Journal of Information Technology and Applications Research
Volume 1 Nomor 2 November 2025
e-ISSN: 3089-8013; p-ISSN:3089-8021, Page 18-28
DOI: https://doi.org/10.63956/jitar.v1i2.32
Available online at: https://jitar.org/index.php/jitar/article/view/32

Implementation of Support Vector Machine (SVM) Algorithm in Sentiment Analysis of the Mobile JKN Application Using the CRISP-DM Approach

Nurdiyanto Yusuf [1], Ilham Bintang Prakoso Siregar [2]
Gunadarma University

Abstract
Technological advances have facilitated various sectors, including healthcare in Indonesia. One innovation is the Mobile JKN application developed by BPJS Kesehatan to ease access to National Health Insurance (JKN) services. This study aims to analyze user sentiment toward the Mobile JKN application using 1,000 reviews from the Google Play Store. The analysis was carried out using the Support Vector Machine (SVM) method with the Cross-Industry Standard Process for Data Mining (CRISP-DM) approach. The data was collected through scraping and processed through pre-processing stages, TF-IDF, data visualization, and splitting into training and testing sets to classify sentiment into positive and negative categories. The deployment stage was carried out by creating a visualization website using Streamlit, which includes dashboards, word clouds, various chart types, text tables, and a manual sentiment prediction feature based on user input. The website is available at: https://sentimen-mobilejkn-ebd8w9s59ueehw6wjqhoct.streamlit.app/. After all calculations were performed, the results showed that the implementation of the Support Vector Machine (SVM) algorithm to analyze user sentiment of the Mobile JKN application based on reviews from the Google Play Store using a 70% training and 30% testing data split achieved an Accuracy of 86%, Precision of 86%, Recall of 86%, and F1-Score of 87%. These findings may support future development of the Mobile JKN service.
Keywords: *Application, Mobile JKN, Sentiment analysis, Support Vector Machine, Crisp-DM, Google Play Store..*

INTRODUCTION

The development of information technology has brought significant convenience across various sectors, including healthcare services in Indonesia. One of the government's innovations through BPJS Kesehatan is the *Mobile JKN* application, which allows participants to access services such as healthcare facility registration, online queues, doctor consultations, and membership verification digitally. This transformation shortens service processes that were previously only available through direct visits to branch offices or healthcare facilities.

However, the effectiveness and quality of the *Mobile JKN* services depend heavily on user experience and perception. User reviews available on the Google Play Store serve as valuable information sources to measure satisfaction and identify potential issues. Machine learning–based sentiment analysis can automatically classify these reviews into

Received: September 15, 2025; Accepted: October 22, 2025; Published: November 19, 2025
*Corresponding author : nurdiyanto@staff.gunadarma.ac.id

positive or negative sentiments, with the Support Vector Machine (SVM) algorithm being one of the most reliable methods for text classification tasks.

The main problem addressed in this study is how to accurately classify user review sentiments of the *Mobile JKN* application using the SVM method so that the results can provide valuable insights for developers to improve service quality.

The objective of this research is to analyze user sentiments from Google Play Store reviews of the *Mobile JKN* application, classify them into positive and negative categories using the SVM method, and present the results through interactive visualizations.

Previous studies have demonstrated the superiority of the Support Vector Machine (SVM) method in sentiment analysis. Iskandar and Nataliani (2021) compared Naïve Bayes, SVM, and k-NN algorithms for aspect-based gadget sentiment analysis, in which SVM achieved an average accuracy of 96.43% [1]. Furthermore, Singgalen (2022) conducted sentiment analysis on Borobudur Temple visitor reviews using NBC, DT, and SVM within the CRISP-DM framework, showing that SVM achieved the highest accuracy at 99.41% [2]. In addition, Purwanti and Sugiyono (2024) applied SVM for sentiment analysis of the free lunch program on the X social media platform, obtaining 94% accuracy with mostly positive sentiments [3]. These studies collectively confirm that SVM consistently performs well in classifying textual data across various domains, making it highly relevant for sentiment analysis of *Mobile JKN* application reviews.

Based on these prior studies, SVM has proven to be an effective method for text classification and capable of handling large-scale data efficiently. The novelty of this research lies in applying the SVM method specifically to sentiment analysis of the *Mobile JKN* application using the most recent dataset from the Google Play Store, supported by interactive visualizations to facilitate interpretation of the results.

METHODS

This study involved participants who were users of the *Mobile JKN* application and had provided reviews on the Google Play Store. The inclusion criteria required that the reviews be written in Indonesian, contain opinions or experiences related to the application, and be published in April 2025. A total of 1,000 reviews that met these criteria were analyzed. The dependent variable was the users' sentiment (positive or negative), while the independent variables included the review text, star rating score, and keywords produced during text preprocessing.

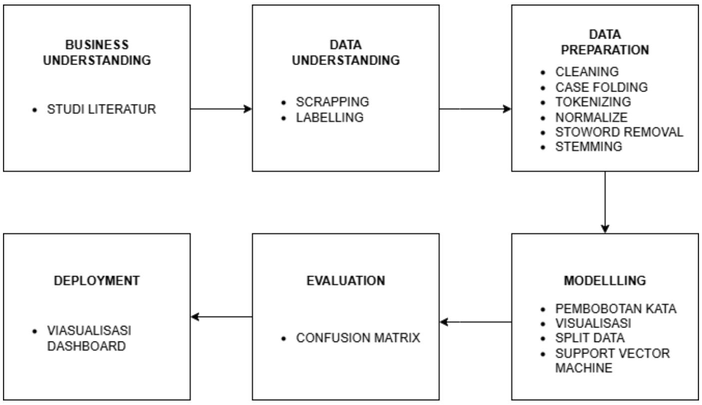

Figure 1. Research stages

The preprocessing stage consisted of several sub-processes: case folding, tokenization, normalization, stopword removal, and stemming. These steps ensured that the text data were clean, standardized, and ready for feature extraction and model training.

The research procedure included several stages: data collection through scraping, sentiment labeling based on user rating scores, preprocessing for text cleaning, word weighting using the Term Frequency–Inverse Document Frequency (TF-IDF) method, splitting the dataset into training and testing sets, and building a classification model using the Support Vector Machine (SVM) algorithm.

The data analysis phase was conducted using a confusion matrix to calculate accuracy, precision, recall, and F1-score as performance evaluation metrics. The final results were visualized through an interactive Streamlit-based dashboard that displayed sentiment distribution, word clouds, and comparative charts across sentiment categories.

RESULTS

Business Understanding

The Business Understanding phase began with a comprehensive literature review focusing on the concepts of sentiment analysis, the application of the Support Vector Machine (SVM) algorithm, and the stages of data mining based on the CRISP-DM framework. This review involved examining previous studies relevant to sentiment analysis of application reviews and the implementation of SVM in text classification tasks.

The purpose of this stage was to understand the strengths, weaknesses, and best practices of the SVM method that could be adapted to this research. Based on the literature findings, the research objectives, data limitations, and analytical approaches were determined, focusing on the sentiment classification of *Mobile JKN* application reviews.

The scope of the study includes the use of Indonesian-language data, the retrieval of reviews from the Google Play Store during a specific period, and the evaluation of model performance using accuracy, precision, recall, and F1-score as key performance metrics.

Data Understanding

The Data Understanding stage consisted of two main processes: Data Collection and Data Labeling.

1. Data Collection
   This process utilized the google-play-scraper library to automatically collect data from the Google Play Store, eliminating the need for manual retrieval. Through this process, a total of 1,000 reviews that met the research criteria were obtained. These criteria included reviews written in Indonesian that contained users' opinions or experiences in using the Mobile JKN application.

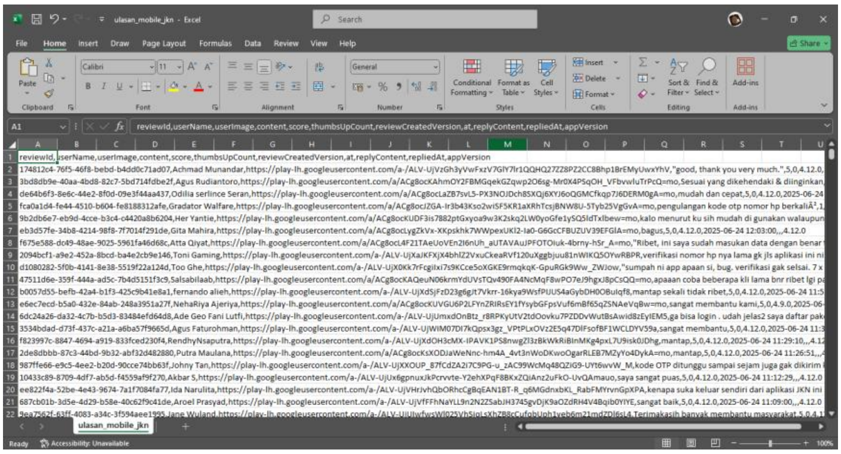

Figure 2. Result of Data Scraping

2. Data Labeling
   After data collection, a labeling process was performed to assign sentiment categories to each review. Labels were determined based on the user's star rating: reviews with scores of 3 to 5 were classified as positive sentiment, while those with scores of 1 to 2 were classified as negative sentiment. This stage was crucial because the assigned labels served as the foundation for the model's learning and classification process.

	content	score	label
0	tiba tiba log out.. udah masukin NIK dengan be...	1	Negatif
1	memudahkan masyarakat	5	Positif
2	kembangkan terus untuk kesehatan yang lebih baik	5	Positif
3	ok	5	Positif
4	bolak blik buka apk jkn suruh login ulang trus...	1	Negatif
5	disuruh buat passwor yg susah diingat dan logi...	1	Negatif
6	tidak bisa verif wajah, kode otp sms sangat la...	1	Negatif
7	sempat bingung dgn APL ini,, karna baru pertam...	5	Positif
8	oke	5	Positif
9	oke	5	Positif

Figure 3. Result of Data Labeling

Data Preparation

The Data Preparation stage involved transforming raw data collected from the Google Play Store into a clean and structured dataset ready for modeling. This process consisted of several key steps conducted sequentially to ensure data quality and reduce noise that could affect classification performance.

1. Data Cleaning

   In this step, irrelevant elements such as punctuation marks, numbers, special symbols, and emojis were removed to eliminate noise from the text. For example, excessive use of exclamation marks or special symbols, which do not contribute meaningfully to sentiment, was cleaned.

	content	cleaning
0	tiba tiba log out.. udah masukin NIK dengan be...	tiba tiba log out udah masukin nik dengan bena...
1	memudahkan masyarakat	memudahkan masyarakat
2	kembangkan terus untuk kesehatan yang lebih baik	kembangkan terus untuk kesehatan yang lebih baik
3	ok	ok
4	bolak blik buka apk jkn suruh login ulang trus...	bolak blik buka apk jkn suruh login ulang trus...
5	disuruh buat passwor yg susah diingat dan logi...	disuruh buat passwor yg susah diingat dan logi...
6	tidak bisa verif wajah, kode otp sms sangat la...	tidak bisa verif wajah kode otp sms sangat lam...
7	sempat bingung dgn APL ini,, karna baru pertam...	sempat bingung dgn apl ini karna baru pertama ...
8	oke	oke
9	oke	oke

Figure 4. Result of Data Cleaning

2. Case Folding

   All characters were converted to lowercase to prevent differences in meaning due to capitalization. For instance, the words “BPJS” and “bpjs” were treated as identical after this process.

	cleaning	case_folding
0	tiba tiba log out udah masukin nik dengan bena...	tiba tiba log out udah masukin nik dengan bena...
1	memudahkan masyarakat	memudahkan masyarakat
2	kembangkan terus untuk kesehatan yang lebih baik	kembangkan terus untuk kesehatan yang lebih baik
3	ok	ok
4	bolak blik buka apk jkn suruh login ulang trus...	bolak blik buka apk jkn suruh login ulang trus...
5	disuruh buat passwor yg susah diingat dan logi...	disuruh buat passwor yg susah diingat dan logi...
6	tidak bisa verif wajah kode otp sms sangat lam...	tidak bisa verif wajah kode otp sms sangat lam...
7	sempat bingung dgn apl ini karna baru pertama ...	sempat bingung dgn apl ini karna baru pertama ...
8	oke	oke
9	oke	oke

Figure 5. Result of Case Folding

3. Tokenization

   This proces split review sentences into individual word units (tokens). For example, the sentence “Aplikasinya sangat membantu” was tokenized into ["aplikasinya", "sangat", "membantu"]. Tokenization allows each word to be analyzed and processed individually.

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

	case_folding	tokenisasi
0	tiba tiba log out udah masukin nik dengan bena...	[tiba, tiba, log, out, udah, masukin, nik, den...
1	memudahkan masyarakat	[memudahkan, masyarakat]
2	kembangkan terus untuk kesehatan yang lebih baik	[kembangkan, terus, untuk, kesehatan, yang, le...
3	ok	[ok]
4	bolak blik buka apk jkn suruh login ulang trus...	[bolak, blik, buka, apk, jkn, suruh, login, ul...
5	disuruh buat passwor yg susah diingat dan logi...	[disuruh, buat, passwor, yg, susah, diingat, d...
6	tidak bisa verif wajah kode otp sms sangat lam...	[tidak, bisa, verif, wajah, kode, otp, sms, sa...
7	sempat bingung dgn apl ini karna baru pertama ...	[sempat, bingung, dgn, apl, ini, karna, baru, ...
8	oke	[oke]
9	oke	[oke]

Figure 6. Result of Tokenization

4. Normalization
   Normalization was applied to convert non-standard words into standard forms following Indonesian language rules. For example, “nggak” and “gk” were normalized to “tidak”. This ensured consistency among words with similar meanings but different spellings.

	case_folding	normalize
0	tiba tiba log out udah masukin nik dengan bena...	tiba tiba log out sudah memasukkan nik dengan ...
1	memudahkan masyarakat	memudahkan masyarakat
2	kembangkan terus untuk kesehatan yang lebih baik	kembangkan terus untuk kesehatan yang lebih baik
3	ok	oke
4	bolak blik buka apk jkn suruh login ulang trus...	bolak balik buka apakah jkn suruh login ulang ...
5	disuruh buat passwor yg susah diingat dan logi...	disuruh buat passwor yang susah diingat dan lo...
6	tidak bisa verif wajah kode otp sms sangat lam...	tidak bisa verifikasi wajah kode otp sms sanga...
7	sempat bingung dgn apl ini karna baru pertama ...	sempat bingung dengan apl ini karena baru pert...
8	oke	oke
9	oke	oke

Figure 7. Result of Normalization

5. Stopword Removal
   Stopwords are common words (e.g., “dan”, “yang”, “di”, “ke”) that frequently appear but do not contribute significantly to sentiment meaning. These words were removed so that the model could focus on more informative words relevant to sentiment detection.

	normalize	stopwordsremoval
0	tiba tiba log out sudah memasukkan nik dengan ...	[tiba, tiba, out, memasukkan, nik, dinyatakan,...
1	memudahkan masyarakat	[memudahkan, masyarakat]
2	kembangkan terus untuk kesehatan yang lebih baik	[kembangkan, kesehatan, baik]
3	oke	[]
4	bolak balik buka apakah jkn suruh login ulang ...	[bolak, balik, suruh, giliran, nik, sesuai, sa...
5	disuruh buat passwor yang susah diingat dan lo...	[disuruh, passwor, susah, diingat, biometrik, ...
6	tidak bisa verifikasi wajah kode otp sms sanga...	[sangat, macam]
7	sempat bingung dengan apl ini karena baru pert...	[sempat, bingung, apl, pertama, kali, memakai,...
8	oke	[]
9	oke	[]

Figure 8. Result of Stopword Removal

6. Stemming
   Stemming converted words to their root forms. For instance, “membantu”, “dibantu”, and “membantunya” were all reduced to the root word “bantu”. This process helped reduce redundancy and produced a more compact and structured dataset.

	stopwordsremoval	stemming
0	[tiba, tiba, out, memasukkan, nik, dinyatakan,...	[tiba, tiba, out, memasukkan, nik, dinyatakan,...
1	[memudahkan, masyarakat]	[memudahkan, masyarakat]
2	[kembangkan, kesehatan, baik]	[kembangkan, kesehatan, baik]
3	[]	[]
4	[bolak, balik, suruh, giliran, nik, sesuai, sa...	[bolak, balik, suruh, giliran, nik, sesuai, sa...
5	[disuruh, passwor, susah, diingat, biometrik, ...	[disuruh, passwor, susah, diingat, biometrik, ...
6	[sangat, macam]	[sangat, macam]
7	[sempat, bingung, apl, pertama, kali, memakai,...	[sempat, bingung, apl, pertama, kali, memakai,...
8	[]	[]
9	[]	[]

Figure 9. Result of Stemming

Modelling

The Modeling stage aimed to build a sentiment classification model using the Support Vector Machine (SVM) algorithm based on the preprocessed dataset. This stage consisted of several essential steps to ensure optimal model performance.

1. Term Weighting
   Term weighting was conducted using the Term Frequency–Inverse Document Frequency (TF-IDF) method, which assigns numerical weights to words based on their frequency within a review (term frequency) and how rare the word appears across the entire dataset (inverse document frequency).

	abal	abalabal	abjad	acces	acount	adahal	adakah	adanya	admin	administrasi	...	wkwk	woi	work	yadan	yamg	yamh	yasampai	yasy	youtube	zolim
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 1531 columns

Figure 10. Result of Term Weighting

2. Visualization
   Before training the model, a word cloud visualization was generated to display the most frequent words for each sentiment class. Positive reviews commonly featured words like “easy”, “fast”, and “helpful”, while negative reviews included terms such as “error”, “difficult”, and “slow”. These visualizations provided an initial overview of the distinguishing vocabulary within each sentiment category.

Figure 11. Result of Visualization

3. Data Splitting
   The dataset was divided into two subsets, a training set and a testing set, using different split ratios: 90:10, 80:20, and 70:30. This step aimed to evaluate the model's ability to generalize to unseen data. The best-performing ratio was later selected based on model evaluation metrics.

```
Jumlah data latih: 900
Jumlah data uji: 100
```
```
Jumlah data latih: 800
Jumlah data uji: 200
```
```
Jumlah data latih: 700
Jumlah data uji: 300
```

Figure 12. Result of Data Splitting

4. Support Vector Machine (SVM)
   The SVM algorithm was used to build the sentiment classification model. SVM operates by finding the optimal hyperplane that best separates the data points into two classes, positive and negative sentiments, with the maximum margin.

Table 1. SVM Model Accuracy Results

Data Split Ratio	SVM Model Accuracy
90% training data and 10% testing data	0.85
80% training data and 20% testing data	0.84
70% training data and 30% testing data	0.86

After obtaining the results from the implementation of the Support Vector Machine (SVM) model using three different data split ratios, it can be concluded that the best-performing configuration was the 70:30 ratio, consisting of 70% training data and 30% testing data. This ratio achieved the highest accuracy of 0.86 or 86%. Following this stage, the data were prepared for use in the subsequent evaluation process.

Evaluation

In the evaluation stage, the model's performance in predicting sentiment data was assessed based on the results obtained from the previous process. The evaluation used the dataset with the highest-performing ratio from the modeling stage, which was the 70:30 split, consisting of 70% training data and 30% testing data. The results of this evaluation are presented using a classification report and a confusion matrix. The confusion matrix is used to evaluate the performance of a machine learning model and to calculate several performance metrics, including accuracy, precision, recall, and the F1-score[4].

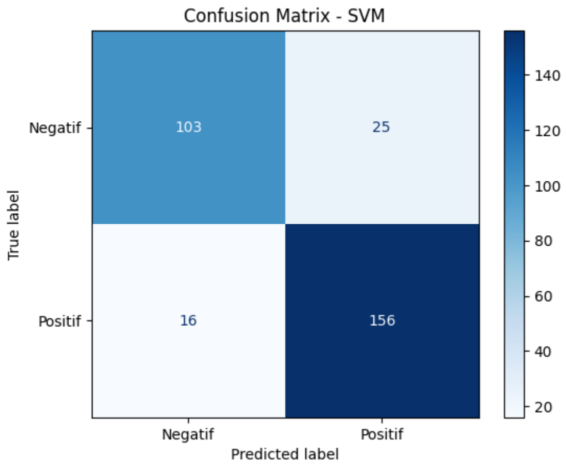

Figure 12. Confusion Matrix Visualization

Based on the values obtained from the confusion matrix visualization, the following calculations were performed to determine accuracy, precision, recall, and F1-score as indicators of the model's performance.

Accuracy
Accuracy = (156 + 103) / (156 + 103 + 25 + 16) = 259 / 300 = 0.863 ≈ 0.86 × 100% = 86%

Precision
Negative Class = 103 / (103 + 16) = 103 / 119 = 0.865 ≈ 0.87 × 100% = 87%
Positive Class = 156 / (156 + 25) = 156 / 181 = 0.861 ≈ 0.86 × 100% = 86%

Recall
Negative Class = 103 / (103 + 25) = 103 / 128 = 0.804 ≈ 0.80 × 100% = 80%
Positive Class = 156 / (156 + 16) = 156 / 172 = 0.906 ≈ 0.91 × 100% = 91%

F1-score
Negative Class = (2 × 0.80 × 0.87) / (0.80 + 0.87) = 1.392 / 1.67 = 0.833 ≈ 0.83 × 100% = 83%
Positive Class = (2 × 0.91 × 0.86) / (0.91 + 0.86) = 1.5646 / 1.77 = 0.883 ≈ 0.88 × 100% = 88%

After calculating the accuracy, precision, recall, and F1-score, the overall and per-class model performance could be evaluated. Additionally, macro-average values were computed to show the average performance across all classes without considering class imbalance, while weighted-average values accounted for the proportion of data in each class, providing a more representative evaluation of the model's performance.

Classification Report:

	precision	recall	f1-score	support
Negatif	0.87	0.80	0.83	128
Positif	0.86	0.91	0.88	172
accuracy			0.86	300
macro avg	0.86	0.86	0.86	300
weighted avg	0.86	0.86	0.86	300

Figure 13. Classification Report Results

The figure above presents the classification report generated from the Support Vector Machine (SVM) model. Based on the results, the SVM algorithm achieved an accuracy of 86%, precision of 86%, recall of 86%, and an F1-score of 86% in classifying user sentiment toward the Mobile JKN application, based on reviews collected from the Google Play Store.

Deployment

The deployment stage involved implementing the trained Support Vector Machine (SVM) model into a web-based application to allow users to interact with the system directly. In this study, the deployment was carried out using Streamlit, a Python-based web framework designed to simplify the presentation of models and data visualizations. Streamlit is known for its clean, user-friendly interface and its ability to run without requiring advanced front-end development skills. It provides various built-in interactive widgets such as file uploaders, sliders, text inputs, checkboxes, and radio buttons.

When a user interacts with the application interface, the entire Python script is re-executed from start to finish. This re-execution concept is crucial to ensure the proper management of the application's state and responsiveness to user input [5]. The process began by saving the trained SVM model and TF-IDF vectorization in pickle (.pkl) format so that the model could be reloaded without retraining.

The deployed dashboard includes several main features, such as a data input page for uploading review datasets in CSV format and a text input field for manual sentiment prediction. Each input, whether from a file or text entry, undergoes the same preprocessing stages used in the training phase, including cleaning, tokenization, normalization, stopword removal, and stemming. The text is then transformed using TF-IDF vectorization before being classified by the SVM model.

The prediction results are displayed in real-time on the same page, showing the sentiment category and the confidence level of each prediction. Additionally, the dashboard presents visualizations of sentiment analysis results, including bar charts and pie charts for sentiment distribution, word clouds for dominant positive and negative words, and classification tables displaying review text, predicted labels, and probability scores.

After successful testing in a local environment, the application was deployed to Streamlit Cloud, making it publicly accessible via a web link without requiring Python installation on the user's side. Through this deployment, developers and stakeholders can easily and interactively monitor user sentiment toward the Mobile JKN application and utilize the results as a data-driven decision-support tool. The final deployed sentiment analysis dashboard for the Mobile JKN application using the Support Vector Machine (SVM) method can be accessed via the following URL: https://sentimen-mobilejkn-ebd8w9s59ueehw6wjqhoct.streamlit.app/

CONCLUSION

This study successfully processed a structured dataset for sentiment analysis of the Mobile JKN application. The review data were obtained from the Google Play Store using the google-play-scraper library and analyzed through the Cross-Industry Standard Process for Data Mining (CRISP-DM), which consists of Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment stages. The analysis results show that the Support Vector Machine (SVM) algorithm was able to classify positive and negative sentiments with good accuracy. The model was tested using three data split scenarios: 90:10, 80:20, and 70:30, where the 70:30 ratio produced the highest accuracy of 86%. Therefore, it can be concluded that the 70:30 ratio using the SVM algorithm is more optimal compared to the other ratios. However, the model still has limitations, particularly in detecting negative sentiments, indicating the need for further improvements in future research.

SUGGESTIONS

Future research is expected to improve model accuracy by exploring other classification algorithms such as Naïve Bayes, K-Nearest Neighbor (KNN), and Random Forest. In addition, the number of review datasets for the Mobile JKN application should be increased to achieve a more balanced distribution between positive and negative data. This enhancement would enable the model to process and detect sentiments more accurately, particularly in identifying negative sentiment categories.

REFERENCES

Iskandar, J.W. and Natalia, Y., Comparison of Naïve Bayes, SVM, and k-NN for Aspect-Based Sentiment Analysis of Gadgets, JURNAL RESTI (Journal of Systems Engineering and Information Technology), 5(2), pp.1120–1126. DOI: https://doi.org/10.29207/resti.v5i6.3588, 2021.

Singgalen, Y.A., Performance Analysis of NBC, DT, and SVM Algorithms in Classifying Visitor Review Data of Borobudur Temple Based on CRISP-DM, Building of Informatics, Technology and Science (BITS), 4(3), pp.1634–1646. DOI: https://doi.org/10.47065/bits.v4i3.2766, 2022.

Purwanti, Z. and Sugiyono, Text Mining Modeling for Sentiment Analysis of the Free Lunch Program on X Social Media Using the Support Vector Machine (SVM) Algorithm, Jurnal Indonesia: Management Informatics and Communication, 5(3), pp.3065–3077. DOI: https://journal.stmiki.ac.id/index.php/jimik/, 2024.

Munawaroh, A., Ridhoi, R. and Rudiman, Sentiment Analysis Using Naïve Bayes Based on Orange for IKN Development Risk, Jurnal Mahasiswa Teknik Informatika, 8(1), pp.587–592. DOI: https://doi.org/10.36040/jati.v8i1.8454, 2024.

Hastono, W., Aini, N., Karno, A.S.B. and Rere, L.M.R., Machine Learning Methods for Predicting Manure Management Emissions, Jurnal Nasional Teknik Elektro dan Teknologi Informasi, 11(22), pp.2460–5719. DOI: http://dx.doi.org/10.22146/jnteti.v11i2.2586, 2022.