International Journal of Research in Engineering, Science and Management
Volume 9, Issue 7, July 2026
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing | www.resaim.com

Analyzing Public Sentiment Toward Prabowo Subianto on X: A Multinomial Naive Bayes Approach with SMOTE

Johanes Eka Priyatma[1*], Ariyova Banua[1]
[1]*Department of Informatics, Sanata Dharma University, Yogyakarta, Indonesia*

***Abstract*: Social media platform X serves as a major source of public opinion on political figures, including Indonesian President Prabowo Subianto. This study aims to conduct sentiment analysis on X user posts related to Prabowo Subianto during the period from June 2023 to June 2024 using the Multinomial Naive Bayes algorithm. The dataset consists of 42,199 posts collected through crawling X. The data were processed through text preprocessing stages, including cleansing, tokenization, normalization, stopword removal, stemming, and text translation, followed by feature weighting using Term Frequency–Inverse Document Frequency (TF-IDF). This study compares model performance with and without data balancing using the Synthetic Minority Over-sampling Technique (SMOTE) and applies K-Fold Cross-Validation for model validation. Model evaluation uses accuracy, precision, recall, and F1-score metrics. The results show that applying SMOTE improves model performance and enables more balanced sentiment classification. Overall, the Multinomial Naive Bayes algorithm proves effective for sentiment analysis of X social media data related to President Prabowo Subianto during the 2023–2024 period.**

***Keywords*: Multinomial Naive Bayes, Prabowo Subianto, Sentiment Analysis, SMOTE, TextBlob, TF-IDF, X (Twitter), VADER Lexicon.**

1. Introduction

The use of social media as a medium for measuring public sentiment continues to increase alongside the growing intensity of online opinion expression. Platform X (Twitter) has become one of the primary data sources because it provides large volumes of real-time opinion text, particularly related to political issues [6]. Data from X reflect spontaneous public responses to political events, public figures, and government policies. Sentiment analysis of such data plays an important role in understanding public political preferences and the factors that influence them. This information holds strategic value for political actors and policymakers in formulating decisions based on public opinion.

In computer science, sentiment analysis is a field within natural language processing and machine learning. It aims to identify and classify opinions or emotions in text into specific categories, such as positive, negative, or neutral [5]. One of the most widely used algorithms in sentiment analysis is Naïve Bayes. This algorithm classifies text based on the probability of word occurrences within each sentiment class [8]. The main advantages of Naive Bayes include its simple model structure, computational efficiency, and stable performance when applied to large-scale textual data [12].

Previous studies indicate that Naive Bayes has been extensively applied to analyze political sentiment in Indonesia, yielding varying levels of accuracy. A research [6] identified a dominance of negative sentiment regarding alleged violations in the 2024 General Election, achieving an accuracy of 84.13%. Another research [1] reported an increase in negative sentiment following the announcement of Gibran as a vice-presidential candidate, with accuracy ranging from 42.86% to 67.80%. Meanwhile, other research [15] demonstrated that Naive Bayes outperformed SVM in analyzing election postponement issues, achieving an accuracy of 98.80%. Other studies reported accuracies ranging from 71% to 95.71%, while highlighting challenges such as class bias and data imbalance. Overall, these studies confirm the effectiveness of Naive Bayes for political sentiment analysis, although performance is influenced by dataset size, class distribution, and labelling methods.

This study focuses on analyzing public sentiment toward Prabowo Subianto from June 2023 to June 2024 using data from platform X. Unlike previous studies that typically examined specific issues or limited timeframes, this research analyzes sentiment toward a political figure over a one-year period, covering both pre-election and post-election phases. The study employs a Multinomial Naive Bayes classifier combined with text preprocessing, TF-IDF feature weighting, data imbalance handling using SMOTE, and model evaluation through K-fold cross-validation. This approach aims to improve sentiment classification quality and produce a more objective performance evaluation.

Through this research, a more comprehensive understanding of public sentiment toward Prabowo Subianto based on social media data is expected. Additionally, the findings are anticipated to contribute empirically to the development of political sentiment analysis methods in Indonesia, particularly in applying the Naive Bayes algorithm to Indonesian-language text data.

*Corresponding author: maseko1963@gmail.com

2. Literature Review

The study of public sentiment toward Prabowo Subianto as Indonesia's president and political leader remains academically relevant because his presidency marks a major political transition after the 2024 election. Prabowo-Gibran were officially declared winners by The Indonesian Commission on Elections (ICE) after securing 58.6% of the vote, while post-election legal challenges and debates over political legitimacy, dynastic politics, public trust, and democratic consolidation continued to shape public discussion [9]. Therefore, analyzing public sentiment toward Prabowo is not merely a technical exercise, but also a way to understand how citizens interpret leadership, legitimacy, and policy direction in Indonesia's digital public sphere.

Recent studies show that social media X/Twitter has become an important arena for Indonesian political expression, especially during and after the 2024 presidential election. Research has examined sentiment toward presidential candidates, election results, Prabowo's inauguration, and the early Prabowo-Gibran administration [11], [16], [17]. These studies demonstrate that political sentiment is dynamic and event-driven. Public opinion can change from campaign enthusiasm to post-election evaluation, especially when citizens respond to cabinet formation, leadership image, campaign promises, and policy implementation. This creates a clear academic need for updated sentiment analysis focusing specifically on Prabowo as both president and political leader.

Furthermore, recent research on the first 100 days and policy implementation of the Prabowo-Gibran government indicates that public responses are not limited to electoral preference, but extend to governance performance, communication strategy, and policy acceptance [7], [10]. Such findings suggest a research gap: many studies analyze the election period broadly, but fewer focus on Prabowo's evolving image after assuming national leadership. This is important because sentiment toward a president can influence political trust, policy legitimacy, polarization, and future electoral behavior.

The use of the Naive Bayes algorithm is also academically justified. Although more complex models such as SVM, LSTM, GRU, and transformer-based models are increasingly used, Naive Bayes remains relevant because it is simple, efficient, interpretable, and widely applied in Indonesian-language sentiment classification [4], [13], [16], [13]. Its probabilistic approach is suitable for classifying large volumes of short social media texts into positive, negative, and neutral categories. For political communication research, interpretability is especially valuable because researchers need not only prediction results, but also explainable patterns of public opinion.

Therefore, sentiment analysis toward Prabowo using Naive Bayes is still needed academically for three reasons. First, Prabowo's presidency represents a new political phase in Indonesia that continues to generate public debate. Second, social media provides real-time evidence of citizen reactions to leadership and policy. Third, Naive Bayes offers a practical and interpretable method for mapping public sentiment in Indonesian political discourse. This study can contribute to computational social science, political communication, and Indonesian digital democracy studies.

3. Methodology

This study employs a quantitative approach using sentiment analysis and follows the Knowledge Discovery in Databases (KDD) framework, consisting of data selection, preprocessing, transformation, data mining, and evaluation. The objective is to analyze public sentiment toward Prabowo Subianto as President of Indonesia and political leader based on discussions on X (formerly Twitter).

The data used in this study were obtained from the Twitter (X) social media platform using the Tweet Harvest API. The collected data consist of tweets containing the keyword "Prabowo" within the period from June 2024 to June 2025. The harvested data were stored in separate CSV files for each day, using the file naming format prabowo-(year)-(month)-(day).csv (e.g., *prabowo-2024-12-31.csv*).

The dataset comprises 361 CSV files, with each file representing tweet data for a single day. After merging all files, the dataset contained 42,199 tweet records. The primary feature used in this analysis is the tweet text. These data are utilized to analyze public sentiment toward Prabowo Subianto during the specified period.

The preprocessing stage consisted of text cleaning, case folding, tokenization, normalization using the Colloquial Indonesian Lexicon, stopword removal, and stemming using the Sastrawi library. These procedures were conducted to improve text quality and reduce linguistic variations common in Indonesian social media communication. After preprocessing, sentiment labels were generated automatically using the VADER sentiment analyzer. VADER was selected because previous studies have demonstrated its effectiveness in social media sentiment classification and political discourse analysis.

The Sentipol Dataset consists of 3,400 labeled comment records collected from Facebook posts of the 2014 Indonesian presidential election candidates is used to evaluate the performance of automatic labeling methods. The labels available in the Sentipol Dataset are divided into three main classes: positive, negative, and neutral [2].

The labeled dataset was then transformed into numerical features using the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme. To address potential class imbalance among positive, negative, and neutral sentiments, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training dataset. The dataset was subsequently divided into training data (80%) and testing data (20%).

Sentiment classification was conducted using the Multinomial Naïve Bayes algorithm. This algorithm was selected because it performs efficiently on high-dimensional text data, requires relatively low computational resources, and remains one of the most widely used methods for sentiment classification in Indonesian-language social media research. Several smoothing parameter (alpha) values were tested to identify the optimal model configuration.

One commonly used variant of Naive Bayes is Multinomial

Naive Bayes, a supervised learning method that requires labeled data during the training phase (Manning et al., 2008). The probability that a document d belongs to class c is calculated using the following equation:

$$P(c|d) = P(c)\prod_{k=1}^{n} P(t_k|c)$$

Where P(c) represents the prior probability of class c, t_k denotes a token in the document, and $P(t_k|c)$ represents the probability of the occurrence of token t_k given class c. The classification result is determined by selecting the class with the highest posterior probability according to the Maximum A Posteriori (MAP) principle:

$$c_{map} = \arg\max_{c \in C} \hat{P}(c)\prod_{k=1}^{n} \hat{P}(t_k|c)$$

Model performance was evaluated using 10-fold cross-validation to obtain a robust estimation of classification performance. The final model was assessed using a confusion matrix and four evaluation metrics: accuracy, precision, recall, and F1-score.

A confusion matrix is a table used to evaluate the performance of a classification model by displaying the number of correct and incorrect predictions for each class. In binary classification, a confusion matrix consists of four components: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) (Fawcett, 2006).

Table 1
Confusion matrix

Actual class label	Predicted class label	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Based on the confusion matrix shown in Table 1, the main evaluation metrics obtained are accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correct predictions out of all data instances using the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision measures the accuracy of positive predictions using the following formula:

$$Precision = \frac{TP}{TP + FP}$$

Recall measures the model's ability to identify positive data instances using the following formula:

$$Recall = \frac{TP}{TP + FN}$$

F1-score is the harmonic mean of precision and recall and is used to evaluate the balance between the two, using the following formula:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In addition to classification performance, descriptive sentiment distribution analysis was performed to identify dominant public perceptions toward Prabowo during the study period. The findings are expected to provide insights into public opinion, political communication, and digital democracy in contemporary Indonesia.

4. Experiment Design

This study employs the Multinomial Naive Bayes algorithm to classify sentiment toward Prabowo Subianto. The experiment is designed to evaluate the impact of smoothing parameters (alpha) and class balancing methods on classification performance.

First, the dataset is divided into training and testing sets. Text data are preprocessed through cleaning, tokenization, stopword removal, and feature extraction using TF-IDF. The Multinomial Naive Bayes model is then trained using different alpha values (e.g., 0.1, 0.5, 1.0, and 2.0) to examine the effect of Laplace/Lidstone smoothing on classification accuracy.

To address potential class imbalance, several balancing strategies are tested, such as no balancing (original dataset), random oversampling, and SMOTE (Synthetic Minority Oversampling Technique). Each balancing method is combined with each alpha value, producing multiple experimental scenarios as follows.

Model performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix. The results are then compared to identify the optimal combination of alpha value and balancing method for sentiment classification. This experimental design enables a systematic assessment of how smoothing and class balancing influence the effectiveness of the Naive Bayes classifier.

Table 2
Multiple experimental scenario

No.	Parameter	Multinomial Naive Bayes Model	
		Weighting Method	Data Balancing Method
1.	Alpha = 1	TF-IDF	Without Balancing
2.	Alpha = 0,5	TF-IDF	Without Balancing
3.	Alpha = 0.1	TF-IDF	Without Balancing
4.	Alpha = 0.01	TF-IDF	Without Balancing
5.	Alpha = 1	TF-IDF	SMOTE Oversampling
6.	Alpha = 0,5	TF-IDF	SMOTE Oversampling
7.	Alpha = 0.1	TF-IDF	SMOTE Oversampling
8.	Alpha = 0.01	TF-IDF	SMOTE Oversampling

5. Analysis

A. Preprocessing Result of Tweeted Data

The preprocessing results shown in Table 3 indicate that the tweet texts were successfully cleaned of URLs, symbols, and case differences, leaving only the opinion content. The tokenization process split sentences into individual words,

Table 3
Sample of Preprocessing X Dataset X (Twitter)

Sample of Preprocessing X Dataset X (Twitter)	
Features Name	**Value**
full_text	Bagi pendukung @prabowo monggo kita jabarkan berbagai keberhasilan Jokowi Semangaaat Pendukung GP dan ARB jgn komen tapi bantu RT dan ya Prabowo: Kepemimpinan Jokowi Berhasil Saya Berniat Meneruskan https://t.co/iwj2A3zU4B
text_cleaned	Bagi pendukung monggo kita jabarkan berbagai keberhasilan Jokowi Semangaaat Pendukung GP dan ARB jgn komen tapi bantu dan ya Prabowo Kepemimpinan Jokowi Berhasil Saya Berniat Meneruskan
text_lowercase	bagi pendukung monggo kita jabarkan berbagai keberhasilan jokowi semangaaat pendukung gp dan arb jgn komen tapi bantu dan ya prabowo kepemimpinan jokowi berhasil saya berniat meneruskan
text_tokenized	['bagi', 'pendukung', 'monggo', 'kita', 'jabarkan', 'berbagai', 'keberhasilan', 'jokowi', 'semangaaat', 'pendukung', 'gp', 'dan', 'arb', 'jgn', 'komen', 'tapi', 'bantu', 'dan', 'ya', 'prabowo', 'kepemimpinan', 'jokowi', 'berhasil', 'saya', 'berniat', 'meneruskan']
text_normalisasi	['bagi', 'pendukung', 'monggo', 'kita', 'jabarkan', 'berbagai', 'keberhasilan', 'jokowi', 'semangaaat', 'pendukung', 'gp', 'dan', 'arb', 'jangan', 'komen', 'tapi', 'bantu', 'dan', 'ya', 'prabowo', 'kepemimpinan', 'jokowi', 'berhasil', 'saya', 'berniat', 'meneruskan']
text_stopword_remove	['pendukung', 'monggo', 'jabarkan', 'keberhasilan', 'jokowi', 'semangaaat', 'pendukung', 'gp', 'arb', 'komen', 'bantu', 'ya', 'prabowo', 'kepemimpinan', 'jokowi', 'berhasil', 'berniat', 'meneruskan']
text_stemming	['dukung', 'monggo', 'jabar', 'hasil', 'jokowi', 'semangaaat', 'dukung', 'gp', 'arb', 'komen', 'bantu', 'ya', 'prabowo', 'pimpin', 'jokowi', 'hasil', 'niat', 'terus']
text_translated	Support Monggo West Java Results Jokowi Semangaaat Support GP ARB Assisting Comments Yes Prabowo leads Jokowi as a result of continuous intention

while normalization corrected non-standard words, such as changing *"jgn"* to *"jangan"* ("don't"). The stopword removal stage eliminated common words that have little influence on sentiment, leaving important terms such as *"supporter," "success," "Jokowi,"* and *"Prabowo."* Stemming converted words into their root forms; for example, *"pendukung"* became *"dukung"* ("support") and *"kepemimpinan"* became *"pimpin"* ("lead"). The text was then translated into English to enable processing by the labeling method. This preprocessing pipeline made the data more concise, consistent, and ready for sentiment analysis.

The missing value inspection results shown in Figure 1 indicate that almost all columns contain no missing values, except for the text translated column, which has 145 missing entries. These missing values were generated during the process of translating the text into English.

```
missing_values_df_twitter = df_twitter_preprocessed.isnull().sum()
print("Missing Value Count for Each Column in df_twitter:")
print(missing_values_df_twitter)

Missing Value Count for Each Column in df_twitter:
full_text              0
text_cleaned           0
text_lowercase         0
text_tokenized         0
text_normalisasi       0
text_stopword_remove   0
text_stemming          0
str_normalisasi        0
text_translated      145
dtype: int64
```

Fig. 1. The missing value after preprocessing

To maintain data consistency for the labelling and modelling stages, all rows containing missing values were removed. After the removal process, the dataset size decreased from 42,199 to 42,054 records, as shown in Figure 2.

```
Shape of df_twitter after dropping rows with missing values: (42054, 9)
```

Fig. 2. The number of data after omitting the missing values

The results of the missing value inspection on the Sentipol dataset show that the comment and label columns contain no missing values. However, the text_cleaned and text lowercase preprocessing output columns each contain 34 missing entries. Rows containing these missing values were removed to prevent them from affecting the labeling and evaluation processes.

```
missing_values_df_sentipol = df_sentipol_preprocessed.isnull().sum()
print("Missing Value Count for Each Column in df_sentipol:")
print(missing_values_df_sentipol)

Missing Value Count for Each Column in df_sentipol:
komentar             0
label                0
text_cleaned        34
text_lowercase      34
text_tokenized       0
text_normalisasi     0
str_normalisasi      0
dtype: int64
```

Fig. 3. Missing values after dataset sentipol preprocessing

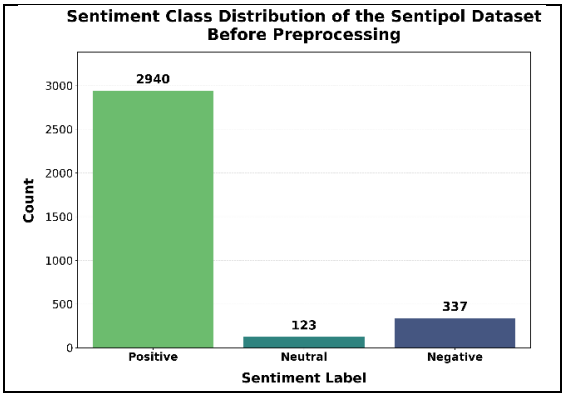

Fig. 4. Sentipol label distribution before preprocessing

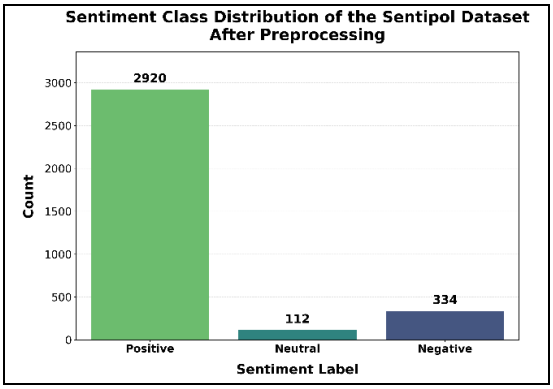

Fig. 5. Sentipol labels distribution after preprocessing

The sentiment class distribution before preprocessing, as shown in Figure 4, consisted of 2,940 positive**,** 337 negative, and 123 neutral records. After preprocessing and the removal of problematic data, as shown in Figure 5, the dataset contained 2,920 positive**,** 334 negative, and 112 neutral records.

The reduction in the number of records was relatively small and did not significantly alter the overall class distribution pattern. However, it improved the quality of the data for the sentiment labeling evaluation stage.

B. Results of VADER and TextBlob Implementation on the Sentipol Dataset

Table 3
Comparison of the original sentiment label distribution with VADER and TextBlob predictions

	Actual Label	VADER Prediction	TextBlob Prediction
Positive	2920	2191	1214
Neutral	112	832	1988
Negative	334	343	164

The sentiment class distribution presented in Table 3 shows that VADER produces a distribution that is closer to the original labels than TextBlob, particularly for the positive and negative classes. VADER correctly predicted 2,191 positive and 343 negative instances, which are relatively close to the original labels of 2,920 positive and 334 negative instances.

In contrast, TextBlob tends to classify more data as neutral, as indicated by the substantially higher number of neutral predictions compared to the original labels. These findings suggest that VADER is more sensitive in distinguishing sentiment polarity, whereas TextBlob is more conservative and exhibits a stronger tendency toward neutral predictions.

C. Comparison of Evaluation Results between VADER Lexicon and TextBlob

The evaluation results presented in Figures 6 through 8 indicate that VADER outperforms TextBlob. VADER achieved an accuracy of 0.6248, which is significantly higher than TextBlob's accuracy of 0.3526. The confusion matrix shows that most positive instances were correctly classified by VADER, although misclassifications still occurred in the negative and neutral classes, with many of these instances being incorrectly assigned to the positive class. VADER's precision, recall, and F1-score ranged from 0.36 to 0.44, indicating a moderate level of performance.

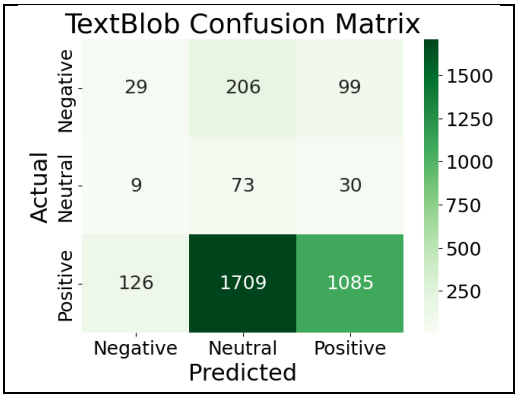

Fig. 6. Confusion matrix results of vader lexicon classification

In contrast, TextBlob exhibited a strong tendency to predict the neutral class. A large number of positive and negative instances were incorrectly classified as neutral, resulting in lower overall accuracy. TextBlob's precision, recall, and F1-score ranged only from 0.23 to 0.37, reflecting less balanced classification performance across sentiment classes. Overall, VADER demonstrated greater consistency and a stronger ability to distinguish sentiment polarity than TextBlob, making it a more suitable choice for the dataset used in this study.

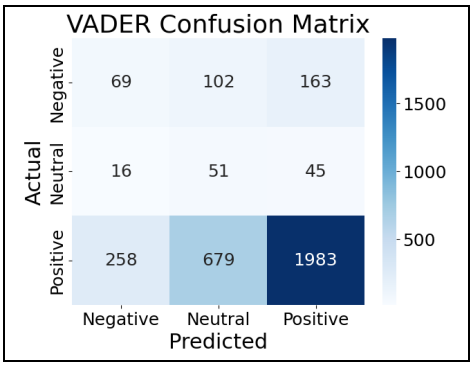

Fig. 7. Confusion Matrix Resulted from textblob Classification

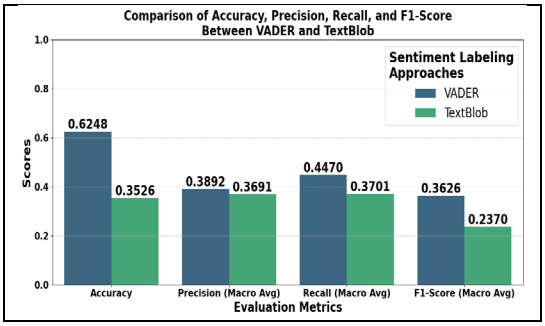

Fig. 8. Comparison of accuracy, Precision, Recall, and F1-Score between VADER and textblob

D. Results of VADER Implementation on the Twitter Dataset

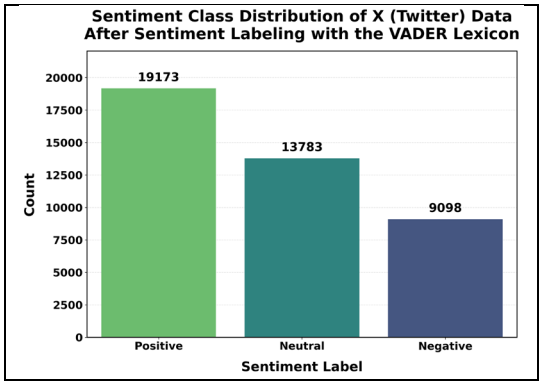

Fig. 9. Sentiment class distribution of the x (twitter) dataset after vader lexicon implementation

The sentiment distribution resulting from VADER-based labeling of the preprocessed X (Twitter) dataset shows that the positive class dominates with 19,173 records, followed by the neutral class with 13,783 records, and the negative class with 9,098 records, as shown in Figure 9. This pattern indicates that user opinions regarding the analyzed topic tend to be more positive than negative. However, the differences in the number of instances across sentiment classes reveal the presence of class imbalance. Therefore, a data balancing step is required to

prevent the classification model from becoming biased toward the majority class.

E. Data Splitting Results

Based on Figure 10, the data splitting process was performed after the stemmed text tokens were recombined into string format and paired with their corresponding sentiment labels.

```
df_twitter_preprocessed['str_stemming'] = \
df_twitter_preprocessed['text_stemming'].apply(lambda x: gabung(x))

X = df_twitter_preprocessed['str_stemming']
y = df_twitter_preprocessed['sentiment_label']

# Splitting the Data into Training and Testing Sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(f"Data shapes:")
print(f"  X_train: {X_train.shape}")
print(f"  X_test: {X_test.shape}")
print(f"  y_train: {y_train.shape}")
print(f"  y_test: {y_test.shape}")
```

```
Data shapes:
  X_train: (33643,)
  X_test: (8411,)
  y_train: (33643,)
  y_test: (8411,)
```

Fig. 10. Data splitting implementation on the x (twitter) dataset hasil pembobotan TF-IDF

The dataset was then divided using an 80% training set and 20% testing set scheme, employing a stratified split technique to ensure that the sentiment class proportions remained balanced across both the training and testing datasets.The splitting process resulted in 33,643 records for training and 8,411 records for testing.

The TF-IDF transformation results indicate that the 33,643 training records and 8,411 testing records were successfully converted into numerical representations containing 3,423 features (terms), as shown in Figure 11. The identical number of features in both the training and testing datasets indicates that the vectorization process was consistent and utilized the same vocabulary across both datasets.

```
tfidf = TfidfVectorizer(min_df=10, max_df=0.9)

X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)

print("X_train_tfidf shape:", X_train_tfidf.shape)
print("X_test_tfidf shape:", X_test_tfidf.shape)
```

```
X_train_tfidf shape: (33643, 3423)
X_test_tfidf shape: (8411, 3423)
```

Fig. 11. TF-IDF weighting implementation

The data balancing process was conducted under two scenarios: a dataset without balancing and a dataset balanced using SMOTE. In the dataset without balancing, the class distribution was imbalanced, consisting of 15,338 positive records, 11,026 neutral records, and 7,279 negative records, as shown in Figure 12.

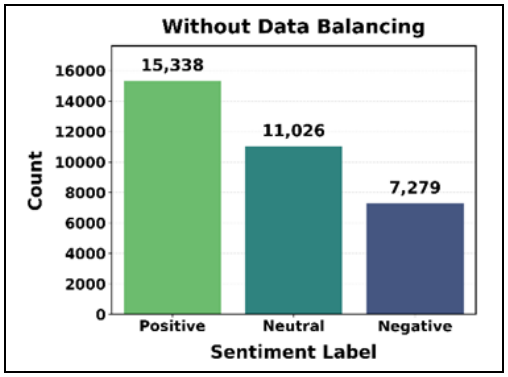

Fig. 12. Class distribution of the training dataset without balancing

This imbalance may cause the model to focus more on the majority class while reducing its ability to accurately recognize and classify instances from the minority classes.

To address this issue, the SMOTE (Synthetic Minority Over-sampling Technique) method was applied by generating synthetic samples for the minority classes. After the SMOTE process, as shown in Figure 15, the number of records in each class became equal, with 15,338 records per class.

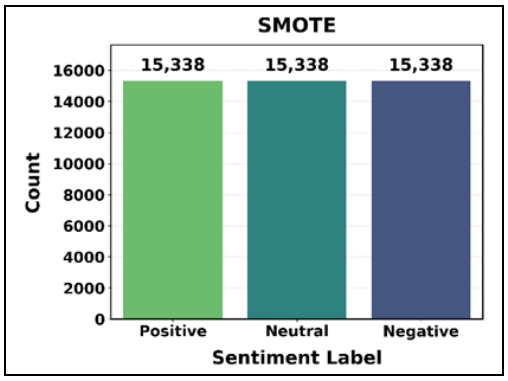

Fig. 13. Class distribution of the training dataset after smote implementation

These results indicate that the class distribution was successfully balanced without reducing the number of instances in the majority class, thereby providing the model with a fairer opportunity to learn the patterns of each sentiment class.

F. Evaluation Results of the Multinomial Naive Bayes Model

The evaluation results presented in Table 4 show variations in model performance across the two data balancing scenarios without balancing and with SMOTE using four different values of the smoothing parameter α (1.00, 0.50, 0.10, and 0.01).

In the without-balancing scenario, the accuracy ranged from 0.6769 to 0.6845. Reducing the alpha value from 1.00 to 0.01 resulted in gradual improvements in accuracy, recall, and F1-score, although the increases were relatively small. Precision tended to be higher than recall, indicating that the model focused more on the correctness of predictions for specific classes than on its ability to identify all instances within each class. This pattern reflects the impact of class imbalance in the dataset. The mean cross-validation (CV) accuracy, ranging from 0.6668 to 0.6737, suggests that the model was relatively stable, although its overall performance remained limited.

Table 4
Evaluation results of the multinomial naive bayes model

No.	Data Balancing Method	Alpha	Accuracy	Precision	Recall	F1-Score	CV Mean Accuracy
1	Without Balancing	1.00	0.6769	0.7227	0.6244	0.6447	0.6668
2	Without Balancing	0.50	0.6818	0.7207	0.6340	0.6537	0.6718
3	Without Balancing	0.10	0.6842	0.7161	0.6397	0.6583	0.6737
4	Without Balancing	0.01	0.6845	0.7147	0.6407	0.6590	0.6731
5	SMOTE	1.00	0.7006	0.6867	0.7035	0.6891	0.7230
6	SMOTE	0.50	0.6966	0.6828	0.6994	0.6849	0.7212
7	SMOTE	0.10	0.6928	0.6789	0.6949	0.6808	0.7193
8	SMOTE	0.01	0.6920	0.6781	0.6938	0.6798	0.7187

After applying SMOTE, the model's performance improved across all evaluation metrics. Accuracy increased, reaching its highest value of 0.7006 at $\alpha = 1.00$. Recall improved significantly to 0.7035, indicating that the model became more capable of recognizing all sentiment classes in a balanced manner. The F1-score also increased to 0.6891, reflecting a better balance between precision and recall. The mean CV accuracy reached 0.7230, demonstrating stronger generalization capability across different validation folds. These improvements highlight the important role of data balancing in addressing class distribution imbalance.

The variation in alpha values on the SMOTE-balanced dataset exhibited a different pattern from the non-balanced scenario. Larger alpha values produced better overall performance, $\alpha = 1.00$ yielded the best results for nearly all evaluation metrics, while decreasing alpha to 0.50, 0.10, and 0.01 led to gradual declines in accuracy, precision, recall, F1-score, and mean CV accuracy. This finding suggests that stronger smoothing helps the model perform more effectively when the dataset has already been balanced.

Overall, the application of SMOTE consistently improved recall and F1-score compared with the model trained without balancing. Although precision decreased slightly, the improvement in recall resulted in more balanced and stable classification performance. Based on all evaluation metrics, the combination of Multinomial Naive Bayes, SMOTE, and $\alpha = 1.00$ demonstrated the most optimal and stable performance for sentiment classification tasks involving imbalanced class distributions

6. Conclusion

This study applied the Multinomial Naive Bayes algorithm to classify positive, neutral, and negative sentiments from 42,199 posts collected from X (formerly Twitter) related to President Prabowo Subianto between June 2023 and June 2024. The dataset underwent text preprocessing, TF-IDF weighting, and evaluation with and without data balancing. Model performance was assessed using accuracy, precision, recall, and F1-score.

Without data balancing, the best result was achieved at alpha = 0.01, with an accuracy of 0.6845, precision of 0.7147, recall of 0.6407, and F1-score of 0.6590. K-Fold Cross Validation produced a mean accuracy of 0.6737, indicating good prediction accuracy but limited ability to recognize all sentiment classes due to class imbalance.

After applying SMOTE, performance improved significantly. The highest results were obtained at alpha = 1.00, with an accuracy of 0.7006, precision of 0.6867, recall of 0.7035, and F1-score of 0.6891. K-Fold Cross Validation yielded a mean accuracy of 0.7230. These findings demonstrate that Multinomial Naive Bayes can effectively classify sentiment data, while SMOTE enhances model balance, stability, and overall performance.

References

[1] A. Gaizka, A. R. Dzikrillah, and E. Sinduningrum, "Analisis sentimen masyarakat sebelum dan sesudah terpilihnya Gibran sebagai cawapres Prabowo menggunakan Naïve Bayes," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 6, no. 6, pp. 2830–2841, Jun. 2024.

[2] A. Rachmat and Y. Lukito, "SENTIPOL: Dataset sentimen komentar pada kampanye PEMILU Presiden Indonesia 2014," in *Konferensi Nasional Teknologi Informasi dan Komunikasi*, Dec. 2016, pp. 218–228.

[3] A. R. Sembiring and C. K. Dewa, "Sentiment analysis on Indonesian tweets about the 2024 election," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 1, pp. 413–422, Jan. 2025.

[4] A. S. Muliana, D. Lestarini, and S. P. Raflesia, "Analysis of public sentiment on election results using Naïve Bayes," *SISTEMASI: Jurnal Sistem Informasi*, vol. 13, no. 6, pp. 2467–2478, Oct. 2024.

[5] D. Faroek, M. Yusuf, and G. Syatauw, "Sentiment analysis of the popularity of parties supporting the 2024 presidential candidates on Twitter using the Naive Bayes classifier algorithm," *Antivirus: Jurnal Ilmiah Informatika*, vol. 17, no. 2, pp. 216–227, Nov. 2023.

[6] D. S. Nugroho, I. F. Hanif, M. A. Hasbi, *et al.*, "Sentiment analysis of alleged 2024 election fraud based on tweets using the Naïve Bayes classifier," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 1169–1176, Jul. 2024.

[7] I. Najiyah and M. Rizal, "Analisis sentimen media sosial X terhadap kebijakan Presiden Prabowo Subianto," *Jurnal FASILKOM (Teknologi Informasi dan Ilmu Komputer)*, vol. 15, no. 3, pp. 443–455, Dec. 2025.

[8] M. M. Ismail and K. M. Lhaksmana, "Sentimen analisis pada media online mengenai pemilihan presiden 2019 dengan menggunakan metode Naive Bayes," *eProceedings of Engineering*, vol. 6, no. 2, pp. 9997–10004, Aug. 2019.

[9] N. Karmini and F. Syam, "Indonesia declares Prabowo Subianto president-elect after court rejects rivals' appeal," *Associated Press*, Apr. 24, 2024.

[10] R. Afifah and Sugiyono, "Social media X-based public opinion analysis of Prabowo-Gibran government's first 100 days using Naïve Bayes and K-nearest neighbor classification methods," *Journal Innovations Computer Science*, vol. 4, no. 2, pp. 213–222, Nov. 2025.

[11] R. Cahyanti, "Twitter sentiment analysis towards candidates of the 2024 Indonesian presidential election," *RESTI (Rekayasa dan Teknologi Informasi)*, vol. 8, no. 4, pp. 516–524, Aug. 2024.

[12] R. Z. P. Dwitama, Y. Yusra, M. Fikry, F. Yanto, and E. Budianita, "Klasifikasi sentimen masyarakat terhadap Prabowo Subianto bakal calon presiden 2024 di Twitter menggunakan Naïve Bayes classifier," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 5, no. 2, pp. 366–375, Dec. 2023.

[13] S. Prayudani, D. R. B. Situmorang, R. Hidayah, and H. S. Ginting, "Sentiment analysis of social media X in the 2024 Indonesian presidential election using the Naive Bayes algorithm: Candidates' backgrounds and political promises," *Journal of Applied Informatics and Computing*, vol. 8, no. 2, pp. 291–295, Dec. 2024.

[14] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[15] W. Septian and F. M. Sarimole, "Komparasi analisis sentimen masyarakat terhadap isu penundaan pemilu 2024 pada Twitter dengan metode Naive Bayes dan support vector machine," *Jurnal Sains dan Teknologi*, vol. 5, no. 3, pp. 890–899, Feb. 2024.

[16] Y. Gunawan, I. Purnama, and R. Rohani**,** "Sentiment analysis of Twitter towards the 2024 Indonesian presidential candidates using the Naïve Bayes algorithm," *International Journal of Science Technology & Management*, vol. 5, no. 4, pp. 953–961, Jul. 2024.

[17] Y. Mayasari and Y. R. Nasution**,** "Post-election sentiment analysis 2024 via Twitter/X using Naïve Bayes classifier algorithm," *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, vol. 4, no. 2, pp. 123–134, Aug. 2024.