

Sentiment Analysis of User Reviews for Health Technology Applications (Halodoc, Alodokter, and SatuSehat) Using the Multinomial Naive Bayes Algorithm

Adhi Bahtiar^{1*}, Eko Darmanto², Muhammad Arifin³

^{1*, 2, 3} Information Systems Study Program, Faculty of Engineering, Universitas Muria Kudus, Kudus Regency, Central Java Province, Indonesia

Email: 202253112@std.umk.ac.id^{1*}, ekodarmanto@umk.ac.id², arifin.m@umk.ac.id³

Article history:

Received September 30, 2026

Revised October 11, 2026

Accepted October 16, 2026

Abstract

The growth of digital health services in Indonesia has driven increased usage of health applications such as Halodoc, Alodokter, and SatuSehat. These applications generate thousands of user reviews on the Google Play Store annually. As these reviews consist of unstructured text, manual sentiment identification is time-consuming and labor-intensive. This study aims to perform automated sentiment analysis on user reviews for these three applications using the Multinomial Naive Bayes algorithm combined with Term Frequency-Inverse Document Frequency (TF-IDF) weighting. The research employs the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework, encompassing data collection via scraping; text preprocessing (case folding, cleaning, tokenizing, stopword removal, and stemming); initial labeling using the InSet lexicon; TF-IDF weighting with a combination of unigrams and bigrams; Multinomial Naive Bayes model training; and result presentation via a Streamlit-based interactive dashboard. The final dataset comprises 10,739 reviews, split into 80% training data and 20% testing data. Test results indicate that the model achieved an accuracy of 48.3%, precision of 91.6%, recall of 48.3%, and an F1-score of 59.8%. The best performance was observed in the positive class (precision of 0.98), whereas the neutral class showed poor performance due to data imbalance. Analysis results reveal that SatuSehat had the highest proportion of negative sentiment (70.1%), while Halodoc had the highest proportion of positive sentiment (88.6%). This study provides insights into user perceptions of the three applications and offers valuable information for developers to improve their services.

Keywords:

Sentiment Analysis; Halodoc; Alodokter; SatuSehat; Multinomial Naive Bayes; TF-IDF.

1. INTRODUCTION

Advancements in information technology have brought about changes across various sectors, including healthcare. The advent of digital technology has made it easier for the public to access healthcare services through various applications. Some widely used health apps in Indonesia include Halodoc, Alodokter, and SatuSehat. Users can submit reviews on the Google Play Store based on their experiences using these applications. These user reviews serve as a valuable source of information for developers, helping them gauge satisfaction levels and user experiences, as well as identify any ongoing issues or obstacles faced by users (Hendra, 2021).

As the number of reviews grows, the volume of data increases, typically taking the form of unstructured text. This data requires processing to yield information that is easier to comprehend. Manual processing and analysis would be time-consuming, particularly as the volume of reviews rises. One approach to addressing this challenge is sentiment analysis. Sentiment analysis is a subset of Natural Language Processing (NLP) used to identify and classify the sentiment within a text into categories such as positive, negative, and neutral

(Rahman et al., 2024). Research by (Hendra, 2021) analyzed reviews of the Halodoc application on the Google Play Store using the Naïve Bayes Classifier algorithm, achieving an accuracy of 81.68% and an AUC value of 0.756. Additionally, (Indrayuni et al., 2021) compared the Support Vector Machine and K-Nearest Neighbors algorithms to perform sentiment analysis on the Halodoc application.

Sentiment analysis of the SatuSehat application has also been conducted in several previous studies. (Nugraha et al., 2023) utilized 16,245 SatuSehat reviews and achieved 93% accuracy using the Naïve Bayes Classifier under the optimal data-splitting scenario. Another study was conducted by (Ridho et al., 2026), combining Support Vector Machines and Naïve Bayes while employing SMOTE optimization. Additionally, (Qumaini, 2024) analyzed the SatuSehat application using a machine learning approach combined with Particle Swarm Optimization. Regarding other health applications, (Putra et al., 2025) used Multinomial Naive Bayes to analyze Mobile JKN reviews, employing several preprocessing stages: cleansing, case folding, tokenizing, stopword removal, and stemming. Similarly, (Ramadhan et al., 2025) used Naive Bayes to analyze user reviews from the Google Play Store. The reviews underwent a pre-processing stage before being classified into sentiment categories, demonstrating that Naive Bayes can be applied to text-based review data. Sentiment analysis research has also been conducted on user reviews of the Flo app using the Naive Bayes algorithm employed this method to classify the sentiments found in user reviews of the Flo app. Kurniawati et al., (2026) and online doctor consultation services (Wardani & Erfina, 2021). Based on these studies, Google Play Store reviews can serve as a data source for understanding user feedback and sentiment trends regarding health applications. Furthermore, Multinomial Naive Bayes represents a viable option for classifying text-based review data.

However, most previous studies have focused on a single health application or service, leaving sentiment comparisons across multiple health apps relatively unexplored. Therefore, this study examines three health applications Halodoc, Alodokter, and SatuSehat using user reviews from the Google Play Store from 2025. These reviews are classified into three sentiment categories: positive, neutral, and negative. Based on this context, the research addresses the following questions: (1) how to perform preprocessing and TF-IDF weighing on review data for sentiment classification; (2) how to apply the Multinomial Naive Bayes algorithm to classify reviews into the three sentiment categories; and (3) how the algorithm performs based on accuracy, precision, recall, F1-score, and the confusion matrix.

The objectives of this study are: (1) to process public reviews from the Halodoc, Alodokter, and SatuSehat applications into a more structured dataset; (2) to perform text preprocessing and TF-IDF weighting to generate features suitable for the classification process; (3) to apply the Multinomial Naive Bayes algorithm to classify reviews based on sentiment; (4) to assess model performance using a confusion matrix; and (5) to present a comparison of sentiments across the three applications via an interactive dashboard that can serve as a reference for digital healthcare service developers. Beyond its practical utility, this study aims to contribute to the body of knowledge regarding the application of Multinomial Naive Bayes for Indonesian text classification. The article is organized into several sections: the research methodology, detailing the CRISP-DM stages and system architecture; the results and discussion, covering dataset characteristics, sentiment distribution, and model evaluation outcomes; and a conclusion summarizing the research findings and identified limitations.

2. RESEARCH METHOD

2.1. Research Type and Design

This study employs a quantitative method with a descriptive-analytical approach. The data analyzed consists of unstructured text derived from user reviews on the Google Play Store. The study focuses on three widely used health applications in Indonesia: Halodoc, Alodokter, and SatuSehat. The review data used is limited to the year 2025.

To ensure a structured research process, this study utilizes the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. This framework comprises six stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The research workflow based on the CRISP-DM stages is illustrated in Figure 1.

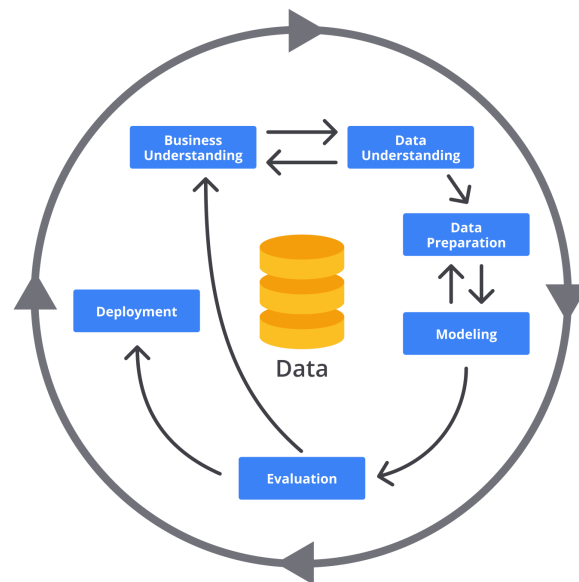


Figure 1. CRISP-DM Workflow

2.2. Business Understanding and Data Understanding

In the Business Understanding phase, the research is motivated by the large volume of user reviews available as unstructured text. The substantial amount of data renders manual analysis inefficient and time-consuming. Consequently, this research develops a system capable of automatically processing user reviews to determine their sentiment tendencies.

The system accepts input in the form of pre-cleaned user review text. It then generates sentiment predictions categorized as positive, negative, or neutral. These results provide valuable insights for application developers whether in the private or public sector to help them identify areas for service improvement, resolve system issues, and develop new features.

The Data Understanding phase is carried out using the `scraping.py` module. This module retrieves review data for the three applications from the Google Play Store with the aid of the `google-play-scraper` library. The acquired data comprising review text, star ratings, and review dates is saved in CSV format.

The primary attributes in the dataset are Review Text (the user's comment content, serving as the independent variable) and Rating Star (a star rating from 1 to 5, used as a reference for the labeling process). Once obtained, the dataset is uploaded to the dashboard via the Upload Dataset page. This page also displays preliminary information such as the total number of reviews, empty reviews, duplicate reviews, and the average rating for each application as shown in Figure 2.

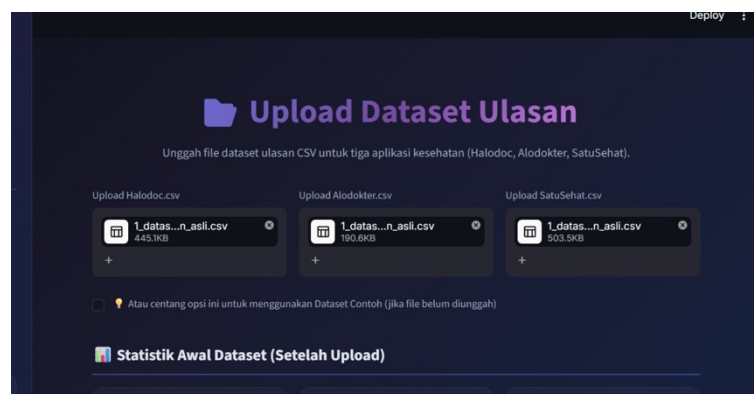


Figure 2. Review Dataset Upload Interface on the SentiHealth Dashboard

The results of the initial data collection are presented in Table 1. The obtained dataset comprises 5,019 reviews for Halodoc, 2,114 for Alodokter, and 4,053 for SatuSehat. The average ratings for the respective applications are 4.55 for Halodoc, 4.24 for Alodokter, and 2.24 for SatuSehat. These differences in average ratings indicate a tendency toward lower scores for SatuSehat compared to the other two applications.

Table 1. Initial Statistics of the Review Dataset per Application

Application	Number of Reviews	Empty Reviewss	Duplicate Reviews	Average Ratinng	Positive	Negative	Neutral
Halodoc	5.019	0	1.220	4,55	4.464	500	55
Alodokter	2.114	0	394	4,24	1.707	357	50
SatuSehat	4.053	0	304	2,24	1.104	2.765	184

2.3. Data Preparation

The data preparation stage involves transforming raw data into a dataset ready for the classification process using the Multinomial Naive Bayes algorithm. Initially, the data is inspected and cleaned to remove null values, text consisting solely of non-textual characters, and duplicates. Unnecessary data is removed to ensure the quality and validity of the dataset are maintained.

Similar preprocessing steps have been employed in previous sentiment analysis studies. (Hal et al., 2025) used case folding, tokenization, filtering, and stemming to prepare review data prior to the classification stage.

Once the data cleaning process is complete, the review text is processed using the preprocessing.py module. Preprocessing is carried out in five stages: case folding, cleaning, tokenizing, stopword removal, and stemming. Descriptions of each stage are provided in Table 2.

Table 2. Text Preprocessing Stages

Stage	Process Description
Case Folding	Change all letters in the review text to lowercase
Cleaning	Removing URLs, mentions (@user), hashtags (#tag), numbers, punctuation, emojis, and other special characters
Tokenizing	Splitting a review sentence into multiple words or tokens
Stopword Removal	Removing common words considered not to contribute significant information regarding sentiment, using the Sastrawi stopwords list and an additional dictionary created for app reviews
Stemming	Converting words with affixes into their root forms using the PySastrawi library for example, changing the word “pelayanan” to “layan”

Once the preprocessing stage yields cleaned text, the next step is to assign an initial sentiment label. This labeling is performed via the lexicon.py module using the InSet Lexicon (Indonesia Sentiment Lexicon). The lexicon contains positive and negative words; each associated with a specific sentiment weight or value.

The values of the words within a review are summed to determine its sentiment class. Reviews with a score greater than zero are categorized as positive, those with a score less than zero as negative, and those with a score of zero as neutral. If the calculated score is zero or if the review contains no words found in the lexicon the Google Play Store rating serves as the basis for labeling: ratings of 4 and 5 are categorized as positive, a rating of 3 as neutral, and ratings of 1 and 2 as negative.

The labeled data is then converted from text format into a numerical representation using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This process is carried out so that the data can be used as input for classification algorithms.

2.4. Modelling

The modeling stage is carried out using the model.py module. In this stage, the preprocessed text data is converted into a numerical format using TF-IDF and subsequently used to train a classification model. A TF-IDF Vectorizer is employed to generate a word-weighting matrix that considers combinations of unigrams and bigrams. Meanwhile, Label Encoding is used to convert sentiment labels into a numerical format.

The configurations used for model construction and training are shown in Figure 3 and summarized in Table 3.



Figure 3. TF-IDF Weighting Parameters and Multinomial Naïve Bayes Algorithm

Table 1. Model Training Configuration

Parameter	Value
N-gram range	(1,2) unigrams and bigrams
Max features	5,000 most frequent words
Min document frequency	2
Sublinear TF	True (logarithmic word frequency scaling)
Classification algorithm	Multinomial Naïve Bayes (MultinomialNB)
Alpha hyperparameter	1.0 (Laplace Smoothing)
Target encoding	LabelEncoder – Positive, Negative, Neutral -> 0,1,2
Data splitting scheme	Stratified split, 80% training data and 20% test data

Multinomial Naive Bayes is a classification algorithm that employs Bayes' Theorem to determine the probability of a data point belonging to a specific class based on the features present in that data. In this study, the features used consist of words from user reviews. The calculation of the posterior probability is shown in Equation (1).

$$\mathcal{P}(C_k | x) = \frac{\mathcal{P}(C_k) \times \mathcal{P}(x | C_k)}{\mathcal{P}(x)} \quad (1)$$

Explanation:

- $\mathcal{P}(C_k | x)$ is the posterior probability of sentiment class C_k given review document x
- $\mathcal{P}(C_k)$ is the prior probability of sentiment class C_k
- $\mathcal{P}(x | C_k)$ is the probability of the occurrence of word features in a document, given that the document belongs to class C_k
- $\mathcal{P}(x)$ is the predictor probability, which is constant

The data is split into 80% training data and 20% test data using the stratified split method. The training data is used to build the model based on word probability patterns within each class, while the test data is used to assess the model's predictive performance on data not included in the training process.

Upon completion of the training process, the model and the components required for prediction are saved in the 'model/' folder. The saved artifacts include 'multinomial_nb_model.pkl' (the Multinomial Naive Bayes model), 'tfidf_vectorizer.pkl' (the TF-IDF Vectorizer), 'label_encoder.pkl' (the Label Encoder), and 'metrics.json' (containing the evaluation metrics).

2.5. Evaluation

The evaluation stage is conducted to determine how well the Multinomial Naive Bayes model classifies review sentiment. Evaluation is performed using a confusion matrix, yielding True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) components. These components are subsequently used to calculate accuracy, precision, recall, and F1-score, as shown in Equations (2) through (5).

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

$$\text{Precision} = TP / (TP + FP) \quad (3)$$

$$\text{Recall} = TP / (TP + FN) \quad (4)$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (5)$$

Accuracy is used to assess the overall correctness of the model's predictions. Precision indicates the accuracy of the model's predictions for a specific class, whereas recall measures the model's ability to identify data that actually belongs to that class. Meanwhile, the F1-score is used to evaluate the balance between precision and recall.

In addition to evaluating overall performance, this study also examines prediction results for each individual application. This is done to determine whether the model yields consistent results when applied to review data from Halodoc, Alodokter, and SatuSehat.

2.6. Deployment

The final stage of CRISP-DM is Deployment. In this study, this stage is implemented through two primary modules: 'predict.py' and 'app.py'. The 'predict.py' module is used to load the saved model and vectorizer, preprocess new reviews, convert text into TF-IDF features, and generate sentiment class predictions along with confidence scores.

Meanwhile, the 'app.py' module is used to build an interactive dashboard based on Streamlit. This dashboard features several pages: Home, Upload Dataset, Preprocessing, Training, Dashboard, Prediction, and About. Each module serves a distinct function in supporting the workflow, ranging from data processing to sentiment prediction. Details regarding the functions of each module are presented in Table 4. The configurations used for model construction and training are shown in Figure 3 and summarized in Table 3.

Table 4. System Module Components and Their Functions

Module	Main Function
scraping.py	Retrieves review data, ratings, and dates from the Google Play Store for the Halodoc, Alodokter, and SatuSehat apps, then saves them in CSV format
preprocessing.py	Performs case folding, cleaning, tokenizing, stopword removal, and stemming on review text
lexicon.py	Assigns an initial sentiment label based on scores from the InSet Lexicon, comprising positive, negative, and neutral classes
model.py	Performs TF-IDF vectorization, label encoding, and Multinomial Naïve Bayes training, and saves the model and evaluation results
predict.py	Performs real-time prediction on new reviews using the saved model and vectorizer
app.py	Displays a Streamlit-based interactive dashboard for data processing, model evaluation, and manual sentiment prediction

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1. Characteristics of the Final Dataset

Following the data cleaning process—which involved removing null entries, duplicate reviews, and unprocessable data and the subsequent assignment of sentiment labels based on lexicons and ratings, a final dataset comprising 10,739 reviews was obtained. This dataset consists of 4,851 reviews for Halodoc, 2,003 for Alodokter, and 3,885 for SatuSehat.

The final dataset was then divided into training and testing sets using an 80:20 split ratio. Specifically, 8,591 reviews were allocated for training and 2,148 for testing. A stratified split method was employed to ensure the balance of sentiment classes was maintained across both the training and testing sets.

3.1.2. Sentiment Distribution Across Applications

The labeling results reveal differences in sentiment trends among the three applications studied. This comparison is illustrated in Figures 4 and 5. Halodoc exhibits the highest percentage of positive sentiment, at 88.6% (4,298 reviews). Negative sentiment for Halodoc stands at 10.3% (498 reviews), while neutral sentiment accounts for 1.13% (55 reviews).

For Alodokter, positive reviews account for 80.3% (1,609 reviews), whereas negative reviews make up 17.5% (351 reviews). In contrast to the other two applications, SatuSehat shows a higher tendency toward negative sentiment. Of the total data for SatuSehat, 70.1% (2,725 reviews) reflect negative sentiment, while positive sentiment accounts for only 25.3% (983 reviews).

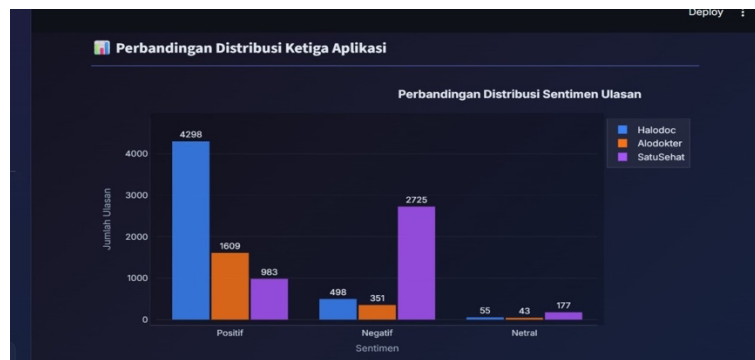


Figure 4. Comparison of Review Sentiment Distribution Across the Three Applications

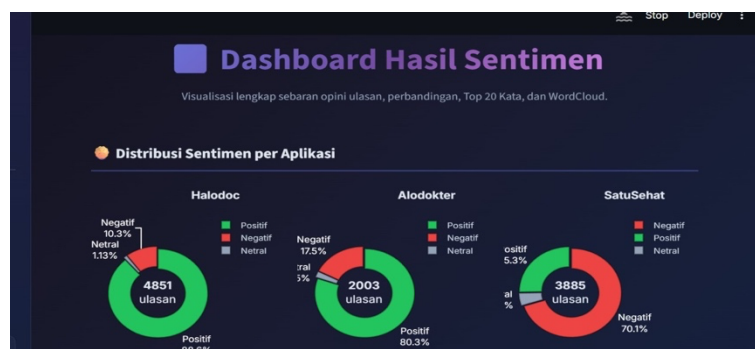


Figure 5. Dashboard: Percentage Distribution of Sentiment per Application

The high volume of negative sentiment regarding SatuSehat aligns with the average rating obtained during the initial phase specifically 2.24 out of 5. This indicates a tendency for users to rate the SatuSehat app lower than Halodoc and Alodokter. Such ratings may reflect suboptimal user experiences, whether stemming from technical issues or the services provided within the app.

These findings are consistent with the study by (Nugraha et al., 2023), which also identified a predominance of negative sentiment in reviews of the SatuSehat app. However, as that study utilized a different dataset size and timeframe, the results cannot be directly compared.

3.1.3. Dominant Keyword Analysis

Word frequency analysis and WordClouds were used to identify the most frequently occurring words within each sentiment category. Based on Figure 6, words commonly found in positive reviews include "help" (bantu), "doctor" (dokter), "easy" (mudah), "medicine" (obat), "fast" (cepat), "friendly" (ramah), and "great" (mantap). The appearance of these words indicates that some users responded positively to the ease of accessing services, particularly the consultation process with doctors.

Meanwhile, reviews with negative sentiment frequently featured words such as "register" (daftar), "login," "try" (coba), "pay" (bayar), "email," and "difficult" (sulit). These words point to complaints regarding the registration process, login or authentication, and payment. In neutral reviews, frequently occurring words included "doctor" (dokter), "vaccine" (vaksin), "register" (daftar), and "healthy" (sehat). The usage of these words tended to be informative or descriptive, reflecting neither strong positive nor strong negative sentiment.



Figure 6. Word Cloud by Sentiment Category (Positive, Negative, Neutral)

To identify the most frequently used words overall, an analysis was conducted on the top 20 words by frequency, as shown in Figure 7. The results indicate that the word "dokter" (doctor) appeared 2,001 times, making it the most frequent word, followed by "bantu" (help) at 1,994 occurrences. Other frequently appearing words included "bagus" (good), "obat" (medicine), "cepat" (fast), "daftar" (register), and "login".

The high frequency of the words "daftar" and "login" suggests that the registration and authentication processes are significant topics for users when writing reviews. This highlights an area for application developers to evaluate regarding the ease and stability of user access processes.

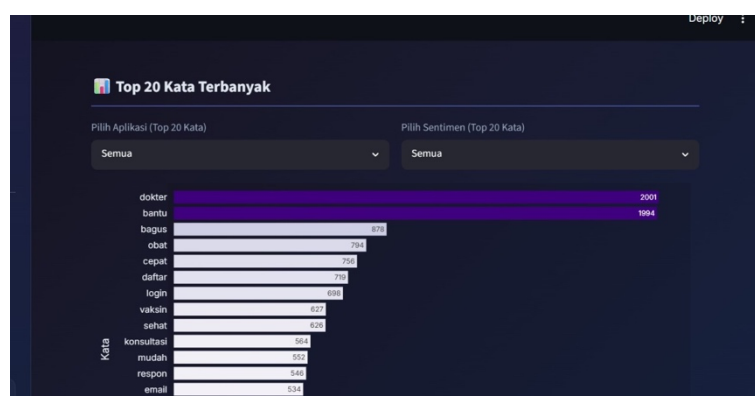


Figure 7. The 20 Words with the Highest Frequency of Occurrence in the Review Dataset

3.1.4. Classification Report Model Multinomial Naïve Bayes

The Multinomial Naive Bayes model was trained using combined data from the three applications and subsequently tested using 2,148 test samples, representing 20% of the entire dataset. Based on the test results, the model achieved an accuracy of 48.3%, precision of 91.6%, recall of 48.3%, and an F1-score of 59.8%. These results are presented in Figure 8.



Figure 8. Summary of Model Evaluation Metrics on Test Data

The confusion matrix and classification report results shown in Figure 9 and Table 5 indicate that the model's ability to recognize each sentiment class varies.



Figure 9. Confusion Matrix and Classification Report for the Multinomial Naïve Bayes Model

Table 5. Model Classification Report on Test Data (2,148 Reviews)

Class	Precision	Recall	F1-Score	Support
Negative	0.86	0.21	0.34	715
Neutral	0.04	0.80	0.08	55
Positive	0.98	0.61	0.75	1,378
Accuracy	-	-	0.48	2,148
Macro avg	0.63	0.54	0.39	2,148
Weighted avg	0.92	0.48	0.60	2,148

For the positive class, the model achieved a precision of 0.98, a recall of 0.61, and an F1-score of 0.75. These results indicate that the model performs well in making positive predictions when it classifies a review as positive; however, some positive reviews are still misclassified into other sentiment categories.

The negative class showed a precision of 0.86, yet the recall was only 0.21. This means that while the model is quite accurate when making negative predictions, a significant number of reviews that are actually negative fail to be identified as such. According to the confusion matrix, out of 715 negative reviews in the test data, 547 were predicted as neutral.

A more noticeable issue arises with the neutral class, which exhibits a precision of 0.04 and a recall of 0.80. These figures indicate that while the model successfully identifies the majority of neutral data, it also frequently predicts the neutral label for data actually belonging to other classes. Consequently, the precision for the neutral class is very low.

This situation stems from the imbalanced distribution of data across the classes; the volume of neutral data is significantly lower than that of the positive and negative data. This class imbalance means the model has far more examples to learn from regarding the language patterns of the positive and negative classes compared to the neutral class. This factor also contributes to the overall accuracy, which stands at only 48.3%.

The discrepancy between a high overall precision score and lower recall and accuracy scores indicates that the model is quite precise when predicting a specific class but still struggles to identify all the data points that should belong to that class. Consequently, data imbalance is a factor that must be addressed to improve the model's performance.

Data imbalance was also observed in the study by (Cikania, 2021), which analyzed sentiment in reviews of the Halodoc telemedicine service. In that study, the volume of positive data exceeded that of negative data;

consequently, model evaluation relied not only on accuracy but also on AUC and G-Mean metrics. One method to address data imbalance is oversampling specifically, the Synthetic Minority Over-sampling Technique (SMOTE). (Ridho et al., 2026) also employed this technique in their sentiment analysis of the SATUSEHAT Mobile application to balance the dataset prior to classification using SVM and Naïve Bayes.

3.1.5. Real-Time Sentiment Prediction Module

In addition to testing on datasets, the developed system also provides a real-time sentiment prediction feature. This feature is available on the Prediction page and allows users to input new reviews directly. Once the user enters text, the system displays the sentiment classification result along with a confidence score, as shown in Figure 10.

The prediction process utilizes the 'predict.py' module. This module re-runs the preprocessing steps on the input text and converts it into TF-IDF features before processing it using the Multinomial Naive Bayes model. The final output is the sentiment label predicted by the model.

This feature offers convenience, as users do not need to undergo a retraining process when testing a new review. Developers or application administrators can utilize this feature to quickly observe sentiment trends in specific reviews.

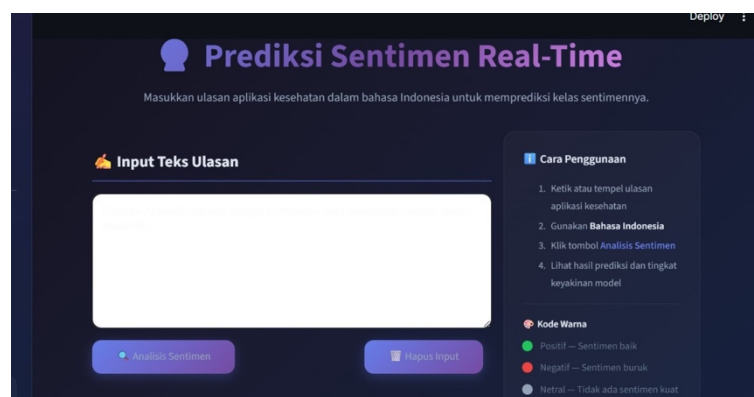


Figure 10. Real-Time Sentiment Prediction page on the SentiHealth Dashboard

3.2. Discussion

Based on the research results, a combination of TF-IDF and Multinomial Naive Bayes can be applied to build a sentiment analysis system for health app reviews. The developed system encompasses several stages, ranging from data acquisition via scraping, text cleaning and preprocessing, sentiment labeling, model training, and evaluation, to the presentation of results through an interactive dashboard.

Test results indicate that the model achieves relatively high precision at 91.6%, suggesting that its predictions for specific classes are quite accurate. However, accuracy and recall values of 48.3% indicate that the model still struggles to correctly identify the data as a whole. This performance is lower than that reported in previous studies, such as (Hendra, 2021), who achieved 81.68% accuracy in a binary (positive and negative) classification task, and (Nugraha et al., 2023), who achieved 93% accuracy under specific data-splitting scenarios.

These differing results may be attributed to several factors. This study employs three sentiment classes positive, negative, and neutral whereas the study by (Hendra, 2021) utilized only two. Furthermore, the data in this study originated from three different applications, each with distinct review characteristics and sentiment distributions. Data imbalance, particularly regarding the neutral class, also played a role in influencing the model's classification performance.

For future research, model performance could be improved by employing data balancing techniques such as SMOTE, increasing the volume of data for the neutral class, or developing separate models for each application. Other algorithms, such as Support Vector Machines, could also be used for comparison, as demonstrated by (Indrayuni et al., 2021) and (Ridho et al., 2026).

Despite the limitations in classification performance, the results regarding sentiment distribution and word analysis still provide insight into user reactions to the three health applications. A notable finding is the frequent occurrence of terms related to login and registration processes within negative reviews. Developers can use this information to evaluate the services and identify areas requiring attention.

4. CONCLUSION

This research has developed a sentiment analysis system for user reviews of the Halodoc, Alodokter, and SatuSehat applications, employing TF-IDF for feature weighting and the Multinomial Naive Bayes algorithm for classification. The research process followed the CRISP-DM framework, encompassing data collection,

text preprocessing, sentiment labeling using the InSet Lexicon, model development, evaluation, and deployment via a Streamlit-based dashboard.

The final dataset used in the study comprised 10,739 reviews. The analysis results indicate that Halodoc had the highest percentage of positive sentiment (88.6%), whereas SatuSehat had the highest percentage of negative sentiment (70.1%). These findings reveal differences in user opinion trends across the respective applications. The Multinomial Naive Bayes testing yielded an accuracy of 48.3%, precision of 91.6%, recall of 48.3%, and an F1-score of 59.8%. The model demonstrated better performance in identifying the positive class, whereas the neutral class remained the most challenging to handle. One contributing factor is the class imbalance in the dataset, particularly the significantly lower volume of neutral data.

In addition to performing sentiment classification, the system provides a dashboard for visualizing sentiment distribution, frequently occurring words, and model evaluation results, as well as a feature for real-time prediction on new reviews. These insights can serve as valuable information for health application developers to gauge user feedback and identify service areas requiring evaluation.

This study still has several limitations, particularly regarding the data imbalance across classes and suboptimal model performance. Therefore, future research could consider employing data balancing techniques such as SMOTE, augmenting data for minority classes, segmenting models based on applications, extending the data collection period, and comparing the results with other classification algorithms—such as Support Vector Machines or deep learning methods.

ACKNOWLEDGEMENTS

The author expresses gratitude to the Information Systems Study Program, Faculty of Engineering, Universitas Muria Kudus, for the support provided during the conduct of this research. Thanks, are also extended to the academic supervisor for providing guidance, suggestions, and input throughout the research process and the preparation of this article.

REFERENCES

- Cikania, R. N. (2021). Implementasi Algoritma Naive Bayes Classifier dan Support Vector Machine Pada Klasifikasi Sentimen Review Layanan Telemedicine Halodoc. *Jambura Journal of Probability and Statistics*, 2(2), 96-104. <https://doi.org/10.34312/jjps.v2i2.11364>
- Hal, D., Wibowo, R. S., Rakan, M., & Ramadhan, W. (2025). Analisis Sentimen Menggunakan Support Vector Machine dan Naive Bayes Pengguna Game Roblox. *JUKOMIKA (Jurnal Ilmu Komputer Dan Informatika)*, 8(2), 106–112. <https://doi.org/10.54650/jukomika.v8i2.618>
- Hendra, A. (2021). Analisis Sentimen Review Halodoc Menggunakan Naive Bayes Classifier. (2021). *JISKA (Jurnal Informatika Sunan Kalijaga)*, 6(2), 78-89. <https://doi.org/10.14421/jiska.2021.6.2.78-89>
- Indrayuni, E., Nurhadi, A., & Kristiyanti, D. A. (2021). Implementasi algoritma Naive Bayes, support vector machine, dan K-nearest neighbors untuk analisa sentimen aplikasi Halodoc. *Faktor Exacta*, 14(2), 64-71. <http://dx.doi.org/10.30998/faktorexacta.v14i2.9697>
- Kurniawati, E., Irma Purnamasari, A., Ali, I., Kurniawan, R., & Nurdian, O.. (2026). Analisis Sentimen Ulasan Pengguna Aplikasi Flo di Google Play Store Dengan Menggunakan Algoritma Naive Bayes. *Jurnal Informatika Dan Teknik Elektro Terapan*, 14(1). <https://doi.org/10.23960/jitet.v14i1.8776>
- Nugraha, M. T., Sulistiyowati, N., Enri, U., Karawang, U. S., Timur, T., Sehat, S., & Classifier, N. B. (2023). Analisis Sentimen Ulasan Aplikasi Satu Sehat Pada Google Play Store Menggunakan Naive Bayes Classifier. *JATI: Jurnal Mahasiswa Teknik Informatika*, 7(5), 3593–3601. <https://doi.org/10.36040/jati.v7i5.7753>
- Putra, I. E., Asriyanik, A., & Azzahra, F. F. (2025). Pemanfaatan Multinomial Naive Bayes Untuk Analisis Sentimen Ulasan Aplikasi Mobile Jkn. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(2), 3283-3291. <https://doi.org/10.36040/jati.v9i2.13311>
- Qumaini, R. (2024). Analisis Sentimen Aplikasi SatuSehat Menggunakan Metode Naive Bayes dan Particle Swarm Optimization. <http://repository.unsri.ac.id/id/eprint/158138>

- Rahman, J., Riaz, A., Malakar, P., & Kabir, M. (2024). Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. *Natural Language Processing Journal*, 6, 100059. <https://doi.org/10.1016/j.nlp.2024.100059>
- Ramadhan, W., Arifin, M., Fithri, D. L., & Setiaji, P. (2025). Analisis Sentimen Ulasan Pengguna Aplikasi Spotify Di Google Play Store Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(2), 3600-3607. <https://doi.org/10.36040/jati.v9i2.13543>
- Ridho, M. R., Revindo, M. K., & Saputra, D. M. (2026). Analisis Sentimen Pengguna Aplikasi SatuSehat Mobile Menggunakan SVM dan Naive Bayes Dengan Optimasi SMOTE. *ATI (Jurnal Mahasiswa Teknik Informatika)*, 10(2), 3480-3487. <https://doi.org/10.36040/jati.v10i2.17629>
- Wardani, N. R., & Erfina, A. (2021). Analisis Sentimen Masyarakat Terhadap Layanan Konsultasi Dokter Menggunakan Algoritma Naive Bayes. *SISMATIK: Seminar Nasional Sistem Informasi Dan Manajemen Informatika*, 11–18.