

## COMPARISON OF XGBOOST AND SVM FOR SENTIMENT ANALYSIS MERAH PUTIH COOPERATIVE POLICY

Wahyu Pratama Lasaleng<sup>1</sup>, Fahrezal Zubedi<sup>2\*</sup>, Siti Nurmardia Abdussamad<sup>3</sup>

<sup>1,2,3</sup>Department of Statistics, Faculty of Mathematics and Natural Sciences, Gorontalo State University  
Jl. Prof. Dr. Ing. B.J. Habibie, Moutong, Bone Bolango, 96554, Gorontalo

Corresponding author's e-mail: \* [fahrezal@ung.ac.id](mailto:fahrezal@ung.ac.id)

### ABSTRACT

#### Article History:

Received: May 25, 2026

Revised: June 06, 2026

Accepted: June 16, 2026

Published: June 30, 2026

Available online.

#### Keywords:

Extreme Gradient Boosting;

Hyperparameter Tuning;

Sentiment Analysis;

Support Vector Machine;

TF-IDF.

Large volumes of textual data have been generated by the rapid growth of social media, making sentiment analysis an effective approach for understanding public perceptions of government policies. However, text classification still faces challenges such as high feature dimensionality, class imbalance, and non-linear relationships within the data. This study used data from the X social media platform to evaluate the performance of the Support Vector Machine (SVM) and XGBoost algorithms in classifying public sentiment toward the Koperasi Desa Merah Putih policy. The dataset consisted of 1,074 tweets collected through a scraping technique between July 21 and November 4, 2025, comprising 800 positive tweets and 274 negative tweets. The research process included data preprocessing, feature extraction using TF-IDF, data splitting with an 80:20 ratio, and hyperparameter tuning using GridSearchCV with 5-fold cross-validation. The models were evaluated using accuracy, precision, recall, and F1-score. Hyperparameter tuning successfully enhanced the performance of both models, with SVM benefiting the most from the optimisation process. The findings demonstrated that both models achieved strong classification performance; however, SVM outperformed XGBoost. The SVM model achieved an accuracy of 95%, with more balanced precision, recall, and F1-score values across both sentiment classes, whereas XGBoost achieved an accuracy of 91% and showed limitations in detecting negative sentiment as the minority class. The data exploration results also indicated that most users expressed positive sentiment toward the Koperasi Desa Merah Putih policy. Nevertheless, this study has several limitations, particularly the use of TF-IDF-based feature representation, which does not capture semantic relationships or sarcasm in textual data. The novelty of this study lies in the comparison of SVM and XGBoost with hyperparameter tuning using GridSearchCV in the context of sentiment analysis of the Koperasi Desa Merah Putih policy, a topic that has received limited attention in previous studies.



This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-ShareAlike 4.0 International License.

#### How to cite this article:

Lasaleng, W. P., Zubedi, F., Abdussamad, S. N., "COMPARISON OF XGBOOST AND SVM FOR SENTIMENT ANALYSIS MERAH PUTIH COOPERATIVE POLICY", Journal Statistika dan Aplikasinya, vol. 10, iss. 1, pp. 54 - 67, June 2026

Copyright © 2026 Author(s)

Journal homepage: <https://journal.unj.ac.id/unj/index.php/statistika>

Journal e-mail: [jsa@unj.ac.id](mailto:jsa@unj.ac.id)

Research Article · Open Access

## 1. INTRODUCTION

Advances in digital technology have brought significant impacts on the development of data, both in terms of its form and characteristics. Data growth is no longer limited to numerical data alone but has expanded to include text-based data generated through various social media platforms. Platforms like X are increasingly playing a key role as a vital channel for gauging public sentiment, thereby enabling analysis of public opinion on various policies, public figures, and social events [1]. In this context, sentiment analysis has become an important approach for identifying and classifying public sentiment expressed through digital text [2]. Sentiment analysis generally integrates machine learning techniques into the processing and classification of text data, thereby enabling the identification of sentiments into positive and negative categories [3].

Text classification is a method in supervised machine learning that employs annotated data to identify specific patterns, allowing it to be applied to predict outcomes for unseen data [4]. The text classification process typically involves multiple steps, beginning with preprocessing, followed by feature extraction, training the model, and ending with model evaluation [2]. Several algorithms of machine learning, including Naïve Bayes, Extreme Gradient Boosting (XGBoost), Random Forest, Logistic Regression, and Support Vector Machine (SVM), have been applied to text classification. However, some of these algorithms still face limitations when applied to text data with very high feature dimensions. For instance, Naïve Bayes performs less well on high-dimensional text data because it relies on the assumption of feature independence and is highly sensitive to feature selection [5], [6]. Logistic Regression has limitations in modeling non-linear relationships [7]. Meanwhile, the Random Forest model tends to experience a decrease in accuracy on imbalanced datasets because the model shows a stronger tendency to classify data into the dominant class, resulting in weaker recognition of the minority class [8]. These conditions show that modern text data classification still faces challenges, such as complexity and high feature dimensionality, class imbalance, and non-linear relationships between features.

The SVM and XGBoost algorithms have been shown to effectively overcome the challenges of sentiment classification in prior research. Previous research shows that SVM successfully outperforms Random Forest with an accuracy of up to 93% in sentiment analysis of application reviews, and remains stable even when the amount of data used is very large [9]. Meanwhile, other empirical studies show that XGBoost demonstrates very good performance with an F1-Score value of 0.941, exceeding Naïve Bayes on an imbalanced dataset [10]. These findings indicate that the SVM and XGBoost algorithms each have different advantages, so a comparison between the two is relevant and important to be studied more deeply, especially in the context of sentiment analysis of government policies.

Although literature on sentiment analysis using SVM and XGBoost has been sufficiently developed, there has been no research that specifically compares these two algorithms in the context of public perception toward newly launched government policies, particularly within the Indonesian policy landscape. In addition, previous studies generally rely on default parameters or manual tuning, which do not necessarily produce optimal model performance. To fill this gap, this study conducts a structured hyperparameter tuning process by employing a grid search method alongside cross-validation across both algorithms, thereby producing a more rigorous, fair, and reproducible comparison. This approach represents a methodological contribution that distinguishes this study from previous works, while also offering a more solid empirical basis for choosing machine learning-based, accurate, and adaptive text classification models.

The object of this study is the Koperasi Desa Merah Putih, a national economic policy launched by the Indonesian government in mid-2025 that has triggered various public responses on social media. Some people view this policy as a strategic step in strengthening the village economy, while others criticize the potential overlap of institutions and the risk of misuse of public funds at the village level [11]. This polarization of perception makes the Koperasi Desa Merah Putih policy a highly suitable object for sentiment analysis, as it naturally contains public expressions that are both positive and negative. Analysis of these sentiments provides an objective overview of the level of public acceptance and trust in this policy, and can also serve as an early social indicator of the potential success or resistance of the policy in the field.

This study uses text data from the social media site X to examine how well the SVM and XGBoost algorithms categorize public opinions of the Koperasi Desa Merah Putih policy, which turned into

evaluated primarily based on accuracy, precision, recall, and F1-score metrics, as well as to determine the high quality model in describing the general sentiment of the general public towards the policy.

## 2. METHODS

### Material and Data

This study utilised secondary data collected from user tweets containing the phrase “Koperasi Merah Putih” on the social media platform X. A Python-based tweet harvesting tool was used to collect the data. Data collection was conducted between 21 July and 4 November 2025, with nationwide coverage across Indonesia. A total of 1,074 tweets were successfully collected, comprising 800 positive tweets and 274 negative tweets, resulting in an imbalanced class distribution ratio of approximately 3:1 (positive: negative). All collected tweets were included in this study using a saturation sampling approach, in which all members of the population are included as study samples [12].

### Research Method

#### Research Flow

This study’s main goal is to assess and compare the efficacy of two categorization algorithms, XGBoost and SVM, in classifying public sentiment about Koperasi Desa Merah Putih’s (KDMP) policies on the social media platform X. The performance of the algorithms will be assessed using accuracy, precision, recall, and F1-score to determine the most effective model. Figure 1 shows the steps taken in the research that has been conducted.

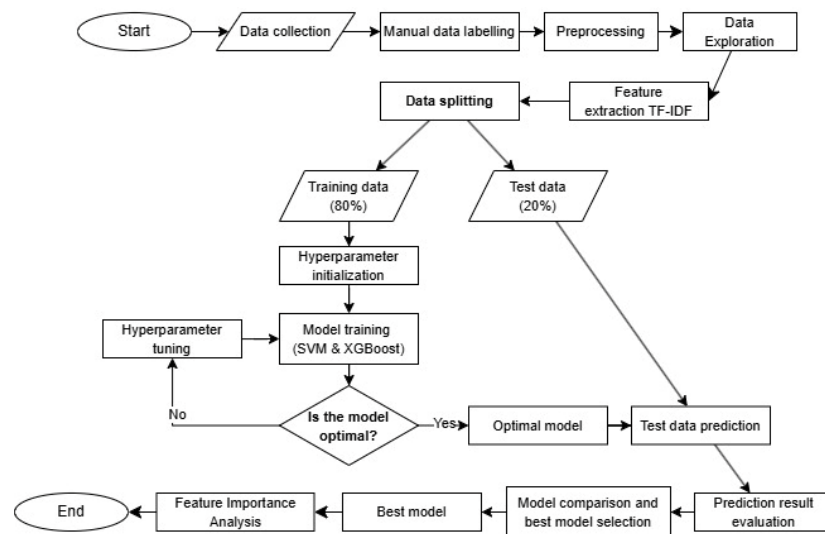


Figure 1. Research Flow

#### Preprocessing

Preprocessing is a crucial stage that aims to transform unprocessed data into a more structured form, thereby improving data quality and accuracy. This process ensures that machine learning models can function optimally and produce more accurate and reliable analyses [13]. In sentiment analysis, preprocessing generally consists of several procedures, such as converting text to lowercase, removing unnecessary characters or symbols, tokenizing the text, eliminating stopwords, and applying stemming [14]. In the case folding stage, all letters in the text are converted into lowercase. Next, data cleaning is performed to remove elements that are irrelevant or unimportant for the analysis process. After that, stopwords removal is used to get rid of frequently used terms that don't have much significance in the study. Tokenization is then used to divide the text into smaller components, such as words or tokens. Finally, stemming is applied to remove affixes from each word so that the root form of the word is obtained [14].

*TF-IDF Feature Extraction*

Feature extraction is carried out to transform textual data into numerical representations so that it can be processed by machine learning algorithms [15]. In this study, Term Frequency–Inverse Document Frequency (TF-IDF) is applied as the feature representation method. TF-IDF assigns a weight to each term by considering two main aspects: the frequency of a term within a document and the importance of that term across the entire corpus. TF indicates how often a specific term appears in a document, while IDF reflects the degree of uniqueness or significance of a term among all documents [16]. Mathematically, it is shown in the following equation [16]:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \tag{1}$$

where:

$$TF(t, d) = \frac{\text{Number of occurrences of term } t \text{ in document } d}{\text{Total number of term in document } d} \tag{2}$$

$$IDF(t) = \log \frac{\text{Total number of documents}}{\text{The number of documents countaining term } t} \tag{3}$$

*Extreme Gradient Boosting (XGBoost)*

Extreme Gradient Boosting, commonly known as XGBoost, is a machine learning algorithm that belongs to the ensemble learning approach and is built upon the gradient boosting framework [17]. Its training process is carried out iteratively using the gradient boosting principle, which optimises the loss function to minimise prediction errors through gradient calculations at each iteration [18]. The final prediction result in XGBoost is obtained from the summation of the prediction outputs generated by each decision tree, which are constructed step-by-step [17]. Mathematically, the XGBoost model is represented as follows [19]:

$$\hat{y} = \sum_{t=1}^T f_t(x) \tag{4}$$

with the objective function, which has been optimised at each iteration, formulated as follows [19]:

$$L(t) = \sum_{i=1}^N l(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)) + \gamma L + \frac{1}{2} \lambda \sum_{j=1}^L w_j^2 \tag{5}$$

Notation :

- $l(y_i, \hat{y})$  : loss function between the expected value  $\hat{y}$  and the true label  $y_i$
- $L$  : number of leaves in the tree
- $w_j$  : weight of the j-th leaf
- $\gamma, \lambda$  : regularization parameters

*Support Vector Machine (SVM)*

SVM is an approach for maximizing the margin or distance from the nearest observations from each category in order to find the optimal dividing boundary between two classes. Support vectors refer to the observations located nearest to the hyperplane, whereas the decision boundary separating these classes is the hyperplane itself [20]. In principle, SVM operates in a linear manner; however, it has been adapted to handle non-linear problems using the kernel trick, which maps the dataset into a higher-dimensional feature space so that the classification process can be performed optimally in that space.

Kernels commonly used in SVM include linear, polynomial, radial basis function, and sigmoid [21]. Mathematically, the equation forming the hyperplane in the SVM model is expressed as follows [22]:

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (6)$$

Notation [23]:

- $\mathbf{w}$  : weight vector of the support vector machine
- $\mathbf{x}$  : input data or feature vector
- $b$  : bias term

Finding the ideal hyperplane that optimizes the separation between the nearest data points from the two categories, which are referred to as support vectors, serves as the primary goal of the SVM method. Mathematically, the margin value in the SVM model can be calculated using the following equation [22]:

$$Margin = \frac{2}{\|\mathbf{w}\|} \quad (7)$$

To obtain the maximum margin, SVM formulates the process of finding the optimal hyperplane as an optimization problem as follows [22]:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad (8)$$

Notation [23]:

- $\mathbf{w}$  : weight vector of the support vector machine
- $\mathbf{x}_i$  : input data or feature vector
- $y_i$  : true label
- $b$  : bias term

Because the optimization function has constraints, the Lagrange Multiplier method is used, which is formulated as follows [20]:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \quad (9)$$

Notation :

- $w$  : weight vector of the support vector machine
- $x_i$  : input data or feature vector
- $y_i$  : true label
- $b$  : bias term
- $\alpha_i$  : Lagrange multiplier

### *Hyperparameter Tuning*

Finding the ideal combination of hyperparameters is known as hyperparameter tuning, which aims to identify the optimal configuration to significantly improve model performance [24]. One commonly used method in this process is Grid Search. To identify the optimal combination based on model performance, this algorithm evaluates all possible combinations of predetermined hyperparameter values and selects the one that produces the lowest error rate [25]. To obtain more accurate evaluation results, Grid Search is often combined with Cross Validation, which is known as Grid Search Cross Validation [26].

### 3. RESULTS

#### Data Collection and Labeling

The data used in this study were obtained from user tweets on the social media platform X containing the keyword “Koperasi Merah Putih,” posted within the period from July 21 to November 4, 2025. The collected data amounted to 1,074 tweets, which were then labeled into two sentiment categories, namely positive and negative. The labeling process was carried out manually by the researcher by reading and understanding the content of each tweet to determine the sentiment expressed within it. The labeling results were then validated by an Indonesian language expert lecturer to minimize errors in the labeling process.

#### Data Preprocessing

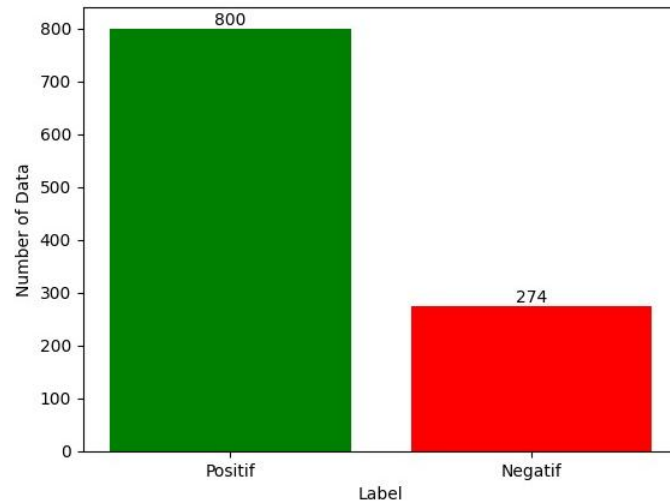
Before entering the model training stage, the dataset must go through a preprocessing stage first so that the data can be optimally utilized in the analysis process as well as in subsequent processing stages. This study employs five preprocessing steps, namely case folding, cleaning, stopword removal, tokenization, and stemming. An example of the output from each preprocessing stage is presented in Table 1 below.

**Table 1. Example of Preprocessing Results**

Process	Input	Output
Case Folding	Setidaknya terdapat 13 manfaat dari kehadiran Koperasi Merah Putih yang telah diresmikan oleh Presiden Prabowo pad <a href="https://t.co/A5or3h8RWt">https://t.co/A5or3h8RWt</a>	setidaknya terdapat 13 manfaat dari kehadiran koperasi merah putih yang telah diresmikan oleh presiden prabowo pad <a href="https://t.co/a5or3h8rwt">https://t.co/a5or3h8rwt</a>
Cleaning	setidaknya terdapat 13 manfaat dari kehadiran koperasi merah putih yang telah diresmikan oleh presiden prabowo pad <a href="https://t.co/a5or3h8rwt">https://t.co/a5or3h8rwt</a>	setidaknya terdapat manfaat dari kehadiran yang telah diresmikan oleh presiden pad
Stopword Removal	setidaknya terdapat manfaat dari kehadiran yang telah diresmikan oleh presiden pad	terdapat manfaat kehadiran telah diresmikan presiden pad
Tokenizing	terdapat manfaat kehadiran telah diresmikan presiden pad	['terdapat', 'manfaat', 'kehadiran', 'telah', 'diresmikan', 'presiden', 'pad']
Stemming	['terdapat', 'manfaat', 'kehadiran', 'telah', 'diresmikan', 'presiden', 'pad']	dapat manfaat hadir telah resmi presiden pad

#### Data Exploration

Data exploration is conducted to understand the distribution and characteristics of sentiment within the dataset. At this stage, visualizations are utilized to give a preliminary overview of the data patterns and trends before going on to the modelling step. The sentiment distribution is visualized using a bar chart to show the proportion of data in each sentiment category. In addition, a word cloud is used to display the dominant words that frequently appear in each sentiment category within the dataset. The sentiment distribution visualization can be seen in Figure 2.



**Figure 2. Sentiment Distribution**

Figure 2 shows that as many as 800 users in the dataset expressed positive sentiment toward the Koperasi Desa Merah Putih program, while the remaining 274 users expressed negative sentiment. This indicates that the majority of users expressed positive sentiment toward the Koperasi Desa Merah Putih program. To obtain a clearer picture of the characteristics of user sentiment, a word cloud visualization was performed for each sentiment category in the dataset.



**Figure 3. Wordcloud Visualization of Positive and Negative Sentiments**

Figure 3 shows that in the positive sentiment word cloud category, the dominant words that appear include “program,” “presiden,” “buat,” “ekonomi,” “desa,” “jadi,” “makin,” “mandiri,” “lewat,” “usaha,” “bareng,” “rakyat,” “dukung,” and “bantu.” The emergence of these words reflects users’ positive responses toward the Koperasi Desa Merah Putih program, particularly related to its benefits, support, and its impact on community life in improving the village economy.

Meanwhile, in the negative sentiment word cloud category, the dominant words that appear include “dana,” “program,” “buat,” “korupsi,” “kalau,” “ga,” “jadi,” “bunga,” “lebih,” and “banyak.” The emergence of these words reflects users’ negative responses toward the Koperasi Desa Merah Putih program, particularly in terms of fund management, potential misuse, and concerns that interest or financial burdens may become greater if the program is not implemented properly.

### TF-IDF Feature Extraction

This study uses the TF-IDF method in the feature extraction process. This method is used to calculate the weight of each word by considering its frequency of occurrence in a document as well as across the entire collection of documents (corpus). The resulting weights indicate the level of importance of a word in representing the content of a document, so that more important words will have higher values compared to others [27]. The results of this method are a two-dimensional matrix in which each row shows the TF-IDF weight assigned to every term in each tweet, while each column corresponds to a particular term from the total dataset. The resulting image generated in this study has a size of  $1,074 \times$

2,600. This size indicates that there are 1,074 documents or tweets and 2,600 features or words. An example of the TF-IDF feature extraction output is presented in Table 2.

**Table 2. Example of Preprocessing Results**

Index	abah	abai	...	ad	ada	...	zulhas
0	0	0	...	0	0	...	0
1	0	0	...	0	0	...	0
2	0	0	...	0	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1071	0	0	...	0.1885	0	...	0
1072	0	0	...	0	0.0392	...	0
1073	0	0	...	0	0.2744	...	0

**Data Splitting**

In this study, data splitting was performed using an 80:20 ratio. This ratio was chosen because, based on previous research [28] an 80:20 data split is able to produce more stable and accurate model performance. From this ratio, 859 data or 80% of the total dataset were obtained as training data used to build and tune the model parameters, whereas 20% of the entire dataset, or 215 data, was utilized as testing data to assess how well the trained model performed.

**Model Training**

To maximize both models’ performance during model training, the researcher employed a hyperparameter tuning technique. The optimization process was carried out using Grid Search Cross Validation, a method that works by testing all combinations of predefined hyperparameters to obtain the most optimal configuration. The determination of the XGBoost model hyperparameters in this study refers to the works of Zhao et al. [29] and Kahraman [30]. In this study, the XGBoost model produced 972 hyperparameter combinations from the hyperparameter tuning process. Table 3 shows the XGBoost model hyperparameters.

**Table 3. XGBoost Hyperparameters**

Parameter	Value
n_estimators	(100, 200, 300)
learning_rate	(0.01, 0.1, 1)
max_depth	(3, 5, 7)
min_child_weight	(1, 3, 5)
gamma	(0, 0.1, 1)
subsample	(0.8, 1.0)
colsample_bytree	(0.8, 1.0)

Meanwhile, for the SVM model, 600 hyperparameter combinations were obtained from the hyperparameter tuning process. The determination of the SVM model parameters in this study refers to the works of Hidayaturohman [31] and Adi et al. [32]. The hyperparameters used in the SVM model are presented in Table 4.

**Table 4. SVM Hyperparameters**

Parameter	Value
C	(0.1, 1, 10, 100, 1000)
gamma	('scale', 'auto', 0.001, 0.01, 0.1, 1)
kernel	('rbf', 'linear', 'poly', 'sigmoid')
degree	(2, 3, 4, 5, 6)

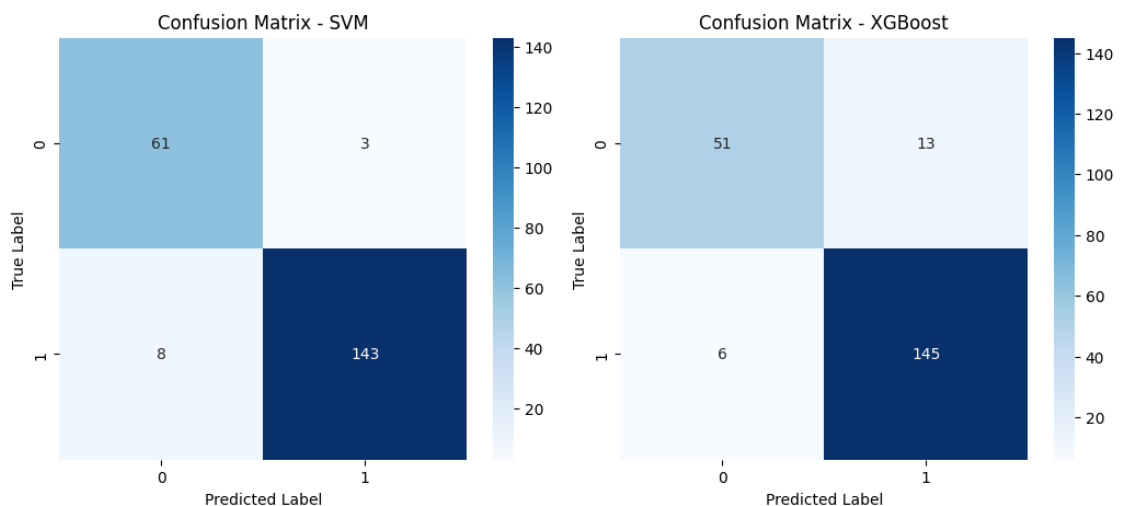
After determining the initial hyperparameters for both models, GridSearchCV applies each hyperparameter combination to the training data and evaluates it using a five-fold cross-validation scheme. After the training dataset is separated into five subsets of the same size, in each iteration, one part is used as the test set, while the other four parts are used as the training set. This procedure is repeated in turn so that each partition has been used as the test set exactly once. In each iteration, the accuracy of each fold is computed and subsequently averaged to yield a single performance score for each parameter combination. This process is repeated for all predefined hyperparameter combinations. After all parameter combinations have been evaluated, GridSearchCV compares the average accuracy values of each combination and selects the best-performing hyperparameter combination, namely the combination with the highest average accuracy value.

With an average accuracy of 0.878, this procedure yielded the ideal hyperparameter combination for the XGBoost model. `n_estimators = 300`, `learning_rate = 0.1`, `max_depth = 3`, `min_child_weight = 1`, `gamma = 0`, `subsample = 1.0`, and `colsample_bytree = 0.8`, are the optimal hyperparameter combinations. The XGBoost model is constructed using this hyperparameter combination, and its generalization performance is assessed using test data.

Meanwhile, for the SVM model, the optimal hyperparameter combination was obtained with an average accuracy of 0.919. There were five hyperparameter combinations that produced identical performance values; therefore, the selection of the best combination was not only based on the highest average accuracy but also considered model simplicity. The best hyperparameter combination is as follows: `C = 1000`, `degree = 2`, `gamma = 0.01`, and `kernel = 'rbf'`. The SVM model is then constructed using this combination of hyperparameters, and its generalization performance is assessed using the test data.

## Model Evaluation

In this study, the effectiveness of the overall model was assessed through a confusion matrix and other performance measures, namely accuracy, precision, recall, and F1-score. The predictions generated from the testing dataset are used to create the confusion matrix, using the optimal model obtained from the previous training process. The confusion matrix for the XGBoost and SVM models is presented in Figure 4.

**Figure 4. Confusion Matrices of SVM and XGBoost Models**

The results of the confusion matrix in Figure 4 are used as the basis for calculating model evaluation metrics, namely accuracy, which shows the overall level of correctness of the model in classifying data; precision, a measures the proportion of correct predictions among all predicted instances of a sentiment category; recall, which demonstrates the models’s ability to distinguish between real positive and negative data; and F1-score, which is the recall and precision harmonic mean.

**Table 5. Evaluation Metrics of XGBoost and SVM Models**

Sentiment Category	XGBoost				SVM			
	P	R	F1	A	P	R	F1	A
Negative (0)	0.89	0.80	0.84	0.91	0.88	0.95	0.92	0.95
Positive (1)	0.92	0.96	0.94		0.98	0.95	0.96	

Notes :

- P : Precision      F1 : F1-score
- R : Recall        A : Acuraccy

Table 5 shows that the XGBoost and SVM perform well in classification, with respective accuracies of 0.91 and 0.95. However, since there is an imbalance in the distribution between sentiment categories, the evaluation does not only focus on the accuracy metric, but also uses additional evaluation criteria, including the F1-score, precision, and recall, to provide a more comprehensive assessment of model performance.

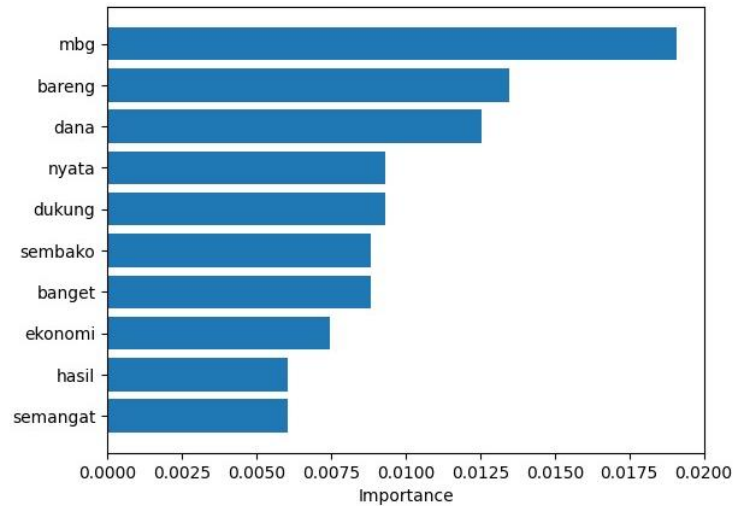
In the XGBoost model, the evaluation results show fairly good performance, particularly in the positive sentiment category. With a recall of 0.96 and an accuracy value of 0.92 for the positive sentiment category, the model is highly successful in identifying positive data with a comparatively low mistake rate. However, as seen by a recall value of 0.80, the model’s performance in the negative sentiment category is still inferior to that of the positive sentiment category, although the precision reaches 0.89. This condition indicates that some negative data has not been properly recognized and tends to be misclassified as positive data. As a result, the F1-score for the negative sentiment category only reaches 0.84, which is lower than the F1-score for the positive sentiment category, which reaches 0.94. The difference in F1-score values between the two categories reflects the limitations of the XGBoost model in handling data imbalance, particularly in improving prediction performance for the minority class.

Meanwhile, in the SVM model, the evaluation results show more stable and balanced performance across both sentiment categories. In the negative sentiment category, the model obtained a precision value of 0.88, a recall of 0.95, and an F1-score of 0.92. Meanwhile, the positive class’s precision values reach 0.98, with recall and F1-score values of 0.95 and 0.96, respectively. This model can accurately identify the majority of the observations, as indicated by its strong recall scores in both emotion categories, while the high precision value, particularly in the positive sentiment category, indicates a low level of prediction error. Overall, the SVM model demonstrates better generalization ability in handling imbalanced data compared to the XGBoost model, with more consistent performance across each sentiment category.

Thus, it can be concluded that although both models show good performance, the SVM model has more balanced performance between the positive and negative sentiment categories, whereas XGBoost tends to perform better in the positive sentiment category and still has limitations in detecting the negative (minority) sentiment category.

**Feature Importance Analysis**

At this stage, an analysis was conducted on the important features that contributed to the formation of the model with the best performance, namely the SVM model. To obtain a deeper insight into the data patterns and characteristics of the data used by the model in the sentiment classification process, this study focuses on identifying the factors that have the greatest influence on sentiment classification.



**Figure 5. Feature Importance of the Best-Performing SVM Model**

According to Figure 5, the features that contribute most significantly to the development of the SVM model include “mbg”, “bareng”, “dana”, “nyata”, “dukung”, “sembako”, “banget”, “ekonomi,” and “hasil”, “semangat”. These findings are consistent with the previous word cloud visualization, where features such as “dana” and “mbg” were more dominant in negative sentiments, while features such as “dukung,” “bareng,” and “nyata” appeared more frequently in positive sentiments. The feature “dana” indicates discussions related to budget management and the potential misuse of the program, whereas “mbg” reflects discussions linking the Koperasi Desa Merah Putih program with the MBG program, which has generated various criticisms from the public. On the other hand, features such as “dukung” and “bareng” represent public support and positive perceptions regarding the benefits of the program. Overall, the identified important features demonstrate that the SVM model is capable of capturing linguistic characteristics that distinguish positive and negative sentiments within the dataset used.

#### 4. DISCUSSIONS

The results of the study show that the Support Vector Machine (SVM) algorithm performs better than the Extreme Gradient Boosting (XGBoost) algorithm in classifying Indonesian-language social media text sentiment. SVM achieved an accuracy of 95% with more balanced precision, recall, and F1-score values across both sentiment categories, while XGBoost achieved an accuracy of 91% but experienced limitations in detecting the negative (minority) sentiment category. These findings are consistent with the study [9], which shows that the SVM algorithm has superior performance with an accuracy of 93% in sentiment classification. In contrast to XGBoost, the findings of this study are inconsistent with previous research [10], which states that XGBoost is able to handle imbalanced data effectively.

The superiority of SVM in this study is influenced by its ability to construct an optimal hyperplane through the use of an RBF kernel with optimal parameters  $C = 1,000$ ,  $\text{degree} = 2$ , and  $\text{gamma} = 0.01$ , which is effective in separating data in a high-dimensional and sparse TF-IDF feature space. Furthermore, the hyperparameter optimization process using Grid Search Cross-Validation also improves the model to generalize more effectively and achieve better accuracy. On the other hand, XGBoost, as a model that is based on decision trees, is less optimal in handling this type of feature representation, especially when the data distribution is imbalanced. The imbalance between positive sentiment (800 tweets) and negative sentiment (274 tweets), with a ratio of approximately 3:1, is the main factor contributing to the decrease in recall for the negative sentiment category, as the model tends to be biased toward the dominant class.

In addition, the data exploration results in this study show that out of 1,074 tweets, 800 users expressed positive sentiment and 274 users expressed negative sentiment toward the Koperasi Desa Merah Putih program. This indicates that public responses to the policy are dominated by positive sentiment. Furthermore, based on the word cloud visualization, it can be seen that the dominant words in the positive sentiment category include “program,” “ekonomi,” “desa,” “mandiri,” “dukung,” and

“bantu”. The emergence of these words reflects positive user responses toward the Koperasi Desa Merah Putih program, particularly related to its benefits, support, and its impact on community life in improving the village economy. Meanwhile, in the negative sentiment category, the dominant words that appear include “dana,” “korupsi,” and “bunga.” These words indicate public concerns regarding fund management, the potential for misuse of authority, and the possibility of increased financial burdens if the program is not implemented effectively. In addition, the frequent occurrence of the term “mbg” indicates that discussions on the Koperasi Desa Merah Putih program are often associated with the Makan Bergizi Gratis (MBG) program, which has received public criticism due to issues related to its implementation and supervision [33].

Overall, this study provides empirical evidence that SVM with hyperparameter tuning optimization is a more optimal choice compared to XGBoost for sentiment classification of Indonesian-language social media text based on TF-IDF. The findings of this study can be used by the government as data-driven evaluation material regarding public acceptance of the Koperasi Desa Merah Putih (KDMP) policy, as well as provide a guide for scholars and professionals creating more flexible public sentiment tracking systems. Further research is recommended to develop feature extraction techniques that can better capture the semantic relationship between words, such as word embeddings (Word2Vec and FastText), or contextual embedding-based models such as IndoBERT. In addition, it is also suggested to perform fine-tuning to make the model more sensitive to sarcastic sentences, so that classification errors caused by ambiguity in meaning can be minimized.

## 5. CONCLUSION

In this study, we used text data from the social media site X to assess how well the Support Vector Machine (SVM) and XGBoost algorithms perform in categorizing public attitudes regarding the Koperasi Desa Merah Putih regulation, as well as to determine the best model in representing public sentiment toward the policy. Based on the results of this study, both models show good classification performance; however, Support Vector Machine demonstrates superior performance compared to Extreme Gradient Boosting. SVM achieved an accuracy of 95%, while XGBoost achieved 91%. In addition, SVM also shows more balanced, consistent results based on recall, precision, and F1-score across both sentiment categories, whereas XGBoost has limitations in detecting negative sentiment as the minority class. Thus, it can be concluded that the research objectives have been achieved, and the Support Vector Machine is the best model for classifying public sentiment toward the Koperasi Desa Merah Putih policy in this study.

## 6. ACKNOWLEDGMENTS

This research would not have been possible without the support and contributions of various parties. The author would like to express sincere gratitude to the Statistics Study Program at the Faculty of Mathematics and Natural Sciences, Universitas Negeri Gorontalo (UNG), for providing academic resources and support throughout the research process. The author also expresses gratitude to the supervisors for their invaluable advice, helpful criticism, and unwavering commitment during the study process.

## 7. REFERENCES

- [1] H. Permadi, “Penambangan Opini Pada Data Multidomain Memanfaatkan Stream Big Data Twitter,” *Semin. Nas. Off. Stat.*, vol. 2022, no. 1, pp. 751–758, 2022, doi: 10.34123/semnasoffstat.v2022i1.1144.
- [2] K. L. Tan, C. P. Lee, and K. M. Lim, “A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research,” *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074550.
- [3] R. Cahyani, I. Rozas, dan N. Yalina, “Analisis Sentimen Pada Media Sosial Twitter Terhadap Tokoh Publik Peserta Pilpres 2019,” *MATICS J. Ilmu Komput. dan Teknol. Inf. (Journal Comput. Sci. Inf. Technol.)*, vol. 12, no. 1, pp. 79–86, 2020, doi: 10.18860/mat.v12i1.8356.
- [4] D. Purnamasari *et al.*, *Pengantar Metode Analisis Sentimen*. Depok: Gunadarma, 2024.
- [5] A. F. Cahyanti, R. Saptono, and S. W. Sihwi, “Penentuan model terbaik pada metode Naïve

- Bayes Classifier dalam menentukan status gizi balita dengan mempertimbangkan independensi parameter,” *ITSMART J. Teknol. dan Inf.*, vol. 4, no. 1, pp. 28–35, 2016.
- [6] R. Wibowo, M. A. Soeleman, and A. Affandy, “Hybrid Top-K Feature Selection to Improve High-Dimensional Data Classification Using Naïve Bayes Algorithm,” *Sci. J. Informatics*, vol. 10, no. 2, pp. 113–120, 2023, doi: 10.15294/sji.v10i2.42818.
- [7] A. M. Wahid, K. A. N. Turino, K. A. Nugroho, T. S. Maharani, D. Darmono, and F. S. Utomo, “Optimasi Logistic Regression dan Random Forest untuk Deteksi Berita Hoax Berbasis TF-IDF,” *J. Pendidik. Dan Teknol. Indones*, vol. 4, no. 8, pp. 1–12, 2024.
- [8] A. Fauziah and J. Hernadi, “Klasifikasi Data Tak Seimbang Menggunakan Algoritma Random Forest dengan SMOTE dan SMOTE-ENN (Studi Kasus pada Data Stunting),” *J. Ris. Sist. dan Teknol. Inf.*, vol. 3, no. 2, pp. 112–121, 2025.
- [9] A. Subandono and D. Ariatmanto, “Optimizing Feature Selection in Sentiment Analysis of Bank Saqu: A Comparative Study of SVM and Random Forest using Information Gain and Chi-Square,” *Sist. J. Sist. Inf.*, vol. 14, no. 3, pp. 1205–1219, 2025, doi: 10.32520/stmsi.v14i3.5106.
- [10] I. R. Hendrawan, E. Utami, and A. D. Hartanto, “Comparison of naïve bayes algorithm and XGBoost on local product review text classification,” *Edumatic J. Pendidik. Inform.*, vol. 6, no. 1, pp. 143–149, 2022.
- [11] R. Ridwan, D. Riswandi, and Y. S. Rachmanda, “Deteksi Dini Kecurangan Pada Program Koperasi Merah Putih: Analisis Kelembagaan Desa Dan Implementasi Model Sipak,” *J. Ekon. PERJUANGAN*, vol. 7, no. 1, pp. 41–53, 2025.
- [12] I. Machali, *Metode Penelitian Kuantitatif*. Yogyakarta: Fakultas Ilmu Tarbiyah dan Keguruan Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta, 2021.
- [13] U. A. Susilo, B. N. Sari, and A. Rizal, “Analisis Sentimen Opini Masyarakat Twitter Terhadap Kasus Diabetes Di Indonesia Menggunakan Algoritma Naïve Bayes Classifier,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, 2025.
- [14] F. Satria, W. H. N. Putra, and B. T. Hanggara, “Analisis Sentimen Mengenai Kepuasan Pengguna terhadap Pelayanan Transportasi Minibus PT Mahkota Tria Wisata dengan menggunakan Metode Support Vector Machine,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 8, pp. 3577–3583, 2023.
- [15] L. Efrizoni, S. Defit, M. Tajuddin, and A. Anggrawan, “Komparasi ekstraksi fitur dalam klasifikasi teks multilabel menggunakan algoritma machine learning,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 3, pp. 653–666, 2022.
- [16] S. Purwaningrum, A. Susanto, and A. S. Prabowo, “Kombinasi Algoritma TF-IDF dan Weighted Dice Similarity untuk Pengukuran Kemiripan Judul Tugas Akhir,” *Infotekmesin*, vol. 16, no. 2, pp. 328–337, 2025.
- [17] S. E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, “Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit,” *J. Math. Theory Appl.*, pp. 21–26, 2022.
- [18] L. G. A. Putri, S. A. Wicaksono, and B. Rahayudi, “Analisis Klasifikasi Spam Email Menggunakan Metode Extreme Gradient Boosting (XGBoost),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 2, 2025.
- [19] J. Mushava and M. Murray, “Flexible loss functions for binary classification in gradient-boosted decision trees: An application to credit scoring,” *Expert Syst. Appl.*, vol. 238, pp. 121876, 2024, doi: <https://doi.org/10.1016/j.eswa.2023.121876>.
- [20] F. Putrawansyah, “Penerapan metode Support Vector Machine terhadap klasifikasi jenis jambu biji,” *JIKO (Jurnal Inform. dan Komputer)*, vol. 8, no. 1, pp. 193–204, 2024.
- [21] M. Affandes et al., “Penerapan Metode Support Vector Machine (SVM) Menggunakan Kernel Radial Basis Function (RBF) Pada Klasifikasi Tweet,” *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 12, no. 2, pp. 189–197, 2015.
- [22] P. A. Octaviani, Y. Wilandari, and D. Ispriyanti, “Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang,” *J. Gaussian*, vol. 3, no. 4, pp. 811–820, 2014.
- [23] A. Desiani et al., “Penerapan Metode Support Vector Machine Dalam Klasifikasi Bunga Iris,” *IJAI (Indonesian J. Appl. Informatics)*, vol. 7, no. 1, pp. 12–18, 2023.
- [24] E. M. Pusung and I. N. Dewi, “Optimasi RoBERTa dengan Hyperparameter Tuning untuk

- Deteksi Emosi berbasis Teks,” *J. Nas. Teknol. dan Sist. Inf.*, vol. 10, no. 3, pp. 240–248, Feb 2025, doi: 10.25077/TEKNOSI.v10i3.2024.240-248.
- [25] M. Fajri and A. Primajaya, “Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search,” *J. Appl. Informatics Comput.*, vol. 7, no. 1, pp. 10–15, 2023, doi: 10.30871/jaic.v7i1.5004.
- [26] W. Nugraha and A. Sasongko, “Hyperparameter tuning pada algoritma klasifikasi dengan grid search,” *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 391–401, 2022.
- [27] K. T. Putra, M. A. Hariyadi, and C. Crysdiyan, “Perbandingan Feature Extraction TF-IDF dan BOW untuk Analisis Sentimen Berbasis SVM,” *J. Cahaya MAndalika*, pp. 1449–1463, 2023.
- [28] Y. A. Prasetyo, E. Utami, and A. Yaqin, “Pengaruh Komposisi Split Data Terhadap Performa Akurasi Analisis Sentimen Algoritma Naïve Bayes dan SVM,” *J. homepage J. Electr. Eng. Comput.*, vol. 6, no. 2, 2024.
- [29] H. Zhao, Y. Wang, X. Li, and P. Guo, “Prediction of Maximum Tunnel Uplift Caused by Overlying Excavation Using XGBoost Algorithm with Bayesian Optimization,” 2023.
- [30] A. Kahraman, “Machine learning techniques for improved prediction of cardiovascular diseases using integrated healthcare data,” pp. 1–14, 2025, doi: 10.3389/frai.2025.1694450.
- [31] Q. A. Hidayaturrohman, “A Comparative Analysis of Hyper-Parameter Optimization Methods for Predicting Heart Failure Outcomes,” pp. 1–17, 2025.
- [32] I. N. Adi, M. Putra, and C. Pramarta, “Optimasi Hyperparameter Algoritma Support Vector Machine dalam Klasifikasi Penyakit  $\beta$ -Thalassemia,” vol. 3, pp. 283–294, 2025.
- [33] N. Prafitri, F. A. Saputra, T. M. Setiyawati, and G. R. M. Ibrahim, “Analisis Big Data Sentimen Publik Terhadap Implementasi Kebijakan Makan Bergizi Gratis di Indonesia,” pp. 153–167, 2025.