INTEK Jurnal Penelitian
Vol. 11, No. 1, pp. 37-43, April 2024

Sentiment Analysis Classification on PLN Mobile Application Reviews using Random Forest Method and TF-IDF Feature Extraction

Muhamad Dafa Rizkiyanto[1,*,a] ,Mahendra Dwifebri Purbolaksono[1,b], Widi Astuti[1,c]
[1] Department School of Computing, Telkom University, Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, Jawa Barat, Indonesia, 40257
[*,a]dafarizkiyanto@student.telkomuniversity.ac.d (Corresponding Author), [b]mahendradp@telkomuniversity.ac.id, [c]widiwdu@telkomuniversity.ac.id

***Abstract*— PT PLN (Persero) has developed the PLN Mobile application to provide electricity services. The large number of users has resulted in various reviews regarding the strengths, weaknesses, and issues of the application. To evaluate the application's quality, sentiment analysis is conducted on user reviews. Review data is obtained through the Google-Play-Scraper API and cleaned through text preprocessing. This study utilizes the TF-IDF feature extraction method and Random Forest for classification. TF-IDF involves weighting each word in a text. This method transforms words into numerical representations, indicating both their frequency and relevance within the document's context. Random Forest is a supervised machine learning algorithm that utilizes ensemble learning which categorizes reviews into positive and negative, This study produced the best model using stemming data and TF-IDF unigram, along with a combination of hyperparameters. The n-estimator was set to 100, max_feature to log2, max_depth to unlimited (none), and entropy criterion, resulting in the highest F1-Score of up to 93.14%.**

Keywords*— *sentiment analysis, user reviews, pln mobile, random forest, tf-idf

I. Introduction

In the digital era, technology has transformed the way we interact with both public and private services, including access to electricity services. PLN Mobile, the latest innovation from PT PLN (Persero), facilitates electricity bill payments, token purchases, meter number recording, power additions, complaints, token purchase monitoring, electricity usage tracking, bill and outage notifications, repair information, and electricity network maintenance monitoring [1]. With a growing user base, reviews of the PLN Mobile application encompass suggestions, strengths, weaknesses, and user experiences, and sentiment analysis aids in evaluating the application's quality based on their opinions.

Sentiment analysis is a machine learning technique that assesses human opinions on entities such as products, services, individuals, or topics through reviews and ratings [2]. Sentiment analysis can provide the necessary information for various purposes [3]. Sentiment analysis is also known as subjective analysis, categorizing text based on the revealed tendencies and direction of opinions into positive, neutral, and negative [4], positive indicates good quality, negative indicates shortcomings, and neutral is an unbiased evaluation. One machine learning technique for sentiment classification is Random Forest. Previous research titled "Sentiment Analysis of Hotel Customers in Purwokerto Using Random Forest and TF-IDF" [5] demonstrated that this method achieved accuracies of 87.23% and 87.01% without stemming. In another study [6], the use of Random Forest for classifying Dana application reviews resulted in precision, recall, F1-Score, and accuracy of 84% each. This study aims to analyze sentiment in PLN Mobile application reviews using TF-IDF and Random Forest methods, providing valuable insights for PLN Mobile to enhance their services based on user feedback.

In a study [7]. This research analyze twitter user sentiment towards Shopeepay using the Random Forest algorithm. Data was collected from Twitter and processed using the TF-IDF method for word weighting. Subsequently, the model underwent assessment utilizing both a Confusion Matrix and K-Fold Cross Validation. The test outcomes reveal the model's excellent performance, achieving a precision of 95%, recall of 94%, F1-Score of 95%, and accuracy of 95%.

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774

INTEK Jurnal Penelitian
Vol. 11, No. 1, pp. 37-43, April 2024

In a study [8], the research compared Naïve Bayes, Random Forest, and SVM algorithms in sentiment analysis of Ruangguru application reviews. The results showed Naïve Bayes with an accuracy of 94.16%, SVM with 96.01%, and Random Forest with the highest, reaching 97.16%. Random Forest stands out as the algorithm with the best performance.

In a study [5], this research compared sentiment analysis of hotel customers in Purwokerto using Random Forest and TF-IDF methods, utilizing data from TripAdvisor. Three experimental scenarios were conducted to analyze the impact of preprocessing on the accuracy of sentiment classification models using the Random Forest algorithm. The results showed that adding custom stop words increased accuracy. Random Forest achieved accuracies of 87.23% with stemming and 87.01% without stemming.

In a study [9], This study centered on analyzing sentiments expressed in reviews of the PeduliLindungi application on the Google Play Store, employing Random Forest and SMOTE techniques. The research yielded an accuracy rate of 71%, accompanied by a recall and precision of 70%.

In a study [6], this research focuses on sentiment analysis of Dana application reviews using the Random Forest method. The Random Forest method is utilized to classify three sentiment classes, namely positive, negative, and neutral. The study also involves evaluation indicators such as accuracy, recall, precision, and F-measure. Testing is conducted with variations in the number and depth of trees on 1354 data, divided into 250 data for each class. Based on the testing and analysis results, comparing training and test data in an 80%:20% ratio, precision, recall, F1-Score, and accuracy values of 84% are obtained.

The main aim of this research is to conduct a sentiment analysis on user reviews of the PLN Mobile application to evaluate its quality and identify areas for improvement based on user opinions. This analysis utilizes TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction and employs the Random Forest algorithm for classification. By systematically analyzing user sentiment, this study seeks to gain insights into user satisfaction and identify specific aspects of the application that require enhancement to improve the overall user experience.

II. Research Methodology

A. System Design

The design of a system model for sentiment analysis on the PLN Mobile application using TF-IDF and Random Forest as depicted in Fig. 1.

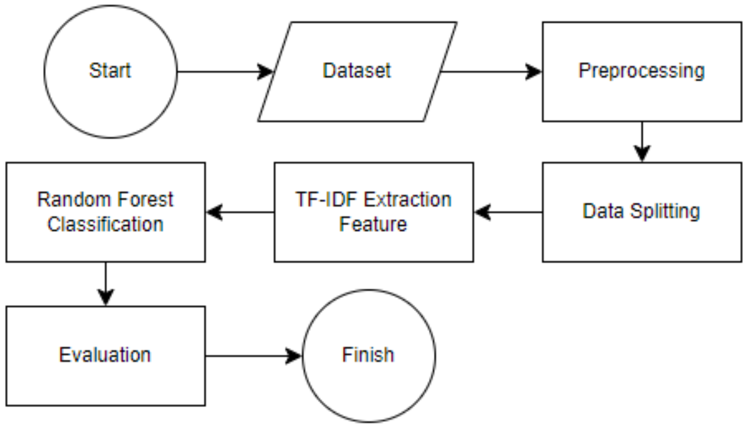

Figure 1 System Model Workflow

B. Data Collection

This research acquired the dataset through web scraping technique using the Google-Play-Scraper API to extract reviews from the PLN Mobile application on the Google Play Store, resulting in approximately 1375 reviews. Subsequently, these reviews were manually reviewed and categorized into two classes, namely positive and negative. As shown in Fig 2.

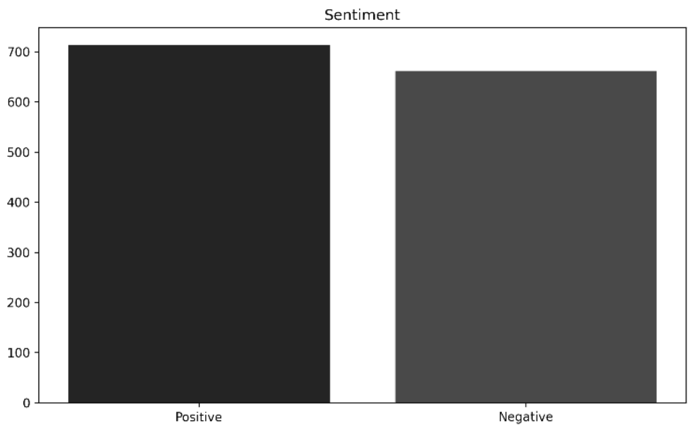

Figure 2. Sentiment Distribution

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774

The labeling process revealed that 713 reviews were classified as positive and 662 reviews were classified as negative. With a balanced dataset between the two sentiment categories, this study is ready to proceed with further analysis to understand patterns and trends in user reviews of the PLN Mobile application.

C. Data Prepocessing

Preprocessing is performed to address issues in data processing. The preprocessing stages involve cleansing, case folding, stopword removal, stemming, and tokenization. The flowchart of the preprocessing steps applied can be seen in Fig. 3, which illustrates the preprocessing workflow

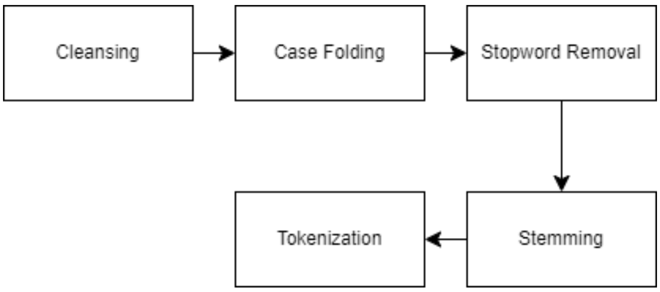

Figure 3. Preprocessing Workflow

1) Cleansing: The cleansing process in preprocessing aims to remove unwanted characters, such as punctuation marks, special characters, or inconsistent formats. This ensures that the data used is clean and ready for further analysis, as shown in Table 1.

Table 1. Cleansing

Review	**Cleansing Result**
“Dengan pln mobile memudahkan dari pembelian pulsa, bayar rekening listrik, tambah daya,pasang baru dan pengaduan gangguan”	“Dengan pln mobile memudahkan dari pembelian pulsa bayar rekening listrik tambah daya pasang baru dan pengaduan gangguan”

2) Case Folding : Case folding is designed to consistently transform all letters to lowercase throughout all words without any exceptions. [10]. An example of the case folding process can be seen in Table 2.

Table 2. Case Folding

Review	Case Folding Result
“Dengan pln mobile memudahkan dari pembelian pulsa bayar rekening listrik tambah daya pasang baru dan pengaduan gangguan”	“dengan pln mobile memudahkan dari pembelian pulsa bayar rekening listrik tambah daya pasang baru dan pengaduan gangguan”

3) Stopword Removal : Stopword removal seeks to eradicate frequently occurring words that hold minimal significance in data analysis, such as conjunctions and prepositions, as shown in Table 3.

Table 3. Stopword Removal

Review	Stopword Removal Result
“dengan pln mobile memudahkan dari pembelian pulsa bayar rekening listrik tambah daya pasang baru dan pengaduan gangguan”	“pln mobile memudahkan pembelian pulsa bayar rekening listrik daya pasang pengaduan gangguan”

4) Stemming : Stemming is the process of mapping and reducing various word forms to their root form [11]. This process involves removing affixes from words so that words with the same root can be treated as a single entity. An illustration of the stemming procedure is presented in Table 4.

Table 4. Stemming

Review	**Stemming Result**
“pln mobile memudahkan pembelian pulsa bayar rekening listrik daya pasang pengaduan gangguan”	“pln mobile mudah beli pulsa bayar rekening listrik daya pasang adu ganggu”

5) Tokenization : The tokenization process aims to separate text into smaller units, such as words or phrases. An example of tokenization can be seen in Table 5.

Table 5. Tokenization

Review	**Tokenization Result**
“pln mobile mudah beli pulsa bayar rekening listrik daya pasang adu ganggu”	“['pln', 'mobile', 'mudah', 'beli', 'pulsa', 'bayar', 'rekening', 'listrik', 'daya', 'pasang', 'adu', 'ganggu']”

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774

D. Data Splitting

During this phase, the dataset will be partitioned into two segments: training data and testing data. The training data comprises 80% of the entire dataset, equivalent to 1,100 PLN Mobile reviews, whereas the testing data constitutes 20% of the total dataset, totaling 275 reviews. The training data is used to build the classification models, while the testing data is used to assess how accurately these models can make predictions [12]. The purpose of dividing the data into different categories is to avoid over-fitting, data imbalance, and to provide an objective evaluation of the model's performance. Data imbalance arises when there is a substantial variance in the sample counts between different classes [13]. The results of the data division can be seen in Table 6.

Table 6. Split Data Result

Dataset	Data Train	Data Test
Sentiment	1100	275
Review	1100	275

E. TF-IDF Extraction Feature

The weighting of words or Term Frequency-Inverse Document Frequency (TF-IDF) is a process to assign values to each word [6]. The TF-IDF weighting method is utilized to transform words into numerical representations [14]. It assigns importance to every word or feature present in the text, indicating both the frequency of word occurrence within documents and their significance within the document's context [15].

While DF reflects how common a word appears across the entire document collection, IDF assigns higher weights to words that occur less frequently. Here is the equation for TF-IDF.

$$IDF = log\frac{D}{DF} \quad (1)$$

$$TF - IDF = tf * idf \quad (2)$$

D represents the total number of documents in the training dataset, while *DF* signifies the number of documents containing the particular word. *tf* represents the frequency of the word within the document, and *idf* stands for the inverse of the document frequency for each word. The weight of a word increases with its high frequency in a document, and conversely, decreases when the word appears in many documents [16].

F. Random Forest Classification

Random Forest is a supervised machine learning algorithm that utilizes ensemble learning, combining various types of algorithms or the same algorithm multiple times to form a stronger predictive model [17]. This confirms that the performance of Random Forest is superior compared to other classification algorithms [8]. It constructs multiple classification trees, enhancing accuracy by generating child nodes for each node and conducting random selection.

In this study, Random Forest is used to predict text into positive and negative classes. This is achieved by combining the results from each decision tree, where decision trees calculate entropy as an indicator of attribute impurity and information gain value [18], as shown in Equation 3 [19]:

$$Entropy\ (Y) = -\sum_i p(c|Y)log_2 p(c|Y) \quad (3)$$

In the explanation provided, *Y* represents the set or collection of cases, while *P(c|Y)* is the ratio of the value *Y* to class *c*. Information gain is a measure of the effectiveness of features in distinguishing different classes in the data. As shown in Equation 4 [19]:

$$Information\ Gain\ (Y, a) = Entropy(Y) - \sum_{v \in Values(a)} \frac{|Y_v|}{|Y_a|} Entropy(y_v). \quad (4)$$

Values(a) refers to the possible values in the set of cases a, while Y_v indicates class v related to class a as a subclass of *Y*, and Y_a encompasses all values corresponding to a. High information gain signifies that a feature is more effective in distinguishing different classes. Therefore, features with higher information gain are typically chosen in the construction of a Random Forest model.

G. Confusion Matrix

The confusion matrix, a tabular representation employed for assessing the effectiveness of classification models in machine learning, illustrates the categorization of both accurately and inaccurately classified test data [20]. This matrix compares the predicted values of the model with the actual values from the training data. It contains four main cells: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The confusion matrix table can be seen in Table 7 [3].

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774

Table 7. Confusion Matrix

Actual Label	Predicted Label	
	Positive (+)	Negative (-)
Positive (+)	TP	FN
Negative (-)	FP	TN

- True Positives (TP) are the number of positive data points classified as positive.
- False Positives (FP) are the number of negative data points classified as positive.
- False Negatives (FN) are the number of positive data points classified as negative.
- True Negatives (TN) are the number of negative data points classified as negative.

The evaluation of the confusion matrix is a matrix used to test and estimate the correctly and incorrectly classified objects, thereby producing values for accuracy, precision, and recall [18]. Based on these values, we can calculate accuracy, recall, precision, and F1-score using the following equation [3]:

1) Accuracy measures the extent to which the classifi-cation model provides correct predictions overall as shown in Equation 5.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

2) Recall measures the extent to which the model can identify or capture all actual positive cases as shown in Equation 6.

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

3) Precision measures how accurate the model's positive predictions are, or the percentage of positive predictions that are correct as shown in Equation 7.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

4) The F1-score combines both precision and recall, providing a balanced metric between the two as shown in Equation 8.

$$F1 = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (8)$$

III. Results and Discussion

In this study, 1375 data points were obtained from user comments on the PLN Mobile application in the Google Playstore. The dataset underwent preprocessing stages including cleaning, case folding, stopword removal, stemming, and tokenization. Subsequently, the data was split into 80% training data (1100 data points) and 20% testing data (275 data points). TF-IDF feature extraction was then employed to convert the text into numerical representations. After successful extraction, the data was classified using the Random Forest algorithm. This research comprises three scenarios to analyze the influence of various factors on the performance of the TF-IDF and Random Forest model in text classification, as shown in Table 8.

Table 8. Experiment Scenario

Scenario	Experiment
1	Comparison of Stemming and Non-Stemming Performance
2	Performance Comparison of TF-IDF Unigram and Bigram
3	The Performance of Hyperparameter Tuning in Random Forest Classification Model

For the analysis, the following conditions were applied: performing stemming during preprocessing, using TF-IDF with n-gram = 1, and applying Random Forest without hyperparameter tuning.

A. Scenario 1 : Comparison of Stemming and Non-Stemming Performance

In the first experimental scenario, a comparison of classification results was conducted using the Random Forest algorithm on two different datasets: data that had undergone stemming and data that had not undergone stemming. The objective of this study was to assess how stemming affects the F1-Score performance in Random Forest classification. The outcomes of this investigation are presented in Table 9.

Table 9. Scenario 1 Result

Pre-processing	F1-Score
Non-Stemming	91.1%
Stemming	**91.81%**

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774

Based on the table above, it can be seen that the data which has undergone the stemming process exhibits slightly better performance compared to the non-stemmed data. This is evidenced by the F1-Scores obtained, namely 91.81% for the stemmed data and 91.1% for the non-stemmed data. The results suggest that utilizing stemming and stopword elimination techniques in the preprocessing phase does not have a substantial impact on the accuracy of sentiment analysis [21]. In specific instances, stemming might not notably affect the precision of sentiment analysis, owing to differences in text attributes, language usage, sentiment context, and data integrity.

B. Scenario 2 : Performance Comparison of TF-IDF Unigram and Bigram

In the second experimental scenario, a comparison of Random Forest classification results was conducted using two sets of data: one with TF-IDF unigram features and another with TF-IDF bigram features. The objective of this experiment is to evaluate the impact of using unigram and bigram features on F1-Score performance. The analysis results are expected to provide insights into which features are more effective in enhancing model performance. The results of this experiment can be seen in Table 10.

Table 10. Scenario 2 Result

TF-IDF	**F1-Score**
Unigram	**92.25%**
Bigram	83.33%

Based on the table above, it can be observed that the utilization of TF-IDF with unigram features resulted in an F1-Score of 92.25%, whereas the utilization of TF-IDF with bigram features only yielded an F1-Score of 83.33%. The use of N-Gram such as Unigram and Bigram in the transformation process using TF-IDF has the potential to influence the classification accuracy [22]. The TF-IDF unigram often proves more effective than bigram in certain scenarios due to its evaluation of words independently without additional complexity. Although bigram considers word sequence to capture more context, in small document sets, this can result in infrequent feature occurrences and reduced accuracy.

C. Scenario 3 : The Performance of Hyperparameter Tuning in Random Forest Classification Model

In the third experimental scenario, a comparison of classification results was conducted using the Random Forest algorithm with various hyperparameter settings to evaluate their impact on model performance. The parameters tested included the number of decision trees (*n_estimators*: 50, 100, 200), the number of features considered for the best split (*max_features*: auto, sqrt, log2), the maximum depth of the trees (*max_depth*: 10, 20, 30, None), and the criteria for splitting (*criterion*: gini, entropy). The objective of this experiment was to identify the optimal combination of hyperparameters to improve the F1-Score in Random Forest classification. The top five results from these experiments can be seen in Table 11.

Table 11. Scenario 3 Result

Parameter				Metric
n_estimator	max_feature	max_depth	criterion	F1-score
200	log2	none	gini	92.09%
100	**log2**	**none**	**entropy**	**93.14%**
200	log2	none	entropy	92.75%
100	log2	30	gini	90.78%
50	log2	none	gini	93.04%

Table 11 above presents various parameter combinations tested to optimize model performance by examining the resulting F1-Score metric. Different parameter combinations yielded varying results, with the highest F1-Score obtained in the configuration with an *n-estimator* of 100, *max_feature* log2, *max_depth* unlimited (none), and *criterion* entropy. This combination resulted in an F1-Score of 93.14%, which is the highest value among all tested combinations.

Another configuration that approached the best performance is the utilization of an n-estimator parameter of 50, max_feature log2, max_depth unlimited (none), and criterion gini, which resulted in an F1-Score of 93.04%. Despite being slightly lower, this result still demonstrates good performance.

Conclusion and Future Work

The results of this study on sentiment analysis of user reviews for the PLN Mobile application using the TF-IDF and Random Forest methods indicate that preprocessing

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774

with stemming significantly improves model performance. Through a series of experiments, it was found that the utilization of TF-IDF unigrams yielded excellent performance with an F1-Score of 92.25%. Additionally, the combination of hyperparameters with *n-estimator* set to 100, *max_feature* to log2, *max_depth* unlimited (none), and *criterion* entropy resulted in the highest F1-Score, reaching 93.14%.

For future research, it is recommended to increase the amount of data analyzed so that the model can learn from more examples and enhance generalization capabilities. Furthermore, it is suggested to explore alternative algorithms to compare their performance with Random Forest, providing deeper insights into the effectiveness of various sentiment analysis techniques. Finally, conducting various other testing scenarios may be necessary to identify the optimal configuration that can achieve a higher F1-Score and improve the overall model performance.

Acknowledgement

I wish to extend my heartfelt thanks to my two supervisors for their significant role in this research, offering invaluable feedback and corrections. I am also grateful to my colleagues for their support and contributions to this study. Furthermore, I appreciate the authors whose works were cited in this research. I hope this research will prove to be beneficial in the future.

References

[1] Y. Asri, W. N. Suliyanti, D. Kuswardani, and M. Fajri, “Analisis Sentimen Pelabelan Otomatis Lexicon Vader dan Klasifikasi Naive Bayes dalam menganalisis sentimen data ulasan PLN Mobile: Analisis Sentimen”, petir, vol. 15, no. 2, pp. 264–275, Nov. 2022.
[2] B. Liu, Sentiment analysis and opinion mining. Springer, 2012.
[3] I. Afdhal et al., “Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia,” Jurnal Nasional Komputasi dan Teknologi Informasi, vol. 5, no. 1, 2022.
[4] P. Karthika, R. Murugeswari, and R. Manoranjithem, ‘Sentiment Analysis of Social Media Network Using Random Forest Algorithm’, in 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2019, pp. 1–5.
[5] B. Baskoro, I. Susanto, and S. Khomsah, “Analisis Sentimen Pelanggan Hotel di Purwokerto Menggunakan Metode Random Forest dan TF-IDF (Studi Kasus: Ulasan Pelanggan Pada Situs TRIPADVISOR)”, INISTA, vol. 3, no. 2, pp. 21-29, Jun. 2021.
[6] F. A. Larasati, D. E. Ratnawati, dan B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest”, J-PTIIK, vol. 6, no. 9, hlm. 4305–4313, Sep 2022.
[7] T. F. Basar, D. E. Ratnawati, dan I. Arwani, “Analisis Sentimen Pengguna Twitter terhadap Pembayaran Cashless menggunakan Shopeepay dengan Algoritma Random Forest”, J-PTIIK, vol. 6, no. 3, hlm. 1426–1433, Feb 2022.
[8] E. Fitri, ‘Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine’, Jurnal Transformatika, vol. 18, p. 71, 07 2020.
[9] M. Pribadi, D. Manongga, H. Purnomo, I. Setyawan, and H. Hendry, ‘Sentiment Analysis of the PeduliLindungi on Google Play using the Random Forest Algorithm with SMOTE’, 07 2022, pp. 115–119.
[10] M. R. Adrian, M. P. Putra, M. H. Rafialdy, and N. A. Rakhmawati, ‘Perbandingan Metode Klasifikasi Random Forest dan SVM Pada Analisis Sentimen PSBB’, J. Inform. UPGRIS, vol. 7, no. 1, Jun. 2021.
[11] M. U. Albab, M. N. Fawaiq, and Others, ‘Optimization of the Stemming Technique on Text Preprocessing President 3 Periods Topic’, Jurnal Transformatika, vol. 20, no. 2, pp. 1–12, 2023.
[12] K. Hashim, Y. Sibaroni, and S. Prasetyowati, ‘The Effectiveness of the Ensemble Naive Bayes in Analyzing Review Sentiment of the Lazada Application on Google Play’, 01 2024, pp. 1–5.
[13] Z. Xiao, L. Wang, and J. Y. Du, ‘Improving the performance of sentiment classification on imbalanced datasets with transfer learning’, IEEE Access, vol. 7, pp. 28281–28290, 2019.
[14] A. Wardani, K. Adiwijaya, and M. Dwifebri Purbolaksono, ‘Sentiment Analysis on Beauty Product Review Using Modified Balanced Random Forest Method and Chi-Square’, Journal of Information System Research (JOSH), vol. 4, pp. 1–7, 10 2022.
[15] J. A. Septian, T. M. Fachrudin, and A. Nugroho, “Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor”, INSYST, vol. 1, no. 1, pp. 43–49, Aug. 2019.
[16] V. Amrizal, “Penerapan metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada sistem temu kembali informasi untuk mengetahui syarah hadits berbasis web (Studi Kasus: Hadits Shahih Bukhari-Muslim),” JURNAL TEKNIK INFORMATIKA, vol. 11, pp. 149–164, 11 2018.
[17] N. Bahrawi, “Sentimen Analysis Using Random Forest Algorithm-Online Social Media Based”, JITU, vol. 2, no. 2, pp. 29–33, Dec. 2019.
[18] G. Sandag, ‘Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest’, CogITo Smart Journal, vol. 6, p. 167, 12 2020.
[19] Y. Nugroho and N. Emiliyawati, ‘Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest’, Jurnal Teknik Elektro, vol. 9, pp. 24–29, 06 2017.
[20] D. Normawati and S. A. Prayogi, ‘Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter’, J-SAKTI (Jurnal Sains Komputer Dan Informatika), vol. 5, no. 2, pp. 697–711, 2021.
[21] A. Pradana and M. Hayati, ‘The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts’, Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol. 4, 10 2019.
[22] N. Arifin, U. Enri, and N. Sulistiyowati, ‘Penerapan Algoritma Support Vector Machine (SVM) dengan TF-IDF N-Gram untuk Text Classification’, STRING (Satuan Tulisan Riset Dan Inovasi Teknologi), vol. 6, no. 2, pp. 129–136, 2021

DOI: http://dx.doi.org/10.31963/intek.v11i1.4774