ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Comparative Analysis of Machine Learning Algorithms for Sentiment Classification of Discord App Reviews

Rani Rosita[1], Putri Taqwa Prasetyaningrum[2]

[1,2] Information Systems, Mercu Buana University Yogyakarta, Yogyakarta, Indonesia

Email: [1] 211210014@student.mecubuana-yogya.ac.id, [2] putri@mercubuana-yogya.ac.id,

Received: Nov 4, 2025
Revised: Nov 26, 2025
Accepted: Dec 21, 2025
Published: Dec 31, 2025

Corresponding Author:
Author Name*:
Rani Rosita
Email*:
211210014@student.mecubuana-yogya.ac.id

DOI:
10.63158/journalisi.v7i4.1367

Abstract. The increasing use of digital communication applications such as Discord has generated diverse user opinions expressed through reviews on the Google Play Store. This study aims to analyze user sentiment toward the Discord application using text mining and machine learning techniques. A total of 3,000 reviews were collected through web scraping, pre-processed, labeled using a lexicon-based approach with TextBlob, and balanced using the SMOTE-Tomek method. Sentiment classification was performed into positive, negative, and neutral categories using Decision Tree, Logistic Regression, Support Vector Machine (SVM), and an Ensemble method. The Ensemble model achieved the highest accuracy of 98.67%, followed by Decision Tree (96.50%), SVM (95.83%), and Logistic Regression (90.33%). Limitations of this study include the use of lexicon-based sentiment labeling, machine translation from Indonesian to English, and initial class imbalance. Despite this strong performance, the study has limitations related to lexicon-based labeling, translation of reviews into English, and the presence of a highly imbalanced class distribution in the original dataset. Overall, the findings demonstrate that the Ensemble approach effectively improves sentiment classification accuracy and can support data-driven decision-making in application development.

Keywords: Sentiment Analysis; Ensemble Learning; Machine Learning; Discord Reviews; Google Play Store

1. INTRODUCTION

Technology is developing rapidly in line with increasing life demands and the growing mobility of human labor, with major advances occurring in communication technology, including the Discord application [1]. Discord is a VoIP-based communication platform initially designed for gaming communities and distinguished by its channel-based system, which enables structured interactions similar to online forums [2]. In addition to text and voice communication, Discord supports features such as screen sharing, live streaming, bot integration, and customizable channels, making it widely adopted for various purposes beyond gaming [3].

Despite these advantages, Discord also has limitations, particularly the dependence of voice calls and screen-sharing features on internet stability. These strengths and weaknesses have prompted users to express diverse opinions through reviews on the Google Play Store. Such reviews provide valuable insights into user satisfaction and expectations however, their large volume and unstructured nature make manual analysis inefficient, highlighting the need for automated sentiment analysis techniques.

Although user reviews can provide valuable insights into user satisfaction, pain points, and expectations, extracting actionable information from thousands of raw reviews is difficult without automation. Traditional manual methods are inefficient, prone to bias, and incapable of processing data at scale. Therefore, an automated sentiment analysis approach is necessary to categorize these reviews into positive, negative, and neutral sentiments, allowing for better understanding and response to user feedback.

Sentiment analysis, a branch of text classification, aims to identify and categorize user opinions into sentiment classes such as positive, negative, and neutral [4]. Previous studies have applied various machine learning algorithms, including Naive Bayes and K-Nearest Neighbor, as well as lexicon-based approaches for sentiment labeling. Effective sentiment analysis requires appropriate text preprocessing steps, including case folding, tokenization, filtering, stemming, and stopword removal [5].

Several machine learning algorithms have demonstrated effectiveness in sentiment classification, such as Decision Tree, Support Vector Machine (SVM), Logistic Regression, and Ensemble learning. Decision Tree offers interpretable classification rules [6], SVM performs well on high-dimensional text data by optimizing decision margins [7], Logistic Regression provides a simple probabilistic classification framework [8], and Ensemble learning improves prediction accuracy by combining multiple models through techniques such as bagging and boosting [9]. Prior research has shown that integrating dimensionality reduction methods like Principal Component Analysis (PCA) with machine learning models can enhance classification accuracy and efficiency [10]. Most existing studies on sentiment analysis either focus on English-language datasets or limit their experiments to a single machine learning (ML) algorithm. Very few works have examined sentiment classification of Indonesian-language reviews, particularly on platforms like Discord. Moreover, studies rarely address class imbalance problems by combining SMOTE-Tomek resampling with multiple machine learning models. This gap leaves questions about model generalizability and performance under imbalanced conditions unanswered.

This study aims to conduct a comparative analysis of four machine learning algorithms—Support Vector Machine (SVM), Decision Tree, Logistic Regression, and an Ensemble method—to classify user sentiments expressed in Indonesian-language reviews of the Discord application. The study also integrates a SMOTE-Tomek technique to handle class imbalance, enhancing the robustness and fairness of model evaluation. The findings are expected to support the development of intelligent sentiment classification tools that aid in product evaluation and user experience improvement. [11].

2. METHODS

Research methods is a process or method scientific to obtain data that will be used for purposes research. Learning model development method Machines for sentiment analysis [12]. Learning method machine has proven effective in classify review customers and provide insights that can followed up to guide improvement quality service.

2.1. Study Literature

This study begins with a literature review on sentiment analysis, text preprocessing, feature extraction, and classification methods. User reviews of the Discord application are collected and preprocessed through cleaning, tokenization, stopword removal, and stemming. The data are then labeled and transformed using TF-IDF feature extraction.

To handle class imbalance, the SMOTE-Tomek technique is applied prior to model training. The balanced dataset is used to develop and compare Support Vector Machine, Decision Tree, Logistic Regression, and ensemble models. Model performance is evaluated and analyzed to identify the most effective approach for classifying sentiments in Discord application reviews. Flow diagram methodology of this study as shown in Figure 1.

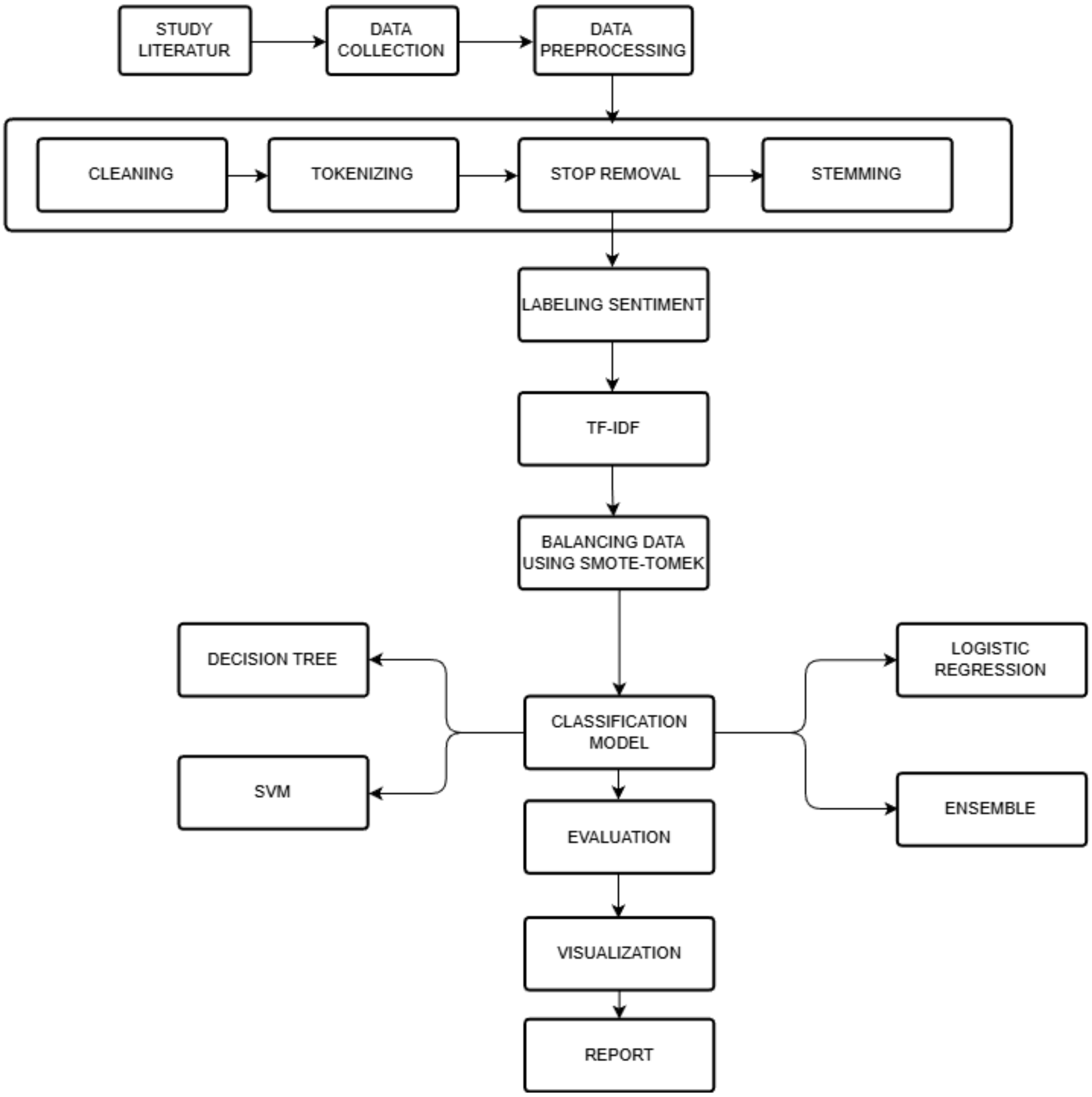


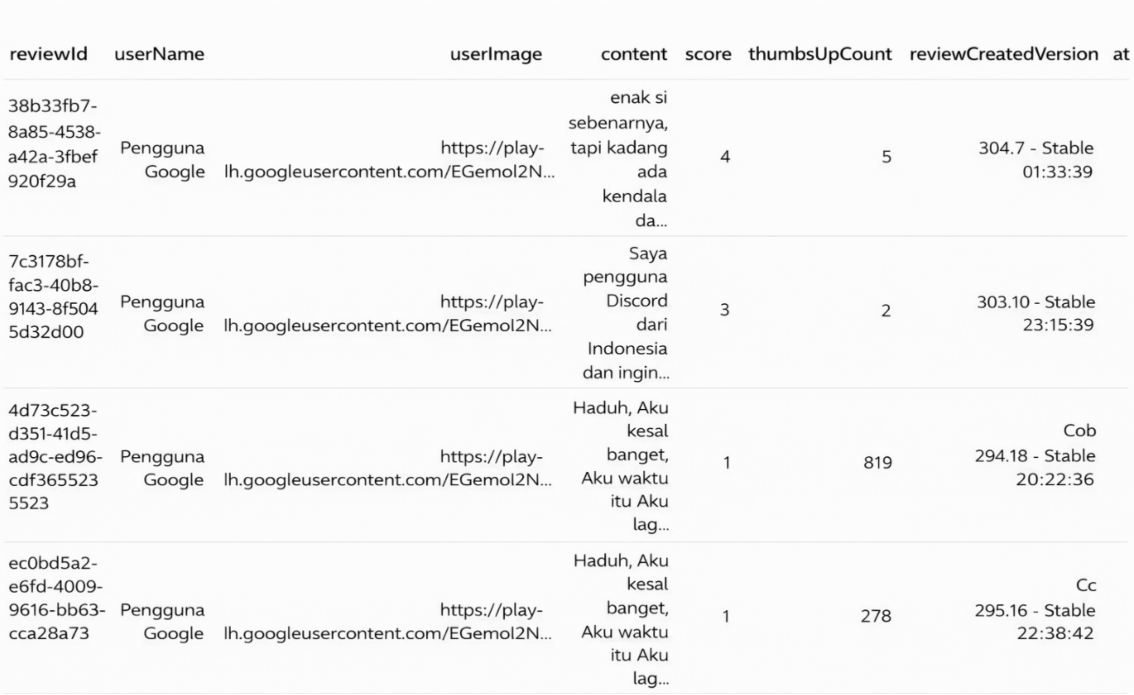
Figure 1. Research Methodology Flowchart

2.2. Library

This research uses various Python libraries with stable version to support data processing and machine learning modeling. Management as well as data cleansing is performed using pandas version 2.1.x and NumPy version 1.26.x which functions for data manipulation and computation numerical. At this stage processing text, research utilizing NLTK version 3.8.x and Sastrawi version 1.0.x for tokenization process, deletion stopwords, and Indonesian stemming. Transformation text become representation numeric done through TfidfVectorizer is available in scikit-learn version 1.3.x. Problem imbalance class handled with SMOTE-Tomek technique using imbalanced-learn library version 0.12.x. Classification model development covers Support Vector Machine, Decision Tree, and Logistic Regression algorithms, which are implemented through module sklearn.svm, sklearn.tree, and sklearn.linear_model on scikit-learn version 1.3.x. For visualization research result using matplotlib version 3.8.x and seaborn version 0.13.x so that evaluation model performance can displayed in a way informative. This entire library ensures the research process walk systematic, consistent, and appropriate standard modern analysis.

2.3. Data Collection

This study followed the research workflow depicted in Figure 1. A total of 3,000 user reviews of the Discord application were collected from the Google Play Store using an automated Python-based web scraping tool. The data collection process was conducted between October and November 2025. To maintain linguistic consistency and minimize translation ambiguity, only reviews written in the Indonesian language were included in the analysis. All data were sourced from publicly accessible platforms, and the scraping process strictly adhered to ethical research guidelines. No personal, sensitive, or identifiable user information was collected, stored, or processed at any stage, thereby ensuring compliance with data privacy and ethical standards. Following data acquisition, the reviews underwent a multi-stage preprocessing pipeline, including text cleaning, tokenization, stopword removal, and stemming using Indonesian-specific NLP tools [13]. The processed texts were then transformed into numerical representations using the TF-IDF (Term Frequency–Inverse Document Frequency) method. Subsequently, sentiment labels were assigned using a lexicon-based approach via TextBlob, following automatic translation of the reviews into English. To address the severe class imbalance inherent in the original dataset, the SMOTE-Tomek resampling technique was applied to balance the sentiment classes. Finally, sentiment classification was performed using four machine learning algorithms: Decision Tree, Support Vector Machine (SVM), Logistic Regression, and an Ensemble model. The results of each classifier were evaluated and compared using multiple performance metrics, and the findings were visualized to support interpretation and analysis.



reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at
38b33fb7-8a85-4538-a42a-3fbef920f29a	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	enak si sebenarnya, tapi kadang ada kendala da...	4	5	304.7 - Stable 01:33:39	
7c3178bf-fac3-40b8-9143-8f5045d32d00	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Saya pengguna Discord dari Indonesia dan ingin...	3	2	303.10 - Stable 23:15:39	
4d73c523-d351-41d5-ad9c-ed96-cdf3655235523	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Haduh, Aku kesal banget, Aku waktu itu Aku lag...	1	819	Cob 294.18 - Stable 20:22:36	
ec0bd5a2-e6fd-4009-9616-bb63-cca28a73	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Haduh, Aku kesal banget, Aku waktu itu Aku lag...	1	278	Cc 295.16 - Stable 22:38:42	

Figure 2. Data Collection

2.4. Data Preprocessing

Data preprocessing is a crucial step in data analysis and machine learning to transform raw text into structured data suitable for sentiment classification [14]. In this study, preprocessing includes text cleaning (removal of emojis, special characters, URLs), case folding, tokenization, stopword removal, and stemming to standardize Indonesian user reviews collected from the Google Play Store.

Sentiment labeling is performed using a single, consistent lexicon-based scheme with TextBlob. Since the reviews are written in Indonesian, automatic translation into English is conducted using the Google Translate API prior to labeling. TextBlob generates polarity scores ranging from −1 to +1, which are categorized as negative (polarity < −0.1), neutral (−0.1 ≤ polarity ≤ 0.1), and positive (polarity > 0.1). This process produces 237 positive, 323 negative, and 2,440 neutral reviews after preprocessing. However, translation-based sentiment labeling may introduce limitations such as loss of contextual nuance and potential lexicon mismatch between Indonesian expressions and English sentiment vocabularies. To illustrate the preprocessing flow, Table 3 presents examples of review transformation from raw text to labeled sentiment.

Table 1. Review Transformation in Preprocessing Stage

Raw Review (ID)	Cleaned Text	Stemmed Text	Sentiment Label
"aplikasinya, bagus tapi sering keluar sendiri"	*aplikasinya bagus tapi sering keluar sendiri*	*aplikasi bagus tapi sering keluar sendiri*	Negative
"suka sekali fitur voicenya!"	*suka sekali fitur voicenya*	*suka kali fitur voice*	Positive
"biasa aja sih, lumayan tetapi banyak bug"	*biasa aja sih lumayan tapi banyak bug*	*biasa saja lumayan tapi banyak bug*	Neutral

2.5. Labeling Sentiment

After the data cleaning process, each review labeled sentiment use approach lexicon-based via the TextBlob library. Sentiment classified into three category:

Positive: Score > 0

Neutral: Score = 0

Negative: Score < 0

2.6. Feature Extraction with TF-IDF

Feature extraction is performed using the Term Frequency–Inverse Document Frequency (TF-IDF) technique to convert preprocessed text into numerical feature vectors suitable for machine learning models [15]. In this study, TF-IDF is implemented using the scikit-learn library with parameters unigram (n-gram = 1), max_df = 0.95, min_df = 2, and L2 normalization to reduce the influence of extremely frequent or rare terms [16]. The TF component reflects the frequency of a term within a review, while the IDF component reduces the weight of terms that frequently appear across many documents, thereby improving feature discriminability [17]. The resulting TF-IDF vectors are then used as input for Decision Tree, Logistic Regression, Support Vector Machine, and Ensemble classifiers to capture sentiment patterns effectively while minimizing noise from non-informative terms.

2.7. Balancing data Using SMOTE-Tomek

To address class imbalance—where neutral reviews substantially outnumbered positive and negative ones—the SMOTE-Tomek technique was applied. Initially, the dataset consisted of 1,094 neutral reviews, 291 positive reviews, and only 28 negative reviews, a distribution that could lead to biased classification results. SMOTE (Synthetic Minority Over-sampling Technique) was first used to generate synthetic samples for minority classes through interpolation between existing samples and their nearest neighbors, increasing class representation [18]. After oversampling, Tomek Links were employed to remove borderline samples, particularly overlapping pairs between neutral–positive and neutral–negative classes, thereby clarifying decision boundaries and reducing noise, as introduced by Tomek (1976). This combined approach produced a balanced dataset with approximately 1,000 samples per sentiment class. While SMOTE improves class balance, it also introduces a risk of overfitting due to synthetic sample generation; therefore, the Tomek cleaning stage plays a crucial role in mitigating this risk by eliminating ambiguous instances. Overall, SMOTE-Tomek enhances both class distribution and data quality, enabling classifiers to learn sentiment patterns more fairly and robustly.

2.8. Algorithm Classification

This study applies four classification algorithms: Support Vector Machine (SVM), Decision Tree, Logistic Regression, and an Ensemble model. All algorithms were implemented using the scikit-learn library with fixed hyperparameters to ensure consistent evaluation. SVM was selected due to its strong capability in handling high-dimensional text representations such as TF-IDF and its ability to achieve high classification accuracy even with limited training data; however, its performance is highly sensitive to kernel selection and parameter tuning [19]. SVM was configured with a linear kernel to handle high-dimensional TF-IDF features efficiently [20]. Logistic Regression served as a baseline classifier and was combined with other models in an Ensemble framework to improve classification stability and accuracy [21]. Decision Tree was implemented to capture non-linear patterns in sentiment data through recursive attribute splitting [22]. The Ensemble model integrates predictions from individual classifiers to obtain more robust sentiment classification results.

Table 2. Hyperparameter Setting of Classification Algorithms

Algortihm	Hyperparameters
SVM	kernel = linear, C = 1.0
Decision Tree	criterion = gini, max_depth = None
Logistic Regression	penalty = l2, C = 1.0, solver = lbfgs, max_iter = 1000
Ensemble	voting = soft, estimators = SVM, Decision Tree, Logistic Regression

2.9. Model Evaluation and Visualization

Before model evaluation, the Discord review dataset was split into training and testing sets using an 80:20 ratio to ensure objective and unbiased assessment. A 5-fold cross-validation scheme was applied during training to improve generalization and reduce overfitting across all classifiers. Model performance was evaluated using accuracy, precision, recall, macro F1-score, and micro F1-score, providing a comprehensive assessment of each algorithm in classifying positive, negative, and neutral sentiments.

Data visualization is used in This research is to clarify findings sentiment analysis through presentation wordcloud, diagram, and chart model performance. Wordcloud displays the most frequent words appear in review Discord users, while chart distribution sentiment show proportion review positive, negative, and neutral. In addition, the visualization results model evaluation, such as comparison accuracy and metrics performance others, giving a clearer picture informative about the effectiveness of each algorithm in do classification. This visual approach helps strengthen interpretation results and make it easier reader in understanding patterns as well as existing trends in the dataset.

3. RESULTS AND DISCUSSION

In this study, the data used in the form of review users collected Discord applications from Google Play Store as basis for doing sentiment analysis. The data collection process is carried out use Language Python programming through web scraping techniques, so that all over reviews that are explicit discuss experience users can collected in a way automatically. After the data is collected, the next stage is the beginning of what was done is data cleansing to remove irrelevant elements like character special, duplication reviews, as well as other noise that can interfere with the analysis process. This cleaning is important to ensure that the data is processed truly clean and decent used for stages pre-processing. next step focus mainly is compare the four method, after through stage preprocessing, steps furthermore is do classification use a number of method learning machines namely Support Vector Machine (SVM), Decision Tree, Logistic Regression, as well as Ensemble method to improve accuracy and stability prediction.

3.1. Support Vector Machine (SVM) Classification Performance

The SVM model is evaluated using 2,400 samples balanced review. This model shows the total accuracy was 94.83%. Precision, recall, and F1-score values for each category sentiment shown in Table 3.

Table 3. Support Vector Machine Test Data Classification

Sentiment Class	Precision	Recall	F1-Score
Negative (-1)	89 %	53 %	67 %
Neutral (0)	95 %	9 8%	97 %
Positive (1)	96 %	92 %	94 %

Performance per sentiment class is presented in Table 3. The negative class obtained a precision of 0.89, recall of 0.53, and F1-score of 0.67, indicating that a considerable portion of negative reviews was correctly identified. The neutral class showed very strong performance, with precision 0.95, recall 0.98, and F1-score 0.97, while the positive class achieved precision 0.96, recall 0.92, and F1-score 0.94, demonstrating reliable classification of positive sentiment.

However, the relatively low recall for the negative class can be attributed to the characteristics of synthetic negative samples generated during SMOTE-Tomek balancing. These synthetic instances tend to lie close to the decision boundary, affecting the maximum-margin optimization of SVM and causing overlap with neutral samples. As shown in the confusion matrix (Figure 4), only 18 negative reviews were correctly classified, while a portion of negative samples was misclassified as neutral. This indicates that although SMOTE improves class balance, it may introduce borderline instances that reduce the separability of the negative class within the SVM margin, particularly when sentiment expressions are lexically similar.

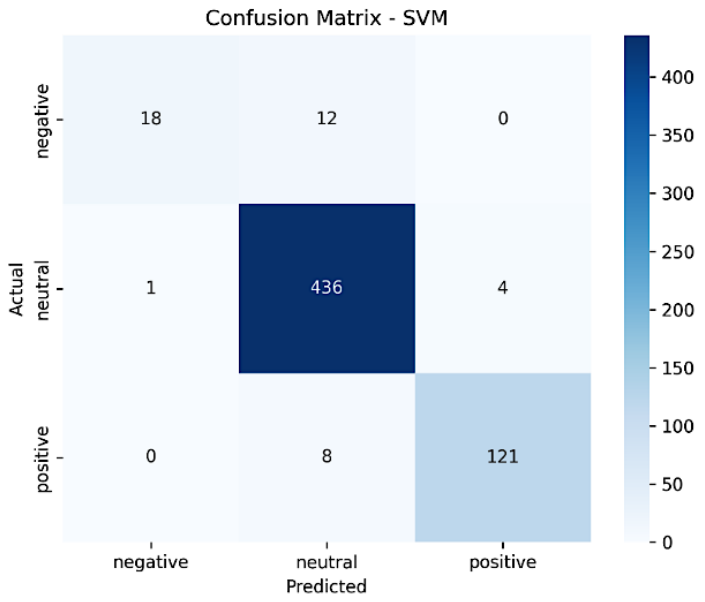


Figure 4. Confusion Matrix SVM

3.2. Logistic Regression Classification Performance

The Logistic Regression model achieved an overall accuracy of 0.9033. As shown in Table 3, the model demonstrates strong performance in classifying neutral and positive sentiments, with F1-scores of 0.94 and 0.86, respectively. However, the performance on the negative class remains low, with an F1-score of only 0.29, indicating limited effectiveness in detecting negative reviews.

Table 4. Logistic Regression Test Data Classification

Sentiment Class	Precision	Remember	F1-Score
Negative (-1)	1%	17%	29%
Neutral (0)	89%	99%	94%
Positive (1)	94%	78%	86%

This performance pattern suggests that Logistic Regression is likely experiencing underfitting, particularly for the negative sentiment class. The use of default L2 regularization may overly constrain the model, preventing it from learning more complex decision boundaries required to distinguish subtle negative expressions [23]. As illustrated by the confusion matrix in Figure 5, only 5 negative reviews and 101 positive reviews were correctly classified, while a substantial portion of these instances was misclassified as neutral. This indicates that the model is highly biased toward the dominant neutral class.

To improve performance, especially for negative sentiment detection, further regularization tuning is recommended, such as adjusting the C parameter to reduce regularization strength or exploring alternative penalty schemes. Such optimization may help Logistic Regression capture richer sentiment patterns and achieve more balanced classification results across all sentiment classes.

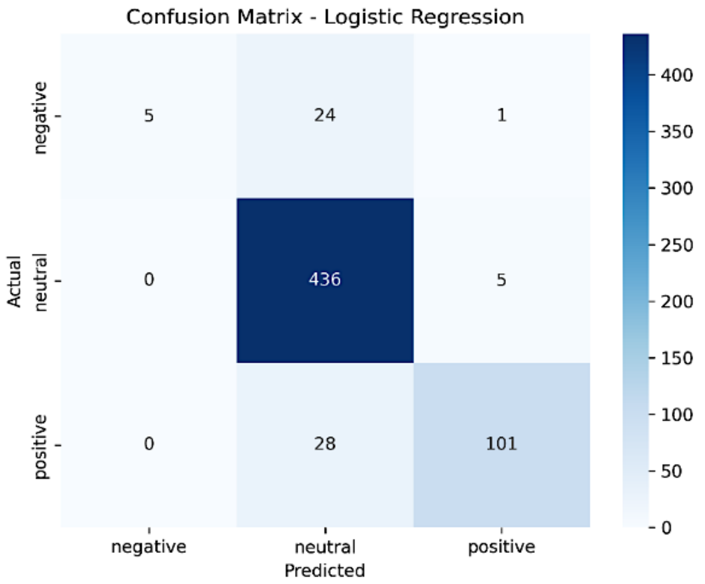


Figure 5. Confusion Matrix Logistic Regression

Matrix The confusion in Figure 5 shows that the Logistic Regression model has accuracy high on sentiment neutral, with 436 identified data in accordance category. However, the model's ability in recognize sentiment negative and positive Still Limited. Only 5 reviews negative and 101 reviews positive detected with right, while part big reviews on two category the still readable as neutral. Findings This show that the model is more sensitive to sentiment neutral compared to two sentiment other.

3.3. Ensemble Classification Performance

The Ensemble model achieved an overall accuracy of 98.67%, with class-wise performance summarized in Table 5. The model shows excellent performance on the neutral and positive classes, achieving F1-scores of 99% and 97%, respectively. This indicates that the Ensemble approach is highly effective in capturing dominant sentiment patterns within the dataset.

Table 5. Classification of Ensemble test data

Sentiment Class	Precision	Remember	F1-Score
Negative (-1)	1%	97%	98%
Neutral (0)	99%	1%	99%
Positive (1)	98%	96%	97%

For the negative class, the model exhibits very high recall (97%) but low precision (1%), indicating that the Ensemble prioritizes detecting almost all negative reviews, although some non-negative samples are misclassified as negative. This behavior arises from the soft voting Ensemble strategy, which combines probability outputs from multiple base classifiers. In this configuration, predictions tend to favor recall for minority or hard-to-separate classes, especially after SMOTE-Tomek balancing, where synthetic negative samples increase class sensitivity but also introduce overlap with neutral samples.

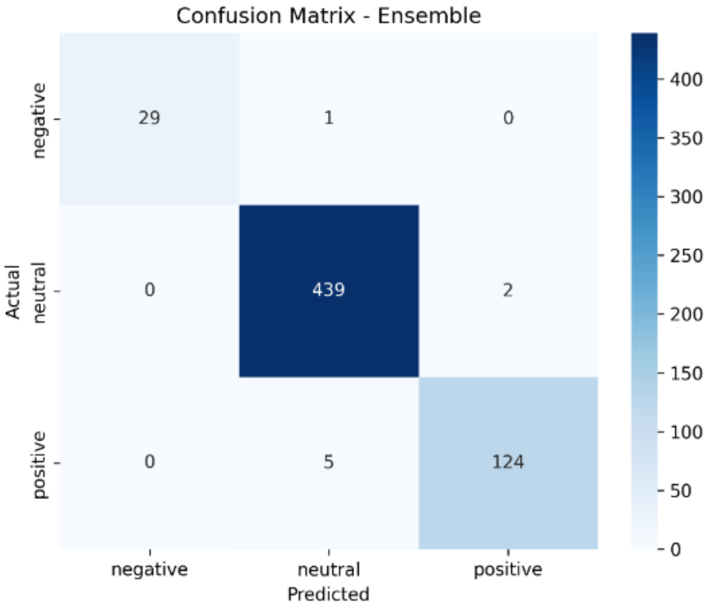


Figure 6. Confusion Matrix Ensemble

As shown in the confusion matrix (Figure 6), most predictions lie along the main diagonal, confirming strong overall classification performance. The neutral class achieved the highest number of correct predictions (439 samples), followed by the positive class (124 samples), while 29 negative samples were correctly identified. Although recall-oriented behavior slightly reduces precision for the negative class, this trade-off is acceptable in sentiment analysis scenarios where identifying negative feedback is prioritized for application improvement. Overall, the soft voting Ensemble demonstrates superior stability and robustness compared to individual models in classifying Discord user reviews.

3.4. Classification Performance Decision Tree

The Decision Tree model achieved an overall accuracy of 96.50%, with class-wise performance summarized in Table 6.

Table 6. Classification of Decision Tree test data

Sentiment Class	Precision	Remember	F1-Score
Negative (-1)	1%	93%	97%
Neutral (0)	98%	97%	98%
Positive (1)	91%	95%	93%

The model demonstrates strong performance in classifying neutral and positive sentiments, with the neutral class achieving 98% precision and 97% recall, and the positive class obtaining 91% precision and 95% recall, resulting in high F1-scores for both classes. In contrast, the negative class exhibits very low precision (1%) but high recall (93%), indicating that the model successfully identifies most negative reviews but also misclassifies a number of non-negative samples as negative. This behavior suggests that the Decision Tree is biased toward capturing minority class instances, likely influenced by the balanced dataset produced through SMOTE-Tomek.

As shown in the confusion matrix (Figure 7), most predictions lie on the main diagonal, confirming stable and accurate classification overall. The neutral class recorded 429 correct predictions, while the positive class achieved 122 correct predictions with only 7 misclassifications as neutral. The negative class was also reasonably detected, with 28 correct predictions and 2 misclassifications as neutral. However, the imbalance between precision and recall for the negative class indicates that further refinement is needed. To address this issue, applying class weighting or adjusting decision thresholds may help improve precision for negative sentiment without significantly reducing recall. Such adjustments could enable the Decision Tree model to produce more balanced classification outcomes across all sentiment categories while maintaining its interpretability and stability.

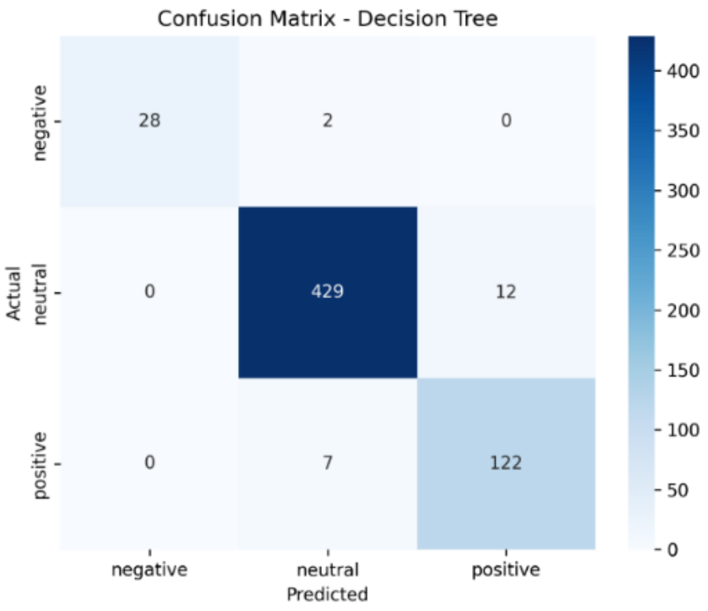


Figure 7. Confusion Matrix Decision Tree

Table 7. Overall Performance of Sentiment Classification Models

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.965	0.965	0.965	0.965
SVM	0.958	0.958	0.958	0.956
Logistic Regression	0.909	0.909	0.903	0.888
Ensemble	0.986	0.986	0.986	0.986

3.5. Comparison Testing Algorithm

At this stage it is carried out comparison results between the four-method algorithm mentioned. Comparison results testing with use a comparison as shown in Figure 6.

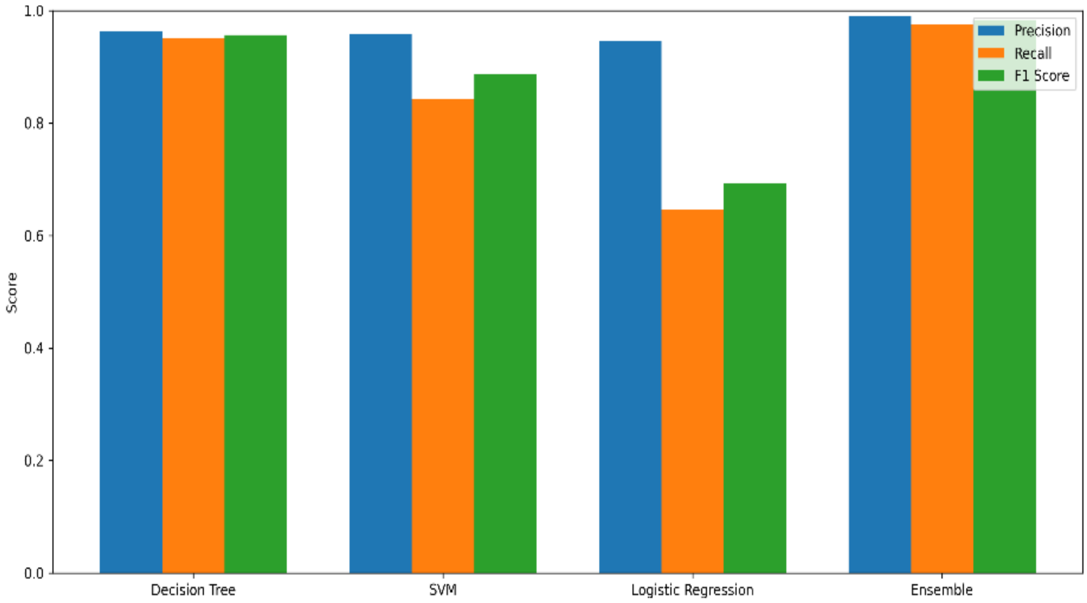


Figure 8. Comparison Diagram Algorithm

From the comparison diagram Figure 6. Based on results evaluation to four classification models, namely Decision Tree, Support Vector Machine (SVM), Logistic Regression, and Ensemble method, are seen that each model shows different performance on precision, recall, and F1-score metrics. Decision Tree produces sufficient performance stable in third metrics, with precision and F1-score values are in the range high, so that show good ability in classify data consistent. Meanwhile, SVM has high precision, but its recall is lower compared to other models, which indicates that this model tends to more selective and less capable catch all over variation class in a way comprehensive. Logistic Regression shows the lowest performance among three single models, especially on the recall value and F1-score, so this model is less effective in recognize all over samples on each class. In Overall, the Ensemble method provides results best in all metric evaluation, with the highest precision, recall, and F1-score values. This shows that merger some models can increase ability generalization and accuracy prediction in a way significant compared to single model usage. The evaluation results summarized in Table 6 show the advantages and disadvantages of each model, esp in detect sentiment negative, neutral, and positive in unbalanced data.

Table 1. Sentiment Classification Model

Algorithm	Accuracy (%)	Negative Precision	Remember the Negative	Neutral Precision	Remember Neutral	Positive Precision	Remember Positive
SVM	95.8%	9.5 %	60 %	96 %	99 %	97 %	94 %
LR	90.3%	1 %	17 %	89 %	99 %	94 %	78 %
Ensemble	98.6%	1%	97%	99%	1%	98%	96%
DT	96.5%	1 %	9 3 %	98 %	97 %	91 %	95 %

The results show a number of findings important:

1. Ensemble: Is the best model with Accuracy highest (98.67%) and dominant global performance (highest Precision, Recall, F1 Score). This model is very superior in *remember* sentiment Negative (Remember) Negative 97%) even though Precision The negative low.
2. Decision Tree: Being a single model best with Accuracy high (96.50%) and very good global performance balanced. He effective and reliable in classify all class, including Remember High negative (93 %).
3. SVM: Achieving Accuracy high (95.83%) but own weakness significant: Remember The negative very low (60%). This shows that this model often fails to identify sentiment data existing negatives.
4. Logistic Regression: Being the weakest model with Accuracy lowest (90.33%) and lowest global score. Weakness mainly is total failure in identify sentiment Negative (Remember) Negative only 17%).

Ensemble Model proven become the most effective and superior architecture for the task classification this sentiment, consistent reach metric Accuracy highest (98.67%) and best global performance. The Decision Tree Model is a single model with performance strongest and most balanced. It is important to note that SVM method and especially Logistic Regression shows significant difficulties in identify sentiment Negative, which underlines importance use Ensemble or Decision Tree approach to achieve optimal model generalization and sensitivity.

3.6. WordCloud Reviews

This analysis was conducted using a word cloud to display frequently occurring words in reviews [24]. Figure 7 shows WordCloud for everyone reviews, most frequently used words used by users in comment they. Words that are often appear such as "account", login, "but", "application", " so " to show all over reviews on discord sentiment.

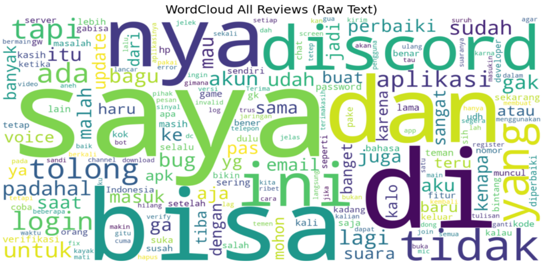

WordCloud All Reviews (Raw Text)

Figure 9. All Reviews Before Cleaning

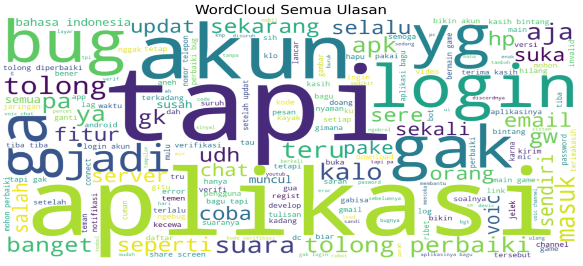

WordCloud Semua Ulasan

Figure 10. All Reviews After Cleaning

Figure 8. shows WordCloud for everyone negative, the most frequent words used by users in comment they. Words that are often appear such as "bug", "login", "fix " most often appear to show that problem technical and disruption access is complaint main user.

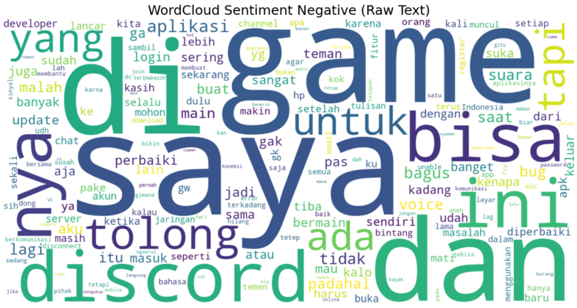

WordCloud Sentiment Negative (Raw Text)

Figure 11. Sentiment Negative Before Cleaning

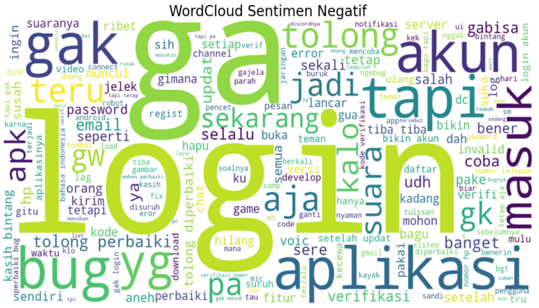
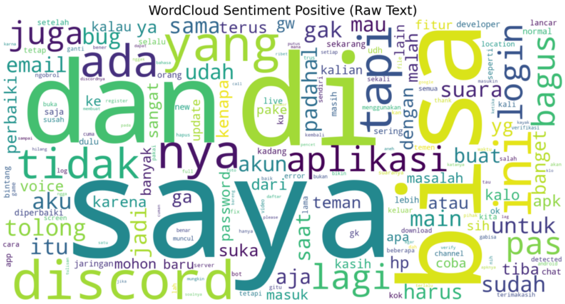

WordCloud Sentimen Negatif

Figure 12. entiment Negative After Cleaning

Figure 9. shows WordCloud for everyone sentiment positive, the most frequent words used by users in comment they. Words that are often appear such as "voice", "application", "if", "friends" most often appear generally used to convey appreciation or experience positive related use application. This shows appreciation to feature interactive and quality communication.


WordCloud Sentiment Positive (Raw Text)

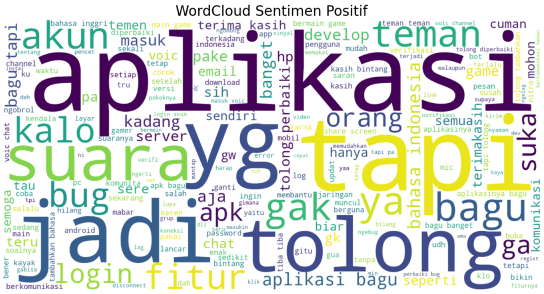
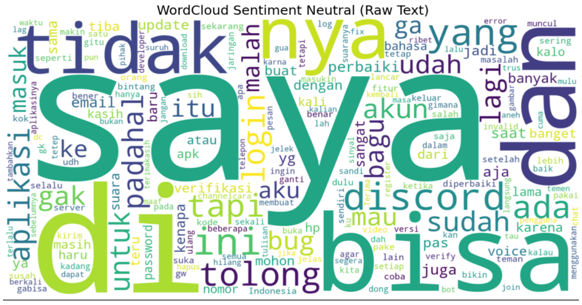

WordCloud Sentimen Positif

Figure 13. Sentiment Positive Before Cleaning **Figure14.** Sentiment Positive After Cleaning

Figure 10 shows WordCloud for everyone sentiment neutral, the most frequent words used by users in comment they. Words that are often appear like "but", "account", login", " please" most often appear generally used to convey statement without show evaluation positive and negative to application. This reflects doubts or comments without evaluation explicit.


WordCloud Sentiment Neutral (Raw Text)

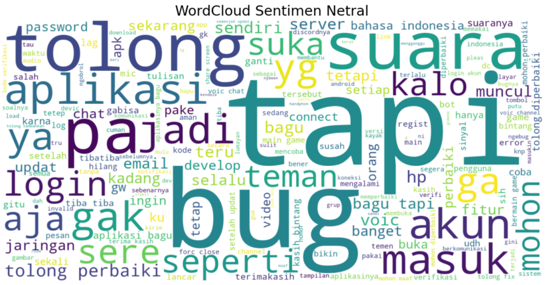

WordCloud Sentimen Netral

Figure 15. Sentiment Neutral Before Cleaning **Figure 16.** Sentiment Neutral After Cleaning

3.7. Discussion

The comparative analysis of the four sentiment classification methods reveals substantial differences in modeling effectiveness. The Ensemble model consistently outperforms the other approaches by achieving the highest global performance across precision, recall, and F1-score. Its primary advantage lies in the significantly improved recall for the minority (negative) class, which can be attributed to the aggregation of multiple base learners that produces more flexible decision boundaries. This ensemble mechanism, combined with SMOTE-Tomek balancing, enhances the model's sensitivity to rare complaint patterns. However, this emphasis on recall results in lower precision for the negative class, indicating a common trade-off where increased sensitivity leads to a higher rate of false positive predictions.

Among the single-model approaches, the Decision Tree demonstrates the most balanced and robust performance across all sentiment classes. Its hierarchical rule-based structure enables adaptive feature splitting that effectively captures sentiment variations, including minority-class patterns, without overfitting. In contrast, Support Vector Machine (SVM) and Logistic Regression (LR) exhibit notable limitations in handling negative sentiment. Although SVM maintains relatively strong overall accuracy, its recall for the negative class declines sharply, suggesting that the decision margin is still influenced by the dominant neutral class despite data balancing. Logistic Regression shows the weakest performance, indicating underfitting, as the model tends to classify most reviews as neutral. This behavior reflects the limitations of linear decision boundaries and insufficient regularization tuning when dealing with complex and context-dependent sentiment expressions.

Model performance is further affected by the translation-based lexicon labeling approach. Translating Indonesian reviews into English prior to sentiment labeling introduces semantic distortion and loss of emotional nuance, particularly for informal expressions and local language patterns. This limitation reduces the reliability of sentiment polarity assignments, especially for negative reviews. The WordCloud visualization supports these findings, where dominant terms such as "bug," "login," and "fix" clearly indicate that technical stability and access issues are the primary sources of user complaints. Therefore, the results confirm that Ensemble and Decision Tree models provide the most reliable and actionable insights for understanding user sentiment and guiding service improvement strategies, consistent with previous studies [25].

4. CONCLUSION

This study contributes to the comparison of various machine learning (ML) algorithms for sentiment classification on Indonesian-language Discord application reviews. The results highlight the advantages of the Ensemble and Decision Tree models in addressing imbalanced data issues. The Ensemble model performed the best overall, achieving higher accuracy and better recall for the negative sentiment class, albeit at the expense of precision. On the other hand, the Decision Tree model also demonstrated strong performance, balancing recall and precision, although it still exhibited weaknesses in precision for the negative class.

Suggestions for Future Research, this study has limitations particularly regarding the use of deep learning-based models. Therefore, we suggest that future research explore the use of BERT or LSTM models, which have proven effective in capturing more complex semantic contexts within text. Additionally, the application of a native Indonesian sentiment lexicon could further enhance the accuracy of sentiment classification, making it more sensitive to local context and reducing issues arising from translation mismatches and lexicon inconsistencies.

REFERENCE

[1] R. Q. Rohmansa, N. Pratiwi, and M. J. Palepa, "Analisis Sentimen Ulasan Pengguna Aplikasi Discord Menggunakan Metode K-Nearest Neighbor," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 1, pp. 368–378, Feb. 2024, doi: 10.29100/jipi.v9i1.4943.

[2] P. W. Ciady and S. Hariyanto, "Implementasi Sentimen Emosi Pada Lirik Lagu Menggunakan Bot Discord Dengan Metode Analisis Sentimen Berbasis Leksikon," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 5, Okt. 2024, doi: 10.36040/jati.v8i5.11103

[3] M. Minarni, "Pelatihan Pemanfaatan Aplikasi Discord Sebagai Kelas Virtual Bagi Guru Se-Kotawaringin Timur," *Dinamisia : Jurnal Pengabdian Kepada Masyarakat*, vol. 6, no. 4, Aug. 2022, doi: 10.31849/dinamisia.v6i4.5865.

[4] S. N. Salsabila, B. N. Sari, and R. Mayasari, "Klasifikasi Ulasan Pengguna Aplikasi Discord Menggunakan Metode Information Gain Dan Naive Bayes Classifier," *Infotech journal*, vol. 9, no. 2, pp. 383–392, Jul. 2023, doi: 10.31949/infotech.v9i2.6277.

[5] M. Rizky Pratama, Y. R. Ramadhan, and M. A. Komara, "Analisis Sentimen BRImo dan BCA Mobile Menggunakan Support Vector Machine dan Lexicon Based", *Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi,* vol. 12, no. 3, pp. 1439-1450, Nov. 2023, doi: 10.35889/jutisi.v12i3.143.

[6] S. E. Situmeang and N. P. Savina, "Analisis Perbandingan Metode Decision Tree, Random Forest, dan Support Vector Machine (SVM) dalam Memprediksi Kesehatan Janin," *Dept. Stat., Institut Teknologi Sepuluh Nopember, Surabaya*, Indonesia, Tech. Rep., pp. 1–8, 2024.

[7] RA, R. D. Y., R. A. Muhadi, A. Fitrianto, and P. Silvianti, "Analisis Regresi Logistik Biner dan Random Forest untuk Prediksi Faktor-Faktor Stunting di Pulau Jawa," *Euler: Jurnal Ilmiah Matematika, Sains dan Teknologi*, vol. 13, no. 2, pp. 147–156, 2025.

[8] A. Diki Prasetyo, F. T. Anggraeny, and R. Mumpuni, "Metode Ensemble Weighted Voting Untuk Deteksi Risiko Diabetes", *JIP (Jurnal Informatika Polinema)*, vol.11, no. 4, pp. 385-390, Agustus. 2025, doi: 10.33795/jip.v11i4.7353

[9] V. I. Yani, A. Aradea, and H. Mubarok, "Optimasi Prakiraan Cuaca Menggunakan Metode Ensemble pada Naïve Bayes dan C4.5," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 3, Dec. 2022, doi: 10.28932/jutisi.v8i3.5455.

[10] P. T. Prasetyaningrum, P. Purwanto, and A. F. Rochim, "Consumer Behavior Analysis in Gamified Mobile Banking: Clustering and Classifier Evaluation," *Online) Journal of System and Management Sciences*, vol. 15, no. 2, pp. 290–308, 2025, doi: 10.33168/JSMS.2025.0218.

[11] L. Maretva Cendani and A. Wibowo, "Perbandingan Metode Ensemble Learning pada Klasifikasi Penyakit Diabetes," vol. 13, no. 1, Mei. 2022, doi: 10.14710/jmasif.13.1.42912

[12] Steven Joses, D. Yulvida, and S. Rochimah, "Pendekatan Metode Ensemble Learning untuk Prakiraan Cuaca menggunakan Soft Voting Classifier," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 72–80, Jun. 2024, doi: 10.52158/jacost.v5i1.741.

[13] S. D. Parameswari *et al.*, "Studi Perbandingan Naïve Bayes dan Support Vector Machine (SVM) dalam Analisis Sentimen Pengguna Metaverse," *Jurnal Teknologi dan Manajemen Industri Terapan (JTMIT)*, vol. 4, no. 3, pp 1059-1065, Sept. 2025, doi: 10.55826/jtmit.v4i3.1122

[14] Sudarto and Kusrini, "JIP (Jurnal Informatika Polinema) Klasifikasi Tsunami Gempa Bumi Dengan Teknik Stacking Ensemble Machine Learning", vol. 10, no. 4, pp. 511-520, August. 2024, doi: 10.33795/jip.v10i4.5655

[15] O. I. Gifari, M. Adha, I. Rifky Hendrawan, F. Freddy, and S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *JIFOTECH (Journal of Information Technology*, vol. 2, no. 1, pp 36-40, Maret. 2022, doi: 10.46229/jifotech.v2i1.330

[16] G. R. Ati and P. T. Prasetyaningrum, "Analysis of Community Sentiment Towards Free Nutrition Meal Programs on Twitter Using Naïve Bayes, Support Vector Machine, K-Nearest Neighbors, and Ensemble Methods," *Journal of Information*

Systems and Informatics, vol. 7, no. 2, pp. 1443–1460, Jul. 2025, doi: 10.51519/journalisi.v7i2.1098.

[17] A. R. Isnain, H. Sulistiani, B. M. Hurohman, A. Nurkholis, and S. Styawati, "Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 8, no. 2, pp. 299–303, 2022.

[18] R. Perangin-angin, E. Julia Gunawati Harianja, I. Kelana Jaya, and B. Rumahorbo, "Penerapan Algoritma Safe-Level-SMOTE Untuk Peningkatan Nilai G-Mean Dalam Klasifikasi Data Tidak Seimbang," *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 4, no. 1, 2020, doi: 10.46880/jmika.vol4no1.pp67-72.

[19] H. K. Saka and P. T. Prasetyaningrum, "Sentiment Analysis and Classification of User Reviews of the 'Access by KAI' Application Using Machine Learning Methods to Improve Service Quality," *Journal of Information Systems and Informatics*, vol. 7, no. 2, pp. 1418–1442, Jun. 2025, doi: 10.51519/journalisi.v7i2.1099.

[20] I. D. Fareza and E. T. E. Handayani, "Analisis Sentimen Kualitas Aplikasi Discord Menggunakan Algoritma Naïve Bayes dan Support Vector Machine," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 4564–4571, 2025.

[21] S. Usman and F. Aziz, "Analisis Perilaku Pelanggan menggunakan Metode Ensemble Logistic Regression," *Jurnal Teknologi dan Ilmu Komputer Prima (Jutikomp)*, vol. 6, no. 2, pp. 90–97, 2023, doi: 10.34012/jutikomp.v6i2.4258

[22] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiasari, "Analisis sentimen pada rating aplikasi Shopee menggunakan metode Decision Tree berbasis SMOTE," *AITI: Jurnal Teknologi Informasi*, vol. 18, no. Agustus, pp. 173–184, 2021, doi: 10.24246/aiti.v18i2.173-184.

[23] E. R. Putri, D. A. Prasetya, and A. Junaidi, "Klasifikasi Perulangan Kanker Tiroid Menggunakan Stack Ensemble dan SMOTE," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 4211–4216, 2025, doi: 10.36040/jati.v9i3.13616.

[24] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 9, pp. 4305–4313, 2022.

[25] M. Idris, A. Rifai, and K. D. Tania, "Sentiment Analysis of Tokopedia App Reviews using Machine Learning and Word Embeddings," *sinkron*, vol. 9, no. 1, pp. 210–219, Jan. 2025, doi: 10.33395/sinkron.v9i1.14278.