Comparison of Naïve Bayes, Random Forest, and SVM Algorithm Performance in Analyzing Sentiment Regarding the Aceh Floods on Platform X

Amalia Khoirunnisa [1]*, Novita Kurnia Ningrum [2]**
* Teknik Informatika, Politeknik Negeri Batam
** Teknik Multimedia Jaringan, Politeknik Negeri Batam
111202012962@mhs.dinus.ac.id [1], mail2@polibatam.ac.id [2]

Article Info

Article history:

Received 2026-01-13
Revised 2026-04-08
Accepted 2026-04-18

Keyword:

Aceh Floods,
Naïve Bayes,
Random Fores,
SMOTE,
Support Vector Machine.

ABSTRACT

Social media has become a means for the public to express their opinions on various events, including the floods in Aceh. This study aims to analyze public sentiment and compare the performance of the Naïve Bayes, Random Forest, and Support Vector Machine (SVM) algorithms in classifying sentiment on the X (Twitter) platform. In addition, this study also evaluates the effect of applying the Synthetic Minority Over-sampling Technique (SMOTE) on improving model performance.The dataset was collected using a crawling method utilizing Twitter Harvest on the X (Twitter) platform during the period from November 18, 2025, to January 5, 2026. The data collection process yielded 1,971 Indonesian-language data points, which after preprocessing stages such as text cleaning, stemming, and duplicate removal resulted in 1,874 data points. The dataset was then divided into 80% training data (1,499 data points) and 20% test data (375 data points). The analysis results show that the majority of public opinion has a positive sentiment of 77.9% (1460 data), while negative sentiment is 22.1% (414 data). The model evaluation results show that the application of SMOTE can improve the performance of the three algorithms. The algorithm with the best performance is Support Vector Machine (SVM) with an accuracy value of 83% and an F1-score of 75% after the application of SMOTE. Based on the results of the study, the SMOTE technique has been proven to help improve the model's ability to recognize minority classes, resulting in better classification performance.

I. INTRODUCTION

With the advancement of technology, human interaction is no longer limited to face-to-face encounters, but also through social media. Social media allows for discussions on current issues without the constraints of distance and time. One of the most widely used social media platforms is X (Twitter). This platform was chosen as a data source because its users are active in expressing their opinions, responses, and reactions to various issues that are developing in Society[1]. The issue chosen for this study was the Aceh floods, with comments (*Tweets*) collected automatically using the *Crawling* method. The comment collection period was from November 18, 2025, to January 5, 2026, and the results of applying this method yielded 1971 data points. Sentiment analysis is a process of identifying, extracting, and grouping public

opinions or views into specific categories, such as positive and negative sentiments, based on comments (text) generated by users[1]. In its application, *machine learning* is used to perform classification[2] by utilizing various algorithms such as *Naïve Bayes*, *Random Forest*, and SVM. Each algorithm has different characteristics and performance. *Naïve Bayes* is a relatively simple algorithm that performs well in probability-based classification[3], , although it is sensitive to assumptions of independence between features. *Random Forest* is an ensemble-based algorithm that can improve accuracy by combining multiple decision trees, but it has higher computational complexity. Meanwhile, *Support Vector Machine* (SVM) is known to be effective in handling high-dimensional data such as text data, and is capable of producing good classification performance with optimal separation margins.

Previous research conducted by Ikhsani et al. (2024) compared SVM with *Naïve Bayes* using two online loan datasets, often abbreviated as pinjol (legal and illegal). The results of this study showed that SVM outperformed *Naïve Bayes* on the legal pinjol dataset with an accuracy of 70%, while *Naïve Bayes* outperformed SVM on the illegal pinjol dataset with an accuracy of 75%[4]. Another study by Salmon et al. (2024) compared the performance of the KNN and Random Forest algorithms using data on kidney failure, with the *Random Forest* algorithm performing better with an accuracy of 99.75%, while the KNN algorithm had the highest accuracy of 93.75% [5]. Then, research by Rani et al. (2025) compared *Naïve Bayes* and SVM against the sentiment of PP NO.82 of 2021 on Twitter, with SVM superior with an accuracy of 100% while *Naïve Bayes* had an accuracy of 95%[6]. Another study by Sulhan et al. (2025) compared *Naïve Bayes* with SVM on the sentiment of the Ferizy ferry ticket application, with SVM achieving an accuracy of 82.62% and *Naïve Bayes* achieving an accuracy of 79.27% [7]. As well as research by Virgiawan et al. (2025) compared the *Naïve Bayes* and *Random Forest* algorithms on a stroke disease dataset for classification. The results of this study showed that *Random Forest* was superior in accuracy and *recall* with a value of 92% compared to *Naïve Bayes*, and had the same *f1- score* value of 88% [8]. Another study by Dicky et al. (2024) compared the Support Vector Machine (SVM) and Naïve Bayes algorithms in relation to public sentiment about the Megathrust Earthquake in Indonesia. The results showed that the Support Vector Machine (SVM) algorithm performed better with an accuracy rate of 84%, while the Naïve Bayes algorithm achieved an accuracy rate of 71%. This study also applied the Synthetic Minority Over-sampling Technique (SMOTE) method to address data imbalance. The application of this method had a positive impact on the performance of both algorithms. After applying SMOTE, the accuracy of the SVM algorithm increased to 85%, while the accuracy of Naïve Bayes increased to 74%[9].

Based on previous studies, the *Naïve Bayes, Random Forest*, and *Support Vector Machine* (SVM) algorithms have been widely used in text classification processes, particularly in sentiment analysis. However, there are still relatively few studies that specifically compare the performance of these three algorithms in the same study, especially when considering the issue of data imbalance. In addition, the application of *the Synthetic Minority Over-sampling Technique* (SMOTE) as a method for balancing class distribution has not been comprehensively studied in the context of disaster sentiment analysis.

Therefore, this study aims to compare the performance of the *Naïve Bayes, Random Forest*, and SVM algorithms in analyzing public sentiment towards the floods in Aceh based on Twitter data. This study also evaluates the effect of applying SMOTE on improving the performance of each algorithm. With this study, it is hoped that accurate and relevant results can be obtained, thereby contributing to determining a more effective algorithm for identifying public sentiment towards flood disasters.

II. Methods

The research stage consists of a series of processes carried out from the beginning to the end of the research. This stage aims to ensure that the research is conducted in a structured manner, so that it is more effective and efficient in achieving the predetermined objectives. The research stages can be seen in Figure 1

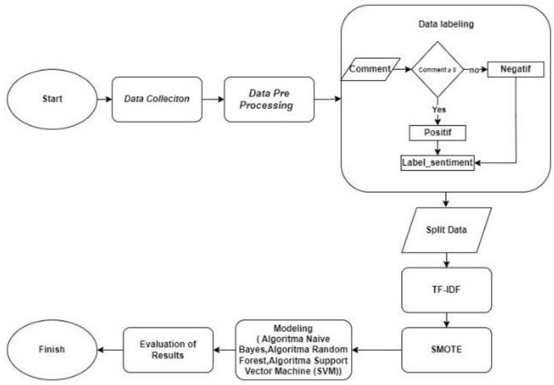

Figure 1. Research Stages

A. *Data Colleciton*

The data collection process is the initial stage of research where data is taken from X comments in the form of qualitative data, which is relevant to user sentiment regarding the Aceh floods. Using the data *crawling* method run using *Google Colab* by utilizing *the Twitter Harvest library* and *Node.js* [6][10][11][12]. For time efficiency, this method was chosen over manual data input. The data collection process was carried out to automatically collect relevant comments and save them as a dataset in .csv format. Data collection was limited to comments containing the keyword "banjir aceh" with the language used is Indonesian. The data collection period was from November 18, 2025, to January 5, 2026. Based on this process, 1,971 comment data were obtained. The resulting dataset is still raw data, so further processing or preprocessing is needed to improve data quality and prepare it for use in the next stage of analysis. The crawling process produces a dataset consisting of several attributes, namely "conversation_id_str", "created_at", "full_text", "id_str", 'tweet_url', and "user_id_str". A more detailed explanation of each attribute in the dataset can be found in Table I.

Table I shows the contents of the dataset with six attributes, namely conversation_id, created_at_str, full_text, id_str, tweet_url, and user_id_str. All six have the same number, which is 1971. For conversation_id_str, id_str, and , user_id_str is of the int64 data type, while created_at, full_text, and tweet_url are of the object data type.

TABLE I
DESCRIPTION OF DATASET

Atribut	Jumlah	Tipe Data
conversation_id_str	1971	int64
created_at	1971	object
full_text	1971	object
id_str	1971	int64
tweet_url	1971	object
user_id_str	1971	int64

B. Data Pre Processing

At this stage, the dataset will be processed before entering the next process so that the data is ready and of good quality so that it does not interfere with the data training process. The processes carried out at this stage are :

1) *Dataset cleaning*: The dataset cleaning process involves checking for missing values and duplicate data, removing links, emojis, symbols, and numbers. If abnormal data is found , appropriate action can be taken to ensure that the sentences are meaningful [13] .

2) *Case Folding*: This stage involves converting all words to lowercase letters to ensure consistency and avoid bias, such as words that use capital letter[14].

3) *Text normalization*: The text normalization process converts abbreviations, *slang*, and non-standard words into standard words using a standard dictionary and *slang* dictionary used in the research process.

4) *Tokenizing* : This stage involves separating sentences into individual words (tokens) to facilitate the analysis of each word element.

5) *Stopword removal*: This stage involves eliminating words that are not particularly significant in the analysis, such as "in," "or," and similar words.

6) *Stemming*: Stemming is the process of changing each word into its root word by removing conjunctions or affixes. For example, "hunian" becomes "huni" or another word, "terdampak," is changed to "dampak." This process simplifies variations of words that actually originate from the same root word, thereby improving the model generalization process.

C. Data Labeling

At this stage, comments are grouped into two categories: *positive* and *negative*. The method used is a *lexicon-based* process carried out by matching words based on the InSet (Indonesia Sentiment *Lexicon*) dictionary so that each word has a score[14][15][16] . This score is then used as a reference for labeling based on formula 1.

$$Komentar = \begin{cases} Negatif, & x < 0 \\ Positif, & x \geq 0 \end{cases} \quad (1)$$

Based on formula 1, comments are categorized as Positive, if the score is greater than or equal to zero, while comments with a score less than zero are grouped as negative

TABLE II
LABELING RESULTS

setelah_stemming_string	score	label_sentimen
respons perintah aceh pas banjir bikin masyarakat percaya	12	positif
aksi dinkes sisir area isolir aceh pascabanjir komitmen perintah pusat adil sosial layan sehat rata tinggal salut tim juang medan berat	20	positif
dasar lapor resmi dinkes aceh kemenkes respon banjir aceh cepat tim medis kerah desember buka posko sehat pidie jaya lhokseumawe dll obat ribu korban perintah potong birokrasi akses cepat	16	positif
warga korban bencana banjir aceh tamiang benarbenar kecewa bulan banjir hantam perintah bantu	-4	negatif

Table II presents several examples of labeling results, which are divided into two groups: negative and positive. The categorization of the score is determined by the value itself. Specifically, a score that is greater than or equal to zero is categorized as positive, while a score that is less than 0 is categorized as negative.

D. Data Split

The next stage is Data Split, where the dataset will be separated into two parts, namely training data and test data, using an 80:20 ratio. Based on this ratio, 80% of the data will be used as training data and the remaining 20% will be used as test data. This process is necessary to see the model's ability to identify new data.

E. TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is a method that calculates the importance of a word by determining its frequency and comparing it to the frequency of all words in a text. The findings of TF-IDF are elucidated in Table III.

TABLE III
TF-IDF RESULTS

Term	TF-IDF
aceh	0.063641
banjir	0.055312
pemerintah	0.038234
bantu	0.033926

F. SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE is a *machine learning* method commonly used to address class imbalance. It is suitable for this study because the amount of *negative* class data is smaller than that of the *positive* class. This method can increase the amount of minority data by generating new data based on the proximiy of the original data, there by improving the model's ability to recognize minority classes.

G. Naïve Bayes

Naïve Bayes is a probability-based classification method based on Bayes' theorem, which assumes that each feature is independent [15][16]. The advantage of using this method is that the prediction results are quite good even though it uses simple assumptions, and the process is relatively short and efficient. However, with the assumption of independence, the results are less realistic, because in the real world, each feature (word) must be connected or continuous in order to form a complete sentence.

H. Random Forest

Random Forest algorithm is an extension of *Classification* and *Decision Tree* (CART), combining several *decision trees* to form a forest, thereby improving prediction accuracy[5]. Each *Decision Tree* is trained using a different subset of data, with the final result based on a majority voting process [17].

I. Support Vector Machine(SVM)

Support Vector Machine (SVM) is a machine learning algorithm that works by constructing a *hyperplane* as a separator between classes in feature space [18][19]. The prediction process is carried out by determining the position of a data point relative to the hyperplane, so that the data will be labeled according to the class to which it belongs. In the classification process using the Support Vector Machine (SVM) algorithm, the kernel used is LinearSVC, because this method is considered suitable for handling high-dimensional data, such as text data generated from the TF-IDF weighting process. The use of a linear kernel is also effective in processing text features that have a large number of dimensions. The advantage of the SVM algorithm lies in its ability to handle high-dimensional data and determine the optimal separation margin between classes. However, this algorithm has a weakness, namely that it requires relatively longer computation time and its performance can decline if the data contains a high level of noise.

J. Evaluation of resultsl

The evaluation process is the final stage in this study, which aims to assess the performance of the classification model that has been built. The evaluation was carried out using *a confusion matrix* to determine the number of correct and incorrect predictions in each class, so that it could be used as a basis for comparing the performance of the *Naïve Bayes, Random Forest,* and *Support Vector Machine* algorithms. *The confusion matrix* can be seen in table IV.

TABLE IV
CONFUSION MATRIX

Predicted Values ↓ / Actual Values →	**Positif**	**Negatif**
Positif	True Positive(TP)	False Positive(FP)
Negatif	False Negative(FN)	True Negative(TN)

Table IV is *a confusion matrix* containing information on the number of correct and incorrect predictions in each class, which can be used as a reference in calculating *evaluation metrics* such as *accuracy, precision, recall,* and *F1-score*. The following are the general formulas for evaluation matrices used in research to measure the performance of classification algorithms.

1. *Accuracy*

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

The *accuracy* value is obtained based on formula 2, the value is based on how much data is predicted from the total prediction.

2. *Precision*

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

In formula 3, the *precision* value is obtained from the number of true positives compared to the total number *of true positives* and *false positives*.

3. *Recall*

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

In formula 4, the *recall* value is obtained from the number of *true positives* divided by the total number *of true positives* and *false negatives.*

4. *F1-Score*

$$f1 - Score = \frac{2\ X\ Precison\ X\ Recall}{Precision+Recall} \quad (5)$$

In formula 5, the *F1-score* value is calculated by multiplying precision by *recall* twice, then comparing it to the total sum of *precision* and *recall.*

5. *Macro Average*

$$Macro\ Averag = \frac{1}{N}\sum_{i=1}^{N} Matric_i \quad (5)$$

The macro average is calculated by averaging the metric values in each class. In this study, *the macro average F1-score* was obtained by summing *the F1-scores* of the *positive* and *negative* classes, then dividing by the number of classes.

III. RESULTS AND DISCUSSION

Data Collection

Data collection is the initial stage of this research. The data used consists of user comments on platform X (Twitter).Data collection was limited to comments containing the keyword “banjir aceh” with the language used is Indonesian. The data collection period was from November 18, 2025, to January 5, 2026. The manual data collection process would take a long time, especially for large amounts of data. Therefore, in this study, data was collected automatically using *crawling* techniques to save time. The data *crawling* process utilizes *Node.js-based tweet harvesting*, which is run using *Python* on the *Google Colaboratory* platform to collect comments according to keywords. Using this technique, 1,971 data points were successfully collected, which were then stored in .CSV format to be used as a dataset. The dataset obtained is not the final data and requires further processing to maximize model performance.

A. Data Pre Processing

This stage is a continuation of the data collection process, where dirty datasets and many unnecessary attributes will be separated into a dataset *called df_clean*, which contains comments (*full_text*). This action is necessary to remove irrelevant data that could potentially interfere with the classification process, so that the dataset used is of better quality. In data pre-processing, actions such as dataset cleaning, *case folding*, text normalization, *tokenizing*, *stopword removal*, and *stemming* can be seen in Table V.

TABLE V
DATA PRE PROCESSING RESULTS

Proses	Hasil
full_text	@enoyazoyus72535 Setuju kesehatan memang prioritas utama dalam penanganan bencana seperti banjir dan longsor di Aceh. BNPB melaporkan mobilisasi medis sedang gencar dengan ribuan terdampak mendapat bantuan. Semangat untuk tim medis dan penyintas!
Setelah_Pembersihan_dataset	Setuju kesehatan memang prioritas utama dalam penanganan bencana seperti banjir dan longsor di Aceh BNPB melaporkan mobilisasi medis sedang gencar dengan ribuan terdampak mendapat bantuan Semangat untuk tim medis dan penyintas
Setelah_Case_Folding	setuju kesehatan memang prioritas utama dalam penanganan bencana seperti banjir dan longsor di aceh bnpb melaporkan mobilisasi medis sedang gencar dengan ribuan terdampak mendapat bantuan semangat untuk tim medis dan penyintas
Setelah_Normalisasi_teks	setuju kesehatan memang prioritas utama dalam penanganan bencana seperti banjir dan longsor di aceh bnpb melaporkan mobilisasi medis sedang gencar dengan ribuan terdampak mendapat bantuan semangat untuk tim medis dan penyintas
Setelah_Tokenizing	setuju,kesehatan,memang,prioritas,utama,dalam,penanganan,bencana,seperti,banjir,dan,longsor,di,aceh,bnpb,melaporkan,mobilisasi,medis,sedang,gencar,dengan,ribuan,terdampak,mendapat,bantuan,semangat,untuk,tim,medis,dan,penyintas
Setelah_Stopword_removal	setuju,kesehatan,prioritas,utama,penanganan,bencana,banjir,longsor,aceh,bnpb,melaporkan,mobilisasi,medis,gencar,ribuan,terdampak,bantuan,semangat,tim,medis,penyintas
Setelah_Stemming	tuju,sehat,prioritas,utama,tangan,bencana,banjir,longsor,aceh,bnpb,lapor,mobilisasi,medis,gencar,ribu,dampak,bantu,semangat,tim,medis,sintas
Setelah_Stemming_string	tuju sehat prioritas utama tangan bencana banjir longsor aceh bnpb lapor mobilisasi medis gencar ribu dampak bantu semangat tim medis sintas

Table V shows the results of the dataset cleaning process, which removed "@enoyazoyus72535" and punctuation marks. During this process, duplicate data was also found. The initial *df_clean* dataset consisted of 1971 data points. After identifying and removing duplicate data, the number of data points decreased to 1,874, meaning that 97 duplicate data points were deleted. The process continued with *case folding*, where capital letters were changed to lowercase letters, such as "Setuju" to "setuju". Then, *Tokenizing* changed sentences into words (*tokens*), such as "setuju kesehatan memang" to "setuju,kesehatan,memang". The next stage was *stopword removal*, where common words such as "memang", "dalam", "di", and "dan" were deleted. Then the *Stemming* process changes each word into its root form, such as "setuju" and "kesehatan" are changed to "tuju" and "sehat". The final stage is to change the *Stemming* results back into a *string*.

B. Data Labelling

Data is labeled based on formula 1. Data will be categorized *as positive* if the score is greater than or equal to zero, while data will be labeled *as negative* if the score is less than zero. The labeling results can be seen in Figure 2.

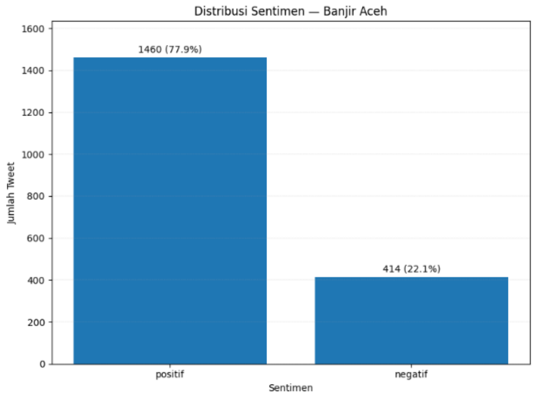

Figure 2. Labeling Data Results

Figure 2 shows that public sentiment towards the Aceh floods on platform X is dominated by positive sentiment at 77.9% or 1460 data points, while the remaining 22.1% or 414 data points are classified as negative sentiment.

C. Split Data

The dataset is divided into two parts: training data and test data, with a ratio of 80:20. A total of 80% (1499 data) is used as training data and 20% (375 data) as test data. The attribute "setelah_stemming_string" is used as a feature or input data, while "label_sentimen" is used as an object.

D. TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF (*Term Frequency–Inverse Document Frequency*) is a word weighting method used to convert text data into numerical representations so that it can be processed by *machine learning* algorithms. This method assigns a weight to each word based on its frequency of occurrence in a document and the importance of that word to the document as a whole.

E. SMOTE (Synthetic Minority Over-sampling Technique)

In this study, the Aceh flood sentiment analysis dataset consists of only two classes, *positive* and *negative*, with an unbalanced data distribution. This can reduce the performance of the algorithm during the classification process, especially in terms of the minority *f1-score*. By using this method, the *negative* class will obtain new data so that the data is equal or balanced with the *positive* class. The results of the Synthetic Minority Over-sampling Technique (SMOTE) can be seen in Figure 4.

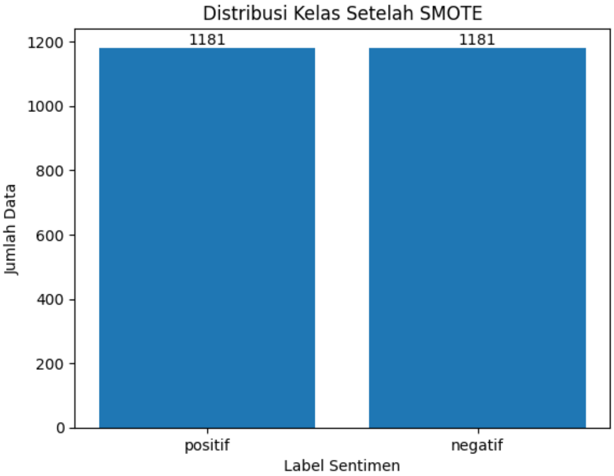

Figure 4 shows the SMOTE results on the training data label, which consists of 1499 data points, where the positive data contains 1181 data points while the negative data only contains 318 data points. By applying the SMOTE method, the data becomes balanced, with both positive and negative labels containing 1181 data points.

F. Evaluasi Model dan Hasil

Pada ketiga model algoritma, yaitu Naïve Bayes, Random Forest, dan Support Vector Machine (SVM), dilatih menggunakan dua kondisi data yang berbeda, yaitu data dengan distribusi label yang tidak seimbang serta data yang telah diseimbangkan menggunakan teknik Synthetic Minority Over-sampling Technique (SMOTE).

G. Naïve Bayes

In the Naïve Bayes model, classification is built using the Multinomial Naïve Bayes algorithm implemented through the Scikit-learn library in Python. The Multinomial Naïve Bayes algorithm is suitable for use with text data such as TF-IDF results. For the first test, the model was trained using an unbalanced label distribution. The results of the model testing process are presented in the form of a classification report shown in Figure 5.

=== Naive Bayes Sebelum SMOTE ===

	precision	recall	f1-score	support
negatif	1.00	0.05	0.10	96
positif	0.75	1.00	0.86	279
accuracy			0.76	375
macro avg	0.88	0.53	0.48	375
weighted avg	0.82	0.76	0.67	375

Gambar 5. Naïve Bayes Evaluation Before SMOTE

Based on Figure 5, the model performance evaluation results show an accuracy value of 76%, precision of 88%, recall of 53%, and an F1-score of 48%, which illustrates the model's performance in classifying sentiment data. The relatively high precision value indicates that the model is quite good at identifying data predicted to be of a certain class. However, the lower recall value indicates that there is still some data that should be included in that class but was not detected by the model. This is also reflected in the F1-score value, which is not very high, indicating that the balance between precision and recall still needs to be improved.

Next, the second test was conducted using the Naïve Bayes model on data that had been balanced by applying the Synthetic Minority Over-sampling Technique (SMOTE). At this stage, hyperparameter tuning was also carried out using the Grid Search method with 5-fold cross-validation to obtain the best parameters. The parameter optimized in the Multinomial Naïve Bayes model was the alpha value with several variations, namely 0.1, 0.5, and 1.0. Based on the results of the best parameter search process, an alpha value of 0.1 was obtained as the optimal parameter that provided the best performance in the model. Evaluation of results in the form of a classification report shown in Figure 6.

Naive Bayes Sesudah SMOTE

	precision	recall	f1-score	support
negatif	0.56	0.55	0.56	96
positif	0.85	0.85	0.85	279
accuracy			0.78	375
macro avg	0.71	0.70	0.70	375
weighted avg	0.77	0.78	0.78	375

Figure 6. Naïve Bayes Evaluation After SMOTE

Figure 6 shows the results of the model performance evaluation with an accuracy value of 78%, precision of 71%, recall of 70%, and F1-score of 70%. These results indicate that the model's performance is more balanced in detecting each sentiment class. The relatively increased recall and F1-score values indicate that the model is able to better identify data in minority classes after applying the SMOTE method.

H. Random Forest

The next model used in this study is Random Forest. In this model, two classification testing scenarios were conducted using different training data. The first scenario used data with an unbalanced label distribution, while the second scenario used data that had been balanced through the application of the Synthetic Minority Over-sampling Technique (SMOTE) method. For the first scenario, the model was built using the Random Forest algorithm with 100 decision trees (n_estimators) and a random_state parameter of 42 to maintain consistency in the results during the model training and testing process. The results of the model performance evaluation are presented in the form of a classification report shown in Figure 7

=== Random Forest Sebelum SMOTE ===

	precision	recall	f1-score	support
negatif	0.88	0.15	0.25	96
positif	0.77	0.99	0.87	279
accuracy			0.78	375
macro avg	0.82	0.57	0.56	375
weighted avg	0.80	0.78	0.71	375

Figure 7. Random Forest Evaluation Before SMOTE

Based on Figure 7, the model performance evaluation results show an accuracy value of 78%, precision of 82%, recall of 57%, and an F1-score of 56%. The relatively high precision value indicates that the model is quite good at predicting data that belongs to a certain class. However, the lower recall value indicates that there is still some data that the model fails to identify. This is also reflected in the F1-score value, which is still moderate, indicating that the balance between precision and recall in the model still needs to be improved. Furthermore, in the second scenario, the Random Forest model was trained using labeled data that had been balanced through the application of the Synthetic Minority Over-sampling Technique (SMOTE) method. At this stage, hyperparameter tuning was also carried out using the Grid Search method with 5-fold cross-validation to obtain the best parameter combination. The parameters optimized in the Random Forest model included the number of decision trees (n_estimators), the maximum tree depth (max_depth), and the minimum number of samples for splitting (min_samples_split). Based on the results of the best parameter search process, the optimal parameter combination obtained was n_estimators of 500, max_depth of None, and min_samples_split of 2. The results of the model performance evaluation are presented in the form of a classification report shown in Figure 8.

Random Forest Sesudah SMOTE

	precision	recall	f1-score	support
negatif	0.71	0.30	0.42	96
positif	0.80	0.96	0.87	279
accuracy			0.79	375
macro avg	0.75	0.63	0.65	375
weighted avg	0.78	0.79	0.76	375

Figure 8. Random Forest Evaluation After SMOTE

Figure 8 shows the results of the model performance evaluation with an accuracy value of 79%, precision of 75%, recall of 63%, and F1-score of 65%. These results indicate that the model has a fairly good ability to classify sentiment data after applying the Synthetic Minority Over-sampling Technique (SMOTE) method and hyperparameter tuning process. The increased recall and F1-score values indicate that the model is able to identify data in the minority class better than in unbalanced data conditions.

I. Support Vector Machine (SVM)

The next and final model in this study is Support Vector Machine (SVM). In this model, two classification testing scenarios were conducted using different training data. The first scenario used data with an unbalanced label distribution (before SMOTE), while the second scenario used data that had been balanced through the application of the Synthetic Minority Over-sampling Technique (SMOTE) method. In the first scenario, the model was trained using data with an unbalanced label distribution. The model was implemented using the Linear Support Vector Classification (LinearSVC) algorithm with a random_state parameter of 42 to maintain consistency of results during the model training and testing process. The results of the model performance evaluation are presented in the form of a classification report shown in Figure 9.

=== SVM Sebelum SMOTE ===

	precision	recall	f1-score	support
negatif	0.75	0.42	0.54	96
positif	0.83	0.95	0.89	279
accuracy			0.82	375
macro avg	0.79	0.69	0.71	375
weighted avg	0.81	0.82	0.80	375

Figure 9. Evaluation of Support Vector Machine (SVM) before SMOTE

Figure 9 shows the results of the model performance evaluation with an accuracy value of 82%, precision of 79%, recall of 69%, and F1-score of 71%. These results indicate that the model has a fairly good ability to classify sentiment data. The relatively balanced precision and recall values show that the model is not only able to predict classes fairly accurately, but also able to identify most of the data. Furthermore, in the second scenario, the Support Vector Machine (SVM) model was trained using data that had been balanced through the application of the Synthetic Minority Over-sampling Technique (SMOTE) method. At this stage, hyperparameter tuning was also carried out using the Grid Search method with 5-fold cross-validation to obtain the best parameter combination. The parameters optimized in the Linear Support Vector Classification (LinearSVC) model included the regularization parameter (C) and the maximum number of iterations (max_iter). Based on the results of the best parameter search process, the optimal parameter combination was obtained, namely a C value of 1 and a max_iter of 1000. The results of the model performance evaluation are presented in the form of a classification report shown in Figure 10.

SVM Sesudah SMOTE

	precision	recall	f1-score	support
negatif	0.71	0.55	0.62	96
positif	0.86	0.92	0.89	279
accuracy			0.83	375
macro avg	0.78	0.74	0.75	375
weighted avg	0.82	0.83	0.82	375

Figure 10. Evaluation of Support Vector Machine (SVM) after SMOTE

Figure 10 shows the results of the model performance evaluation with an accuracy value of 83%, precision of 78%, recall of 74%, and F1-score of 75%. These results indicate an increase in the recall and F1-score values, which indicates that the model has become better at identifying data in the minority class after data balancing.

Based on the results of testing conducted on the three algorithms, namely Naïve Bayes, Random Forest, and Support Vector Machine (SVM), both on unbalanced and balanced data using the SMOTE technique, differences in performance were obtained for each model. A comparison of the evaluation results of the three algorithms is summarized in Table V.

TABLE VI
COMPARISON OF THREE ALGORITHM

Condition SMOTE	**Algoritma**	**Akurasi**	**Presisi**	**recall**	**F1-Score**
Before SMOTE	Naïve Bayes	76%	88%	53%	48%
	Random Forest	78%	82%	57%	56%
	SVM	82%	79%	69%	71%
After SMOTE	Naïve Bayes	78%	71%	70%	70%
	Random Forest	79%	75%	63%	65%
	SVM	83%	78%	74%	75%

Based on the comparison table of model evaluation results, it can be seen that the application of the SMOTE method and the hyperparameter tuning process provides improvements in several evaluation metrics, particularly in recall and F1-score values. This shows that data balancing using SMOTE can help the model identify data in minority classes better. In the Naïve Bayes model, there was a significant increase in recall and F1-score values after applying SMOTE. The recall value increased from 53% to 70%, while the F1-score increased from 48% to 70%. This shows that the SMOTE method can help the model detect more data in the minority class, resulting in more balanced classification performance. In the Random Forest model, the application of SMOTE also improved performance, although not significantly. The recall value increased from 57% to 63%, and the F1-score value increased from 56% to 65%. This shows that data balancing helps the model recognize patterns in minority classes better than before the oversampling process. Meanwhile, the Support Vector Machine (SVM) model showed the best performance compared to other models. This can be seen from the highest accuracy value of 83% and an F1-score of 75% after applying SMOTE and hyperparameter tuning. In addition, the recall value also increased from 69% to 74%, indicating that the model was able to better identify data in each class.

In this study, word visualization using the word cloud method was used to display the distribution of words that appeared in the dataset. The visualization results were divided into two main categories, namely positive sentiment and negative sentiment. Through this visualization, it can be seen which words most dominantly appeared in each sentiment category in the dataset related to the flooding in Aceh. The word cloud visualization results for positive sentiment can be seen in Figure 11.

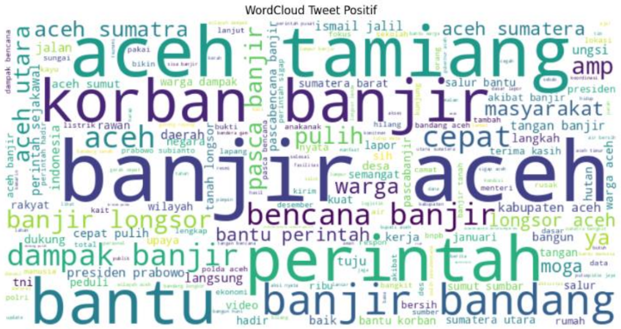

Figure 11. Positive Word Cloud

In Figure 11 (positive sentiment), some of the most frequently appearing words include “aceh” (1640), “banjir” (1404), “bantu” (469), “perintah” (450), “pulih” (300), “bencana” (279), and “korban” (269). The appearance of these words shows that positive sentiment discussions are mostly related to support, assistance, and post-disaster recovery efforts. Words such as “bantu”, “pulih”, “cepat”, “bangun”, “salur”, and “hadir” indicate the public's appreciation or expectations regarding disaster management efforts and assistance to victims. In addition, words such as “presiden”, “prabowo”, “tni”, and “pemerintah” indicate that positive sentiment is also related to the public's response to the role of the government and related parties in dealing with the impact of disasters.

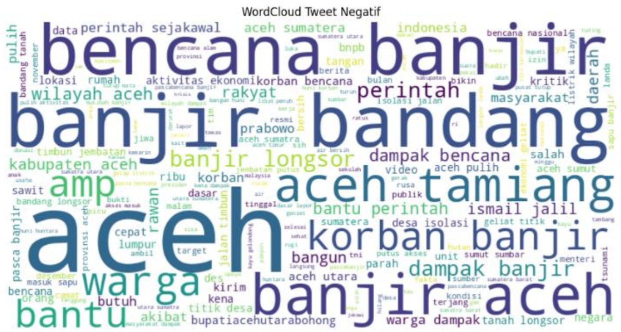

Figure 12. Negative Word Cloud

In Figure 12 (negative sentiment), the most frequently appearing words include “aceh” (471), “banjir” (380), “bencana” (218), “dampak” (109), “bandang” (109), “korban” (107), dan “perintah” (98). The appearance of these words shows that negative sentiment is more related to the adverse effects caused by disasters, such as damage to infrastructure, community losses, and the conditions of the affected areas. Words such as “rawan”, “listrik”, “ekonomi”, “rumah”, dan “jembatan” describe the complaints or concerns of the community regarding post-disaster conditions that affect their daily lives.

When comparing the two sentiments, there are several words that appear in both categories, such as "aceh", "banjir", "bencana", "korban", "wilayah", dan "masyarakat". This shows that both sentiments discuss the same main topic, namely the flood disaster, but from different perspectives. Positive sentiment highlights the support, assistance, and recovery efforts undertaken after the disaster, while negative sentiment highlights the damage and various problems caused by the disaster.

Overall, the results of the word cloud analysis show that public conversations in this research dataset are dominated by discussions about flooding in Aceh, including the impacts and the responses of the government and community in dealing with these conditions. Based on the sentiment classification results in the dataset used, it was found that positive sentiment accounted for 77.9% or 1,460 data points, while negative sentiment accounted for 22.1% or 414 data points. This shows that most public opinion in the research dataset tended to respond positively to the floods in Aceh, especially in the context of support, assistance, and post-disaster management efforts.

IV. CONCLUSION

The research specifically analyzes and compares the Naïve Bayes, Random Forest, and Support Vector Machine (SVM) algorithms in relation to public opinion based on the topic of flooding in Aceh on the X (Twitter) platform. It also evaluates the effect of SMOTE implementation on improving the performance of the three algorithms. The dataset was collected automatically using the Crawling method by utilizing Twitter Harvest on the X (Twitter) platform. The dataset contains comments related to the topic of flooding in Aceh, with a collection period ranging from November 18, 2025, to January 5, 2026. Comments were taken specifically in Indonesian, and the results of applying this method yielded 1971 data points. After data cleaning, stemming, and duplicate removal, the data was reduced to 1874. The clean dataset was then split into 80% (1499 data) as training data and 20% (375 data) as test data applied to the three algorithms.

The results of sentiment distribution analysis show that the majority of public opinion in the dataset has a positive sentiment of 77.9% or 1,460 data points, while negative sentiment accounts for 22.1% or 414 data points. This indicates that most public conversations related to flooding in Aceh tend to be positive, especially regarding support, assistance, and post-disaster management and recovery efforts.

Based on the results of the evaluation, the application of the SMOTE technique was proven to increase the accuracy and F1-score of the three algorithms. In the Naïve Bayes algorithm, the accuracy value increased from 76% to 78%, while the F1-score increased from 48% to 70%. In the Random Forest algorithm, the accuracy value increased from 78% to 79%, and the F1-score increased from 56% to 65%. Meanwhile, the algorithm that showed the best performance among the three algorithms tested was Support Vector Machine (SVM). After applying the SMOTE technique, the accuracy value in the SVM algorithm increased from 82% to 83%, while the F1-score increased from 71% to 75%. Based on these test results, all three algorithms showed an increase in F1-score after applying the SMOTE technique. This indicates that the SMOTE technique helps the model recognize patterns in minority classes better, resulting in more balanced classification performance. Further research is recommended to increase the amount of data and expand the time range of data collection in order to improve the model's ability to generalize and represent public sentiment more broadly. It is also recommended to expand the data source platforms, such as YouTube or TikTok, so that the sentiment analysis results can provide a more comprehensive and diverse picture of public opinion. Furthermore, the development of classifications with more than two sentiment classes, such as positive, negative, and neutral, is expected to produce a more comprehensive analysis of public opinion. With this development, future research is expected to produce a more accurate and relevant sentiment analysis model.

REFERENCES

[1] T. Safitri, Y. Umaidah, and I. Maulana, "Analisis Sentimen Pengguna Twitter Terhadap BTS Menggunakan Algoritma Support Vector Machine," vol. 7, no. 1, pp. 34–41, 2023.

[2] S. M. Putri, R. Novita, and M. Afdal, "Perbandingan Algoritma Linear Regression , Support Vector Regression , dan Artificial Neural Network untuk Prediksi Data Obat," vol. 6, no. 1, pp. 54–63, 2024, doi: 10.47065/bits.v6i1.5184.

[3] B. Z. Ramadhan, I. Riza, and I. Maulana, "Analisis Sentimen Ulasan Pada Aplikasi E-Commerce Dengan Menggunakan Algoritma Naïve Bayes," vol. 6, no. 2, pp. 220–225, 2022.

[4] Y. Ikhsani, I. Permana, F. N. Salisah, and N. E. Rozanda, "Perbandingan Algoritma Support Vector Machine dan Naïve Bayes dalam Menganalisis Sentimen Pinjaman Online di Twitter," vol. 6, no. 3, 2024, doi: 10.47065/bits.v6i3.6106.

[5] H. Ekawati, "Perbandingan Kinerja Algoritma K-Nearest Neighbor dan Algoritma Random Forest Untuk Klasifikasi Data Mining Pada Penyakit Gagal Ginjal," vol. 6, no. 3, pp. 1943–1953, 2024, doi: 10.47065/bits.v6i3.6476.

[6] A. M. Rani and N. Hendrastuty, "Perbandingan Algoritma NBC Dan SVM Untuk Melakukan Analisis Sentimen Terhadap PP NO . 82 Tahun 2021," vol. 6, no. 4, pp. 2139–2151, 2025, doi: 10.47065/bits.v6i4.6496.

[7] M. Sulhan, "Perbandingan Metode Naïve Bayes Dengan SVM Pada Analisis Sentimen Aplikasi Pemesanan Tiket Kapal Ferizy," vol. 6, no. 4, pp. 0–9, 2025, doi: 10.47065/bits.v6i4.6715.

[8] I. Virgiawan, "Analisis Perbandingan Algoritma Naïve Bayes dan Random Forest Dalam Klasifikasi Penyakit Stroke Pada Puskesmas," vol. 6, no. 4, pp. 2807–2814, 2025, doi: 10.47065/bits.v6i4.6771.

[9] D. Kurniawan, M. Najib, and D. Satria, "Analisis Sentimen Opini Publik Tentang Gempa Megathrust di Indonesia Menggunakan Metode Support Vector Machine dan Naïve Bayes," vol. 6, no. 3, 2024, doi: 10.47065/bits.v6i3.6213.

[10] R. S. Andarujaya and R. R. Suryono, "Perbandingan Kinerja Algoritma Random Forest , KNN , dan SVM dalam Analisis Sentimen Cryptocurrency," vol. 6, no. 4, pp. 2288–2299, 2025, doi: 10.47065/bits.v6i4.6572.

[11] I. T. Rahmawati and D. Alita, "Perbandingan Algoritma SVM ,

Random Forest , KNN untuk Analisis Sentimen Terhadap Overclaim Skincare pada Media Sosial X," vol. 6, no. 4, pp. 2390–2402, 2025, doi: 10.47065/bits.v6i4.6782.

[12] L. Kencono and D. Darwis, "Perbandingan Algoritma NBC , SVM dan Random Forest untuk Analisis Sentimen Implementasi Starlink pada Media Sosial X," vol. 6, no. 4, pp. 2288–2300, 2025, doi: 10.47065/bits.v6i4.6813.

[13] I. Arfyanti, T. Bustomi, and I. Haristyawan, "Perbandingan Kinerja Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Darah Tinggi," vol. 6, no. 3, pp. 1987–1994, 2024, doi: 10.47065/bits.v6i3.6477.

[14] M. R. Raihandika and R. R. Suryono, "Perbandingan Algoritma Random Forest , KNN , SVM Untuk Analisis," vol. 6, no. 4, pp. 2403–2412, 2025, doi: 10.47065/bits.v6i4.6797.

[15] R. T. Aldisa and P. Maulana, "Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi Booster COVID-19 Dengan Perbandingan Metode Naive Bayes, Decision Tree dan SVM," *Build. Informatics, Technol. Sci.*, vol. 4, no. 1, pp. 106–109, 2022, doi: 10.47065/bits.v4i1.1581.

[16] R. Windari and H. W. Nugroho, "Prediksi Rekomendasi Pemilihan Kejuruan pada Sekolah Menengah Kejuruan Menggunakan Perbandingan Metode Decision Tree C4 . 5 dan Naïve Bayes," vol. 6, no. 4, pp. 2700–2708, 2025, doi: 10.47065/bits.v6i4.6928.

[17] E. Danuarta and D. Alita, "Perbandingan Algoritma Naïve Bayes dan Random Forest untuk Melakukan Analisis Sentimen Cyberbullying Generasi Z Pada Twitter," vol. 6, no. 4, pp. 2448–2458, 2025, doi: 10.47065/bits.v6i4.6909.

[18] Y. Wiratama and R. Z. A. Aziz, "Perbandingan Prediksi Penyakit Stunting Balita Menggunakan Algoritma Support Vektor Machine dan Random Forest," vol. 6, no. 2, 2024, doi: 10.47065/bits.v6i2.5543.

[19] E. A. Pranata, F. Budiman, and D. Kurniawan, "Analisis Sentimen Ulasan Mobile JKN pada Playstore dengan Perbandingan Akurasi Algoritma Naïve Bayes dan SVM," vol. 7, no. 1, 2025, doi: 10.47065/bits.v7i1.7334.