

Comparison of Naive Bayes, SVM, and Logistic Regression for Sentiment Analysis of the Makan Bergizi Gratis Program

Andika*¹, Julio Cesar Alessandro², Adjie perkasa Tarigan³, Zaqi Kurniawan⁴

Information System^{1*, 2,3,4}
Universitas Budi Luhur, Jakarta, Indonesia^{1*, 2,3,4}
<https://www.budiluhur.ac.id/>^{1*, 2,3,4}

2412500999@student.budiluhur.ac.id¹, 2412501609@student.budiluhur.ac.id², 2412500619@student.budiluhur.ac.id³,
zaqi.kurniawan@budiluhur.ac.id⁴

Abstract. The Makan Bergizi Gratis Program (MBG) became one of the widely discussed public issues on platform X and generated diverse responses from users. These responses included supportive, critical, and neutral opinions, making sentiment analysis relevant for understanding public opinion toward the program. This study compares three text classification algorithms, namely Naive Bayes, LinearSVC based Support Vector Machine, and Logistic Regression, to analyze the sentiment of tweets concerning the Makan Bergizi Gratis Program. Data were collected from platform X on 19 June 2026 using tweet crawling and were subsequently filtered and manually annotated. The final dataset consisted of 1,061 Indonesian language tweets classified into negative, neutral, and positive sentiment. Two annotators independently assigned sentiment labels, followed by discussion to resolve disagreements. The inter annotator agreement reached 94.62%, with a Cohen's Kappa coefficient of 0.9097. Text preprocessing consisted of cleaning, normalization, tokenization, stopword removal, and stemming. TF-IDF features were generated within a pipeline to ensure that feature fitting was performed only on the training portion of each evaluation fold. The data were divided into training and testing sets using an 80:20 stratified split. Hyperparameter optimization was performed for each classifier, followed by evaluation using accuracy, macro precision, macro recall, macro F1 score, confusion matrix, and stratified 5 fold cross validation. The test results show that SVM achieved the highest accuracy of 72.77%, precision of 75.11%, recall of 65.17%, and macro F1 score of 68.28%. Logistic Regression obtained a macro F1 score of 67.05%, while Naive Bayes obtained 63.69%. SVM also achieved the highest average macro F1 score in cross validation at 60.25%, compared with 59.94% for Logistic Regression and 56.53% for Naive Bayes. These results indicate that SVM provided the strongest overall performance for the dataset examined in this study.

Key words: Sentiment Analysis; MBG Program; Naive Bayes; SVM; Logistic Regression

1. INTRODUCTION

Early 2025 marked something uncommon in Indonesian public policy: a government program immediately drew the attention of millions of social media users even before its implementation was fully underway. The Makan Bergizi Gratis Program (MBG) was not merely a question of school children's nutrition; it also served as a mirror reflecting various social, economic, and political tensions that surfaced on platform X. Notably, what flowed through the timeline was not simply enthusiasm or outright rejection. Many posts appeared in far more ambiguous forms, such as news excerpts reshared without comment, laughter difficult to interpret as either genuine or sarcastic, and questions left without firm answers.

This is where sentiment analysis becomes relevant. As a branch of Natural Language Processing, sentiment analysis enables the extraction and classification of opinions from textual data in a systematic and measurable manner. Social media platforms provide a large volume of user generated content that can be examined to identify patterns of public opinion toward particular issues [1][2]. On platform X, however, short informal expressions, contextual language, and ambiguous statements can complicate sentiment classification. Consequently, the interpretation of sentiment cannot always rely on individual words alone. Within the landscape of text classification algorithms, the available options are numerous. Yet three names recur persistently across the sentiment analysis literature: Naive Bayes, Support Vector Machine, and Logistic Regression.



Their continued presence is not arbitrary; each offers distinct advantages. Naive Bayes appeals through its efficiency, training quickly with minimal computational overhead. SVM, on the other hand, is built to perform effectively in high-dimensional feature spaces such as TF-IDF representations, making it well-suited for sparse text data. Logistic Regression is often associated with relatively stable performance and can handle multiclass classification directly, making it relevant for sentiment classification tasks [3][4].

Prior studies have begun to explore this territory. Both Logistic Regression and SVM have been reported to produce adequate classification on social media data with varying characteristics

[5][6]. Comparisons of Naive Bayes, SVM, and Logistic Regression have also been conducted in other sentiment analysis studies, and results consistently indicate that each model's performance can differ substantially depending on the characteristics of the data at hand [7]. Nonetheless, a specific gap remains. A direct comparison of these three algorithms on X platform data concerning MBG, particularly under an imbalanced class distribution, has not been widely explored. That is the motivation behind this study.

Concretely, this study compares the performance of Naive Bayes, SVM, and Logistic Regression in classifying the sentiment of MBG tweets into three categories: positive, negative, and neutral. Beyond simply examining accuracy figures, this study also seeks to understand how class distribution imbalance affects each algorithm, and which evaluation metric is more appropriate for representing model performance under such conditions. The findings are intended to serve as a practical reference for other researchers working with similarly structured datasets.

2. RELATED WORK

2.1. Sentiment Analysis on Social Media

Sentiment analysis is a branch of Natural Language Processing that focuses on identifying, extracting, and classifying opinions embedded in text data [1]. The challenge, however, is that data from platforms such as X is far from homogeneous. Posts tend to be short, multilingual, and filled with abbreviations and terminology that exist primarily within the Twitter ecosystem; at times, a single word can entirely shift the meaning of a sentence. These conditions place heavier demands on the preprocessing stage, because a misstep at this point will leave the model built on top of it impaired from the very start[1][2]. In the domain of public policy, social media data has long served as a kind of opinion thermometer. To be sure, users on platform X do not proportionally represent the entire Indonesian population. Nevertheless, the speed and volume of conversations taking place there lend it a distinct value: not as a representative sample, but as an early indicator of how policies are received by the most vocal segments of the digital public [1]. Various studies on other public

issues have demonstrated that this approach, though not without limitations, is capable of capturing relatively consistent opinion patterns from social media data [2].

2.2. Text Classification Algorithms

Naive Bayes rests on Bayes' Theorem with an assumption its very name concedes as "naive": that each feature in the data is independent of the others. This assumption is rarely fully satisfied in real-world text, yet it curiously does not significantly diminish the algorithm's effectiveness. Training is fast, memory requirements are low, and results across various document classification tasks are often surprisingly competitive, even against far more complex models. This is why it continues to serve frequently as a baseline in sentiment analysis studies [3]. Support Vector Machine operates on a fundamentally different logic, seeking a hyperplane, a decision boundary in high-dimensional space, that maximizes the margin between distinct classes. For text data represented in sparse formats such as TF-IDF, the LinearSVC variant is preferred for its computational efficiency without sacrificing separation quality. Its track record in text classification is substantial; SVM regularly emerges as the preferred option when feature dimensionality is large and the data does not lend itself to clean linear separation [4]. Unlike the previous two algorithms, Logistic Regression approaches the problem from a probabilistic angle: it models the likelihood of a document belonging to a particular class based on the weights of its features. Its implementation is straightforward, its interpretation relatively transparent, and its performance tends to remain stable across varied data conditions. What makes it particularly suitable for the present study is its native capacity to handle multiclass classification directly, without requiring additional engineering to separate the positive, negative, and neutral classes [8].

2.3. Class Imbalance

Class imbalance refers to a condition in which the distribution of samples across classes within a dataset is uneven. This condition can lead models to favor learning patterns from the majority class, producing high accuracy values while performing poorly on minority classes. In social media-based sentiment analysis, uneven class distributions are frequently encountered because user interaction patterns and data characteristics differ across the topics being discussed [9].

Several strategies have been developed to address this problem. Oversampling synthetically adds minority class samples, undersampling reduces the majority class, and class weighting adjusts penalties during training. SMOTE, one of the most widely cited techniques, interpolates new samples between existing minority data points. Each of these approaches carries its own advantages and risks, and the most appropriate choice depends heavily on data characteristics and the intended purpose of the analysis [10]. In this study, no resampling technique was applied.



This was not an oversight but rather a deliberate methodological choice: by allowing the class distribution to remain as is, we could directly observe how each algorithm responds to that imbalance.

3. METHODS

3.1. Data Collection

Data were collected from platform X using Tweet Harvest version 2.6.1. The crawling process was conducted on 19 June 2026 using the query “mbg OR makanan bergizi gratis” with the latest tweet sorting option. A collection limit of 1,080 tweets was specified. The initial crawling process produced 1,080 tweet records.

The collected dataset contained tweet identifiers, upload dates, usernames, tweet text, and interaction information. The cleaning stage first checked for empty tweet records. No tweet was removed because of empty tweet content at this stage, so the dataset remained at 1,080 records. Duplicate tweet content was then removed, resulting in 1,079 unique tweets.

The resulting dataset was retained for the annotation process after checking its relevance to the Makan Bergizi Gratis Program and its Indonesian language content. No specific user account or account category was intentionally excluded during crawling. After text preprocessing, 18 records produced empty cleaned text and were excluded from model development. Therefore, 1,061 tweets were used in the final machine learning experiments.

3.2. Sentiment Labeling

In this study, Sentiment labeling was performed manually by two annotators. Each tweet was independently examined and assigned to one of three sentiment categories: positive, negative, or neutral. The positive category represented support, appreciation, approval, or favorable evaluation of the Makan Bergizi Gratis Program. The negative category included criticism, rejection, complaints, concerns, and statements indicating disagreement with the program or its implementation. The neutral category was assigned to informative statements, factual content, ambiguous expressions, or tweets without a clear evaluative position.

The annotation process considered the overall context of each tweet rather than relying solely on individual sentiment words. This approach was important because social media text may contain informal expressions, abbreviations, irony, and sarcasm whose intended meaning cannot always be determined from individual words [1][2]. Manual annotation also allows sentiment labels to be assigned according to the contextual meaning of an utterance rather than through direct word matching [11].

For sarcastic tweets, the annotators considered the intended meaning expressed in the overall context. When sarcasm conveyed criticism, dissatisfaction, rejection, or a

negative evaluation of the Makan Bergizi Gratis Program, the tweet was assigned to the negative category. Criticism concerning food quality, program implementation, distribution, management, or other shortcomings was also classified as negative. Therefore, the occurrence of a positive word alone did not automatically result in a positive label.

The two annotators first labeled the tweets independently. Their labels were then compared to identify disagreements. A total of 58 disagreements were identified among 1,079 annotated tweets. The annotators discussed the disagreement cases and established a final agreed label for each case. The resulting final labels were stored in the Label column and used as the target variable in the subsequent machine learning experiments.

Inter annotator agreement was measured using Cohen's Kappa before the disagreement resolution process. The observed agreement was 94.62%, while the Cohen's Kappa coefficient was 0.9097. The use of two annotators and agreement measurement provides an additional assessment of consistency in the manual annotation process [11].

The final labeling results showed that, from the 1,079 annotated tweets, 568 tweets were categorized as negative sentiment, 355 as neutral sentiment, and 156 as positive sentiment. Negative sentiment represented the largest class with 52.64% of the dataset, followed by neutral sentiment at 32.90% and positive sentiment at 14.46%. This distribution indicates that negative sentiment was the most frequently observed category in the annotated dataset, while positive sentiment represented the smallest class. The resulting class imbalance was subsequently considered during model evaluation by emphasizing macro based performance metrics.

TABLE 1. Sentiment Class Distribution of Annotated Dataset

Sentiment Class	Count	Percentage
Negative	568	52,54%
Neutral	355	32,90%
Positive	156	14,46%
Total	1.079	100%

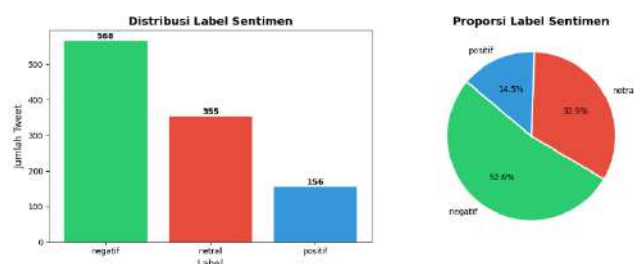


Fig. 1. Sentiment label distribution of the MBG tweet dataset

3.3. Text Preprocessing

Preprocessing was carried out in five sequential stages. The first was cleaning: text was converted to lowercase,



followed by the removal of URLs, mentions, hashtags, RT symbols, HTML entities, emojis, numbers, and punctuation. Characters repeated more than twice were also normalized, so that "wahhhhh," for instance, was reduced to "wahh." The second stage was normalization using a manually constructed slang dictionary comprising more than 80 pairs of non-standard words and their standard equivalents, including common Twitter abbreviations and contextual terms specific to MBG discussions. Emojis were removed to maintain a consistent text representation during normalization, tokenization, and stemming, as the classification pipeline in this study was designed around normalized textual tokens rather than emoji specific features.

Tokenization was the third stage, using the NLTK library to split text into sequences of tokens. Next, stopwords were removed using a combined list sourced from Sastrawi, NLTK, and a custom list specifically designed to capture interjections, particles, and neutral words that carry no sentiment weight in the Twitter context. Finally, stemming was performed using Sastrawi to reduce each word to its base form.

After the preprocessing stages, 18 of the 1,079 annotated tweets produced empty or non informative text representations. Some tweets contained only terms or symbols that were removed during preprocessing. For example, a tweet containing only "MBG" could lose its remaining content after the term was removed during preprocessing, leaving no meaningful textual information for sentiment classification. These records were excluded from model development because they no longer contained sufficient textual information to generate meaningful features. The removal was therefore part of the data filtering process rather than a technical error. After this filtering stage, 1,061 tweets remained for the machine learning experiments.

3.4. Feature Extraction

Once the text was cleaned, each tweet was converted into a numerical vector using TF-IDF (Term Frequency-Inverse Document Frequency), a weighting technique that assigns higher values to words appearing frequently within a specific document but rarely across the broader corpus, thereby giving greater weight to terms that are genuinely distinctive and informative [12]. This method is widely applied in text classification for its capacity to generate effective numerical representations compatible with various machine learning algorithms [13]. The TF-IDF weighting formula is defined as [12]:

$$TF-IDF(d, t) = TF(d, t) \cdot \log(N/df(t)) \quad (1)$$

where N denotes the total number of documents and $df(t)$ denotes the number of documents containing term t . This study used a combination of unigrams and bigrams with `ngram_range = (1,2)`, a maximum feature count of 5,000, and `min_df = 2`. Sublinear TF was also applied to reduce the influence of terms with very high occurrence

frequencies. These configurations were applied consistently across the three classification models.

To prevent information leakage, the TF-IDF vectorizer was implemented within a Pipeline together with the classification model. During stratified 5 fold cross validation, the TF-IDF vectorizer was fitted only on the training portion of each fold and then used to transform the corresponding validation portion. For the final evaluation, TF-IDF was fitted using the training data and subsequently applied to the independent test data. Therefore, vocabulary construction and inverse document frequency statistics from the validation and test data were not used during model training.

3.5. Data Splitting and Evaluation

The final dataset consisted of 1,061 tweets and was divided into training and testing sets using an 80:20 stratified split. This resulted in 848 training samples and 213 testing samples. Stratified sampling was applied to preserve the relative representation of the three sentiment classes in both subsets.

The training data were used for hyperparameter optimization and stratified 5 fold cross validation, while the test data were kept separate and used only for final evaluation. Accuracy, macro precision, macro recall, and macro F1 score were used as evaluation metrics. Macro F1 score remained the primary metric because the dataset contained an uneven class distribution, making accuracy alone insufficient for evaluating performance across all sentiment classes [9].

In addition, the confusion matrix of each model was analyzed to examine error patterns at a more granular level, identifying where each model failed and whether the failure patterns were consistent across the three algorithms.

3.6. Model Implementation

All three models were implemented using Scikit learn in a Python environment. Naive Bayes used the MultinomialNB class, SVM was implemented using LinearSVC, and Logistic Regression was configured using the lbfgs solver with a maximum of 1,000 iterations. These models were selected because they are established approaches for text classification and sentiment analysis [3][4][7].

Hyperparameter optimization was performed using GridSearchCV on the training dataset. Stratified 5 fold cross validation with macro F1 score as the optimization criterion was used to select the best configuration for each model. For Naive Bayes, the alpha parameter was optimized, with 0.1 selected as the best value. For SVM, the C parameter was optimized, with 0.5 selected as the best value. For Logistic Regression, the C parameter was optimized, with 5.0 selected as the best value. The optimized configurations were subsequently used for the final evaluation on the independent test set [7]



4. RESULTS AND DISCUSSIONS

4.1. Exploratory Data Analysis

The dataset used in this study consists of 1,061 Indonesian language tweets related to the Makan Bergizi Gratis Program. After the preprocessing stage, the resulting class distribution remained dominated by the negative class, although all three classes were represented in different proportions. Negative sentiment accounted for the largest share with 566 tweets (53.35%), followed by neutral sentiment with 340 tweets (32.05%), and positive sentiment with 155 tweets (14.61%). The visualization of this distribution is shown in Figure 1.

This distribution pattern is important to consider before interpreting the classification results. The dominance of negative sentiment indicates that the dataset contains substantially more critical or unfavorable responses than explicitly positive responses. The positive class, which represents the smallest proportion of the dataset, is therefore more vulnerable to being overlooked by a classifier. This imbalance makes macro based evaluation important because performance should be considered across all sentiment classes rather than through accuracy alone [9].

The word cloud per class (Figure 2) provides an initial overview of the vocabulary associated with each sentiment category. Among positive tweets, the most prominent words were "dukung" and "jalan", followed by "manfaat", "badan", and "penuh". The prominence of "dukung" reflects explicit expressions of support toward the program, while the occurrence of "jalan" may relate to expectations regarding the continuation or implementation of the program. In the negative class, "korupsi" appeared as the most prominent word, followed by "orang" and "henti". The frequent occurrence of "korupsi" suggests that concerns about governance and implementation were prominent within negative discussions of MBG. The neutral class showed a broader vocabulary, including words such as "layan", "badan", "satu", and "susu", which is consistent with a more informative rather than strongly evaluative pattern.



Fig. 2. Word cloud per sentiment class (positive, negative, neutral)

The bar chart of the 15 most frequent words per class (Figure 3) provides a more detailed view of the dominant vocabulary within each sentiment category. In the positive class, "dukung", "jalan", and "manfaat" occupy the highest positions, followed by words such as "penuh", "laksana", and "lanjut". The frequent occurrence of "dukung" indicates explicit expressions of support toward the MBG Program, while "manfaat" reflects discussion of the perceived benefits of the program. In the negative class,

"korupsi" appears as the most frequent word with a substantial gap from the following terms, particularly "orang" and "henti". This pattern indicates that governance related criticism and calls for change were prominent themes within the negative class. In the neutral class, "my", "kasih", "banget", "satu", and "layan" were among the most frequent terms. The neutral vocabulary is more varied and does not show a single dominant topic comparable to "korupsi" in the negative class, which is consistent with the broader and less explicitly evaluative nature of neutral tweets.

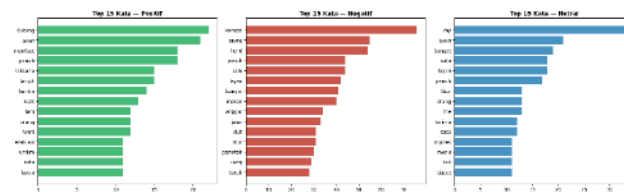


Fig. 3. Fifteen most frequent words per sentiment class

4.2. Model Evaluation Results

Table 2 summarizes the evaluation results of all three models on the 213-sample test set. The figures presented there need to be read together rather than in isolation, as each metric captures a different aspect of performance.

TABLE 2. Model Performance Evaluation Results on Test Data

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	68,54%	74,04%	60,94%	63,69%
SVM	72,77%	75,11%	65,17%	68,28%
(LinearSVC)				
Logistic Regression	71,36%	72,69%	64,29%	67,05%

Table 2 summarizes the evaluation results of all three models on the 213 sample test set. The results show that SVM achieved the highest performance across all four evaluation metrics. SVM recorded an accuracy of 72.77%, macro precision of 75.11%, macro recall of 65.17%, and macro F1 score of 68.28%.

Naive Bayes achieved an accuracy of 68.54%, with a macro precision of 74.04%, macro recall of 60.94%, and macro F1 score of 63.69%. Its relatively high precision indicates that the predictions assigned by the model were comparatively precise. However, its lower recall shows that the model still failed to identify a considerable number of samples from some sentiment classes.

SVM produced the highest macro precision, macro recall, and macro F1 score among the three models. Its macro F1 score of 68.28% indicates that the model provided the most balanced classification performance across the positive, neutral, and negative categories in the test set. Logistic Regression produced results close to SVM, with an accuracy of 71.36% and macro F1 score of 67.05%.

When macro F1 score is considered as the primary comparison metric, SVM achieved the highest result at 68.28%, followed by Logistic Regression at 67.05% and



Naive Bayes at 63.69%. The comparison of all four evaluation metrics is presented in Figure 4.

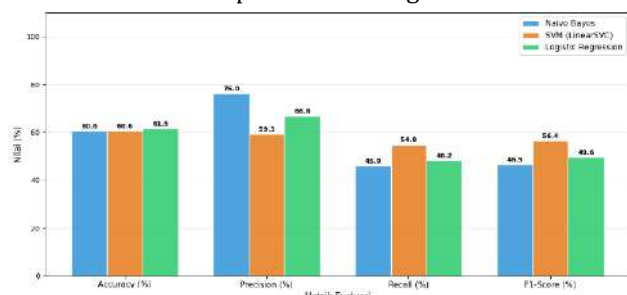


Fig. 4. Model performance comparison across four evaluation metrics

4.3. Confusion Matrix Analysis

The confusion matrix (Figure 5) provides a more detailed view of the prediction patterns produced by each model. The matrices show that all three classifiers recognized the negative class more effectively than the neutral and positive classes, while the main differences appeared in their ability to identify the minority sentiment categories. For Naive Bayes, 104 of 114 negative samples were correctly classified. The model correctly identified 25 of 68 neutral samples and 17 of 31 positive samples. The largest classification error occurred in the neutral class, where 42 neutral samples were predicted as negative. In the positive class, 13 samples were also classified as negative. This pattern indicates that Naive Bayes was relatively effective at recognizing the majority negative class but had greater difficulty identifying neutral and positive tweets.

SVM correctly classified 102 of 114 negative samples, 37 of 68 neutral samples, and 16 of 31 positive samples. The model correctly identified more neutral samples than Naive Bayes, increasing the number of correct neutral predictions from 25 to 37. However, 28 neutral samples were still classified as negative. In the positive class, 16 samples were correctly identified, while 13 were classified as negative. This distribution reflects a more balanced classification pattern across the three sentiment categories and is consistent with the highest macro F1 score obtained by SVM.

Logistic Regression correctly classified 99 of 114 negative samples, 37 of 68 neutral samples, and 16 of 31 positive samples. Its predictions for the neutral and positive classes were identical to those of SVM in this test set. The main difference appeared in the negative class, where Logistic Regression correctly classified 99 samples compared with 102 samples for SVM. This difference contributed to the slightly lower overall performance of Logistic Regression.

Overall, the confusion matrices show that all three models experienced greater difficulty distinguishing neutral tweets from negative tweets. The majority negative class was recognized most effectively, while neutral and positive tweets produced more classification errors. This

pattern is consistent with the distribution of the test data, where negative sentiment represented the largest class.

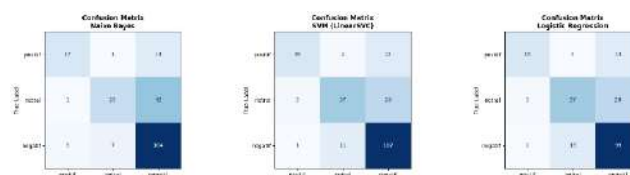


Fig. 5. Confusion matrix of Naive Bayes, SVM, and Logistic Regression

4.4. Cross-Validation Results

Table 3 presents the results of stratified 5 fold cross validation using macro F1 score. The cross validation process was performed on the training data as part of the hyperparameter optimization procedure.

TABLE 3. 5-Fold Cross-Validation Results (Macro F1)

Model	Average F1	Std. Deviasi
Naive Bayes	56,53%	3,80%
SVM (LinearSVC)	60,25%	2,76%
Logistic Regression	59,94%	3,00%

SVM achieved the highest average macro F1 score at 60.25%, followed closely by Logistic Regression at 59.94%. Naive Bayes obtained the lowest average at 56.53%. SVM also produced the smallest standard deviation at 2.76%, compared with 3.00% for Logistic Regression and 3.80% for Naive Bayes. This result indicates that SVM showed the smallest variation in macro F1 score across the five folds.

The per fold results shown in Figure 6 provide additional information about the variation across the training folds. Naive Bayes obtained macro F1 scores of 51.86%, 59.68%, 54.34%, 54.61%, and 62.17% across folds 1 to 5. SVM obtained 56.28%, 60.87%, 62.10%, 58.05%, and 63.96%, while Logistic Regression obtained 55.00%, 59.85%, 62.44%, 58.86%, and 63.56%.

The results show that the performance of all three models varied across folds. SVM remained competitive across all folds and achieved the highest score in folds 1, 2, 4, and 5. Logistic Regression obtained the highest score in fold 3. The variation across folds indicates that the classification performance changed according to the composition of the training and validation subsets.

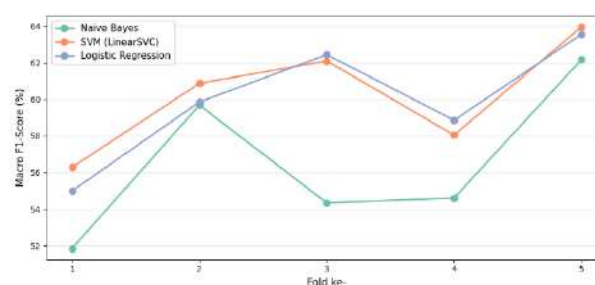


Fig. 6. Macro F1-score comparison per fold in 5-fold cross-validation



4.5. Discussion

Several findings from the evaluation stages warrant closer examination rather than a surface level comparison of numerical results. The final test results show that the three models produced different performance profiles across accuracy, precision, recall, and macro F1 score. SVM achieved the highest values across all four metrics, with an accuracy of 72.77%, precision of 75.11%, recall of 65.17%, and macro F1 score of 68.28%. Logistic Regression followed with an accuracy of 71.36% and macro F1 score of 67.05%, while Naive Bayes obtained an accuracy of 68.54% and macro F1 score of 63.69%. The difference in performance becomes more meaningful when the uneven class distribution is considered. The negative class accounted for 53.35% of the modeling dataset, whereas the positive class represented only 14.61%. Under this condition, macro F1 score provides a more informative basis for comparison because it assigns equal importance to each sentiment category, including the minority class[9].

Naive Bayes exhibited a distinct performance pattern in this study. It achieved a macro precision of 74.04%, which was lower than SVM at 75.11% but higher than Logistic Regression at 72.69%. However, its macro recall was only 60.94% and its macro F1 score reached 63.69%, both lower than the corresponding values of the other two models. This pattern indicates that Naive Bayes produced relatively precise predictions but had greater difficulty identifying samples consistently across all sentiment categories. The confusion matrix also showed that many neutral and positive tweets were classified as negative. Therefore, in the present dataset, Naive Bayes was less effective at maintaining balanced recognition across the three sentiment classes. A similar dependence of Naive Bayes performance on dataset characteristics and textual representation has been reported in previous sentiment classification research [14].

A different pattern was observed for SVM. The model achieved the strongest overall performance on the independent test set, with the highest accuracy, precision, recall, and macro F1 score among the three classifiers. Its macro F1 score reached 68.28%, which was 1.23 percentage points higher than Logistic Regression and 4.59 percentage points higher than Naive Bayes. SVM also achieved the highest average macro F1 score during stratified 5 fold cross validation, reaching 60.25%, compared with 59.94% for Logistic Regression and 56.53% for Naive Bayes. Its standard deviation of 2.76% was also the smallest among the three models. These results indicate that SVM provided the strongest and relatively stable performance under the experimental conditions used in this study. The result is also consistent with previous research on sentiment analysis of the Makan Bergizi Gratis Program, where SVM demonstrated competitive classification performance [6]. The effectiveness of SVM is also relevant to the use of TF-IDF features because SVM is well suited to classification

problems involving high dimensional text representations [4].

Logistic Regression produced results that were close to those of SVM. Its test accuracy reached 71.36%, while its macro precision, macro recall, and macro F1 score were 72.69%, 64.29%, and 67.05%, respectively. Its average macro F1 score in 5 fold cross validation was 59.94%, only 0.31 percentage points below the SVM result. This relatively small difference indicates that Logistic Regression remained a competitive classifier for the dataset. However, SVM maintained a consistent advantage across both the independent test evaluation and cross validation. The result suggests that Logistic Regression can perform effectively for this task, although its performance was slightly lower than that of SVM under the feature representation and optimized parameters used in this research.

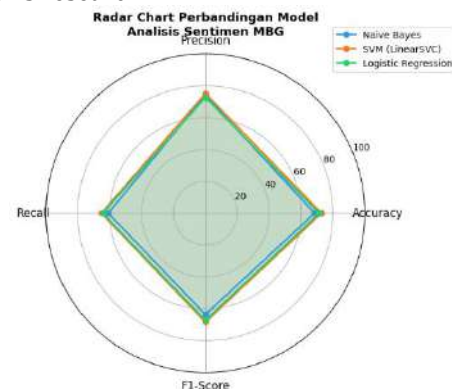


Fig. 7. Radar chart: comprehensive comparison of the three models

The differences among the three models become clearer in the radar chart presented in Figure 7. SVM forms the widest performance profile because it records the highest value on each of the four evaluation metrics. Logistic Regression follows closely, particularly on precision, recall, and F1 score, while Naive Bayes shows a more pronounced difference between its precision and recall. The radar chart therefore supports the numerical comparison presented in Table 2, while also illustrating the relative balance of the models across the evaluated metrics. These findings reinforce the importance of considering multiple evaluation measures rather than relying on accuracy alone, particularly when the class distribution is uneven [9]. The results should not be interpreted as evidence that SVM is universally superior to Naive Bayes or Logistic Regression. The relative performance of a classifier can depend on dataset characteristics, class distribution, preprocessing procedures, feature representations, and model configurations [15]. In the present study, SVM produced the strongest overall results after hyperparameter optimization and evaluation under the specific dataset and experimental design used. The conclusion that SVM was the best performing model therefore applies specifically to the MBG dataset examined in this research



rather than to sentiment classification tasks in general [15].

5. CONCLUSION

This study compared three models, namely Naive Bayes, Support Vector Machine, and Logistic Regression, for three class sentiment classification of Indonesian language tweets concerning the Makan Bergizi Gratis Program. The final modeling dataset consisted of 1,061 tweets. The sentiment annotation process involved two annotators and achieved an observed agreement of 94.62% with a Cohen's Kappa coefficient of 0.9097. Disagreement cases were subsequently discussed between the annotators to establish the final labels used in the classification experiments.

The experimental results showed that SVM achieved the strongest overall performance on the independent test set. SVM obtained an accuracy of 72.77%, precision of 75.11%, recall of 65.17%, and macro F1 score of 68.28%. Logistic Regression achieved an accuracy of 71.36% and macro F1 score of 67.05%, while Naive Bayes achieved an accuracy of 68.54% and macro F1 score of 63.69%. The stratified 5 fold cross validation produced a similar ranking, with SVM obtaining the highest average macro F1 score of 60.25%, followed by Logistic Regression at 59.94% and Naive Bayes at 56.53%.

Hyperparameter optimization also produced distinct model configurations, with $\alpha = 0.1$ for Naive Bayes, $C = 0.5$ for SVM, and $C = 5.0$ for Logistic Regression. These optimized configurations were evaluated using an independent test set after TF-IDF processing was integrated into the classification pipeline to prevent information leakage.

Based on the experimental conditions used in this study, SVM can be considered the strongest performing model among the three evaluated classifiers for the MBG dataset. This finding is specific to the dataset, preprocessing procedure, TF-IDF configuration, class distribution, and optimized hyperparameters applied in this research. It should not be interpreted as evidence that SVM is universally superior to Naive Bayes or Logistic Regression.

Future research may investigate class weighting, resampling strategies, and transformer based models to improve the recognition of minority sentiment classes, particularly positive and neutral tweets.

REFERENCES

- [1] M. Rodríguez-Ibáñez, A. Casáñez-Ventura, F. Castejón-Mateos, and P. M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Syst. Appl.*, vol. 223, no. March, 2023, doi: 10.1016/j.eswa.2023.119862.
- [2] B. Setiawan, "A Review of Sentiment Analysis Applications in Indonesia Between 2023-2024," *J. Inf. Eng. Educ. Technol.*, vol. 8, no. 2, pp. 71–83, 2025, doi: 10.26740/jieet.v8n2.p71-83.
- [3] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "Survey on Text Classification Algorithms: From Text to Predictions," *Inf.*, vol. 13, no. 2, pp. 1–39, 2022, doi: 10.3390/info13020083.
- [4] S. N. Khan, S. U. Khan, H. Aznaoui, C. B. Şahin, and Ö. B. Dinler, "Generalization of linear and non-linear support vector machine in multiple fields: a review," *Comput. Sci. Inf. Technol.*, vol. 4, no. 3, pp. 226–239, 2023, doi: 10.11591/csit.v4i3.p226-239.
- [5] M. Dawud, "Klasifikasi sentimen terhadap program makan bergizi gratis menggunakan metode Logistic Regression," pp. 400–412, 2025, doi: <https://doi.org/10.35889/progresif.v22i2.3610>.
- [6] M. Arif Firmansyah, Asep Saeppani, and Irfan Fadil, "Analisis Sentimen Publik Program Makan Bergizi Gratis Menggunakan Support Vector Machine," *JPNM J. Pustaka Nusantara Multidisiplin*, vol. 3, no. 4, pp. 14–20, 2025, doi: 10.59945/jpnm.v3i4.805.
- [7] S. Jatmiko and C. Dometian, "Analisis Sentimen Ulasan Google Play Store: Studi Komparatif Algoritma SVM, Naive Bayes, dan Logistic Regression," *J. FASILKOM*, vol. 15, no. 3, pp. 610–620, Jan. 2026, doi: 10.37859/jf.v15i3.10016.
- [8] S. Pradeepa *et al.*, "HGATT_LR: transforming review text classification with hypergraphs attention layer and logistic regression," *Sci. Rep.*, vol. 14, no. 1, pp. 1–14, 2024, doi: 10.1038/s41598-024-70565-6.
- [9] R. Obiedat *et al.*, "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution," *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.
- [10] H. Hairani, T. Widiyaningtyas, and D. D. Prasetya, "Addressing Class Imbalance of Health Data: a Systematic Literature Review on Modified Synthetic Minority Oversampling Technique (SMOTE) Strategies," *Int. J. Informatics Vis.*, vol. 8, no. 3, pp. 1310–1318, 2024, doi: 10.62527/joiv.8.3.2283.
- [11] I. Amal and Jayanta, "Perbandingan Pelabelan Otomatis Dan Manual Untuk Analisis Sentimen Terhadap Kenaikan Harga BBM Pertamina Pada Twitter Menggunakan Algoritma Support Vector Machine," *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, vol. 4, no. 2, pp. 473–487, 2023, doi: 10.1109/SENAMIKA57008.2023.10338748.
- [12] H. D. Abubakar and M. Umar, "Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec," *SLU J. Sci. Technol.*, vol. 4, no. 1&2, pp. 27–33, 2022, doi: 10.56471/slujst.v4i.266.
- [13] I. Hendrawan Rifky, E. Utami, and A. Hartanto Dwi, "Analisis Perbandingan Metode Tf-Idf dan Word2vec pada Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal di Indonesia," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 11, no. 3, pp. 497–503, 2022, doi: 10.30591/smartcomp.v11i3.3902.
- [14] S. Syafrizal, M. Afdal, and R. Novita, "Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naive Bayes Classifier dan K-Nearest Neighbor," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 10–19, Dec. 2023, doi: 10.57152/malcom.v4i1.983.
- [15] K. Walji, A. Erraissi, A. Zakrani, and M. Banane, "From Review to Practice: A Comparative Study and Decision-Support Framework for Sentiment Classification Models," vol. 16, no. 9, pp. 699–709, 2025, doi: 10.14569/IJACSA.2025.0160967.

