Volume XI Issue 3 Year 2026 | Page 928-939 | ISSN: 2527-9866
Received: ***20-06-2026*** | *Revised:* ***15-07-2026*** | *Accepted:* ***15-08-2026***

Sentiment Analysis of User Reviews for the Locket Widget Application on Google Play Store Using the Naïve Bayes Classifier

Athia Aisy Bakhita[1], Arif Setiawan[2], Eko Darmanto[3]
[1,2,3] Muria Kudus University, Kudus, Central Java, Indonesia, 59327
e-mail: 202153123@std.umk.ac.id[1], arif.setiawan@umk.ac.id[2], eko.darmanto@umk.ac.id[3]
*Correspondence: 202153123@std.umk.ac.id

Abstract: *User reviews on the Google Play Store contain valuable opinions that can be utilized to evaluate application quality, including the Locket Widget application. This study aims to classify user review sentiment into positive, neutral, and negative categories using the Naïve Bayes Classifier algorithm with Term Frequency–Inverse Document Frequency (TF-IDF) weighting. The data were collected through web scraping, resulting in 589 user reviews. The dataset then underwent preprocessing, sentiment labeling, TF-IDF weighting, and sentiment classification. The model was evaluated using an 80:20 stratified train–test split. The results showed that the proposed model achieved an accuracy of 76.27%, with a weighted precision of 0.69, weighted recall of 0.76, and weighted F1-score of 0.72. The findings indicate that the combination of TF-IDF and the Naïve Bayes classifier is effective in classifying positive and negative user review sentiments. However, the model was unable to effectively classify the neutral class, resulting in an F1-score of 0.00 for this category. This finding indicates that further improvements are needed to enhance the model's ability to distinguish neutral sentiment from positive and negative classes.*

Keywords: *Sentiment Analysis, Google Play Store, Locket Widget, Naïve Bayes, TF-IDF*

1. Introduction

Google Play Store function not only as an application distribution platform but also as a medium for users to express their experiences, criticisms, and suggestions through the review feature. These user reviews provide valuable information that can be utilized to evaluate application quality and identify aspects requiring further improvement[1]. One of the applications that has attracted considerable user attention is Locket Widget, a widget-based application that enables users to share photos directly through their device's home screen, creating a more personal sharing experience than conventional social media platforms[2]. In addition to providing ratings, users can also express their opinions and share their experiences with the application through the review feature available on the Google Play Store[3].

Based on observations on the Google Play Store, Locket Widget has received more than 364,000 user reviews. The increasing number of reviews has generated a substantial amount of textual data with diverse characteristics, making manual analysis inefficient and time-consuming. At the same time, these reviews provide valuable insights into user satisfaction, feature quality, and various issues encountered during application use. Therefore, an automated approach is needed to process user reviews efficiently, enabling the extracted information to be utilized as a basis for application evaluation and continuous improvement[4].

One approach that can be applied is sentiment analysis, a text mining technique used to identify the tendency of user opinions based on textual content[5]. In this study, sentiment classification was performed using the Naïve Bayes Classifier algorithm combined with Term Frequency–Inverse Document Frequency (TF-IDF) weighting. This algorithm was selected because it has demonstrated competitive performance in text classification while maintaining relatively low computational complexity. In addition, Naïve Bayes Classifier is well suited for sparse feature

representations generated by TF-IDF, making it an appropriate approach for sentiment classification tasks[6],[7]. Therefore, this study aims to classify user reviews of the Locket Widget application into positive, neutral, and negative sentiment categories. The findings are expected to provide insights into user opinion trends that can serve as a reference for application evaluation and future development [8].

Previous studies have demonstrated that various machine learning algorithms, including Naïve Bayes Classifier, Support Vector Machine, Random Forest, and deep learning approaches, have been successfully applied to sentiment analysis across different application domains[9], [10], [11],[12],[13]. However, the reported performance varies considerably depending on the characteristics of the dataset, preprocessing techniques, and class distribution[9], [10], [11]. Most existing studies have focused on e-commerce platforms, financial services, transportation applications, or social media, while sentiment analysis on widget-based social applications remains limited[10],[11],[12],[13]. Furthermore, previous studies primarily emphasize classification accuracy, with limited discussion regarding the challenges posed by class imbalance and the performance of individual sentiment categories. Therefore, further evaluation is required to examine the effectiveness of the Naïve Bayes Classifier algorithm on Locket Widget user reviews, which exhibit different user interaction patterns and sentiment characteristics [14].

Based on these research gaps, this study evaluates the performance of the Naïve Bayes Classifier algorithm combined with TF-IDF weighting for sentiment classification of Locket Widget user reviews collected from the Google Play Store. This study contributes by providing an evaluation of Naïve Bayes Classifier performance on a widget-based social application with distinct user interaction characteristics, as well as offering insights into the distribution of user sentiment that may support future sentiment analysis research and application improvement [15].

2. Literature Review

Sholeha et al[9] investigated sentiment analysis on user comments regarding three online travel agencies in Indonesia using the Naïve Bayes Classifier and K-Nearest Neighbor (KNN) algorithms. The Facebook comments were first processed through several preprocessing stages before being classified and evaluated using 10-fold cross-validation. The experimental results showed that both algorithms achieved a highest accuracy of 52.35%, while KNN demonstrated slightly better average performance than Naïve Bayes. These findings indicate that both algorithms are applicable to sentiment classification, although their performance is strongly influenced by the characteristics of the dataset.

A comparative study was also conducted by Putri and Rozi[10] who analyzed user reviews of the Telegram application collected from the Google Play Store. The study applied TF-IDF weighting before performing sentiment classification using Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM). The evaluation results revealed that SVM achieved the highest accuracy of 89.73%, followed by Logistic Regression with 87.49%, while Naïve Bayes Classifier obtained an accuracy of 75.61%. These results suggest that the effectiveness of each algorithm depends on the characteristics of the textual data.

Similarly Indriyani et al[11] compared the performance of Naïve Bayes Classifier and SVM in sentiment analysis of TikTok user reviews. After applying preprocessing and TF-IDF weighting, SVM achieved an accuracy of 84%, whereas Naïve Bayes Classifier reached 79%. Although Naïve Bayes Classifier did not produce the highest accuracy, it remained competitive while offering a simpler and more computationally efficient classification process.

In addition to probabilistic approaches, other machine learning methods have also been explored for sentiment classification. Dasilva et al[12] analyzed tourist reviews of Bali beach destinations using the Random Forest algorithm combined with TF-IDF feature extraction and achieved the best accuracy of 94.7% on a 75:25 training–testing data split. Meanwhile, Alghifari et al[13] utilized the Bidirectional Long Short-Term Memory (BiLSTM) model to classify sentiment in Grab Indonesia user reviews, achieving an accuracy of 91%. These studies demonstrate that both conventional machine learning and deep learning approaches can produce promising results, although their performance largely depends on the characteristics of the dataset and the preprocessing techniques applied.

Based on these previous studies, the selection of a classification algorithm should consider not only predictive performance but also model complexity and data characteristics. Therefore, this study employs the Naïve Bayes Classifier algorithm combined with TF-IDF weighting, as it provides a simple, efficient, and suitable approach for text classification. Moreover, studies focusing on sentiment analysis of Locket Widget user reviews on the Google Play Store remain limited. Consequently, this research is expected to contribute to the development of sentiment analysis studies using Locket Widget as the research object.

3. Methods

This study employed a quantitative approach to analyze the sentiment of Locket Widget user reviews collected from the Google Play Store using the Naïve Bayes Classifier algorithm. The research consisted of several stages, including data collection, sentiment labeling, text preprocessing, TF-IDF weighting, sentiment classification, and model evaluation. The overall research workflow is illustrated in Figure 1.

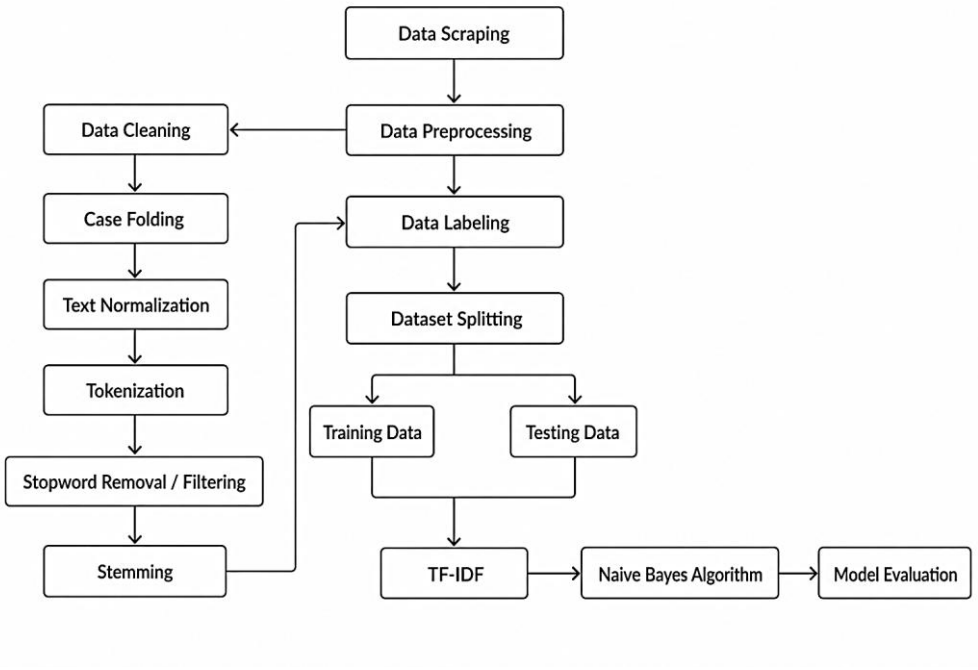

Figure 1. Research Flowchart

A. Data Collection

The data were collected through a web scraping process from the Google Play Store. The scraping process was carried out using the Python programming language with the Google Play Scraper library. All collected reviews were then stored in CSV format and used as the dataset for the sentiment analysis process.

The data retrieval process was configured to collect up to 3,000 user reviews posted between 2020 and 2026. However, based on the specified retrieval criteria, only 589 reviews were successfully retrieved and used as the dataset for this study.

B. Data Preprocessing

The preprocessing stage was performed to clean and transform the textual data into a more structured format, thereby improving data quality before the classification process[11]. The preprocessing steps included:

1) Cleaning: Removing unnecessary elements such as punctuation marks, numbers, URLs, emojis, and special characters.
2) Case Folding: Converting all text into lowercase to ensure consistency.
3) Normalization: Transforming non-standard words into their standard forms. Normalization was conducted using a predefined Indonesian slang-word dictionary to convert informal and abbreviated words into their standard forms. This process was intended to improve text consistency before feature extraction.
4) Tokenization: Splitting text into individual words or tokens.
5) Stopword Removal*:* Eliminating common words that do not contribute significant meaning to the analysis.
6) Stemming: Reducing words to their root forms to unify different word variations. Stemming was performed using the Sastrawi stemming library to reduce inflected words to their root forms. This step helps minimize lexical variations and improve the effectiveness of TF-IDF feature extraction.

C. Sentiment Labeling

Sentiment labeling was performed based on the rating assigned by users to each review. Reviews with ratings of 1 and 2 were labeled as negative, a rating of 3 was labeled as neutral, while ratings of 4 and 5 were labeled as positive. This rating-based labeling approach was adopted because user ratings generally represent the overall opinion toward the application and have been widely applied in previous sentiment analysis studies involving online reviews[16]. The assigned labels were used as the ground truth for model training and evaluation. However, no manual validation was performed to verify the correspondence between the assigned ratings and the sentiment expressed in individual review texts.

D. Data Splitting

After the preprocessing stage, the dataset was divided into training and testing sets using an 80:20 ratio. The training set was used to develop the classification model, whereas the testing set was used to evaluate its performance[17]. The data partitioning process was carried out using the (`train_test_split`) function with a fixed (`random_state`) of 42 to ensure reproducibility. Furthermore, stratified sampling based on the sentiment labels was applied to preserve the class distribution in both the training and testing datasets.

E. Feature Weighting Using TF-IDF

Feature weighting was performed using the Term Frequency–Inverse Document Frequency (TF-IDF) method. This technique assigns a weight to each word based on its frequency within a document and its distribution across the entire document collection[18]. Consequently, words that appear frequently in a particular document but rarely in other documents receive higher weights, allowing them to contribute more significantly to the sentiment classification process[19]. The TF-IDF formula is presented as follows:

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

The resulting TF-IDF weights were then used as feature representations and served as input for the Naïve Bayes Classifier algorithm during the sentiment classification process. TF-IDF feature extraction was implemented using unigram and bigram features (`ngram_range=1,2`)) with a maximum of 100,000 features (`max_features=100000`). In addition, sublinear term frequency scaling (`sublinear_tf=True`) was applied to reduce the influence of highly frequent terms. No additional stop-word removal was performed during vectorization because this process had already been completed during the preprocessing stage.

F. Sentiment Classification Using Naïve Bayes Classifier

The sentiment classification process was performed using the Naïve Bayes Classifier algorithm, specifically the Multinomial Naïve Bayes Classifier variant, which is well suited for text classification tasks involving TF-IDF feature representations. This algorithm was selected because of its computational efficiency, simple implementation, and widespread use in text-based classification problems[20],[21]. The classification model was trained using the TF-IDF feature vectors derived from the training dataset and subsequently applied to predict the sentiment categories of the testing dataset[1]. To obtain the optimal model, hyperparameter tuning was conducted using (`GridSearchCV`) with five-fold-cross-validation (`cv=5`), and the model with the highest accuracy score was selected for the final classification. The Naïve Bayes Classification formula is presented as follows:

$$P(Y|X) = \frac{P(X|Y).P(Y)}{P(X)} \quad (2)$$

where:

P(Y|X) = the probability that data X belongs to class Y,

P(X|Y) = the probability of observing feature X given class Y,

P(Y) = the prior probability of class Y,

P(X) = the probability of observing data X

The probability values for all classes were then compared, and the class with the highest probability was selected as the final sentiment classification result.

G. Model Evaluation

Model performance was evaluated using a confusion matrix, which compares the predicted labels with the actual labels. Based on this comparison, the model performance was assessed using four evaluation metrics, namely accuracy, precision, recall, and F1-score. These metrics were employed to measure the effectiveness of the proposed classification model[22].

Accuracy was used to measure the overall proportion of correctly classified instances, as presented in Equation (3).

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \quad (3)$$

Precision was then used to measure the proportion of correctly predicted instances for each sentiment class, as presented in Equation (4).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (4)$$

Recall was used to evaluate the model's ability to identify all actual instances belonging to each sentiment class, as presented in Equation (5).

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

Meanwhile, *F1-score* was employed to provide a balanced evaluation by combining precision and recall into a single metric, as calculated using Equation (6).

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

4. Result and Discussion

A. Data Collection

The research data were collected through a web scraping process from the Google Play Store using the Google Play Scraper library on the Google Colab platform. At the time of data collection, the Locket Widget application had accumulated approximately 364,000 user reviews. The data retrieval process was configured to collect up to 3,000 reviews posted between 2020 and 2026. However, based on the specified retrieval criteria, only 589 user reviews were successfully retrieved and used as the dataset for further analysis.

userName	score	at	content	probable_user_id
Wullan Bessie	5	2025-12-31T15:45:28	sukaaa bgttttt	name:wullan_bessie
Rahma Wnn	4	2025-12-29T08:26:50	kadang apk nya lag padahal hp ku gda masalah	name:rahma_wnn
Sabrina Aleesya	1	2025-12-26T12:23:25	KEBALIK HASIL FOTONYA	name:sabrina_aleesya
Marsya Aulia	1	2025-12-25T05:40:09	gk bisa masuk	name:marsya_aulia
Ulma Jaya	5	2025-12-21T06:03:42	baguuss	name:ulma_jaya

Figure 2. Data Collection Results

B. Data Preprocessing

The preprocessing stage was performed to produce structured textual data that were suitable for feature extraction. During this stage, each review underwent several preprocessing steps to standardize the text format, remove unnecessary elements, and reduce words to their root forms. As a result, the processed data became more consistent, minimizing textual noise and improving the learning process of the classification model.

content	cleaning	case_folding	normalisasi	tokenize	stopword removal	steming_data
sukaaa bgttttt	sukaaa bgttttt	sukaaa bgttttt	suka banget	[suka, banget]	[suka, banget]	suka banget
kadang apk nya lag padahal hp ku gda masalah	kadang apk nya lag padahal hp ku gda masalah	kadang apk nya lag padahal hp ku gda masalah	kadang aplikasi nya lag padahal handphone ku t...	[kadang, aplikasi, nya, lag, padahal, handphon...	[kadang, aplikasi, lag, handphone, tidak, masa...	kadang aplikasi lag handphone tidak masalah
KEBALIK HASIL FOTONYA	KEBALIK HASIL FOTONYA	kebalik hasil fotonya	kebalik hasil fotonya	[kebalik, hasil, fotonya]	[kebalik, hasil, fotonya]	balik hasil foto
gk bisa masuk	gk bisa masuk	gk bisa masuk	tidak bisa masuk	[tidak, bisa, masuk]	[tidak, bisa, masuk]	tidak bisa masuk
baguuss	baguuss	baguuss	bagus	[bagus]	[bagus]	bagus

Figure 3. Data Preprocessing Results

C. Sentiment Labeling

Sentiment labels were assigned based on the ratings provided by users. Reviews with ratings of 1–2 were categorized as negative, a rating of 3 was categorized as neutral, while ratings of 4–5 were categorized as positive. The labeling results showed that, of the 589 reviews analyzed, 249 reviews (42.28%) were classified as negative, 208 reviews (35.31%) as neutral, and 132 reviews (22.41%) as positive, as illustrated in Figure 4.

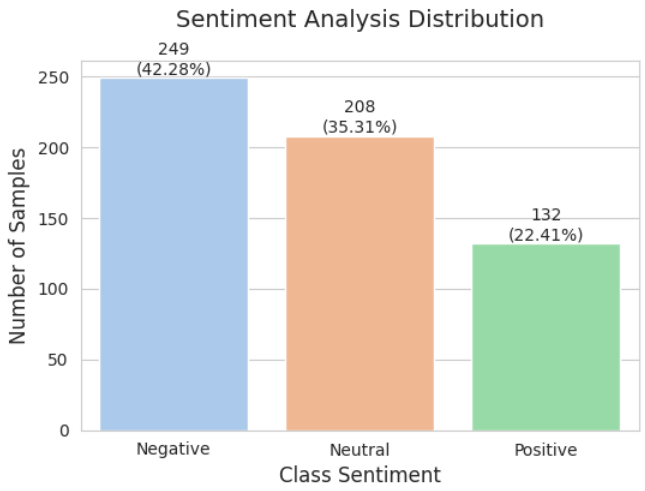

Figure 4. Sentiment Analysis Distribution

The distribution indicates that negative reviews constitute the largest proportion of the dataset compared with the other sentiment categories. This imbalance in class distribution should be considered during model development to ensure that the classification model can adequately represent all sentiment classes.

D. Data Splitting

After the sentiment labeling process, the dataset was divided into 471 training samples (79.97%) and 118 testing samples (20.03%), as presented in Figure 5. The training dataset was used to train the classification model by learning the characteristics of each sentiment category, whereas the testing dataset was used to evaluate the model's ability to classify previously unseen data. An 80:20 data split was adopted to achieve an appropriate balance between model training and performance evaluation.

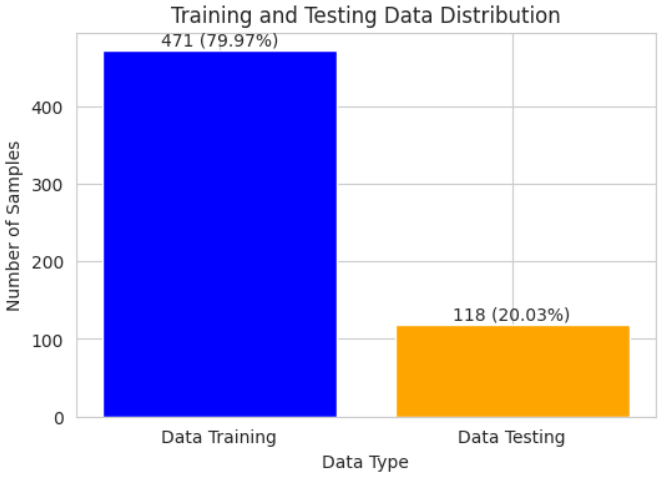

Figure 5. Training and Testing Data Distribution

E. Feature Weighting Using TF-IDF

Text representation was performed using the Term Frequency–Inverse Document Frequency (TF-IDF) weighting method. The results showed that the word “tidak” (not) obtained the highest TF-IDF weight 0.050, followed by “bagus” (good) with a weight of 0.042 and “bisa” (can) with a weight of 0.045. In addition to individual terms, the phrase “tidak bisa” (cannot) also received a relatively high weight. These findings indicate that user reviews were largely dominated by discussions of application-related issues, although positive opinions were also identified.

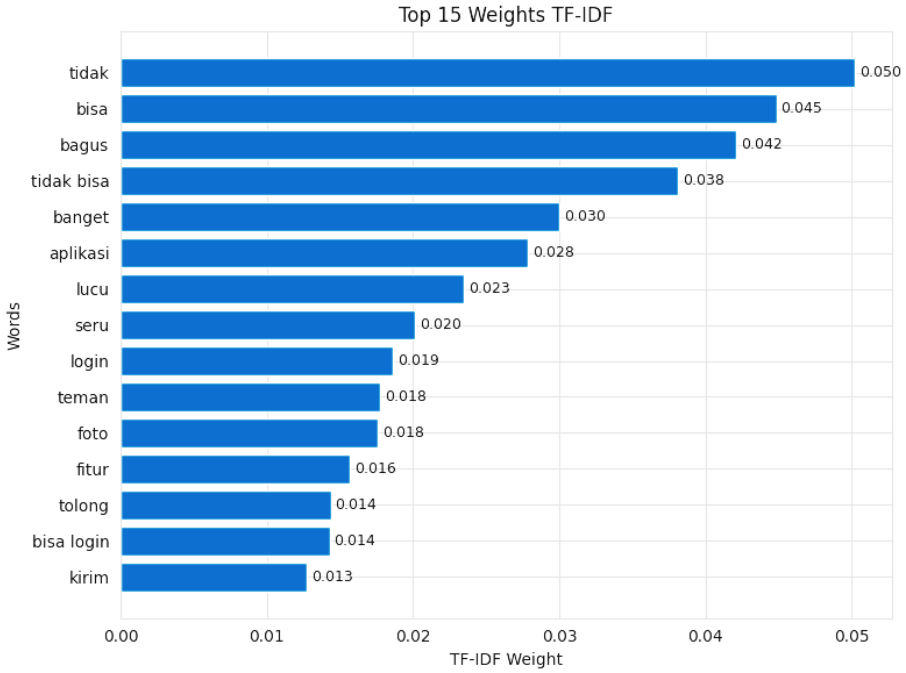

Figure 6. Top 15 TF-IDF Weights

F. Naïve Bayes Classification and Model Evaluation

Sentiment classification was carried out using the Naïve Bayes Classifier algorithm with TF-IDF feature weighting. The proposed model achieved an accuracy of 76.27%, with weighted precision, recall, and F1-score values of 0.69, 0.76, and 0.72, respectively.

```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.79      0.74      0.76        46
     Neutral       0.00      0.00      0.00        12
    Positive       0.75      0.93      0.83        60

    accuracy                           0.76       118
   macro avg       0.51      0.56      0.53       118
weighted avg       0.69      0.76      0.72       118

==================================
Accuracy (Optimized Naive Bayes Classifier): 0.7627
==================================
```

Figure 7. Classification Report Naïve Bayes Classifier

According to the classification report, the positive class achieved the best performance with an F1-score of 0.83, followed by the negative class with 0.76. In contrast, the neutral class obtained an F1-score of 0.00, indicating that the model failed to correctly classify any neutral reviews. Although the overall classification performance was acceptable, the model showed considerable difficulty in recognizing the neutral sentiment category. The poor performance of the neutral class can be attributed to several factors. First, the number of neutral reviews was considerably smaller than the number of positive reviews, resulting in an imbalanced class distribution that limited the model's ability to learn representative patterns for neutral sentiment. Second, neutral reviews often contain ambiguous or mixed expressions that share vocabulary with both positive and negative reviews. Consequently, the classifier tended to assign neutral reviews to the positive or negative class instead of identifying them as neutral. This limitation suggests that further improvements are needed, particularly in handling class imbalance and improving the representation of neutral sentiment in the dataset.

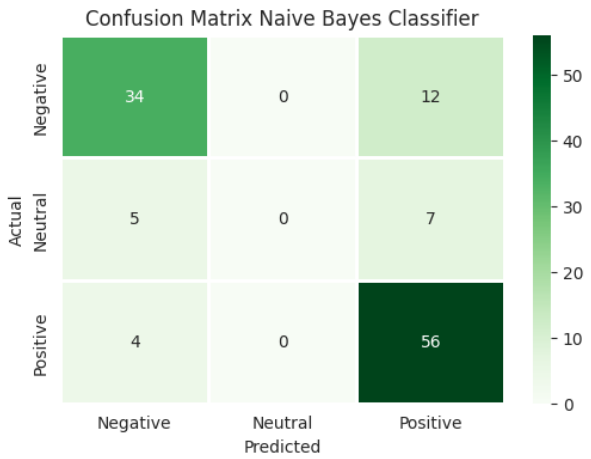

Figure 8. Confusion Matrix Naïve Bayes Classifier

Figure 8 illustrates the confusion matrix generated by the Naïve Bayes Classifier. The model correctly classified 34 negative reviews and 56 positive reviews, while none of the 12 neutral reviews were correctly classified. Among the three sentiment categories, the positive class achieved the highest number of correct predictions, indicating that reviews containing clear positive expressions were easier for the model to identify. In contrast, the model failed to recognize the neutral class, resulting in no correct predictions for this category.

The confusion matrix further shows that 5 neutral reviews were misclassified as negative and 7 neutral reviews were misclassified as positive. These findings indicate that the classifier tended to assign neutral reviews to the positive or negative class rather than identifying them as neutral. This limitation is reflected in the neutral class obtaining precision, recall, and F1-score values of 0.00. One possible explanation is that neutral reviews often contain ambiguous or mixed expressions that overlap with both positive and negative sentiment. Furthermore, the relatively smaller number of neutral reviews compared with the positive class may have limited the model's ability to learn representative patterns for this sentiment category. Therefore, future studies should consider improving the handling of class imbalance and exploring more effective feature representation or classification methods to enhance the recognition of neutral sentiment.

Table 1. Examples of Incorrectly Classified Reviews

	Review	Actual	Predicted
1	Fitur kurang android tidak bisa balas chat (The android version lacks a chat reply feature)	Neutral	Positive
2	Gakbisa kirim pesan (Unable to send messages)	Neutral	Negative
3	Tidak bisa login internet bagus (Unable to log in despite having a stable internet connection)	Neutral	Negative

Table 1 presents several examples of incorrectly classified reviews. Most misclassified instances belong to the neutral class, where the reviews contain both positive and negative expressions within the same sentence. Such lexical ambiguity makes it difficult for the Naïve Bayes classifier to assign the correct sentiment category. Compared with previous studies that applied Support Vector Machine (SVM), Random Forest, and BiLSTM to sentiment classification tasks, the performance achieved in this study is relatively lower[10], [11],[12]. Nevertheless, the Naïve Bayes Classifier algorithm remains an appropriate choice for text classification because it offers a simple implementation, computational efficiency, and competitive performance. It should also be noted that differences in dataset characteristics, preprocessing techniques, and sentiment distribution can substantially influence classification results across different studies.

5. Conclusions

The sentiment classification model using the Naïve Bayes Classifier algorithm with TF-IDF feature weighting achieved an accuracy of 76.27%, with weighted precision, recall, and F1-score values of 0.69, 0.76, and 0.72, respectively. While the model demonstrated good performance in classifying positive and negative sentiments, it failed to correctly classify the neutral class, resulting in an F1-score of 0.00 for this category. This limitation may be attributed to the use of rating-based sentiment labeling without manual validation, the relatively small number of neutral reviews, and the ambiguous characteristics of neutral expressions. Therefore, future studies are recommended to improve the sentiment labeling process, address class imbalance, and compare the proposed approach with other classification algorithms to enhance the classification performance, particularly for neutral sentiment.

References

[1] A. Nurian, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naïve Bayes," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 3s1, pp. 829–835, 2023, doi: 10.23960/jitet.v11i3s1.3348.

[2] A. Kamal, "Inovasi Media Digital dalam Pembelajaran Pendidikan Agama Islam," *Sinergi : Jurnal Ilmiah Multidisiplin*, vol. 1, no. 1, pp. 1–11, Feb. 2025, doi: 10.66914/88ft4r24.

[3] A. H. Kamilah, D. Sugiana, and A. Setiaman, "Motif Penggunaan Media Sosial Locket Widget Di Kalangan Generasi Z," pp. 521–535, 2026, doi: https://doi.org/10.61104/alz.v4i3.5330.

[4] M. K. Khoirul Insan, U. Hayati, and O. Nurdiawan, "ANALISIS SENTIMEN APLIKASI BRIMO PADA ULASAN PENGGUNA DI GOOGLE PLAY MENGGUNAKAN ALGORITMA NAIVE BAYES," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, pp. 478–483, Mar. 2023, doi: 10.36040/jati.v7i1.6373.

[5] K. Kevin, M. Enjeli, and A. Wijaya, "Analisis Sentimen Pengunaaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes," *Jurnal Ilmiah Computer Science*, vol. 2, no. 2, pp. 89–98, Jan. 2024, doi: 10.58602/jics.v2i2.24.

[6] B. Ramadhani and R. R. Suryono, "Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse," vol. 8, no. April, pp. 714–725, 2024, doi: 10.30865/mib.v8i2.7458.

[7] A. R. Damanik and G. W. Nurcahyo, "Jurnal Sistim Informasi dan Teknologi Prediksi Tingkat Kepuasan dalam Pembelajaran Daring Menggunakan Algoritma Naïve Bayes," vol. 3, pp. 88–94, 2021, doi: 10.37034/jsisfotek.v3i3.49.

[8] M. I. Bimasena, I Kadek Yogi Prayoga, I Gusti Agung Putu Mahendra, and Purnama Sidik, "Benchmarking IndoBERT and Multilingual BERT for Indonesian Financial News Sentiment Classification," *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 9, no. 1, pp. 164–176, Jun. 2026, doi: 10.36378/jtos.v9i1.5583.

[9] E. W. Sholeha, S. Yunita, R. Hammad, V. C. Hardita, and K. Kaharuddin, "Analisis Sentimen Pada Agen Perjalanan Online Menggunakan Naïve Bayes dan K-Nearest Neighbor," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 3, no. 4, pp. 203–208, 2022, doi: 10.35746/jtim.v3i4.178.

[10] A. S. Putri and A. F. Rozi, "Analisis Sentimen Kepuasan Pengguna Aplikasi Telegram pada Google Play Store Menggunakan Metode Naïve Bayes, Logistic Regression, dan SVM," *Jurnal Inovtek Polbeng - Seri Informatika*, vol. 10, no. 2, pp. 857–867, 2025, doi: https://doi.org/10.35314/r6cyb589.

[11] F. A. Indriyani, A. Fauzi, and S. Faisal, "Analisis sentimen aplikasi tiktok menggunakan algoritma naïve bayes dan support vector machine Tiktok application sentiment analysis using naïve bayes algorithm and support vector machine," vol. 10, pp. 176–184, 2023, doi: 10.37373/tekno.v10i2.419.

[12] E. F. Yohana Dasilva, E. H. Susanto, and Y. A. Pranoto, "ANALISIS SENTIMEN ULASAN WISATAWAN TERHADAP DESTINASI WISATA PANTAI BALI DENGAN MENGGUNAKAN METODE RANDOM FOREST," *Jurnal Informatika Teknologi dan Sains (Jinteks)*, vol. 7, no. 4, pp. 2113–2120, Dec. 2025, doi: 10.51401/jinteks.v7i4.6901.

[13] D. R. Alghifari, M. Edi, and L. Firmansyah, "Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 12, no. 2, pp. 89–99, Sep. 2022, doi: 10.34010/jamika.v12i2.7764.

[14] F. Profesio Putra, G. Agung, P. Mahendra, A. Tedyyana, and M. Noor, "IMPROVING FAQ RETRIEVAL FOR ACADEMIC REGULATIONS USING SEMANTIC EMBEDDINGS AND LLM QUESTION AUGMENTATION."

[15] A. Tedyyana, I. G. Agung Putu Mahendra, F. Ratnawati, and O. W Purbo, "Modelling the Hatching Success of Sea Turtle Eggs Using Long Short-Term Memory (LSTM) for Conservation Oriented Ecotourism," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 16, no. 2, pp. 162–176, Nov. 2025, doi: 10.31849/digitalzone.v16i2.29045.

[16] Christ Mario and Ryan Randy Suryono, "Public Sentiment Analysis on Dirty Vote Movie on YouTube using Random Forest and Naïve Bayes," *INOVTEK Polbeng - Seri Informatika*, vol. 10, no. 1, pp. 111–122, Mar. 2025, doi: 10.35314/ev9j2g33.

[17] A. Perdana, A. Hermawan, and D. Avianto, "Analisis Sentimen Terhadap Isu Penundaan Pemilu di Twitter Menggunakan Naive Bayes Clasifier," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 2, pp. 195–200, 2022, doi: 10.32736/sisfokom.v11i2.1412.

[18] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, "Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 375–384, 2024, doi: 10.57152/malcom.v4i2.1206.

[19] H. Zhou, "Research of Text Classification Based on TF-IDF and CNN-LSTM," *J. Phys. Conf. Ser.*, vol. 2171, no. 1, 2022, doi: 10.1088/1742-6596/2171/1/012021.

[20] Gilbert, Syariful Alam, and M. Imam Sulistyo, "Analisis Sentimen Berdasarkan Ulasan Pengguna Aplikasi Mypertamina Pada Google Playstore Menggunakan Metode Naïve Bayes," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 2, no. 3, pp. 100–108, 2023, doi: 10.55123/storage.v2i3.2333.

[21] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, 2021, doi: 10.1186/s13634-021-00742-6.

[22] C. Cahyaningtyas, Y. Nataliani, and I. R. Widiasari, "Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE," *AITI*, vol. 18, no. 2, pp. 173–184, Nov. 2021, doi: 10.24246/aiti.v18i2.173-184.