INTEK Jurnal Penelitian
Vol. 12, No. 1, pp. 25-32, April 2025

Comparative Sentiment Analysis of SIREKAP Application Reviews Using Support Vector Machines and Naive Bayes

Khalid Khalid [1,*,a], Rifki Wijaya[1,b] and Moch Arif Bijaksana[1,c]
[1]Department School of Computing, Telkom University, Jalan. Telekomunikasi No. 1, Bandung, Jawa Barat, Indonesia, 40257
[*,a] khalid@student.telkomuniversity.ac.id (Corresponding Author), [b]rifkiwijaya@telkomuniversity.ac.id
[c]arifbijaksana@telkomuniversity.ac.id

***Abstract*—This study analyzes user sentiment toward the SIREKAP (Recapitulation Information System) application using reviews obtained from the Google Play Store. Two machine learning algorithms, Naïve Bayes and Support Vector Machine (SVM), were employed to classify sentiments in a dataset of 19,925 reviews. The data were preprocessed through text normalization, stopword removal, stemming, and tokenization. To mitigate class imbalance, both oversampling and undersampling techniques were implemented. Subsequently, feature extraction was performed using the Term Frequency–Inverse Document Frequency (TF–IDF) method, which transforms textual input into numerical feature vectors. The dataset was then divided into 80% for training (15,940 entries) and 20% for testing (3,985 entries). The findings indicate that oversampling yields superior model performance compared to undersampling. Using the oversampled dataset, the SVM algorithm achieved the highest classification accuracy of 95%, along with consistently strong precision, recall, and F1-scores across all sentiment categories. The Naïve Bayes classifier also demonstrated satisfactory performance with an accuracy of 77%, whereas both algorithms achieved only 61% accuracy under the undersampling condition. These findings suggest that combining the oversampling technique with the SVM model is the most effective strategy for handling imbalanced datasets and offers valuable insight into user sentiment regarding the SIREKAP platform.**

***Keywords*—Sentiment Analysis; Naïve Bayes; Support Vector Machine; Oversampling; SIREKAP**

I. Introduction

Elections in Indonesia represent a fundamental pillar of democracy, emphasizing direct citizen participation as a manifestation of popular sovereignty. Through these elections, Indonesians are granted the opportunity to select their political representatives, thereby reinforcing democratic consolidation and broadening political engagement across diverse social groups [1]. This participatory framework underscores Indonesia's commitment to ensuring fair representation and inclusivity in the national decision-making process [2].

Despite the democratic significance of Indonesia's elections, managing large-scale electoral processes presents considerable logistical and administrative challenges, especially in ensuring accurate and efficient vote recapitulation. To overcome these barriers, the General Election Commission (KPU) introduced the SIREKAP (Recapitulation Information System) as part of the digital transformation of election administration [3]. This innovation aims to enhance transparency, integrity, and efficiency during vote counting and reporting stages [4]. However, while the SIREKAP application has improved administrative efficiency at polling stations, its deployment continues to face technical and infrastructural obstacles. Several studies highlight persistent system instability and data upload errors that compromise the accuracy of vote recapitulation [5]. Furthermore, uneven internet connectivity across Indonesia has hindered election officers, particularly those in rural regions, from operating the system effectively [6]. These technological limitations mirror findings in broader digital election studies, which note that the success of e-recapitulation systems heavily depends on infrastructure readiness and stable connectivity [7]. In addition, the scanned results often fail to synchronize with the physical data, while poor image quality, improper capture angles, and inconsistent lighting conditions hinder the ability of Optical Character Recognition (OCR) and Optical Mark Recognition (OMR) technologies to reliably interpret and extract information [8].

DOI: http://dx.doi.org/10.31963/intek.v12i1.5196

Given these operational challenges, understanding user perceptions and interactions with digital election systems such as SIREKAP is crucial for identifying functional improvements and ensuring successful implementation in future elections. One effective approach to capturing user perceptions is sentiment analysis, a data-driven technique in natural language processing (NLP) that systematically extracts and interprets emotional cues, opinions, and attitudes expressed in textual data [9, 10]. Through the integration of machine learning algorithms, sentiment analysis can automatically classify user-generated content into positive, negative, or neutral sentiments, offering quantifiable insights for evaluating and improving digital systems [11, 12]. Applying this method to user feedback and app reviews, particularly from the SIREKAP application, can provide valuable evidence of user satisfaction, technical challenges, and potential enhancement opportunities [13].

In this research, sentiment analysis will focus on two main data sources. First, user reviews on the Play Store provide rich information about users' experiences and perceptions of the SIREKAP app [14]. Through sentiment analysis of these user reviews, we can identify the level of user satisfaction, obstacles faced, and suggestions for improvement. Second, reviews related to the SIREKAP app provide a broader picture of user perceptions of the SIREKAP app. By analyzing the sentiment of Play Store interactions, it is possible to identify the topics discussed, the opinions expressed, and the sentiment contained in these interactions.

Using a variety of techniques and platforms, sentiment analysis of mobile applications has been the focus of numerous studies. For example, previous research analyzed user evaluations of ride-hailing applications such as Gojek and Grab from the Google Play Store using the Naïve Bayes and Support Vector Machine (SVM) approaches, revealing that most users expressed positive sentiments toward the applications [15, 16]. In addition, studies employing SVM-based sentiment analysis on social media posts related to Indonesia's presidential elections have identified strong opinion polarization among users, demonstrating the model's capability in capturing political sentiment dynamics [17, 18].

By utilizing various sentiment analysis methods, previous studies provide valuable insights into users' perceptions and opinions towards mobile applications as well as important events in the social and political context. The purpose of this study is to evaluate the accuracy of the Support Vector Machine (SVM) and Naïve Bayes algorithms for sentiment analysis of the Sirekap application. The study will determine the best algorithm for categorizing user attitude about the program by means of this comparison.

II. Research Methodology

A. System Design

This study aims to analyze the tone of Google Play Store reviews of the SIREKAP application. The system design is illustrated in Figure 1.

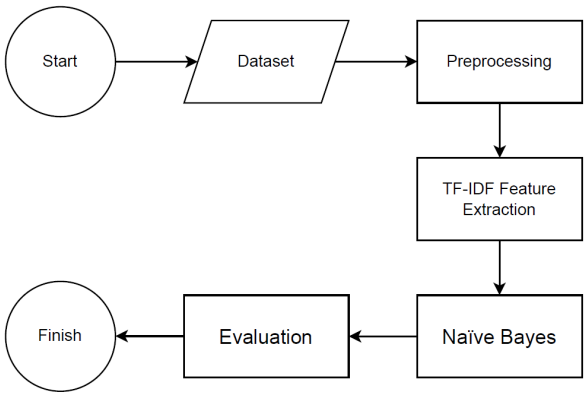

Figure 1. Flowchart of system design

B. Dataset

The dataset used in this study was collected through Python-based web scraping techniques. It consists of approximately 19,925 user reviews of the SIREKAP application from the Google Play Store. Each review has been categorized into one of three sentiment classes: positive, negative, or neutral. Specifically, the dataset includes 15,434 negative reviews, 3,529 positive reviews, and 962 neutral reviews. An example of the labeled dataset is illustrated in Figure 2. These labeled reviews form the foundation for sentiment analysis in this study.

DOI: http://dx.doi.org/10.31963/intek.v12i1.5196

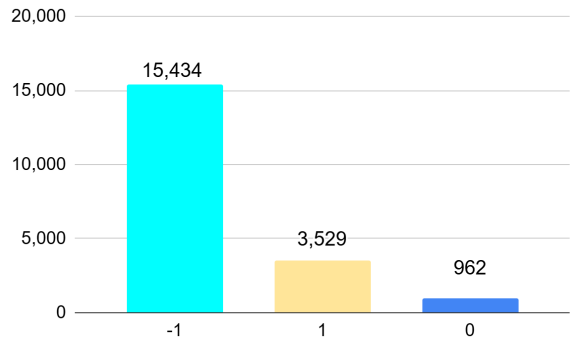

Figure 2. Dataset based on class distribution

C. Preprocessing

The purpose of preprocessing is to solve issues with data processing, particularly with comments. Cleansing, Case Folding, Tokenization, Stopword Removal, and Stemming are the preprocessing steps used [19]. A flowchart of the preparation steps is shown in Figure 3.

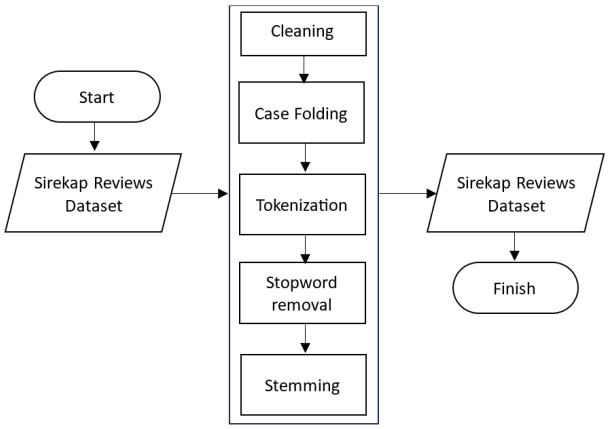

Figure 3. Processing flowchart

1) Cleansing

One preprocessing technique, called cleaning, seeks to eliminate superfluous words and punctuation [20]. Table 1 provides examples of the cleaning procedure.

Table 1. Cleansing

Before Cleansing	After Cleansing
For the future, if possible, just upload PDFs. If you upload photos one by one, it will be slow unless we have a good server. This is accessed by the whole of Indonesia, not one district.	*In the future, we can just upload photos one by one, it will be slow unless we have a server that...this is accessed by all of Indonesia, not one district.*

2) Case Folding

Converting every capital or uppercase letter in a word to a lowercase letter is known as case folding [19]. Table 2 shows instances of case folding.

Table 2. Case folding

Before Case Folding	After Case Folding
For the future, if possible, just upload PDFs. If you upload photos one by one, it will be slow unless we have a good server. This is accessed by the whole of Indonesia, not one district.	*In the future, if possible, just upload PDFs. If you upload photos one by one, it will be slow unless we have a good server. This is accessed by the whole of Indonesia, not by one district.*

3) Stopword Removal

The process of removing words and phrases that have little bearing on the classification process and are meaningless or irrelevant is known as stopword elimination [21]. Table 3 provides an illustration of the Stopword Removal procedure.

Table 3. Stopword removal

Before Stopword Removal	After Stopword Removal
For the future, if possible, just upload PDFs. If you upload photos one by one, it will be slow unless we have a good server. This is accessed by the whole of Indonesia, not one district.	*In the future, we can just upload photos one by one, it will be slow unless we have a server that...this is accessed by all of Indonesia, not one district.*

4) Stemming

Discovering a word's core form is the purpose of the stemming process. Using certain algorithms, affixes like prefixes, suffixes, and combinations will be eliminated during this procedure [22]. Table 4 provides examples of the stemming procedure.

Table 4. Stemming

Before Stemming	After Stemming
In the future, we can just upload photos one by one, it's slow unless we have a server that... this is accessed by all of Indonesia, not one district.	*For the front, you can just upload photos one by one, it's slow unless we have a server that...this is accessed by all of Indonesia, not one district.*

DOI: http://dx.doi.org/10.31963/intek.v12i1.5196

5) Tokenization

The technique of splitting sentences up into individual words is known as tokenization. Examples of the tokenization process can be seen in Table 5.

Table 5. Tokenization

Before Tokenization	After Tokenization
For the front, you can just upload photos one by one, it's slow unless we have a server that...this is accessed by all of Indonesia, not one district.	*"For", "front", "can", "that", "upload", "just", "photo", "one", "one","slow", "except", "we", "have", "server", "that",..", "this", "that", "access", "one", "Indonesia", "not", "one", "district"*

D. Data Splitting

The dataset used in this study consists of 19,925 entries. These entries were divided into training and testing subsets with an 80:20 ratio (as shown in Table 6). Specifically, 15,940 entries (80%) are utilized for training, while 3,985 entries (20%) are reserved for testing. This split was performed using the train_test_split function with a test_size parameter of 0.2. The training set is used to optimize the model, whereas the testing set serves to evaluate the model's performance on unseen data, providing an indication of its generalization ability.

Table 6. Split data result

Train Data	Test Data
15.940	3.985

E. TF-IDF Extraction Feature

A technique for assigning weight to a word or term in a document is called TF-IDF. This algorithm calculates weight by combining two ideas. The inverse of the frequency of documents containing a word, or IDF, and the frequency of occurrence of that term in a specific document, or TF. The TF-IDF extraction feature equation is as follows [23].

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (1)$$

If the frequency of term (t) in document (d) is represented by TF(t,d), which can be computed as:

$$tf = \frac{Number\ of\ Occurrences\ of\ t\ in\ d}{Total\ number\ of\ words\ in\ d} \quad (2)$$

IDF(t,D) is the inverse of the frequency of documents containing a term (t) across all documents (D), which can be calculated as:

$$IDF(t, D) = (\frac{Total\ number\ of\ documents}{number\ of\ documents\ containing\ t}) \quad (3)$$

F. Naïve Bayes

This study utilizes the Naïve Bayes algorithm to perform text classification on a preprocessed dataset that has undergone procedures such as tokenization, stopword removal, and text cleaning. To address class imbalance, an oversampling technique was applied to ensure an equal distribution of data across sentiment categories. The textual data was then converted into a numerical format using Term Frequency–Inverse Document Frequency (TF-IDF) vectorization, which assigns weights to words based on their frequency of occurrence [24]. Naïve Bayes, which is grounded in Bayes' Theorem and assumes conditional independence among features [25]. includes several variants such as Multinomial, Bernoulli, Poisson, and Ensemble Naïve Bayes. This study adopted the Multinomial Naïve Bayes model, as it is particularly well-suited for handling discrete data, especially in text classification tasks. Model evaluation was conducted using performance metrics such as precision, recall, F1-score, and a confusion matrix. The experimental results indicated that the Naïve Bayes algorithm, known for its efficiency, simplicity, and minimal training data requirements, achieved competitive performance in text classification tasks [26].

The subsequent equation illustrates the general formulation of Naïve Bayes:

$$(c|x) = \frac{P(X|c)\ x\ P(\mathrm{c})}{P(x)} \quad (4)$$

Description:

$P(c|x) : Probability\ of\ class\ c\ if\ word\ x\ appears$
$P(x|c) : Probability\ of\ word\ x\ if\ class\ c\ occurs$
$P(c) : Probability\ class\ c$
$P(x) : Probability\ of\ class\ x$

DOI: http://dx.doi.org/10.31963/intek.v12i1.5196

G. Support Vector Machine

Support Vector Machine (SVM) is a machine learning technique widely used for both regression and classification tasks [27]. The core principle of SVM is to identify the optimal hyperplane that separates data into distinct classes. As illustrated in Figure 4, SVM determines a decision boundary that maximizes the margin, which is the distance between the data points of each class and the hyperplane, to improve model generalization [28].

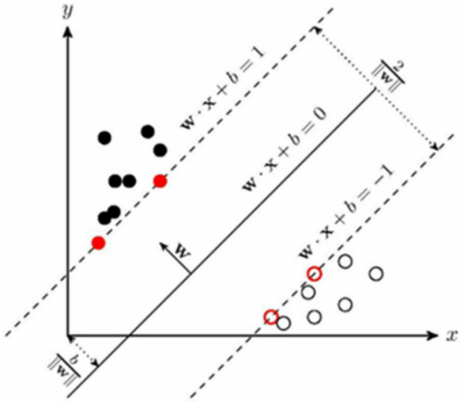

Figure 4. Kernel support vector machine [29]

In cases where linear separation is not feasible, SVM employs kernel functions, such as linear, polynomial kernels, or Radial Basis Function (RBF), to project the data into a higher-dimensional feature space where separation is possible [30]. SVM is particularly effective in high-dimensional spaces, where it can handle large feature sets and complex decision boundaries efficiently.

H. Confusion Matrix

During the evaluation phase, the confusion matrix was used to assess the model's performance. It generates four key outcomes: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Using these values, metrics such as accuracy, recall, precision, and F1-score can be calculated, with the corresponding formulas based on the TP, TN, FP, and FN values.

1) Recall

$$R = \frac{TP}{TP + FN} \quad (5)$$

Recall (R) is a metric that measures the model's ability to identify all positive class instances in the database Precision

2) Precision

$$P = \frac{TP}{TP + FN} \quad (6)$$

Precision A parameter called precision (P) gauges how well the model can detect positive class instances.

3) F1-Score

$$F1 = 2\,x\,\frac{(precision\ x\ recall)}{(precision + recall)} \quad (7)$$

F1-score (F1) is the mean of the recall and precision values.

4) Accuracy

$$\frac{TP + TN}{FP + FNTP + TN} \quad (8)$$

Accuracy (A) is the proportion of accurate forecasts (positive and negative) to all predictions.

III. Results and Discussion

The study evaluated the effectiveness of two classification algorithms, SVM and Naïve Bayes, for a sentiment analysis task. To address the imbalance inherent in the data, both oversampling and undersampling techniques were employed. The performance of each model was assessed using standard metrics, including accuracy, precision, recall, and F1-score [32].

This study utilized a dataset comprising 19,925 user reviews of the SIREKAP application collected from the Google Play Store. Prior to analysis, the data underwent several preprocessing steps, including text cleaning, case folding, stopword removal, stemming, and tokenization. The dataset was then split into training and testing sets, with 80% (15,940 entries) allocated for training and 20% (3,985 entries) for testing. To convert textual data into a numerical format, the Term Frequency–Inverse Document Frequency (TF-IDF) technique was applied.

DOI: http://dx.doi.org/10.31963/intek.v12i1.5196

After feature extraction, the Naïve Bayes algorithm was implemented for classification. The study further explored three different experimental scenarios to assess the impact of various conditions on the text classification performance of both the Naïve Bayes and Support Vector Machine (SVM) models using TF-IDF representations.

A. Comparison of Model Performance Based on Evaluation Table

The evaluation results from Table 7 indicate the performance of the classification model after applying oversampling to balance the data distribution across sentiment classes. The Naive Bayes algorithm achieved an accuracy of 77%, demonstrating its capability to classify sentiments effectively. Additionally, other metrics such as recall, precision, and F1-score were consistently balanced, ranging between 77% and 79%. These results highlight that the Naive Bayes model is reliable for datasets with a balanced class distribution.

On the other hand, SVM outperformed Naive Bayes significantly, achieving an impressive accuracy of 95%. Metrics such as recall, precision, and F1-score also reached 95%, underscoring the model's ability to classify all three sentiment categories (positive, neutral, and negative) consistently and with high accuracy. This performance demonstrates the robustness of SVM in handling sentiment classification tasks on balanced datasets.

Table 8 evaluates the performance of the Naïve Bayes and SVM algorithms under the undersampling condition, based on accuracy, recall, precision, and F1-score metrics. The results show that both algorithms have the same accuracy and recall of 61%, indicating an equal ability to classify the overall data and recognize positive data in each sentiment class. However, in terms of precision, Naïve Bayes is slightly superior with a value of 67% compared to SVM, which reaches 63%, indicating Naïve Bayes' positive predictions are more accurate. The F1-score of both algorithms is almost the same, which is 62% for Naïve Bayes and 61% for SVM, indicating that both models are able to handle validation data well although they still have limitations in capturing patterns on small datasets due to undersampling.

Table 7. Oversampling Evaluation

Metric	Naïve Bayes (%)	SVM (%)
Accuracy	77	95
Recall	77	95
Precision	79	95
F1-Score	77	95

Table 8. Undersampling evaluation

Metric	Naïve Bayes (%)	SVM (%)
Accuracy	61	61
Recall	61	61
Precision	67	63
F1-Score	62	61

B. Model Evaluation Based on Multi-Class ROC Curve

Figure 5 presents the ROC curves, offering valuable insights into the performance of the multi-class classification model.

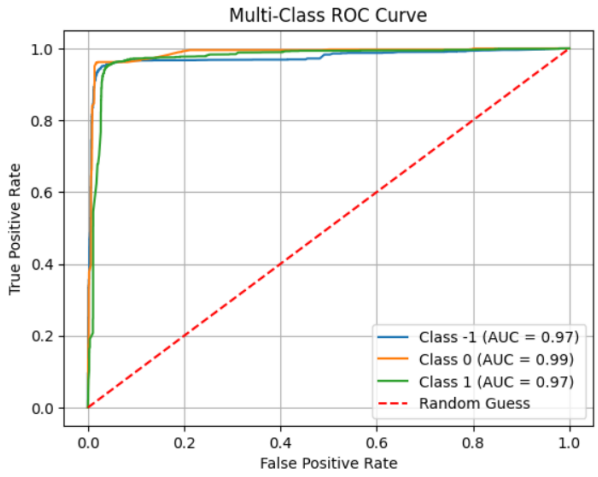

Figure 5. Multi-Class ROC curve

The Area Under the Curve (AUC) scores were notably high, 0.97 for class -1, 0.99 for class 0, and 0.97 for class 1, indicating the model's exceptional ability to differentiate between the sentiment categories. These near-perfect AUC values suggest that the classifier is highly effective in identifying both positive and negative instances across all classes. An AUC approaching 1 reflects stronger discriminative power, reinforcing the model's reliability in this context. Despite the overall strong performance, the slight variation in AUC values among the classes may point to marginal inconsistencies in classification accuracy. Such variations could stem

DOI: http://dx.doi.org/10.31963/intek.v12i1.5196

from complexities in the dataset's feature distribution or from imbalanced class representation. Additionally, the red dashed line in the graph represents the baseline performance of random guessing, which is clearly inferior compared to the model's output. Collectively, these results affirm the robustness and precision of the proposed model in handling multi-class classification tasks.

IV. Conclusion

This research investigated sentiment classification of user feedback on the SIREKAP application by employing Naïve Bayes and Support Vector Machine (SVM) algorithms, with a focus on addressing class imbalance through oversampling and undersampling strategies. The dataset comprised 19,925 user reviews, which were preprocessed through steps such as text cleaning and tokenization, and transformed into numerical features using the TF-IDF method. The data was then split into training and testing sets. Under the oversampling scenario, the SVM model demonstrated the best performance, achieving 95% in accuracy, precision, recall, and F1-score. In comparison, Naïve Bayes reached 77% accuracy, with corresponding performance metrics of 79% precision, 77% recall, and 77% F1-score. Conversely, with undersampling, both models performed less effectively. SVM attained 61% accuracy, with 63% precision, 61% recall, and a 61% F1-score, while Naïve Bayes matched the 61% accuracy but had slightly better precision (67%) and F1-score (62%). These findings highlight that the combination of oversampling and SVM offers the most promising approach for this sentiment analysis task, although it may be prone to overfitting. Moreover, while TF-IDF effectively represents text in numerical form, it lacks the ability to capture deeper semantic relationships between words. To improve model performance and generalizability, future research should explore more advanced text representation techniques, such as word embeddings, and evaluate the models on larger and more diverse datasets.

References

[1] K. J. Rajagukguk, S. Aripin, dan H. Wahyudi, "Simultaneous general election: It is fair for democracy in Indonesia," *J. Ilmu Pemerintahan: Kajian Ilmu Pemerintahan dan Politik Daerah*, vol. 6, no. 1, pp. 56–64, Mar. 2021, doi: 10.24905/jip.6.1.2021.56-64.

[2] W. Tuanaya, M. Wance, and Muhtar, "Analysis on election models: Examining the Indonesian case", *J. of Law and Sust. Develop.*, vol. 11, no. 8, p. e1080, Oct. 2023.

[3] T. Haryadi, A. Nurmandi, I. Muallidin, D. Kurniawan, and Salahudin, "Implementing "SIREKAP" application based on election for improving the integrity of election administrators and increasing public trust," in *Human Interaction, Emerging Technologies and Future Systems* V, T. Ahram and R. Taiar, *Ed. Cham: Springer*, 2022, pp. 248–258, doi: 10.1007/978-3-030-85540-6_21.

[4] D. Huda, A. E. Winarto, and L. Lestariningsih, "Analysis of 2024 general election digitalization system as an effort to improve the quality of democracy in Indonesia," *J. Dev. Res.*, vol. 7, no. 2, pp. 272–282, Nov. 2023, doi: 10.28926/jdr.v7i2.313.

[5] M. Habibi, A. Mahadika, and W. Astuti, "Digital dilemma: Technology in the vote counting process for general elections and local head elections in Indonesia," *Jurnal Ilmu Pemerintahan,* vol. 13, no. 3, pp. 377-389, 2023, doi: 10.26618/ojip.v13i3.12729.

[6] H. Indriyani and A. Meyer, "Use of information technology in counting (Situng) and recapitulating (recapitalized) votes for the 2024 election," *Jurnal Administrasi Publik*, vol. 3, no. 1, pp. 38-52, May. 2023.

[7] Y. Djuyandi, A. Herdiansah, I. Yulita, and S. Sudirman, "Using vote e-recapitulation as a means to anticipate public disorders in election security in Indonesia," *Humanities & Social Sciences Reviews*, vol. 7, no. 5, pp. 111–122, Sept. 2019, doi: 10.18510/hssr.2019.7515.

[8] R. R. Danti, H. Hilman, and M. I. Rantau, "The impact of using the vote recapitulation information system (SIREKAP) application on the vote recapitulation process in the 2024 election (case study of the Tangerang City General Election Commission)," *Dialektika J. Ilmu Sos*., vol. 22, no. 2, pp. 980–990, 2024.

[9] M. Ahad, "A Review article: Use of sentiment analysis in social media," *Int. J. Eng. Appl. Sci. Technol.*, vol. 7, no. 9, pp. 171–176, 2023, doi: 10.33564/ijeast.2023.v07i09.026.

[10] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," *Procedia Comput. Sci.*, vol. 57, pp. 821–829, 2015, doi: 10.1016/j.procs.2015.07.523.

[11] A. Romadhony, S. A. Faraby, R. Rismala, U. N. Wisesti, and A. Arifianto, "Sentiment analysis on a large Indonesian product review dataset," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 1, pp. 167–178, 2024, doi: 10.20473/jisebi.10.1.167-178

[12] F. Y. A'la, "Indonesian sentiment analysis towards mypertamina application reviews by utilizing machine learning algorithms," *Journal of Informatics Information System Software Engineering and Applications.*, vol. 5, no. 1, pp. 080–091, Nov. 2022, doi: 10.20895/INISTA.V5I1.

[13] R. Setiyawan and Z. Mustofa, "Comparison of the performance of naive bayes and support vector machine in sirekap sentiment analysis with the lexicon-based approach," *J. Soft Comput. Explor.*, vol. 5, no. 2, pp. 122–132, 2024, doi: 10.52465/joscex.v5i2.367.

[14] R. Setiyawan and Z. Mustofa, "Comparison of the performance of Naïve Bayes and Support Vector Machine in SIREKAP sentiment analysis with the lexicon-based approach," *Journal of Soft Computing Exploration*, vol. 5, no. 2, 2024, doi: 10.52465/joscex.v5i2.367.

[15] B. F. Wiguna, H. Herlawati, and A. Y. P. Yusuf, "Sentiment analysis of on-demand ride-hailing systems Using Support Vector Machine and Naïve Bayes," *PIKSEL: Penelit. Ilmu Komput., Sist. Embedded dan Logic.*, vol. 11, no. 2, pp. 401–414, Sep. 2023, doi: 10.33558/piksel.v11i2.738.

[16] N. Suhaimi and M. Lestari, "Sentiment analysis of TikTok app reviews on Google Play using several machine learning methods," *Int. J. Global Oper. Res.*, vol. 5, no. 4, pp. 275–287, Des. 2024, doi: 10.47194/ijgor.v5i4.343.

[17] N. Mardiah, L. Marlina, K. Z. Sitorus, and M. Iqbal, "Analysis of indonesian people's sentiment towards 2024 presidential candidates on social media using Naïve Bayes Classifier and Support Vector Machine," *Build. Inf., Technol. Sci. (BITS)*, vol. 6, no. 2, pp. 950–960, Sep. 2024, doi: 10.47065/bits.v6i2.5766.

[18] W. Dirgantara, F. I. Maulana, Subairi, and R. Arifuddin, "The performance of machine learning model Bernoulli Naïve Bayes, Support Vector Machine, and Logistic Regression on COVID-19 in Indonesia using sentiment analysis," *TECHNE: J. Ilmiah Elektrotek.*, vol. 23, no. 1, Art. 446, 2024, doi: 10.31358/techne.v23i1.446.

[19] Yuyun, A. D. Latief, T. Sampurno, Hazriani, A. O. Arisha and Mushaf, "Next Sentence Prediction: The Impact of Preprocessing Techniques in Deep Learning," *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Bandung, Indonesia, 2023, pp. 274-278, doi: 10.1109/IC3INA60834.2023.10285805.

[20] W. F. Satrya, R. Aprilliyani, and E. H. Yossy, "Sentiment analysis of Indonesian police chief using multi-level ensemble model," *Procedia Comput. Sci.*, vol. 216, pp. 620–629, 2022, doi: 10.1016/j.procs.2022.12.177.

[21] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts", *KINETIK*, vol. 4, no. 4, pp. 375-380, Oct. 2019.

[22] M. U. Albab, Y. Karuniawati P, and M. N. Fawaiq, "Optimization of the stemming technique on text preprocessing President 3 periods topic," *J. Transform.*, vol. 20, no. 2, pp. 1–10, 2023.

[23] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A Survey on sentiment analysis methods, applications, and challenges,"*Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/S10462-022-10144-1.

[24] S. Hidayat, N. Rahaningsih, R. D. Dana, and Mulyawan, "Improvement of user sentiment classification model for the Indomaret Poinku application using the Naïve Bayes method," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 4, no. 2, February. 2025, doi: 10.59934/jaiea.v4i2.937.

[25] S. Rahman, M. Hasan, and A. K. Sarkar, "Prediction of brain stroke using machine learning algorithms and deep neural network techniques," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 1, pp. 23–30, 2023, doi: 10.24018/ejece.2023.7.1.483.

[26] A. Sitepu, W. Wanayumini, and Z. Situmorang, "Determining bullying text classification using Naïve Bayes classification on social media," *VARIAN: Jurnal Ilmiah Ilmu Komputer*, vol. 4, no. 2, pp. 133-140, 2021, doi: 10.30812/varian.v4i2.1086

[27] S. N. Khan, S. U. Khan, H. Aznaoui, C. B. Şahín, and Ö. Dinler, "Generalization of linear and non-linear support vector machine in multiple fields: A review," *Computer Science and Information Technologies*, vol. 4, no. 3, pp. 226-239, 2023, doi: 10.11591/csit.v4i3.p226-239.

[28] K.-L. Du, B. Jiang, J. Lu, J. Hua, and M. N. S. Swamy, "Exploring kernel machines and support vector machines: Principles, Techniques, and Future Directions", *Mathematics*, vol. 12, no. 24, p. 3935, Dec. 2024, doi: 10.3390/math12243935.

[29] N. Chen, Y. Sun, Z. Wang, and C. Peng, "Improved LS-SVM method for flight data fitting of civil aircraft flying at high plateau," *Electronics*, vol. 11, no. 10, Art. no. 1558, 2022, doi: 10.3390/electronics11101558.

[30] C. Ding, T.Y. Bao, and H. L. Huang , "Quantum-inspired support vector machine," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 33, no. 12, pp. 2423–2434, Dec. 2022, doi: 10.1109/TNNLS.2021.3084467.

[32] B. S. Nugroho and W. Maharani, "Support vector machine and Naïve Bayes for personality classification based on social media posting patterns," *Build. Informat. Technol. Sci*., vol. 6, no. 3, pp. 1717–1731, Dec. 2024, doi: 10.47065/bits.v6i3.6411.