IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.19, No.1, January 2025, pp. 13~24
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.101910

Sentiment Analysis Mobile JKN Reviews Using SMOTE Based LSTM

Ghufron Tamami*[1], Wiwit Agus Triyanto[2], Syafiul Muzid[3]
[1,2,3]Department of Information System, FT UMK, Kudus, Indonesia
e-mail: ***[1]202153111@std.umk.ac.id**, [2]at.wiwit@umk.ac.id, [3]syafiul.muzid@umk.ac.id

Abstrak

Aplikasi Mobile JKN berperan penting dalam memberikan akses yang mudah dan cepat terhadap layanan kesehatan bagi pengguna JKN-KIS. Namun, ulasan pengguna menunjukkan ketidakpuasan terhadap beberapa aspek aplikasi, seperti masalah login dan kode OTP, yang dapat memengaruhi pengalaman pengguna secara keseluruhan. Tantangan lain yang dihadapi adalah ketidakseimbangan kelas pada dataset ulasan, yang dapat memengaruhi kinerja analisis sentimen. Penelitian ini menggunakan Long Short-Term Memory (LSTM) yang dikombinasikan dengan Synthetic Minority Oversampling Technique (SMOTE) untuk mengatasi ketidakseimbangan kelas. Data ulasan dikumpulkan dari Google Play Store dan Kaggle, kemudian dilakukan preprocessing mencakup lemmatization, tokenization, dan padding. Kinerja model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa LSTM dengan SMOTE mencapai akurasi 88%, presisi 90%, recall 88%, dan F1-score 89%. SMOTE berhasil meningkatkan kinerja pada kelas minoritas meskipun terdapat sedikit penurunan pada akurasi dibandingkan model tanpa SMOTE. Visualisasi word cloud mengungkapkan sentimen positif terkait kemudahan penggunaan aplikasi, sementara sentimen negatif menunjukkan area yang memerlukan perbaikan. Penelitian ini menegaskan pentingnya penanganan dataset tidak seimbang untuk menghasilkan analisis sentimen yang lebih akurat.

Kata kunci— *Analisis Sentimen, Data Tidak Seimbang, LSTM, Mobile JKN, SMOTE*

Abstract

The JKN Mobile application plays an important role in providing JKN-KIS users with easy and fast access to health services. However, user reviews indicate dissatisfaction with several aspects of the application, such as login issues and OTP codes, which can affect the overall user experience. Another challenge faced is class imbalance in the review dataset, which can affect the performance of sentiment analysis. This research is using Long Short-Term Memory (LSTM) combined with Synthetic Minority Oversampling Technique (SMOTE) to manage the class imbalance. Review data is collected from the Google Play Store and Kaggle platform, then preprocessed including lemmatization, tokenization, and padding. Model performance was evaluated using the metrics accuracy, precision, recall, and f1-score. The research results show that LSTM with SMOTE achieves 88% accuracy, 90% precision, 88% recall, and 89% F1-score. SMOTE successfully improved performance in the minority class although there was a slight decrease in accuracy compared to the model without SMOTE. Word cloud visualization reveals positive sentiments regarding the ease of use of the application, while negative sentiments indicate areas that need improvement. This study emphasizes the importance of handling imbalanced datasets to produce more accurate sentiment analysis.

Keywords— *Sentiment Analysis, Imbalanced Data, LSTM, Mobile JKN, SMOTE*

Received November 26th,2024; Revised Desember 20th, 2024; Accepted January 2nd, 2025

1. INTRODUCTION

BPJS Kesehatan plays in important role in enhancing the quality of health services for Indonesian citizens [1], such as developing a mobile application services for healthcare services. Mobile JKN is one of the most applications that gave a mobile access to health facilities and services. This app have been released by the BPJS Kesehatan for health facilities and services in July 2017, the aim of this product is enhancing services for JKN-KIS users [2]. Through Mobile JKN, users can easily access a variety of health services, such as registering for health facilities, viewing membership status, and making direct payments for healthcare services [3]. This mobile application innovation is expected to improve users efficiency and comfort in accessing healthcare.

However, over time, it's very important for service providers to continue improving the application quality based on user feedback and needs. Feedback is crucial for evaluating the current application quality. Feedback can containing insights into user satisfaction and suggestions for improvement or even complaints about performance of the application [4]. According to Presidential Regulation Number 49 of 2024, consumer empowerment who are able to make optimal decisions and understand their preferences [5]. Based on this information, application developers need to gain knowledge on app shortcomings or suggestions, which can be analyzed using sentiment analysis.

Sentiment analysis is the study of text mining and natural language processing with the aim of automatically extracting insights from text data, such as in review text [6]. This analysis often used for find out how users respond to a product on the market [7]. In addition, sentiment analysis is an automatic extraction process that extracting attitudes, opinions and emotions packaged in textual data [8]. By using sentiment analysis to analyze user review data will provide important information for application development [9]. Advanced analytics with Long Short-Term Memory (LSTM) can improve this process by effectively handling sequential data and can capture long-term dependencies in text, which can provide deeper insight into user sentiment [10].

User review data can contain information that is very important for assessing whether a product is well received or not by users. By using machine learning, this review data can be processed into valuable insights that regarding customer satisfaction with an application service. Review data is usually grouped into a satisfaction index ranging from 1 to 5, but it is often imbalanced. Because users usually give a rating of more than 4 or 5, even though the contents of the review show complaints or dissatisfaction with the app service. This issue is known as imbalanced dataset. For example, in the review data Mobile JKN on play store, there are many more ratings 4 and 5 than other ratings. This imbalance can have a negative impact on sentiment analysis results.

This Imbalance dataset can causes a problem known as data bias [11], where the majority of the data has high values, while data with the low values become minority class. This condition has the potential to affect the performance of machine learning model, this is because the model will only prioritize majority data and ignore important information from minority data. This can reduce the effectiveness of the model in objectively assessing customer satisfaction.

Based on this issue, an oversampling method is needed during the data preprocessing stage of sentiment analysis to handle imbalanced datasets and ensure that the model can work optimally with the data. One possible solution is using Synthetic Minority Oversampling Technique (SMOTE). SMOTE is one of the oversampling methods that adds synthetic data to the minority class, so that the number is the same as the majority class. Several previous research have shown that SMOTE can effectively manage class imbalance problems and improve model performance.

There are several previous studies related to sentiment analysis using SMOTE for imbalance data that have been carried out in several studies, such as research [12][13][14], which applies Naive Bayes and SMOTE to handle the imbalance data and shows significantly improved accuracy. On the other research using SMOTE to manage imbalance data with other models like random forest [15] and K-Nearest Neighbors [16]. This research has shown that SMOTE kindly improves recall and provides table performance across the models. The consistent improvement across variant studies demonstrates the reliability of SMOTE in addressing class imbalance, making it a better choice for managing the imbalance dataset in sentiment analysis and process them with LSTM model.

This research will consist of several stages, including data collection from Google Play Store and Kaggle, data preprocessing consisting of labeling, lemmatization tokenization and padding, implementation of the SMOTE method to handle class imbalance, model training using the LSTM model, and the final step is evaluating model performance using performance metrics (accuracy, precision, recall, and f1 score). The dataset consists of 100,000 user reviews of the Mobile JKN application collected from Google Play Store and Kaggle.

2. METHODS

This research method provides a detailed description of the data collection used in this research. There are two main processes in this research, namely: data preprocessing and building sentiment analysis using the SMOTE method based on the LSTM model. This research will use Python as a programming language and Jupyter Notebook as an IDE. The research method will be shown in Figure 1 below.

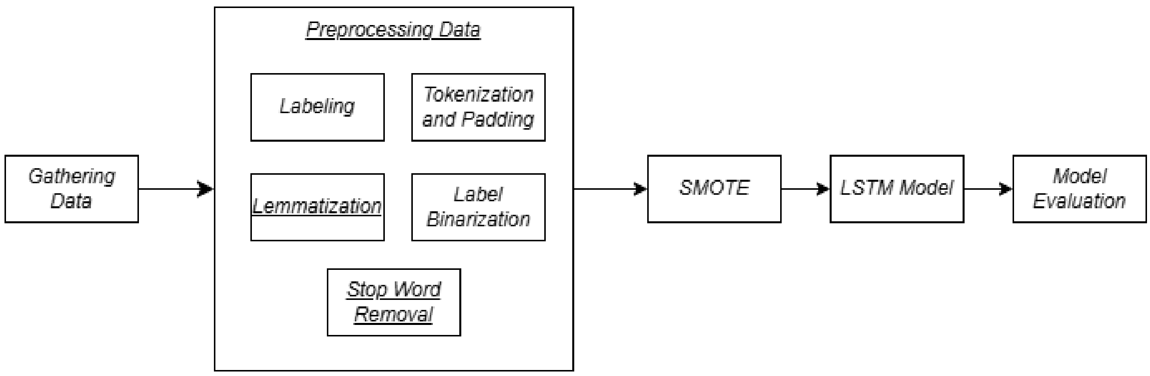

Figure 1 Research Method

Based on Figure 1 above, this research method will contain several important stages of sentiment analysis, starting from data collection, preprocessing data, and evaluation of results. These several stages aim to ensure that the data obtained can be accepted as input for the model and provide the best results. The research method will be explained in detail below.

2.1 Gathering Data

This research begins with collecting a dataset. The data collected is sourced from the Kaggle platform, using the relevant keyword “Ulasan Mobile JKN” to gather the dataset from Mobile JKN Google Play Store reviews. The dataset can be accessed from Kaggle platform at this link (https://www.kaggle.com/datasets/nuricahyono/mobile-jkn). Below is an example of a dataset that will be displayed in Table 1.

Table 1 Dataset Kaggle Mobile JKN

UserName	score	at	content
Pengguna Google	1	2024-09-08 01:36:21	“Aplikasi Rusak”
Pengguna Google	5	2024-09-08 01:20:04	”alhamdulillah aplikasi ini sudah berjalan dengan lancar,semoga tambah maju dan tidak mengecewakan”
Pengguna Google	4	2024-09-08 01:11:55	“Sangat baik”
Pengguna Google	2	2024-09-08 01:03:25	“tidak bisa perifikasi no hp”
Pengguna Google	5	2024-09-08 00:41:17	”Baru pake bkm terlalu tahu”

After getting the dataset from Kaggle, the data will then continue with preprocessing so that it can be used in machine learning models. Apart from that, a dataset containing user reviews of the Mobile JKN application collected from the Google Play Store will also be utilized, this process will be using Google Play Scrapper library with Python. Below is the sample of the dataset will be shown in Table 2.

Table 2 Dataset Google Play Store Mobile JKN

userName	score	at	content
Pengguna Google	5	2024-10-28 10:02:21	“Mantap mantap di tempat yang penting”
Pengguna Google	5	2024-10-28 09:53:23	”Mantap..aplikasi yang sangat berguna”
Pengguna Google	1	2024-10-28 09:51:19	“Tolong kpd developer untuk mengatasi masalah masuk ke aplikasi karena aplikasi mengganggap masuk menggunakan emulator atau HP ny di root.”
Pengguna Google	5	2024-10-28 09:46:10	“Mohon maaf kok aplikasi tdk bisa kebuka , ket anda sedang ofline gmn pak bu . Tolong seger adinperbaiki”
Pengguna Google	1	2024-09-08 00:41:17	” Data Perseta BPJS saya ada 4 anggota ko tibÂ² hilang 1 ya data nya jadi ga bisa di cek , ini kenapa ya pihak BPJS sedangkan iuran sudah sya bayarkan”

2.2 Preprocessing Data

In this section, there will be several processes will be carried out with the aim of cleaning text and preparing data, so the Mobile JKN data review can be processed by the machine learning models.

2.2.1 Labeling

In this section, the collected data will be labeled based on review scores ranging from 1 to 5 and categorized into three sentiment classes, namely “negative” for scores 1 and 2, “neutral” for scores 3, and “positive” for scores 4 and 5.

2.2.2 Lemmatization

This process converts words in the text data into their root forms using the WordNet Lemmatizer from python library. The goal of this process is to standardize variations of a word (e.g., “apk” and “app”) into a single base form (“aplikasi”), this will ensure consistency and improving analysis accuracy.

2.2.3 Stop Word Removal

In this step, common words that provide little value in sentiment analysis are removed. The removed words are based on a predefined list of Indonesian stop words. This process is implemented using the NLTK library, which includes stop words removal tools. This process aims to reduce noise in text data, allowing the model to focus on more meaningful and discriminating words that contribute to the sentiment analysis task.

2.2.4 Tokenization and Padding

The review text will be converted into a numeric sequence using the Keras Tokenizer library. Each word in the text will be put into a specific index based on its frequency, this known as tokenization process. Tokenization aims to prepare text data so that it can be accepted by machine learning models, especially for sentiment analysis.

Then padding process will be applied to ensure all review sequences have the same length, as reviews vary in length. This process will use the Pad Sequences function from the Keras library. This step is very important for further preprocessing in machine learning models.

2.2.5 Label Binarization

Three sentiment categories (negative, neutral and positive) are binarized into a numeric format using LabelBinarizer from python library. This process will ensure that the target variable is compatible with multiclass classification models and optimizing the training process.

2.3 Synthetic Minority Oversampling Technique

At this process, the data will be balanced across classes by adding synthetic samples to the minority classes. Synthetic Minority Oversampling Technique (SMOTE) will identify sample from the minority class and generate data with the K-Nearest Neighbor algorithm to balance data in the minority class. This process will be using Scikit-Learn from python library. SMOTE will create new additional synthetic data based on the nearest neighbors of the minority data. For example, if there is only a small amount of neutral class data, SMOTE will create additional samples that resemble the pattern of the neutral data. The oversampled synthetic data is combined with the majority class data to create a more balanced data distribution.

This step is only performed on the training data, so the test data still represents the actual conditions. This approach ensures the data set is more evenly distributed and improves the ability of the model to generalize between classes.

2.4 Long Short-Term Memory Model

This is the main stage of this research, that is building a Long Short-Term Memory model for sentiment analysis. LSTM is a type of Recurrent Neural Network that designed to process sequential data with time series, such as textual reviews, by capturing temporal relationship between words [17]. There are several steps to building the model architecture and to preparing the data for training.

2.4.1 Data Splitting

To ensure the reliability and generalization ability of the model, the dataset will be divided into two types, namely training data and validation data. The training data will use Mobile JKN review data sourced from Kaggle and the validation data will use Mobile JKN review sourced data from Google Play Store Review, this validation data will represent real world review and evaluate the model's generalization capability. This approach ensures an accurate sentiment analysis model.

2.4.2 LSTM Architecture

The LSTM architecture implemented in this research consists of the following layers, as shown in Table 3 below.

Table 3 LSTM Architecture

Layer	Configuration
Input Layer	Input shape (200,)
Embedding Layer	Embedding Dimension: 128
LSTM Layer	Units: 64, Dropout: 0.2
Dense Layer	Units: 64, Activation: ReLU
Output Layer	Units: 3, Activation: Softmax

This architecture is designed to handle sequential data more effectively. Based on Table 3 above, the embedding layer converts words into dense vector representations and captures the semantic relationships between them. The LSTM layer, with 64 units and a dropout rate of 0.2, processes the sequential data while preventing overfitting by ignoring random neurons during training. The dense layer, configured with 64 units and ReLU activation, ensures efficient learning of high-level features. Finally, the output layer, using a Softmax activation function, enables multi-class classification by producing probabilities for each sentiment class.

This architecture is chosen for handling imbalanced datasets, making it well-suited for the sentiment analysis tasks in this study. The model will capture the temporal dependencies within review text to predict sentiment in all class (negative, neutral and positive).

2.5 Model Evaluation

The evaluation of this LSTM model is conducted using a test dataset consisting of 20,000 recent reviews collected from the Google Play Store. The model test data results will be evaluated using performance metrics [18]. The performance metrics will be calculated using weighted accuracy to account for class imbalance This metrics will provide a more understandable picture of how big the impact of an imbalance class distribution. The model performance will be assessed based on the following performance metrics.

2.5.1 Accuracy

This metrics will evaluate how well the model predicts the correct output based on the input data. The following is the calculation formula is shown in the Eq. (1) below:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

2.5.2 Precision

This metrics will measure how accurate the model in making predictions in the three classes (negative, neutral and positive). The following is the calculation formula is shown in the Eq. (2) below:

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

2.5.3 Recall

This metrics will measure how correct the predictions are in the three classes (negative, neutral and positive). The following is the calculation formula is shown in the Eq. (3) below:

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

2.5.4 F1-Score

This metrics is the result of an average calculation of precision and recall. The following is the calculation formula is shown in the Eq. (4) below:

$$F1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

After the evaluation is carried out with these metrics, i tis hoped that the results will provide information regarding the performance of the LSTM model in handling imbalance dataset in the user review data.

3. RESULTS AND DISCUSSION

This section discusses the results of testing the sentiment analysis model that has been built in this research. The results will be discussed in several stages. Each phase will discuss the results of implementing SMOTE, LSTM model performance results and model comparison. The following is a more detailed discussion of the phases of this model.

In the first stage, training is carried out to get the best model based on the LSTM model. In the second phase, it contains a comparison of the LSTM model with the Naive Bayes and Random Forest models. The third stage of this step is a discussion of the comparison of performance results with previous research. This research does not involve a stemming process, as it focuses solely on analyzing the impact of SMOTE on the dataset.

Table 4 Dataset distribution before SMOTE

Sentiment Polarity	Amount Of the Data
Negative	29,553
Neutral	3,071
Positive	67,376

As can be seen in Table 4 above, the distribution of the dataset before implementing SMOTE was very unbalanced, the amount of data on neutral and negative sentiment was much less than on positive sentiment.. Then using SMOTE to balancing all classes, SMOTE will only be used on the training process.

Table 5 Dataset distribution after SMOTE

Sentiment Polarity	Amount Of the Data
Negative	67,376
Neutral	67,376
Positive	67,376

After applying SMOTE, as shown in Table 5, the distribution of the dataset becomes balanced across all sentiment classes, ensuring that each class contains the same number of samples. This balancing expected to improve the model’s ability to generalize across sentiment categories. The impact of this balanced dataset on model performance will be analyzed and discussed further below.

3.1 Training and Model Optimization

Hyperparameter testing will aim to find the best configuration for the accuracy performance of the LSTM architectural model. Testing was conducted to determine the most effective settings for this research.

3.1.1 Hyperparameter Selection

The selection of parameters was based on the results of experiment conducted during this study, to achieve the highest model performance. The LSTM model will be optimized using parameters, several parameters are chosen to optimize the model's training process more efficiently. The parameters are listed in Table 6 below.

Table 6 Training parameters

Parameter	Value
Loss Function	Categorical Crossentropy
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Epochs	20

This combination of parameters was carefully selected to optimize the training process in developing the sentiment classification. The categorical crossentropy loss function was chosen because it is well suited for multi-class classification tasks, ensuring precise accounting of the difference between predicted and actual class probabilities. The Adam optimizer was chosen for its adaptive learning capabilities, which allow efficient updates to the model's weights, accelerating convergence. The learning rate of 0.001 was set to balance the speed of learning and the stability of the optimization process, preventing overshooting the optimal solution. Batch size 32 was chosen to provide a balance between computational efficiency and model accuracy during training. Finally, the model was trained over 20 epochs to ensure sufficient exposure to the dataset for effective learning without overfitting.

3.1.2 Model Performance Evaluation

After training the LSTM model with the selected hyperparameters, the performance results are evaluated to find out how effective the model is in classifying imbalanced data. Confusion metrics are also used to gain a deeper understanding of the model's classification performance. The confusion metrics evaluates the distribution of predictions across true positives, true negatives, false positives, and false negatives. These metrics will provide a detailed picture of the model's ability to correctly classify each sentiment class (negative, neutral, and positive). This metrics is crucial for identifying specific areas where the model may struggle, such as misclassifying neutral sentiments as either negative or positive. The test results before and after using SMOTE will be shown in Table 7 below.

Table 7 Model performance

Algorithm	Accuracy	Precision	Recall	F1-Score
SMOTE	88%	90%	88%	89%
Non-SMOTE	91%	88%	91%	90%

The test results show that SMOTE can help overcome the problem of class imbalance by generating synthetic sample data for minority classes. However, its use causes an impact on overall performance. The decrease in accuracy and recall observed after implementing SMOTE suggests that although the model becomes more balanced in terms of class distribution, it may struggle to maintain the performance across all metrics. This trade-off reflects the challenge of improving generalization on imbalanced data while maintaining the power of the model and effectively distinguishing between different sentiment classes. Despite these changes, the model still performed well, with an F1 score of 88%, indicating that SMOTE had a positive impact on the model's ability to handle class imbalance without greatly reducing its classification ability.

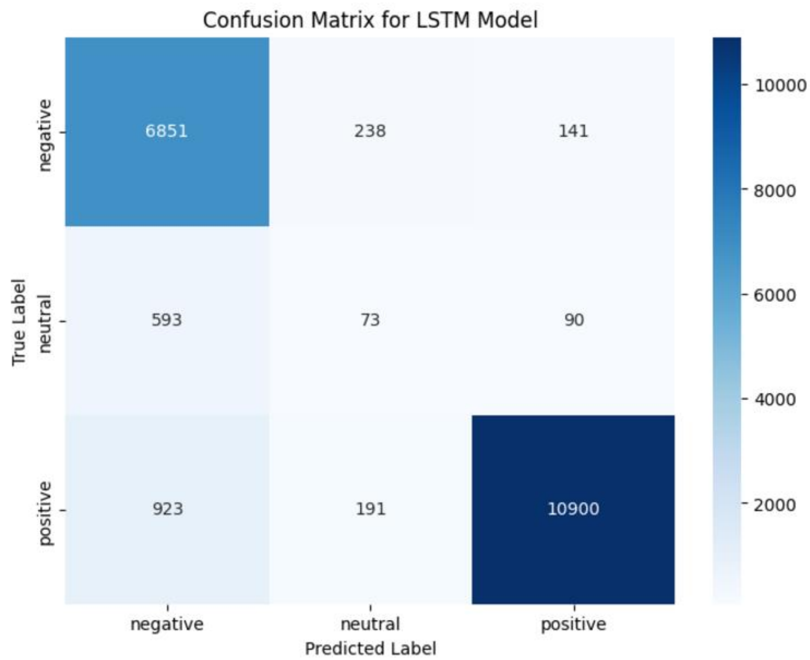

Figure 2 Confusion metrics for SMOTE – LSTM model

The confusion metrics results from the LSTM module shown in Figure 2 above, provide details of the model predictions for the three sentiment classes (negative, neutral and positive). This model has good performance in predicting negative and positive sentiment, as indicated by the high positive value in this category. However, the neutral class struggle as a model seems to have difficulty in classifying sentiments into negative and positive categories. This suggests that the model struggles to capture subtle differences in neutral sentiment, which is likely due to overlapping features between the neutral sentiment class and the other sentiment classes.

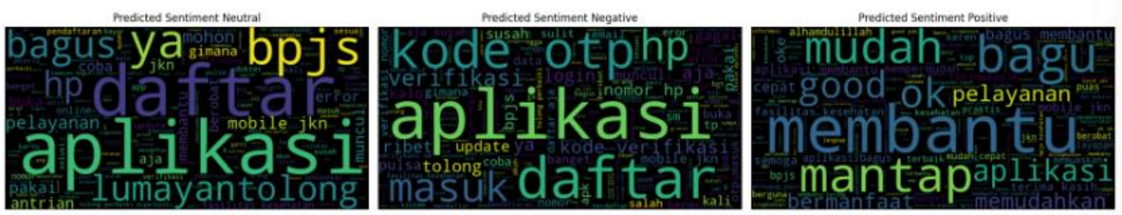

Figure 3 Word cloud sentiment analysis for each class

A word cloud visualization for each predicted sentiment, presented in Figure 3, highlights the words that appear most frequently in classified reviews. For positive sentiment, words such as “membantu” and “bagus” appear prominently, reflecting a positive user experience. Similarly, in the negative sentiment class, terms such as “susah” and “otp” dominate, showing general user frustration with login issues and technical issues. The neutral sentiment word cloud includes words such as “daftar” and “lumayan,” indicating mixed or moderate reactions that suggest adequate but improvable features.

This visualization provides insight into user feedback trends and reveals key terms associated with each sentiment class, aiding in a deeper understanding of the user satisfaction perspective on the object. By analyzing these frequently occurring keywords, app developers can prioritize improvements based on issues that relate to user preferences. Addressing login and OTP issues identified in negative sentiment can significantly improve user satisfaction and reduce frustration. This analysis allows for a balanced approach to improving app quality and ensures that both strengths and weaknesses of the app are addressed effectively.

3.2 Model Comparison

At this stage, the LSTM model prediction results that have been obtained will be compared with other models, namely Naive Bayes and Random Forest with the same dataset and using SMOTE. This stage aims to determine the difference in model performance between the deep learning model and other models, whether there is a significant difference or not. The comparison results between these models will be shown in Table 8 below.

Table 8 Model comparison

Methods	Accuracy	Precision	Recall	F1-Score
LSTM + SMOTE	88%	90%	88%	89%
Naïve Bayes + SMOTE	70%	80%	70%	74%
Random Forest + SMOTE	83%	81%	83%	81%

Based on the results, the LSTM model outperforms both Naive Bayes and Random Forest overall on each performance metrics, especially in precision, recall and f1-score. The LSTM model achieved 86% accuracy, 90% precision, 86% recall, and 88% f1 score. Meanwhile, the Naive Bayes model shows the lowest performance in classifying imbalanced data with 70% accuracy, 80% precision, 70% recall and 74% f1-score. Random Forest has quite good performance in classifying imbalanced data with 83% accuracy, 81% precision, 83% recall and 81% f1-score, but it is still inferior to LSTM compared to precision and f1-score.

These results show that the LSTM model, with the ability to capture temporal dependency patterns in sentiment data and provides the best performance for sentiment classification in this research, especially when handling imbalanced data.

4. CONCLUSIONS

The conclusion of this study provides information that the use of SMOTE can help overcome class imbalance by increasing precision, recall, and F1 score in minority classes. However, the results also show a trade-off, as the overall accuracy and recall of the model are slightly lower than when SMOTE is not used. However, the LSTM model with SMOTE showed strong performance, achieving 88% accuracy, 90% precision, 88% recall, and 89% F1 score. These results show that SMOTE contributes to balancing the distribution of each class without significantly reducing classification ability. Additionally, the use of word clouds based on sentiment predictions provides valuable insights into user feedback. Positive sentiment often highlights ease of use and helpful features, while negative sentiment focuses on login issues and technical errors with OTP codes. Neutral sentiment reflects areas where the application is performing well but there is still room for improvement.

This study shows the importance of effectively handling imbalanced datasets while maintaining high model performance. Mobile JKN application developers can leverage these insights to improve user experience by addressing key issues identified through sentiment analysis. Future research could explore alternative techniques for dealing with class imbalance, such as ensemble learning or advanced sampling methods, and focus on optimizing models for better generalization on imbalanced datasets, especially on classification on neutral classes.

REFERENCES

[1] L. N. Hapsari and N. Rokhman, "Anomaly Detection of Hospital Claim Using Support

Vector Regression," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 18, no. 1, p. 1, 2024, doi: 10.22146/ijccs.91857.

[2] A. Ramadhan and H. R. Sari, "BPJS kesehatan luncurkan aplikasi 'Mobile JKN,'" ANTARA. [Online]. Available: https://www.antaranews.com/berita/665018/bpjs-kesehatan-luncurkan-aplikasi-mobile-jkn

[3] W. M. E. Lumi, R. A. Musak, F. F. Tumiwa, M. Y. Waworuntu, and W. S. Surya*, "Edukasi Tentang Penggunaan Aplikasi Mobile JKN Pada Pasien Rawat Jalan RSUD Anugerah Tomohon," *J. Pengabdi. Kpd. Masy. Nusant.*, vol. 4, no. 3, pp. 1620–1626, 2023.

[4] S. F. Maulani *et al.*, "Analisis Costumer Service Post Transaction Melalui Google Ulasan Pada PT Indah Logistik Cargo cabang Serang : Product Tracking," vol. 4, pp. 5395–5404, 2024.

[5] J. Menengah, "Peraturan Presiden (Perpres) Nomor 49 Tahun 2024 tentang Strategi Nasional Perlindungan Konsumen," *Bpk Ri*, no. 198461, 2024.

[6] N. L. Ratniasih and N. W. Ninik Jayanti, "Sentiment Analysis of Stakeholder Satisfaction Measurement," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 16, no. 2, p. 217, 2022, doi: 10.22146/ijccs.72245.

[7] H. Utami, "Analisis Sentimen dari Aplikasi Shopee Indonesia Menggunakan Metode Recurrent Neural Network," *Indones. J. Appl. Stat.*, vol. 5, no. 1, p. 31, 2022, doi: 10.13057/ijas.v5i1.56825.

[8] R. Md, R. D. Restiyan, and H. Irsyad, "Analisis Sentimen Masyarakat terhadap Perilaku Lawan Arah yang diunggah pada Media Sosial Youtube Menggunakan Naïve Bayes," *BANDWIDTH J. Informatics Comput. Eng.*, vol. 02, no. 02, pp. 75–83, 2024.

[9] I. K. A. G. Wiguna, P. Sugiartawan, I. G. I. Sudipa, and I. P. Y. Pratama, "Sentiment Analysis Using Backpropagation Method to Recognize the Public Opinion," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 16, no. 4, p. 423, 2022, doi: 10.22146/ijccs.78664.

[10] M. Z. Rahman, Y. A. Sari, and N. Yudistira, "Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 11, pp. 5120–5127, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[11] E. Tasci, Y. Zhuge, K. Camphausen, and A. V. Krauze, "Bias and Class Imbalance in Oncologic Data—Towards Inclusive and Transferrable AI in Large Scale Oncology Data Sets," *Cancers (Basel).*, vol. 14, no. 12, 2022, doi: 10.3390/cancers14122897.

[12] E. F. Saputra and M. R. Pribadi, "Analisis Sentimen Komentar Pada Kanal Youtube The Lazy Monday Menggunakan Algoritma Naive Bayes," *MDP Student Conf.*, vol. 2, no. 1, pp. 17–23, 2023, doi: 10.35957/mdp-sc.v2i1.4283.

[13] N. S. Sediatmoko, Y. Nataliani, and I. Suryady, "Sentiment Analysis of Customer Review Using Classification Algorithms and SMOTE for Handling Imbalanced Class," *Indones. J. Inf. Syst.*, vol. 7, no. 1, pp. 38–52, 2024, doi: 10.24002/ijis.v7i1.8879.

[14] Eskiyaturrofikoh and R. R. Suryono, "Analisis Sentimen Aplikasi X Pada Google Play Store Menggunakan Algoritma Naïve Bayes Dan Support Vector Machine (Svm)," *JIPI(Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 3, pp. 1408–1419, 2024, [Online]. Available: https://www.jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/view/5392

[15] N. Istiqamah and M. Rijal, "Klasifikasi Ulasan Konsumen Menggunakan Random Forest dan SMOTE," *J. Syst. Comput. Eng.*, vol. 5, no. 1, pp. 66–77, 2024, doi: 10.61628/jsce.v5i1.1061.

[16] C. Candra, K. W. Chandra, and H. Irsyad, "Efektifitas SMOTE dalam Mengatasi Imbalanced Class Algoritma K-Nearest Neighbors pada Analisis Sentimen terhadap Starlink," *J. Ilmu Komput. dan Inform.*, vol. 4, no. 1, pp. 31–42, 2024, doi: 10.54082/jiki.132.

[17] G. Tamami and M. Arifin, "Penggunaan LSTM dalam Membangun Prediksi Penjualan

untuk Aplikasi Laptop Lens," vol. 14, no. 2, pp. 301–308, 2024.
[18] M. Fadli and R. A. Saputra, "Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke," *JT J. Tek.*, vol. 12, no. 02, pp. 72–80, 2023, [Online]. Available: http://jurnal.umt.ac.id/index.php/jt/index