

Evaluating SVM and Naïve Bayes Classifiers with Resampling Techniques for Sentiment Analysis of Indonesian PayLater Users on Social Media



Aditya Fathan Santoso¹, Erick Fernando^{2*}

Faculty of Engineering and Informatics, Information System Program, Universitas Multimedia Nusantara, Banten 15180, Indonesia

Corresponding Author Email: erick.fernando@umn.ac.id

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310402>

ABSTRACT

Received: 13 August 2025

Revised: 1 November 2025

Accepted: 20 April 2026

Available online: 30 April 2026

Keywords:

PayLater services, sentiment analysis, social media mining, Support Vector Machine, Naïve Bayes classifier, resampling techniques, semi-supervised learning, class imbalance handling

Advances in financial technology have popularized PayLater services in Indonesia, enabling online consumers to pay in installments. While offering convenience, these services also pose challenges such as impulsive behavior and default risks. This study analyzes user sentiment toward PayLater by comparing the performance of Support Vector Machine (SVM) and Naïve Bayes Classifier (NBC) algorithms with resampling techniques. Data were collected from social media, preprocessed, and partially labeled using semi-supervised learning. Tweets were classified as positive (support or satisfaction) or negative (criticism or complaint). Class imbalance was addressed using Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN), and hyperparameters were optimized via GridSearch. Model performance was evaluated with 10-fold cross-validation, considering accuracy, precision, recall, and F1-score. Results indicate that SVM combined with SMOTE and GridSearch achieved the highest accuracy (94.31%) and balanced performance across all metrics. Comparisons show that semi-supervised labeling improves model generalization compared to manually labeled data. Visual analyses, including word clouds and sentiment distribution, provide insights into dominant user concerns and satisfaction patterns. This research demonstrates the effectiveness of combining SVM, Naïve Bayes, and resampling strategies for sentiment classification in imbalanced social media datasets. Findings can guide fintech companies, regulators, and researchers in monitoring public sentiment, improving user engagement, and informing policy development. The study also illustrates a practical deployment workflow for integrating predictive sentiment analysis with web-based applications.

1. INTRODUCTION

The rapid advancement of digital technology has brought significant changes to various aspects of life, including commerce. Innovations in communication and information technology have made economic activities easier and more efficient. One major impact is the shift in buying and selling transactions from conventional methods to online systems. Consumers can now purchase goods or services more conveniently through marketplaces and e-commerce platforms, which offer various supporting features, such as search filters, shipping estimates, and product reviews to aid decision-making. Along with the rapid growth of digital transactions, online payment methods have also undergone significant development. Advances in financial technology (fintech) have provided people with more flexible payment options across various marketplaces. Currently, users can choose from a variety of payment methods, from bank transfers, digital wallets (e-wallets), and PayLater [1]. Since 2018, PayLater technology has been introduced as a new innovation in e-commerce and marketplace payment systems, gaining popularity. This method allows users to make transactions in advance and pay later, providing convenience

and flexibility in digital shopping [2].

The rapid growth of PayLater services in Indonesia cannot be separated from the strategic collaborations established between fintech companies and e-commerce platforms, which have played a significant role in expanding digital payment accessibility and enhancing consumer purchasing convenience. The first e-commerce platform to introduce PayLater in Indonesia was Traveloka, partnering with fintech company PT Dana Pasar Pinjaman. As it developed, more and more e-commerce companies adopted this technology, making it increasingly well-known to the public. Public enthusiasm for PayLater is evident in the continued growth in the number of its users from year to year [3]. Based on data from the Financial Services Authority (OJK), the number of PayLater financing contracts in Indonesia reached 79.92 million in 2023, a drastic increase compared to only 4.63 million contracts in 2019. The average annual growth rate reached 144.35%. As of March 2024, outstanding PayLater financing receivables were recorded at IDR 6.13 trillion, an annual increase of 23.9% year-on-year, reflecting the high public interest in this payment system. According to Agusman, Chief Executive of the Financial Institutions Supervision Agency of the OJK, PayLater's performance is projected to continue to

improve in line with technological developments that increasingly facilitate online shopping transactions [4].

The significant increase in the use of PayLater services in Indonesia is driven by several positive factors. Ease of access and payment flexibility make this service increasingly popular, especially among people who want to meet their needs without having to wait for funds [5]. Furthermore, the relatively quick and easy registration process, coupled with various attractive promotions, also increases PayLater's appeal among consumers. The low credit card penetration in Indonesia also makes PayLater a more practical alternative for those who want to access credit facilities without having a credit card [6]. However, despite its benefits, the increasing use of PayLater also has negative impacts that require attention. The ease of conducting transactions without immediate payment can encourage consumer behavior and impulsive buying, potentially leading to uncontrolled debt accumulation. If not managed properly, users can become trapped in financial burdens due to late payments, which incur additional fees or penalties. Furthermore, uncontrolled PayLater use can increase the risk of bad debt, which can ultimately impact the stability of the broader financial system [2].

The rapid growth of PayLater services in Indonesia has not only impacted people's consumption patterns but has also sparked diverse opinions regarding its benefits and risks. This is because PayLater directly influences individuals' financial habits, particularly in terms of the ease of unsecured debt. On the one hand, many see it as a practical solution to meet needs without having to have a credit card. However, on the other hand, this convenience also raises concerns about increasing consumer behavior and the risk of default [2]. These differences in experience and perspectives are what make PayLater a hot topic of discussion, especially on social media X (formerly Twitter) [7]. Figure 1, the choice of platform x as a data source is because the X social media application is a popular social media in Indonesia with a significant number of users, reaching 79.32 million in April 2023 [8]. As a social media that allows its users to express opinions directly, X becomes a space for the public to share experiences, discuss, or even express complaints regarding PayLater services [9]. Using the x platform allows users to share experiences directly, either in the form of reviews, discussions, or complaints. With the widespread reach of social media, opinions about PayLater can spread quickly and influence public perception, creating a growing debate within the community [9].

Given the large volume of opinions about PayLater on social media, effective methods are needed to understand public sentiment at large [10]. One widely used method is sentiment analysis, which functions to identify and categorize opinions or emotions within a text. Sentiment analysis is a branch of natural language processing (NLP) that aims to identify and categorize opinions or emotions within a text [10]. This concept began to develop in the early 2000s, driven by the increasing volume of digital data and the need for companies to analyze customer opinions through online reviews. In the early stages, sentiment analysis was conducted using a lexicon-based approach, where words with positive or negative connotations were manually classified. However, with advances in artificial intelligence technology, approaches based on machine learning and deep learning have become increasingly used to improve the accuracy of sentiment analysis [11]. Sentiment analysis enables fintech companies,

e-commerce companies, and financial regulators to understand public response to PayLater through data from social media platforms. This method maps public opinion patterns to identify trust trends and user satisfaction levels, thus assisting the industry and regulators in developing more appropriate policies [9].

Previous research has shown that various algorithms can be used in sentiment analysis, such as Naïve Bayes (NB) and Support Vector Machine (SVM). In sentiment analysis of PayLater services, the SVM algorithm achieved an accuracy of 89.74% on Shopee PayLater [10], while NB showed an accuracy of 87% in sentiment analysis of Indodana: PayLater & Pinjaman application users [12]. In this study, both algorithms were chosen because of the specific advantages they offer in the context of text-based sentiment analysis. SVM was prioritized for its ability to handle high-dimensional data and produce consistent text classification accuracy. This algorithm works by optimizing the hyperplane with the largest margin, so that the model remains stable even though the number of features is very large, such as in text representation using Term Frequency-Inverse Document Frequency (TF-IDF). Various studies have also proven that SVM is able to maintain reliable performance in various text data scenarios. The use of the NB algorithm is considered due to its simplicity in calculating probabilities and its efficiency in handling large datasets quickly. This algorithm is effective in classifying text based on word distribution and still provides accurate results even though the amount of training data is limited. Another advantage lies in its ability to handle sparse and high-dimensional data, which is often encountered in text-based sentiment analysis. The combination of SVM's stability on complex data and Naïve Bayes' flexibility in limited data conditions makes these two algorithms relevant for further testing in this research.

The study aims to investigate the Semi-Supervised Learning Method in the context of Indonesian public opinion about the PayLater service through sentiment analysis of comments on the social media platform X (Twitter). The main objective is to evaluate how public opinion influences perceptions of PayLater. Moreover, this research compares NB and SVM to analyze sentiment. Predicted results will help to increase the accuracy of sentiment analysis methods and should be beneficial to the fintech businesses, regulators, and researchers who want to understand public sentiment.

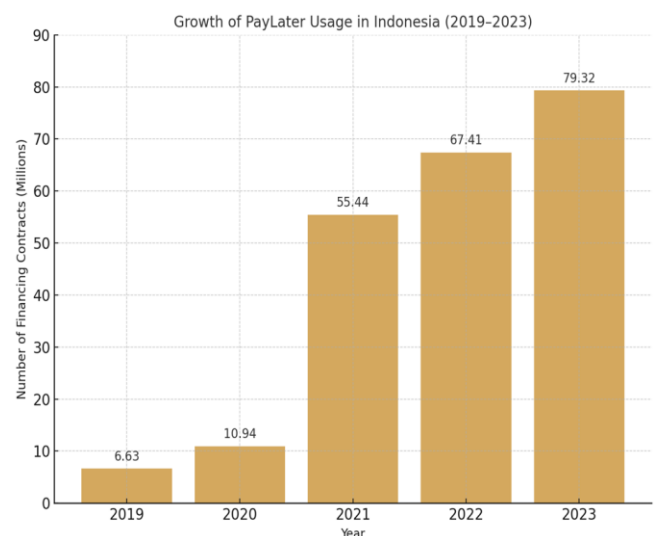


Figure 1. Growth in PayLater usage in Indonesia

2. RELATED WORK

Several previous studies have made important contributions to analyzing sentiment toward PayLater services, both in terms of classification methods and addressing challenges such as data imbalance and labeling.

One study applied the NB algorithm to classify sentiment from PayLater opinions derived from Twitter social media data collected through web scraping techniques, achieving 91% accuracy [13]. NB operates on probability principles based on Bayes' Theorem, assuming that features (words) are independent. This technique is particularly effective for short texts and relatively clean datasets. The study's findings revealed that negative sentiment dominated public conversations, particularly regarding high interest rates and late fees. NB also proved superior to lexicon-based approaches such as TextBlob, which only achieved 61% accuracy on a balanced dataset.

This labeling approach was also evaluated in a study comparing rating-based labeling methods with lexicon methods using the Indonesian Sentiment Lexicon (InSet) on Indodana app reviews [12]. By utilizing TF-IDF for feature extraction and NB as a classification model, the results show that lexicon-based labeling produces higher precision and recall (86%) than rating-based labeling, although accuracy is slightly lower (87%). TF-IDF plays a crucial role in balancing frequently occurring words with distinctive, informative words, improving the quality of feature representation.

The problem of class imbalance in review data is addressed by integrating SVM, Synthetic Minority Oversampling Technique (SMOTE), and a tuning process using GridSearch [10]. SVM works by constructing an optimal hyperplane that maximally separates two classes in a high-dimensional space, suitable for sparse text data. SMOTE adds synthetic data for the minority class by interpolating nearby points, thus improving the representation of the sparse class. GridSearch is used to select the best combination of parameters, such as C-value and kernel, to optimize model performance. This combination successfully increased accuracy to 90.27%, particularly in classifying negative sentiment.

Another study confirmed the effectiveness of SMOTE in addressing imbalance, with an SVM accuracy increase of up to 93% [14]. Furthermore, the Adaptive Synthetic Sampling (ADASYN) method was also used to adjust the number of synthetic samples based on data complexity, and demonstrated an 8.8% accuracy increase over models using word embeddings [15]. Word embeddings, such as Word2Vec or FastText, map words into fixed-dimensional vectors that represent semantic meaning, enabling better modeling of word context than frequency-based methods. However, for short review contexts such as those on PayLater services, TF-IDF is still considered more reliable because it focuses on explicit keywords.

A study developing an SVM model with a Radial Basis Function (RBF) kernel and parameter optimization through GridSearch achieved 97.5% accuracy on the Twitter dataset [16]. The RBF kernel enables a non-linear mapping of the original features to a higher-dimensional space, allowing the model to separate non-linearly separable data. Although not specifically targeting PayLater data, this approach serves as an important reference for applying hyperparameter optimization techniques and selecting appropriate kernels in social media sentiment analysis.

An SVM model with a linear kernel was also applied in the

context of P2P lending and achieved an accuracy of 82.33% [17]. Linear kernels are quite effective for text data because they naturally have a large number of features and are often already linearly separable in the feature space. This demonstrates that, despite limited data, SVMs remain relevant in the financial domain.

A hybrid integration of NB and a lexicon approach was applied to analyze 4,059 comments related to online loans [18]. This combination aimed to leverage the classification speed of NB and the sensitivity of dictionary-based approaches in capturing explicit sentiment. Although accuracy reached 82.06%, limitations lie in its reliance on static lexicons, which are less adaptable to the dynamics of social media language.

In an analysis of digital banking services, NB again demonstrated high performance with an accuracy of 88.12% [19]. The labeling process was performed manually, and the Twitter data was processed through text cleaning, tokenization, and stopword removal. The predominance of positive sentiment (54%) reflects user trust in digital banking innovations. In this context, NB demonstrates its consistency as a robust baseline model in the financial domain, especially when the class distribution is relatively balanced.

An exploration of semi-supervised learning (SSL) methods was conducted by combining the Random Forest and NB algorithms through pseudo-labeling techniques [20]. Pseudo-labeling works by assigning predictive labels to unlabeled data, which are then used for model retraining. Despite using two-class data, this approach achieved an F1-score of 0.76. SSL has great potential in leveraging the vast amounts of unlabeled data often available on digital financial platforms. The use of SSL also opens the possibility of further integration with SVM to improve accuracy in scenarios with limited and unlabeled data.

3. METHODOLOGY

The implementation stages in this study refer to the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. The selection of the CRISP-DM framework in this study is based on its systematic, flexible structure. The stages in CRISP-DM consist of six main steps [21], as shown in Figure 2.

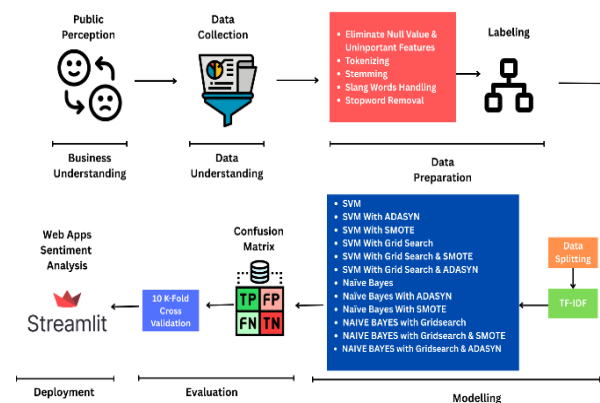


Figure 2. Research workflow

3.1 Business understanding

In the business understanding stage, the researcher gathered

literature on sentiment analysis of the PayLater service, data collection methods on social media platform X, and preprocessing techniques such as data labeling and handling dataset imbalance. The researcher reviewed the literature related to automatic labeling and how to handle class imbalance. Next, the researcher compared two machine learning methods, the Naïve Bayes Classifier (NBC) and the SVM, to select the most appropriate one for the dataset and research objectives. The primary objective of this research is to analyze user sentiment toward the PayLater service to understand public perception and provide insights for more informed decision-making.

3.2 Data understanding

The data retrieval method from platform X (Twitter) presented will use the Python programming language with the tweet-harvest library, executed through a terminal command or a Python environment that supports shell execution. This tool allows for specific data scraping based on keywords and specific criteria, as shown in Table 1. The search timeframe was also strictly defined, from January 1, 2022, to February 5, 2025, to ensure the data captured covers the PayLater policy implementation period. The dataset can be seen at the following open source link <https://data.mendeley.com/preview/3hwmz87fpc>.

Table 1. Detail query keywords

Query Section	Function	Description
"PayLater"	Mencari tweet yang mengandung kata "PayLater".	Kata kunci utama
-gestun	Menghapus tweet yang mengandung kata "gestun".	Filter negatif
-cair	Menghapus tweet yang mengandung kata "cair".	Filter negatif
-gesek	Menghapus tweet yang mengandung kata "gesek".	Filter negatif
-tunai	Menghapus tweet yang mengandung kata "tunai".	Filter negatif
Since: 2022-01-01	Mencari Tweet Mulai 1 Januari 2022.	Rentang Waktu Awal
Until: 2025-02-06	Mencari Tweet Hingga 5 Februari 2025.	Rentang Waktu Akhir

3.3 Data preparation

Text preprocessing is a crucial stage in converting natural language into a format that can be recognized and processed by machines. This stage involves a series of methods aimed at cleaning, normalizing, and simplifying text data to improve the quality of the analysis. Common methods used in text preprocessing include Eliminating Null Values and Unimportant Features, Data Cleaning, Tokenizing, Stemming, Slang Word Removal, and Labeling [10].

3.4 Modeling

The modeling stage is carried out after the preprocessing and feature representation process in the text data. The cleaned and structured data is used to train a sentiment classification model to recognize public opinion regarding the PayLater service. The primary goal is to identify sentiment patterns in

user comments and then classify them into positive or negative categories based on the learned data distribution. The modeling process is carried out using two algorithms: SVM and Naive Bayes, with two main approaches: without parameter tuning and with parameter tuning to achieve optimal performance. Model evaluation also involved three approaches to imbalanced data: unbalanced, with oversampling using SMOTE, and with ADASYN [10]. The formulas used in the modeling process are as follows:

3.4.1 Term Frequency-Inverse Document Frequency

The TF-IDF method is used to convert text into numeric vectors. TF can be calculated using Eq. (1).

$$tf(t, d) = \frac{n_{ij}}{\sum_k n_{ij}} \quad (1)$$

The $tf(t,d)$ function quantifies the frequency of term t in document d , computed as the ratio of the term's occurrences to the overall word count in the document. In this context, n_{ij} represents the frequency of the i -th term in the j -th document, whereas $\sum_k n_{ij}$ denotes the aggregate number of words within that document. Inverse Document Frequency (IDF) is employed to ascertain the frequency of each word within a collection of categories. The IDF computation can be executed with Eq. (2).

$$idf = \log \log \frac{N}{df_j} \quad (2)$$

In the idf calculation, N is the total of classes, and df_j shows the number of classes j that contain or have word i appear in them. After calculating TF and IDF, the next step is to add TFIDF. The TFIDF calculation can be done using Eq. (3).

$$W_{ij} = tf_{ij} \times idf \quad (3)$$

In the TF-IDF summation, W_{ij} is the sum of the weights of word I in class J , tf_{ij} is the total occurrence of word I in class j , and df_j is the total number of classes j containing word i .

3.4.2 Synthetic Minority Oversampling Technique

SMOTE is able to generate synthetic samples from the minority class by linearly interpolating between the minority samples that are already available and the ones that are closest to them. It is possible to express the SMOTE computation using Eq. (4).

$$x_{new} = x_i + \alpha \cdot (x_{mn} - x_i) \quad (4)$$

A new synthetic sample (x_{new}) in the SMOTE method is generated by performing a linear interpolation between the feature vectors of a minority sample (x_i) and one of its nearest neighbors (x_{mn}) from the same minority class. This synthetic value is calculated by adding x_i to the product of the difference ($x_{mn} - x_i$) with a random number (α) chosen uniformly from the interval $[0,1]$.

3.4.3 Adaptive Synthetic Sampling

ADASYN generates synthetic samples from minority classes by focusing the synthetic sample generation process on areas that are more difficult to separate, particularly data from minority classes located near the boundary between two classes or in areas where other samples are rarely found.

$$g_i = \left(\frac{r_i}{\sum_{j=1}^m r_j} \right) \cdot G \quad (5)$$

The number of synthetic samples (g_i) generated for each minority sample i in the ADASYN method is determined by multiplying the classification difficulty ratio of that sample r_i by the total difficulty of all minority samples ($\sum_{j=1}^m r_j$), then multiplying it by the total number of synthetic samples required (G).

3.4.4 Support Vector Machine

SVM is used to solve classification problems using linear equations or inequalities. By applying kernel techniques to SVM, this model can map data into a higher-dimensional vector space, allowing for optimal data separation with a hyperplane. Thus, problems that initially could not be solved linearly can be solved. The SVM kernel calculation can be seen in Table 2.

In the SVM kernel formulas, there are several important symbols that have special meanings. The vectors x and x' represent the features of the two data being compared, while $x \cdot x'$ represents the dot product or dot product between the two. In the polynomial kernel, the constant c is used to shift

the value of the dot product result and the power d to determine the degree of the polynomial. The RBF or Gaussian kernel involves $\|x - x'\|^2$, which is the squared Euclidean distance between two vectors, and the gamma parameter (γ) which determines how much influence one point has on another. The larger the gamma, the more local the influence. In the sigmoid kernel, the tanh function (hyperbolic tangent) is used with the parameter alpha (α) as a scale for the dot product and beta (β) as a bias or shift. All of these parameters are used to adjust the shape of the kernel function in mapping data to a higher-dimensional space to make it easier to separate by the SVM. Table 3 shows the pseudocode of the SVM algorithm that illustrates the main steps in the training and classification process. This pseudocode SVM algorithm.

Table 2. Support Vector Machine (SVM) kernel equation

Kernel	Equation
Linear	$K(x, x') = x \cdot x'$
Polynomial	$K(x, x') = (x \cdot x' + c)^d$
RBF Gaussian	$K(x, x') = \exp(-\gamma \ x - x'\ ^2)$
Sigmoid	$K(x, x') = \tanh(\alpha x \cdot x' + \beta)$

Note: RBF: Radial Basis Function.

Algorithm 1. Pseudocode Support Vector Machine (SVM) Algorithm

Input:

$D = \{(x_1, y_1), \dots, (x_n, y_n)\}, x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$

Kernel type $K(x_i, x_j)$, e.g.:

- Linear: $K(x_i, x_j) = x_i^T x_j$

- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + \text{coef0})^{\text{degree}}$

- RBF: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$

Hyperparameters: $C > 0, \gamma > 0$ (for RBF), degree (for poly), coef0

Output:

Support vectors x_s , coefficients α_i , bias term b

1: Compute Gram matrix $K \in \mathbb{R}^{n \times n}$:

For $i = 1$ to n :

For $j = 1$ to n :

$K_{ij} \leftarrow K(x_i, x_j)$

2: Solve the following quadratic program:

Maximize: $L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K_{ij}$

Subject to: $0 \leq \alpha_i \leq C, \forall i$

$\sum_i \alpha_i y_i = 0$

3: Select support vectors:

SupportVectors $\leftarrow \{x_i \mid \alpha_i > 0\}$

4: Compute bias b :

Choose any support vector x_s such that $0 < \alpha_s < C$

$b \leftarrow y_s - \sum_i \alpha_i y_i K(x_i, x_s)$

5: Return model:

$f(x) = \sum_i \alpha_i y_i K(x_i, x) + b$

3.4.5 Naïve Bayes Classifier

This algorithm works based on the principle of Bayes' Theorem, which relates the probability of an event to previously known information. The NB kernel calculation can be done using Eq. (6).

$$P(c | x) = \frac{P(x|c) \cdot P(c)}{P(x)} \quad (6)$$

Eq. (5) is a form of Bayes' Theorem, which is used in Naive Bayes classification to calculate the posterior probability of a class c based on an observed feature x . Here, $P(c)$ represents

the probability that data x belongs to class c , $P(c)$ is the likelihood or probability of data x appearing if it is known to be from class c , $P(x)$ is the prior or initial probability of class c , and $P(c)$ is the overall probability of data x across all classes, which serves as a normalization factor to ensure the posterior result is valid as a probability. In practice, since $P(x)$ is the same for all classes when making predictions, it is often ignored and classification is simply done by finding the maximum value of $P(x|c) \cdot P(c)$. Table 4 shows the pseudocode of the NB algorithm that illustrates the main steps in the training and classification process. This pseudocode MVB algorithm.

Algorithm 2. Multinomial Naive Bayes (MNB)

Input:

 $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ // Training data
 $x_i = [x_{i1}, \dots, x_{id}] \in \mathbb{N}^+$ // Count vector of d features (e.g. word counts)
 $y_i \in \{c_1, \dots, c_K\}$ // Class labels
Hyperparameters:
 $\alpha \geq 0$ // Laplace smoothing parameter (e.g. $\alpha = 1$)

Output:

Class prior probabilities $P(c)$,
Feature likelihoods $P(x_j | c)$

Training Phase:

```
1: For each class  $c \in \{c_1, \dots, c_K\}$ :
2:    $N_c \leftarrow$  number of samples with label  $y_i = c$ 
3:    $P(c) \leftarrow N_c / n$  // Prior probability
4:    $T_c \leftarrow 0$  // Total token count for class  $c$ 
5:   For each feature  $j = 1$  to  $d$ :
6:      $T_{\{c_j\}} \leftarrow$  sum of  $x_{ij}$  for all  $x_i$  where  $y_i = c$ 
7:      $T_c \leftarrow T_c + T_{\{c_j\}}$ 
8:   For each feature  $j = 1$  to  $d$ :
9:      $P(x_j | c) \leftarrow (T_{\{c_j\}} + \alpha) / (T_c + \alpha \cdot d)$  // Likelihood with smoothing
```

Prediction Phase:

```
10: Function Predict( $x_{new}$ ):
11:   For each class  $c \in \{c_1, \dots, c_K\}$ :
12:      $\log\_prob[c] \leftarrow \log P(c)$ 
13:     For  $j = 1$  to  $d$ :
14:        $\log\_prob[c] \leftarrow \log\_prob[c] + x_{new}[j] \cdot \log P(x_j | c)$ 
15:   Return class  $c^* = \operatorname{argmax}_c \log\_prob[c]$ 
```

3.4.6 Varian modeling, parameter, and hyperparameter

This study uses two main models, namely SVM and NB, each of which is tested in its basic form, 12 variants as combined with data balancing techniques (SMOTE and ADASYN), and optimized using Grid Search. In SVM, this combination aims to find the best parameters and improve performance on imbalanced data through variations such as

SVM+SMOTE, SVM+ADASYN, and SVM+Grid Search. In NB, a similar approach is applied to assess the effect of balancing and optimization on the stability of probabilistic estimates. Overall, each variant is designed to test how data balancing and hyperparameter optimization contribute to improving minority class detection and overall classification performance. All can see details in Tables 3-5.

Table 3. Varian modeling

No	Varian Model	Description
1	SVM	Baseline SVM without resampling and without parameter optimization.
2	SVM with ADASYN	SVM trained on balanced data using ADASYN to handle adaptive class imbalance.
3	SVM with SMOTE	SVM used a dataset balanced with SMOTE before training.
4	SVM with Grid Search	SVM without resampling, but hyperparameters optimized using GridSearchCV.
5	SVM with Grid Search & SMOTE	SMOTE was applied before training, then GridSearch was used to find the best SVM parameters.
6	SVM with Grid Search & ADASYN	ADASYN was used to balance the data, followed by GridSearch SVM to find the optimal configuration.
7	Naïve Bayes	Baseline Naïve Bayes model without balancing or hyperparameter optimization.
8	Naïve Bayes with ADASYN	Naïve Bayes trained on balanced data using ADASYN.
9	Naïve Bayes with SMOTE	Naïve Bayes trained on the SMOTE dataset.
10	Naïve Bayes with Grid Search	Baseline Naïve Bayes with hyperparameter tuning using GridSearch.
11	Naïve Bayes with Grid Search & SMOTE	Combination of SMOTE and GridSearch in Naïve Bayes to improve generalization.
12	Naïve Bayes with Grid Search & ADASYN	ADASYN is used to balance the data before GridSearch is performed on Naïve Bayes.

Note: SVM: Support Vector Machine; ADASYN: Adaptive Synthetic Sampling; SMOTE: Synthetic Minority Oversampling Technique.

Table 4. Parameter GridSearch (Support Vector Machine & Naïve Bayes)

Algorithm	Parameter	Grid	Function
SVM	C	[0.1, 1, 10, 50, 100]	Controls the error penalty; large values tight margins.
	gamma	['scale', 'auto', 0.01, 0.001]	Manages the influence of each point in the RBF kernel.
	kernel	['rbf']	The best nonlinear kernel for complex patterns.

Naïve Bayes	class_weight var_smoothing pha	[None, 'balanced'] [1e-9, 1e-8, 1e-7] [0.1, 0.5, 1.0]	Automatic weighting option for imbalanced data. Stability of probability calculations (preventing division by zero). Laplace/Lidstone smoothing to prevent zero probabilities.
-------------	--------------------------------------	---	--

Note: SVM: Support Vector Machine; RBF: Radial Basis Function.

Table 5. Optimal hyperparameter

Variance	Optimal Hyperparameter	Value	F1-Weighted	Interpretation
SVM with Grid Search	C	10	0.88	SVM is more stable after parameter tuning. The boundary is smoother and more general.
SVM with Grid Search & SMOTE	gamma	0.01	0.89	SMOTE provides a stable distribution for SVM. Suitable for data with moderate feature variation.
SVM with Grid Search & ADASYN	C	10	0.91	ADASYN helps focus the model on difficult minority points.
Naïve Bayes with Grid Search	gamma	0.01	0.8	Provides optimal separation between classes. Moderate smoothing produces more stable probabilities.
Naïve Bayes with Grid Search & SMOTE	alpha	0.5	0.82	SMOTE improves minority representation.

Note: SVM: Support Vector Machine; ADASYN: Adaptive Synthetic Sampling; SMOTE: Synthetic Minority Oversampling Technique.

3.5 Evaluation

The SVM and Naive Bayes models used a pre-trained dataset of PayLater service reviews, both with and without the Grid Search, ADASYN, and SMOTE methods. Model performance was assessed and compared using several evaluation metrics, namely precision, accuracy, recall, and f1-score. This evaluation process is an essential stage in machine learning model development, as it allows for a comprehensive assessment of model effectiveness. Various metrics were used to gain a deeper understanding of model performance, particularly when dealing with unlabeled or imbalanced data. The following metrics served as the primary reference for the analysis and comparison of results in this study [22-24].

1. Accuracy is the ratio of the number of accurate or correct predictions, based on the accuracy of predicting positive and negative labels, to the total amount of available data. It can be calculated using Eq. (7).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

2. Precision is a measurement parameter used to evaluate the model's ability to make accurate predictions for the positive class, by considering the total number of data identified as the positive class. Precision calculations are carried out by dividing the number of correct positive predictions by the total number of positive predictions. The calculation can be done using Eq. (8).

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

3. Recall is a metric that reflects the model's ability to recognize or identify data that is actually classified as positive. In its calculation, Recall measures the extent to which the model is able to correctly identify the number of positive data from the total sample that is actually labeled positive. The calculation can be done using Eq. (9).

$$Recall = TP/(TP + FN) \quad (9)$$

4. F1-score is a measurement parameter that reflects the

relationship between precision and recall that has been previously calculated. This parameter provides an overview of how the model achieves a balance between the ability to correctly classify the positive class (precision) and in comprehensively identifying all entities that are actually positive (recall). The calculation can be done using Eq. (10).

$$F1 - Score = \frac{Precision*Recall}{Precision+Recall} \quad (10)$$

5. True Positive (TP) refers to a positive condition that is accurately predicted by the model, as shown in the previous equation. Meanwhile, True Negative (TN) represents a negative condition that is also correctly predicted. Conversely, False Positive (FP) occurs when the model incorrectly predicts a negative case as positive. False Negative (FN) is a prediction error where a positive case is incorrectly classified as negative.
6. Deploy. In the deployment phase, the developed model is implemented as a web application using the Streamlit library and the Python programming language. This application is designed to automatically predict user sentiment toward the PayLater service. The prediction process utilizes the best algorithm selected based on previous model performance evaluations, allowing the analysis results to be used as a basis for decision-making.

4. RESULTS AND EVALUATION

4.1 Business understanding

The goal of this study is to analyze how the general public of Indonesia perceives the PayLater service by analyzing the sentiment in the messages left on social media platform X (Twitter). The goal is to evaluate PayLater's effect based on the opinions of the general people. The goal of this research was to build an analysis model (Sentiment Analysis) by comparing the performance of two types of machine learning algorithms. Those algorithms were the NBC and the SVM. The model was evaluated comprehensively with the help of accuracy, precision, recall, and F1-score metrics extracted from the confusion matrix. Due to the fact the dataset was in a

category imbalance, SMOTE and ADASYN oversampling methods were used. These techniques were subsequently tuned through parameter tuning by using a grid search method for parameter fine tuning. The best performing model was then integrated into a website-based system that was integrated with SahabatAI's Large Language Model (LLM) Gamma 9:b model. Furthermore, this was equipped with data preprocessing capabilities (noise cleaning and text normalization) and visualization of analysis results (word cloud, sentiment distribution and the dominant keyword bar chart). It is hoped that the results of this study will serve to formulate more accurate techniques for sentiment analysis, and that they will be a means of reference for fintech companies, regulators and those studying the public's response towards PayLater services.

4.2 Data understanding

The data collection results indicate that 1,052 tweets relevant to the PayLater service topic were successfully obtained through an extraction process using the Tweet-Harvest tool with pre-defined search parameters. The data was stored in Comma-Separated Values (CSV) format for compatibility and ease of processing using analytical tools such as Python and Excel. Figure 3 displays a sample of the collected data.

conversation_id_str	created_at	favorite_count	full_text	id_str	image_url	in_reply_to_screen_name	lang	location
0	Thu Feb 21 08:21:27 +0000 2025	0	@SapangyaCamen Baru lgd nyweet ini kalo em...	18924873266712960	NaN	SapangyaCamen	in	NaN
1	Thu Feb 20 13:25:31 +0000 2025	0	@renesant Gak ada notif ah lagi kalo buka L...	18925942541682008	NaN	renesant	in	Bandung, Indonesia
2	Thu Feb 20 08:13:54 +0000 2025	1	Sobat L@j@ja bir ga bener @ba sba mending...	18924873266712960	https://pbs.twimg.com/media/GK4Sc0rfsJAMn4Zh.jpg	NaN	in	NaN
3	Wed Feb 19 03:55:20 +0000 2025	1	@ulfahd di domi ada flour b@k@ nomor t@k...	189215591419082096	NaN	ulfahd	in	NaN
4	Wed Feb 19 03:55:20 +0000 2025	0	Aktikan Flour DAMA Paylater Lewat E-mail T@p...	1892121042560153904	NaN	NaN	in	NaN

Figure 3. Data collection result

4.3 Data preparation

The data pre-processing process in sentiment analysis involves several important steps to improve data quality and model accuracy. The first step is removing null values and unimportant features, which removes rows with empty (null) values and features that are irrelevant or do not contribute significantly to the analysis results. In this case, only the full_text feature is retained because it contains Twitter user comments regarding the PayLater service, which is the primary object of the analysis. Data cleaning is then performed to ensure the data is clean and ready for further processing. After that, tokenization is performed, which breaks down the text into smaller units such as words, phrases, or characters so that the text structure can be analyzed more systematically. The process continues with stemming, which returns each word to its base form by removing affixes. Slang word handling is then performed to transform non-standard or slang words into formal forms so that the meaning and sentiment in the text can be recognized more accurately. The final stage in pre-processing is stopword removal, which removes common words that do not contribute significantly to the analysis, allowing the model to focus more on meaningful words. After this stage, the detokenizing process is performed because the TF-IDF method is implemented using the scikit-learn library,

which does not require tokenized data; it automatically performs tokenization during the feature extraction process. The final results of all preprocessing stages are shown in Table 6.

Table 6. Preparation result

Before Preparation	After Preparation
@n4ruebet Iya yang paling penting jangan pakai fitur PayLater karena kepepet. Udah pasti jadi boomerang.	iya pakai fitur PayLater pepet boomerang
@tanyarlfs kalo bisa hindari pake PayLater/kartu kredit nder. emg sih awal2 membantu tp lama2 malah gaji keburu abis bayarin bunganya sayang bgt. apalagi sPayLater bunganya kan gede nder lumayan buat nambahin jajan2 mah	hindar pakai PayLater kartu kredit bang emang sih bantu gaji terlanjur habis bayarin bunga sayang banget sPayLater bunga gede bang lumayan nambahin jajan mah
@4deruby PayLater hadir untuk membantu kebutuhan anda	PayLater hadir bantu butuh

After the data was prepared, the next step was data labeling using a semi-supervised learning method with the SVM algorithm, combined with a Self-Training Classifier and TF-IDF text representation. A total of 598 of the 1,052 data sets were manually labeled as initial training data, while the rest were used as unlabeled data. The trained SVM model then predicted the labels of the unlabeled data, and only predictions with a confidence level of $\geq 70\%$ were automatically labeled. This process was carried out iteratively to expand the amount of labeled data without extensive manual labeling.

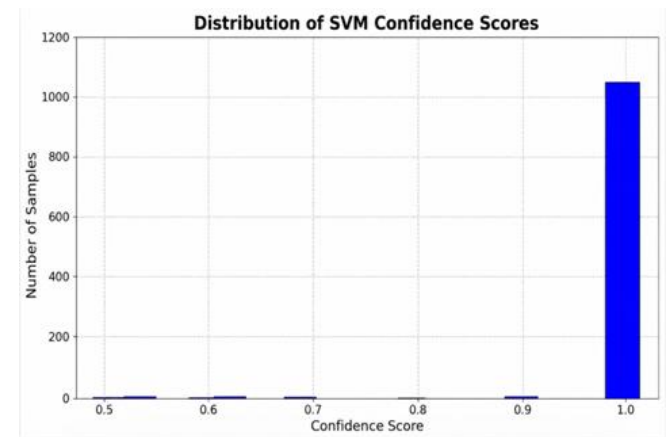


Figure 4. Semi-supervised Support Vector Machine (SVM) result

The classification results show that the Self-Training SVM model produces very high prediction scores for the automatically labeled data. Based on the prediction score distribution shown in Figure 4, most confidence values exceed the 0.7 threshold. With the majority of samples reaching the maximum value of 1.0, the histogram shows that more than 1,000 samples fall exactly at this value and only a small fraction fall below it, reflecting the model's success in selectively and reliably recognizing sentiment patterns and effectively extending automatic labeling with minimal error based on a certain confidence threshold.

The sentiment distribution generated by the semi-supervised SVM model in Figure 5 shows a higher number of negative opinions (65.30% of the data) compared to positive

4.4.6 Support Vector Machine algorithm modeling with grid search

The classification model was built using the SVM algorithm without applying any data sampling techniques. Parameter optimization was performed using GridSearchCV to explore various combinations of C, kernel, and gamma values. The C values tested were [0.1, 1, 10, 100], with kernel options including ['linear', 'rbf', 'poly', 'sigmoid'], and the gamma values tested were [0.01, 0.1, 1, 10]. The optimization process was performed with cross-validation ($cv = 10$) to ensure more stable results and avoid overfitting. Of the various parameter combinations tested, the best configuration was obtained with $C = 10$, $\gamma = 0.1$, and RBF kernel.

4.4.7 Support Vector Machine algorithm modeling with grid search and Adaptive Synthetic Sampling

The oversampling process was performed using ADASYN with the parameters $\text{sampling_strategy} = 0.9$ and $n_neighbors = 5$. The sampling_strategy parameter determines the proportion of the minority class to the majority class after resampling, which in this case is close to balanced (90%). Meanwhile, $n_neighbors$ indicates the number of nearest neighbors used to generate synthetic data based on the local distribution.

After the data was balanced, the SVM model was optimized using GridSearchCV. The tested parameters included C ([0.1, 1, 10, 100]), kernel (['linear', 'rbf', 'poly', 'sigmoid']), and gamma ([0.01, 0.1, 1, 10]). This process used 10-fold cross-validation ($cv = 10$) to improve model reliability, as well as $n_jobs = -1$ to speed up computation and $verbose = 1$ to display the process. The best results were obtained with $C = 10$, $\gamma = 0.1$, and kernel = 'rbf'.

4.4.8 Modeling the Support Vector Machine algorithm with grid search & Synthetic Minority Oversampling Technique

SMOTE was used with $\text{sampling_strategy} = 1$, meaning the minority class was increased to 100% of the majority class. The $k_neighbors = 19$ parameter determines the number of nearest neighbors used in generating synthetic samples. After the data was balanced, the optimal parameters for the SVM model were searched using GridSearchCV. The search space included values of C ([0.1, 1, 10, 100]), kernel (['linear', 'rbf', 'poly', 'sigmoid']), and gamma ([0.01, 0.1, 1, 10]). This process used 10-fold cross-validation, $\text{random_state} = 42$ for consistency, and $n_jobs = -1$ to speed up computation. The final results showed that the best parameter combination was $C = 10$, $\gamma = 0.1$, and kernel = 'rbf'.

4.4.9 Naive bayes algorithm modeling

The classification model development process was carried out using the Naive Bayes algorithm without applying data sampling techniques or parameter optimization. The model was built using the default configuration, without adjustments to parameters such as α , fit_prior , or force_alpha . Model validation was performed using a 10-fold cross-validation technique ($cv = 10$) to ensure the reliability of the results and reduce the risk of overfitting.

4.4.10 Naive bayes algorithm modeling with Adaptive Synthetic Sampling

The Naive Bayes model was implemented using the ADASYN method with the default configuration, namely a linear kernel, without any parameter optimization. The parameters used to balance the class distribution in the training

data were $\text{sampling_strategy} = 0.9$ and $n_neighbors = 4$. Model validation was performed using a 10-fold k-fold cross-validation method to evaluate model performance.

4.4.11 Naive bayes algorithm modeling with Synthetic Minority Oversampling Technique

The Naive Bayes model was implemented using the SMOTE method with the default configuration, namely a linear kernel, without any parameter optimization. The parameter used is $\text{sampling_strategy} = 0.85$, meaning the minority class is increased to 85% of the majority class. The parameter $K_neighbors = 3$ functions to determine the number of nearest neighbors used in the formation of synthetic samples. Model validation is carried out using the k-fold cross-validation method with 10 folds to evaluate model performance.

4.4.12 Naïve Bayes algorithm modeling with grid search

The NB model was built without applying data balancing techniques. To improve model performance, hyperparameter optimization was performed using GridSearchCV by exploring various combinations of alpha values ([0.01, 0.05, 0.1, 0.5, 1.0, 1.5, 2.0, 5.0, 10.0]), fit_prior ([True, False]), and force_alpha ([True, False]). The optimal parameter search process was performed using 10-fold cross-validation to ensure model stability and reduce the risk of overfitting. Based on the optimization results, the best parameter combination was $\alpha = 1.0$, $\text{fit_prior} = \text{False}$, and $\text{force_alpha} = \text{True}$.

4.4.13 Naïve Bayes algorithm modeling with grid search & Adaptive Synthetic Sampling

The NB model was optimized after applying the ADASYN method, which was used to balance the class distribution in the training data with parameters $\text{sampling_strategy} = 0.9$ and $n_neighbors = 4$. ADASYN increased the number of minority class samples to approximate the number of majority class samples. Hyperparameter optimization using Grid Search was performed to improve model accuracy, exploring combinations of alpha, fit_prior , and force_alpha values. This process was validated using 10-fold cross-validation to ensure model generalization and reduce the risk of overfitting. The best results were obtained at $\alpha = 1.5$, $\text{fit_prior} = \text{True}$, and $\text{force_alpha} = \text{True}$.

4.4.14 Naïve Bayes algorithm modeling with grid search & Synthetic Minority Oversampling Technique

SMOTE was applied to the training data with parameters $\text{sampling_strategy} = 0.90$, $k_neighbors = 3$, and $\text{random_state} = 42$. This configuration aims to increase the number of samples in the minority class to reach 90% of the number of samples in the majority class, by considering the 3 nearest neighbors in the synthetic data generation process. After the class distribution is balanced using SMOTE, optimization is carried out on the Multinomial NB model using the Grid Search method. The goal is to find the best combination of hyperparameters to improve the performance of the classification model. The parameters tested include alpha with a value range of [0.01, 0.05, 0.1, 0.5, 1.0, 1.5, 2.0, 5.0, 10.0], as well as fit_prior and force_alpha each with a value of [True, False]. Validation is carried out using the 10-fold cross-validation technique. Based on the search results, the best parameter combination was obtained at $\alpha = 1.0$, $\text{fit_prior} = \text{True}$, and $\text{force_alpha} = \text{True}$.

4.5 Evaluation

The evaluation of the NB and SVM algorithms in this study will be compared with the results of previous studies, as shown in Table 7. Paramitha [10], and Rifqi Rizaldi and Riska Aryanti [12]. This comparison aims to evaluate the algorithm's performance in analyzing public sentiment towards PayLater services, taking into account the application of data balancing techniques (SMOTE and ADASYN) and hyperparameter optimization through GridSearch.

Table 7. Comparison of evaluation result with previous research

Authors	Ferra Junian Wahidna & Paramitha	Rifqi Rizaldi & Riska Aryanti
Algorithm	SVM	Naïve Bayes
Accuracy	90,27%	87%
Precision	92,28%	86%
Recall	90,27%	92%
F1-Score	89,84%	89%

The model accuracy results can be seen in Figure 8. The SVM algorithm combined with Grid Search and hyperparameter tuning in SMOTE produced an accuracy of 94.31%, indicating strong classification performance. Compared to previous studies using a similar approach—such as the study by Ferra Junian Wahidna and Paramitha [10], which reported an accuracy of 90.27% using SVM and SMOTE—this finding suggests potential performance improvements that could be attributed to the more comprehensive optimization process employed in this study.

Similarly, compared to the study by Rifqi Rizaldi and Riska Aryanti [12], which used the NB algorithm without optimization and only achieved an accuracy of 87%, the NB variant in this study, equipped with Grid Search and SMOTE, achieved an accuracy of 91.94%. This difference indicates a trend of improved performance thanks to the implementation of optimization methods and data balancing.

Table 8. Models performance result

Algorithm	Accuracy	Precision	Recall	F1-Score
SVM	93.36%	94%	91%	92%
SVM with ADASYN	92.42%	92%	92%	92%
SVM with SMOTE	93.36%	93%	92%	93%
SVM with Grid Search	93.84%	94%	92%	93%
SVM with Grid Search & SMOTE	94.31%	94%	94%	94%
SVM with Grid Search & ADASYN	93.84%	93%	93%	93%
Naïve Bayes	73.93%	83%	62%	61%
Naïve Bayes with ADASYN	89.57%	91%	86%	88%
Naïve Bayes with SMOTE	91.94%	92%	90%	91%
NAIVE BAYES with Gridsearch	86.26%	90%	81%	83%
NAIVE BAYES with Gridsearch & SMOTE	91.94%	92%	90%	91%
NAIVE BAYES with Gridsearch & ADASYN	91.47%	93%	89%	90%

Note: SVM: Support Vector Machine; ADASYN: Adaptive Synthetic Sampling.

Based on the performance results of all models shown in Table 8, the evaluation was conducted using four main metrics, namely accuracy, precision, recall, and F1-score. The SVM model without hyperparameter optimization produced an accuracy of 93.36%, precision of 93%, recall of 91%, and F1-score of 93%. After the application of GridSearch, the model performance improved with an accuracy of 93.84%, precision of 94%, recall of 92%, and F1-score of 93%. The application of SVM with SMOTE without Grid Search showed lower performance than the optimized version, with an accuracy of 93.36%, precision of 93%, recall of 92%, and F1-score of 93%. While the SVM with ADASYN without Grid Search only achieved an accuracy of 92.42%, precision of 92%, recall of 92%, and F1-score of 92%. The application of the SMOTE and ADASYN data balancing techniques with GridSearch produced similar results: 94.31% accuracy, 94% precision, 94% recall, and 94% F1-score. While both data balancing techniques were effective in improving model performance, SMOTE slightly outperformed the former, producing higher training data accuracy, precision, and overall more balanced results in analyzing public sentiment toward PayLater services.

The NB model in this study performed relatively lower, with 73.93% accuracy, 83% precision, 62% recall, and 61% F1-score. However, after applying GridSearch, Naïve Bayes' performance improved to 86.26% accuracy, 90% precision, 81% recall, and 83% F1-score. Using ADASYN without Grid Search actually decreased performance, with 89.57% accuracy, 91% precision, 86% recall, and 88% F1-score. Using NB with SMOTE achieved 91.94% accuracy, 92% precision, 90% recall, and 91% F1-score. The addition of SMOTE and ADASYN data balancing techniques to Naïve Bayes also resulted in significant improvements, with 91.94% accuracy and 91% F1-score for SMOTE, and 91% accuracy and 89% F1-score for ADASYN. However, NB still lags behind SVM, especially in terms of accuracy and F1-score,

Accuracy Comparison (Ascending) of SVM and Naive Bayes Variants

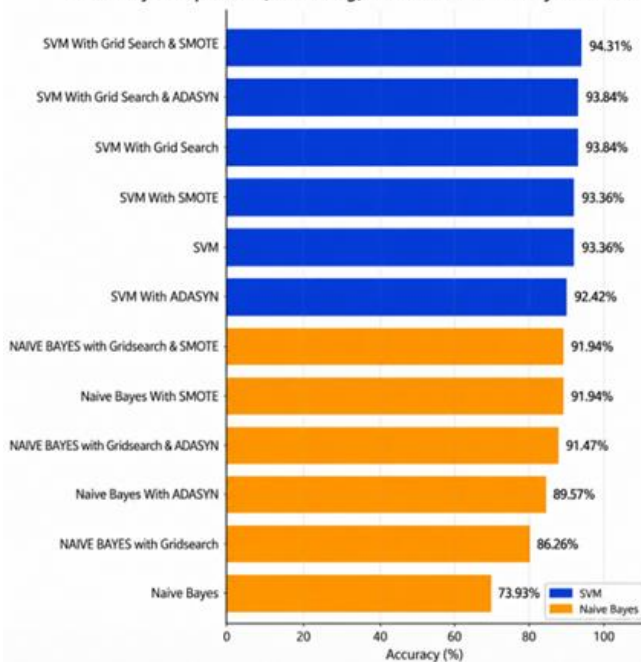


Figure 8. Comparison algorithm

which are crucial for ensuring accuracy in analyzing public sentiment towards PayLater services.

Based on the results obtained, the SVM model with SMOTE and GridSearch proved to be the best model in this study. In addition to producing the highest accuracy (94.31%), this model also demonstrated superior performance in terms of precision, recall, and F1-score (94%) compared to the SVM model using ADASYN. When compared to NB, which produced lower scores in almost all evaluation metrics, SVM with SMOTE demonstrated its superiority in addressing data imbalance issues, with SMOTE providing slightly higher precision and more consistent results.

In addition, this study also presents a Comparison of Model Performance (Manual vs. Semi-Supervised) from result analysis a good algorithm is SVM, can see in Table 9.

A comparison of the training results shows that using only manually labeled data yields an accuracy of 91.82%. After expanding the dataset through a semi-supervised step, accuracy increases to 94.31%. Similar improvements are seen in precision, recall, and F1-score, each of which increases by approximately 2&3 points. These results indicate that the semi-supervised process provides significant added value by enriching the variety of text patterns in the training data. With a larger and more diverse dataset, the model can recognize sentiment patterns more accurately.

Table 9. Model performance comparison (manual vs. semi-supervised)

Model / Dataset	Accuracy	Precision	Recall	F1-Score
SVM – Manual Label Only	91.82%	92.10%	91.40%	91.74%
SVM – Semi-Supervised	94.31%	94.60%	93.90%	94.24%

Note: SVM: Support Vector Machine.

4.6 Deployment

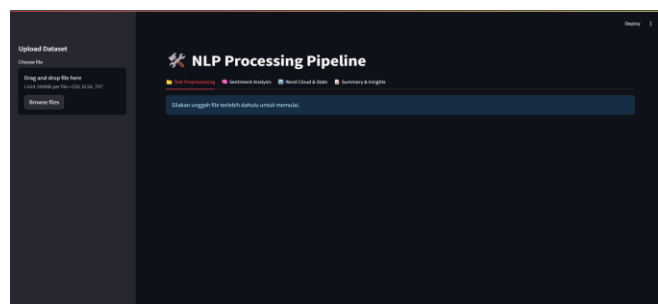


Figure 9. Input file deploy

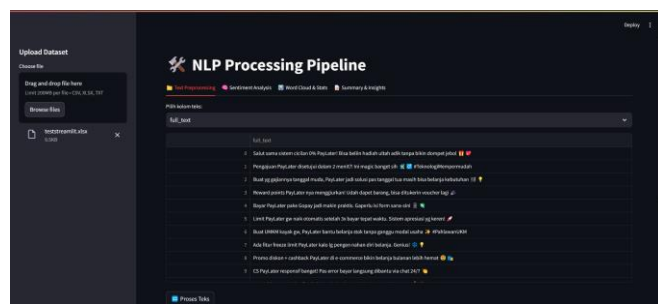


Figure 10. Before processing deploy

During this stage, a simple website was developed using the Streamlit library. This application also features explanations for each word in the text based on the LLM Gemma 2:9b Sahabat AI model, so users can understand the reasoning behind the classification results. The application begins the process by uploading comment or tweet data via the Upload Dataset menu. The raw data is displayed in the Text Processing tab, then cleaned using the Process Text button. The preprocessed results are used in the Sentiment Analysis tab, where the system classifies sentiment using an SVM model and provides an option to download the results. The Word Cloud & Stats tab displays sentiment distribution, word clouds, word frequencies, and word networks to understand user patterns. Finally, the Summary & Insight tab presents a summary of keywords, review citations, sentiment impact, and recommendations. All can see in Figures 9–17 for all actives application.

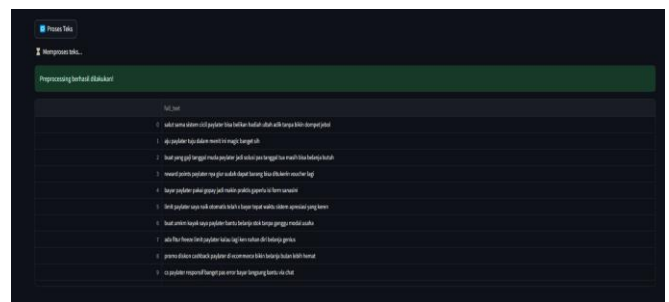


Figure 11. After processing deploy

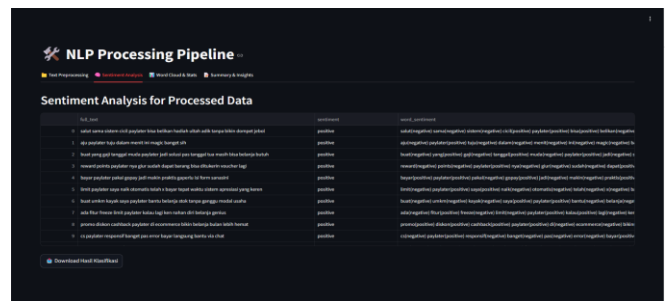


Figure 12. Tab sentiment analysis deploy

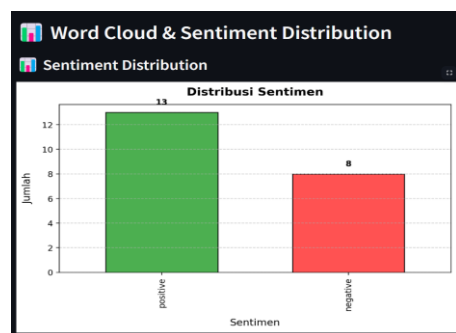


Figure 13. Sentiment distribution deploy



Figure 14. Word cloud deploy

The final stage of this research was the deployment process.

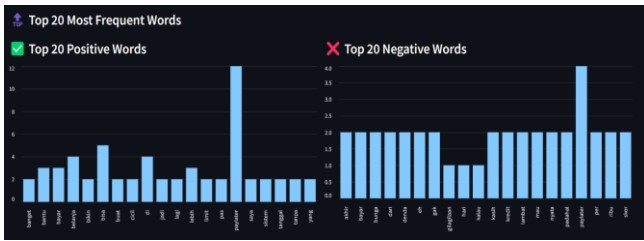


Figure 15. Barplot frequent words deploy

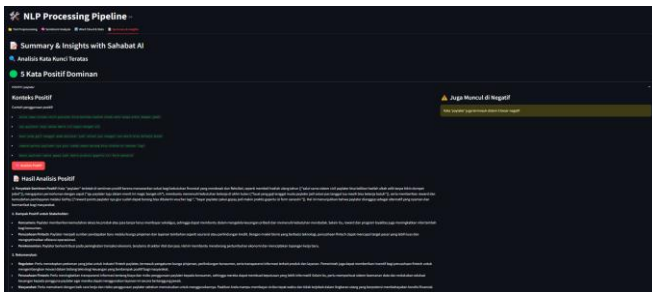


Figure 16. Summary output positive deploy

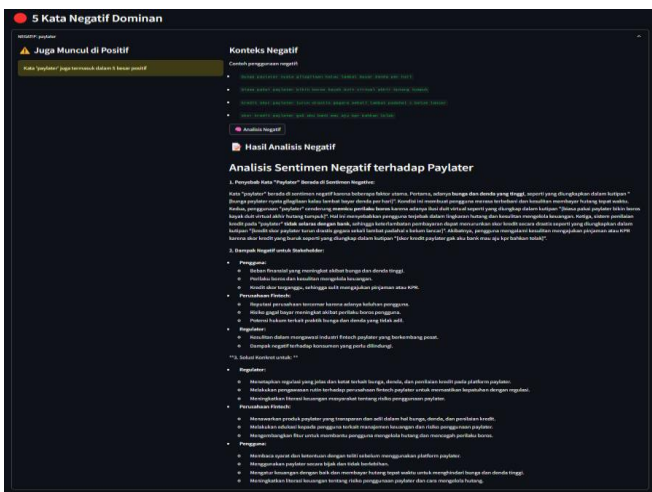


Figure 17. Summary output negative deploy

5. CONCLUSIONS

The results research in the Machine Learning modeling, a comparison between the SVM and NB algorithms, the ADASYN and SMOTE data balancing techniques, and parameter optimization using grid search, indicates that the best model for sentiment analysis of opinions related to PayLater services is SVM with SMOTE and optimization with Grid Search. This model is able to handle data imbalance more effectively and produces superior performance, with an accuracy of 94.31%, a precision of 94%, a recall of 94%, and an F1-score of 94%. This performance consistently outperforms the NB model. These findings indicate that the SVM algorithm is superior and more efficient at classifying sentiment in imbalanced datasets than NB.

The researchers have also successfully implemented the model in a website-based system that allows users to perform sentiment analysis independently. This system is equipped with data preprocessing features, visualizations such as word clouds, sentiment distributions, and dominant word bar charts, as well as integration with SahabatAI's LLM Gamma 9:b

model. This integration feature allows users to obtain recommendations, identify positive or negative impacts, and understand the causes of the emergence of keywords within a particular sentiment. This system overcomes technical challenges in data processing, and can be a valuable tool for regulators and fintech companies to monitor public opinion in real time, identify root causes, and formulate more effective, evidence-based policies. Overall, this research successfully integrates data analysis, predictive modeling, and technology implementation, which can significantly contribute to improving the quality of PayLater services through a data-insight-driven approach.

ACKNOWLEDGMENT

Thank you to Multimedia Nusantara University for providing space and time to conduct research and funding the publication process.

REFERENCES

- [1] Prasetya, A.N.E. (2023). Analisis adanya pay later dalam marketplace terhadap daya beli masyarakat. *Jurnal Revenue: Jurnal Ilmiah Akuntansi*, 3(2): 593-601.
- [2] Sari, R. (2021). Pengaruh Penggunaan PayLater TERHADAP perilaku impulse buying pengguna e-commerce di Indonesia. *Jurnal Riset Bisnis Dan Investasi*, 7(1): 44-57. <https://doi.org/10.35313/jrbi.v7i1.2058>
- [3] Habiba, S., Sissah, S., Siregar, E.S. (2024). Analisis penggunaan fitur shopee PayLater dalam perspektif mahasiswa perbankan syariah febi uin sts jambi. *Jurnal Akademik Ekonomi Dan Manajemen*, 1(3): 170-184. <https://doi.org/10.61722/jaem.v1i3.2582>
- [4] Yonatan, A.Z. (2024). Pengguna PayLater Indonesia tumbuh 17 kali lipat dalam 5 tahun terakhir. *GoodStats*.
- [5] Hardhika, R.E.B. (2021). Pengalaman pengguna PayLater mahasiswa di Surabaya. *The Commercialium*, 4(2): 19-32. <https://doi.org/10.26740/tc.v4i2.41291>
- [6] Alvida, D., Maya, A.N., Riska, A., Vyanara, A., Radita, D.E.M., Rama, W.A.R. (2024). Dampak penggunaan aplikasi PayLater terhadap gaya hidup masyarakat. *Akuntansi Pajak Dan Kebijakan Ekonomi Digital*, 1(2): 51-60. <https://doi.org/10.61132/apke.v1i2.75>
- [7] Aditiya, V., Sari, N., Suryani, L. (2024). Pengaruh media sosial dan gaya hidup terhadap perilaku konsumtif pada pengguna SPayLater di Shoope. *Innovative: Journal of Social Science Research*, 4(1): 10429-10441. <https://doi.org/10.31004/innovative.v4i1.9085>
- [8] Kristiyanti, D.A., Sitanggang, I.S., Nurdiani, S. (2023). Feature selection using new version of v-shaped transfer function for salp swarm algorithm in sentiment analysis. *Computation*, 11(3): 56. <https://doi.org/10.3390/computation11030056>
- [9] Hermawan, M.G. (2023). Klasifikasi bot twitter dengan menggunakan random forest classifier. *Politeknik Negeri Jember*.
- [10] Wahidna, F.J., Nerisafitra, P. (2023). Analisis sentimen pengguna sistem pay later menggunakan support vector machine metode pembobotan lexicon. *Journal of Informatics and Computer Science (JINACS)*, 4(3): 334-343. <https://doi.org/10.26740/jinacs.v4n03.p334-343>

- [11] Mahayani, I., Agushinta, D., Supriyadi, M.E. (2020). Analisis sentimen twitter terhadap pembayaran ShopeePayLater pada aplikasi belanja online (Shopee) menggunakan metode lexicon based dan naive bayes classifier. *Jurnal Ilmiah KOMPUTASI*, 19(4): 545-558. <https://doi.org/10.32409/jikstik.19.4.293>
- [12] Rizaldi, R., Aryanti, R. (2024). Analisis sentimen pengguna terhadap aplikasi indodana di google play store menggunakan metode naive bayes classifier. *Journal of Informatics Management and Information Technology*, 4(3): 98-105. <https://doi.org/10.47065/jimat.v4i3.400>
- [13] Safira, A., Hasan, F.N. (2023). Analisis sentimen masyarakat terhadap PayLater menggunakan metode naive bayes classifier. *ZONASI: Jurnal Sistem Informasi*, 5(1): 59-70. <https://doi.org/10.31849/zn.v5i1.12856>
- [14] Kristiyanti, D.A., Sanjaya, S.A., Tjokro, V.C., Suhali, J. (2024). Dealing imbalance dataset problem in sentiment analysis of recession in Indonesia. *IAES International Journal of Artificial Intelligence*, 13(2): 2058-2070. <https://doi.org/10.11591/ijai.v13.i2.pp2060-2072>
- [15] Cahyana, N.H., Fauziah, Y., Wisnalmawati, W., Aribowo, A.S., Saifullah, S. (2025). The evaluation of effects of oversampling and word embedding on sentiment analysis. *Jurnal Infotel*, 17(1): 54-67. <https://doi.org/10.20895/infotel.v17i1.1077>
- [16] Kumar, A., Dutt, V., García-Díaz, V., Narang, S.K. (2021). Twitter sentimental analysis from time series facts: The implementation of enhanced support vector machine. *Bulletin of Electrical Engineering and Informatics*, 10(5): 2845-2856. <https://doi.org/10.11591/eei.v10i5.3078>
- [17] Kusnawi, K., Rahardi, M. (2023). Sentiment analysis of neobank digital banking using support vector machine algorithm in Indonesia. *JOIV: International Journal on Informatics Visualization*, 7(2): 377-383. <https://doi.org/10.30630/joiv.7.2.1652>
- [18] Putra, A.D., Pudjiantoro, T.H., Renaldi, F., Hadiana, A.I. (2022). Analyzing sentiments on official online lending platform in Indonesia with a combination of naive bayes and lexicon based method. In 2022 International Conference on Science and Technology (ICOSTECH), Batam City, Indonesia, pp. 1-6. <https://doi.org/10.1109/ICOSTECH54296.2022.9829141>
- [19] Karmagatri, M., Aziz, C.F.A., Asih, W.R.P., Jumbri, I.A. (2023). Uncovering user perceptions toward digital banks in Indonesia: A Naïve Bayes sentiment analysis of Twitter data. *Journal of Theoretical and Applied Information Technology*, 101(12): 4960-4968.
- [20] Wisnalmawati, W., Aribowo, A.S., Herawati, Y. (2022). Semi-supervised learning models for sentiment analysis on marketplace dataset. *International Journal of Artificial Intelligence & Robotics*, 4(2): 78-85. <https://doi.org/10.25139/ijair.v4i2.5267>
- [21] Putri, D.D., Nama, G.F., Sulistiono, W.E. (2022). Analisis sentimen kinerja dewan perwakilan rakyat (DPR) pada twitter menggunakan metode naive bayes classifier. *Jurnal Informatika dan Teknik Elektro Terapan*, 10(1): 34-40. <https://doi.org/10.23960/jitet.v10i1.2262>
- [22] Nuraeni, A., Erfina, A., Sukmawan, D. (2023). Sentiment analysis of Indonesian people's response against the PayLater payment method using the naive bayes algorithm. *Jurnal Bisnis dan Manajemen*, 3(4): 566-573.
- [23] Kurniadi, D., Fernando, E., Fauziyah, A., Mulyani, A. (2025). Improving low-light face recognition using deepface embedding and multi-layer perceptron. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 9(5): 1047-1055. <https://doi.org/10.29207/resti.v9i5.6797>
- [24] Fernando, E., Ikhsan, R.B. (2024). Exploring integration AI-powered in supply chain management. In 2024 International Conference on Decision Aid Sciences and Applications (DASA), Manama, Bahrain, pp. 1-5. <https://doi.org/10.1109/DASA63652.2024.10836368>