Hyperparameter Tuning for Optimizing Stunting Classification with KNN, SVM, and Naïve Bayes Algorithms

Wawan Firgiawan[1,*], Dita Yustianisa[2], Nurrahmi Afiah Nur[3], Gabrelia[4]

[1,2,3,4] Fakultas Teknik, Program Studi Teknik Informatika, Universitas Sulawesi Barat, Majene, Indonesia
Email: [1,*] wawanfirgiawan@unsulbar.ac.id, [2]yustianisadita@gmail.com, , [3]nurrahmi.afiahnur@gmail.com, [4]gabreliageby49@gmail.com

Abstrak– Tujuan dari penelitian ini adalah untuk mengevaluasi dan membandingkan kinerja tiga algoritma klasifikasi, yaitu *Naive Bayes* (NB), *Support Vector Machine* (SVM), *dan K-Nearest Neighbors* (KNN) dalam mengklasifikasikan data stunting. Dengan menggunakan ukuran evaluasi seperti akurasi, presisi, *recall,* dan *F1 score*, kinerja masing-masing algoritma dinilai sebelum dan sesudah penyesuaian hyperparameter. Hasil eksperimen menunjukkan bahwa SVM memberikan keseimbangan yang kuat antara presisi dan *recall* sebelum penyesuaian hyperparameter, KNN unggul dalam *recall,* dan NB mencapai presisi tertinggi. Setelah penyesuaian hyperparameter, semua model menunjukkan peningkatan kinerja, dengan SVM mencapai akurasi dan *F1 score* terbaik. Sementara NB tetap sangat presisi, mengurangi kesalahan positif, KNN terus menang dalam *recall*. Hasil menunjukkan bahwa penyesuaian hyperparameter sangat penting untuk mengoptimalkan kinerja algoritma, dan bahwa algoritma harus dipilih sesuai dengan tujuan spesifik penelitian untuk memaksimalkan akurasi deteksi dan mencapai keseimbangan antara recall dan presisi.

Kata Kunci: Stunting*, Hyperparameter Tuning, K-Nearest Neighbors, Support Vector Machine, Naïve Bayes.*

Abstract– The purpose of this study is to illuminate and compare the performance of three classifiers, namely Naive Bayes (NB), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), in classifying stunting data. Using evaluation measures such as accuracy, precision, recall, and F1 score, the performance of each algorithm is measured before and after hyperparameter adjustment. The experimental results show that SVM provides a strong balance between precision and recall before hyperparameter adjustment, KNN excels in recall, and NB achieves the highest precision. After hyperparameter adjustment, all models show improved performance, with SVM achieving the best accuracy and F1 score. While NB remains highly precise and reduces false positives, KNN continues to win the recall. The results show that hyperparameter adjustment is critical to optimizing algorithm performance and that algorithms should be selected according to specific research objectives to maximize detection accuracy and balance recall and precision.

Keywords: Stunting, Hyperparameter Tuning, K-Nearest Neighbors, Support Vector Machine, Naïve Bayes.

1. INTRODUCTION

Stunting remains a significant nutritional problem for infants and children under two in Indonesia. According to information from the Ministry of Communication and Informatics (Source: www.kominfo.go.id), the prevalence of stunting reached 37.2% in 2020, increasing from 35.6% in 2019 to 24.4% in 2021. According to the Indonesian Ministry of Health, the prevalence of growth retardation even reached 38.9% in 2020 [1]. This condition needs to be addressed immediately so as not to hinder the momentum of Indonesia's golden generation in 2045. Stunting affects not only children's physical growth but also their cognitive development, which impacts future productivity and quality of life. Therefore, the government needs to strengthen nutritional intervention efforts, increase access to health services, and provide intensive dietary education to the community to reduce stunting rates and ensure optimal development of Indonesian children. Stunting is when a baby has a length and height shorter than the standards set by the World Health Organization (WHO), which is more than minus two standard deviations [2].

Anthropometric measurements are used to assess the stunting status in early children. Anthropometric measurements encompass age (A), weight (BW), and height (H). Infants who undergo stunting exhibit reduced length and height compared to infants of equivalent age [3]. Stunting arises from persistent starvation, which impacts the development of the brain, intelligence, and body metabolism. This condition, known as fetal and neonatal malnutrition, occurs during the prenatal period and the initial days after birth [4]. Over time, stunting can have a significant impact on multiple facets of a child's health and development. Children who suffer from stunting typically have diminished cognitive capacities and reduced learning capacity compared to their peers of the same age who have normal growth. This disorder can also compromise their immune system, rendering them more vulnerable to infection and disease. Furthermore, stunting elevates the likelihood of having diverse chronic ailments in the future, such as diabetes, obesity, cardiovascular disease, cancer, and stroke. Individuals who endure stunting in childhood have a greater likelihood of developing disabilities in old age [5]. Thus, it is crucial to address stunting at an early stage in order to mitigate its long-term adverse effects and enhance the overall well-being of young children.

Data mining is the process of utilizing pattern recognition technology, together with statistical and mathematical methodologies, to identify new and significant similarities, patterns, and trends. This is achieved by categorizing vast quantities of data that are stored in repositories [6]. Classification is the process of assigning data to specified classes and labels. For example, it can be used to determine if a kid is stunted or not [7].

This work employ many algorithms, including K-Nearest Neighbors, Support Vector Machine, and Naïve Bayes, to construct a stunting classification model. K-Nearest Neighbors (KNN) is well-suited for classifying complicated data due to its flexibility and ability to adjust to changes in the data. The Support Vector Machine (SVM) algorithm is highly effective in dealing with datasets that have a large number of attributes and is particularly adept at handling overlapping classes [8]. Naïve Bayes is a suitable option for cases where the premise of feature independence is not completely applicable, as it effectively handles interdependent data features [9]. This algorithm was chosen after comparing its performance with other classifiers from several previously conducted studies [10] [11] [12], taking into account factors such as accuracy, computational efficiency, and suitability to the specific characteristics of the stunting data set.

Each algorithm will undergo hyperparameter tuning approaches in order to enhance the performance of the model. Hyperparameter tuning involves the adjustment of model parameters in order to maximize its performance [13]. By employing this methodology, it is anticipated that the model will yield more precise outcomes in detecting young children who are at a higher likelihood of experiencing stunted growth. Hyperparameter tuning is essential for optimizing the model to improve the accuracy and reliability of predictions, particularly in the context of child health and stunting prevention.

This study will gather data from several sources to guarantee that the generated model has an ample amount of data for training and testing purposes. The data will encompass anthropometric data, nutritional history, and environmental and social determinants that impact the nutritional status of children. Data preparation will be conducted to cleanse and format the data in order to meet the requirements of the algorithm. Subsequently, the data will be partitioned into training and test sets in order to assess the model's efficacy.

Utilizing different algorithms and optimizing hyperparameters are anticipated to offer a more thorough understanding of the elements that contribute to stunting and aid in the development of more efficient intervention techniques. This study seeks to create a precise predictive model and offer comprehensive insights into the reasons and potential remedies for the issue of stunting in Indonesia. By leveraging the advantages of each method and fine-tuning hyperparameters, the findings of this research can offer more effective recommendations for policymakers in developing precise and data-driven actions.

2. RELATED RESEARCH

A study conducted by Gibran Nasrizal Masacgi and Muhammad Syaifur Rohman in 2023 investigated the optimization of classification algorithm models for toddler stunting using the bagging method. The study found that by combining three algorithms: K-Nearest Neighbors, Support Vector Machine, and Naïve Bayes - a precision of 89.77%, a recall of 95.57%, and the highest F1-score of 90.27% were achieved. This demonstrates that employing bagging as an ensemble technique can greatly enhance the precision of forecasting infant stunting. This study is deemed creative due to its integration of three algorithms and implementation of a novel optimization strategy, which has the potential to effectively prevent stunting in infants by improving the accuracy of identification [14].

In 2023, Septi Kenia Pita Loka and Arif Marsal conducted a study that compared the K-Nearest Neighbor and Naïve Bayes Classifier algorithms in order to classify the nutritional health of toddlers. The K-nearest neighbors (KNN) approach achieved an accuracy rate of 96.10%, whereas the Naive Bayes classifier (NBC) technique achieved an accuracy rate of 90.94% when utilizing the toddler status anthropometric standard with body mass to body height ratio (BB/TB). These findings indicate that the KNN algorithm outperforms the NBC algorithm [6] in classifying toddler mass-weighing data.

In 2023, Dewi Nurmalasari, Teguh Imam Hermanrto, and Imam Ma'ruf Nugroho conducted a study that examined the SVM, KNN, and NBC algorithms for sentiment analysis of loan service applications. The study revealed promising findings. Analyzed data from loan service applications, including Kredivo, Akulaku, and Indodana evaluations. The Kredivo program obtained an accuracy of 84% using the K-Nearest Neighbors (KNN) algorithm, 88% using the Naïve Bayes algorithm, and 89% using the Support Vector Machine (SVM) algorithm. The accuracy rates for the Akulaku application were 79% for K-Nearest Neighbors (KNN), 86% for Naïve Bayes, and 87% for Support Vector Machines (SVM). The accuracy rates for the Indodana application were 81% for KNN, 88% for Naïve Bayes, and 88% for SVM. The results suggest that the Support Vector Machine method outperforms the K-Nearest Neighbor and Naïve Bayes methods in accurately analyzing the sentiment of loan service applications [15].

3. METHOD

3.1 Research Stages

The research stages necessary to finish the study are outlined based on the research stages depicted in Figure 1:

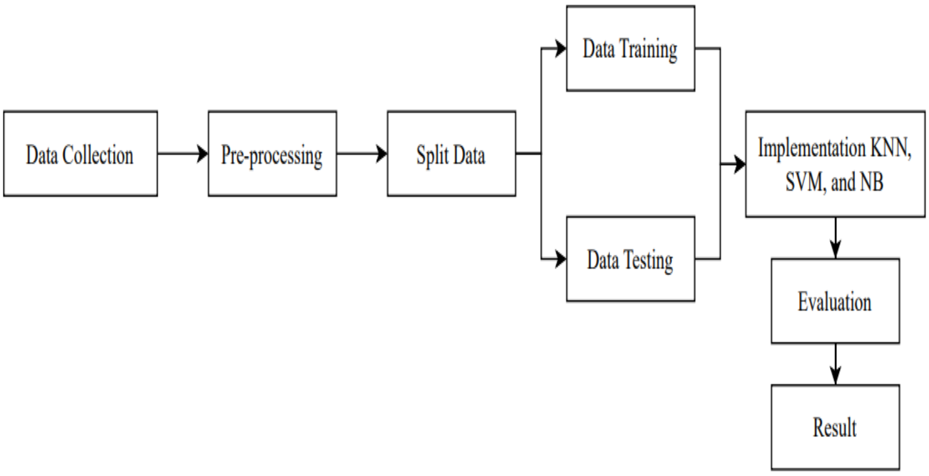

Figure 1. Research Stages

a. Data Collection
Data collection generally consists of data acquisition, labeling, and improving existing data or models. Integrating machine learning and data management for data collection is an essential part of the process because it relates to the results obtained [16]. The data used in this study came from Kaggle.com. The dataset has eight parameters, including gender (male and female), age, birth weight, birth length, current weight, current length, breastfeeding status, and stunting status. The label "Stunting" in this dataset indicates whether a child is stunted (Yes) or not (No).

b. *Pre-Processing*
Data pre-processing is an exciting part of machine learning. Raw data from the field must be processed using special techniques to improve its quality before being applied to the prediction method [17]. In this study, a per-processing procedure was run to prepare the data for analysis and modeling. An essential stage in pre-processing involves converting categorical attributes into numerical representations.

c. Data Splitting
Penelitian tentang pemisahan data untuk mengevaluasi kinerja model machine learning merupakan hal yang menarik, terdapat beberapa penelitian yang mengungkapkan pengaruh rasio data pelatihan dan pengujian yang berbeda pada kinerja model dapat memberikan hasil yang berbeda [18]. dengan mempertimbangkan rasio pemisahan data. Data partitioning, also known as data division or data splitting, is a crucial research procedure that involves dividing the dataset into two primary components: the training data and the testing data. This stage guarantees that the constructed model may be evaluated using previously unseen data, hence offering a more precise assessment of its performance in real-life scenarios.

d. Algorithm Implementation
At this stage, the KNN, SVM, and NB algorithms are used to implement the classification model. The Grid Search and Random Search approaches apply each algorithm with the most suitable hyperparameter modifications.

e. Evaluation
Various studies use techniques such as accuracy, recall, precision, and F1-score, K-Fold to measure algorithm performance due to their ability to assess the algorithm's success [19] [20]. During this stage, an assessment is conducted to determine the values of accuracy, recall, precision, and F1-score. The evaluation is conducted using a 10-fold cross-validation technique to assure the dependability and broad applicability of the model.

f. Results
The model that yielded the highest F1-score value was acquired after conducting tests. Subsequently, this model is employed to detect toddlers who are susceptible to experiencing stunting.

3.2 Classification Model

A multitude of machine learning algorithms have been created and employed to tackle the difficulties associated with data classification and predictive analysis. Regarding the classification of toddler stunting data, the three primary algorithms to be examined are Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest Neighbor (KNN). Each of these methods possesses distinct benefits and drawbacks, as well as applications that are appropriate for different datasets and analysis requirements. This text provides a comprehensive analysis and investigation of the three algorithms.

3.2.1 K-Nearest Neighbor (KNN)

The K-Nearest Neighbor (KNN) algorithm is a non-parametric technique that is based on examples and is widely regarded as the simplest approach in the data mining process. KNN, or k-nearest neighbors, is a data classification technique that is particularly suitable for small datasets due to its simplicity and ease of implementation. Nevertheless, KNN exhibits certain shortcomings, particularly in terms of inefficiency when applied to extensive and intricate datasets [21].

The KNN approach categorizes new data by determining its proximity to existing data points. In order to classify new data, we calculate the distance between the new data and its K nearest neighbors. We then forecast the class label of the new instance based on the class label of the nearest neighbor. The choice of the number of K nearest neighbors has a significant impact on the accuracy of the K-nearest neighbors (KNN) algorithm. When the K number is too little, the model becomes more vulnerable to noise. Conversely, if the K value is too large, it can be less responsive to changes in the data [22].

KNN is a memory-based algorithm that iteratively searches through the data to find the nearest data attribute or parameter. The test data is evaluated against the training data using the metric of minimum distance. The KNN algorithm commonly employs the Euclidean distance formula, which is expressed as:

$$\boldsymbol{D_i} = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{x_i} - \boldsymbol{y_i})^{\boldsymbol{2}}} \tag{1}$$

Description:

- *d : Distance*
- *x : Training Data*
- *y : Test Data*
- *n : Amount of data*

Some important hyperparameters tuned in KNN include:

a. **Number of neighbors (k):** Determines the nearest neighbors to be used for classification.
b. **Distance measurement method:** Euclidean and Manhattan are commonly used to calculate the distance between data points.
c. **Weight:** Determines whether all neighbors have the same contribution in classification (uniform) or contribution based on their distance (distance).

3.2.2 Support Vector Machine (SVM)

The Support Vector Machine (SVM) technique seeks to identify the optimal hyperplane that may effectively distinguish two classes. A hyperplane is a mathematical function that divides data into two distinct groups by maximizing the margin or the distance between the closest training samples from each class and the decision boundary. This method possesses numerous benefits, rendering it highly efficient in diverse applications [23]. Firstly, Support Vector Machines (SVM) exhibit high performance with data sets of varying sizes, ranging from tiny to large. This attribute renders it a versatile option for a wide range of data sets. Furthermore, Support Vector Machines (SVM) exhibit efficacy when dealing with datasets including several attributes, as they are capable of effectively managing high-dimensional data without compromising performance. Furthermore, this technique is reasonably straightforward to develop and can be utilized in a wide range of practical applications [11]. Originally, Support Vector Machines (SVM) were limited to doing binary classification tasks, where input is divided into two distinct classes. However, SVM has undergone significant advancements that encompass regression, outlier detection, and classification. Methods such as multi-class SVM and Support Vector Regression (SVR) enable the application of SVM in more intricate scenarios, encompassing multi-class classification and the prediction of continuous values. Furthermore, Support Vector Machines (SVM) possess the benefit of effectively addressing the issue of overfitting, particularly in datasets that contain noise or irregular data. SVM can utilize a suitable kernel to transform data into a higher dimensional space, enabling the separation of more intricate data [24]. In summary, Support Vector Machines (SVM) are a valuable tool in the field of machine learning, providing robust performance and adaptability in a wide range of classification and regression tasks. Its efficacy in managing datasets with numerous features and its capacity to mitigate overfitting render it one of the most prevalent and extensively employed algorithms in data analysis and machine learning. This is the testing phase of the Support Vector Machine (SVM) approach used to determine the value of (x) in equation (2) [25].

$$\boldsymbol{f}\,(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{\alpha_i y_i K}(\,\boldsymbol{x_i}, \boldsymbol{x}^{+}) + \boldsymbol{b} \tag{2}$$

To find the value of b, can use equation 3.

$$\boldsymbol{b} = -\frac{1}{2}\left(\sum_{i=1}^{n} \boldsymbol{\alpha_i y_i K}(\,\boldsymbol{x_i}, \boldsymbol{x}^{+}) + \boldsymbol{\alpha_i y_i K}(\boldsymbol{x_i}, \boldsymbol{x}^{-})\right) \tag{3}$$

Some of the important hyperparameters tuned in SVM include:

a. **Kernel:** Kernel functions such as linear, radial basis function (RBF), and polynomial map the data to a higher dimensional space.
b. **Regularization parameter (C):** Determines the trade-off between maximizing margin and minimizing classification error.
c. **Kernel parameter (gamma):** This parameter determines how far the influence of a single data point extends in the feature space. A small value means the 'influence' of a single data point is far-reaching, while a large value means the influence is close.

3.2.3 Naïve Bayes (NB)

Naïve Bayes is a classification algorithm that employs probability and statistical techniques, initially introduced by British scientist Thomas Bayes. This program utilizes previous data to make predictions about future opportunities. An important benefit of this algorithm is its capacity to utilize a little amount of training data in order to ascertain the necessary parameters for the classification procedure [26]. The subsequent sequence illustrates the progression of the NB technique [27]:

a. **Reading data**
b. **Calculating the number and probability:** If the data is numerical, do the following steps:
   - Find the mean and standard deviation: Calculate the mean and standard deviation for each parameter, which is numerical data.
   - Find the probabilistic value: Calculate the probability by dividing the number of appropriate data in the same category by the total number of data in that category.
c. **Get the value in the table:** Generate a table containing the mean, standard deviation, and probability values used in the classification process.

The Bayes theorem formula in equation (4) is [28]:

$$\boldsymbol{P}(\boldsymbol{H}|\boldsymbol{X}) = \frac{\boldsymbol{P}(\boldsymbol{X}|\boldsymbol{Ci}).\boldsymbol{P}(\boldsymbol{Ci})}{\boldsymbol{P}(\boldsymbol{X})} \tag{4}$$

Description:

- *X* : *Data with unknown class*
- *Ci* : *Hypothesis on a specific class*
- *P(Ci|X)* : *Probability of hypothesis Ci based on condition X*
- *P(Ci)* : *Probability of hypothesis Ci*
- *P(X|Ci)* : *Probability of X based on condition on hypothesis Ci*
- *P(X)* : *Probability of X*

Some commonly used NB variants include:

a. **Kernel:** Kernel functions such as linear, radial essential function (RBF), and polynomial map data to a higher dimensional space.
b. **Regularization parameter (C):** Determines the trade-off between maximizing margin and minimizing classification error.
c. **Kernel parameter (gamma):** This parameter determines how far the influence of a single data point extends in the feature space. A small value means the influence reaches far, while a large value means the influence is close.

These algorithms were implemented using Python 3.9 with scikit-learn version 1.2.0. The choice of software and version ensures compatibility with modern data science libraries and leverages the latest advances in machine learning techniques. Data preprocessing was performed using pandas 1.5.0 and NumPy 1.21.0, while visualizations were generated using Matplotlib 3.5.1. The experiments were conducted on a machine equipped with an AMD A10 Radeon R5 Graphics, 8GB of RAM, and Windows 10. These algorithms were selected after comparing their performance against other classifiers, considering factors such as accuracy, computational efficiency, and suitability for the specific characteristics of the stunting dataset. The specific tools and software versions were chosen to maximize reproducibility and ensure the analysis was performed with up-to-date and reliable computational resources.

3.3 Hyperparameter Tuning

Hyperparameter tuning is conducted to identify the most optimal settings for each method. The study employs two techniques: grid search A method that exhaustively checks all possible combinations of explicitly specified hyperparameters. While this procedure is thorough, it can be somewhat time-consuming and need a significant amount of resources. Random search is a more efficient approach that involves testing random combinations of hyperparameters selected from a preset distribution. This approach frequently yields results that are near ideal, while requiring less time and resources compared to a grid search. The objective of this tuning process is to enhance the accuracy, precision, recall, and F1-score of the model, guaranteeing that the resulting model is the most optimal in detecting children who are at risk of stunting.

3.4 Model Evaluation

The model underwent evaluation using diverse performance indicators to guarantee its dependability and efficacy in categorization. The metrics utilized comprise accuracy, denoting the ratio of correct predictions to the total number of predictions; precision, representing the ratio of correct positive predictions to the total number of positive predictions; recall, indicating the ratio of correctly predicted positive cases to the total number of actual positive cases; and F1-score, denoting the harmonic mean of precision and recall, which offers a balanced assessment of both metrics.

The model is assessed using a Confusion matrix, which is an essential evaluation tool in data categorization analysis. The confusion matrix is visually represented as a 2 x 2 matrix that categorizes the model's prediction outputs into four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [29]. True Positives refer to the count of data accurately classified as positive in a classification task. On the other hand, True Negatives refer to the count of accurately predicted negative data. False Positives are instances where data is inaccurately classified as positive, when it should instead be classified as negative. False Negatives refer to the instances where data is wrongly classified as negative when it should instead be classified as positive [12].

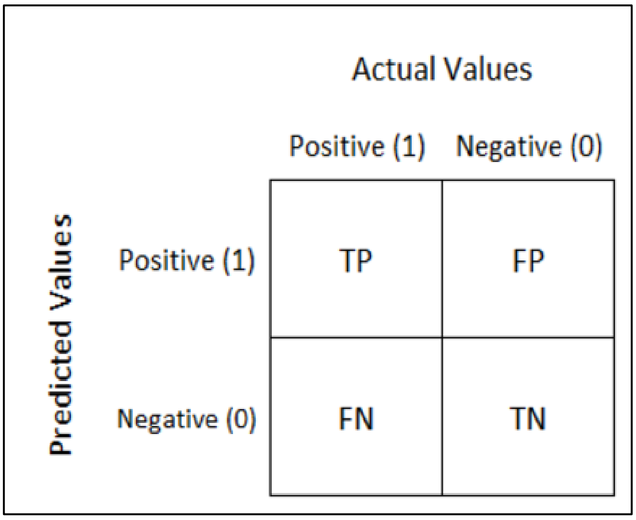

Figure 2. Confusion Matrix

Description:

- *TP (True Positive): Prediction 1, actually 1*
- *TN (True Negative): Prediction 0, actually 0*
- *FP (False Positive): Prediction 1, actually 0*
- *FN (False Negative): Prediction 0, actually 1*

A confusion matrix provides a visual representation of the ability of a classification model to accurately differentiate between positive and negative classes. Multiple evaluation metrics can be derived from this matrix, including accuracy, precision, recall, and F1-score. Every indicator offers a more intricate representation of the model's effectiveness in categorizing data [30].

The utilization of a confusion matrix not only yields insights into the overall correctness of the model, but also facilitates the identification of specific areas where the model tends to err. For instance, if a model exhibits a high number of False Negatives, it suggests that the model is less responsive to identifying true positive cases. By comprehending the organization and analysis of a confusion matrix, data analysts can make more informed decisions when assessing and enhancing classification models, and get a more profound comprehension of how the model performs on various test data [31]. The formula for computing accuracy, precision, recall, and F1-score based on the confusion matrix is as follows [32]:

a. **Accuracy** is the degree of closeness between predicted and actual values.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

b. **Precission** compares TP data and the number of data predicted to be positive.

$$precision = \frac{TP}{TP + FP} \quad (6)$$

c. **Recall** is the comparison between TP and the number of positive data.

$$recall = \frac{TP}{TP + FN} \quad (7)$$

d. **F1-score** is the harmonic mean between precision and recall values.

$$F1 - score = \frac{2 * Recall * precision}{Recall + precission} \quad (8)$$

3.5 Data Sources

The “Faktor Stunting” dataset is sourced from Kaggle.com and contains 10.000 data on factors that influence toddlers' stunting. This dataset is designed to assist in analyzing and predicting stunting conditions. This data can be accessed through the following link: https://www.kaggle.com/datasets/harnelia/faktor-stunting. The attributes of this dataset include:

a. **Gender:** (Male or Female). This feature is crucial for determining whether there is a disparity in the prevalence of stunting between male and female children.
b. **Age:** Starting from 6 to 48 months. Age is an essential factor in child growth, and this age range covers a critical period in toddler development.
c. **Birth Weight:** Starting from 2 to 3.1 kg. Birth weight is often used as an early indicator of infant health and can affect the risk of stunting.
d. **Birth Length:** Starting from 48 to 50 cm. Birth length is also an indicator of infant health related to growth.

The dataset exhibits a notable imbalance in data distribution, with 79,55% of the total data indicating instances of stunting, whereas only 20,45% do not exhibit stunting. The study must take into account this disparity since it has the potential to impact the performance of the prediction model. To prevent bias towards the majority class, it may be necessary to employ distinct methods, such as suitable evaluation criteria or data balancing procedures. Table 1 below is a table displaying the distribution of stunting data in the dataset.

Table 1. Data Distribution by Class

Stunting	Amount	Percentage (%)
Yes	7.955	79,55
No	2.045	20,45
Total	10.000	100,00

This table displays the distribution of stunting data in a dataset of 10,000 data elements. Out of the dataset, a total of 7.955 toddlers were categorized as stunted, accounting for 79,55% of the entire dataset. Simultaneously, a total of 2.045 young children were categorized as not experiencing stunted growth, which accounts for 20,45% of the entire dataset. The distribution of this dataset indicates that the majority of the data consists of stunting patients. This suggests that stunting is a notable concern in the population represented by this dataset. The uneven distribution of data must also be taken into account during the modelling and analysis process, since it might lead to bias in the prediction model towards the dominant class (stunting). Hence, it is imperative to employ data balancing approaches or modify model evaluation criteria to ensure the model's ability to recognize cases of stunting while also considering minority data (non-stunting).

4. RESULTS AND DISCUSSIONS

This study involved conducting trials and tests to assess the efficacy of the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB) algorithms in identifying stunting data. Testing was conducted utilizing the Hyperparameter tuning technique to achieve optimal outcomes from each algorithm by employing the appropriate parameters. The table displays the evaluation results, encompassing accuracy, precision, recall, and F1-score, both before and after hyperparameter adjustment.

4.1 Without Hyperparameter Tuning

Initially, a test was performed on three classification methods, including KNN, SVM, and NB, to categorize stunting data without any hyperparameter tuning. Hyperparameter tuning involves the adjustment of parameters in order to optimize the performance of a model. Nevertheless, throughout this phase, testing was conducted using the default settings in order to assess the fundamental performance of each method. The outcomes are visible in table 2.

Table 2. Accuracy Test Results Without Hyperparameter Tuning

Method	Accuracy (%)
KNN	82,15
SVM	82,80
Naïve Bayes	80,35

The accuracy test results indicate that KNN earned an accuracy rate of 82,15%, whilst SVM attained the best accuracy rate of 82,8%. Despite the absence of any tuning, Support Vector Machines (SVM) demonstrated a remarkable capacity to accurately distinguish between different classes. However, Naïve Bayes achieved an accuracy of 80,35%, which is slightly lower but still satisfactory given its reliance on the assumption of independence across features. The outcomes are visible in table 3.

Table 3. Precision Test Results Without Hyperparameter Tuning

Method	Precision (%)
KNN	86,18
SVM	87,23
Naïve Bayes	88,26

Table 3 displays the precision values of three machine learning techniques, namely KNN, SVM, and NB, without the utilization of hyperparameter adjustment. The Naïve Bayes algorithm achieved the greatest precision score of 88,26, demonstrating its superior capability to reduce false positive mistakes. The Support Vector Machine (SVM) achieved a precision of 87,23%, while the K-Nearest Neighbors (KNN) algorithm achieved a precision of 86,18%. The results suggest that all three algorithms show promising promise in identifying stunted data, with Naïve Bayes performing the most effectively in this particular case. The outcomes are visible in table 4.

Table 4. Recall Test Results Without Hyperparameter Tuning

Method	Recall (%)
KNN	92,40
SVM	91,83
Naïve Bayes	86,88

Prior to implementing hyperparameter adjustment, Table 4 displays the recall values for three machine learning algorithms: KNN, SVM, and NB. The KNN algorithm demonstrates a recall value of 92,40, which signifies its exceptional capability to identify nearly all instances of stunting. The Support Vector Machine (SVM) method ranks second with a recall value of 91,83, demonstrating its strong ability to accurately identify affirmative cases. Naïve Bayes has a recall value of 86,88, which is marginally lower than the recall values of the other two algorithms. These findings suggest that, without additional adjustments, KNN and SVM outperform Naïve Bayes in identifying stunting instances. The outcomes are visible in table 5.

Table 5. F1-Score Test Results Without Hyperparameter Tuning

Method	F1-Score (%)
KNN	89,18
SVM	89,48
Naïve Bayes	87,56

Table 5 displays the F1-score values for the KNN, SVM, and NB algorithms prior to hyperparameter adjustment. The Support Vector Machine (SVM) method achieves an F1-score of 89,48, demonstrating a favorable equilibrium between precision and recall for categorizing stunting data. The KNN method performs marginally worse than SVM, with an F1-score of 89,18, suggesting nearly equivalent performance. Naïve Bayes achieved an F1-score of 87,56, suggesting that although it has good precision, its recall is inferior compared to KNN and SVM. The F1-score is crucial for evaluating the model's ability to accurately identify cases of stunting, taking into account both the accuracy of positive predictions and the completeness of detection.

Based on the test findings, four evaluation metrics were used: accuracy, precision, recall, and F1-score. Based on the results of these experiments, it is evident that the Support Vector Machine (SVM) consistently demonstrates superior performance compared to the other two algorithms, namely K-Nearest Neighbors (KNN) and Naïve Bayes (NB). The Support Vector Machine (SVM) exhibits the best accuracy and F1-score, indicating its strong capability in classifying stunting. The KNN algorithm yields competitive results, particularly in terms of recall, which demonstrates its usefulness in identifying cases of stunting. Although Naïve Bayes exhibits a high accuracy value, it demonstrates inferior performance in recall and F1-score. This suggests that the model is more likely to make accurate positive predictions, but it is less effective at identifying all instances of stunting.

4.2 Using Hyperparameter Tuning

This section assesses the effectiveness of the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB) algorithms at identifying stunting data following hyperparameter adjustment. Hyperparameter tuning is a method used to discover the most effective parameter configurations in order to enhance the performance of a model. The evaluation results after hyperparameter adjustment include numbers for accuracy, precision, recall, and F1-score. This data is presented in table 6.

Table 6. Accuracy Test Results Using Hyperparameter Tuning

Method	Accuracy (%)
KNN	85,50
SVM	86,20
Naïve Bayes	82,10

Following the implementation of hyperparameter tweaking, the Support Vector Machine (SVM) algorithm achieved the greatest accuracy of 86,20%, surpassing the accuracy of the K-Nearest Neighbors (KNN) algorithm, which achieved an accuracy of 85,50%. The accuracy of NB was measured at 82,10%, suggesting that even after tweaking, SVM remains the most efficient method for classifying stunted data. This data is presented in table 7.

Table 7. Precision Test Results Using Hyperparameter Tuning

Method	Precision (%)
KNN	88,75
SVM	89,50
Naïve Bayes	89,00

Following the tuning process, the Naive Bayes (NB) algorithm demonstrates exceptional precision, achieving a value of 89,00%. This figure indicates its superior capability to eliminate false positive errors, thereby enhancing its accuracy. The precision values for SVM and KNN are 89,50% and 88,75%, respectively. This data is presented in table 8.

Table 8. Recall Test Results Using Hyperparameter Tuning

Method	Recall (%)
KNN	93,20
SVM	92,90
Naïve Bayes	88,30

The K-nearest neighbors (KNN) algorithm had a recall rate of 93,20%, signifying its robust capability to identify nearly all instances of stunting. The Support Vector Machine (SVM) achieved a recall rate of 92,90%, placing it in second position, while the Naive Bayes (NB) classifier had a recall rate of 88,30%. This data is presented in table 9.

Table 9. F1-Score Test Results Using Hyperparameter Tuning

Method	F1-Score (%)
KNN	90,92
SVM	91,18
Naïve Bayes	88,65

Following the tuning process, the Support Vector Machine (SVM) once again demonstrated the greatest F1-score of 91,18%, indicating a favorable equilibrium between precision and recall. The K-nearest neighbors (KNN) algorithm achieved an F1-score of 90,92%, whereas the Naive Bayes (NB) algorithm obtained an F1-score of 88,65%.

Conducting hyperparameter tuning experiments demonstrated a substantial improvement in the performance of the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB) algorithms, as compared to the results obtained without tuning. The Support Vector Machine (SVM) attained an accuracy of 86,20% and a precision of 89,50%, demonstrating its capability to effectively distinguish between different classes of stunting data and reduce the occurrence of false positive errors. The KNN algorithm achieved a recall rate of 93,20%, demonstrating its remarkable efficacy in recognizing nearly all cases of stunting. By comparison, the Support Vector Machine (SVM) achieved the greatest F1-score of 91,18%, signifying an ideal equilibrium between precision and recall. This improvement can be attributed to the refinement of parameter modification through tuning. This refinement enables the model to more effectively adapt to the complexity of the data, resulting in enhanced accuracy and generalization ability. Additionally, it optimizes the balance between key performance measures.

4.3 Comparison Test Results

After testing without and without hyperparameter tuning, the results are compared, as shown in Figure 3 below.

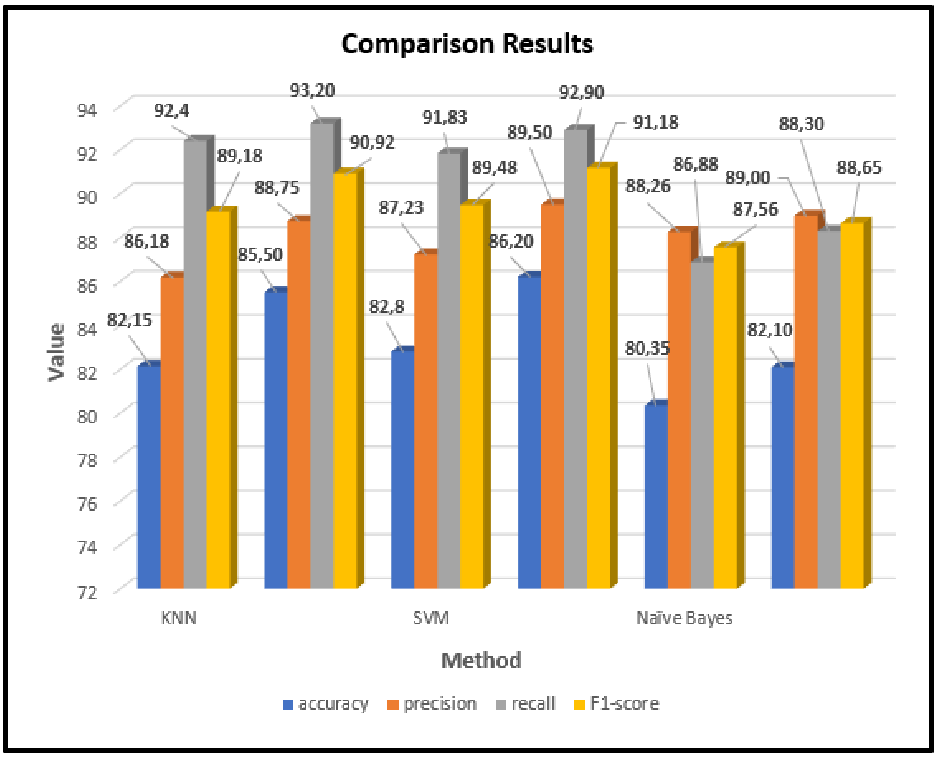

Figure 3. Comparison of Test Results Before and After *Hyperparameter Tuning*

According to Figure 3, the pre-tuning test results indicate that the K-Nearest Neighbor (KNN) algorithm achieved an accuracy of 82,15%, a precision of 86,18%, a recall of 92,40%, and an F1-score of 89,18%. The Support Vector Machine (SVM) algorithm achieved an accuracy rate of 82,80%, a precision rate of 87,23%, a recall rate of 91,83%, and an F1-score of 89,48%. The Naïve Bayes (NB) algorithm achieved an accuracy of 80,35%, a precision of 88,26%, a recall of 86,88%, and an F1-score of 87,56%. Following the implementation of hyperparameter adjustment, there was a notable enhancement observed across all algorithms. The accuracy of the K-Nearest Neighbor (KNN) algorithm improved to 85,50%, while the precision increased to 88,75%, the recall to 93,20%, and the F1-score to 90,92%. The Support Vector Machine (SVM) also experienced an increase, achieving an accuracy of 86,20%, precision of 89,50%, recall of 92,90%, and an F1-score of 91,18%. The Naïve Bayes (NB) model achieved an accuracy of 82,10%, precision of 89,00%, recall of 88,30%, and F1-score of 88,65%.

A ten-fold cross-validation analysis was conducted to enhance result accuracy and prevent overfitting. The subsequent data presents the mean outcomes of 10-fold cross-validation for each algorithm following the optimization of hyperparameters.

Table 10. K-Fold Cross validation results

Algorithm	Average Accuracy (%)	Average Precission (%)	Average Recall (%)	Average F1-score (%)
KNN	85,50	88,75	93,20	90,92
SVM	86,20	89,50	92,90	91,18
NB	82,10	89,00	88,30	88,65

The K-Fold Cross Validation test results indicate that the SVM algorithm had superior performance in terms of average accuracy and average F1-score, achieving values of 86,20% and 91,18% respectively. This demonstrates that Support Vector Machines (SVM) may achieve an optimal equilibrium between precision and recall while classifying stunting data. The KNN algorithm demonstrated exceptional performance, particularly in terms of average recall, with a remarkable 93,20% accuracy. This makes it highly successful in identifying cases of stunting. Despite Naïve Bayes having a lower average accuracy of 82,10%, it demonstrated strong performance with the greatest average precision value of 89,00%. This indicates its capability to generate precise positive predictions. In summary, these findings suggest that the selection of an algorithm for stunting classification should be tailored to individual requirements, whether the focus is on precise identification (recall) or a combination of precise identification and prediction (F1-score).

5. CONCLUSION

The experimental results indicate that each algorithm, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB), possesses distinct strengths in the context of stunting classification. Before hyperparameter tuning, SVM demonstrated a strong balance between precision and recall, KNN excelled in recall, and NB achieved the highest precision. After hyperparameter tuning, all models showed improved performance, with SVM achieving the highest average accuracy and F1 score, making it the most balanced and reliable choice. KNN continued to excel in the recall, effectively identifying stunting cases, while NB maintained vital precision, reducing false positives. In summary, our findings emphasize the significance of hyperparameter tuning in optimizing the performance of algorithms. It is crucial to select an algorithm that aligns with the specific objectives of the study, whether it is to maximize detection accuracy or strike a balance between precision and recall.

ACKNOWLEDGEMENT

We would like to extend our profound appreciation to the Informatics Engineering Study Program of West Sulawesi University for their invaluable support and provision of facilities during this research endeavor. This research could be effectively conducted with the assistance of resources provided by the study program. It is expected that the Informatics Engineering Study Program of Universitas Sulawesi Barat persistently progress and achieve success in generating future breakthroughs.

REFERENCES

[1] Kesehat. Republik Indones., pp. 77–77, 2023, [Online]. Available: https://promkes.kemkes.go.id/materi-hasil-survei-status-gizi-indonesia-ssgi-2022.

[2] H. Hatijar, "The Incidence of Stunting in Infants and Toddlers," J. Ilm. Kesehat. Sandi Husada, vol. 12, no. 1, pp. 224–229, 2023, doi: 10.35816/jiskh.v12i1.1019.

[3] P. Apriyani, A. R. Dikananda, and I. Ali, "Penerapan Algoritma K-Means dalam Klasterisasi Kasus Stunting Balita Desa Tegalwangi," Hello World J. Ilmu Komput., vol. 2, no. 1, pp. 20–33, 2023, doi: 10.56211/helloworld.v2i1.230.

[4] Vega Herliansyah, Roswan Latuconsina, and Ashri Dinimaharawati, "Prediksi Stunting Pada Balita Dengan Menggunakan Algoritma Klasifikasi *Naïve Bayes*," e-Proceeding Eng., vol. 8, no. 5, p. 6642, 2021.

[5] E. Prasetyo and K. Nugroho, "Optimasi Klasifikasi Data Stunting Melalui Ensemble Learning pada Label Multiclass dengan Imbalance Data," Techno.Com, vol. 23, no. 1, pp. 1–10, 2024, doi: 10.62411/tc.v23i1.9779.

[6] S. K. P. Loka and A. Marsal, "Perbandingan Algoritma K-Nearest Neighbor dan *Naïve Bayes* Classifier untuk Klasifikasi Status Gizi Pada Balita," MALCOM Indones. J. Mach. Learn. Comput. Sci., vol. 3, no. 1, pp. 8–14, 2023, doi: 10.57152/malcom.v3i1.474.

[7] A. Z. M S. Widodo, A. P. Kusuma, and W. D. Puspitasari, "Analisis Algoritma *Naïve Bayes* Classifier (Nbc)," JATI (J. Mahasiswa Teknik Informatika), vol. 7, no. 1, pp. 87–94, 2023.

[8] A. Y. Labolo, S. Mooduto, A. Bode, and I. C. R. Drajana, "Penerapan Algoritma Spport Vector Machine dan K-Nearest Neighbor Menggunkan Feature Selection Backward Elimination Untuk Prediksi Status Penderita Stunting Pada Balita," J. Tecnoscienza, vol. 6, no. 2, pp. 374–388, 2022, doi: 10.51158/tecnoscienza.v6i2.713.

[9] M. Y. Titimeidara and W. Hadikurniawati, "Implementasi Metode *Naïve Bayes* Classifier Untuk Klasifikasi Status Gizi Stunting Pada Balita," J. Ilm. Inform., vol. 9, no. 01, pp. 54–59, 2021, doi: 10.33884/jif.v9i01.3741

[10] A. Putri et al., "Komparasi Algoritma K-NN, *Naïve Bayes* dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir," MALCOM Indones. J. Mach. Learn. Comput. Sci., vol. 3, no. 1, pp. 20–26, 2023, doi: 10.57152/malcom.v3i1.610.

[11] A. I. Putri, Y. Syarif, P. Jayadi, F. Arrazak, and F. N. Salisah, "Implementasi Algoritma Decision Tree dan *Support Vector Machine* (SVM) untuk Prediksi Risiko Stunting pada Keluarga," MALCOM Indones. J. Mach. Learn. Comput. Sci., vol. 3, no. 2, pp. 349–357, 2024, doi: 10.57152/malcom.v3i2.1228.

[12] M. Fluorida Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode *Naïve Bayes* Dan *Support Vector Machine* (SVM)," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 2, no. 9, pp. 3112–3123, 2018, [Online]. Available: http://j-ptiik.ub.ac.id.

[13] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," Ecological Modelling, Vol. 406, pp. 109-120, 2019, doi: 10.1016/j.ecolmodel.2019.06.002.

[14] G. N. Masacgi and M. S. Rohman, "Optimasi Model Algoritma Klasifikasi menggunakan Metode Bagging pada Stunting Balita," Edumatic J. Pendidik. Inform., vol. 7, no. 2, pp. 455–464, 2023, doi: 10.29408/edumatic.v7i2.23812.

[15] D. Nurmalasari, T. I. Hermanto, and I. M. Nugroho, "Perbandingan Algoritma SVM , KNN dan NBC Terhadap Analisis Sentimen Aplikasi Loan Service," J. Media., vol. 7, pp. 1521–1530, 2023, doi: 10.30865/mib.v7i3.6427.

[16] Y. Roh, G. Heo and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347, 1 April 2021, doi: 10.1109/TKDE.2019.2946162.

[17] E. S. K. Tiu, Y. F. Huang, J. L. Ng, N. AlDahoul, A. N Ahmed and A. Elshafie, "An evaluation of various data pre-processing techniques with machine learning models for water level prediction," Nat Hazards, vol. 110, pp: 121–153, 2022, doi: 10.1007/s11069-021-04939-8.

[18] Q. H. Nguyen, HB. Ly, L. S. Ho, N. Al-Ansari, H. V. Le, V. Q. Tran, I. Prakash, and B. T. Pham, "Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil," Math. Problems in Engineering, 2021, doi: 10.1155/2021/4832864.

[19] Ž. Đ. Vujović, "Classification Model Evaluation Metrics," Int. J. of Advanced Computer Science and Applications, Volume 12, No. 6, 2021.

[20] M. R. Wayahdi, D. Syahputra, and S. Hafiz Nanda Ginting, "Evaluation of The K-Nearest Neighbor Model with K-Fold Cross Validation on Image Classification", infokum, vol. 9, no. 1,Desember, pp. 1-6, Dec. 2020.

[21] Salsabila, S. Martha, and W. Andani, "Komparasi Algoritma K-Nearest Neighbor Dengan Euclidean Distance Dan Manhattan Distance Untuk Klasifikasi Stunting Balita," Bul. Ilm. Math. Stat. dan Ter., vol. 13, no. 2, pp. 285–292, 2024.

[22] F. M. N. Akbar, "Metode KNN (K-Nearest Neighbor) untuk Menentukan Kualitas Air" Jurnal Tekno Kompak," vol. 18, no. 1, 2024.

[23] M. Banurea, D. Betaria Hutagaol, and O. Sihombing, "Klasifikasi Penyakit Stunting Dengan Menggunakan Algoritma *Support Vector Machine* Dan Random Forest," J. TEKINKOM, vol. 6, no. 2, pp. 540–549, 2023, doi: 10.37600/tekinkom.v6i2.927.

[24] G. Sanhaji, A. Febrianti, and Hidayat, "Aplikasi DIATECT Untuk Prediksi Penyakit Diabetes Menggunakan SVM Berbasis Web" Jurnal Tekno Kompak," vol. 18, no. 1, 2024.

[25] W. Agustina, M. T. Furqon, and B. Rahayudi, "Implementasi Metode *Support Vector Machine* (SVM) Untuk Klasifikasi Rumah Layak Huni (Studi Kasus: Desa Kidal Kecamatan Tumpang Kabupaten Malang)," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 2, no. 10, pp. 3366–3372, 2018, [Online]. Available: http://j-ptiik.ub.ac.id

[26] W. Cholid Wahyudin, "Klasifikasi Stunting Balita Menggunakan *Naïve Bayes* Dengan Seleksi Fitur Forward Selection," J. Bisnis Digit. dan Sist. Inf., vol. 1, pp. 71–74, 2020, [Online]. Available:

https://ejr.stikesmuhkudus.ac.id/index.php/jikoma/article/view/1220%0Ahttps://ejr.stikesmuhkudus.ac.id/index.php/jikoma/article/viewFile/1220/760.

[27] A. Damuri, U. Riyanto, H. Rusdianto, and M. Aminudin, "Implementasi Data Mining dengan Algoritma *Naïve Bayes* Untuk Klasifikasi Kelayakan Penerima Bantuan Sembako," JURIKOM (Jurnal Ris. Komputer), vol. 8, no. 6, p. 219, 2021, doi: 10.30865/jurikom.v8i6.3655.

[28] E. Martantoh and N. Yanih, "Implementasi Metode *Naïve Bayes* Untuk Klasifikasi Karakteristik Kepribadiaan Siswa Di Sekolah MTS Darussa'adah Menggunakan Php Mysql," J. Teknol. Sist. Inf., vol. 3, no. 2, pp. 166–175, 2022, doi: 10.35957/jtsi.v3i2.2896.

[29] Karsito and S. Santi, "Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma *Naïve Bayes* Di Perumahan Azzura Residencia," J. Teknol. Pelita Bangsa, vol. 9, pp. 43–48, 2019.

[30] I. W. Saputro and B. W. Sari, "Uji Performa Algoritma *Naïve Bayes* untuk Prediksi Masa Studi Mahasiswa," Creat. Inf. Technol. J., vol. 6, no. 1, p. 1, 2020, doi: 10.24076/citec.2019v6i1.178.

[31] P. Romadloni, B. Adhi Kusuma, and W. Maulana Baihaqi, "Komparasi Metode Pembelajaran Mesin Untuk Implementasi Pengambilan Keputusan Dalam Menentukan Promosi Jabatan Karyawan," JATI (Jurnal Mhs. Tek. Inform., vol. 6, no. 2, pp. 622–628, 2022, doi: 10.36040/jati.v6i2.5238.

[32] W. I. Rahayu, C. Prianto, and E. A. Novia, "Perbandingan Algoritma K-Means dan *Naïve Bayes* untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan pada PT. Pertamina (Persero)," J. Tek. Inform., vol. 13, no. 2, pp. 1–8, 2021.