International Journal of Electrical Engineering, Mathematics and Computer Science

E- ISSN: 3048-1945
P- ISSN: 3048-1910

Implementation of the Extreme Gradient Boosting(XGBoost) Method in the Classification of Recipients of Habitable Housing Rehabilitation in Central Aceh Regency

Ira Zulfa [1*], Hendri Syahputra[2], Fitranuddin [3], Adellia Divandariga S[4]

[1] Universitas Gajah Putih, Indonesia ; ira.zulfaa@gmail.com
[2] Universitas Gajah Putih, Indonesia ; andreseptian905@gmail.com
[2] Universitas Gajah Putih, Indonesia ; fitranuddin1234@gmail.com
[2] Universitas Gajah Putih, Indonesia ; adeliadivariga@gmail.com
* Corresponding Author : Ira Zulfa

Abstract. In Central Aceh Regency, many households still live in uninhabitable conditions. The government is running a program to rehabilitate habitable houses, but the selection of recipients is still done manually, causing inefficiency and inconsistency. This study implements the Extreme Gradient Boosting (XGBoost) algorithm to classify aid recipients automatically and accurately. Using a machine learning approach, data is collected based on variables of structural conditions, building materials, ventilation, lighting, and sanitation. Hyperparameter tuning is performed to optimize model performance. The implementation results show that XGBoost is able to support fair, efficient, and transparent decision making in housing assistance programs.

Keywords: XGBoost; Classification; Habitable Houses; Rehabilitation; Machine Learning

1. Introduction

house is a basic need that must be met to ensure a good quality of life for every individual and family. In Central Aceh Regency, many households live in less than adequate conditions, both in terms of the physical structure of the house and the adequacy of the surrounding environment. The local government has attempted to carry out a livable house renovation program to improve the quality of life of the community.

Extreme Gradient Boosting (XGBoost) method has become one of the most popular and effective machine learning algorithms for classification and regression tasks. XGBoost is known for its ability to overcome overfitting, computational efficiency, and its ability to handle data with complex characteristics. To determine whether it is sufficient or not, residents must meet the standards that have been applied by the sovereign in the form of house conditions (buildings) that cover the conditions of the Room Area, Condition Type, Floor, Roof Condition, Final Waste (WC), and Drinking Water Source [1] . Several previous studies have shown the effectiveness of the XGBoost method in various fields of classification. Comparing tree-based classification methods such as Decision Tree, Random Forest, Gradient Boosting, and XGBoost to classify Body Mass Index (BMI) based on features such as gender, height, and weight [2] . Comparison between XGBoost and Decision Tree in music genre classification using the GITZAN dataset with accuracy results of up to 72% [3] . Meanwhile, Yulianti (2022) applied XGBoost for credit card customer classification to detect the risk of bad credit, with high accuracy results after tuning using GridSearchCV [4] . Admajayanti (2021) used the AHP and SAW methods for a decision support system in a livable housing assistance program, focusing on a combination of weight-based criteria. Finally, Shafila (2020) used XGBoost for bioinformatics data classification in a case study of the Ebola disease, highlighting the strength of this algorithm in handling complex and high-dimensional data. The similarities between the five studies and this study lie in the use of the XGBoost algorithm for classification and the importance of evaluating model performance, although applied to different domains [5] .

Received: April 30, 2025
Revised: May 15, 2025
Accepted: June 11, 2025
Published: June 13, 2025
Curr. Ver.: June 13, 2025

Rehabilitation eligibility criteria often involve various factors such as the physical condition of the house, the number of occupants, the family's income level, and so on. Decisions made manually can be time-consuming and resource-intensive, and potentially inconsistent. Therefore, a system is needed that can help classify habitable houses automatically and accurately.

2. Preliminaries or Related Work or Literature Review

This literature review is part of a scientific work (such as a thesis, dissertation, or dissertation) which functions to review and analyze literature relevant to the research topic.

Extreme Gradient Boosting (XGBoost)

The XGBoost method is a development of gradient boosting proposed by Dr. Tianqi Chen from the University of Washington in 2014. Gradient boosting is an algorithm that can find optimal solutions to various problems, especially in regression, classification and ranking. The basic concept of this algorithm is to adjust the learning parameters repeatedly to reduce the loss function (model evaluation mechanism). XGBoost uses a more regular model to build a regression tree structure, so that it can provide better performance and reduce model complexity to avoid overfitting. The final prediction result of XGBoost is the sum of the prediction results from each regression tree. Decision tree-based algorithms have good performance on data with categorical features and do not have much effect on data with unbalanced classes [4] .

Rehabilitation

Rehabilitation is the process of returning something to normal or at least having a replacement [6] **.** It is known that poverty will have an impact on basic human needs that are not met, including physical, psychological, social and spiritual needs. One of them is the unfulfilled need for decent housing. This occurs because of the inability of the community to meet decent housing because the community's standard of living is poor and the community's knowledge about realizing decent housing is still limited, so that the community still has difficulty building a house that is considered habitable [7] .

Rutilahu Social Rehabilitation is a process of restoring the social function of the poor through efforts to improve the condition of Rutilahu, either partially or completely, which is carried out through mutual cooperation in order to create a house condition that is suitable as a place to live. The Rutilahu rehabilitation program aims to rehabilitate uninhabitable houses, increase the comfort of the house, and foster the values of mutual cooperation, participation, concern and social solidarity [8] **.**

Data Classification

Classification is the process of placing a member into a particular class, but the class is determined first, then the members of the class are placed in a class based on the same data characteristics [9] .

Data classification is the process of finding models or functions that describe and distinguish data classes and their concepts [10] .

Data classification is the process of separating and organizing data into relevant groups ("classes") based on shared characteristics, such as their sensitivity level, the risks they pose, and the compliance regulations that protect them. To protect sensitive data, it must be located, classified according to its sensitivity level, and tagged accurately. Then, companies must handle each class of data in a way that ensures only authorized individuals can gain access, both internally and externally, and that the data is always handled in full compliance with all relevant regulations

Python

Python is one of the high-level programming languages that is interpreted, interactive, *object-oriented* and can operate on almost all platforms such as *the Linux family, Windows, Mac* , and other platforms. *Python* is one of the high-level programming languages that is easy to learn because of its clear and elegant syntax, combined with the use of modules that have high-level data structures, are efficient, and ready to use immediately. *The source code* of the application in the *Python programming language* will usually be compiled into an intermediate format known as *byte code* which will then be executed [11] .

Flowchart

Flowchart is a way of writing algorithms using graphic notation. *Flowchart* is a picture or diagram that shows the sequence or steps of a program and the relationship between processes and their statements [12] . Flowchart also describes the logical sequence of a problem-solving procedure, so that flowchart can be understood as problem-solving steps written in certain symbols. And this flowchart will represent the flow in the program logically [13] .

Visual Studio Code

Visual Studio Code can be used for various programming languages such as JavaScript, HTML, CSS, PHP, Python, C++, and many more. *Visual Studio Code* works on various operating systems such as Windows, macOS, and Linux. In addition, *VisualStudio Code* provides a *Live Share feature* that allows multiple developers to work on the same project simultaneously from different locations [14] .

3. Proposed Method

This method includes data preprocessing, categorical variable encoding, and classification using XGBoost. Evaluation is done with accuracy, precision, recall, and F1-score metrics. Hyperparameter tuning is done using GridSearchCV. The research method used in this study is a quantitative approach with explanatory and evaluative research types. This study aims to classify habitable houses that are eligible for rehabilitation using the XGBoost algorithm, explain the factors that influence house eligibility, and evaluate the effectiveness of the habitable house rehabilitation program in Central Aceh Regency.

Confusion Matrix

Confusion Matrix is a classification model evaluation method in the form of a matrix table to compare model prediction results with actual data. This matrix provides a detailed description of correct and incorrect predictions for each class, making it easier to analyze model performance [15] **.** ***This*** *Confusion Matrix* is an evaluation tool used in classification problems to assess model performance. This matrix provides an overview of how the classification model works on test data by comparing model predictions to actual values.

Table 1. Contingencies

Classification	True Livability = Yes	True_Livable = No	Total
Habitable Prediction = Yes	***True Positive (TP)***	***False Positive (FP)***	**TP+FP**
Habitable Prediction = No	***False Negatives (FN)***	***True Negative (TN)***	**FN+TN**
Total	***True Positive (TP) + False Negatives (FN***	***False Positives (FP) + True Negative (TN)***	**TP+FP+FN+TN**

Where:
True Positives (TP) - instances where Habitable = Yes and predicted as Yes
False Positives (FP) - instances where Habitable = No and predicted as Yes
False Negatives (FN) - instances where Habitable = Yes and predicted as No
True Negatives (TN) - instances where Habitable = No and predicted as No
Then based on *the instances* or objects from the dataset, it can be calculated that:
(TP): 63
(FP): 24
(FN): 13
(TN): 50

Table 2. Confusion Matrix Prediction Table

Classification	Prediction Habitable = Yes	Prediction Habitable = No
True Livable = Yes	63	13
True_Livable = No	24	50

confusion matrix prediction table above, here are the manual calculations for the levels of accuracy, precision, *recall* , and *f1-score.*

Accuracy, one of the most commonly used evaluation metrics to assess the performance of a classification model. Accuracy measures the percentage of correct predictions out of the total predictions made by the model.

Accuracy formula:

$$\text{Accuracy} = \frac{\textit{True Positives} + \textit{True Negatives}}{\text{Total Observations}}$$

$$\text{Accuracy} = \frac{TP + TN}{\text{Total Observations}}$$

$$= \frac{113}{150}$$

Accuracy = 0.7467 = 74.67%

Information:

True Positive (TP) : Number of correct positive predictions (class positive that is truly positive).

True Negatives (TN) : Number of correct negative predictions (class negative that is truly negative).

False Positives (FP) : Number of false positive predictions (class negative predicted as positive).

False Negatives (FN) : Number of incorrect negative predictions (class positive predicted as negative).

Precision , *precision* measures how accurate the model's predictions are by showing the percentage of correct positive predictions.

$$\textit{Precision} = \frac{\textit{True Positives}}{\textit{True Positives} + \textit{False Positives}}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$= \frac{63}{87}$$

Precision = 0.7241 = 72.41%

Recall , The proportion of positive cases that were actually detected by the model.

$$\textit{Recall} = \frac{\textit{True Positives}}{\textit{True Positives} + \textit{False Negatives}}$$

$$\textit{Recall} = \frac{TP}{TP + FN}$$

$$= \frac{63}{76}$$

Recall = 0.8289 = 82.89%

F1-Score, the harmonic mean of precision and recall. *The F1-Score value* ranges between 0 and 1, where 1 is the best value.

$$F1\text{-}Score = 2 \times \frac{Precision + Recall}{Precision \times Recall}$$

$$F1\text{-}Score = 2 \times \frac{0.7241 + 0.8289}{0.7241 \times 0.8289}$$

$$= 0.7742 = 77.42\ \%$$

F1-Score is used to provide an overview of the balance between precision and recall. This is especially useful when there is an imbalance between positive and negative classes.

Model Implementation, the model that has been evaluated and declared good will be implemented to classify new data. The model works quite well, but it is not perfect. There is room for improvement, especially in terms of precision, because the model still makes false positives. The model could also be better at detecting all houses that are truly habitable so that recall can be higher.

Extreme Gradient Boosting Analysis Flowchart

The following is a flow diagram of the classification process using the Extreme Gradient Boosting method.

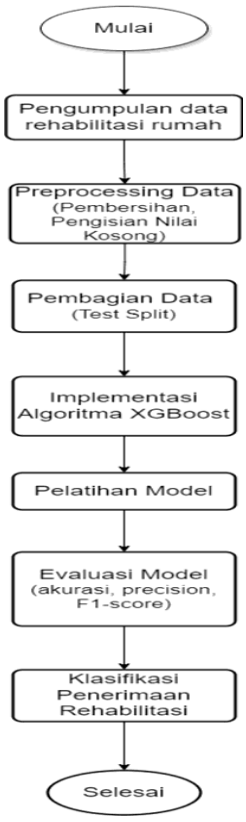

Figure 1. Flowchart of the Extreme Gradient Boosting Method

Based on *the flowchart* above, it can be explained that the research process begins with collecting data related to the rehabilitation program for habitable houses in Central Aceh, including demographic information and house conditions. After the data is collected, preprocessing is carried out such as data cleaning, filling in empty values, and normalization. The dataset is then divided into training and testing data to ensure fair model evaluation. The XGBoost algorithm is applied and the model is trained using the training data. Model performance is evaluated with the testing data using the metrics of accuracy, *precision* , *recall* , and *F1-Score* . The trained model is used to classify houses that are eligible for rehabilitation. This study ends by concluding the results of the classification and analysis.

4. Results and Discussion

Implementation begins with the process of collecting data from various sources including data on structural conditions, building materials, ventilation, lighting and sanitation. The data then goes through a preprocessing process to ensure completeness and cleanliness of the data, including handling missing values and normalization. The XGBoost model is then trained using the shared training data, with accuracy testing carried out using test data.

The given data consists of 150 samples with 9 features each:

- ID (unique identification)
- Name
- Address
- Condition_Structure
- Building Materials
- Ventilation
- Explanation
- Sanitation

These percentages help to understand the proportion of each category value in those features.

```python
import pandas as pd

# Membaca dataset
df = pd.read_csv('dataset_rumahlayakhuni1.csv', delimiter=';')

# Kolom yang akan dihitung persentasenya
columns = ['Kondisi_Struktur', 'Bahan_Bangunan', 'Ventilasi', 'Penerangan', 'Sanitasi']

# Melakukan perhitungan persentase untuk setiap kolom
for col in columns:
    percentage = df[col].value_counts(normalize=True) * 100
    print(f'Persentase untuk {col}:\n{percentage}\n')
```

Figure 2. Calculation of feature percentage

With the following percentage results:

```
Persentase untuk Kondisi_Struktur:
Kondisi_Struktur
BAIK     51.333333
RUSAK    48.666667
Name: proportion, dtype: float64

Persentase untuk Bahan_Bangunan:
Bahan_Bangunan
RUSAK    54.0
BAIK     46.0
Name: proportion, dtype: float64

Persentase untuk Ventilasi:
Ventilasi
RUSAK    53.333333
BAIK     46.666667
Name: proportion, dtype: float64

Persentase untuk Penerangan:
Penerangan
BAIK     62.0
RUSAK    38.0
Name: proportion, dtype: float64

Persentase untuk Sanitasi:
Sanitasi
RUSAK    52.0
BAIK     48.0
Name: proportion, dtype: float64
```

Figure 3. Calculation Results

Implementation of XGBoost Method

The implementation process of the XGBoost (*Extreme Gradient Boosting*) method is used to classify data related to the suitability of a house to live in. This implementation process involves several important steps, namely:

Importing Libraries

The following are the initial steps for including the libraries needed in the XGBOOST method.

- **pandas and numpy** : Used for data manipulation and operations. pandas for dataframe based data processing and numpy for numerical computation.
- ***train_test_split*** : Splits the dataset into training data and testing data.
- ***accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix*** : Evaluation metrics to measure the performance of a classification model.
- **xgboost** : Implementation of the XGBoost algorithm for classification.
- ***LabelEncoder :*** To convert categorical data into numbers.
- ***GridSearchCV*** : Used to find the best combination of hyperparameters through cross-validation.
- **matplotlib.pyplot and *seaborn*** : For graphic visualization such as *confusion matrix* and ROC curve.

Model Evaluation Results

These results show that the model trained using XGBoost has very high performance, with all evaluation metrics showing perfect scores:

```
Akurasi: 1.0
Presisi: 1.0
Recall: 1.0
F1-score: 1.0
AUC-ROC: 1.0
Hiperparameter terbaik: {'colsample_bytree': 0.5, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50, 'subsample': 0.5}
```

Figure 5. XGBoost model evaluation results

The best hyperparameters found by GridSearchCV are:

- *'colsample_bytree'* : 0.5: Only 50% of the features are selected to create each decision tree, which helps prevent *overfitting* .
- ' *gamma* ': 0: No additional regularization is applied to prune the tree on internal nodes.
- *'learning_rate'* : 0.1: A relatively low learning rate, helps the model learn gradually.
- *'max_depth'* : 3: The maximum tree depth is 3, which keeps the model simple.
- *'n_estimators'* : 50: The model was built using 50 decision trees.
- ' *subsample* ': 0.5: Only 50% of the training data is used for each tree, helping to reduce the chance of *overfitting* .

ROC (Receiver Operating Characteristic) Curve Visualization Results

AUC is a metric that shows the performance of a model in classifying positive and negative samples.

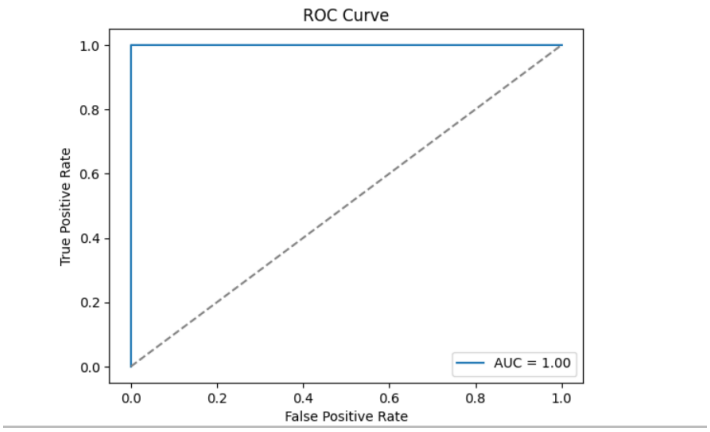

Figure 6. ROC curve

This graph shows ***the Area Under The Curve (AUC)*** for a binary classification model. This graph shows a **very good AUC value (1.00)** . This means that the model can perfectly distinguish between positive and negative samples.

ROC *Curve* with **AUC = 1.00** confirms that the model is very good in separating between the two classes (Livable and Unlivable) at all probability *thresholds* .
Here are some key points from the chart:

- The blue line is the ROC curve of the model.
- The dashed gray line is a diagonal line indicating random classification.
- The further the blue curve is from the diagonal line, the better the model performance.

Confusion Matrix Visualization Results

The results of the Confusion Matrix graph are the results of manual calculations of raw data for model evaluation against the existing raw dataset.

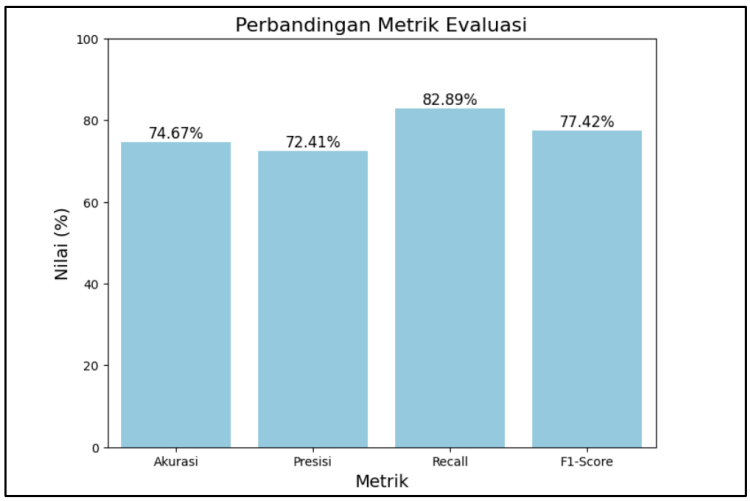

Figure 7. Confusion Matrix Results Graph of Manual Calculation

From the figure, it can be seen that the model has good performance with an accuracy value of 74.67%, precision of 72.41%, recall of 82.89%, and F1-Score of 77.42%.

Here is the interpretation of each metric:

- **Accuracy:** 74.67% means the model was able to correctly predict 74.67% of all cases.
- **Precision:** 72.41% means that of all cases that the model predicted as "admitted to rehab", 72.41% of them were actually admitted to rehab.
- ***Recall* :** 82.89% means the model was able to identify 82.89% of all cases that were actually admitted to rehab.
- ***F1-Score* :** 77.42% is the average value of precision and recall, which gives an overview of the model performance.

From the results, this classification model is quite good in predicting the acceptance of habitable housing rehabilitation. This model has a good ability to identify cases that are actually accepted for rehabilitation (high recall), and is relatively accurate in its predictions (relatively high accuracy and precision).

Classification of Acceptance of Rehabilitation of Habitable Homes

The following is an implementation of the Xgboost method in classification to determine the number of houses that are acceptable and unacceptable for rehabilitation.

The data is divided into input (X) and target (y) features. The target is the variable Habitable, which indicates whether a house is fit for rehabilitation or not (Yes or No). The XGBoost model is trained with optimized parameters to speed up the training time, such as reducing the number of estimators to 50 and setting the maximum tree depth to 3. The model is trained to predict which classes of houses are fit (1) and not fit (0) for rehabilitation.

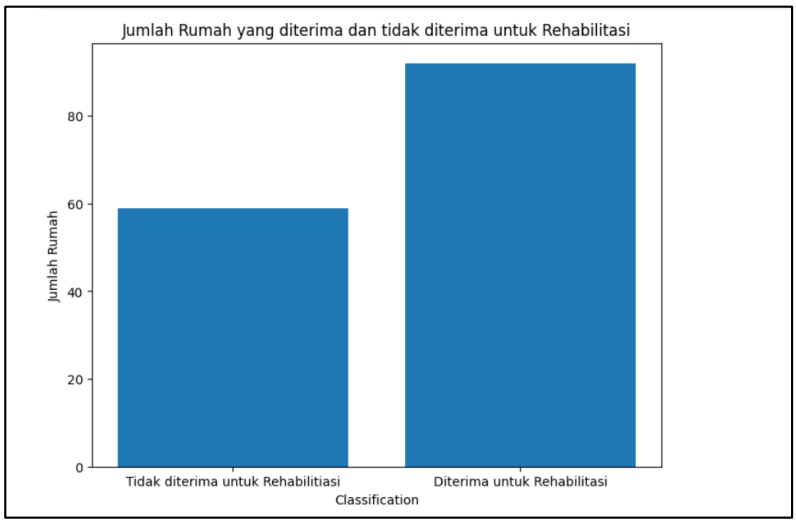

Figure 8. Results of the Classification Graph for Acceptance of Rehabilitation of Habitable Houses

From the results of the classification using the XGBoost method on this dataset, the following is the number of houses accepted and not accepted for rehabilitation:

Number of houses accepted for rehabilitation: 92

Number of houses not accepted for rehabilitation: 58

This shows that based on the existing features, 92 houses were classified as suitable for rehabilitation, while 58 houses were classified as unsuitable for rehabilitation.

5. Conclusions

From the results of the analysis of the implementation of the XGBoost method for the classification of habitable housing rehabilitation acceptance in Central Aceh Regency, it can be concluded that this model is able to produce very high accuracy, especially after hyperparameter tuning with GridSearchCV, which improves performance based on evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. Features such as structural conditions, building materials, ventilation, lighting, and sanitation have been shown to have a significant effect on the classification of housing eligibility. Although there are slight differences between the manual results and the model results, these differences do not affect the validity of the conclusions. Based on the classification results, there were 92 applications accepted and 58 rejected, mostly due to incomplete data, ineligibility, or other administrative constraints.

References

[1] F. Atmajayanti, A. Qashlim, and B. Burhanuddin, "Decision Support System for Accepting Livable Housing Assistance Using the Ahp Saw Method," *J. Peqguruang Conf. Ser.*, vol. 3, no. 1, p. 115, 2021, doi: 10.35329/jp.v3i1.1117.

[2] R. N. Alifah *et al.*, "Comparison of Tree Based Classification Methods for Body Mass Index Data Classification Problems," *Indones. J. Math. Nat. Sci.*, vol. 47, no. 1, p. 2024, 2024. [Online]. Available: https://journal.unnes.ac.id/journals/JM/index

[3] R. Oktafiani, A. Hermawan, and D. Avianto, "Max Depth Impact on Heart Disease Classification: Decision Tree and Random Forest," *J. RESTI (Syst. Eng. Inf. Technol.)*, vol. 8, no. 1, pp. 160–168, 2024, doi: 10.29207/resti.v8i1.5574.

[4] S. E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, "Application of Extreme Gradient Boosting (XGBOOST) Method on Credit Card Customer Classification," *J. Math. Theory Appl.*, vol. 4, no. 1, pp. 21–26, 2022, doi: 10.31605/jomta.v4i1.1792.

[5] G. A. Shafila, "Implementation of the Extreme Gradient Boosting (XGBOOST) Method for Classification of Bioinformatics Data (Case Study: Ebola Disease, GSE 122692)," *Dspace.Uii.Ac.Id*, pp. 1–77, 2020.

[6] A. Fadhilah, R. Ramdani, and M. F. Rizki, "Social Rehabilitation of Homeless and Beggars at the Harapan Jaya Bina Karya Social Home," *Community Dev. J.*, vol. 5, no. 3, pp. 5115–5119, 2024.

[7] D. Di, K. Cibadak, S. A. Nurulita, and D. Kurnia, "Journal of Praxisidealis," vol. 01, no. 01, 2024.

[8] T. Y. Tursilarini and T. Udiati, "The Impact of Uninhabitable Housing Assistance (RTLH) on the Social Welfare of Beneficiary Families in Bangka Regency," *Media Inf. Penelit. Kesejaht. Sos.*, vol. 44, no. 1, pp. 1–21, 2020. [Online]. Available: https://ejournal.kemsos.go.id/index.php/mediainformasi/article/view/1973/pdf

[9] S. Rahayu and Y. Yamasari, "Stroke Disease Classification with Support Vector Machine (SVM) Method," *J. Informatics Comput. Sci.*, vol. 5, no. 03, pp. 440–446, 2024, doi: 10.26740/jinacs.v5n03.p440-446.

[10] N. B. Putri and A. W. Wijayanto, "Comparative Analysis of Data Mining Classification Algorithms in Phishing Website Classification," *Komputika J. Sist. Comput.*, vol. 11, no. 1, pp. 59–66, 2022, doi: 10.34010/komputika.v11i1.4350.

[11] S. Ratna, "Digital Image Processing and Histogram with Python and PyCharm Text Editor," *Technol. J. Ilm.*, vol. 11, no. 3, p. 181, 2020, doi: 10.31602/tji.v11i3.3294.

[12] J. R. Fauzi, "Algorithms and Flowcharts in Solving a Problem Compiled by Janabadra University Yogyakarta 2020," *J. Tech. Inform.*, no. 20330044, pp. 4–6, 2020.

[13] ADAN Programming, "Pseudocode," *Definitions*, 2020, doi: 10.32388/tf77dy.

[14] M. N. Syarif, N. Pambudiyatno, W. Utomo, J. I. Jemur Andayani No, and K. Siwalankerto, "Design of Attendance System and Recapitulation of OJT Activity Journal Using Web-Based Visual Studio Code at Airnav, Matsc Branch," *Pros. Seminar. Inov. Technol. Aviation Year*, p. 2023, 2023.

[15] F. R. Valerian *et al.*, "Classification of obesity levels using the GBM method and confusion matrix," vol. 9, no. 2, pp. 2242–2249, 2025.