Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Sentiment Analysis of Transjakarta App Reviews Using the Naive Bayes Algorithm

Ahmad Fikri Haikal[1]*, Yosep Nuryaman[2]

[1,2]*Universitas Bina Sarana Informatika*
fikaalm26@gmail.com[1*], yosep.ynu@bsi.ac.id[2]

Abstract

In utilizing digital technology in the transportation sector, Transjakarta has introduced a mobile-based application to facilitate public mobility. The number of Transjakarta passengers has increased significantly, making it important to know whether users are satisfied with the application or not. This study aims to classify sentiments and identify the aspects and issues that frequently arise in user reviews of the Transjakarta app using the Naive Bayes algorithm. This study employs the CRISP-DM methodology. Analysis was conducted by scraping 1,000 Google Play Store review data based on MOST RELEVANT, following preprocessing, TextBlob data labeling, TF-IDF weighting, and oversampling (SMOTE) methods. The implementation of the Naive Bayes algorithm with an 80:20 resulted in 684 positive data and 315 negative data, yielding a model accuracy of 78%. For the positive sentiment class, the precision was 82%, recall was 85%, and the F1-score was 84%. For the negative sentiment class, the precision was 67%, recall was 62%, and the F1-score was 65%. Based on the visualization, the words that frequently appear in positive reviews are “bagus”, “lengkap”, and “mudah”, and negative reviews are “ribet”, “susah”, and “error”.

Keywords: *crisp-dm, naive bayes, sentiment analysis, text mining, transjakarta*

1. Introduction

In this ever-evolving digital age, the use of mobile applications has become an important element in the daily lives of urban communities, including in the field of public transportation [1]. One concrete example of the use of digital technology in the transportation sector is the Transjakarta app. Transjakarta itself is the first Bus Rapid Transit (BRT) system in Southeast and South Asia, operating since 2004 in Jakarta, and has become a crucial element in the public transportation infrastructure managed by the DKI Jakarta government [2].

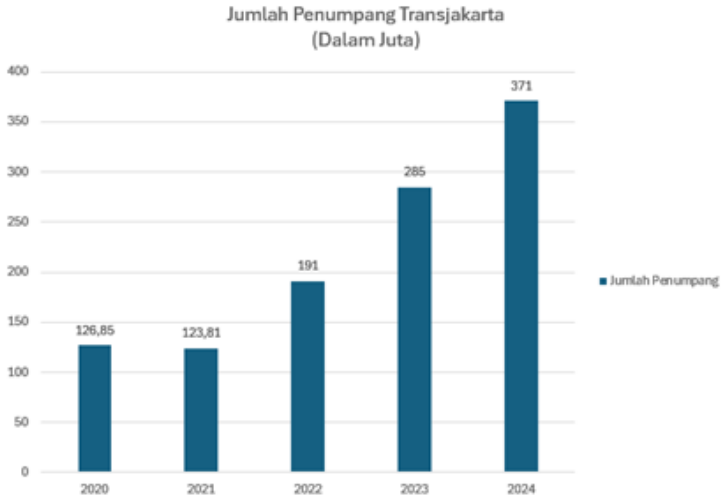

Fig. 1: Number of Transjakarta Passengers

Fig 1 shows data on the number of Transjakarta passengers, which has increased year on year, proving that the people of Jakarta are highly dependent on Transjakarta as their main mode of transportation. As the number of users increases, so does the number of reviews of the Transjakarta app, both in the form of appreciation and complaints. This is why it is important to know whether users are satisfied with the

app or not. Dissatisfaction or mistakes in service, if not addressed immediately, can risk reducing public trust in Transjakarta services, which could result in many passengers switching back to using private vehicles [2]. User reviews serve as a source of information/data that can provide direct insight into issues arising in the application [3]. Therefore, to see how users perceive the application, sentiment analysis was conducted based on user reviews taken from the Google Play Store.

Sentiment analysis is a field of study that analyzes people's opinions, sentiments, attitudes, evaluations, surveys, and emotions towards entities such as issues, events, topics, and their attributes [4]. Several previous studies have discussed related studies, including; Research from [2] on Consumer Satisfaction Analysis of Transjakarta Services analyzed general service satisfaction, such as the comfort of bus stops and punctuality, but did not touch on aspects of digital services such as applications. Then, research from [5] on User Sentiment Analysis of Jakarta's Public Transportation System Performance Using the Naive Bayes Algorithm to classify the sentiment of netizen comments on public transportation in the Jakarta area with a high accuracy rate of 94.22%. Then, research from [6] on Sentiment Analysis of the KRL Access Online Transportation Application Using the Naive Bayes Method provided results with an accuracy of up to 84.00% for test data on positive reviews in Indonesian on the selection of land transportation applications on smartphones.

From the results of several studies, it can be concluded that the Naive Bayes algorithm has proven to be successful and accurate in classifying data in various contexts. By utilizing the Naive Bayes algorithm, researchers will group user reviews into positive and negative sentiment categories, thereby providing an overview of their satisfaction or dissatisfaction. Therefore, this study aims to analyze the sentiment of Transjakarta app user reviews by utilizing the Naive Bayes algorithm, classifying user opinions into positive and negative sentiment categories, and providing data-driven recommendations for improving the quality of Transjakarta services.

2. Research Methodology

The method used in this study is the Cross Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM is a framework that is often used in data science and data mining projects [7]. This method consists of six main stages, namely Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment, as shown in Fig 2.

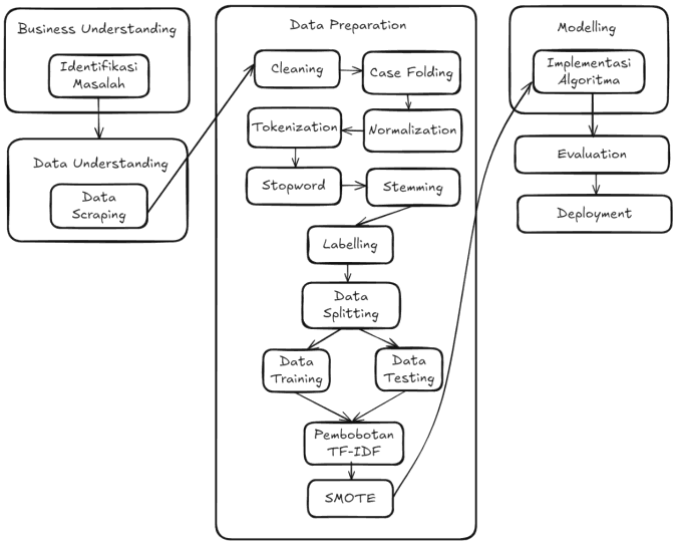

Fig. 2: Research Stages

At the business understanding stage, we identified key issues by classifying user sentiment toward the Transjakarta app and conducting a literature review to strengthen the theoretical basis and understand appropriate methods such as sentiment analysis and the Naive Bayes algorithm.

The next stage is data understanding, which includes the process of collecting data through web scraping techniques from the Google Play Store platform. The data obtained consists of 1000 Transjakarta app reviews that have been filtered as “MOST RELEVANT” by the system.

In the data preparation stage, the collected data are processed for modeling through several important steps, including data cleaning, case folding, normalization, tokenization, stopword removal, stemming, data labeling, data splitting, TF-IDF weighting, and oversampling. The initial step, data cleaning, is performed to eliminate irrelevant characters such as symbols, numbers, and punctuation marks from the dataset. This step is essential because raw data often contains incomplete, inaccurate, or poorly structured information. The second stage of case folding, the process of converting all uppercase text to lowercase text [8]. This is done to standardize characters in documents or text data [9]. The third stage of normalization, the process of converting non-standard terms or abbreviations into words that conform to standard/correct spelling or dictionaries [10]. The fourth stage of tokenization is the process of splitting/separating text into individual words. This breaks the text down into small units called tokens [11] The fifth stage is stopword removal. This process removes common words that have no significant meaning or are considered less important in text analysis [12]. The sixth stage of stemming, the process of mapping and analyzing a word into its basic form, where words containing affixes are filtered into the standard form of the word [13] The seventh stage of data labeling is the process of automatically labeling sentiments as positive or negative using TextBlob tools. TextBlob is a well-known lexicon-based sentiment analysis library model available in Python, which offers simpler text processing [14]. The eighth stage is data splitting, a process that divides the data into two parts, namely training data and testing data. The ninth stage is TF-IDF weighting, a process of assigning weights to each term based on its importance in the document relative to the entire data set. The weights

are assigned to assess or score the frequency of occurrence of a word (TF) and the frequency of occurrence of a word or term in all documents (IDF) [10]. And the final stage is oversampling (SMOTE), a process to balance class distribution by sampling from minority classes and generating synthetic samples in those minority classes. SMOTE is an oversampling technique designed to address class imbalance by utilizing interpolation to generate synthetic samples from underrepresented classes [15].

Next is the modeling stage, which involves applying the Naive Bayes algorithm as the main classification model due to its simplicity and effectiveness in analyzing text. The model will be created and the data from the preprocessing stage will be trained.

Then, in the evaluation stage, the performance of the Naive Bayes algorithm model will be tested. The evaluation method applied is the confusion matrix. In the confusion matrix, the test will calculate accuracy, recall, precision, and f1-score values, which will be visualized in percentage form.

And in the final stage of deployment, the model evaluation results are used to understand user sentiment towards the Transjakarta application. The classification model, which has been evaluated using a confusion matrix, serves as a reference for assessing the model's ability to accurately categorize reviews. To identify the dominant factors or main themes in each sentiment category, wordcloud-based visualization is used. Wordclouds help to find the words or phrases that appear most frequently in positive and negative reviews, which can reflect the main concerns or issues of users.

3. Results and Discussion

The results obtained from applying the Naive Bayes method in sentiment analysis of Transjakarta app reviews, based on the stages in the CRISP-DM framework, which include Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The following are the steps in the research process using the CRISP-DM framework:

A. Business Understanding

At this stage, an analysis was conducted on user reviews of the Transjakarta application on Google Play Store, with the aim of understanding user perceptions of the application. The algorithm used is Naive Bayes with the Python programming language. The results of this sentiment analysis are expected to provide information to the app developers and Transjakarta management regarding the quality of the app's services based on user reviews, so that it can be used as a basis for decision-making regarding improvements and future app development.

B. Data Understanding

At this stage, the review data collection process was carried out using web scraping techniques from the Google Play Store platform, collecting 1,000 review data as research material using the google-play-scraper library automatically. The data collected was saved in a CSV file format, and only the Review ID, Username, Rating, Review Text, and Date columns were taken.

Table 1: Data Collection Results

Review ID	Username	Rating	Review Text	Date
07c1be5f-4b81-47b4-8bc9-e6e840c85f4f	Adi Candra	2	Masih perlu banyak peningkatan, peta yg loading ny lama dan gak bisa zoom out, posisi bus suka muncul hilang dan gak akurat, app kurang enak tampilannya. Kelebihan dari app yg lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute ny. Tapi tetap trafi yg terbaik	2024-08-29 17:30:57
…	…	…	…	…
J753dd01-22tf-3yfj-b5e3-40gasg3575f	Ci Kakya	5	Skrg jadi lebih mudah cari rute , halte, dan jadwal bus...	2024-11-21 13:41:19

C. Data Preparation

The cleaned data subsequently undergoes preprocessing, which comprises several stages, including case folding, normalization, tokenization, stopword removal, stemming, data labeling, TF-IDF weighting, and SMOTE-based oversampling.

1. Data Cleaning
   This process is to correct, remove characters/symbols/emojis, and delete unclear words from the results that researchers previously obtained from scraping.

Table 2: Data Cleaning Results

Before	After
Masih perlu banyak peningkatan, peta yg loading ny lama dan gak bisa zoom out, posisi bus suka muncul hilang dan gak akurat, app kurang	Masih perlu banyak peningkatan peta yg loading ny lama dan gak bisa zoom out posisi bus suka muncul hilang dan gak akurat app kurang enak

enak tampilannya. Kelebihan dari app yg lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute ny. Tapi tetap trafi yg terbaik	tampilannya Kelebihan dari app yg lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute ny Tapi tetap trafi yg terbaik
…	…
Skrg jadi lebih mudah cari rute , halte, dan jadwal bus...	Skrg jadi lebih mudah cari rute halte dan jadwal bus

2. Case Folding
   This process will convert all uppercase text to lowercase text. This is to ensure that all text is processed under the same conditions.

Table 3: Case Folding Results

Before	After
Masih perlu banyak peningkatan peta yg loading ny lama dan gak bisa zoom out posisi bus suka muncul hilang dan gak akurat app kurang enak tampilannya Kelebihan dari app yg lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute ny Tapi tetap trafi yg terbaik	masih perlu banyak peningkatan peta yg loading ny lama dan gak bisa zoom out posisi bus suka muncul hilang dan gak akurat app kurang enak tampilannya kelebihan dari app yg lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute ny tapi tetap trafi yg terbaik
…	…
Skrg jadi lebih mudah cari rute halte dan jadwal bus	skrg jadi lebih mudah cari rute halte dan jadwal bus

3. Normalization
   This process serves to replace non-standard words with standard words using the kamus_slag that researchers can obtain from the kaggle.com website to normalize the words in the reviews in this study.

Table 4: Normalization Results

Before	After
masih perlu banyak peningkatan peta yg loading ny lama dan gak bisa zoom out posisi bus suka muncul hilang dan gak akurat app kurang enak tampilannya kelebihan dari app yg lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute ny tapi tetap trafi yg terbaik	masih perlu banyak peningkatan peta yang loading nya lama dan tidak bisa zoom out posisi bus suka muncul hilang dan tidak akurat app kurang enak tampilannya kelebihan dari app yang lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute nya tapi tetap trafi yang terbaik
…	…
skrg jadi lebih mudah cari rute halte dan jadwal bus	sekarang jadi lebih mudah cari rute halte dan jadwal bus

4. Tokenization
   This process serves to separate each review from the document used to separate review sentences into words and remove blank spaces.

Table 5: Tokenization Results

Before	After
masih perlu banyak peningkatan peta yang loading nya lama dan tidak bisa zoom out posisi bus suka muncul hilang dan tidak akurat app kurang enak tampilannya kelebihan dari app yang lama cuma bisa lihat posisi bus real time tanpa perlu pilih rute nya tapi tetap trafi yang terbaik	['masih', 'perlu', 'banyak', 'peningkatan', 'peta', 'yang', 'loading', 'nya', 'lama', 'dan', 'tidak', 'bisa', 'zoom', 'out', 'posisi', 'bus', 'suka', 'muncul', 'hilang', 'dan', 'tidak', 'akurat', 'app', 'kurang', 'enak', 'tampilannya', 'kelebihan', 'dari', 'app', 'yang', 'lama', 'cuma', 'bisa', 'lihat', 'posisi', 'bus', 'real', 'time', 'tanpa', 'perlu', 'pilih', 'rute', 'nya', 'tapi', 'tetap', 'trafi', 'yang', 'terbaik']
…	…
sekarang jadi lebih mudah cari rute halte dan jadwal bus	['sekarang', 'jadi', 'lebih', 'mudah', 'cari', 'rute', 'halte', 'dan', 'jadwal', 'bus']

5. Stopword Removal
   This process removes meaningless words that have no meaning and only contain informative words. Researchers use the nltk library for tools in this process.

Table 6: Stopword Removal Results

Before	After
['masih', 'perlu', 'banyak', 'peningkatan', 'peta', 'yang', 'loading', 'nya', 'lama', 'dan', 'tidak', 'bisa', 'zoom', 'out', 'posisi', 'bus', 'suka', 'muncul', 'hilang', 'dan', 'tidak', 'akurat', 'app', 'kurang', 'enak', 'tampilannya', 'kelebihan', 'dari', 'app', 'yang', 'lama', 'cuma', 'bisa', 'lihat', 'posisi', 'bus', 'real', 'time', 'tanpa', 'perlu', 'pilih', 'rute', 'nya', 'tapi', 'tetap', 'trafi', 'yang', 'terbaik']	['peningkatan', 'peta', 'loading', 'nya', 'zoom', 'out', 'posisi', 'bus', 'suka', 'muncul', 'hilang', 'akurat', 'app', 'enak', 'tampilannya', 'kelebihan', 'app', 'lihat', 'posisi', 'bus', 'real', 'time', 'pilih', 'rute', 'nya', 'trafi', 'terbaik']
…	…
['sekarang', 'jadi', 'lebih', 'mudah', 'cari', 'rute', 'halte', 'dan', 'jadwal', 'bus']	['mudah', 'cari', 'rute', 'halte', 'jadwal', 'bus']

6. Stemming
   This process serves to convert affixes into their base words/original forms. Researchers use the Sastrawi library because the review data is in Indonesian.

Table 7: Stemming Results

Before	After
['peningkatan', 'peta', 'loading', 'nya', 'zoom', 'out', 'posisi', 'bus', 'suka', 'muncul', 'hilang', 'akurat', 'app', 'enak', 'tampilannya', 'kelebihan', 'app', 'lihat', 'posisi', 'bus', 'real', 'time', 'pilih', 'rute', 'nya', 'trafi', 'terbaik']	tingkat peta loading nya zoom out posisi bus suka muncul hilang akurat app enak tampil lebih app lihat posisi bus real time pilih rute nya trafi baik

…	…
['mudah', 'cari', 'rute', 'halte', 'jadwal', 'bus']	mudah cari rute halte jadwal bus

7. Data Labelling
   Data labeling was performed using the TextBlob library as a tool to automatically label sentiment as positive or negative. When the polarity is greater than 0, it indicates positive sentiment, while a polarity less than or equal to 0 indicates negative sentiment. Fig 3 shows the amount of data in each sentiment class obtained from the labeling results using TextBlob.

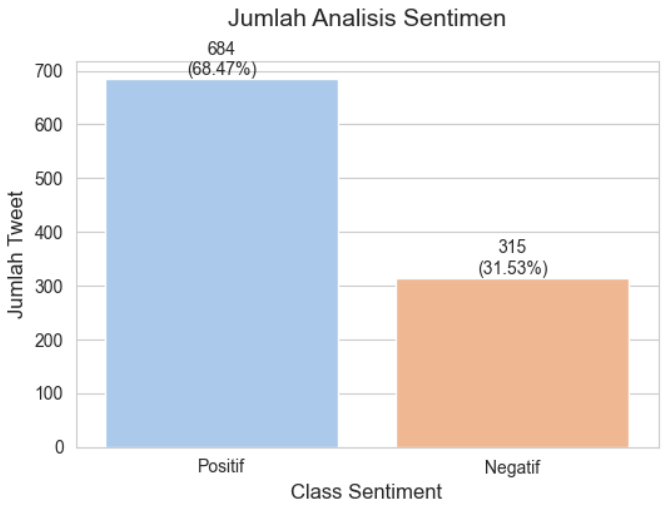

Fig. 3: Comparison of Positive and Negative Class Results

8. TF-IDF Weighting
   This process involves weighing each word in the dataset and assessing it based on how important it is in a document. It is also useful for converting data into a vector (numeric) matrix so that machines can process it. However, before weighing, the data is divided into 80% training data and 20% testing data.

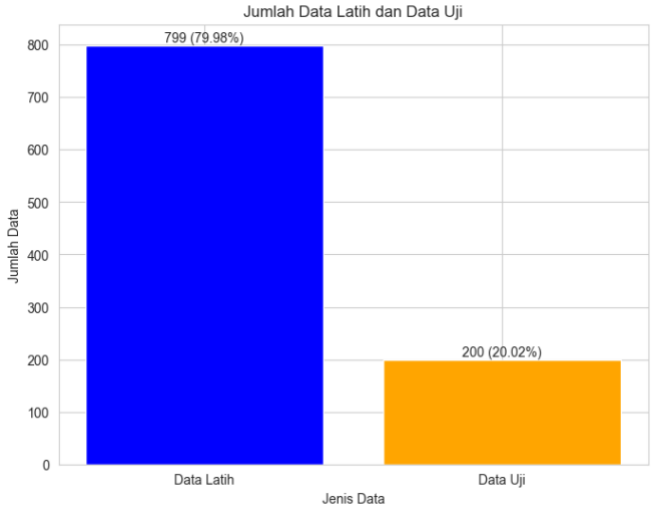

Fig. 4: Visualization of Training and Testing Data Distribution Diagram

9. SMOTE
   The final process in data preparation is SMOTE. This process balances imbalanced data. This is useful so that the data is not biased towards the minority class.

D. Modelling

At this stage, researchers begin building models using the Naive Bayes Algorithm with the MultinomialNB type because it is suitable for text-based data such as reviews. The stages of the Naive Bayes classification process are to initialize MultinomialNB() for the Naive Bayes model used. Then train the model with input data X_train and label y_train and to perform model testing predictions with input data X_test.

E. Evaluation

After completing the modeling process, the researchers proceeded to the next stage, which was to evaluate the output results from the implementation of the Naive Bayes algorithm model. This evaluation was conducted on 200 test data from Transjakarta app reviews. The evaluation used a confusion matrix to obtain more accurate model results. The confusion matrix produces values for model accuracy, recall, precision, and F1 score.

Table 8: Classification Report Results from the Naive Bayes Algorithm

Sentiment	Test Accuracy	Precision	Recall	F1-Score
Positive	0.775	0.82	0.85	0.84
Negative	0.775	0.67	0.62	0.65

F. Deployment

In this final stage, researchers visualize frequently occurring words with the aim of providing analysis results in a format that can be used practically by end users. The visualization uses the WordCloud library, which represents the frequency of words appearing in various positive and negative sentiment categories. WordCloud effectively presents textual information clearly and intuitively, with the size of the words indicating their frequency of appearance in reviews within a particular sentiment category.

Fig. 5: Positive Sentiment WordCloud

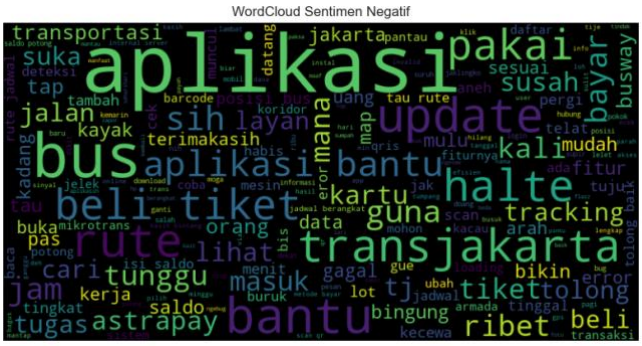

Fig. 6: Negative Sentiment WordCloud

4. Conclusion

Based on the results of sentiment analysis of Transjakarta app reviews using the Naive Bayes algorithm, which was conducted based on 1000 user review data from Google Play Store platform, there were 684 positive review sentiments and 315 negative review sentiments. Then, the results of the classification report using the Naive Bayes algorithm obtained an accurate value of 78%. For the positive sentiment class, the precision value was 82%, the recall was 85%, and the f1-score was 84%, while for the negative sentiment class, the precision value was 67%, the recall was 62%, and the f1-score was 65%. Furthermore, for the results of classification and visualization, information was obtained that the aspects or issues arising from the emergence of phrases in negative sentiments that often appeared were that users still had difficulty in using the Transjakarta application, especially the features in it, such as balance top-ups, ticket purchases, and the application still often experiencing errors, while for positive sentiments, users felt that the application was good, easy to use, and facilitates users in finding bus route schedules and arrival times.

Acknowledgement

I would like to thank my parents for always supporting me. I would also like to thank Mr. Yosep Nuryaman as the supervisor who has guided me in conducting this research.

References

[1] A. T. Aurahman, "IMPLEMENTASI NAIVE BAYES UNTUK ANALISIS SENTIMEN TWEET PELAYANAN TRANSPORTASI PT TRANSJAKARTA," 2024.

[2] T. Nur *et al.*, "ANALISIS KEPUASAN KONSUMEN TERHADAP KUALITAS PELAYANAN TRANSJAKARTA," *Kinerja*, vol. 7, no. 02, pp. 341–349, Jan. 2024, doi: 10.34005/KINERJA.V7I02.4372.

[3] H. Junianto, P. Arsi, B. A. Kusuma, D. Intan, and S. Saputra, "Evaluasi Aplikasi Raileo Melalui Analisis Sentimen Ulasan Playstore Dengan Metode Naive Bayes," *SINTECH (Science and Information Technology) Journal*, vol. 7, no. 1, pp. 27–40, Apr. 2024, doi: 10.31598/SINTECHJOURNAL.V7I1.1505.

[4] C. A. Iglesias and A. Moreno, *Sentiment Analysis for Social Media*. 2020. doi: https://doi.org/10.3390/app9235037.

[5] M. I. Santoso and A. R. Dzikrillah, "Analisis Sentimen Pengguna Terhadap Kinerja Sistem Transportasi Umum Jakarta Menggunakan Algoritma Naive Bayes," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 6, pp. 3032–3043, Jun. 2024, doi: 10.30865/KLIK.V4I6.1936.

[6] S. Nurwahyuni, "ANALISIS SENTIMEN APLIKASI TRANSPORTASI ONLINE KRL ACCESS MENGGUNAKAN METODE NAIVE BAYES," *Swabumi*, vol. 7, no. 1, pp. 31–36, May 2019, doi: 10.31294/SWABUMI.V7I1.5575.

[7] R. Winurputra, D. E. Ratnawati, and P. Korespondensi, "Peramalan Penjualan Produk Menggunakan Extreme Gradient Boosting (XGBoost) dan Kerangka Kerja CRISP-DM untuk Pengoptimalan Manajemen Persediaan (Studi Kasus: UB Mart)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 12, no. 2, pp. 417–428, Apr. 2025, doi: 10.25126/JTIIK.2025129451.

[8] S. Syafrizal, M. Afdal, and R. Novita, "Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 10–19, Dec. 2023, doi: 10.57152/malcom.v4i1.983.

[9] F. Ladayya, D. Siregar, W. E. Pranoto, and H. D. Muchtar, "Analisis Sentimen pada Program Transportasi Publik JakLingko dengan Metode Support Vector Machine," *Jurnal Statistika dan Aplikasinya*, vol. 6, no. 2, pp. 381–392, Dec. 2022, doi: 10.21009/JSA.06221.

[10] D. N. N. Husnina, D. E. Ratnawati, and B. Rahayudi, "Analisis Sentimen Pengguna Aplikasi RedBus berdasarkan Ulasan di Google Play Store menggunakan Metode Naive Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 2, pp. 737–743, Mar. 2023, Accessed: Apr. 26, 2025. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12297

[11] A. Komarudin and A. M. Hilda, "Analisis Sentimen Ulasan Aplikasi Identitas Kependudukan Digital Pada Play Store Menggunakan Metode Naïve Bayes," *Computer Science (CO-SCIENCE)*, vol. 4, no. 1, pp. 28–36, Jan. 2024, doi: 10.31294/COSCIENCE.V4I1.2955.

[12] M. S. Santoso, J. J. Suryadi, K. Marchellino, G. Z. Nabiilah, and Rojali, "A Comparative Analysis of Decision Tree and Support Vector Machine on Suicide Ideation Detection," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 518–523. doi: 10.1016/j.procs.2023.10.553.

[13] J. Elektronik, I. K. Udayana, Y. Anugrah, V. Gunawan, N. Agus, and S. Er, "Analisis Sentimen Ulasan Aplikasi Transportasi Online Menggunakan Multinomial Naïve Bayes dan Query Expansion Ranking", [Online]. Available: http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/

[14] W. Aljedaani *et al.*, "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry," vol. 255, p. 109780, 2022, doi: 10.1016/j.knosys.2022.109780.

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