International Journal Software Engineering and Computer Science (IJSECS)
6 (2), 2026, 404-416
Published Online August 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i2.6956.

RESEARCH ARTICLE Open Access

Modeling the Reputation of Digital Banks Based on Public Opinion Using a Text Mining Approach with TF-IDF and the Support Vector Machine (SVM) Algorithm

Muhamad Ihsan Ashari [1*], **Afif Efendi** [2], **Dimas Eko Prasetyo** [3].

[1*,2,3] Department of Information Systems, Universitas Pamulang, South Tangerang City, Banten Province, Indonesia.

**Corresponding author*: dosen03154@unpam.ac.id.

Received: April 14, 2026; Accepted: May 9, 2026; Published: August 1, 2026.

Abstract: This study addressed the growing importance of reputation management in digital banking, where public opinion expressed on social media significantly influences customer trust and business sustainability. The objective of this research was to model the reputation of a digital bank based on public sentiment using a text mining approach. The study employed the CRISP-DM methodology, including data collection, preprocessing, modeling, and evaluation. A total of 1,897 Twitter comments related to the "Jenius" digital banking application were collected from 2023 to 2025. The data underwent preprocessing stages such as case folding, cleansing, tokenizing, normalization, stopword removal, negation handling, and stemming. Feature extraction was performed using Term Frequency–Inverse Document Frequency (TF-IDF), and sentiment classification was conducted using Support Vector Machine (SVM). The performance of SVM was compared with Naïve Bayes and K-Nearest Neighbors (KNN). The results showed that SVM achieved the best performance with an accuracy of 81.58%, outperforming Naïve Bayes (70.26%) and KNN (55.00%). Furthermore, sentiment distribution indicated that positive sentiment dominated public opinion, reflecting a generally favorable perception of the digital bank. In conclusion, the combination of TF-IDF and SVM proved effective for sentiment classification and can be utilized to model digital bank reputation, providing valuable insights for improving service quality and customer satisfaction.

Keywords: Text Mining; Digital Bank; SVM; TF-IDF.

1. Introduction

In today's modern era, technology continues to advance rapidly and shows no sign of slowing, with finance being one of the sectors most strongly affected by this development. Many financial products now incorporate technology—a trend commonly referred to as Financial Technology (FinTech)—and the digital bank stands as one prominent example of such innovation. Financial technology itself refers to companies that offer technological innovations within the financial sector (Astuti *et al.*, 2022). However, the rapid growth of digital banks has also given rise to various issues related to service quality and customer satisfaction. Poor service quality and customer dissatisfaction can lead to a decline in a company's reputation and, in turn, negatively affect its business growth. Banking companies therefore need to evaluate and measure customer satisfaction levels on an ongoing basis, including through sentiment analysis, which is a text analysis technique used to

extract positive, negative, or neutral sentiments from customer-generated text such as product or service reviews. In the context of digital banking, sentiment analysis can be applied to measure customer satisfaction with the services provided by a digital bank, particularly through discussions occurring on the social media platform X.

Among the various approaches available, the weighting method generally favored in previous studies is based on a vector space model, namely Term Frequency–Inverse Document Frequency (TF-IDF) (Fahrezi *et al.*, 2024). Although term frequency is widely used in term-weighting calculations, it only accounts for the proportion of documents that can be retrieved through the search process in an information retrieval system. Meanwhile, the proportion of retrieved documents considered relevant to user needs tends to increase when the weighting vector incorporates terms that rarely appear across the broader document collection (Singgalen, 2023). By combining these two components—term frequency and inverse document frequency, both of which account for how often a word occurs—this study aims to simultaneously increase the proportion of documents that can be retrieved and considered relevant. In this sense, the most appropriate terms are those that appear frequently within individual documents but rarely appear elsewhere in the collection.

This study specifically employs the TF-IDF method, a statistical approach used to quantify the importance of a keyword relative to a specific document. TF-IDF combines two concepts: Term Frequency (TF), which measures how often a word appears within a single document, and Inverse Document Frequency (IDF), which measures the relative importance of that word across an entire document collection. The most appropriate criterion for a word to represent a document well is that it appears frequently within that particular document (high TF) while remaining rare across other documents (high IDF). This principle is supported by findings from a previous study titled "Comparison of Support Vector Machine (SVM) with Naive Bayes for Sentiment Analysis on the BRImo Application," conducted by Astuti *et al.* (2022), which showed that SVM achieved an accuracy of 97.69%, outperforming Naive Bayes at 96.53%. These results indicate that SVM is a superior method for categorizing review data from the BRImo mobile banking application, reinforcing its suitability for similar sentiment classification tasks. Building on this foundation, the present study employs SVM to address non-linear or uneven data classification, with the combination of TF-IDF and SVM expected to yield higher accuracy in sentiment analysis compared to the use of SVM alone.

2. Related Work

Several previous studies have examined text-mining-based sentiment analysis using various machine learning and deep learning algorithms across different domains. In the tourism sector, Ipmawati *et al.* (2024) analyzed sentiment in tourist attraction reviews using a Support Vector Machine (SVM) algorithm combined with TF-IDF weighting. Their study showed that the combination of TF-IDF and SVM was capable of producing strong classification performance with high accuracy, making it effective in identifying user opinions regarding a service. A related study in the e-commerce domain examined Shopee application reviews on the Google Play Store, applying a combination of TF-IDF and Long Short-Term Memory (LSTM) to classify sentiment into three categories: positive, neutral, and negative (Musfiroh *et al.*, 2024). The model achieved an accuracy of 83%, performing reasonably well on the positive and negative classes but less optimally on the neutral class due to data imbalance. This finding suggests that although deep learning models such as LSTM are capable of capturing long-term context, data quality and distribution remain critical factors in determining model performance (Idris *et al.*, 2023).

A simpler yet effective approach was also demonstrated in a sentiment analysis study on game reviews using the Naïve Bayes algorithm with TF-IDF weighting (Wati *et al.*, 2023), which achieved an accuracy of 94.12%. This result indicates that classical algorithms such as Naïve Bayes remain competitive in text classification, particularly when supported by effective preprocessing. The preprocessing steps applied in that study—filtering, case folding, tokenization, stopword removal, and stemming—were shown to improve data quality prior to the classification process (Rahmadani *et al.*, 2024). Beyond sentiment analysis, the application of SVM algorithms extends to other fields, such as online transaction fraud detection (Eldo *et al.*, 2024), where SVM achieved up to 95% accuracy in classifying transactions as legitimate or fraudulent. The strength of SVM in this context lies in its ability to handle high-dimensional data and identify complex patterns through an optimal hyperplane, which further reinforces the rationale for selecting SVM as a reliable algorithm across various classification scenarios, including text-based sentiment analysis.

Taken together, these studies confirm that TF-IDF as a feature extraction technique, combined with classification algorithms such as SVM, Naïve Bayes, and LSTM, has proven effective for sentiment analysis across diverse domains, including tourism, e-commerce, and digital applications. However, most of these studies remain focused on general sentiment classification and have not specifically addressed reputation modeling, particularly within the digital banking sector. This study therefore proposes an approach that extends beyond sentiment classification by utilizing its results to model the reputation of digital banks based

on public opinion. Through the combination of TF-IDF and SVM, this study is expected to contribute to the broader application of text mining within the digital financial industry and offer more strategic insights into public perceptions of digital banking services.

3. Methodology

This study employs the CRISP-DM methodology, which consists of six stages—business understanding, data understanding, data preparation, modeling, evaluation, and deployment—all of which were applied in this research (Singgalen, 2023b). This study adopts a quantitative approach, using a specific population and sample for data collection. Sentiment analysis on digital banking application users was conducted using the Support Vector Machine (SVM) approach for sentiment classification. Data collection was carried out using dedicated research tools, and quantitative data analysis was performed to test the predetermined hypotheses. Tweet data from the Twitter (X) platform containing the keyword "Jenius app" served as the primary data source for this study.

3.1 Business Understanding

At this stage, an understanding of the research object is established by gathering information related to the existing problems. The increasing competition among digital banking products requires companies to move quickly in order to remain competitive and become market leaders. Business Understanding is also conducted to identify the most suitable sentiment analysis approach, which helps determine the appropriate model based on a comparison of algorithm performance results. The machine learning algorithm applied in this study is Support Vector Machine (SVM).

3.2 Data Understanding

Data Understanding is the process of comprehending the data used as the research subject. At this stage, raw data is collected based on the required attributes. This phase helps identify potential issues or opportunities within the data and determines whether the available data can support the objectives of the analysis. The crawling process was carried out using Python and Node.js to access and download data from Twitter, covering the period from January 1, 2023, to December 31, 2025, using the keyword "Jenius application." This process yielded 7,883 comment records. Filtering was then performed to remove duplicate records and comments originating from the official Jenius account, eliminating 5,986 records. As a result, the final dataset consisted of 1,897 records, which were subsequently processed through preprocessing. The initial dataset contained 12 columns; however, not all columns were used in the analysis, which focused only on user comments regarding the Jenius application, the comment date, and the comment ID. A feature selection process was therefore conducted on the dataset, with the selected attributes shown in Figure 1.

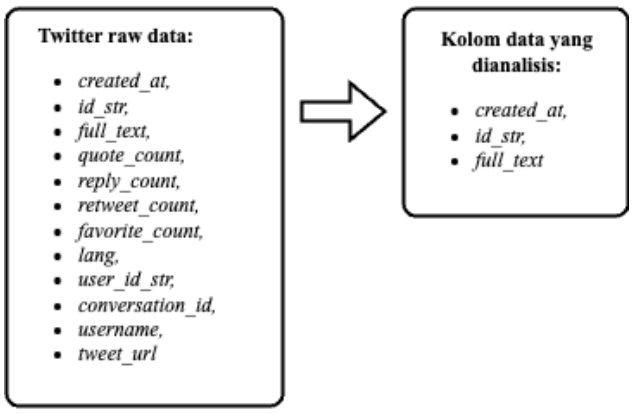


Figure 1. Feature Selection

Feature selection is the process of selecting relevant attributes prior to data analysis. This process is necessary because the available data is large and continuously growing in volume. Feature selection is performed to improve accuracy and enhance performance, particularly when dealing with high-dimensional data (Kartiwi *et al.*, 2019), while also simplifying the data structure and reducing dataset size to make processing more efficient (Hermawan *et al.*, 2023). Attribute selection was applied because not all collected attributes were relevant to the analysis, and reducing the overall dataset size helped accelerate both the data cleaning process and the subsequent text analysis of tweets. The dataset after filtering and feature selection is presented in Table 1.

Table 1. Dataset Before Preprocessing

No.	created_at	id_str	full_text
1	Sat Jan 30 18:42:49 +0000	1355587286713286657	Mudah atur keuangan/kehidupan dengan Jenius. Download aplikasi Jenius di https://t.co/onNfLrYy0b dan masukkan $Cashtag: $dyckha di kolom referral.
2	Sat Jan 30 14:15:06 +0000	1355519915113889794	*menunggu selama 10 menit supaya aplikasi @JeniusConnect bisa ke-load* jenius: https://t.co/Trby3Mh0aE
...	...	...	...
1.897	Fri Des 30 09:31:21 +0000	1355086120191582213	@390Piyo Selamat sore, Bapak/Ibu. Untuk pertanyaan atau pengaduan terkait aplikasi Jenius, Bapak/Ibu dapat menghubungi Jenius Help di 1500365, email: jenius-help@btpn.com, Twitter: @jeniusconnect &amp @jeniushelp, atau chat langsung melalui aplikasi. Terima kasih.

3.3 Data Preparation

The data preparation stage aims to produce clean, ready-to-use data for research purposes by transforming raw data into high-quality data. Preprocessing is necessary because raw data may still be incomplete (e.g., missing attribute values or aggregated placeholders such as full_text = ".."), contain noise (errors and outliers), or be inconsistent in terms of coding and naming standards. The collected dataset was processed through several preprocessing stages to optimize the development of both training and testing classification models, given that user comments do not fully adhere to standard language conventions. All preprocessing stages were implemented using Python libraries, as described below.

3.3.1 Case Folding

Case folding converts all letters in the text into lowercase to ensure consistency in letter usage. This step supports feature engineering techniques such as term frequency and tokenization by preventing duplication of identical words with different capitalization, thereby maintaining a consistent text representation that avoids unintended shifts in meaning. A comparison of text before and after case folding is presented in Table 2.

Table 2. Comparison of the Case Folding Process

Before Case Folding	After Case Folding
Balik ke hp lama setelah kemarin hp tiba-tiba mati total, aplikasi-aplikasi penting ya tinggal install lagi beres. Cuma satu yang bermasalah, Jenius, ga bisa buat login.	balik ke hp lama setelah kemarin hp tiba-tiba mati total, aplikasi-aplikasi penting ya tinggal install lagi beres. cuma satu yang bermasalah, jenius, ga bisa buat login.
@jeniushelp halo mau tanya untuk UNSUBSCRIBE langganan dari aplikasi apakah bisa dibantu dr Jenius ya? Karena gak terdaftar di Google Play	@jeniushelp halo mau tanya untuk unsubscribe langganan dari aplikasi apakah bisa dibantu dr jenius ya? karena gak terdaftar di google play

3.3.2 Cleansing and Tokenizing

Cleansing removes non-essential text elements such as mentions (@), hashtags (#), links, numerals, non-alphabetic characters, punctuation, empty characters, emojis/emoticons, and repeated characters, making the training process more efficient. Dirty data can reduce analysis accuracy, slow down processing, and complicate the interpretation of results. A comparison of text before and after cleansing is presented in Table 3.

Table 3. Comparison of the Cleansing Process

Before Cleansing	After Cleansing
Udah cobain merchant GO-FOOD terbaru deket rumah kamu? Yuk order sekarang juga! Isi saldo GO-PAYnya cukup di aplikasi Jenius by BTPN! @gojekindonesia #MudahPakaiPRIMA	udah cobain merchant gofood terbaru deket rumah kamu yuk order sekarang juga isi saldo gopaynya cukup di aplikasi jenius by btpn mudahpakaiprima

@jeniushelp halo min, saya dr kemarin mau login ulang ke apk jenius ngga bisa, selalu dibilangnya your session has expired". Sudah coba reinstall aplikasi, tp tetap aja gini. https://t.co/9bAu0Dh46o"	halo min saya dr kemarin mau login ulang ke apk jenius ngga bisa selalu dibilangnya your session has expired sudah coba reinstall aplikasi tp tetap aja gini cobaudho

Tokenization converts text into smaller units called "tokens," which may be words, phrases, or characters depending on the required level of detail, allowing text to be broken down into its smallest analyzable units. A comparison of text before and after tokenization is presented in Table 4.

Table 4. Comparison of the Tokenizing Process

Before Tokenizing	After Tokenizing
Udah cobain merchant GO-FOOD terbaru deket rumah kamu? Yuk order sekarang juga! Isi saldo GO-PAYnya cukup di aplikasi Jenius by BTPN! @gojekindonesia #MudahPakaiPRIMA	['udah', 'cobain', 'merchant', 'gofood', 'terbaru', 'deket', 'rumah', 'kamu', 'yuk', 'order', 'sekarang', 'juga', 'isi', 'saldo', 'gopaynya', 'cukup', 'di', 'aplikasi', 'jenius', 'by', 'btpn', 'mudahpakaiprima']
@jeniushelp halo min, saya dr kemarin mau login ulang ke apk jenius ngga bisa, selalu dibilangnya your session has expired". Sudah coba reinstall aplikasi, tp tetap aja gini. https://t.co/9bAu0Dh46o"	['halo', 'min', 'saya', 'dr', 'kemarin', 'mau', 'login', 'ulang', 'ke', 'apk', 'jenius', 'ngga', 'bisa', 'selalu', 'dibilangnya', 'your', 'session', 'has', 'expired', 'sudah', 'coba', 'reinstall', 'aplikasi', 'tp', 'tetap', 'aja', 'gini', 'cobaudho']

3.3.3 Normalization

Normalization converts informal words into formal words according to KBBI (Indonesian Dictionary) standards, addressing the prevalence of foreign terms or informal language in crawled user comments. This step standardizes word forms and improves algorithm performance by reducing bias, accelerating convergence, and increasing model stability. A comparison of text before and after normalization is presented in Table 5.

Table 5. Comparison of the Normalization Process

Before Normalization	After Normalization
['sudah', 'coba', 'merchant', 'gofood', 'terbaru', 'dekat', 'rumah', 'kamu', 'yuk', 'order', 'sekarang', 'juga', 'isi', 'saldo', 'gopaynya', 'cukup', 'di', 'aplikasi', 'jenius', 'by', 'btpn', 'mudahpakaiprima']	['halo', 'min', 'saya', 'dari', 'kemarin', 'mau', 'login', 'ulang', 'ke', 'apakah', 'jenius', 'tidak', 'bisa', 'selalu', 'dibilangnya', 'your', 'session', 'has', 'expired', 'sudah', 'coba', 'reinstall', 'aplikasi', 'tetapi', 'tetap', 'saja', 'begini', 'cobaudho']

3.3.4 Negation Handling

Negation handling addresses negative words that alter sentence polarity. For example, "cannot download the Jenius application" carries a negative meaning; however, standard stopword removal typically eliminates words such as "not," which would unintentionally reverse the sentence's meaning to "can download the Jenius application." If left unaddressed, negative reviews risk being misclassified as positive. Negation handling combines the word following a negation term with an underscore ("_") separator, while temporarily retaining the negation word itself. This ensures that once the negation word is removed during stopword removal, only the modified token remains. A comparison of text before and after negation handling is presented in Table 6.

Table 6. Comparison of the Negation Handling Process

Before Negation Handling	After Negation Handling
gagal dapet promo XXI pake debit Jenius karena kartu debitnya kelamaan nggak dipake jadi kudu reset pin dulu via aplikasi	['gagal', 'dapat', 'promo', 'xxi', 'pakai', 'debit', 'jenius', 'karena', 'kartu', 'debitnya', 'kelamaan', 'enggak', 'enggak_enggak_enggak_dipakai', 'jadi', 'harus', 'reset', 'pin', 'dulu', 'via', 'aplikasi']

@jeniushelp Assalamualaikum Siang Ka. Kenapa saya ga bisa masuk aplikasi Jenius? Apakah sedang eror? Terima kasih 🙏🙏🙏	['assalamualaikum', 'siang', 'kakak', 'kenapa', 'saya', 'tidak', 'tidak_tidak_tidak_bisa', 'masuk', 'aplikasi', 'jenius', 'apakah', 'sedang', 'eror', 'terima', 'kasih']

3.3.5 Stopword Removal

Stopword removal eliminates common words that carry little semantic weight, based on a predefined stopword list. This stage was implemented using the Natural Language Toolkit (NLTK) library in Python. A comparison of text before and after stopword removal is presented in Table 7.

Table 7. Comparison of the Stopword Removal Process

Before Stopword	After Stopword
Udah cobain merchant GO-FOOD terbaru deket rumah kamu? Yuk order sekarang juga! Isi saldo GO-PAYnya cukup di aplikasi Jenius by BTPN! @gojekindonesia #MudahPakaiPRIMA	['coba', 'merchant', 'gofood', 'terbaru', 'rumah', 'yuk', 'order', 'isi', 'saldo', 'gopaynya', 'aplikasi', 'jenius', 'by', 'btpn', 'mudahpakaiprima']
@jeniushelp halo min, saya dr kemarin mau login ulang ke apk jenius ngga bisa, selalu dibilangnya your session has expired". Sudah coba reinstall aplikasi, tp tetap aja gini. https://t.co/9bAu0Dh46o"	['halo', 'min', 'kemarin', 'login', 'ulang', 'jenius', 'tidak_tidak_bisa', 'dibilangnya', 'your', 'session', 'has', 'expired', 'coba', 'reinstall', 'aplikasi', 'cobaudho']

3.3.6 Stemming

Stemming reduces words to their base or root form by removing affixes while preserving the root word, allowing words sharing the same root to be treated as a single entity. For the Indonesian language, common prefixes include *me-*, *mem-*, *meng-*, and *di-*, while common suffixes include *-i*, *-nya*, and *-an*. A comparison of text before and after stemming is presented in Table 8.

Table 8. Comparison of the Stemming Process

Before Stemming	After Stemming
Udah cobain merchant GO-FOOD terbaru deket rumah kamu? Yuk order sekarang juga! Isi saldo GO-PAYnya cukup di aplikasi Jenius by BTPN! @gojekindonesia #MudahPakaiPRIMA	['coba', 'merchant', 'gofood', 'rumah', 'yuk', 'order', 'isi', 'saldo', 'gopaynya', 'aplikasi', 'jenius', 'by', 'btpn', 'mudahpakaiprima']
@jeniushelp halo min, saya dr kemarin mau login ulang ke apk jenius ngga bisa, selalu dibilangnya your session has expired". Sudah coba reinstall aplikasi, tp tetap aja gini. https://t.co/9bAu0Dh46o"	['halo', 'min', 'kemarin', 'login', 'ulang', 'jenius', 'bilang', 'your', 'session', 'has', 'expired', 'coba', 'reinstall', 'aplikasi', 'cobaudho']

3.3.7 Sentiment Labeling Based on a Lexicon Dictionary

Lexicon-based sentiment labeling assigns sentiment values (positive, negative, or neutral) to words using a predefined dictionary, with the overall sentiment of a text calculated by aggregating these values. This process, also known as sentiment annotation, produces the labels "Positive," "Negative," and "Neutral." For example, in the sentence "Yesterday my day went well, but the next day went badly," the lexicon labels "well" as positive and "badly" as negative, while other words remain neutral. The overall sentiment score is calculated using the following formula:

$$Score\ Sentiment = Positive\ Word\ Count - Negative\ Word\ Count \quad (1)$$

A neutral sentiment is assigned when the resulting score equals zero. The dataset labeled through this lexicon-based approach is presented in Table 9.

Table 9. Labeling Results Based on the Lexicon Dictionary

created_at	full_text	token	positif	negatif	sentiment_score	sentiment
Sat Jan 30 14:15:06 +0000	tunggu menit aplikasi keload jenius cotrbymhae	['tunggu', 'menit', 'aplikasi', 'keload', 'jenius', 'cotrbymhae']	1	0	1	Positif
Sat Jan 30 11:30:50 +0000	admin restart device force stop aplikasi reinstalluninstall tidak bisa kendala jenius handphone browser laptop tidak bisa akses	['admin', 'restart', 'device', 'force', 'stop', 'aplikasi', 'reinstalluninstall', 'tidak bisa', 'kendala', 'jenius', 'handphone', 'browser', 'laptop', 'tidak bisa', 'akses']	1	3	-2	Negatif
Sat Jan 30 09:30:46 +0000	lupa password akunaplikasi jenius pilih lupa password jenius kirim notifikasi nomor handphone tidak saya kenal tolong bantu	['lupa', 'password', 'akunaplikasi', 'jenius', 'pilih', 'lupa', 'password', 'jenius', 'kirim', 'notifikasi', 'nomor', 'handphone', 'tidak saya', 'kenal', 'tolong', 'bantu']	2	2	0	Netral
...	...	...	...	...	...	...
Fri Dec 01 04:17:49 +0000	jenius hadir krisflyer miles traveloka points tukar yay points real time aplikasi coycqbrqew	['jenius', 'hadir', 'krisflyer', 'miles', 'traveloka', 'points', 'tukar', 'yay', 'points', 'real', 'time', 'aplikasi', 'coycqbrqew']	2	0	2	Positif

3.3.8 Dataset Splitting

At this stage, the dataset was divided into training and testing subsets to determine which data would be used for classification. An 80:20 split was applied, allocating 80% of the data for training and 20% for testing. Training data was used to build the model, while testing data was used for evaluation. The results of the splitting process are presented in Table 10.

Table 10. Data Splitting Results

Label	Amount of Data
X_train	1.518
X_test	380
y_train	1.518
y_test	380

3.4 Data Collection Summary

This research methodology is grounded in Twitter data analysis. The dataset was collected from the Twitter platform using tweets containing the keyword "Jenius application" as the search query. Data collected from January 2023 to December 2025 formed the dataset used for sentiment analysis and algorithm testing.

3.5 Research Steps

This study follows the six iterative stages of the CRISP-DM framework as its research procedure. Figure 2 illustrates the overall research process flow based on this framework.

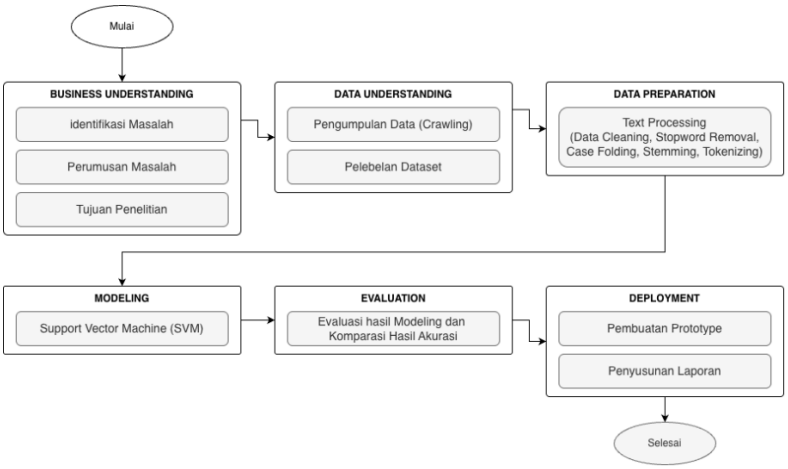


Figure 2. Research Steps (CRISP-DM Framework).

4. Result and Discussion

4.1 Results

4.1.1 Modeling

Modeling is the phase in which data mining techniques are selected by determining the algorithms to be used. In this phase, data classification was performed by comparing three algorithms: Naïve Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). These three algorithms were selected because they are widely used classification methods in data analysis, each employing a different approach to understanding data and generating predictions, yet sharing the same objective of classifying data into the correct class based on existing features. Comparing these algorithms allows the strengths, weaknesses, and characteristics of each to be identified, while also determining which algorithm is most suitable for classifying negative, positive, and neutral sentiment. This comparison further supports the evaluation of relative model performance and the selection of the most appropriate model for this particular use case.

4.1.2 Modeling Using the Naïve Bayes Algorithm

The experimental results using 80% training data and 20% testing data, classified using the Naïve Bayes model, produced an accuracy of 70.26%.

Table 11. Confusion Matrix for Naïve Bayes Classification

Actual Data	Predicted: Negatif	Predicted: Netral	Predicted: Positif	Precision	Recall	F1-score
Negatif	56	5	1	0.4480	0.9032	0.5989
Netral	30	66	4	0.6286	0.6600	0.6439
Positif	39	34	145	0.9667	0.6651	0.7880

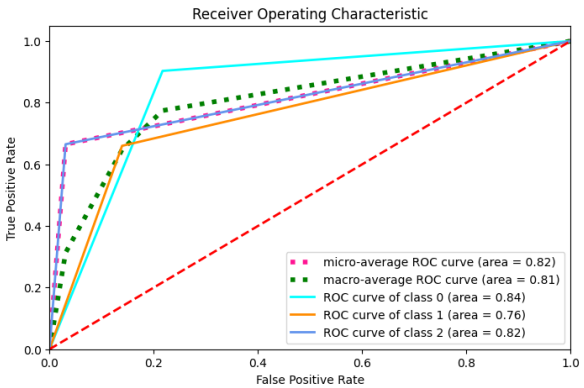


Figure 3. ROC Curve for Naïve Bayes

Figure 3 shows that the ROC value for the Naïve Bayes classification is 0.82, indicating that the algorithm performs well and falls into the "Good" category.

4.1.3 Modeling Using the K-Nearest Neighbors (KNN) Algorithm

The experimental results using 80% training data and 20% testing data, with a K value of 5, classified using the KNN model, produced an accuracy of 55.00%.

Table 12. Confusion Matrix for K-Nearest Neighbors Classification

Actual Data	Predicted: Negatif	Predicted: Netral	Predicted: Positif	Precision	Recall	F1-score
Negatif	51	10	1	0.3248	0.8226	0.4658
Netral	23	74	3	0.5481	0.7400	0.6298
Positif	83	51	84	0.9545	0.3853	0.5490

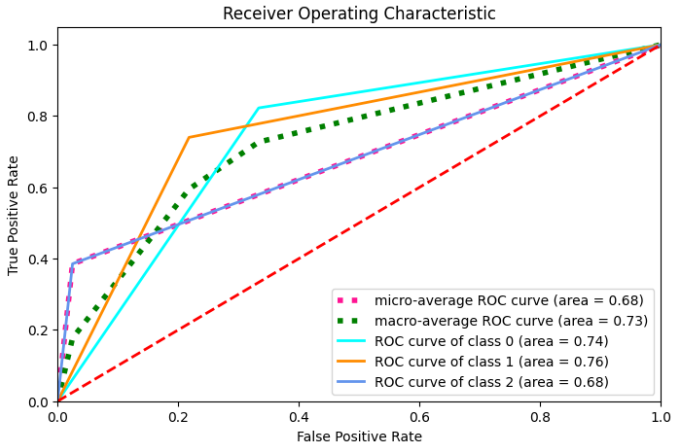


Figure 4. ROC Curve for K-Nearest Neighbors (KNN)

Figure 4 shows that the ROC value for the KNN classification is 0.68, indicating that the algorithm performs poorly and falls into the "Poor" category.

4.1.4 Modeling Using Support Vector Machine (SVM)

The experimental results using 80% training data and 20% testing data, classified using the SVM model, produced an accuracy of 81.58%.

Table 13. Confusion Matrix for Support Vector Machine (SVM) Classification

Actual Data	Predicted: Negatif	Predicted: Netral	Predicted: Positif	Precision	Recall	F1-score
Negatif	52	4	6	0.5843	0.8387	0.6887
Netral	29	57	14	0.8143	0.5700	0.6706
Positif	8	9	201	0.9095	0.9220	0.9157

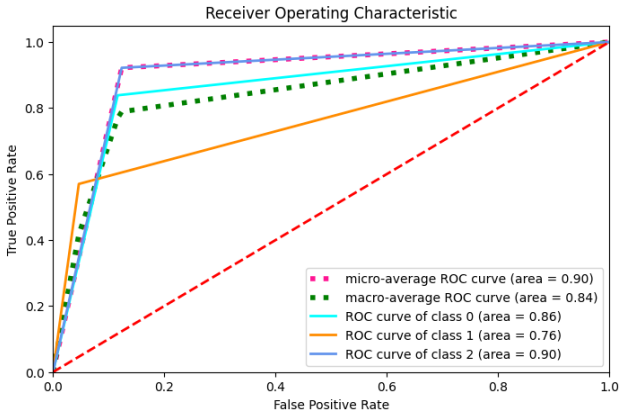


Figure 5. ROC Curve for Support Vector Machine (SVM)

Figure 5 shows that the ROC value for the SVM classification is 0.90, indicating that the algorithm performs very well and falls into the "Excellent" category.

4.1.5 Comparison of Classification Results

After conducting several evaluations, the comparative results of the three algorithms—Naïve Bayes, KNN, and SVM—are summarized in Table 14.

Table 14. Comparison of Classification Results Across Algorithms

Model	Accuracy	Precision	Recall	F1-score
Naïve Bayes	70.26%	68.11%	74.28%	67.70%
KNN	55.00%	60.92%	64.93%	54.82%
SVM	81.58%	76.94%	77.69%	75.83%

Table 14 shows that the SVM algorithm achieves the highest accuracy compared to Naïve Bayes and KNN. In general, SVM is more effective in handling high-dimensional datasets, which commonly occur in text analysis where feature spaces can become very large when using representations such as TF-IDF or word embeddings. SVM is capable of performing non-linear separation and utilizes kernels—such as the RBF kernel—to distinguish between sentiment classes, a capability that Naïve Bayes and KNN may not handle as effectively without additional preprocessing.

4.2 Discussion

The comparison of accuracy, precision, and recall between the standard SVM algorithm and the SVM algorithm enhanced with BERT text embedding is presented in Table 15.

Table 15. Comparison of Accuracy, Precision, and Recall Results

Model	Accuracy	Precision	Recall	F1-score
SVM	81.58%	76.94%	77.69%	75.83%

Text classification models facilitate the identification of sentiment categories such as positive, negative, and neutral. Based on the dataset processed using Python, review data were split into individual tokens, each assigned a specific weight. These weighted tokens were then used to identify frequently occurring sentiment-related terms, providing insight into public opinion regarding user responses to the Jenius application. The results of this study indicate that the SVM algorithm, when optimized using BERT, achieved an accuracy of 85.00%, with a precision of 81.10%, a recall of 82.25%, and an F1-score of 81.65%. These findings suggest that the BERT-optimized SVM classifier performs better in analyzing the Jenius application dataset compared to standard SVM.

4.2.2 Deployment

The deployment stage involves implementing the developed solution into a prototype, integrating the evaluated model into a web application. The analyzed data consisted of raw data obtained from Twitter, including the following features: created_at, full_text, quote_count, retweet_count, favorite_count, lang, user_id_str, username, tweet_url, id, geo, conversation_id, media_url_https, and media_type.

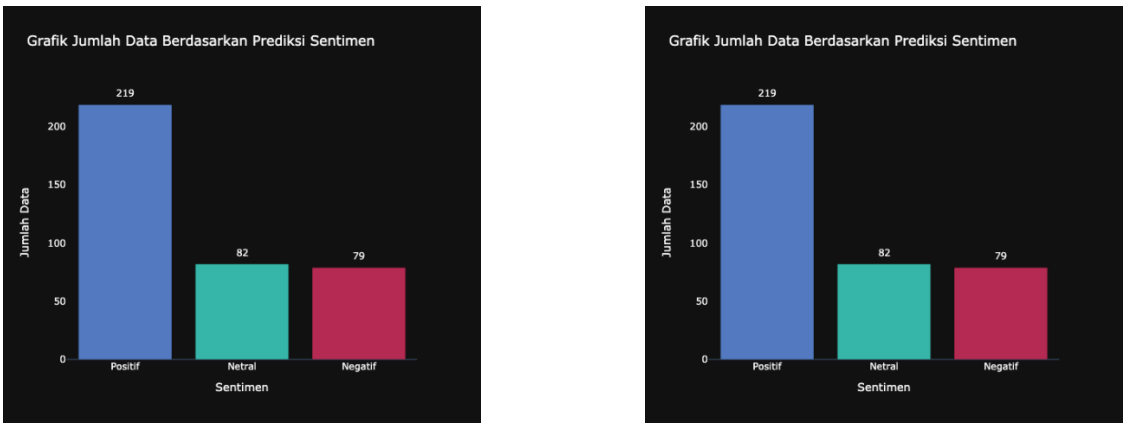
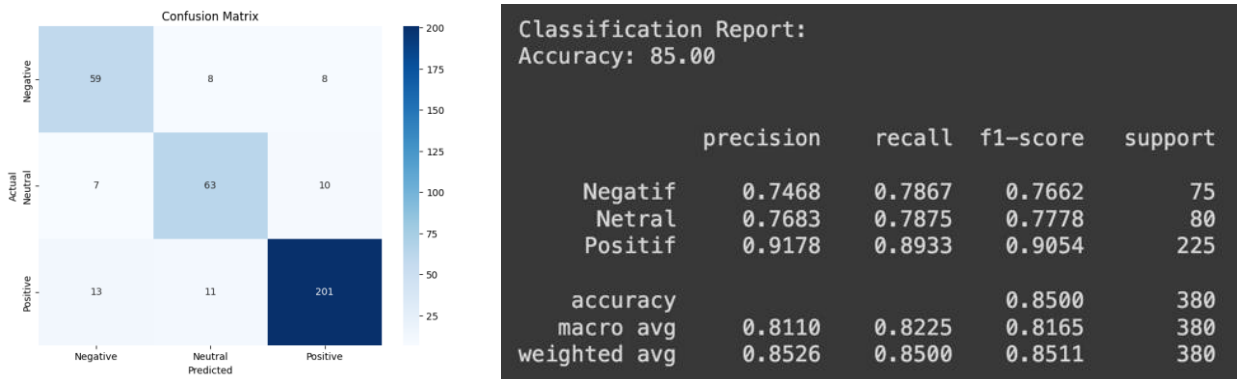


Figure 6. Data Visualization Page

The visualization page displays graphs and plots representing the sentiment distribution within the dataset, such as pie charts or bar charts illustrating the overall proportion of positive, negative, and neutral sentiments. The visualization shows that positive sentiment accounts for 219 records (57.6%), negative sentiment for 79 records (20.8%), and neutral sentiment for 82 records (21.6%).

Figure 7. Data Preprocessing Display

Figure 7 presents the results of the text preprocessing pipeline applied to clean and prepare the data for further analysis, including the removal of special characters, case conversion, stopword removal, stemming to handle word variations, and tokenization to split text into smaller analyzable units.



Classification Report:
Accuracy: 85.00

	precision	recall	f1-score	support
Negatif	0.7468	0.7867	0.7662	75
Netral	0.7683	0.7875	0.7778	80
Positif	0.9178	0.8933	0.9054	225
accuracy			0.8500	380
macro avg	0.8110	0.8225	0.8165	380
weighted avg	0.8526	0.8500	0.8511	380

Figure 8. Model Evaluation Display

Model evaluation using the confusion matrix and classification report for SVM provides a detailed understanding of the text classification model's performance. The confusion matrix shows the number of correct and incorrect predictions for each sentiment class (positive, negative, and neutral), enabling identification of the model's accuracy as well as evaluation of false positives and false negatives. The classification report summarizes the model's performance statistics, including a precision of 81.10%, a recall of 82.25%, and an F1-score of 81.65% across classes. Precision measures the accuracy of positive predictions, recall measures the model's ability to capture all actual positive instances, and the F1-score balances the two. Together, the confusion matrix and classification report support more informed decision-making in optimizing the BERT-SVM model for sentiment classification tasks.

5. Conclusion and Recommendations

Based on the results of this study, it can be concluded that the combination of TF-IDF feature extraction and the Support Vector Machine (SVM) algorithm is effective in classifying public sentiment toward the Jenius digital banking application into three categories: positive, negative, and neutral. On the testing dataset, the model correctly identified 219 positive sentiments, 79 negative sentiments, and 82 neutral sentiments, achieving an accuracy of 81.58%, a precision of 76.94%, a recall of 77.69%, and an F1-score of 75.83%. These results confirm that SVM outperformed both Naïve Bayes and K-Nearest Neighbors in this classification

task, reinforcing its suitability for sentiment analysis involving high-dimensional text data. The dominance of positive sentiment within the analyzed data further suggests a generally favorable public perception of the Jenius application, which can serve as a valuable indicator for evaluating and maintaining the reputation of digital banking services. Despite these promising results, this study has several limitations that should be acknowledged. First, the dataset was limited to tweets collected from a single social media platform (Twitter/X), which may not fully capture public opinion expressed through other channels such as Instagram, app store reviews, or customer service complaints. Second, the relatively lower F1-score for the neutral sentiment class compared to the positive class indicates that the model still faces challenges in distinguishing ambiguous or mixed expressions of sentiment.

Based on these findings and limitations, several recommendations are proposed for future research and practical application. For future studies, it is recommended to explore the use of word embedding techniques or transformer-based models to improve classification performance, particularly for the neutral sentiment category. Expanding the data sources to include multiple platforms would also provide a more comprehensive representation of public opinion toward digital banking services. From a practical standpoint, digital banking providers such as Jenius are encouraged to regularly monitor social media sentiment as part of their reputation management strategy, using the identified negative sentiment patterns as a basis for improving service quality, addressing recurring customer complaints, and enhancing overall customer satisfaction.

References

Astuti, A. P., Alam, S., & Jaelani, I. (2022). Komparasi algoritma Support Vector Machine dengan Naive Bayes untuk analisis sentimen pada aplikasi BRImo. *Jurnal Bangkit Indonesia*, *11*(2), 1–6. https://doi.org/10.52771/bangkitindonesia.v11i2.196

Eldo, H., Ayuliana, A., Suryadi, D., Chrisnawati, G., & Judijanto, L. (2024). Penggunaan algoritma Support Vector Machine (SVM) untuk deteksi penipuan pada transaksi online. *Jurnal Minfo Polgan*, *13*(2), 1627–1632. https://doi.org/10.33395/jmp.v13i2.14186

Fahrezi, I. A., Rudiman, & Verdikha, N. A. (2024). Analisis sentimen Twitter atas isu hak angket menggunakan pembobotan TF-IDF dan algoritma SVM. *Sci-Tech Journal*, *3*(2), 179–192. https://doi.org/10.56709/stj.v3i2.526

Hermawan, A., Jowensen, I., Junaedi, J., & Edy. (2023). Implementasi text-mining untuk analisis sentimen pada Twitter dengan algoritma Support Vector Machine. *JST (Jurnal Sains dan Teknologi)*, *12*(1), 129–137. https://doi.org/10.23887/jstundiksha.v12i1.52358

Idris, I. S. K., Mustofa, Y. A., & Salihi, I. A. (2023). Analisis sentimen terhadap penggunaan aplikasi Shopee menggunakan algoritma Support Vector Machine (SVM). *Jambura Journal of Electrical and Electronics Engineering*, *5*(1), 32–35.

Ipmawati, J., Saifulloh, S., & Kusnawi, K. (2024). Analisis sentimen tempat wisata berdasarkan ulasan pada Google Maps menggunakan algoritma Support Vector Machine. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(1), 247–256.

Kartiwi, M., Gunawan, T. S., Arundina, T., & Omar, M. A. (2019). Feature selection for financial data classification: Islamic finance application. In *2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)* (pp. 1–4). IEEE. https://doi.org/10.1109/ICSIMA.2018.8688803

Musfiroh, M., Tholib, A., & Arifin, Z. (2024). Analisis sentimen terhadap ulasan aplikasi Shopee di Google Play Store menggunakan metode TF-IDF dan Long Short-Term Memory. *Journal of Electrical Engineering and Computer (JEECOM)*, *6*(2), 371–381. https://doi.org/10.33650/jeecom.v6i2.8713

Rahmadani, R., Rahim, A., & Rudiman, R. (2024). Analisis sentimen ulasan "Ojol the Game" di Google Play Store menggunakan algoritma Naive Bayes dan model ekstraksi fitur TF-IDF untuk meningkatkan kualitas game. *Jurnal Informatika dan Teknik Elektro Terapan*, *12*(3). https://doi.org/10.23960/jitet.v12i3.4988

Singgalen, Y. A. (2023a). Analisis sentimen dan sistem pendukung keputusan menginap di hotel menggunakan metode CRISP-DM dan SAW. *Journal of Information System Research (JOSH)*, *4*(4), 1343–1353. https://doi.org/10.47065/josh.v4i4.3917

Singgalen, Y. A. (2023b). Analisis perilaku wisatawan berdasarkan data ulasan di website Tripadvisor menggunakan CRISP-DM: Wisata minat khusus pendakian Gunung Rinjani dan Gunung Bromo. *Journal of Computer System and Informatics (JoSYC)*, *4*(2), 326–338. https://doi.org/10.47065/josyc.v4i2.3042

Wati, R., Ernawati, S., & Rachmi, H. (2023). Pembobotan TF-IDF menggunakan Naïve Bayes pada sentimen masyarakat mengenai isu kenaikan BIPIH. *Jurnal Manajemen Informatika (JAMIKA)*, *13*(1), 84–93. https://doi.org/10.34010/jamika.v13i1.9424.