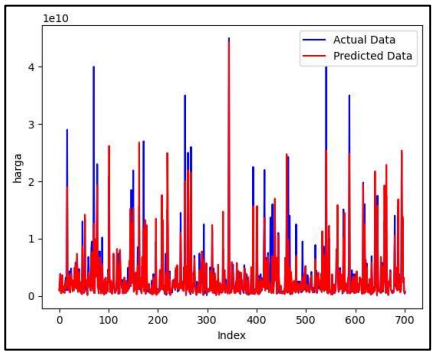
2) *XGBoost*

Pada model *XGBoost* digunakan dengan parameter yang telah diatur menghasilkan evaluasi model yang dapat dilihat pada Tabel 9.

Tabel 9. Hasil evaluasi model *XGBoost*

RMSE	MAE	MAPE (%)	*Training*(s)	*Prediction*(s)
0.33	0.21	1.00	0.2199	0.0050

Visualisasi hasil prediksi dan aktual model *XGBoost* dapat dilihat pada Gambar 15.



Gambar 15. Visualisasi data aktual dan prediksi model *XGBoost default*

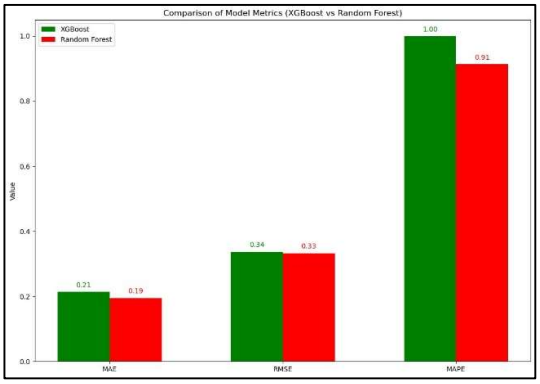
3) Perbandingan Evaluasi Model *Default*

Pada langkah ini dilakukan perbandingan evaluasi model pada kedua model *default Random Forest* dan XGBoost. Hasil perbandingan kedua model dapat dilihat pada Tabel 10.

Tabel 10. Perbandingan evaluasi model *default*

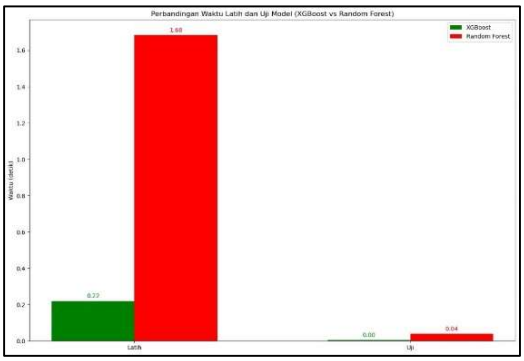
Model	RMSE	MAE	MAPE	*Training*(s)	*Prediction*(s)
Random Forest	0.33	0.19	0.91%	1.6850	0.0389
XGBoost	0.34	0.21	1.00%	0.2199	0.0050

Visualisasi perbandingan evaluasi model dapat dilihat pada Gambar 16.



Gambar 16. Perbandingan Evaluasi Model *Default*

Visualisasi perbandingan waktu dalam proses model *training* dan *predictions* dapat dilihat pada Gambar 17.



Gambar 17. Perbandingan waktu model *default*

Hasil perbandingan evaluasi kedua model *set default* pada Gambar 17 memperoleh hasil bahwa metode *Random fOrest* dan *XGBoost* memiliki akurasi yang sangat tinggi berdasarkan RMSE, MAE dan MAPE.

Model *Random Forest* memiliki sedikit keunggulan pada nilai *error* MAPE yaitu memiliki 0.09 % lebih kecil, dan memiliki nilai *error* 0.01 lebih sedikit pada RMSE dan MAE lebih kecil 0.02 dibandingkan dengan model *XGBoost*. Namun dalam hal kecepatan yang dibutuhkan dalam proses *training* dan *predictions*, *XGBoost* lebih cepat dibandingkan dengan model *random forest*.

4.9.2 Model *Tuning Hyperparameter*

Model *Tuning Hyperparameter* adalah model yang sudah di sesuaikan parameternya sehingga dapat mengoptimalkan kemampuan model dalam prosesnya yang bertujuan untuk meningkatkan kinerja pada model itu sendiri.

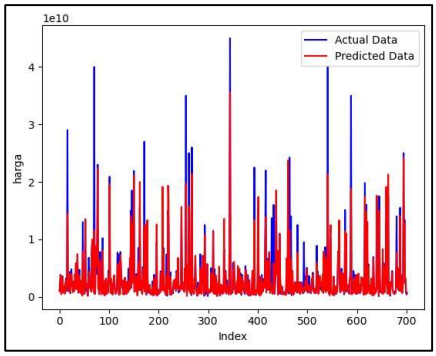
1) *Random Forest*

Pada *Random Forest* digunakan dengan parameter yang telah diatur menghasilkan evaluasi model yang dapat dilihat pada Tabel 11.

Tabel 11. Hasil evaluasi model Random Forest *tuning hyperparameter*

RMSE	MAE	MAPE (%)	*Training*(s)	*Prediction*(s)
0.31	0.19	0.89	3.3859	0.3330

Visualisasi hasil prediksi dan aktual model *Random Forest* dapat dilihat pada Gambar 18.



Gambar 18. Visualisasi data aktual dan prediksi model *Random Forest tuning hyperparameter*

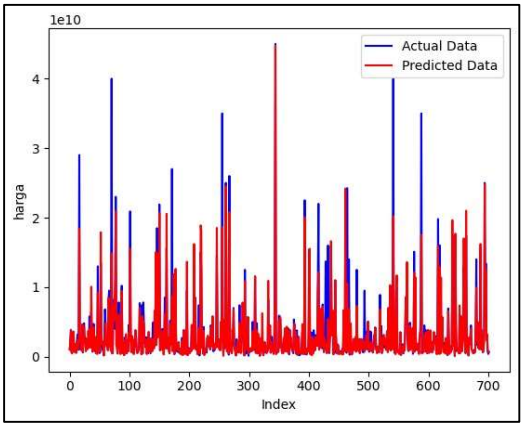
2) *XGBoost*

Pada model *XGBoost* digunakan dengan parameter yang telah diatur menghasilkan evaluasi model yang dapat dilihat pada Tabel 12.

Tabel 12. Hasil evaluasi model *XGBoost tuning hyperparameter*

RMSE	MAE	MAPE (%)	*Training*(s)	*Prediction*(s)
0.30	0.17	0.84	4.7121	0.0090

Visualisasi hasil prediksi dan aktual model *XGBoost* dapat dilihat pada Gambar 19.



Gambar 19. Visualisasi data aktual dan prediksi model *XGBoost tuning hyperparameter*

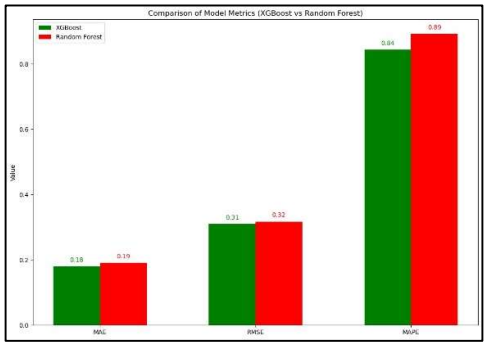
3) Perbandingan Evaluasi Model *Tuning Hyperparameter*

Pada langkah ini dilakukan perbandingan evaluasi model pada kedua model *tuning hyperparameter Random Forest* dan *XGBoost*. Hasil perbandingan kedua model dapat dilihat pada Tabel 13.

Tabel 13. Perbandingan evaluasi model *tuning hyperparameter*

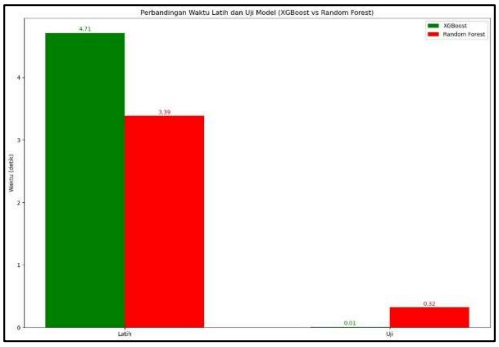
Model	RMSE	MAE	MAPE	*Training*(s)	*Prediction*(s)
Random Forest	0.33	0.19	0.91%	1.6850	0.0389
XGBoost	0.34	0.21	1.00%	0.2199	0.0050

Visualisasi perbandingan evaluasi model dapat dilihat pada Gambar 20.



Gambar 20. Perbandingan Evaluasi Model *Tuning Hyperparameter*

Visualisasi perbandingan waktu dalam proses model *training* dan *predictions* dapat dilihat pada Gambar 21.



Gambar 21. Perbandingan waktu model *tuning hyperparameter*

Hasil perbandingan evaluasi kedua model *set tuning hyperparameter* pada Gambar 21 memperoleh hasil bahwa metode *Random Forest* dan *XGBoost* memiliki akurasi yang sangat tinggi berdasarkan RMSE, MAE dan MAPE.

Model *XGBoost* memiliki sedikit keunggulan pada nilai *error* MAPE yaitu memiliki 0.05% lebih kecil, dan memiliki nilai *error* 0.01 lebih sedikit pada RMSE dan MAE lebih kecil 0.01 dibandingkan dengan model *Random Forest*. Namun dalam hal kecepatan yang dibutuhkan dalam proses *training*, *XGBoost* membutuhkan waktu yang sedikit lebih lama tetapi dalam proses *predictions* atau uji, *XGBoost* lebih cepat dibandingkan dengan model *Random Forest*.

4.10 Prediksi

Pada tahap ini dilakukan prediksi harga jual rumah menggunakan model yang telah dilatih, dilakukan validasi silang kemudian di uji menggunakan data uji yang berjumlah 20% dari total keseluruhan data, lalu di analisis evaluasi model dari masing-masing model yang telah di *set default* dan *set tuning hyperparameter* untuk dipilih dan disimpan. Model akan di lakukan proses prediksi harga jual rumah dengan menggunakan data inputan terbaru. Data inputan

terbaru untuk di lakukan prediksi harga jual rumah di wilayah Jabodetabek dapat dilihat pada Tabel 14.

Tabel 14. Data Baru

Variabel	X_1	X_2
Kota	1 (Bogor)	3 (Tangerang)
Kamar Tidur	3	3
Kamar Mandi	1	1
Luas Tanah (m2)	75	75
Luas Bangunan (m2)	50	50
Kelistrikan (mAh)	1300	1300
Kamar Tidur Pembantu	0	0
Kamar Mandi Pembantu	0	0
Lantai	1	1

Pada model yang di *set default* ini, *Random Forest* yang akan dipilih untuk dilakukan proses prediksi dan pada model yang *set tuning*, *XGBoost* yang akan dipilih karena memiliki nilai *error* yang paling kecil pada masing-masing kedua model. Hasil prediksi yang didapatkan oleh kedua model yang dipilih dapat dilihat pada Tabel 15 dan Tabel 16.

Tabel 15. Hasil prediksi model *Random Forest set default*

Variabel	X_1	X_2
Kota	1	3
Kamar Tidur	3	3
Kamar Mandi	1	1
Luas Tanah (m2)	75	75
Luas Bangunan (m2)	50	50
Kelistrikan (mAh)	1200	1200
Kamar Tidur Pembantu	0	0
Kamar Mandi Pembantu	0	0
Lantai	1	1
Prediksi Harga	Rp. 483.120.000	Rp. 576.080.000

Tabel 16. Hasil prediksi model *XGBoost set tuning parameter*

Variabel	X_1	X_2
Kota	1	3
Kamar Tidur	3	3
Kamar Mandi	1	1
Luas Tanah (m2)	75	75
Luas Bangunan (m2)	50	50
Kelistrikan (mAh)	1200	1200
Kamar Tidur Pembantu	0	0
Kamar Mandi Pembantu	0	0
Lantai	1	1
Prediksi Harga	Rp. 530.970.00	Rp. 602.620.000

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan adalah bahwa metode *Random Forest* menghasilkan prediksi harga jual rumah di wilayah Jabodetabek yang lebih baik daripada *XGBoost*. Hal ini terbukti dengan nilai *error* RMSE sebesar 0.33, MAE sebesar 0.19, dan MAPE sebesar 0.91%. *Random Forest* juga menunjukkan hasil yang cukup baik tanpa adanya kecenderungan *underfitting* atau *overfitting* ketika dilakukan proses menggunakan data uji.

XGBoost perlu dilakukan *set tuning parameter* agar menghasilkan nilai galat lebih kecil dibandingkan dengan model *Random Forest*. Hal ini terbukti nilai *error* yang didapatkan menurun sehingga sangat baik dalam melakukan prediksi. Nilai yang didapatkan RMSE 0.30, MAE sebesar 0.18, dan MAPE sebesar 0.84 %.

Model *XGBoost* sangat perlu diperhatikan dalam *set tuning parameter* sehingga dapat mengoptimalkan model *XGBoost* dengan baik. Berbanding terbalik dengan *Random Forest* model yang sangat mudah diterapkan tanpa perlu memperhatikan *set tuning parameter* lebih lanjut.

5.2 Saran

Penelitian selanjutnya, beberapa hal yang dapat diambil adalah melakukan pengaturan parameter lebih lanjut pada model *Random Forest* dan *XGBoost* untuk meningkatkan nilai akurasi prediksi. Selain penyetelan parameter yang lebih mendalam, penanganan data hilang yang lebih baik, perluasan *dataset* dan penambahan fitur-fitur yang relevan dapat meningkatkan kinerja model.

DAFTAR PUSTAKA

[1] I. Maula, L. U. Hasanah, and A. Tholib, "Analisis Prediksi Harga Rumah Di Jabodetabek Menggunakan Multiple Linear Regression," *J. Inform. Kaputama*, vol. 7, no. 2, pp. 216–224, 2023, doi: 10.59697/jik.v7i2.135.

[2] Kemenpu, "MEMPERSIAPKAN JABODETABEK UNTUK 30 JUTA ORANG," *Kementerian Pekerjaan Umum dan Perumahan Rakyat*, 2020. https://pu.go.id/berita/mempersiapkan-jabodetabek-untuk-30-juta-orang (accessed Dec. 15, 2023).

[3] E. Nuky, "Penduduk 30 Juta Melebihi Australia, Jabodetabek Harus Jadi Jakarta Megapolitan," *Investor Trust*, 2024. https://investortrust.id/news/penduduk-30-juta-melebihi-australia-jabodetabek-harus-dijadikan-jakarta-megapolitan (accessed Dec. 23, 2023).

[4] A. F. Febriyani, "Dampak Pembangunan Apartemen Bagi Kehidupan Masyarakat," *J. Kaji. Komun. Dan Pembang. Drh.* , vol. 10, no. 1, pp. 27–34, 2022.

[5] E. F. Rahayuningtyas, F. N. Rahayu, and Y. Azhar, "Prediksi Harga Rumah Menggunakan General Regression Neural Network," *J. Inform.*, vol. 8, no. 1, pp. 59–66, 2021, doi: 10.31294/ji.v8i1.9036.

[6] A. Aljufri, A. Amalia Meidhani, H. Adriel, and R. Wijaya Limas, "Faktor yang Mempengaruhi Niat Membeli Rumah di Jabodetabek Menurut Teori Model Perilaku Pembeli," *PERWIRA - J. Pendidik. Kewirausahaan Indones.*, vol. 4, no. 1, pp. 1–12, 2021, doi: 10.21632/perwira.4.1.1-12.

[7] A. A. G. S. Utama, "The Best Model and Variables Affecting Housing Values of Big Cities," vol. 10, no. 6, pp. 782–793, 2022.

[8] Y. Li and W. Chen, "A comparative performance assessment of ensemble learning for credit scoring," *Mathematics*, vol. 8, no. 10, pp. 1–19, 2020, doi: 10.3390/math8101756.

[9] B. Bawono and R. Wasono, "Perbandingan Metode Random Forest dan Naive Bayes Untuk Klasifikasi Debitur Berdasarkan Kualitas Kredit," *J. Sains dan Sist. Inf.*, vol. 3, no. 7, pp. 343–348, 2019, [Online]. Available: http://prosiding.unimus.ac.id

[10] S. E. Herni Yulianti, Oni Soesanto, and Yuana Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit," *J. Math. Theory Appl.*, vol. 4, no. 1, pp. 21–26, 2022, doi: 10.31605/jomta.v4i1.1792.
[11] A. Kumar, "Random Forest vs XGBoost: Which One to Use?," *Vital Flux*, 2023. https://vitalflux.com/random-forest-vs-xgboost-which-one-to-use/ (accessed Jul. 15, 2024).
[12] E. Fitri, "Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 58–64, 2023, doi: 10.52158/jacost.v4i1.491.
[13]]P. T. Wibowo, "Apa Itu Kaggle?," *Warta Ekonomi*, 2021. https://wartaekonomi.co.id/read379561/apa-itu-kaggle (accessed Dec. 11, 2023).
[14] [A. B. Adetunji, O. N. Akande, F. A. Ajala, O. Oyewo, Y. F. Akande, and G. Oluwadara, "House Price Prediction using Random Forest Machine Learning Technique," *Procedia Comput. Sci.*, vol. 199, no. February, pp. 806–813, 2021, doi: 10.1016/j.procs.2022.01.100.
[15] J. Melvin and A. Soraya, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *J. Ris. Rumpun Mat. dan Ilmu Pengetah. Alam*, vol. 2, no. 2, pp. 87–103, 2023.