

SENTIMENT ANALYSIS OF JKN MOBILE APPLICATION IN HEALTH TECHNOLOGY USING NAIVE BAYES, GRADIENT BOOSTING, AND ADABOOST

Deni Utama*

Sistem dan Teknologi informasi
Universitas Negeri Jakarta,
Indonesia

Rauhil Fahmi

Sistem dan Teknologi informasi
Universitas Negeri Jakarta,
Indonesia

**Muhammad Ridho Kurniawan
Pratama**

Sistem dan Teknologi informasi
Universitas Negeri Jakarta,
Indonesia

Article Info

Article history:

Received: 7 November 2025

Revised: 28 November 2025

Accepted: 10 Desember 2025

DOI: 10.21009/jvote.v8i2.72073



Keywords:

Sentiment Analysis
JKN Mobile
Naive Bayes
Gradient Boosting
AdaBoost

Abstract

The increasing adoption of mobile health services highlights the need to evaluate user experience through feedback. The JKN Mobile application, developed by BPJS Kesehatan (Indonesia's Social Security Agency for Health), has accumulated thousands of user reviews on the Google Play Store. These reviews provide an opportunity to gain insights into public sentiment regarding the app. This study conducts sentiment analysis using three machine learning algorithms—Naive Bayes, Gradient Boosting, and AdaBoost—to classify user opinions as positive, negative, or neutral. The reviews were preprocessed, vectorized using TFIDF, and evaluated using standard classification metrics. The results indicate that Naive Bayes achieves the highest classification performance, followed by AdaBoost and Gradient Boosting. These findings offer valuable guidance for developers and policymakers in improving the app's usability and functionality.

To cite this article: Utama, D., Fahmi, R., & Pratama, M, R, K. (2025). Sentiment Analysis of JKN Mobile Application in Health Technology Using Naive Bayes, Gradient Boosting, and AdaBoost. *Jurnal Pendidikan Vokasional Teknik Elektronika*, 8(2), 80-90

INTRODUCTION

This research is the first part of a five-year roadmap focusing on System and Technology for Health. The roadmap aims to develop innovative, AI-driven, and inclusive health technologies that support equitable healthcare access, particularly for vulnerable groups such as the visually impaired.

Mobile health applications have become a cornerstone in public healthcare delivery, especially in developing countries. In Indonesia, the JKN Mobile application provides various conveniences for National Health Insurance (JKN) participants in accessing healthcare services. Some of its benefits include easy membership status checks, billing and payment information, finding healthcare facility locations, and registering for services. Given its importance, assessing user feedback is critical to ensure the application's effectiveness and user satisfaction. Online platforms like the Google Play Store provide a rich source of user-generated reviews that can be mined for sentiment analysis.

Corresponding author:

Deni Utama, Universitas Negeri Jakarta, Indonesia (deniutama@unj.ac.id)

This paper applies three machine learning approaches—Naive Bayes, AdaBoost, and Gradient Boosting—to classify the sentiment of JKN Mobile application reviews. By understanding user sentiment, service providers can identify issues, prioritize updates, and enhance user experience.

Sentiment analysis is a rapidly growing area in natural language processing (NLP) and data mining. It focuses on extracting subjective information from textual data to determine users' attitudes toward a particular product, service, or topic. Pang et al. (2002) were among the early researchers who demonstrated the efficacy of machine learning algorithms for sentiment classification tasks. They showed that Naive Bayes performs effectively on structured reviews, especially when combined with proper text preprocessing.

Breiman (2001) introduced Random Forest as an ensemble learning method that builds multiple decision trees and merges their outputs to improve classification performance. This method is robust to noise and suitable for high-dimensional data such as text vectors. Liu (2012) emphasized that machine learning algorithms like Naive Bayes and Random Forest are effective for sentiment analysis when paired with appropriate feature extraction techniques such as TF-IDF.

Gradient Boosting is an ensemble technique that builds models sequentially by optimizing a loss function, improving upon previous errors (Friedman, 2001). It has been successful in many text mining tasks due to its ability to handle complex patterns in data. AdaBoost, introduced by Freund and Schapire (1997), combines weak learners to form a strong classifier. Its application in sentiment classification has shown effectiveness in handling binary and multiclass sentiment data.

Recent work by Fahrezi and Solichin (2024) achieved an impressive 98.57% accuracy using Naïve Bayes on sentiment classification of JKN Mobile reviews. Similarly, Harta et al. (2024) employed Random Forest for aspect-based sentiment analysis, highlighting the model's interpretability in revealing issues like login errors and referral delays.

Other comparative studies, such as that by Miftahusalam et al. (2023), evaluated Naive Bayes and Random Forest on mobile banking apps and observed that while Naive Bayes provided higher precision, Random Forest was more resilient in handling ambiguous sentiment cases. Zaenudin et al. (2023) implemented an ensemble model combining Naive Bayes and logistic regression, showing enhanced performance on multi-source sentiment data.

Table 1. Novelty and Contribution

No	Aspect	Previous Studies	This Study
1	Data Collection	Collected data from Twitter	Collected data from Twitter
2	Methods Used	Only applied one or two methods	Uses multiple methods for comparison
3	Algorithms	Used common algorithms: Random Forest, SVM, KMeans, K-Nearest Neighbor, Logistic Regression	Applies Naive Bayes, Gradient Boosting, and AdaBoost
4	Research Continuity	Standalone studies with limited scope	Has continuity and contributes to the roadmap of Health Technology research
5	Data Size	Limited datasets	Larger dataset with 500 samples
6	Preprocessing Approach	Manual sentiment labeling or simple Bag of Words	Automated preprocessing using Bag of Words
7	Complexity of Processing	Basic preprocessing techniques	More complex preprocessing for better classification performance

Here is the formal academic English translation for the Novelty and Contribution section of your research paper:

1. Novelty and Contribution

This study offers several significant contributions that distinguish it from previous literature in the field of health technology sentiment analysis. The primary novelty of this research is rooted in three main pillars: Boosting-Based Algorithmic Comparison, Data Scalability, and Strategic Research Integration.

2. Multi-Method Algorithmic Comparison

Unlike previous studies that generally focus on a single or limited set of standard methods—such as Random Forest or SVM—this research implements a broader comparative approach. By benchmarking Naive Bayes (as a baseline) against two advanced ensemble learning algorithms, Gradient Boosting and AdaBoost, this study provides deeper insights into the effectiveness of boosting techniques in handling the ambiguous nature of health application reviews.

3. Processing Complexity and Scalability

From a technical perspective, this research enhances data quality standards through:

- a. Data Volume: Utilizing a larger dataset (500 samples) compared to limited previous studies, ensuring better model generalization.
- b. Preprocessing Automation: Moving beyond subjective manual labeling toward a more complex, automated preprocessing pipeline to minimize human error and significantly improve overall classification performance.

4. Sustainability within the Health Technology Roadmap

A crucial differentiator of this research is its alignment with the Health Technology Research Roadmap. Rather than being a standalone study with limited scope, this research is designed for continuity, ensuring that the sentiment analysis of the JKN Mobile application contributes directly to the future development of digital health service strategies.

METHOD

This chapter outlines the systematic framework employed to conduct a sentiment analysis of the JKN Mobile application. The approach utilizes a machine learning-based experiment encompassing the stages of data acquisition, text preprocessing, and feature extraction, leading to classification using three primary algorithms: Naive Bayes, Gradient Boosting, and AdaBoost.

1. Data Collection

Data Collection in this study refers to the process of collecting user reviews from platforms such as the Google Play Store, using scraping methods Node Js, to obtain text data to be analyzed. Data collection from online platforms provides large-scale user opinions that are valuable for sentiment classification (Munir et al., 2021). Effective sentiment analysis depends not only on model selection but also on the quality and diversity of collected data from social platforms (Kharde & Sonawane, 2016).

User reviews were scraped from the Google Play Store using the google-play-scraper Node Js package. A total of 10.000 reviews were gathered, ensuring a variety of app versions and timestamps. Each review was text processing Node Js labeled as either positive, negative, or neutral by bag of word processing.

```
const positifWords = ['mantul', 'bagus', 'mantap', 'cepat', 'mudah', 'keren', 'puas', 'lancar', 'top', 'terbaik', 'hebat', 'recommended'];
const negatifWords = ['lemot', 'blank', 'error', 'gagal', 'susah', 'buruk', 'jelek', 'parah', 'macet', 'kecewa', 'lambat', 'ngelag', 'reset'];
```

After being labeled by the bag of words, the labeler is then validated again manually so that it is more precise if there are words that are sarcastic.

2. Data Collection

Text preprocessing transforms raw text into a clean and structured format, which is crucial for improving model performance in sentiment analysis (Chai et al., 2022). Preprocessing steps like stopwords removal, lowercasing, and stemming significantly affect classification accuracy in Twitter sentiment analysis (Jianqiang & Xiaolin, 2018).

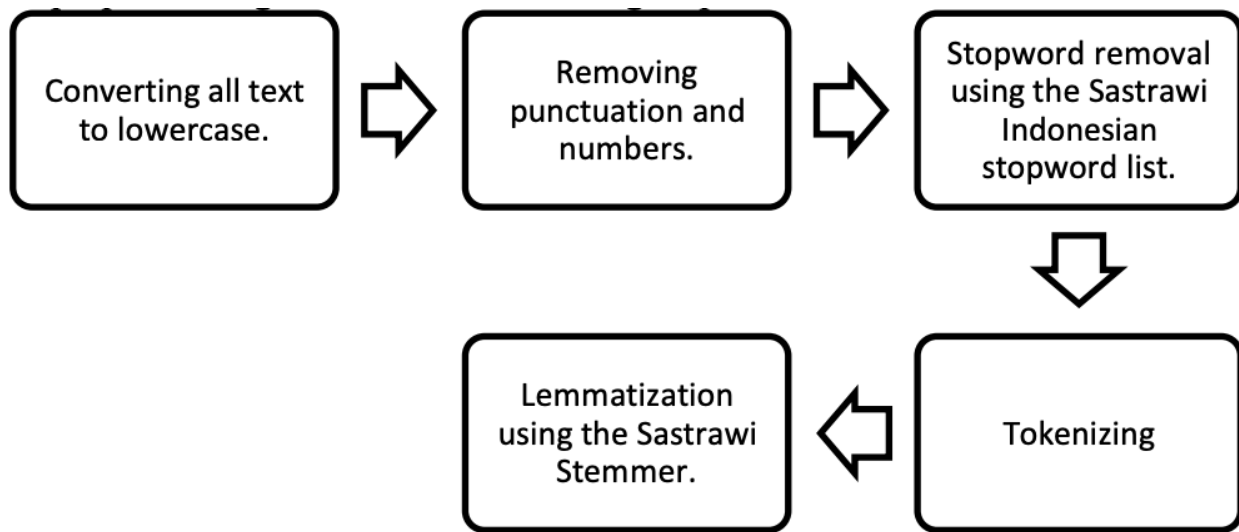


Figure 1. Research Flow

This section describes the structured approach used to process and analyze user feedback. To ensure the machine learning models can effectively interpret the raw data, a rigorous preparation phase was conducted. The preprocessing stage is a critical step in transforming unstructured text data into a meaningful numerical format for machine learning algorithms. The following procedures were implemented to ensure data quality and model consistency:

2.1. Case Folding

The initial step involved converting all text to lowercase. This process standardizes the tokens, ensuring that the system treats words such as **"Health"**, **"HEALTH"**, and **"health"** as the same entity. This prevents the unnecessary expansion of the feature space and maintains consistency across the dataset.

2.2. Noise Removal

User reviews often contain non-alphabetic characters, including punctuation (., !, ?), special symbols (@, #), and numbers. These elements were removed as they do not carry significant sentiment information and are considered "noise" that can hinder the accuracy of the classification models.

2.3. Stopword Removal

Stopwords are high-frequency words that carry minimal information (e.g., "and", "the", or in Indonesian, "yang", "dan", "di"). This study utilized the **Sastrawi** library to filter out these common words. By omitting them, the models can focus on keywords that represent specific opinions or emotional weight.

2.4. Lemmatization and Stemming

To preserve semantics while reducing dimensionality, the **Sastrawi Stemmer** was applied. This process reduces words to their canonical or base forms by stripping Indonesian affixes (prefixes, suffixes, and infixes).

> **Example:** The words **"pelayanan"** (service), **"melayani"** (to serve), and **"dilayani"** (served) are all reduced to the root word **"layan"**.

2.5. Tokenization

Tokenization is the process of splitting the cleaned text strings into discrete units called tokens (individual words). These tokens serve as the fundamental units required for mapping the text into a vector space in the subsequent phase.

2.6 Feature Extraction: TF-IDF Vectorization

Once the text was fully cleaned, it was transformed into a numerical representation using the **TF-IDF (Term Frequency-Inverse Document Frequency)** method. This technique assigns a weight to each word based on its importance within a specific document relative to the entire corpus.

The TF-IDF weight is calculated using the following formula:

$$W_{i,j} = TF_{i,j} \times \log\left(\frac{N}{DF_i}\right)$$

Where:

- * $TF_{i,j}$ represents the frequency of term in document
- * N is the total number of documents in the dataset.
- * DF_i is the number of documents containing term

2.7 Significance of TF-IDF:

Unlike a simple **Bag of Words** approach which only counts occurrences, TF-IDF penalizes words that appear too frequently across all reviews and rewards words that are unique and discriminative (such as "satisfied," "slow," or "disappointed"). This makes it highly effective for identifying specific sentiment patterns in the JKN Mobile user feedback.

3. Model Algorithms

Following the preprocessing and feature extraction phases, the processed dataset is utilized to train and evaluate the classification models. This study employs a comparative analysis between a traditional probabilistic model and advanced ensemble learning techniques to determine the most effective approach for sentiment detection in health technology. The following algorithms were selected based on their proven performance in natural language processing tasks:

3.1. Naive Bayes (MultinomialNB)

Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem, assuming that all features are independent (naive assumption). The Multinomial variant is particularly suitable for text classification tasks where features represent word frequencies, such as TF-IDF vectors (Utama et al., 2025). Advantages: Fast, efficient, and well-suited for short texts and high-dimensional data. Best used for: Text classification, spam detection, sentiment analysis.

Code snippet calling algorithm in python script:

```
from sklearn.naive_bayes import MultinomialNB 'Naive Bayes': MultinomialNB()
```

The following is the equation of the NB algorithm :

$$P(X|H) = P(H|X) P(H) / P(X)$$

3.2 Gradient Boosting

Gradient Boosting is an ensemble method that builds models sequentially. Each new model is trained to reduce the errors of the previous one by minimizing a loss function using gradient descent. It typically uses shallow decision trees as base learners (Utama & Sari, 2026). Advantages: High accuracy, capable of capturing complex patterns in the data. Best used for: Non-linear problems with complex feature interactions.

Code snippet calling algorithm in python script:

```
'Gradient Boosting': GradientBoostingClassifier(n_estimators=100, random_state=42),
```

Annotation:

- `n_estimators=100`: number of trees (boosting iterations).
- `random_state=42`: for consistent results across runs.

The following is the equation of the Gradient Boosting algorithm:

Creating initial constant predicted values:

$$F_0(x) = \log \left[\frac{x}{1-x} \right]$$

For $m = 1$ to M , calculate

$$r_i = y_i - p$$

For $j = 1, \dots, J_m$. Calculate

$$Y_{jm} = \sum (y - p) \quad J_m \quad j=1 \quad \sum p(1 - p)$$

Update model

$$F_m(x) = F_{m-1}(x) + v \sum Y_{jm} J(x \in R_{jm})$$

3.3 AdaBoost (AdaBoostClassifier)

AdaBoost (Adaptive Boosting) is another ensemble method that combines multiple weak learners (usually decision stumps) to form a strong classifier. It increases the weight of misclassified instances so that subsequent models focus more on difficult examples (Freund & Schapire, 1997). Advantages: Simple, effective, and less prone to overfitting. Best used for: Clean datasets and binary or multi-class classification problems.

Code snippet calling algorithm in python script:

```
AdaBoost': AdaBoostClassifier(n_estimators=100, random_state=42)
```

Annotation:

- `n_estimators=100`: the number of weak learners to be merged.
- `random_state=42`: ensures replication of results.

The following is the equation of the AdaBoost algorithm :

$$F_t(x) = \sum_{t=1}^T F_t(x)$$

3.4 Model Evaluation

Model evaluation is the process of measuring the performance of a machine learning model using metrics such as accuracy, precision, recall, and F1-score (Utama et al., 2026a). This evaluation is essential for determining how well the model correctly classifies data. Model evaluation metrics such as accuracy and F1-score provide insights into model performance, enabling comparison between classifiers (Han et al., 2011). Accuracy alone is not sufficient for evaluating sentiment analysis models, especially in the case of imbalanced datasets; precision and recall must be considered (Avinash & Sivasankar, 2019).

The models were evaluated using the following metrics:

- Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

- Precision $\frac{TP}{TP+FP}$
- Recall $\frac{TP}{TP+FN}$
- F1-score $2 * \frac{Precision * Recall}{Precision + Recall}$

An 80/20 stratified train-test split ensured proportional representation of sentiment classes.

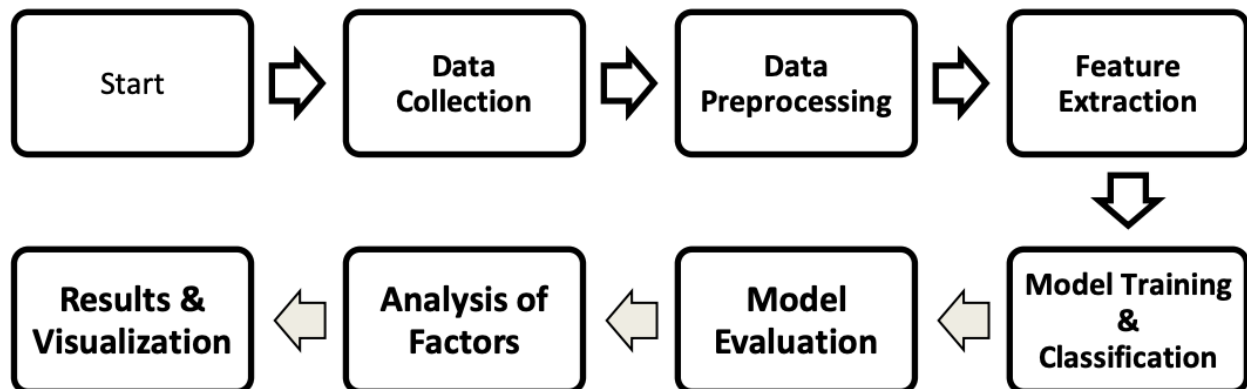


Figure 2. Model Development Flow Chart

The research follows a systematic methodology as illustrated in Figure 2. The process begins with Data Collection, where raw user reviews are acquired, followed by a rigorous Data Preprocessing phase to clean and standardize the text. Subsequently, Feature Extraction is performed to transform the text into a numerical format suitable for analysis. The core of the experiment involves Model Training & Classification using the selected algorithms, which are then subjected to Model Evaluation to measure their performance metrics. Finally, an Analysis of Factors is conducted to interpret the model's behavior, culminating in the Results & Visualization stage to present the comprehensive findings of the sentiment analysis (Utama et al., 2025, 2026a, 2026b; Utama & Sari, 2026).

RESULTS AND DISCUSSION

This section presents the findings obtained from the experimental analysis and provides a detailed interpretation of the performance of the three machine learning models. The primary objective is to evaluate how effectively Naive Bayes, Gradient Boosting, and AdaBoost can classify user sentiments regarding the JKN Mobile application. The results are analyzed based on standard evaluation metrics, followed by a discussion on the implications of these findings within the context of health technology development and user experience.

1. Accuracy and F1-score

The performance of three machine learning algorithms — Naive Bayes, Gradient Boosting, and AdaBoost — was evaluated using Accuracy, Precision, Recall, F1-score (weighted), and 5-Fold Cross-Validation Accuracy. The results are summarized in Table 2.

Table 1. The sample of table format (Center, Cambria, 11)

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-score (Weighted)	CV Accuracy (5-Fold)
Naive Bayes	78%	79%	78%	74%	75.96%
Gradient Boosting	86%	85%	86%	85%	81.63%
AdaBoost	77%	78%	77%	74%	73.74%

Gradient Boosting achieved the highest overall performance with 86% accuracy, an F1-score of 85%, and stable crossvalidation accuracy (81.63%). Naive Bayes achieved 78% accuracy and an F1-score of 74%, demonstrating efficiency on short and sparse text. AdaBoost yielded the lowest overall performance (77% accuracy, 74% F1-score) and struggled particularly with the neutral class.

Impact of class imbalance. The dataset exhibits a strong class imbalance (see Section 4.3). This imbalance affects some metrics: for instance, high overall accuracy can partially reflect the model learning to predict the majority (negative) class. To mitigate misleading conclusions, we emphasize per-class metrics and weighted averages in this study. Notably, Naive Bayes shows extremely high recall for the negative class (0.99), which is consistent with the large negative class proportion and indicates the model is very sensitive to negative signals.

2 Feature Analysis

Feature importance and class-specific indicators were analysed using the trained models and the Naive Bayes class log-probabilities. Key findings:

- Naive Bayes (class logprobabilities):
 - o Negative class top words include: the, not, cant, error, app — reflecting frequent error/complaint tokens and some English stopwords that remain informative in context.
 - o Neutral class top words include: baik, nice, great, login, simple — suggesting descriptive or mixed comments.
 - o Positive class top words include: good, easy, use, mudah, friendly — reflecting satisfaction and usability.
- Gradient Boosting (feature_importances_): Top features included error, thanks, easy, good, mudah, login, showing both appreciation and technical complaint terms are influential.
- AdaBoost (feature_importances_): Top features included error, thanks, good, bisa, mau, again emphasizing technical issues plus appreciation tokens.

Across models, technical/ login/verification-related terms consistently appear as strong predictors of negative sentiment, while words denoting ease, helpfulness, and appreciation predict positive sentiment. The presence of English function words (“the”, “to”, etc.) among top features suggests mixed-language preprocessing leaves some common English tokens that the models still use for discrimination; consider targeted stopword handling in future preprocessing.

3. Sentiment Distribution

The label counts in the dataset are:

- Negative: 635 reviews ($\approx 65.6\%$)
- Positive: 221 reviews ($\approx 22.8\%$)
- Neutral: 112 reviews ($\approx 11.6\%$)

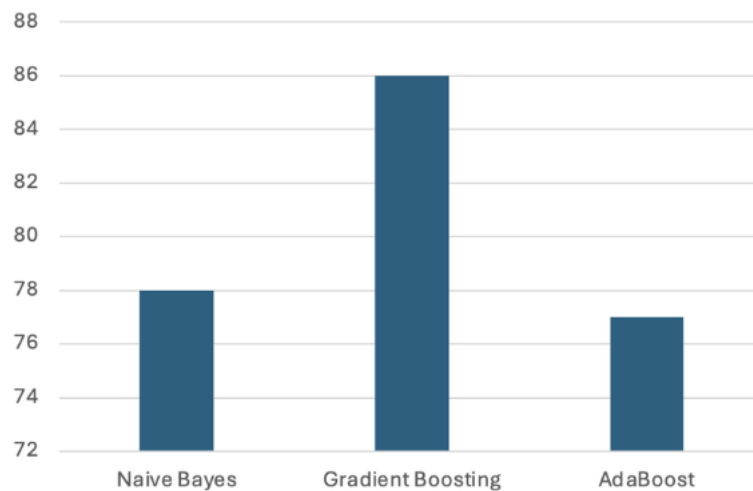
Total = 968 reviews.

This distribution indicates that a majority of reviews express negative sentiment. The dominance of negative reviews helps explain some model behaviors (e.g., high negative recall for Naive Bayes) and underlines the importance of using weighted and perclass metrics rather than relying on accuracy alone.



Figure 3. WordCloud Positif And Negatif Reviews

Interestingly, when the dataset was first collected at 500 reviews, the negative proportion was already about 60%. As more reviews were added (up to 968), the proportion of negative reviews increased further to 65.6%. This consistent trend suggests that the application has genuinely accumulated a large volume of negative feedback. It may also indicate that satisfied users are less motivated to leave reviews, while dissatisfied users are more inclined to express their concerns publicly.

**Figure 4.** Comparison Accuracy of the Three Algorithms

Word Cloud visualizations and the feature lists corroborate the quantitative distribution: negative reviews frequently mention login failures, errors, and waiting times, while positive reviews emphasize ease of use and helpful features. These findings highlight areas where the application could be improved. In the context of public digital services such as healthcare applications, continuous enhancement is important. Addressing recurring technical issues and responding to user complaints would not only improve user satisfaction but also support the broader goal of strengthening public healthcare services in Indonesia.

Based on Figure 4, the classification performance demonstrates significant variation, with Gradient Boosting emerging as the best-performing model achieving 86% Accuracy and an 85% F1-score. Meanwhile, Naive Bayes follows with 78% Accuracy and a 74% F1-score, proving to be efficient for short texts, while AdaBoost records 77% Accuracy and a 74% F1-score despite experiencing difficulties in handling the neutral class optimally.

4. Limitations

Despite the significant findings, this study acknowledges several limitations that should be addressed in future research:

- **Class Imbalance:** The dataset exhibits a notable skew toward negative sentiment (approximately 65.6%). This imbalance may introduce bias during model training, potentially affecting the sensitivity of the classifiers toward positive reviews.
- **Dataset Constraints:** While the 968 reviews provide valuable insights, the sample size remains relatively small. A larger corpus would be required to achieve broader generalization and more robust model validation.
- **Linguistic Complexity:** The presence of code-switching (mixed Indonesian and English) within the reviews introduces noise. Although preprocessing was conducted, the bilingual nature of the data poses challenges for standard monolingual NLP tools.
- **Labeling Subjectivity:** The reliance on human annotators for sentiment labeling may introduce inherent subjectivity and potential inconsistency in the ground-truth data.

- **Preprocessing Noise:** The emergence of English function words among the top features suggests that residual noise remains. Future iterations would benefit from a more refined, bilingual stopwords removal process.

CONCLUSION

This study evaluated Naive Bayes, Gradient Boosting, and AdaBoost for sentiment classification of JKN Mobile app reviews. Gradient Boosting yielded the best overall performance (86% accuracy, 85% F1-score, 81.63% CV accuracy). Naive Bayes provided a lightweight and effective baseline (78% accuracy; very high negative recall), while AdaBoost underperformed relative to the other models. Feature analysis across models indicated that technical issues — particularly login failures, verification problems, and error messages — are primary drivers of negative sentiment, whereas ease of use, speed, and helpfulness fuel positive sentiment. The predominance of negative reviews suggests that addressing these technical problems should be a priority for improving user satisfaction.

Future Work

To strengthen the analysis and model robustness, future work should consider:

- Addressing class imbalance: apply resampling techniques (oversampling minority classes, undersampling the majority, or synthetic methods such as SMOTE/ADASYN), or use classweighting in model training; evaluate with macro-averaged metrics.
- Advanced models and contextual embeddings: experiment with deep learning (LSTM, transformer-based models such as BERT or multilingual BERT) to capture contextual meaning and codeswitching.
- Aspect-based sentiment analysis: identify sentiment on specific app aspects (e.g., login, verification, UI) to provide actionable recommendations.
- Improved preprocessing: refine tokenization, stopwords lists (language-aware), and handling of English tokens to reduce noise.
- Larger and more diverse dataset: collect more reviews across versions/time to improve generalizability and detect temporal trends.
- Threshold tuning and costsensitive learning: tune decision thresholds or use cost-sensitive loss to better handle the operational cost of misclassifying negative feedback.

REFERENCES

- Avinash, C., & Sivasankar, E. (2019). Performance evaluation of machine learning algorithms in sentiment analysis. *Procedia Computer Science*, 165, 682–689. <https://doi.org/10.1016/j.procs.2020.01.054>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Chai, C. P., Shamsuddin, S. M., & Zain, J. M. (2022). Comparison of text preprocessing techniques for sentiment analysis in social media. *Natural Language Engineering*, 28(2), 203–223. <https://doi.org/10.1017/S1351324922000044>
- Fahrezi, A., & Solichin, S. (2024). Sentiment analysis of Mobile JKN app using Naive Bayes. *Proceedings of Seminar Nasional Informatika (SENAFTI)*.
- Freund, Y., & Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139.
- Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Elsevier.
- Harta, Y. J. F., et al. (2024). Aspect-based sentiment analysis of JKN Mobile app. *Jurnal Ilmu Komputer Udayana*, 18(1).
- Jianqiang, Z., & Xiaolin, G. (2018). Comparison research on text preprocessing methods on Twitter sentiment analysis. *Procedia Computer Science*, 127, 17–23. <https://doi.org/10.1016/j.procs.2017.01.123>
- Kharde, V., & Sonawane, P. (2016). Sentiment analysis of Twitter data: A survey of techniques. *International Journal of Computer Applications*, 139(11), 5–15. <https://doi.org/10.5120/ijca2016908625>
-

- Liu, B. (2012). Sentiment analysis and opinion mining. Morgan & Claypool Publishers. <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>
- Miftahusalam, A., et al. (2023). Comparison of Random Forest and Naive Bayes in sentiment analysis of BCA Mobile reviews. *SIPTEK Journal*, 4(1).
- Miftahusalam, A., et al. (2024). Comparison of Random Forest and Naive Bayes in sentiment analysis of Mobile JKN reviews. [Nama jurnal belum tersedia].
- Munir, R., et al. (2021). Data mining techniques for sentiment classification: A comparative analysis. *Procedia Computer Science*, 179, 554–561. <https://doi.org/10.1016/j.procs.2021.01.041>
- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 79–86). <https://doi.org/10.3115/1118693.1118704>
- Utama, D., Fahmi, R., & Pratama, M. R. K. (2025). Performance comparison of 10 machine learning algorithms in sentiment classification on Platform X regarding the government's priority program: Makanan Bergizi Gratis (MBG). *Teknika*, 14(3), 501–508.
- Utama, D., & Sari, D. N. (2026). Multi-horizon LSTM forecasting of Indonesian food commodity prices to support the Free Nutritious Meal (MBG) program. *Teknika*, 15(1), 123–129.
- Utama, D., Sugiyanta, L., Fahmi, R., Falaq, F. I., Josephine, & Putra, M. F. (2026, 28 Juli). Enhancing national cyber resilience: Real-time DDoS detection and mitigation using isolation forest and machine learning techniques. In *2026 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)* (Vol. 8, pp. 137–143). IEEE. <https://ieeexplore.ieee.org/document/116>
- Utama, D., Sugiyanta, L., Fahmi, R., Falaq, F. I., & Sari, D. N. (2026, 22 Juli). Real-time DDoS detection and monitoring dashboard with multi-view statistical visualization: An adaptive approach for Indonesian cyber resilience. In *2026 International Seminar on Intelligent Technology and Its Applications (ISITIA)* (pp. 1000–1005). IEEE.
- Zaenudin, M., et al. (2023). Combining Naive Bayes and logistic regression for sentiment analysis of lending app reviews. *International Journal of Electrical and Computer Engineering*, 12(2), 215–222. <https://doi.org/10.11591/ijes.vol12.i2.pp215-222>