Sentiment Analysis of Grab App Reviews with Machine Learning Approach

Vihi Atina[1*], Prattana Srisuk[2]
Duta Bangsa University[1], Thai Global Business Administration Technological College[2]
*Correspondence Email : vihi_atina@udb.ac.id

ABSTRACT

Technological advances in online transportation services such as Grab facilitated user mobility. User reviews of the application were a valuable source of information for developers to improve service quality and for users to make decisions regarding service use. This research aimed to analyze the sentiment of Grab application user reviews using a machine learning approach. The system development method used in this research was the Agile method with the stages of Planning, Iterative Development, and Testing. The machine learning algorithms applied were Random Forest, Support Vector Machine (SVM), and Naive Bayes. The results of sentiment analysis of Grab application reviews were in the form of classification of reviews into positive, neutral, and negative sentiments. The test results showed that the Random Forest algorithm had the highest accuracy rate of 95.14%. This indicated that Random Forest was effective in identifying sentiment patterns in review data.

KEYWORDS Grab, Machine Learning, Sentiment Analysis, Reviews

INTRODUCTION

Advances in digital technology brought significant changes in various aspects of life, including in the transportation sector. Online transportation service technology such as Grab became a popular solution for people in meeting their needs, especially in facilitating daily mobility (Li and Huang, 2020). With features that made it easy for users, such as fast vehicle booking, digital payments, and responsive customer service, Grab attracted the attention of millions of users in various regions (Chen and Wang, 2022).

However, as usage increased, user reviews of apps became increasingly important as a valuable source of information (Zhang et al., 2019). User reviews reflected users' satisfaction, experience, and complaints with the app services provided (Sun et al., 2021). For app developers, these reviews served as valuable feedback to identify areas that required improvement, thereby improving service quality. Meanwhile, for new users, these reviews could be a reference in determining whether the service was worth using (Singh et al., 2021).

Given the growing volume of reviews, manual methods for analyzing user sentiment became inefficient. Therefore, a more effective and accurate approach, such as machine learning, was needed to classify the sentiment of these reviews (Rahman and Salam, 2022). Machine learning algorithms had a number of advantages, such as the ability to automatically process large amounts of data, recognize complex patterns, and continue to learn and adapt from new data (Kim and Ahn, 2021). Another advantage of machine learning algorithms was their ability to work quickly and efficiently, especially in handling unstructured text data (Liu and Lee, 2022). With the right training process, these algorithms could increase the accuracy of the analysis as the processed data grew. Algorithms such as Naive Bayes, Support Vector Machine (SVM), and Random Forest were not only able to identify positive, negative, or neutral sentiments, but also provided more accurate predictions than conventional methods (Gupta et al., 2020). This was particularly useful in analyzing the context of online transportation applications, where user feedback evolved over time.

Based on this background, the purpose of this research was to analyze the sentiment of Grab application user reviews with a machine learning approach. By using a machine learning approach, sentiment analysis could be done more quickly, accurately, and efficiently, providing significant benefits for developers to improve services and for users to understand service quality based on reviews. This research was expected to contribute to the development of online transportation applications that were more responsive to user needs and expectations, while maximizing the potential of machine learning technology in big data analysis.

RESEARCH METHOD

This research used the Agile system development method which allowed rapid and flexible iteration in the development of sentiment analysis systems. The Agile method was chosen because of its ability to adapt to changing user needs and allow continuous improvement of the system based on feedback (Hoda et al., 2020). The stages in the Agile approach used in this research included planning, iterative development, and testing. The following were the detailed stages carried out:

1. Planning
   In the planning stage, the functional and non-functional requirements of the sentiment analysis system were identified. These needs were obtained from literature studies and analysis of research objectives.
2. Iterative Development
   Development was carried out in iterations or sprints. Each iteration focused on developing and refining system components, including data collection, data preprocessing, and model development.
3. Testing
   Testing was carried out continuously in each iteration to ensure each feature ran as expected. Model performance testing used accuracy, precision, recall, and F1-score metrics to evaluate model performance.

RESULT AND DISCUSSION

1. Planning

At this stage, planning was made to develop the Sentiment Analysis of Grab Application Reviews with a Machine Learning Approach by identifying and defining the functional and non-functional requirements of the system to be developed.

a. Functional Needs

Functional needs were related to the features that were to be presented. The Sentiment Analysis of Grab App Reviews could display analysis results according to positive, negative, and neutral criteria.

b. Non-Functional Needs

Non-functional requirements consisted of hardware requirements and software requirements. The hardware requirements used to create the Grab App Review Sentiment Analysis were a laptop with an Intel Core i5 processor, 8 GB RAM, a 14-inch monitor, and a mouse. The software requirements used were the Windows 11 operating system, Jupyter Notebook, Python, and a web browser.

2. Iterative Development

a. Data Collection

Data collection was done automatically by scraping Grab review data from the Google Play Store. The data collected included content, score, year, month, and day. The review data collected comprised 500 datasets. The results of the review data scraping process could be seen in Figure 1.

	content	score	at	appVersion	Year	Month	Day
499	Mm kkk ni kk ss 3, ga tau ya s. 6-9	3	2024-09-15 10:31:26	5.318.0	2024	9	15
498	good attitude, good service	5	2024-09-15 10:34:13	5.310.0	2024	9	15
497	terima kasih banyak	5	2024-09-15 10:34:17	5.322.0	2024	9	15
496	bagus simple	5	2024-09-15 10:36:27	5.322.0	2024	9	15
495	oke	5	2024-09-15 10:40:44	5.322.0	2024	9	15
...	...	...	...	...	...	...	...
4	terpercaya	5	2024-09-18 17:11:17	5.322.0	2024	9	18
3	cepat	5	2024-09-18 17:14:47	5.322.0	2024	9	18
2	Tengah malam, pencahayaan kurang, buru-buru ma...	1	2024-09-18 17:43:04	None	2024	9	18
1	mantabs,sehat selalu pak,terimakasih	5	2024-09-18 18:06:31	5.322.0	2024	9	18
0	Pelayanan terlalu lama	1	2024-09-18 18:16:20	5.322.0	2024	9	18

500 rows × 7 columns

Figure 1. Data Scraping Results

b. Preprocessing

From the results of the data scraping process, positive, neutral, and negative sentiment labeling was carried out based on the score value. Score values of more than 3 were labeled positive, score values of 3 were labeled neutral, and score values of less than 3 were labeled negative. The results of the data labeling process could be seen in Figure 2.

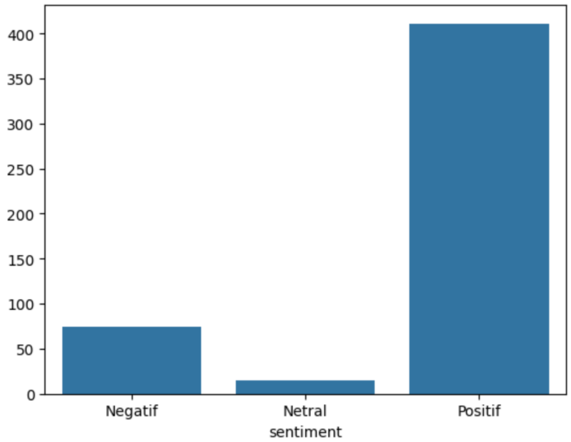

Figure 2. Data Labeling Result

Based on the results of data labeling, the percentage distribution of sentiment could be seen in Figure 3.

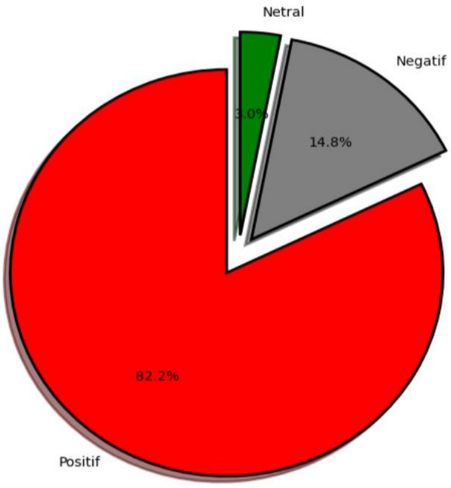

Figure 3. Sentiment Share Result

After the labeling process, the next stage of data preprocessing included the tokenizing process, filtering process, and stemming process. The tokenizing process was done by separating each word connected by a space character into individual words that were collected in an array. The result of tokenizing was an array of data that could be used for the next stage, namely the filtering process. The filtering process was done by eliminating words that had no effect on the sentiment process. The result of filtering was the elimination of words that were detected as stopwords. The results of the filtering process were used in the next stage, namely the stemming process. In the stemming process, it was done by converting words with affixes into base words to facilitate the weighting process later. The results of the preprocessing process could be seen in Figure 4.

	content	score	Year	Month	Day	sentiment	content_token	stemmed	text_string
499	Mm kkk ni kk ss 3, ga tau ya s. 6-9	3	2024	9	15	Netral	[Mm, kkk, ni, kk, ss, 3, ,, ga, tau, ya, s., 6-9]	[mm, kkk, ni, kk, ss, 3, , ga, tau, ya, s, 6-9]	
498	good attitude, good service	5	2024	9	15	Positif	[good, attitude, ,, good, service]	[good, attitude, , good, service]	good attitude good service
497	terima kasih banyak	5	2024	9	15	Positif	[terima, kasih]	[terima, kasih]	terima kasih
496	bagus simple	5	2024	9	15	Positif	[bagus, simple]	[bagus, simple]	bagus simple
495	oke	5	2024	9	15	Positif	[oke]	[oke]	
...	...	...	...	...	...	...	...	...	...
4	terpercaya	5	2024	9	18	Positif	[terpercaya]	[percaya]	percaya
3	cepat	5	2024	9	18	Positif	[cepat]	[cepat]	cepat
2	Tengah malam, pencahayaan kurang, buru-buru ma...	1	2024	9	18	Negatif	[Tengah, malam, ,, pencahayaan, ,, buru-buru, ...	[tengah, malam, , cahaya, , buru, , suruh, sel...	tengah malam cahaya buru suruh selfie mantap
1	mantabs,sehat selalu pak,terimakasih	5	2024	9	18	Positif	[mantabs, ,, sehat, ,, terimakasih]	[mantabs, , sehat, , terimakasih]	mantabs sehat terimakasih
0	Pelayanan terlalu lama	1	2024	9	18	Negatif	[Pelayanan]	[layan]	layan

500 rows × 9 columns

Figure 4. Preprocessing Result

c. Model Development

At this stage, the data from the preprocessing stage was then formed into a classification model by applying machine learning algorithms. The machine learning algorithms applied were Random Forest, Support Vector Machine (SVM), and Naive Bayes. The training and testing process was carried out by changing the test size to compare the three machine learning algorithms. The results of the machine learning model development process could be seen in Figure 5.

	data_train	data_tes	random_fores	SVM	naive_bayes
0	90	10	95.967742	91.935484	94.354839
1	80	20	95.141700	91.497976	91.902834
2	70	30	94.594595	90.810811	88.918919
3	60	40	93.117409	89.676113	87.854251
4	50	50	92.706645	89.951378	91.410049
5	40	60	91.891892	89.324324	89.864865
6	30	70	89.930556	85.995370	87.384259
7	20	80	87.234043	81.560284	79.331307
8	10	90	77.747748	75.765766	64.414414

Figure 5. Results Application of Machine Learning

The results of the training and testing process of 3 machine learning algorithms, namely Random Forest, Support Vector Machine (SVM), and Naive Bayes, could be seen in Figure 6.

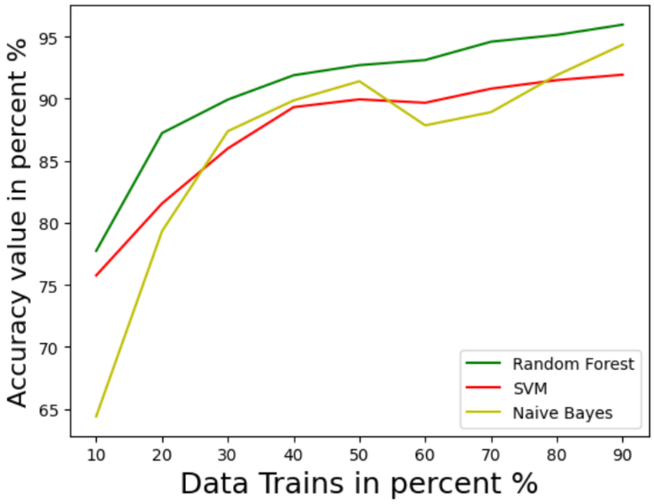

Figure 6. Machine Learning Algorithm Accuracy

3. Testing

At the testing stage, model performance testing was carried out using accuracy, precision, recall, and F1-score metrics to evaluate the performance of the Random Forest, Support Vector Machine (SVM), and Naive Bayes algorithms. Random Forest test results could be seen in Figure 7. SVM test results could be seen in Figure 8. Naive Bayes test results could be seen in Figure 9.

```
RandomForestClassifier Accuracy: 0.951417004048583
RandomForestClassifier Precision: 0.9521843809055195
RandomForestClassifier Recall: 0.951417004048583
RandomForestClassifier f1_score: 0.9515070259291372

confusion_matrix:
 [[83  1  5]
 [ 0 73  2]
 [ 2  2 79]]

              precision    recall  f1-score   support

     Negatif       0.98      0.93      0.95        89
      Netral       0.96      0.97      0.97        75
     Positif       0.92      0.95      0.93        83

    accuracy                           0.95       247
   macro avg       0.95      0.95      0.95       247
weighted avg       0.95      0.95      0.95       247
```

Figure 7. Random Forest Testing Results

```
SVC Accuracy: 0.9149797570850202
SVC Precision: 0.9219380682824361
SVC Recall: 0.9149797570850202
SVC f1_score: 0.915742208466819

confusion_matrix:
 [[86  0  3]
 [ 6 67  2]
 [10  0 73]]
              precision    recall  f1-score   support

     Negatif       0.84      0.97      0.90        89
      Netral       1.00      0.89      0.94        75
     Positif       0.94      0.88      0.91        83

    accuracy                           0.91       247
   macro avg       0.93      0.91      0.92       247
weighted avg       0.92      0.91      0.92       247
```

Figure 8. SVM Testing Results

```
MultinomialNB Accuracy: 0.9190283400809717
MultinomialNB Precision: 0.9254868017289283
MultinomialNB Recall: 0.9190283400809717
MultinomialNB f1_score: 0.9198082504384051

confusion_matrix:
 [[84  4  1]
 [ 0 72  3]
 [ 1 11 71]]
              precision    recall  f1-score   support

     Negatif       0.99      0.94      0.97        89
      Netral       0.83      0.96      0.89        75
     Positif       0.95      0.86      0.90        83

    accuracy                           0.92       247
   macro avg       0.92      0.92      0.92       247
weighted avg       0.93      0.92      0.92       247
```

Figure 9. Naive Bayes Testing Results

The test results showed that the Random Forest algorithm had the highest accuracy rate of 95.14%. This indicated that Random Forest was effective in identifying sentiment patterns in review data (Dandekar et al., 2022).

CONCLUSION

The Agile system development method applied in the development of the Grab App Review sentiment analysis system consisted of 3 stages, namely Planning, Iterative Development, and Testing. The results of sentiment analysis of Grab application reviews were in the form of classification of reviews into positive, neutral, and negative sentiments. The results of testing 3 machine learning algorithms showed that the Random Forest algorithm had the highest accuracy rate of 95.14%. This indicated that Random Forest was effective in identifying sentiment patterns in review data. Further research could explore the latest algorithms and more advanced natural language processing techniques to improve the accuracy of sentiment analysis.

REFERENCES

Chen, L., & Wang, Z. (2022). *User Reviews and Service Quality Analysis of Ride-Hailing Apps: A Case Study of Grab*. Journal of Transportation Research, 45(2), 123-135.

Dandekar, G. B., et al. (2022). *Comparative Performance of Machine Learning Algorithms for Sentiment Analysis.* Journal of Computer Science and Technology.

Gupta, R., Kumar, A., & Sharma, V. (2020). *Machine Learning Approaches for Sentiment Analysis: A Review*. International Journal of Data Mining & Knowledge Management, 12(3), 56-78.

Hoda, R., Noble, J., & Marshall, S. (2020). *Agile in Practice: The Impact of Agile Methods on Productivity and Performance*. Springer.

Kim, S., & Ahn, J. (2021). *Comparative Study of Machine Learning Algorithms for Sentiment Analysis on Ride-Hailing Services*. IEEE Access, 9, 45678-45689.

Li, X., & Huang, Y. (2020). *The Impact of Digital Platforms on Urban Mobility: A Study on Grab and its Competitors.* Journal of Urban Studies, 30(4), 212-230.

Liu, P., & Lee, J. (2022). *Text Mining Techniques for Analyzing User Feedback in Transportation Apps*. Journal of Intelligent Transportation Systems, 19(1), 98-112.

Rahman, M., & Salam, F. (2022). *Efficiency of Machine Learning Algorithms in Sentiment Analysis of Online Reviews.* International Journal of Computer Science, 14(3), 231-243.

Singh, A., Verma, R., & Patel, K. (2021). *Leveraging Machine Learning for Sentiment Analysis of Ride-Sharing Applications*. Data Science Review, 8(2), 67-85.

Sun, Y., Wang, H., & Zhao, L. (2021). *Evaluating Customer Feedback with Machine Learning: A Focus on Ride-Hailing Services*. Journal of Artificial Intelligence Research, 34(7), 445-460.

Zhang, Q., Chen, Y., & Li, M. (2019). *User Sentiment Analysis of Ride-Hailing Applications Using Machine Learning Techniques*. Expert Systems with Applications, 76, 216-224.