Implementasi Machine Learning Untuk Memprediksi Harga Rumah Menggunakan Algoritma Random Forest Regressor dan XGBoost

http://dx.doi.org/10.28932/jutisi.vXiX.X

Riwayat Artikel
Received: xx Bulan 20xx | *Final Revision*: xx Bulan 20xx | *Accepted*: xx Bulan 20xx

David Leonardo Wahyudi[1], Abhivandya Audry Riana[2], Dheandra Halwa Ghassani[3], Tara Mahaputra Genia Mawikere[4],Verline[5]

[#] *Teknik Informatika, Universitas Kristen Maranatha*
Jalan Surya Sumantri no 65, Bandung,40164, Indonesia.

[1]1972017@maranatha.ac.id
[2]2172035@maranatha.ac.id
[3]2172040@maranatha.ac.id
[4]2272901@maranatha.ac.id
[5]2372901@maranatha.ac.id

***Abstrak* —** Harga rumah di kawasan ibukota DKI Jakarta merupakan suatu wadah sebagai investasi bagi para pencari rumah zaman sekarang. Setiap rumah yang ada di wilayah DKI Jakarta memiliki spesifikasi yang berbeda baik dari harga, luas, dan jumlah ruangan yang terdapat di rumah tersebut. Maka, tujuan penelitian ini bertujuan untuk membuat model prediksi harga rumah menggunakan metode pembelajaran mesin. Penelitian ini menggunakan total dataset sebanyak 2409 data dengan 15 variabel yang diperoleh dari rumah123.com. Adapun variabel yang akan kita gunakan untuk dataset harga rumah diantaranya luas tanah, luas bangunan, luas bangunan berdiri, banyaknya kamar tidur, banyaknya kamar mandi, dan ketersediaan tempat parkir mobil, terdapatnya keamanan maupun taman, jarak rumah ke rumah sakit, jarak rumah ke sekolah maupun jarak rumah ke tol terdekat. website Data tersebut akan melalui proses data *preprocessing* sebelum melatih model. Selanjutnya pada tahap evaluasi, Hasil pengujian 5 algoritma tersebut dievaluasi dengan nilai MSE(*Mean Square Error*), MAE*(Mean Absolute Error*), dan akurasi model untuk menilai kinerja model dengan nilai yang terendah. Berdasarkan hasil analisa dari dataset sebanyak 2216 data, dimana data tersebut dibagi menjadi dua bagian yaitu 80% sebagai data latih dan 20% sebagai data uji. Dari hasil analisa, Algoritma *random forest regressor* dan *XGBoos*t dinyatakan menghasilkan prediksi terbaik dibandingkan 5 algoritma tersebut, dengan *Random Forest Regressor* tingkat akurasi sebesar 0.7585, nilai MSE sebesar 3.693697040720869e+19, dan nilai MSE sebesar 3118528783.837409. Sedangkan pada algoritma *XGBoost*, tingkat akurasi yang didapatkan sebesar 0.73715, nilai MSE sebesar 4.021063475901835e+19 dan juga MAE sebesar 3278723463.783784

.

Kata kunci— Linear Regression, XGBoost, Pembelajaran mesin;Prediksi harga rumah.

Implementation of Machine Learning to Predict House Prices Using Linear Regression and XGBoost Algorithms

***Abstract* —** *The price of houses in the capital region of DKI Jakarta represents an attractive investment opportunity for modern home seekers. Each house in the DKI Jakarta area has different specifications in terms of price, size, and the number of rooms. Therefore, the aim of this research is to create a house price prediction model using machine learning methods. This research uses a total dataset of*

2409 data with 15 variables obtained from rumah123.com. The variables used in the house price dataset include land area, building area, built-up area, number of bedrooms, number of bathrooms, availability of parking spaces, presence of security or a garden, distance from the house to the nearest hospital, school, or toll road. The data will undergo preprocessing before training the model. In the evaluation stage, the results of the five algorithms are evaluated using MSE (Mean Square Error), MAE (Mean Absolute Error), and model accuracy to assess the model performance with the lowest values. Based on the analysis of a dataset consisting of 2216 data points, which is divided into two parts: 80% for training data and 20% for testing data. From the analysis results, the Random Forest Regressor and XGBoost algorithms are declared to produce the best predictions compared to the five algorithms, with the Random Forest Regressor achieving an accuracy rate of 0.7585, an MSE value of 3.693697040720869e+19, and an MAE value of 3118528783.837409. Meanwhile, the XGBoost algorithm achieved an accuracy rate of 0.73715, an MSE value of 4.021063475901835e+19, and an MAE value of 3278723463.783784.

Keywords—Linear Regression; XGBoost; Machine learning; Predict House Price.

I. PENDAHULUAN

Harga rumah di kawasan ibukota DKI Jakarta menjadi semakin signifikan sebagai investasi bagi individu pada era saat ini. Akan tetapi, harga rumah dapat berubah dengan waktu dan keadaan tertentu. Variasi spesifikasi setiap rumah di wilayah ini, termasuk harga, luas, dan jumlah ruangan, mencerminkan kompleksitas pasar properti. Maka dari itu, harga rumah sangatlah susah untuk diprediksi. Pasar properti di Jakarta terkenal dengan sifatnya yang dinamis, di mana harga rumah terus mengalami perubahan. Hal ini menjadikan prediksi harga rumah di kota metropolitan ini sebagai topik yang menarik dan penuh tantangan.Memahami faktor-faktor yang mempengaruhi harga rumah dapat memberikan wawasan berharga bagi investor, pemilik rumah, dan pembuat kebijakan. Selain itu, prediksi harga rumah yang akurat dapat membantu dalam mengambil keputusan yang terinformasi terkait investasi dan perencanaan keuangan.

Maka dari itu, penelitian ini dilakukan untuk melakukan sebuah sistem prediksi harga rumah berdasarkan kriteria tertentu menggunakan Machine Learning. Dengan menganalisis sejumlah variabel yang mempengaruhi harga rumah, kami berupaya untuk mengembangkan model yang tidak hanya memiliki performa baik dalam memprediksi harga, tetapi juga memberikan wawasan tentang faktor-faktor yang relatif penting dalam menentukan nilai properti di lingkungan perkotaan ini. Studi ini akan menjawab kebutuhan praktis akan prediksi harga yang dapat diandalkan dalam pasar real estat yang kompleks.

Algoritma yang akan digunakan dalam penelitian ini adalah algoritma *Random Forest Regressor* dan *XGBoost Regression*. Penelitian ini menggunakan algoritma berikut karena algoritma ini merupakan metode yang dapat mendapatkan hasil nilai MAE dan MSE yang lebih kecil daripada algoritma prediksi lainnya. Pada proses penelitian, akan dibandingkan hasil prediksi Algoritma Random Forest Regressor dan XGBoost Regression dengan 3 algoritma lainnya yaitu SVR, Decision Tree dan Linear Regression.

Sebelum proses penelitian ini, ditemukan beberapa penelitian serupa yang memakai algoritma *Random Forest Regressor* dan *XGBoost Regression*. Penelitian tersebut dapat dijadikan alat bantu dalam menerapkan algoritma *Random Forest Regressor* dan *XGBoost Regressio*n dalam penelitian ini.

Penelitian serupa yang pertama ini dilakukan oleh Nicholas Hadi, Jason Benedict dengan judul "Implementasi Machine Learning untuk Prediksi Harga Rumah Menggunakan Algoritma Random Forest"(2024). Penelitian ini dilakukan menggunakan dataset harga rumah di King County, USA dari Kaggle. Variabel luas rumah, *grade*, dan luas atas rumah memiliki pengaruh besar terhadap harga rumah berdasarkan pengujian korelasi. Algoritma *Machine Learning* yang digunakan adalah *Random Forest, Decision Tree,* dan P*olynomial Regression*. Hasil evaluasi menunjukkan bahwa *Random Forest* memberikan prediksi terbaik dengan tingkat akurasi 86,54% dan RMSE sebesar 144,913.73.

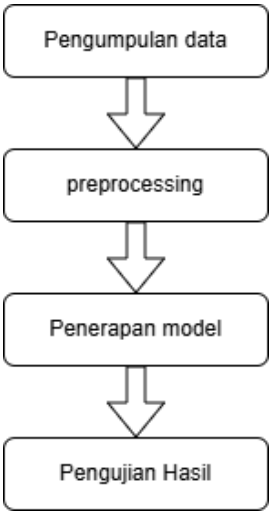
Penelitian serupa selanjutnya ini dilakukan oleh Ega Sri Lestari, Ida Astuti (2016) dengan penelitian yang berjudul "Penerapan Random Forest Regression Untuk Memprediksi Harga Jual Rumah Dan Cosine Similarity Untuk Rekomendasi Rumah Pada Provinsi Jawa Barat". Penelitian ini bertujuan mengimplementasikan *Random Forest Regression* untuk memprediksi harga rumah dan *Cosine Similarity* untuk merekomendasikan rumah. Data diambil melalui *web scraping* dan diolah menggunakan metode CRISP-DM yang meliputi *business understanding, data understanding, data preparation, modeling, evaluation,* dan *deployment*. Prediksi harga rumah dengan tuning parameter mencapai akurasi 85,29%, sementara rekomendasi rumah mencapai akurasi 89,99% pada data uji. Website yang dibangun memungkinkan pengguna mencari harga rumah di Jawa Barat dan memberikan rekomendasi dengan link ke rumah123.com. Pengujian inferensial menunjukkan nilai precision 75%, recall 100%, akurasi 80%, dan f-measure 86% untuk prediksi harga, serta precision 78%, recall 100%, akurasi 80%, dan f-measure 88% untuk rekomendasi. User acceptance test menunjukkan kepuasan pengguna sebesar 89,29% dengan kategori sangat baik.

Penelitian serupa *XGBoost Regression* ini dilakukan oleh Hayqal Hazmi Qastari, dkk berjudul "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit"(2022). .Penelitian ini bertujuan untuk mengklasifikasikan nasabah kartu kredit menjadi tidak macet (0) atau macet (1) menggunakan metode XGBoost. Mereka

menggunakan dataset "*default of credit card clients*" dari *UCI Machine Learning Repository* yang terdiri dari 30,000 data nasabah dengan 24 variabel, tetapi hanya menggunakan 7 variabel dalam analisis mereka. Hasil pertama dengan *parameter default* menunjukkan akurasi model sebesar 80,02%, presisi 85,32%, dan recall 94,86%, yang dikategorikan sebagai *good classification*. Percobaan kedua dengan melakukan hyperparameter tuning berhasil meningkatkan akurasi menjadi 83,42%, presisi 85,36%, dan recall 95,28%, juga termasuk dalam kategori good classification.

Penelitian serupa selanjutnya dilakukan oleh Beno Jange dengan judul "Prediksi Harga Saham Bank BCA Menggunakan XGBoost"(2022). Data yang diambil adalah data harga saham harian (open, high, low, closedan adj close) dan volume selama periode 01/01/2017 hingga 31/12/2020. Berdasarkan hasil pembahasan yang telah dilakukan maka dapat disimpulkan bahwa metode XGBoost cukup baik dalam memprediksi harga saham Bank BCA. Melalui beberapa penyetelan parameter hiper pada metode *XGBoost* akan didapatkan prediksi yang lebih baik yaitu MAPE sebesar 4.01 persen.

II. METODE PENELITIAN



Gambar 1.Tahap metode penelitian

A. Pengumpulan Data

Pada pelaksanaannya penelitian ini menggunakan pengumpulan data untuk prediksi harga rumah yang dilakukan dengan studi dokumenter pada website jual beli rumah yaitu rumah123.com. Karakteristik data terdiri dari 15 variabel. 14 variabel yang dijadikan variabel X atau disebut variabel dependen dan 1 variabel yang dijadikan variabel Y atau variabel independen. jumlah data terkumpul dan digunakan sebanyak 2409 data. variabel yang dikumpulkan untuk memprediksikan harga suatu rumah yaitu harga rumah, luas tanah, luas bangunan, banyaknya kamar tidur, banyaknya kamar mandi, dan ketersediaan tempat parkir mobil, besarnya pasokan listrik,kabupaten/kota, kecamatan, kelurahan, terdapatnya keamanan, terdapatnya taman, jarak rumah ke rumah sakit, jarak rumah ke sekolah maupun jarak rumah ke tol terdekat

B. Preprocessing

Pre-processing adalah tahap untuk membantu metode yang digunakan agar dapat menghasilkan nilai output yang lebih baik, ini merupakan tahap yang tidak dapat dilewatkan dalam pengolahan data. *Pre-processing data* yang dilakukan adalah melakukan *cleaning* pada data dengan maksud untuk memperbaiki atau menghapus data yang rusak maupun data yang tidak relevan. tahapan yang dilakukan pada preprocessing adalah

1. Data yang tidak relevan ataupun hilang akan dihapus.
2. Data yang ada memiliki tipe data *float* dan *integer*. Diberikan pre-processing berupa proses *converting* semua data ke *float* untuk memudahkan dalam proses modeling.
3. Mengubah data ke dalam format yang sesuai untuk analisis yaitu pengkodean variabel kategorikal terhadap beberapa variabel.
4. Normalisasi menggunakan *Standar Scaler* pada variabel X

5. Mengurangi jumlah fitur dalam dataset untuk meningkatkan kinerja model dan mengurangi kompleksitas dengan PCA menjadi hanya 2 komponen utama(*n-components* = 2)

C. Penerapan model

sebelum penerapan model, ada satu tahapan penting yaitu pembagian data. Pada tahap ini data penelitian akan dibagi menjadi 2 bagian yaitu data latih dan data uji.

Pada tahap ini dilakukan penerapan metode Algoritma *Random Forest Regressor, XGBoost Regression, SVR, Decision Tree* dan *Linear Regression*. pada data yang akan diteliti.

Tahapan yang akan dilakukan, sebagai berikut:

1. Melatih berbagai model diantaranya *Linear regression, SVR, Decision tree regression, Random Forest Regression, XGBoost regression* menggunakan data latih.
2. Melakukan proses prediksi menggunakan model yang dilatih sebelumnya

Tahap selanjutnya, kita akan melakukan proses hyperparameter tuning menggunakan metode *grid search cross validation* pada 2 model dengan MSE maupun MAE terendah. Tujuan dari langkah ini adalah untuk menemukan kombinasi optimal dari hyperparameter yang meningkatkan kinerja model *Random Forest Regressor* dan *XGBoost regression* dalam memprediksikan harga rumah. Dengan menggunakan *grid search cross validation*, berbagai kombinasi hyperparameter diuji dan divalidasi untuk memastikan model yang dihasilkan memiliki performa terbaik.

d. Pengujian

Untuk menentukan akurasi prediksi, dan kinerja pemodelan, digunakan beberapa parameter evaluasi, diantaranya :

1. *Mean Square Error* (MSE)

   MSE merupakan error yang dikuadratkan, semakin besar error semakin besar juga nilai MSE yang dihasilkan. Cara perhitungan MSE yang dikuadratkan ini membuat data outlier sangat memiliki sensitivitas. Formula MSE didefinisikan sebagai berikut (So et al., 2013):

   $$MSE = \frac{\sum(Y' - Y)^2}{n} \quad (1)$$

   Dengan keterangan :

   - *Y ′ = Predicted*

   - *Y = Actual*

   - *n = Jumlah Data*

2. *Mean Absolute Error* (MAE)

   MAE menentukan rata-rata kesalahan atau error pada hasil *actual* dan *predicted* saat pemodelan, didefinisikan sebagai berikut (Chai & Draxler, 2014) :

   $$MAE = \frac{\sum|Y' - Y|^2}{n} \quad (2)$$

   Dengan keterangan :

   - *Y ′ = Predicted*

   - *Y = Actual*

   - *n = Jumlah Data*

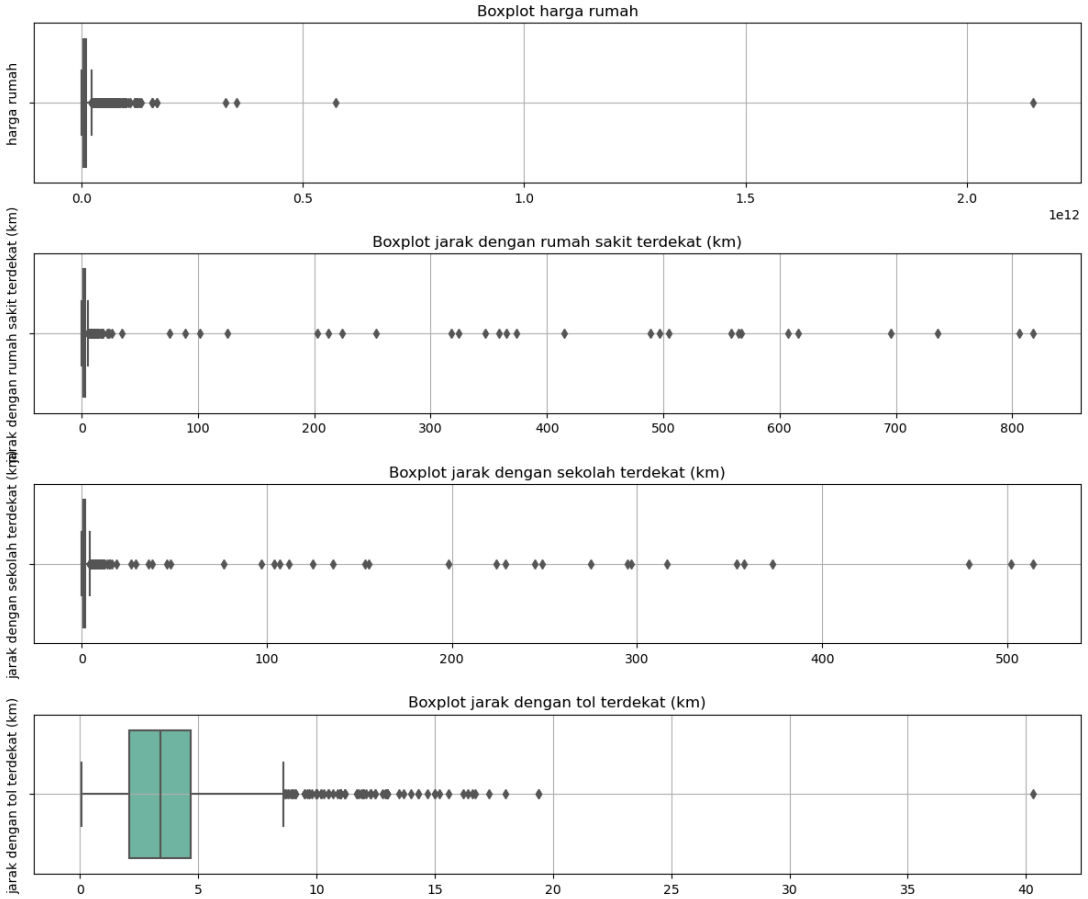
III. HASIL DAN PEMBAHASAN

A. Preprocessing Data

Penelitian ini menggunakan OpenRefine berbasis web yang memudahkan dalam melakukan penelitian prediksi harga rumah. OpenRefine membantu dalam membersihkan dan mengorganisir data yang akan digunakan untuk analisis dan pemodelan prediktif. Terdapat total 2409 baris data dan 15 kolom. Data tersebut sudah dibersihkan dan bisa dilanjutkan ke tahap berikutnya.

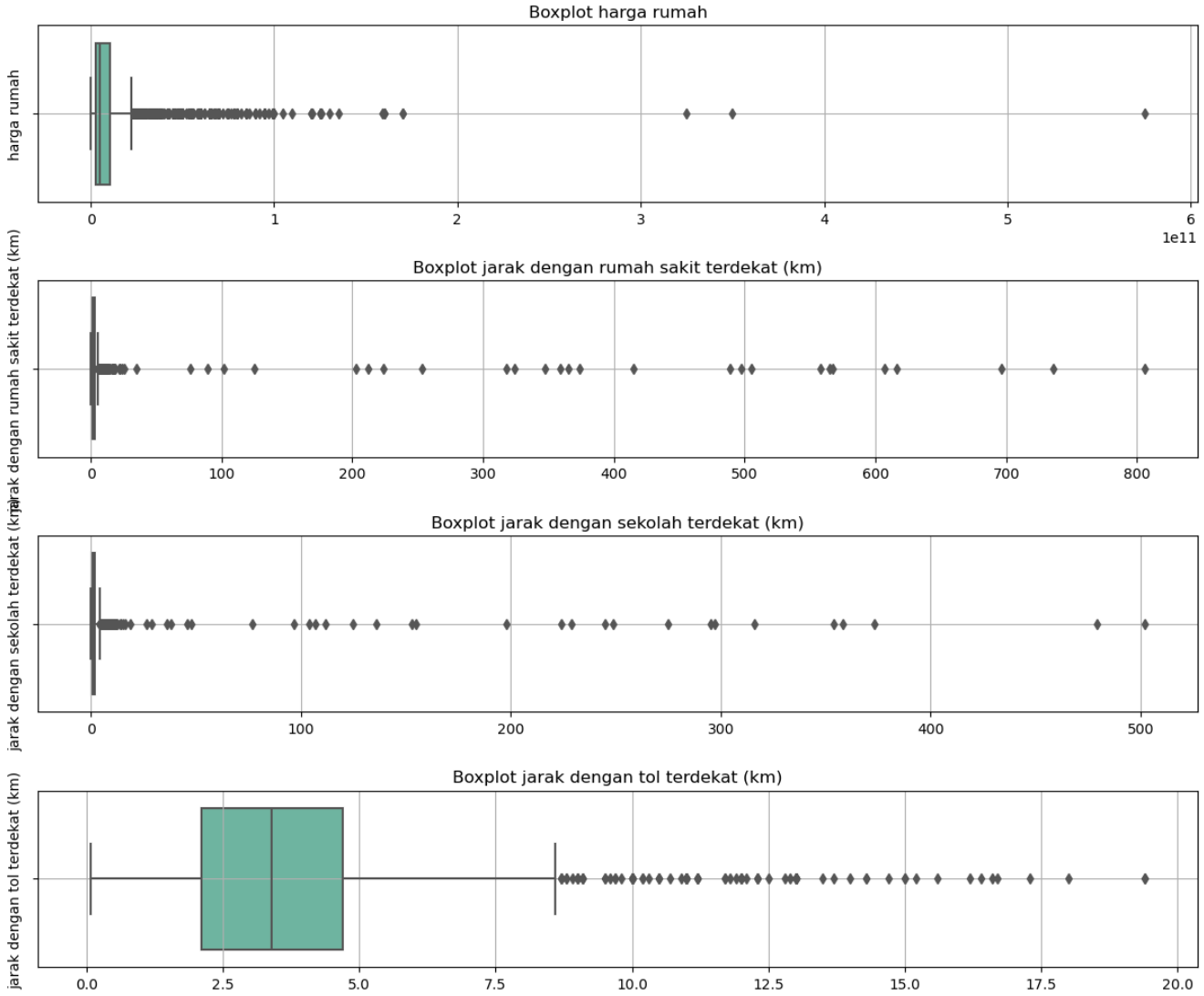
tahapan *preprocessing* yang dilakukan adalah

1. menghapus kolom yang tidak relevan dengan dataset yang akan digunakan yaitu ‘no’, ‘NRP’, ‘nama’, dan ‘link data rumah’
2. menghapus barisan pada semua kolom jika ada data yang kosong
3. Data yang ada memiliki tipe data *float* dan *integer*. Diberikan pre-processing berupa proses *converting* semua data ke *float* untuk memudahkan dalam proses modeling. sebelum itu, akan dilakukan proses pengecekan tipe data terlebih dahulu. kolom yang perlu diubah ke tipe data integer berupa harga rumah, jumlah kamar tidur, jumlah kamar mandi, luas tanah (m2), luas bangunan (m2), carport (mobil), pasokan listrik (watt). Sedangkan data yang perlu diubah ke dalam bentuk float berupa jarak dengan rumah sakit terdekat (km), jarak dengan sekolah terdekat (km), jarak dengan tol terdekat (km)
4. pengkodean variabel kategorikal(one-hot encoding) terhadap beberapa variabel, seperti variabel Kabupaten/Kota, kecamatan, kelurahan, keamanan (ada/tidak), taman (ada/tidak)
5. menghapus beberapa data yang outlier
6. melakukan proses normalisasi menggunakan StandardScaler
7. Mengurangi jumlah fitur dalam dataset untuk meningkatkan kinerja model dan mengurangi kompleksitas dengan PCA menjadi hanya 2 komponen utama(*n-components* = 2).

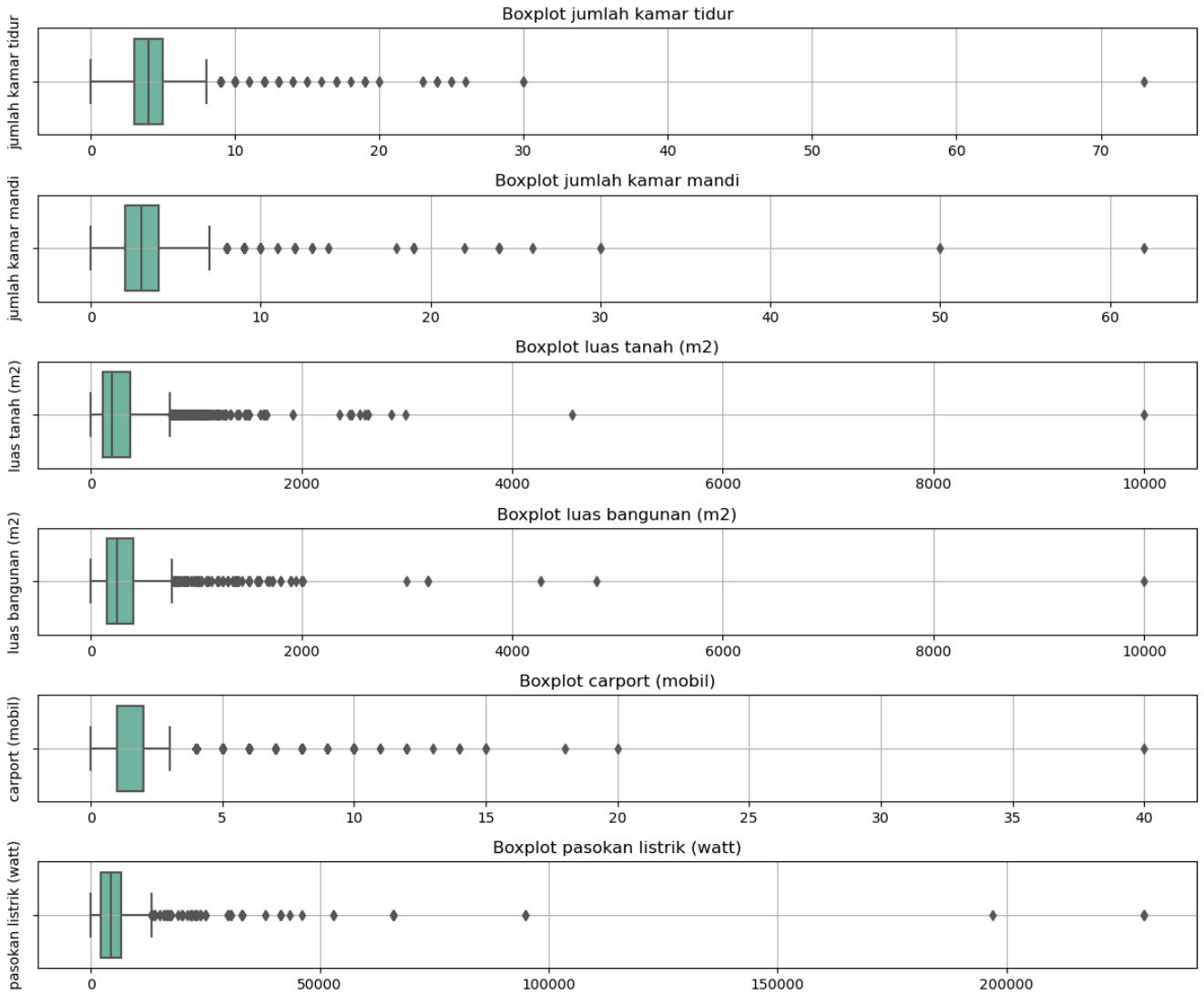


berikut adalah boxplot dari 4 variabel yaitu harga rumah, jarak dengan rumah sakit terdekat, jarak dengan sekolah terdekat dan jarak dengan tol terdekat. berdasarkan boxplot, keempat variabel memiliki banyak *outlier* yang mengindikasikan perlunya langkah berikut yaitu menangani *outlier* atau transformasi data untuk mengurangi *skewness*.

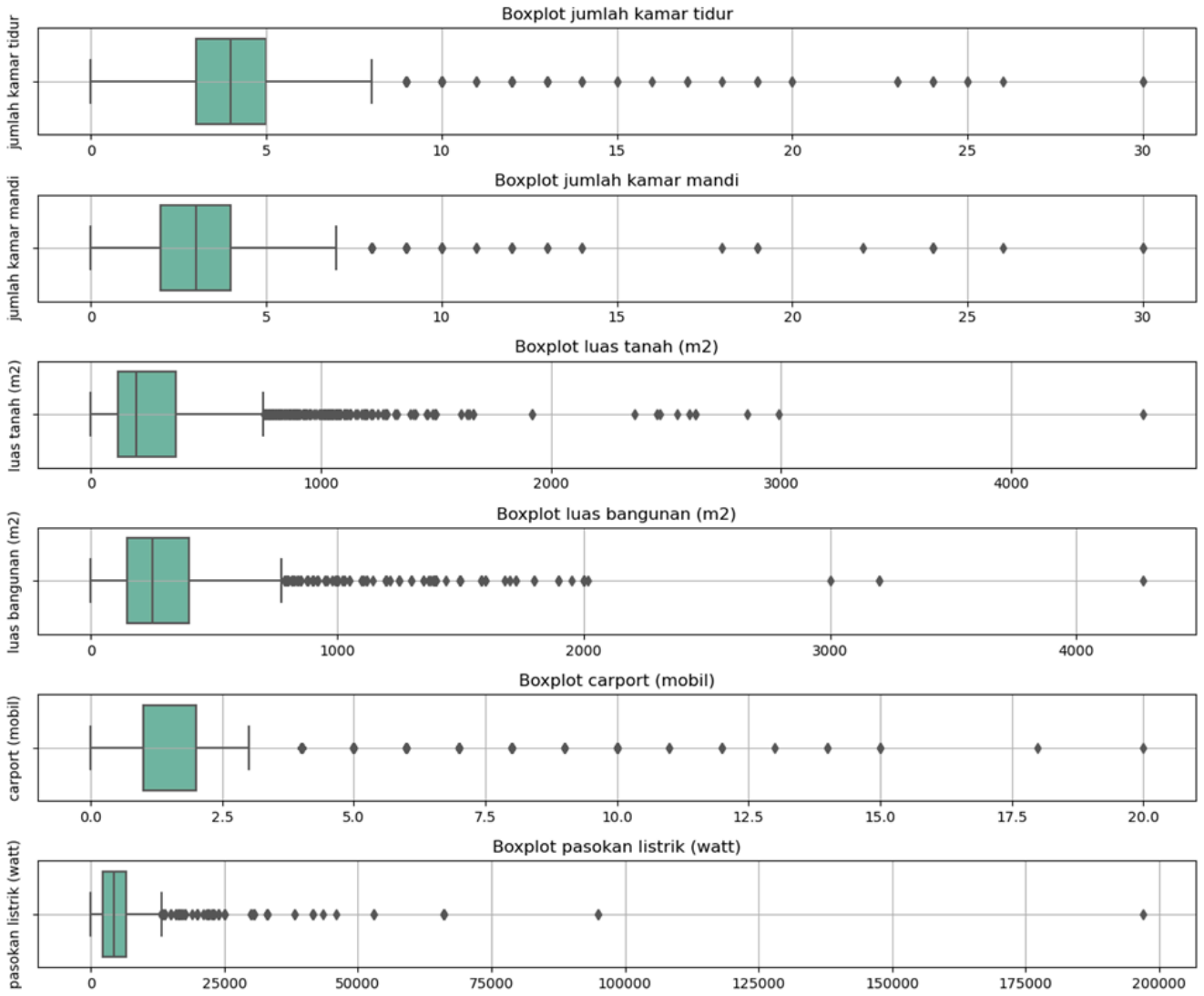
berikut adalah *boxplot* dari *outlier* yang telah ditangani dengan menghapus beberapa data yang memiliki *outlier* yang ekstrem



Dalam gambar pertama, data memiliki rentang yang lebih luas dan banyak *outlier* yang ekstrem pada setiap fitur. Sedangkan dalam gambar kedua, rentang data dan jumlah outlier tampak lebih terkonsentrasi, terutama pada harga rumah dan jarak dengan tol terdekat, yang menunjukkan adanya perbaikan dalam penanganan *outlier*. Misalnya, harga rumah pada gambar kedua menunjukkan rentang hingga sekitar 600 miliar dengan lebih sedikit *outlier*, sedangkan pada gambar pertama rentangnya mencapai lebih dari 2 triliun dengan banyak *outlier* ekstrem. Hal serupa terjadi pada jarak dengan tol terdekat, di mana rentang data pada gambar kedua lebih terkonsentrasi antara 0 hingga 20 km dengan lebih sedikit outlier. Ini menunjukkan bahwa data telah melalui proses pembersihan atau transformasi, sehingga distribusi menjadi lebih terkonsentrasi dan outlier ekstrim berkurang, membantu meningkatkan performa model machine learning.



Boxplot ini menampilkan distribusi dari enam variabel properti: jumlah kamar tidur, jumlah kamar mandi, luas tanah, luas bangunan, kapasitas carport, dan pasokan listrik. Pada setiap plot, kotak menunjukkan interkuartil rentang (IQR) di mana 50% data berada di dalamnya, dengan garis tengah yang menunjukkan median. Garis (whiskers) meluas dari kotak ke nilai minimum dan maksimum yang bukan outlier. Titik-titik di luar *whiskers* adalah *outlier*. Dari plot-plot ini, terlihat bahwa semua variabel memiliki *outlier*, yang tersebar luas terutama pada jumlah kamar tidur, jumlah kamar mandi, luas tanah, luas bangunan, dan pasokan listrik, yang menunjukkan nilai ekstrim yang jauh lebih tinggi dibandingkan dengan mayoritas data. *Outlier* pada jumlah kamar tidur dan kamar mandi mencapai lebih dari 70 dan 60, sedangkan outlier untuk luas tanah, luas bangunan, dan pasokan listrik bahkan mencapai 10,000m2 dan 200,000 watt. dengan ini perlu dilakukan penghapusan data yang memiliki outlier yang jauh dari rentang yang seharusnya.



Setelah penghapusan outlier, semua boxplot menunjukkan distribusi data yang lebih terfokus dengan rentang interkuartil yang lebih representatif dari data inti. Misalnya, jumlah kamar tidur dan kamar mandi sekarang memiliki outlier yang lebih sedikit, memperlihatkan nilai yang lebih wajar, sementara luas tanah, luas bangunan, carport, dan pasokan listrik juga menunjukkan pengurangan signifikan dalam jumlah outlier ekstrem.

Data sudah bersih, tahap selanjutnya yakni pembagian data menjadi 2, yakni data dibagi menjadi data uji dan data latih. Pada proses ini data dibagi meliputi 80% data latih dan 20% data uji. Pembagian ini melihat dari data yang cukup banyak, maka efektif untuk memakai ukuran pembagi 80 : 20 dibanding 70 : 30. Data yang digunakan pada penelitian ini hanya pada 19 variabel yaitu terdiri dari 18 variabel bebas dan 1 variabel terikat. kemudian data akan dilakukan proses normalisasi. Tahapan ini menggunakan standar scaler untuk menstandarisasi data latih dan data uji. Alat ini digunakan untuk menstandarisasi fitur dengan menghilangkan mean dan menskalakan ke unit variance. StandardScaler menghitung mean dan standar deviasi dari data latih, lalu menormalkan data latih agar setiap fitur memiliki mean nol dan standar deviasi satu. Data uji kemudian dinormalisasi menggunakan mean dan standar deviasi yang sama. Proses ini memastikan bahwa data latih dan uji berada pada skala yang sama, meningkatkan kinerja model machine learning. Tahapan selanjutnya adalah mereduksi dimensi fitur dalam dataset. Tahapan ini menggunakan PCA (*Principal Component Analysis*) untuk mereduksi dimensi data yang telah distandarisasi. PCA*(n_components=2*) mengatur agar data direduksi menjadi 2 komponen utama. Data latih yang telah distandarisasi (*X_train_scaled*) dikonversi ke 2 komponen utama menggunakan fit_transform, sedangkan data uji (*X_test_scaled*) dikonversi dengan transform

menggunakan komponen yang sama. Proses ini mengurangi kompleksitas data sambil mempertahankan variansi maksimum, sehingga membantu meningkatkan efisiensi dan performa model machine learning.

B. Hasil model tanpa hyperparameter tuning (baseline)

Tahapan selanjutnya adalah penerapan model. Pada penelitian ini akan dilakukan 5 percobaan yaitu percobaan pertama analisis menggunakan linear regression, percobaan kedua analisis menggunakan model SVR, percobaan ketiga akan menggunakan decision tree regressor, percobaan keempat akan menggunakan model random forest regressor dan yang terakhir akan menggunakan XGBoost regression. Semua prediksi akan dilakukan tanpa hyperparameter tuning. Data dilatih sebanyak 80% dari keseluruhan dataset menggunakan PCA dengan komponen sebanyak 2 (n-component = 2).

TABEL 1

PERBANDINGAN TRAINING 5 MODEL

jenis model	MSE(Mean Square Error)	MAE(Mean Absolute Error)	akurasi(r2-score)
Linear regression	5.580364222439113e+19	4427728846.050124	0.6352174667003623
SVR	1.7262261030422102e+20	6479288702.552059	-0.1284158270236806
Decision Tree Regression	6.092876749153152e+19	4227241441.4414415	0.6017150624861747
Random Forest Regression	3.693697040720869e+19	3118528783.837409	0.7585469137771437
XGBoost Regression	4.021063475901835e+19	3278723463.783784	0.7371473146143512

Berdasarkan analisis perbandingan lima model yang telah evaluasi berdasarkan nilai Mean Square Error (MSE), Mean Absolute Error (MAE), dan akurasi dari model tersebut maka didapatkan ada 2 model terbaik yaitu Random Forest Regressor dan XGBoost Regression. Random Forest Regression menunjukkan performa terbaik dengan nilai MSE dan MAE terendah serta akurasi tertinggi. XGBoost Regression menjadi alternatif terbaik kedua. Linear Regression dan Decision Tree Regression memiliki performa yang cukup baik namun tidak sebaik dua model sebelumnya. SVR memiliki performa terburuk dan tidak direkomendasikan untuk dataset ini. XGBoost Regression menunjukkan nilai MSE dan MAE terendah maupun akurasi tertinggi di antara semua model yang diuji. MSE yang rendah menunjukkan bahwa model ini memiliki kemampuan yang sangat baik dalam memprediksi nilai-nilai data yang tidak dilatih dengan kesalahan kuadrat rata-rata yang kecil. Demikian pula, MAE yang rendah menunjukkan bahwa rata-rata kesalahan absolut juga kecil, yang berarti prediksi model ini sangat dekat dengan nilai aktual. Random Forest Regressor menunjukkan performa yang sangat baik dengan nilai MSE dan MAE yang sedikit lebih tinggi dari XGBoost Regression tetapi masih jauh lebih rendah dibandingkan model lainnya. MAE yang rendah menunjukkan bahwa kesalahan absolut rata-rata juga kecil, yang berarti prediksi model ini cukup akurat dan dekat dengan nilai aktual.

C. Hasil Model dengan Hyperparameter Tuning

Setelah melakukan analisis baseline menggunakan lima model regresi tanpa hyperparameter tuning, hasil menunjukkan bahwa *XGBoost Regression* dan *Random Forest Regressor* memiliki performa terbaik dengan nilai *Mean Square Error* (MSE) dan *Mean Absolute Error* (MAE) terendah. Berdasarkan hasil ini, langkah selanjutnya adalah melakukan

hyperparameter tuning pada kedua model tersebut untuk lebih meningkatkan akurasi prediksi. Pada penelitian ini, akan dilakukan *hyperparameter tuning* pada dua model terbaik, yaitu *XGBoost Regression* dan *Random Forest Regressor.* Berdasarkan model *Random Forest Regressor*, Terdapat 4 parameter yang diperkirakan dapat meningkatkan kinerja model dalam memprediksi menggunakan metode *Random Forest Regressor*. *Hyperparameter tuning* yang dilakukan pada 4 parameter ini menggunakan metode *grid search CV*. *Grid search CV* dikategorikan sebagai metode yang teliti, karena dalam menentukan parameter terbaik dilakukan eksplorasi masing masing parameter dengan mengatur jenis nilai prediksi terlebih dahulu. Konfigurasi hyperparameter optimal dari grid search dipilih berdasarkan nilai akurasi *cross validation* tertinggi dari kandidat *hyperparameter*. Hasil nilai *hyperparameter* terbaik sebagai berikut:

TABEL 2

HASIL TERBAIK RANDOM FOREST REGRESSOR UNTUK HYPERPARAMETER TUNING

Hyperparameter	Grid Search Values Nilai	Hyperparameter Terbaik
n_estimators	100, 200, 300	200
max_depth	None, 10, 20, 30, 40, 50	None
min_samples_split	2, 5, 10	5
min_samples_leaf	1, 2, 4	2

Adapun parameter dan nilai parameter terbaik yang dapat meningkatkan kinerja algoritma dapat dilihat pada Tabel 2. Konfigurasi hyperparameter ini kemudian digunakan untuk melatih ulang model *Random Forest Regressor* menggunakan keseluruhan dari data latih yang sudah direduksi komponennya. Model *Random Forest Regressor* dengan *hyperparameter tuning* kemudian dievaluasi menggunakan data uji untuk mengukur kinerja berupa MSE dan MAE. Adapun untuk hasil kinerja yang dihasilkan pada model *Random Forest Regressor* dengan *hyperparameter tuning* disajikan dalam tabel 4, model *Random Forest Regressor* dengan *hyperparameter tuning* menghasilkan akurasi sebesar 0.7670230830645581, MSE sebesar 3.5640304379727258e+19 dan MAE sebesar 3090478099.1231303

Berdasarkan model *XGBoost regression*, Terdapat 4 parameter yang diperkirakan dapat meningkatkan kinerja model dalam memprediksi menggunakan metode *XGBoost*. Sama dengan *Random Forest Regressor*, hyperparameter tuning yang dilakukan dengan menggunakan metode grid search CV. Hasil nilai hyperparameter terbaik terdapat pada tabel 3 sebagai berikut.

TABEL 3

HASIL TERBAIK XGBOOST UNTUK HYPERPARAMETER TUNING

Hyperparameter	Grid Search Values Nilai	Hyperparameter Terbaik
n_estimators	100, 200, 300	200
learning_rate	0.01, 0.1, 0.2	0.1
max_depth	3, 5, 7	3
subsample	0.8, 1.0	0.8
colsample_bytree	0.8, 1.0	0.8

Konfigurasi hyperparameter terbaik akan digunakan untuk melatih ulang model *XGBoost regression* menggunakan dataset yang telah ada. Tahap selanjutnya, model yang telah dilatih dengan *hyperparameter tuning* akan dievaluasi menggunakan data test untuk mengukur MSE dan MAE, apakah mengalami penurunan atau kenaikan. Adapun untuk hasil kinerja yang dihasilkan dapat dilihat dalam tabel 4, dimana model *XGBoost* dengan *hyperparameter tuning* akan

menghasilkan akurasi sebesar 0.7823362976026847, MSE sebesar 3.329772197135018e+19 dan MAE sebesar 3031818510.4144144.

Pada tabel 4 dapat dilihat tabel perbandingan antara kinerja model *XGBoost* tanpa *hyperparameter tuning* dengan kinerja model *Random Forest Regressor* dan model *XGBoost* dengan hyperparameter tuning.

model	evaluasi	baseline	hyperparameter tuning	perbedaan
Random Forest Regressor	akurasi	0.7585469137771437	0.7670230830645581	0.00847616929
	MSE	3.693697040720869e+19	3.5640304379727258e+19	-0.12966660274814323
	MAE	3118528783.837409	3090478099.1231303	-28050684.714278698
XGBoost Regression	akurasi	0.7371473146143512	0.7823362976026847	0.045188982988333515
	MSE	4.021063475901835e+19	3.329772197135018e+19	-0.6912912787668173
	MAE	3278723463.783784	3031818510.4144144	-246904953.3693695

Guna mengetahui pengaruh *hyperparameter tuning* pada kinerja *Random Forest Regressor* dalam melakukan prediksi pada dataset rumah, maka dilakukan perbandingan antara kinerja model *Random Forest Regressor* dan model *XGBoost* tanpa *hyperparameter tuning* dengan kinerja model *Random Forest Regressor* dan model *XGBoost* dengan *hyperparameter tuning*. Dari hasil MSE dan MAE *Random Forest Regressor* dan *XGBoost* tanpa *hyperparameter tuning* dan hasil model dengan *hyperparameter tuning* pada Tabel 2, dapat dilihat bahwa hyperparameter tuning memberikan pengaruh terhadap model *Random Forest Regressor*, karena terjadinya peningkatan maupun penurunan MSE maupun MAE, yaitu pada akurasi mengalami peningkatan sebesar 0.08476 %, MSE menurun sekitar 0.12966660274814323, dan MAE menurun sekitar 28050684.714278698. Sedangkan pada *XGBoost,* terjadinya peningkatan akurasi sekitar 4.52% dan juga penurunan MSE maupun MAE sebesar 0.6912912787668173 dan 246904953.3693695. Maka dapat disimpulkan bahwa keseluruhan hyperparameter tuning dapat memberikan dampak positif terhadap kinerja model prediksi, terutama pada model *XGBoost*. Hasil dari tuning menunjukkan peningkatan akurasi dan penurunan nilai MSE dan MAE, yang menunjukkan bahwa model menjadi lebih akurat dan memiliki kesalahan prediksi yang lebih kecil setelah dilakukan tuning. Berdasarkan hasil penelitian dapat disimpulkan bahwa *hyperparameter tuning* terbukti mampu meningkatkan kinerja algoritma dalam proses prediksi. Dari uraian diatas dapat dikatakan bahwa *hyperparameter tuning* merupakan hal yang disarankan menjadi salah satu tahapan sebelum melakukan pengujian. Karena sesuai dengan hasil penelitian dan penelitian yang dilakukan bahwa *hyperparameter tuning* dapat meningkatkan kinerja model.

IV. Simpulan

.Penelitian ini membahas implementasi algoritma *Machine Learning*, yaitu *Random Forest Regressor* dan *XGBoost* untuk memprediksi harga rumah di kawasan Jakarta. Dataset yang digunakan berjumlah 2409 data yang diperoleh dari rumah123.com, dengan 15 variabel dependen dan 1 variabel independen. Tahap preprocessing meliputi pembersihan data, konversi tipe data, pengkodean variabel kategorikal, normalisasi, dan pengurangan fitur menggunakan PCA. Lima model regresi yang diuji yaitu *Linear Regression, SVR, Decision Tree Regression, Random Forest Regression,* dan *XGBoost Regression.* Evaluasi awal tanpa *hyperparameter tuning* menunjukkan bahwa *Random Forest Regressor* dan *XGBoost*

Regression memberikan performa terbaik berdasarkan nilai MSE dan MAE. *Hyperparameter tuning* dilakukan pada dua model terbaik (*Random Forest Regressor* dan *XGBoost Regression*) menggunakan metode *grid search cross-validation*. Konfigurasi optimal dari *hyperparameter tuning* ditemukan untuk kedua model. Model *XGBoost Regression* dengan *hyperparameter tuning* menunjukkan peningkatan kinerja yang signifikan dibandingkan model tanpa tuning, dengan akurasi yang meningkat serta penurunan MSE dan MAE. *Random Forest Regressor* juga menunjukkan hasil yang baik, meskipun peningkatan performa setelah hyperparameter tuning tidak sebesar *XGBoost*. *XGBoost Regression* dengan *hyperparameter tuning* memberikan prediksi harga rumah yang paling akurat di antara semua model yang diuji. *Hyperparameter tuning* terbukti mampu meningkatkan kinerja model secara signifikan dan sangat disarankan sebagai langkah penting sebelum pengujian model. Penelitian ini menunjukkan bahwa pemanfaatan *Machine Learning*, khususnya dengan teknik *hyperparameter tuning*, dapat memberikan hasil yang akurat dalam memprediksi harga rumah. Hal ini memberikan wawasan berharga bagi investor, pemilik rumah, dan pembuat kebijakan dalam membuat keputusan yang lebih baik terkait investasi dan perencanaan keuangan di pasar properti Jakarta.

Daftar Pustaka

[1] H. Nadi, and B. Jason, “Implementasi Machine Learning Untuk Prediksi Harga Rumah Menggunakan Algoritma Random Forest, “Computatio: Journal of Computer Science and Information Systems. vol. 8, pp. 50-61, April 2024

[2] H. Tarak, and Athal, “Visualization and Explorative Data Analysis,”International Journal of Enhanced Research in Science, Technology & Engineering, vol. 12, pp. 11–21, March 2023.

[3] J. beno, “Prediksi Harga Saham Bank BCA Menggunakan XGBoost.”ARBITRASE: Journal of Economics and Accounting, vol. 3, pp. 231–237, November 2022.

[4] L. S. Ega, and A.Ida, “Penerapan Random Forest Regression Untuk Memprediksi Harga Jual Rumah Dan Cosine Similarity Untuk Rekomendasi Rumah Pada Provinsi Jawa Barat, “Jurnal Ilmiah FIFO, vol 14, pp. 131-146, November 2022

[5] M. L. Muhammad, D. A. Sekar, Z. N. Hanan, M. Toni, W. Rio, “Analisis Prediksi Harga Rumah Sesuai Spesifikasi Menggunakan Multiple Linear Regression,”Jurnal Informatik, vol.17, pp.238-245, Desember 2021

[6] S. Andi, A. Septi, G. Aris, “Prediksi Harga Rumah Menggunakan Web Scraping Dan Machine Learning Dengan Algoritma Linear Regression,” Jurnal Teknik Informatika dan Sistem Informasi, vol. 8, pp. 41–50, Maret 2021.

[7] Y. H. E. Sri, S. Oni, and S. Yuana, “Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit ,”Journal of Mathematics: Theory and Applications, vol. 4, pp. 21–26, 2022.