**Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences** **E-ISSN 3047-8278**
**Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16**
**DOI** : https://doi.org/10.66341/fusion.v3i1.342
**WEB** : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

# A Comparative Analysis of Naïve Bayes and Random Forest Algorithms for Sentiment Classification of Akulaku User Reviews

**Nur Ferdiansyah[1*], Ariel Mutia Salsabila[2], Putri Wiji Lestari[3]**

[1,2,3]Program Studi Informatika, Universitas Nurul Huda

[1*]nurferdiansyah2010@gmail.com, [2]arielmutia430@gmail.com, [3]putriwijilestari245@gmail.com

***Abstract***

*Abstract User reviews of the Akulaku application on Google Play Store contain important information regarding user satisfaction and complaints. This study compares the performance of Naïve Bayes and Random Forest algorithms in classifying sentiment into three classes: positive, negative, and neutral. A total of 2,000 Indonesian-language reviews were collected using web scraping techniques. Data were processed through case folding, cleaning, normalization, stopword removal, tokenizing, and stemming. Labeling was based on user ratings. After preprocessing, 1,953 data points remained with an imbalanced distribution; SMOTE was applied to balance each class to 1,114 samples. TF-IDF was used for feature weighting with an 80:20 train-test split. Results showed Random Forest achieved 83% accuracy, while Naïve Bayes reached 80%. However, Naïve Bayes outperformed in precision, recall, and F1-score with macro averages of 0.59, 0.70, and 0.60, compared to Random Forest at 0.52, 0.55, and 0.54. Based on these results, the choice of the best algorithm depends on the specific needs. If the priority is overall accuracy, Random Forest is more recommended however, if the priority is balanced performance across sentiment classes, Naïve Bayes performs better.*

***Keyword :*** *Sentiment Analysis, Naïve Bayes, Random Forest*

## 1. INTRODUCTION

The rapid advancement of digital technology has transformed nearly every aspect of daily life, including financial services. One notable example is the emergence of digital financial applications that enable users to conduct transactions, apply for loans, and pay installments without visiting a bank. This development has been driven by the increasing number of internet and smartphone users in Indonesia, leading to the rapid growth of financial technology (fintech) as an essential component of the country's economic ecosystem [1].

One of the most popular fintech applications in Indonesia is Akulaku. The application offers various financial services, including installment payments without credit cards, cash loans, and digital banking services. Since Akulaku is available on the Google Play Store, users can directly provide ratings and reviews regarding their experiences with the application. These reviews contain valuable information, as users express their satisfaction, complaints, and suggestions about the services provided. However, the large volume of user reviews makes manual analysis inefficient and time-consuming.

Sentiment analysis addresses this challenge by automatically identifying and classifying opinions expressed in textual data. It is a computer-based technique used to determine whether a piece of text conveys a positive, negative, or neutral sentiment [2]. Sentiment analysis has been widely applied to analyze reviews of various applications on the Google Play Store, including financial, social media, and public service applications [3]. By utilizing sentiment analysis, application developers can efficiently understand users' perceptions and identify areas requiring improvement without manually reviewing thousands of comments.

CC Attribution-ShareAlike 4.0 License.

Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences E-ISSN 3047-8278
Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16
DOI : https://doi.org/10.66341/fusion.v3i1.342
WEB : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

Various machine learning algorithms have been employed for sentiment classification, among which Naïve Bayes and Random Forest are the most frequently compared. Naïve Bayes is a probabilistic classifier known for its computational efficiency and effectiveness in text classification tasks [3]. In contrast, Random Forest is an ensemble learning algorithm that combines multiple decision trees to produce more accurate and robust predictions, particularly when dealing with complex datasets [4]. Although both algorithms have been extensively studied, previous research has reported inconsistent findings. Several studies have shown that Random Forest outperforms Naïve Bayes, such as in the sentiment analysis of the LinkAja application, achieving an accuracy of 82% compared to 79% for Naïve Bayes [4], and in the Indonesian Digital Identity (IKD) application, where Random Forest achieved 93% accuracy compared to 89% for Naïve Bayes [5]. Conversely, other studies have reported superior performance of Naïve Bayes. For example, sentiment analysis of BRImo application reviews achieved an accuracy of 87% compared to 86% for Random Forest (Agustin et al., 2026), while Naïve Bayes obtained an accuracy of 90.52% in Minecraft user reviews [3] and outperformed Random Forest in the evaluation of PKKMB activities with an accuracy of 81.4% compared to 79.78% [6]. These inconsistent findings suggest that the performance of classification algorithms depends heavily on the characteristics of the dataset and application domain.

Despite the growing body of research on sentiment analysis, few studies have specifically compared the performance of Naïve Bayes and Random Forest on user reviews of fintech applications such as Akulaku available on the Google Play Store. User reviews of online lending applications exhibit distinctive characteristics, often containing emotionally charged expressions related to financial experiences, including complaints about interest rates, loan disbursement delays, and technical issues within the application. Such characteristics may influence the performance of different classification algorithms [6], [7].

Therefore, this study aims to compare the performance of the Naïve Bayes and Random Forest algorithms in classifying sentiment from Akulaku user reviews on the Google Play Store into three sentiment categories: positive, negative, and neutral. To address the class imbalance problem in the dataset, the Synthetic Minority Oversampling Technique (SMOTE) is employed. The review data are processed through text preprocessing and feature extraction using the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme. The classification performance is evaluated using accuracy, precision, recall, and F1-score metrics. The findings of this study are expected to identify the most suitable algorithm for sentiment classification in fintech loan applications and provide valuable insights for Akulaku developers to improve the quality of their services.

## 2. METHODS

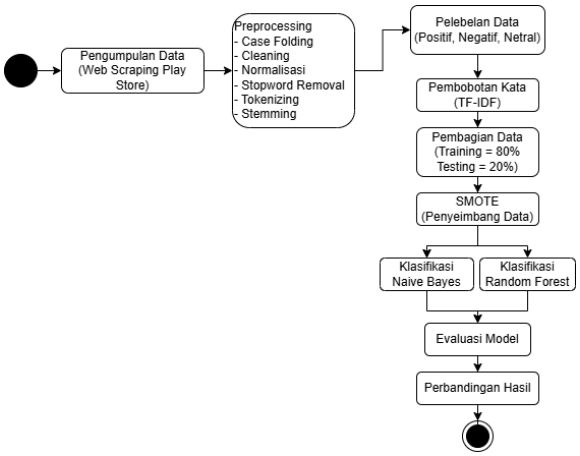


**Figure 1.** Research Workflow

CC Attribution-ShareAlike 4.0 License.

Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences E-ISSN 3047-8278
Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16
DOI : https://doi.org/10.66341/fusion.v3i1.342
WEB : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

Figure 1 illustrates the research workflow, comprising several stages: data collection, preprocessing, labeling, TF-IDF weighting, data splitting, SMOTE, classification, model evaluation, and result comparison.

## Data Collection

Data was collected from Akulaku app user reviews on the Google Play Store using web scraping techniques with Python and the `google-play-scraper` library. This process yielded 2,000 Indonesian-language reviews—including attributes such as username, review content, rating, and date—which were subsequently stored in CSV format [8].

## Preprocessing

The preprocessing stage aims to clean and standardize the text prior to analysis [7]. This stage consists of (1) case folding to convert all text to lowercase for uniformity [9], (2) cleaning to remove unnecessary characters such as punctuation, symbols, numbers, URLs, and emojis [10], (3) normalization to convert non-standard words and abbreviations into their standard forms using a manual normalization dictionary (e.g., changing "gk" to "tidak") [8], (4) stopword removal to eliminate low-semantic-value words like "dan," "di," and "ke" using the PySastrawi library [11], (5) tokenization to break sentences down into individual word units [12], and (6) stemming to reduce words with affixes to their root forms using the Sastrawi algorithm (e.g., changing "memudahkan" to "mudah") [13].

## Labeling

Following preprocessing, labeling was performed based on user ratings: ratings of 4 and 5 were categorized as positive sentiment, ratings of 1 and 2 as negative sentiment, and a rating of 3 as neutral sentiment [7]. The preprocessed text was then converted into a numerical representation using TF-IDF (Term Frequency-Inverse Document Frequency)—where words appearing frequently in a specific review but rarely in others are assigned higher weights [6]—using the following formula:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \quad (1)$$

## Data Splitting and SMOTE

The data was subsequently split using an 80:20 ratio via the `train_test_split` function from scikit-learn, resulting in 1,562 training samples and 391 testing samples [7]. Due to the imbalanced class distribution, the SMOTE (Synthetic Minority Oversampling Technique) was applied to synthesize new data for the minority classes until all three classes were balanced, with 1,114 samples each [14]. SMOTE was applied after TF-IDF, as it operates on numerical data.

## Naïve Bayes Classification

The first algorithm employed was Naïve Bayes—specifically the Multinomial Naïve Bayes approach—which operates based on Bayes' Theorem under the assumption that each feature is independent of the others [3]. This algorithm was implemented using the scikit-learn library in Python, utilizing the following formula:

$$P(C \mid X) = \frac{P(X \mid C) \times P(C)}{P(X)} \quad (2)$$

CC Attribution-ShareAlike 4.0 License.

Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences E-ISSN 3047-8278
Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16
DOI : https://doi.org/10.66341/fusion.v3i1.342
WEB : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

where P(C|X) is the probability of document X belonging to class C, P(X|C) is the probability of feature X occurring given class C, P(C) is the prior probability of class C in the dataset, and P(X) is the overall probability of feature X occurring.

### Naïve Bayes Classification

The second algorithm used is Random Forest, an ensemble learning method that combines multiple decision trees; the final result is determined by a majority vote across all trees, resulting in a more stable model capable of reducing overfitting [7]. In this study, Random Forest was implemented using scikit-learn with 100 decision trees (n_estimators = 100), using the formula:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \dots, h_T(x)\} \quad (3)$$

where y is the final prediction result, h_1(x) to h_T(x) are the predictions from each decision tree, T is the total number of trees, and the mode is the most frequently occurring value from all tree predictions.

### Comparison and Evaluation

The performance of both models was evaluated using a confusion matrix with four main metrics [15]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

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

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

## 3. RESULTS AND DISCUSSION

### Data Collection

The web scraping process successfully collected 2,000 Indonesian-language reviews from Akulaku app users on the Google Play Store. The data includes four attributes: username, review content, rating, and review date, as shown in Table 1.

**Table 1.** Sample Scraping Data

| username | ulasan | rating | tanggal |
|---|---|---|---|
| Nathan Haunatas | semoga amanah | 5 | 27/06/2026 12:34 |
| Hwat lie | akulaku tidak berkualitas dan tidak ada sama ... | 1 | 27/06/2026 12:19 |
| Muhli Son | Menurutku bungaNya terlalu BESAR kalau pinjam ... | 1 | 27/06/2026 12:12 |
| Faiza adila husna Adila | sangat puas | 5 | 27/06/2026 12:07 |

### Preprocessing

The preprocessing stage produces clean, structured text through six stages. The final result of each stage can be seen in the text changes. For example, the sentence "In my opinion, the interest is too BIG"

CC Attribution-ShareAlike 4.0 License.

Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences E-ISSN 3047-8278
Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16
DOI : https://doi.org/10.66341/fusion.v3i1.342
WEB : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

after case folding becomes "In my opinion, the interest is too big," after stopword removal becomes "In my opinion, the interest is borrowing money," and after stemming becomes "Bagian bunga pinjam uang." This process ensures that the text entering the analysis stage is free from noise and has a uniform structure.

## Labeling

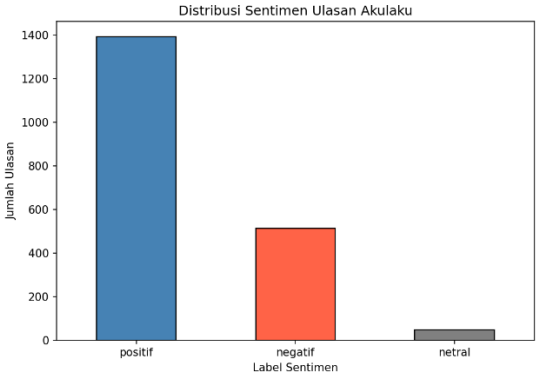


**Figure 2.** Distribution of Akulaku Review Sentiment

Based on the labeling results using rating values, 1,394 positive reviews, 514 negative reviews, and 47 neutral reviews were obtained from a total of 1,953 data points after preprocessing, as shown in Figure 2. These results indicate that the majority of users gave positive reviews of the Akulaku app. However, the significant proportion of negative sentiment still requires attention as evaluation material for app developers. This imbalance in the number of data points between classes is the reason for implementing the SMOTE technique in the next stage.

## Data Distribution and SMOTE

The data was divided in an 80:20 ratio, resulting in 1,562 training data points and 391 testing data points. Before SMOTE implementation, the distribution of the training data was highly imbalanced, with 1,114 data points in the positive class, 410 in the negative class, and only 38 in the neutral class. After SMOTE implementation, the three classes were balanced, with 1,114 data points each, as shown in Table 2.

**Table 2.** Data Distribution Before and After SMOTE

| Sentimen | Sebelum SMOTE | Setelah SMOTE |
|---|---|---|
| positif | 1114 | 1114 |
| negatif | 410 | 1114 |
| netral | 38 | 1114 |

## Naive Bayes Classification

The Naive Bayes model achieved an overall accuracy of 80%. Based on the classification report in Table 3, the model performed best in the positive class with a precision of 0.96, a recall of 0.84, and an F1-score of 0.89. In the negative class, the model achieved a precision of 0.69, a recall of 0.72, and an F1-score of 0.70, which is considered quite good. However, in the neutral class, the model performed less than optimally, with a precision of only 0.12 and an F1-score of 0.20. This was due to the very small amount of neutral data, making it more difficult to learn the patterns of this class. Based on the confusion matrix in Figure 3, the model successfully classified 233 positive reviews, 74 negative reviews, and 5 neutral reviews correctly.

CC Attribution-ShareAlike 4.0 License.

Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences
E-ISSN 3047-8278
Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16
DOI : https://doi.org/10.66341/fusion.v3i1.342
WEB : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

**Tabel 3.** Classification Report Naïve Bayes

| *Class / Metric* | *precision* | *recall* | *f1-score* | *support* |
|---|---|---|---|---|
| negatif | 0.69 | 0.72 | 0.70 | 103 |
| netral | 0.12 | 0.56 | 0.20 | 9 |
| positif | 0.96 | 0.84 | 0.89 | 279 |
| | | | | |
| *accuracy* | | | 0.80 | 391 |
| *macro avg* | 0.59 | 0.70 | 0.60 | 391 |
| *weighted avg* | 0.87 | 0.80 | 0.83 | 391 |

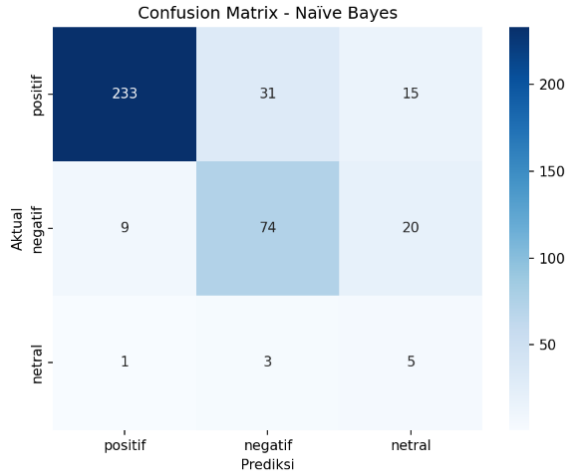


**Figure 3** Naïve Bayes Confusion Matrix

## Random Forest Classification

The Random Forest model produced an overall accuracy of 83%, higher than Naïve Bayes. Based on the classification report in Table 4, the model performed best in the positive class with a precision of 0.90, a recall of 0.89, and an F1-score of 0.90. In the negative class, the model produced a precision of 0.67, a recall of 0.76, and an F1-score of 0.71. However, in the neutral class, the model failed to correctly classify any of the reviews, with a precision of 0.00 and an F1-score of 0.00. This indicates that Random Forest struggled to recognize neutral class patterns despite applying SMOTE. Based on the confusion matrix in Figure 4, the model successfully classified 248 positive and 78 negative reviews correctly, but all nine neutral reviews were misclassified into the positive and negative classes.

Table 4 *Classification Report Random Forest*

| *Class / Metric* | *precision* | *recall* | *f1-score* | *support* |
|---|---|---|---|---|
| negatif | 0.67 | 0.76 | 0.71 | 103 |
| netral | 0.00 | 0.00 | 0.00 | 9 |
| positif | 0.90 | 0.89 | 0.90 | 279 |
| | | | | |
| *accuracy* | | | 0.83 | 391 |
| *macro avg* | 0.52 | 0.55 | 0.54 | 391 |
| *weighted avg* | 0.82 | 0.83 | 0.83 | 391 |

CC Attribution-ShareAlike 4.0 License.

Fusion : Journal of Research in Engineering, Technolgy and Applied Sciences
E-ISSN 3047-8278
Volume 3 ; Nomor 1 ; April 2026 ; Page 08-16
DOI : https://doi.org/10.66341/fusion.v3i1.342
WEB : *https://ejurnal.faaslibsmedia.com/index.php/fusion*

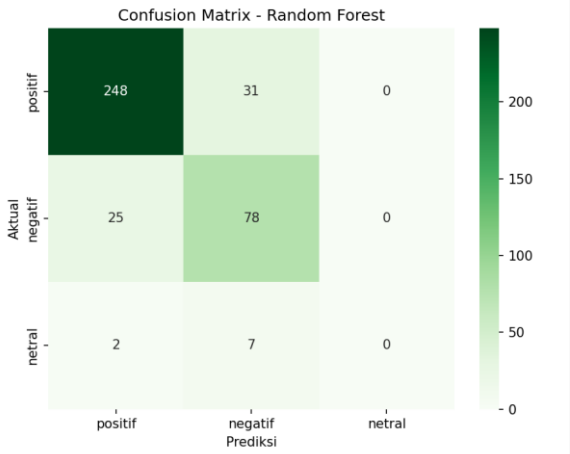


**Figure 4.** Confusion Matrix Random Forest

## Comparison and Evaluation

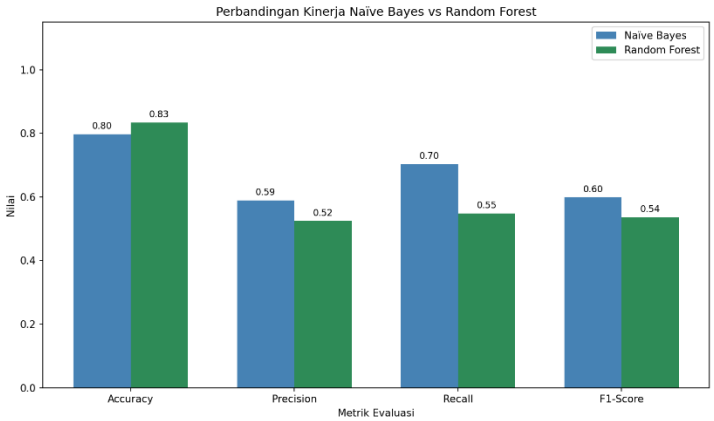


**Figure 5.** Comparison of Naïve Bayes vs Random Forest Performance

Based on Figure 5, the evaluation results show that Random Forest is superior in accuracy with a value of 0.83 compared to Naïve Bayes which reached 0.80. However, Naïve Bayes is superior in precision, recall, and F1-score metrics with macro average values of 0.59, 0.70, and 0.60, respectively, compared to Random Forest which only reached 0.52, 0.55, and 0.54. This difference occurs because the accuracy value is influenced by the majority class, namely positive, so that Random Forest which is better at recognizing the positive class gets higher accuracy. Meanwhile, Naïve Bayes is superior in macro average because it is better able to recognize patterns in minority classes, namely negative and neutral. This is in line with Agustin's findings in 2026 in the BRImo application review and Qomariah in 2026 in the Minecraft game review which proved that Naïve Bayes is more effective in handling word distribution independently in text data [3] [7].

## 4. CONCLUSION

This study compared the performance of the Naïve Bayes and Random Forest algorithms in classifying the sentiment of Akulaku app users on the Google Play Store. Of the 2,000 reviews collected, 1,953 were used after preprocessing, with a distribution of 1,394 positive reviews, 514 negative reviews, and 47 neutral reviews. Data imbalance was addressed using the SMOTE technique to balance the three classes with 1,114 data points each. Feature weighting was performed using TF-IDF.

CC Attribution-ShareAlike 4.0 License.

The test results showed that Random Forest achieved a higher accuracy of 83% compared to Naïve Bayes, which achieved 80%. However, Naïve Bayes outperformed in precision, recall, and F1-score metrics, with macro average values of 0.59, 0.70, and 0.60, respectively, compared to Random Forest, which only achieved 0.52, 0.55, and 0.54. Both algorithms experienced similar difficulty classifying the neutral class, likely due to the small number of neutral reviews, with only 47 reviews, even with SMOTE applied.

Based on these results, the choice of the best algorithm depends on the needs. If overall accuracy is the priority, Random Forest is preferred, but if performance balance between sentiment classes is the priority, Naïve Bayes is superior. Further research is recommended to increase the data set, especially for the neutral class, try other algorithms such as Support Vector Machines or Deep Learning, and explore more accurate labeling methods other than rating-based.

## 5. REFERENCE

[1] L. O. M. H. Abdillah Sam Mongkito, N. Ransi, L. Surimi, A. Tenriawaru, G. Gunawan, and B. Wijaya Rauf, "Analisis Sentimen Aplikasi Peminjaman Online Berdasarkan Ulasan Pada Play Store Menggunakan Metode Naïve Bayes Dan Support Vector Machine (Studi Kasus : Adakami Dan Easycash)," *AnoaTIK J. Teknol. Inf. Dan Komput.*, vol. 2, no. 2, pp. 121–128, Dec. 2024, doi: 10.33772/anoatik.v2i2.71.

[2] Putri Cahyani and Lufty Abdillah, "Perbandingan Performa Algoritma Naïve Bayes, SVM dan Random Forest Studi Kasus Analisis Sentimen Pengguna Sosial Media X," *KALBISCIENTIA J. Sains Dan Teknol.*, vol. 11, no. 02, pp. 12–21, Oct. 2024, doi: 10.53008/kalbiscientia.v11i02.3624.

[3] L. A. Qomariah, A. Eviyanti, H. Setiawan, and N. Ariyanti, "Klasifikasi Sentimen Ulasan Minecraft Pada Google Play Store: Studi Komparatif Naïve Bayes Dan Random Forest," vol. 10, no. 1, 2026.

[4] Kaeren and Andrianingsih, "Analisis Sentimen Aplikasi Linkaja Di Google Play Store Menggunakan Algoritma Naïve Bayes Dan Random Forest," *J. Ris. Dan Apl. Mhs. Inform. JRAMI*, vol. 06, no. 02, pp. 438–447, 2025, doi: 10.30998/jrami.v6i02.13821.

[5] A. K. Nugroho, L. N. Hayati, and S. R. Jabir, "Analisis Perbandingan Metode Naive Bayes dan Random Forest pada Klasifikasi Sentimen Publik terhadap Aplikasi Identitas Kependudukan Digital (IKD)," *J. Algoritma*, vol. 22, no. 2, Nov. 2025, doi: 10.33364/algoritma/v.22-2.2729.

[6] A. Sole and A. Zubair, "Analisis Perbandingan Naïve Bayes dan Random Forest untuk Evaluasi PKKMB Berbasis Sentimen Mahasiswa," *J. Janitra Inform. Dan Sist. Inf.*, vol. 6, no. 1, pp. 20–32, Apr. 2026, doi: 10.59395/nf4xmg03.

[7] S. Agustin, A. E. Pajri, and A. D. Ardianti, "Analisis Sentimen Aplikasi BRImo Pada Google Play Store Menggunakan Algoritma Naïve Bayes Dan Random Forest," *J. Algoritma*, vol. 23, no. 1, May 2026, doi: 10.33364/algoritma/v.23-1.3266.

[8] F. Marzuqi, Y. Purba, T. Syaputra, W. Bismi, I. Kurniawati, and R. Fahlapi, "Perbandingan Algoritma Machine Learning Untuk Sentimen Ulasan Aplikasi Jobstreet," *INFOTECH J.*, vol. 11, no. 2, pp. 416–423, Dec. 2025, doi: 10.31949/infotech.v11i2.16609.

[9] D. N. I. Huda, C. Prianto, and R. M. Awangga, "Analisis Sentimen Perbandingan Layanan Jasa Pengiriman Kurir Pada Ulasan Play Store Menggunakan Metode Random Forest dan Descision Tree," vol. 11, no. 02, 2023.

[10] A. Nurian, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naïve Bayes," *J. Inform. Dan Tek. Elektro Terap.*, vol. 11, no. 3s1, Sep. 2023, doi: 10.23960/jitet.v11i3s1.3348.

CC Attribution-ShareAlike 4.0 License.

[11] F. Riza, D. F. Hendrakusuma, B. Wibowo, and D. Y. A. Afghani, "Comparison Of Machine Learning Classification Algorithm Performance In Wondr By Bni Mobile Banking Review Sentiment Analysis," *J. Inf. Technol. Comput. Sci.*, 2025.

[12] Fenilinas Adi Artanto, "Implementasi Algoritma Random Forest dan Model Bag of Words Dalam Analisis Sentimen Mengenai E-Materai," *SATESI J. Sains Teknol. Dan Sist. Inf.*, vol. 4, no. 2, pp. 139–145, Oct. 2024, doi: 10.54259/satesi.v4i2.3240.

[13] Sari, Pratama, and Wijaya, "Sistem Informasi Pendaftaran Mahasiswa Baru Berbasis Web," *J. Sist. Inf.*, 2021.

[14] F. A. T. Putra, A. B. P. Negara, and H. Sastypratiwi, "Perbandingan Kinerja Naive Bayes Dan Random Forest Dengan Penanganan Imbalance Data," vol. 12, 2026.

[15] M. F. Y. Herjanto and C. Carudin, "Analisis Sentimen Ulasan Pengguna Aplikasi Sirekap Pada Play Store Menggunakan Algoritma Random Forest Classifer," *J. Inform. Dan Tek. Elektro Terap.*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4192.

CC Attribution-ShareAlike 4.0 License.