SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, June 2025, pp.208 - 215
ISSN 2407-0939 print/ISSN 2721-2041 online

Property Price Prediction Using the Random Forest Regression Algorithm

Putri Utami[1], Muhamad Jundi[2], Rahmaddeni[3], Leonardo Sinaga[4]
[1.2.3.4] Department of Informatic Engineering, Universitas Sains dan Teknologi Indonesia
Jl. Purwodadi KM. 10, Pekanbaru, Riau, Indonesia. Zip code 28294
Email: 241705280290@usti.ac.id[1] , 2417052802089@usti.ac.id[2] , rahmaddeni@usti.ac.id[3]
2417052802105@usti.ac.id[4]

ABSTRACT

This study uses an innovative approach to predicting property prices through machine learning with the Random Forest Regression method. The data set was obtained from Kaggle and contains a total of 500 rows with 12 attributes, including 10 numerical attributes and 2 categorical attributes. The evaluation results, measured using the R^2 score on the test dataset, indicate strong performance, achieving the highest R^2 score of 81.88% with a dataset split ratio of 90:10. Although there is a slight difference between the predicted and actual values, the scatter plot visualization shows that the model's predictions generally approach the actual values, indicating good accuracy. The graph of the training data and actual data reflects no significant signs of overfitting or underfitting. This indicates the good accuracy of Random Forest Regression in predicting house prices and its ability to capture the relationship between independent and dependent variables effectively.

Keywords: *Machine Learning, Overfitting, Random Forest Regression, Scatter plot, Underfitting.*

Introduction

Property and real estate companies, particularly in the housing sector, have significant potential in the business and investment world. Factors that influence this include the continuous shortage of housing supply compared to the growing demand and needs of the population, as well as relatively low interest rates on public housing loans [1]. This situation drives property developers to compete in selling houses to the public, either for residential needs or investment purposes. As a result, prospective homebuyers consider whether the house they intend to purchase offers good profitability or not [2], as many houses located in non-strategic areas are priced high and do not align with their actual value [3].

Therefore, it is crucial for individuals who intend to buy a house to predict the price of the property they wish to purchase, as this often requires in-depth and complex analysis. One way to overcome this challenge is by utilizing machine learning techniques, which can process large amounts of data and provide more accurate price predictions. The use of machine learning techniques enables learning with more complex layers to achieve high accuracy [4].

One of the best approaches used for prediction is the Random Forest Regression model. This method is chosen for its ability to handle complex and multi-dimensional data [5]. It is a type of ensemble learning, which means it combines multiple models to achieve more accurate predictions. Essentially, the Random Forest Regression model is built by combining multiple Decision Trees, where each tree acts as an independent decision-maker [6]. The advantage of Random Forest lies in its ability to handle dependencies and interactions between complex variables while also effectively addressing overfitting [7].

The Random Forest Regression-based approach in property price estimation leverages the algorithm's ability to handle various types of data with high complexity, such as property physical characteristics and location attributes. Features used, such as building area, number of rooms, accessibility to the city center, and surrounding public facilities, all contribute to generating more accurate price predictions. In the property market, which heavily depends on local dynamics, Random Forest can capture non-linear relationships and interactions between variables that traditional models struggle to handle. Thus, the use of this technique not only enhances price prediction accuracy but also provides deeper insights into the factors influencing property prices. This approach is essential for helping property developers, real estate agents, and buyers understand price movements more objectively and data-driven, while also creating opportunities for innovation in leveraging technology to enhance the efficiency of the real estate market. Through this approach, the property market can become more transparent, with prices more accurately reflecting the real value of a property based on objective data. It also opens opportunities for improvements in decision-making, whether for investors, developers, or the public, who increasingly require more efficient and reliable methods for assessing property value.

Research Methods

This study aims to develop a house price prediction model using the Random Forest Regression model to determine whether it can accurately predict prices. The model was built through a series of steps, including data collection, data preprocessing, model training, evaluation, and storing the best-performing model. The evaluation was conducted using several performance metrics to ensure prediction accuracy, which was then used to predict new property prices. In other words, this study uses an applied research method aimed at finding answers to specific questions.

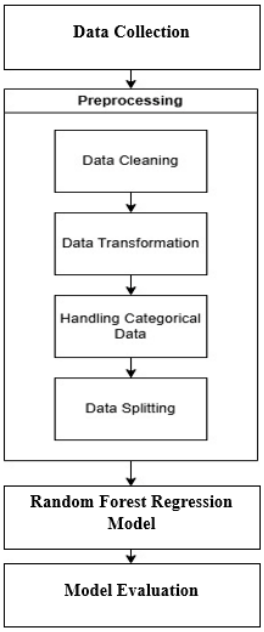

Figure 1. Research methodology

Data Collection

In this study, the Kaggle dataset was used. The values of each house vary, including land area, house location, number of bedrooms, number of bathrooms, living room area, available facilities in the surrounding environment, and so on. Due to the diverse values of each house, this is what causes house prices to become increasingly varied [8]. Therefore, the selected dataset consists of sources relevant to the research topic, as it contains information on house selling prices and various physical property specifications, such as house size, number of bedrooms, number of bathrooms, and the year the house was built, which affects house prices. However, it is important to note that this dataset has several limitations, such as the absence of certain features that may be highly relevant to house prices. For instance, building quality, crime rates in the surrounding neighborhood, or proximity to public facilities are not included in the dataset. Features such as *Has_Garden*, *Has_Pool*, or *Garage_Size* may not fully represent the actual conditions of a house. An explanation of the attributes used can be found on the table below.

Table 1. Dataset

No.	Attribute	Description	Type
1.	*ID*	Unique identification for each property	Numeric
2.	*Square_Feet*	Property building area in square feet	Numeric
3.	*Num_Bedrooms*	Number of bedrooms in the property	Numeric
4.	*Num_Bathrooms*	Number of bathrooms on the property	Numeric
5.	*Num_Floors*	Number of floors on the property	Numeric
6.	*Year_Built*	The year the property was built	Numeric
7.	*Has_Garden*	Whether the property has a garden or not	Categorical
8.	*Has_Pool*	Whether the property has a swimming pool or not	Categorical
9.	*Garage_Size*	Garage capacity in terms of the number of vehicles	Numeric
10.	*Location_Score*	Property location score	Numeric
11.	*Distance_to_Center*	Distance from the property to the city center	Numeric
12.	*Price*	Property price	Numeric

Preprocessing

Preprocessing is the process used to clean, transform, and prepare data before further processing. Data preprocessing is carried out to improve data quality, remove incomplete or invalid values, modify data formats, and adjust the data to match the format required by the model or algorithm to be used [9]. Furthermore, data cleaning is essential to ensure that the model does not receive incomplete data by handling missing values to prevent distortions in the model. In this data-cleaning process, several steps were taken to produce a clean dataset so that it could be used in the next stage with the aim of obtaining useful information and modifying predefined data to be ready for the subsequent processing phase [10]. This technique is used to address issues such as missing values, redundant data, outliers, or data formats that do not align with the model, as these problems can interfere with the output results [11].

After the data cleaning stage, the next equally important step is data transformation. One of the most used normalization methods is decimal scaling, a data transformation method that normalizes values to ensure a consistent range across all attributes by shifting the decimal point in the desired direction. This process was performed to improve data quality and enhance the model's prediction accuracy [12]. Next, categorical data handling was conducted using the one-hot encoding method. Encoding is the process of converting categorical data into a numerical format that can be processed by the model. This step is critical to ensure that the machine learning model can efficiently process the data and avoid bias in categorical variables [13].

Data splitting is a technique used to divide a dataset into two main sets: the training data and the testing data. This division allows the model to be trained on one subset of the data while testing it on another subset that was not used during training. This step is essential to prevent overfitting and to ensure that the model can generalize well to unseen data [14]. Generally, the data is split using ratios of 80:20 [15], 70:30 [16], 60:40, and 90:10 [17] between training and testing data.

Random Forest Regression Model

Random Forest Regression is an ensemble model consisting of multiple decision trees that work collectively to generate predictions. The Random Forest model was developed in the 1990s and has been widely recognized for its advanced capabilities in classification and/or regression, as well as its ability to handle categorical or continuous variables and manage missing data. The Random Forest model has gained rapid popularity in the industry due to its strong performance across various applications. Some of its popular applications include predictive modeling for error diagnosis and root cause analysis, change point detection in data, and diagnostic modeling for improving information retrieval from models. Random Forest is capable of handling complex, non-linear data with many features, making it suitable for various predictive and modeling problems across different industries. This model works by constructing multiple decision trees, each trained using a random subset of the training data. The final prediction is obtained by aggregating the predictions from all trees, which can be averaged in the case of regression. This characteristic makes Random Forest more robust and reduces overfitting or underfitting compared to a single decision tree. This algorithm is unique because it can combine predictions from all trees to generate a final prediction that is more accurate and reliable [18].

Model Evaluation

In evaluating the Random Forest Regression model that was developed, several evaluation methods were used, such as R^2 Score, Mean Absolute Error (MAE), and Mean Squared Error (MSE) [19]. Based on the evaluation results obtained from MAE and MSE, the smaller the resulting value, the better the evaluation. In contrast, for the R^2 Score, a coefficient of determination value closer to 1 indicates a better estimation result [20]. This study used three evaluation methods: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score. MAE is useful for measuring the average absolute difference between the actual value and the model's prediction. MAE was calculated by summing all the absolute differences between actual values and predictions and then dividing by the total number of data points.

$$MAE = \frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{n} \quad (1)$$

In this formula, y_i represents the actual value, $\hat{y}$represents the predicted value, and n is the total number of data points. Mean Absolute Error (MAE) provides a simple measure of prediction accuracy, where a lower value indicates better model performance. Next, Mean Squared Error (MSE) is a metric that measures the average squared difference between actual values and the model's predictions. MSE is calculated by summing all the squared differences between actual values and predictions, then dividing by the total number of data points.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (2)$$

In this formula, y represents the actual value, $\hat{y}$ represents the predicted value, and n is the total number of data points. Mean Squared Error (MSE) provides a measure of how large the prediction errors produced by the model are. The prediction results for MAE and MSE indicate that the smaller the values, the better the prediction performance. In contrast, for the R^2 Score, a coefficient of determination value closer to 1 means that the independent variables provide all the necessary information for estimating the dependent variable.

Results and Discussion

Data Collection

The data set used in this study contains property data, as described in Table 1, and was obtained from Kaggle. This dataset consists of 12 attributes that include various property specifications such as building area, number of bedrooms, number of bathrooms, number of floors, year built, availability of a garden, availability of a swimming pool, garage size, location score, and distance to the city center. The total number of records in this dataset is 500 rows, with each row representing a single property unit. Figure 2 below displays the top 10 data entries, which were extracted using Python through the read excel function with Pandas and the head () method.

	ID	Square_Feet	Num_Bedrooms	Num_Bathrooms	Num_Floors	Year_Built	Has_Garden	Has_Pool	Garage_Size	Location_Score	Distance_to_Center	Price
0	1	143.635030	1	3	3	1967	1	1	48	8.297631	5.935734	602134.816747
1	2	287.678577	1	2	1	1949	0	1	37	6.061466	10.827392	591425.135386
2	3	232.998485	1	3	2	1923	1	0	14	2.911442	6.904599	464478.696880
3	4	199.664621	5	2	2	1918	0	0	17	2.070949	8.284019	583105.655996
4	5	89.004660	4	3	3	1999	1	0	34	1.523278	14.648277	619879.142523
5	6	88.998630	5	3	2	1959	1	1	36	8.994552	17.633250	670386.804433
6	7	64.520903	4	3	1	1938	0	1	32	7.101354	2.429908	523827.125601
7	8	266.544036	5	1	3	1973	1	1	39	9.373784	12.692785	875352.545188
8	9	200.278753	5	1	1	1988	1	1	32	6.032918	11.642876	738269.852342
9	10	227.018144	3	2	1	1917	0	0	29	4.734009	2.368301	490552.681240

Figure 2. Top 10 Data

From Figure 2, the top 10 data entries can be seen, consisting of 12 attributes. Of these 12 attributes used in this study, 2 attributes have categorical data types, while 10 attributes have numerical data types. The total number of data points used is 500.

Preprocessing

Before performing model prediction, the data preprocessing stage must be carried out. The preprocessing step functions to adjust the data into the format required by the system. In this study, only data reduction was performed, which involved removing attributes that were not needed by the system. The attribute removed was "ID", as shown in Figure 3 because the ID feature is not required for model testing. Below is the implementation in Python.

```
# Step 3: Model Building
# Split the data into training and testing sets
X = data.drop(columns=[target, 'ID'])  # Assuming 'ID' is not useful for prediction
y = data[target]
```

Figure 3. Data Reduction in Python

The model evaluation process was carried out using four test-to-training data ratios, namely 90:10, 80:20, 70:30, and 60:40, from a total dataset of 500 rows. The data splitting process can be seen in Figure 3 below, where `X_train` and `y_train` will be used as training data to fit the model. The dataset ratio split, where `test_size=0.1` represents 10% of the data used as test data, while 90% of the total data will be used as training data. The test size value can be adjusted according to the desired splitting ratio. Syntax `random_state=42` is used to ensure reproducibility, allowing the dataset to be split in the same way when running the script.

```
# Splitting Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

Figure 4. Splitting Data in Python

Hyperparameter Optimization

The optimization process was performed using training data, which was repeatedly applied to all grids with the aim of obtaining the best-scoring combination. In completing this study, the method that was used was

GridSearchCV. The implementation of GridSearchCV in Random Forest Regression is shown in Figure 5 below. Hyperparameter optimization is useful for evaluating model performance across various data subsets to prevent overfitting and underfitting. Overfitting occurs when the model is too complex and learns too many details or noise from the training data, resulting in very high accuracy only on the training data. Underfitting arises when the model is too simple and fails to capture patterns in the training data, resulting in poor performance on both the training and test data. The GridSearchCV method can be used to optimize hyperparameters and select parameters that help prevent overfitting or underfitting. The maximum tree depth and the number of trees play a crucial role in preventing both issues.

```
# Step 4: Hyperparameter Tuning
param_grid = {
    'model__n_estimators': [100, 200, 300],
    'model__max_depth': [None, 10, 20, 30],
    'model__min_samples_split': [2, 5, 10],
    'model__min_samples_leaf': [1, 2, 4]
}

grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1, verbose=2)
grid_search.fit(X_train, y_train)

# Best parameters
print("Best Parameters:", grid_search.best_params_)
```

Figure 5. Hyperparameter Tuning

In Figure 5 above, the hyperparameter grid search includes n_estimators, max_depth, min_samples_split, and min_samples_leaf. Then, 5-fold cross-validation was used, and MSE (Mean Squared Error) was applied as the model evaluation metric. The parameter n_jobs = -1 means that all CPU cores were utilized to speed up the search process.

Model Evaluation

After the data splitting stage was completed, the data was ready to be processed using the Random Forest Regression model. Below is the Python implementation for using the Random Forest Regression model, as shown in Figure 6.

```
# Define Random Forest Regressor
model = RandomForestRegressor(random_state=42)
```

Figure 6. Random Forest Regression Model Invocation

This study used several evaluation methods, including Scatter Plot, Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score. MAE was used to measure the average absolute difference between the actual values and the model's predictions. MAE was calculated by summing all the absolute differences between the actual values and predictions, then divided by the total number of test data in the 90:10 split.

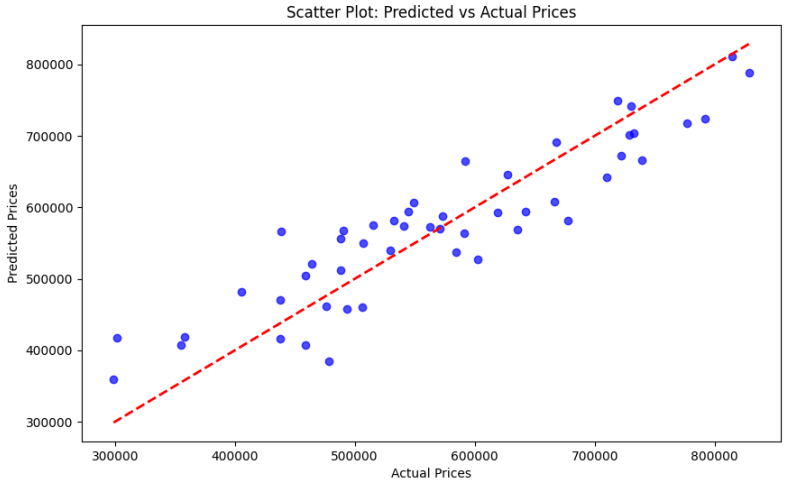

Figure 7. Visualization of Actual vs. Predicted Results (90:10 Test Data)

In Figure 7, the model's predicted values (red diagonal line) closely align with the actual values (blue dots), demonstrating that the model provides good predictions. Additionally, the prediction results are also presented in terms of R^2 Score, Mean Absolute Error (MAE), and Mean Squared Error (MSE), as shown in Figure 8 below.

```
Mean Absolute Error (MAE): 48184.5960850895
Mean Squared Error (MSE): 3073027449.1640625
R² Score: 0.8188430799231146
```

Figure 8. Prediction Results for 10% Test Data

Based on Figure 8, the 90:10 test data split achieved an R^2 Score of 81.88%, with an MAE of 48.184 and MSE of 3,073,027,449. In the data splitting explanation, this study did not rely on a single test data split but also performed multiple test data splits, such as the 60:40 test split, which resulted in an R^2 Score of 81.18%. This value is 0.7% lower than the R^2 Score of the 90:10 test split but higher than the R^2 Scores of the 80:20 and 70:30 test splits. For the 60:40 test split, the MAE was 42.792, and the MSE was 2,726,761,729. More detailed results from these four test data splits can be seen in Table 2 below:

Table 2. Evaluation results

No.	Data Splitting	MAE	MSE	R2 Score
1.	90:10	48184.59608508950	3073027449.164062	0.81884307
2.	80:20	43658.24707434616	2847269193.342156	0.81070998
3.	70:30	45463.47180352078	3017049368.696039	0.79027132
4.	60:40	42792.27546787269	2726761729.625462	0.81180044

From Table 2 above, the R^2 Score in percentage form for the Random Forest Regression model in predicting house prices reached the highest value of 81.88% from the 90:10 training data ratio. For the R^2 Score, the 81.88% value is the closest to 1, indicating the best estimation quality. In this 90:10 ratio, most of the data was used to train the model, meaning the model had more information to learn from. This could result in a better-performing model for predicting training data, based on the evaluation results. Below is a comparison of actual values and predicted values in the training data, which can be seen in Table 3.

Table 3. Comparison of actual values and predicted values

ID	Actual	Predicted
336	828,686.474457	788,767.068814
76	814,846.453707	811,183.609563
316	791,677.204598	724,428.777199
185	776,835.044348	718,049.395150
450	739,265.736526	666,435.304988
388	732,651.692569	703,498.951791
104	730,197.480607	741,543.362512
490	728,995.977430	701,426.304316
408	721,719.204904	672,892.879927
280	718,789.195061	749,235.942554
333	710,152.564843	642,083.802685
11	677,623.466782	581,721.780804
73	667,693.000276	691,734.691340
475	666,405.154193	607,908.719402
101	642,579.094167	593,877.857456
415	635,614.195888	568,558.108702
70	627,395.372381	645,623.407080
440	618,783.083265	592,412.628870
0	602,134.816747	527,440.456702
211	591,635.644373	664,443.256594
361	591,058.708950	563,331.579081
461	584,792.068007	536,936.521497
124	573,486.145827	587,590.174765

371	571,107.834475	569,611.335240
77	562,584.035120	572,775.196417
485	549,076.808610	606,952.489698
491	544,610.970482	593,474.171680
384	540,866.883935	574,412.884166
374	532,377.580195	582,057.601039
68	529,467.599267	540,095.142989
394	515,544.254783	575,690.515584
33	506,721.273236	549,397.691761
63	506,038.288780	460,173.600174
194	493,341.254380	458,229.490197
9	490,552.681240	568,000.573162
495	488,496.350722	511,491.246150
93	488,035.884886	556,654.368732
155	478,489.299027	385,238.825414
173	475,921.448459	462,025.089311
2	464,478.696880	521,039.766614
84	458,704.412271	407,010.366402
334	458,567.790955	504,284.470379
30	438,805.869608	566,867.195914
406	437,797.486635	470,933.256882
22	437,751.643015	416,497.526104
497	405,324.950201	481,845.691523
209	358,006.550936	418,655.587525
377	354,991.349233	407,661.427813
356	301,823.920507	418,029.296245
409	298,871.665267	359,504.694361
336	828,686.474457	788,767.068814

The actual values and predicted values in Table 3 above show that the values in the predicted column closely align with those in the actual column. The visualization results using the scatter plot in Figure 7 make it easier to observe how closely the predicted values approach the actual values. If the points on the scatter plot are close to the diagonal line, the model tends to have good prediction accuracy. Although there is still a slight difference between the predicted prices, this model is overall reliable for analysis and planning in predicting house prices. Although Random Forest is a reliable model, having too many irrelevant variables or features can make the model overly complex and increase the risk of overfitting. This can reduce the model's ability to generalize to unseen data. To overcome these limitations, it is often necessary to use other modeling approaches, data preprocessing techniques, or further refinements such as more careful feature selection and hyperparameter tuning.

Conclusion

Based on the results of this study, Random Forest Regression has been proven to be effective in predicting property prices. The highest R^2 Score, obtained from the calculations, reached 81.88% with a 90:10 dataset split between test and training data. Although there is a slight difference between the predicted and actual values, the scatter plot visualization of predicted and actual values shows the model's predictions generally aligned with the actual values, indicating high accuracy in Random Forest Regression for predicting house prices. Overall, this algorithm has been proven to effectively capture the relationship between independent and dependent variables. When compared to other methods, such as Linear Regression, which tends to be simpler and less capable of capturing non-linear relationships, and Gradient Boosting Machines (GBM), which excel in precision but are more complex in terms of tuning, Random Forest remains a strong choice for this problem. This is due to its ability to handle non-linear and complex features effectively without requiring intricate modeling. Nevertheless, there is still room for improvement in terms of accuracy, and exploring other methods, such as GBM, XGBoost, or deep learning, could yield more optimal results for certain datasets with more precise tuning.

References

[1] K. Anam, M. Nurfadillah, and F. Fauziah, "Analisis kinerja keuangan terhadap return saham perusahaan properti dan real estate Indonesia," *Jurnal Daya Saing*, vol. 7, no. 3, pp. 123-135, 2021.

[2] E.F. Rahayuningtyas, F.N. Rahayu, and Y. Azhar, "Prediksi harga rumah menggunakan general regression neural network," *Jurnal Informatika*, vol. 8, no.1, pp. 59-66, 2021.

[3] N. P. Nainggolan, and Heryenzus, "Analisis faktor-faktor yang mempengaruhi minat beli konsumen dalam membeli rumah di Kota Batam," *Jurnal Ilmiah Manajemen dan Bisnis*, vol. 19, no. 1, pp. 41-54. 2018.

[4] L.P. Nasyuli, Lubis, and A. M. Elhanafi, "Penerapan model machine learning algoritma gradient boosting dan linear regression melakukan prediksi harga kendaraan bekas," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 2, no. 2, pp. 299-310, 2023.

[5] N. Yusuf, "Prediksi produksi daging sapi di Indonesia menggunakan random forest regression: Analisis data 2018-2025," *Jurnal JUIT*, vol. 3, no. 2, pp 134-142, 2024.

[6] M. A. Pratama, M. Munawaroh, and W. J. Pranoto, "Perbandingan performa algoritma linear regresi dan random forest untuk prediksi harga bawang merah di Kota Samarinda", *Jurnal Tektonik*, vol. 3, no. 2, pp. 172-182, 2024.

[7] H. Tantyoko, D. K. Sari, and A. R. Wijaya, "Prediksi Potensial Gempa Bumi Indonesia Menggunakan Metode Random Forest Dan Feature Selection," *Jurnal Informasi System* vol. 6, no. 2, pp. 83–89, 2023.

[8] V. A. Priliaputri, "Perbandingan kinerja algoritme naïve bayes dan k-nearest neighbor (KNN) untuk prediksi harga rumah," Fakultas Teknik, Departemen Teknik Komputer, *Universitas Diponegoro, Semarang*, 2022.

[9] L. U. Hasanah, I. Maula, and A. Tholib, "Analisis prediksi harga rumah di Jabodetabek menggunakan multiple linear regression," *Jurnal Informatika Kaputama*, 2023.

[10] B. S. Purnomo, and P. T. Prasetyaningrum, "Penerapan data mining dalam mengelompokkan kunjungan wisatawan di Kota Yogyakarta menggunakan metode K-means," *JCS-TECH: Journal of Computer Science and Technology*, vol. 1, no.1, pp. 27-32, 2021.

[11] S. Y. Kurniawan, "Klasifikasi kelayakan air minum dengan backpropagation neural network berbasis penanganan missing value dan normalisasi," *Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru*, 2024.

[12] M. R. Kusnaidi, T. Gulo, and S. Aripin, "Penerapan normalisasi data dalam mengelompokkan data mahasiswa dengan menggunakan metode K-means untuk menentukan prioritas bantuan uang kuliah tunggal," *Journal of Computer System and Informatics*, vol. 3, no.4, pp. 330-338, 2022.

[13] F. Septian, "Optimasi klusterisasi pada lama tempo pekerjaan berbasis gradient boost algorithm," *Indonesian Journal of Information Technology*, 2023.

[14] R. Ramadhania, R. Ramadhanua, A. F. Artha Abdillah, and M. Ridwana, "Studi Komparatif Multinomial Naïve Bayes, Decision Tree, dan K-Nearest Neighbor dalam Klasifikasi Validasi Ulasan Clash of Clans oleh Pengguna Ahli," *Jurnal Sains dan Teknologi Indonesia*, vol. 12, no. 4,pp. 653-660, 2024.

[15] Anisatuzzumara, "Implementasi Latent Dirichlet Allocation (LDA) dan K-Nearest Neighbors (KNN) pada Sistem Rekomendasi Jurnal Terindeks GARUDA," *Universitas Islam Sultan Agung Semarang*, 2024.

[16] F. Ibrahim, "Perbandingan Performa Algoritma Random Forest Classifier dan Naive Bayes pada Penyakit Diabetes Melitus*," Universitas Islam Negeri Syarif Hidayatullah Jakarta*, 2024.

[17] A. M. Sarah, B. Kurniadi, and E. Warsini, "Implementasi metode regresi linear dalam memprediksi penyakit anemia secara dini," *Jurnal Teknologi Komputer dan Sistem Informasi*, vol. 3, no.1, pp. 14-23, 2023.

[18] M. A. Pratama, M. Munawaroh, and W. J. Pranoto, "Perbandingan Performa Algoritma Linear Regresi dan Random Forest untuk Prediksi Harga Bawang Merah di Kota Samarinda," *Jurnal Tektonik*, vol. 1, no. 2, pp. 172-182, 2024.

[19] J. Hutahaean, D. Yusup, and Purwantoro, "Perbandingan metode linear regression, random forest & k-nearest neighbor untuk prediksi produksi hasil panen padi di Provinsi Jawa Barat," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 3, pp. 38-95, 2024.

[20] M. A. Sembiring, "Analisis faktor prediksi diagnosis tingkat serangan jantung menggunakan metode regression," *Jurnal Teknologi Komputer dan Sistem Informasi*, vol. 4, no.1, pp. 16-22, 2024.