Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 2 (2025) 195 - 208 e-ISSN: 2580-0760

An In-depth Exploration of Sentiment Analysis on Hasanuddin Airport using Machine Learning Approaches

Lilis Nur Hayati[1], Fitrah Yusti Randana [2]*, Herdianti Darwis[3]
[1]Department of Information System, Faculty of Computer Science, Universitas Muslim Indonesia, Makassar, Indonesia
[2, 3]Department of Informatics, Faculty of Computer Science, Universitas Muslim Indonesia, Makassar, Indonesia
[1]lilis.nurhayati@umi.ac.id, [2]13020210286@umi.ac.id, [3] herdianti.darwis@umi.ac.id

Abstract

Machine learning-based sentiment analysis has become essential for understanding public perceptions of public services, including air transportation. Sultan Hasanuddin Airport, one of the main gateways in eastern Indonesia, faces the challenge of improving services amid changing user needs due to the COVID-19 pandemic. This study aims to compare the effectiveness of three machine learning algorithms- Support Vector Machine (SVM), Naive Bayes Multinomial, and K-Nearest Neighbor (KNN)-in analyzing the sentiment of user reviews related to airport services. The research also explores data splitting techniques, text preprocessing, data balancing using SMOTE, model validation, and method parameterization to ensure optimal results. The review data was retrieved from Google Maps (2021-2024) and underwent manual labelling. Text preprocessing includes normalization, stemming using Sastrawi, and stopword removal. The data-balancing technique uses SMOTE, while model evaluation is done with stratified k-fold cross-validation. SVM with a linear kernel showed the best performance, achieving an F1-score of 98.4%. Naive Bayes performed optimally, achieving an F1-score of 93.9%, while KNN recorded the best F1-score of 92.0%. SMOTE was shown to improve Naive Bayes' performance on unbalanced datasets, although it did not significantly impact SVM. The findings of this study provide data-driven recommendations to improve services at Sultan Hasanuddin Airport, such as the management of cleaning facilities, waiting room comfort, and passenger flow efficiency. In addition, this research opens up opportunities for developing real-time sentiment analysis systems that can be applied in other air transportation sectors.

Keywords: Sentiment Analysis; Support Vector Machine; Naive Bayes; K-Nearest Neighbor; SMOTE; Sultan Hasanuddin Airport.

How to Cite: Fitrah Yusti Randana, Lilis Nur Hayati, and H. Darwis, "An In-depth Exploration of Sentiment Analysis on Hasanuddin Airport using Machine Learning Approaches", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 2, pp. 195 - 208, Mar. 2025.
DOI: https://doi.org/10.29207/resti.v9i2.6253

1. Introduction

Indonesia, with more than 17,000 islands stretching from Sabang to Merauke, makes air transportation the primary means to support the mobility of people and tourists [1]. The speed and efficiency of air transportation are important pillars in driving national economic growth, trade, and tourism. Sultan Hasanuddin Airport, as one of the main gateways in eastern Indonesia, has a strategic role in connecting these regions [2]. The quality of services and facilities at the airport is a crucial factor that affects the experience of service users, which is reflected in customer reviews on cleanliness, comfort, efficiency of passenger flow, and staff attitude [3], [4]. With the COVID-19 pandemic, airport management faces significant challenges, including travel restrictions and health protocol adjustments that affect operations and user perception [5], [6]. Under these conditions, sentiment analysis becomes an essential tool for understanding changes in user perceptions of airport services and facilities, and it provides insights for managers to improve service quality.

Many studies have been related to sentiment analysis in the air transportation sector, and various techniques and algorithms have been applied. Previous research shows that an SVM has an advantage in terms of accuracy compared to a KNN in analyzing the sentiment of traveler reviews, as proven in several studies [7]. In addition, other studies have shown that applying the SMOTE technique to imbalanced data can improve

Received: 23-12-2024 | Accepted: 16-02-2025 | Published Online: 08-03-2025

model performance, especially in algorithms such as Random Forest [8]. However, most previous studies have focused on large international airports or airlines. At the same time, the application of algorithms in the context of Sultan Hasanuddin Airport, Airport-especially after the impact of the COVID-19 pandemic, is still limited. Most studies also only compare two machine learning algorithms, limiting our understanding of the potential of different algorithms in more complex contexts.

There are still several aspects that have not been explored in previous research. Therefore, this study seeks to fill the gap by comparing the effectiveness of three machine learning algorithms, namely SVM, Naive Bayes Multinomial, and KNN, in analyzing the sentiment of user reviews related to services and facilities at Sultan Hasanuddin Airport, considering the impact of the COVID-19 pandemic. This research also explores essential aspects of sentiment analysis, including data splitting, model validation techniques, text preprocessing, data balancing with SMOTE, and method parameterization. These three algorithms are evaluated to find out which is most effective in handling review sentiments that are affected by various factors, such as facility cleanliness, new health protocols, and changes in airport services during the pandemic.

The novelty of this research lies in its more thorough and coordinated approach. By comparing the three algorithms in a broader context, this research can provide deeper insights into the performance of each algorithm on sentiment data faced by Sultan Hasanuddin Airport. In addition, this research also introduces the use of SMOTE to address class imbalance in the dataset, which has not been widely applied in-depth in similar studies. In addition, this research considers the direct impact of the COVID-19 pandemic on user sentiment and tests various preprocessing and model evaluation techniques that can improve the quality of sentiment analysis. This research aims to provide data-driven recommendations that the Sultan Hasanuddin Airport management can use to enhance the quality of service and improve areas of significant concern to users, such as cleanliness, waiting room comfort, and passenger flow efficiency. Thus, this research contributes to the development of machine learning-based sentiment analysis methods and opens up opportunities to develop similar systems at other airports in Indonesia, which can improve user experience more broadly and sustainably.

2. Research Methods

This research is designed to analyze the sentiment of user reviews on services and facilities at Sultan Hasanuddin Airport using a machine learning approach. Several main stages are interrelated, starting with data collection, manual labeling, data splitting, text preprocessing, feature extraction, data balancing, modeling, and prediction on testing data. Each stage is systematically designed to ensure the analysis results are accurate, relevant, and applicable in an authentic context. The entire research flow is organized based on the framework illustrated in Figure 1.

2.1 Data Collection

Data collection is the initial stage in this research, which aims to collect user reviews from the Google Maps platform regarding their experience using services and facilities at Sultan Hasanuddin Airport. The review data includes various categories, such as high-rated, low-rated, recent, and most relevant reviews. The collection was done through two methods: manual and automated. Manually, reviews were collected by copying directly from Google Maps based on sorting such as 'Most Relevant,' 'Latest,' 'Highest Rating,' and 'Lowest Rating' to ensure diversity of user perspectives. The automated method uses the Instant Data Scraper tool, which speeds up extracting data with similar criteria, resulting in a larger and more varied data set [9].

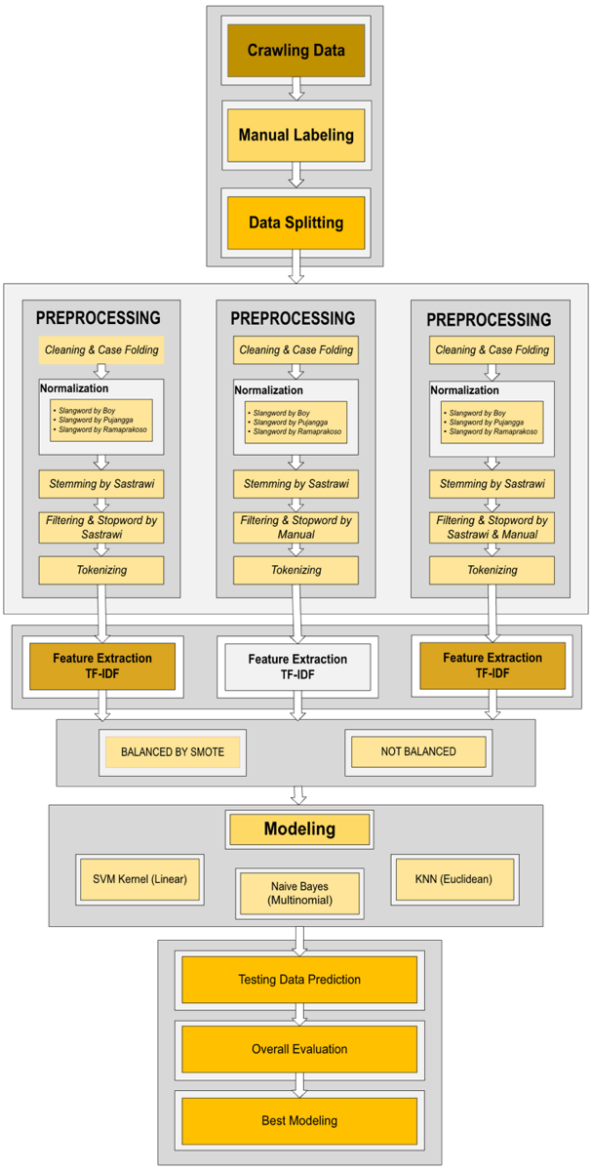


Figure 1. Research Design

The data collected covered 2021-2023 (1,596 raw data) and 2024 (1,829 raw data), bringing the total raw data obtained to 3,525 reviews. A selection was made to use only reviews from the 2021-2024 period to maintain relevance to the current context, especially in reflecting significant changes during the COVID-19 pandemic, such as implementing health protocols and technological adaptations at airports. Before preprocessing, the dataset was refined to 2,804 reviews, consisting of 1,420 from 2021-2023 and 1,384 from 2024, after the removal of irrelevant and duplicate entries.

2.2 Data Labelling

Labeling is done to determine the sentiment of each review, whether positive or negative, with a manual process to ensure the accuracy and consistency of results. This manual approach was chosen because Indonesian contexts often contain slang or informal expressions that are difficult to handle automatically.

Positive sentiment is assigned to reviews that contain appreciation of the service or facility, while negative sentiment is tagged to reviews that reflect criticism or user dissatisfaction. With this approach, the labeling process can capture the nuances of language more accurately, ensuring each review is categorized according to its context.

2.3. Data Splitting and Cross-Validation

After the data labeling stage is complete, we enter the preprocessing stage, but before preprocessing, the dataset is divided into several parts for training, validation, and model testing purposes, as shown in Figure 2.

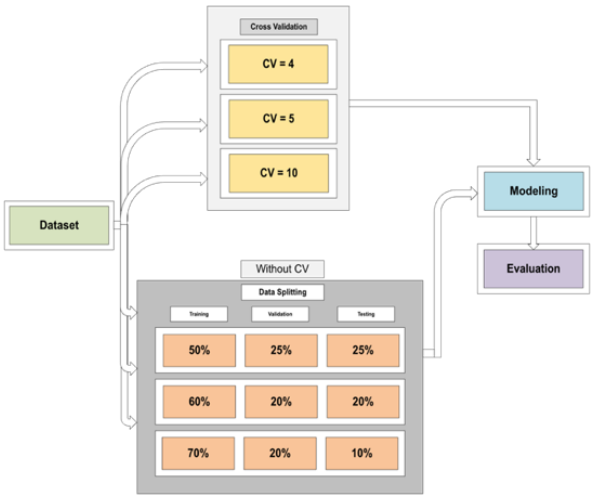


Figure 2. Data Splitting Scenarios

Data splitting is done using two main approaches. Data splitting divides the dataset into three parts: training, validation, and testing, with varying proportions of 50:25:25, 60:20:20, and 70:20:10, to evaluate the effect of training data size on model performance. In addition, stratified K-Fold cross-validation was used with k values of 4, 5, and 10. This technique ensures that each fold has a balanced class distribution, thus improving the reliability of model evaluation and providing more stable results. This approach helps to reduce the risk of bias and ensures that model performance is thoroughly tested on all available data.

Cross-validation is a machine learning model performance evaluation technique that divides a dataset into several subsets or 'folds' to ensure a more accurate and reliable assessment [10]. In this research, stratified K-fold cross-validation is used with 4, 5, and 10 values to keep the class distribution in each fold proportional to the original dataset, as depicted in Figure 3. This technique minimizes data-sharing bias and provides a more representative picture of model performance.

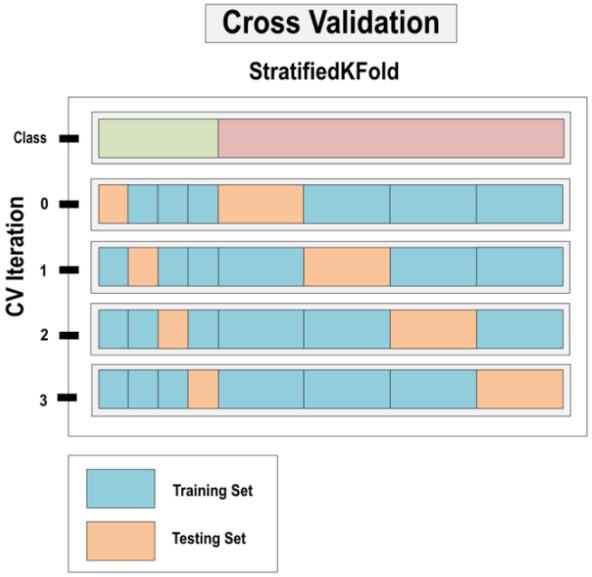


Figure 3. Stratified KFold

The dataset is divided into mutually exclusive subsets, where in each iteration, one of the subsets is used as testing data, while the other subset is used for training. This process is repeated so that each subset is used as testing data once and training data once, with the final result calculated from the average performance of all iterations. The use of 4, 5, and 10 folds provides different levels of evaluation granularity:

4-Fold: The dataset is divided into four subsets, with 25% of the data used for testing and 75% for training. 5-Fold: The dataset is divided into five subsets, with 20% of the data used for testing and 80% used for training.

10-Fold: The dataset is divided into ten subsets, with 10% of the data used for testing and 90% used for training, providing more stable evaluation results due to larger iterations.

The main advantage of this method is that all data is used for training and testing alternately, ensuring a thorough and representative evaluation, especially with

stratification that maintains class distribution in each fold.

2.4 Preprocessing Text

The preprocessing stage aims to clean the text data to make it more structured and ready to be analyzed by machine learning algorithms [11], [12]. Before the cleaning stage, a dominant word calculation, which is the most frequently occurring and rarely occurring words, was performed to avoid deleting important, frequently used words. From the results of this calculation, a slang dictionary named Slangword by Boy was created based on the collected data. Two additional dictionaries, Slangword by Pujangga and Slangword by Ramaprakoso, obtained from GitHub, were also used. These dictionaries were used in the normalization stage to replace nonstandard words with standard words. The preprocessing process includes several stages.

Table 1. Preprocessing

Preprocessing	Before	After
Cleaning	*Luas, Lantai Bersih, Toilet bersih, Banyak jual makanan cmn hargax lumayan mahal dr harga luar... parkiran luas bgt👍👍👍*	*luas lantai bersih toilet bersih banyak jual makanan cmn hargax lumayan mahal dr harga luar parkiran luas bgt*
Normalization	*luas lantai bersih toilet bersih banyak jual makanan cmn hargax lumayan mahal dr harga luar parkiran luas bgt*	*luas lantai bersih toilet bersih banyak jual makanan hanya harganya lumayan mahal dari harga luar parkiran luas banget*
Stemming	*luas lantai bersih toilet bersih banyak jual makanan hanya harganya lumayan mahal dari harga luar parkiran luas banget*	*luas lantai bersih toilet bersih banyak jual makan hanya harga lumayan mahal dari harga luar parkir luas banget*
Filtering & Stopword	*luas lantai bersih toilet bersih banyak jual makan hanya harga lumayan mahal dari harga luar parkir luas banget*	*luas lantai bersih toilet bersih banyak jual makan harga lumayan mahal harga luar parkir luas banget*
Tokenization	*luas lantai bersih toilet bersih banyak jual makan harga lumayan mahal harga luar parkir luas banget*	*luas, lantai, bersih, toilet, bersih, banyak, jual, makan, harga, lumayan, mahal, harga, luar, parkir, luas, banget*

Cleansing and case folding are done by removing irrelevant characters, such as symbols, numbers, and emojis, and converting all text to lowercase to reduce word variation. Normalisation uses three slang dictionaries to align nonstandard words with standard words. Stemming was performed using the Sastrawi library to return words to their base form. Dominant word count was performed before the filtering and stopword removal stages to ensure that important words were not deleted. Instead, unimportant words are collected in a homemade stopword dictionary. This stopword dictionary includes short words (1-3 letters), personal pronouns, conjunctions, prepositions, auxiliaries, adverbs, and interrogatives. Stopword filtering and removal is done through three approaches: sastrawi, which only uses the library without involving the manual dictionary; manual, which uses the homemade stopword dictionary to remove irrelevant words; and a combination of sastrawi and manual, which combines the two for more optimal results. At this stage, words such as *"tidak*", *"baik* ", and "*sangat"* are retained because they play an important role in determining the intensity or direction of sentiment, for example, as negation markers (*not good*) or emotion amplifiers (*very good*). Finally, tokenizing breaks the text into individual tokens using the unigram technique, where each word is treated as a single unit of analysis. This preprocessing is shown in Table 1.

2.5 Feature Extraction

This stage uses the Term Frequency-Inverse Document Frequency (TF-IDF) technique to convert text data into a numerical representation that machine learning algorithms can use. This technique measures the importance of a word in a particular document compared to the entire dataset, thus helping the model recognize sentiment patterns based on significant words. TF-IDF combines two main components: TF, which calculates how often a word appears in a document, and IDF, which measures how unique the word is in the entire dataset [13]. The TF-IDF formula is formulated as shown in Formulas 1, 2, and 3.

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \quad (1)$$

$$\text{TF}(t,d) = \frac{f(t,d)}{\sum_{t' \in d} f(t',d)} \quad (2)$$

$$\text{IDF}(t) = \log\left(\frac{N}{1+|\{d \in D: t \in d\}|}\right) \quad (3)$$

TF-IDF combines the Term Frequency (TF) and the Inverse Document Frequency (IDF) to represent the importance of a word t in a document dd relative to a collection of documents D. This measures how often a word t appears in a document d, normalized by the total number of words in d. where $f(t,d)$ is the raw frequency of word t in document d. This measures the rarity of a word tt across a collection of documents D. where N is the total number of documents in D, and $|\{d \in D: t \in d\}|$ represents the number of documents containing the word t. The product of $\text{TF}(t,d)$ and $\text{IDF}(t)$gives the TF-IDF score, which highlights the importance of a word in a document while penalizing commonly occurring words across documents. The processed text data is converted to numerical form using TF-IDF values for each word, enabling the algorithm to understand important words based on their relative frequency of occurrence in the context of the dataset [14]. This approach is effective in sentiment pattern analysis, as it emphasizes words that are unique and relevant to a particular sentiment rather than commonly used words.

2.6 Balancing Data

Two main approaches are taken to address the class imbalance in the dataset. One technique is the Synthetic Minority Oversampling Technique (SMOTE), which synthetically adds data to the minority class (in this case, positive or negative sentiment, depending on the imbalance) by creating new samples based on combinations of existing data [15]. This technique aims to improve the representation of the minority class so that the class distribution becomes more balanced, which can help machine learning models understand the patterns of both classes more accurately [16]. In addition, experiments were conducted by comparing the model's performance on datasets that have been balanced using SMOTE with datasets without balancing to evaluate the extent to which data balancing affects model performance [17]. This approach ensures that the final result shows high accuracy due to the balanced data distribution and measures the model's ability to handle datasets that reflect real situations with uneven class distribution. This provides a more comprehensive insight into the effectiveness of balancing in improving model generalization.

2.7 Modeling and Classification

Parameter Settings optimize model performance by setting specific values for each algorithm while exploring the effect of various parameter configurations on the analysis results, which is shown in Table 2. In SVM, a linear kernel is used with the C parameter controlling regularization, where small values (0.0001) provide a wider margin, while large values (10000) prioritize more accurate classification [18]. In Naive Bayes, the α parameter is used for Laplace smoothing, preventing zero probabilities on infrequent words, with small values providing minimal impact and large values providing more aggressive smoothing. For KNN, the Euclidean distance metric is used with *n* neighbors (1-31), which determines the number of nearest neighbors, where small values are more sensitive to local data and large values create more stable predictions. Exploration was conducted by trying different combinations of parameter values for each algorithm to understand their effect on model performance and obtain the best configuration that yields optimal accuracy and generalization.

Table 2. Parameter Settings

Algorithm	Function	Parameter	Value
SVM	Linear	C	0.0001; 0.001; 0.01; 0.1; 1; 10; 100; 1000; 10000
Naïve Bayes	Multinomial	Alpha	0.0001; 0.001; 0.01; 0.1; 0.5; 1.0; 5.0; 10.0
KNN	Euclidean	N_neighbors	1, 2, 3, ..., 31

Support Vector Machine (SVM) is a machine learning algorithm for classification and regression that works by finding the best hyperplane that separates data from two or more classes in a high-dimensional space, as shown in Figure 4 [19], [20]. For data that cannot be linearly separated, SVM uses a kernel function to map the data to a higher dimension [21], [22]. This research uses a linear kernel, which is effective for text data as patterns are usually linearly separable. The main parameter optimized is the regularization value, which controls the balance between area margin and classification error, with a value range of 0.0001 to 10000. The SVM model is formulated as described in Formulas 4, 5, and 6.

$$\mathrm{w}^{\mathrm{T}} \cdot x + b = 0 \quad (4)$$

w is the weight vector, x is the feature vector, b is the bias.

The maximum margin is calculated by minimizing the loss function (Formula 5).

$$\min \frac{1}{2}|w|^2 \quad (5)$$

With conditions as in Formula 6.

$$y_i(\mathrm{w}^{\mathrm{T}} \cdot x_i + b) \geq 1, \quad \forall i \quad (6)$$

y_i is the class label (+1 or -1), x_i is the i-th data.

This formula describes optimizing the maximum margin while ensuring all data is correctly classified according to class.

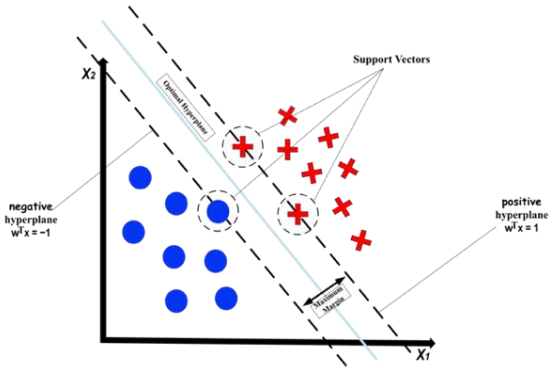


Figure 4. SVC Algorithm

Naive Bayes (Multinomial) is a probabilistic classification algorithm based on Bayes' theorem, which assumes that each feature (word in a review) is mutually independent [23], [24]. The SVM model is formulated as shown in Formula 7.

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (7)$$

$P(C|X)$ is the probability that the riviews X belongs to class (C), $P(X|C)$ represent the probability of observing feature X given class (C), $P(C)$ is the prior probability of class (C), and $P(X)$ refers to the probability of observing the entire dataset.

Multinomial distribution is used to handle text data, as the algorithm considers the frequency of occurrence of words in the document [25]. The α smoothing parameter is optimized from 0.0001 to 10.0 to determine the best value that results in optimal performance.

KNN is an instance-based (lazy learning) algorithm that determines the class of a data point based on its proximity to other labeled data points, as shown in Figure 5 [26], [27]. The Euclidean distance is used as the main metric and is computed using the formulated as shown in Formula 8 where x_i and x_j are the data points, and (x_i, k) and x_j, k are the features of the respective data points [28].

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \quad (8)$$

Once the distances are computed, the algorithm selects the k nearest neighbors based on the smallest distance values [29]. The number k determines how many neighbors are considered when classifying a data point. The algorithm utilizes the Euclidean distance metric for classification, and the parameter n neighbors (the number of neighbors to consider) is typically optimized with values such as 1,2,3,… , 31.

The formula for Euclidean distance between two data points $x_i = x_{i,1}, x_{i,2}, \ldots, x_{i,n}$ and $x_j = x_{j,1}, x_{j,2}, \ldots, x_{j,n}$ in an n-dimensional space is shown in Formula 9.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \quad (9)$$

This formula measures the distance between each test data point and the training data points to identify the nearest neighbors.

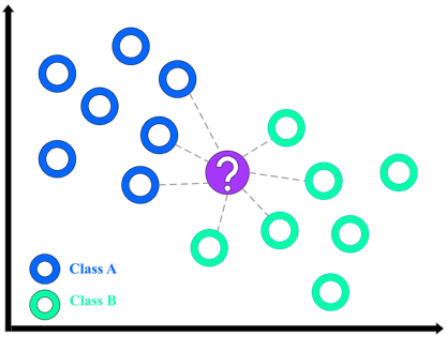


Figure 5. KNN Algorithm

Training data is used to train the model, validation data is used to find the best parameters through grid search, and testing data is used to evaluate the model's performance on new, unseen data.

2.8 Model Evaluation

Model evaluation is performed using various metrics to assess the performance of the classification algorithm. The main metrics used include accuracy, which is the percentage of correct predictions against the overall data shown in Formula 10; precision, which measures the proportion of correct positive predictions shown in Formula 11; recall, which evaluates the model's ability to detect positive classes shown in Formula 12; and F1-Score, which is the harmonic mean between precision and recall shown in Formula 13. In addition, evaluation is also conducted using the confusion matrix. This analytical tool provides a detailed picture of model performance by comparing model predictions against actual data, as shown in Figure 6. In the context of sentiment analysis, the main components of the confusion matrix include True Positive (TP), which is the amount of positive sentiment data that was correctly classified; True Negative (TN), the amount of negative sentiment data that was correctly classified; False Positive (FP), the amount of negative sentiment data that was misclassified as positive; and False Negative (FN), the amount of positive sentiment data that was misclassified as negative. The evaluation formulas based on the confusion matrix are shown in Formulas 10 to 13.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (10)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (11)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (12)$$

$$\text{F1} - \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}} \quad (13)$$

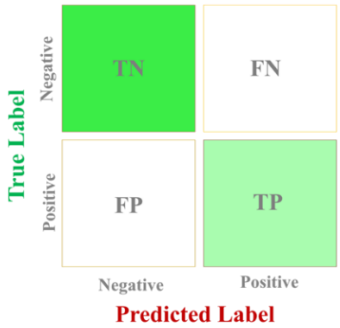


Figure 6. Confusion Matrix

Evaluations were conducted on cross-validation results and testing data to compare the performance of SVM, Naive Bayes, and KNN algorithms. Thus, the evaluations provided insight into each algorithm's strengths and weaknesses in effectively detecting positive and negative sentiment. The confusion matrix becomes an important tool for understanding the distribution of model errors, especially on unbalanced datasets.

The best model was selected based on the highest performance (high accuracy, precision, and recall) and used to predict the sentiment of new data. This model provides data-driven recommendations to the Sultan Hasanuddin Airport management to improve their services.

3. Results and Discussions

3.1 Results

SVM with linear kernel consistently performed best in various preprocessing stages, especially in the sastrawi & manual SMOTE stage with 70:20:10 data split. SVM showed high stability and superior accuracy on heavily preprocessed datasets. Naive Bayes performed very well on simple preprocessing, such as SMOTE

Sastrawi, and combined preprocessing, such as SMOTE Sastrawi & Manual, especially on the 2024 dataset. However, its performance tends to decrease on the 2021-2023 dataset with a 50:25:25 data split. KNN shows performance that is highly dependent on data distribution and parameter k. KNN has the best performance in the Manual SMOTE stage with a data split of 70:20:10 on the 2024 dataset but tends to perform poorly in deep preprocessing stages such as Sastrawi & Manual SMOTE on the 2024 dataset with a data split of 60:20:20. These details can be seen in the Table 3.

Table 3. Balancing by SMOTE

Dataset	Stages	Split data	Algorithm	Function	Parameter	Value	Validation				Testing			
							Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
2021 - 2023	SMOTE Sastrawi	50/25/25	SVM	Linear	C	1	91.9	91.9	91.9	91.9	89.6	89.5	89.6	89.6
			NB	Multinomial	Alpha	5.0	91.3	91.7	91.3	91.4	89.1	89.1	89.1	89.1
			KNN	Euclidean	K	1	66.2	83.4	66.2	66.9	69.9	81.3	69.9	70.9
		60/20/20	SVM	Linear	C	1	95.0	95.0	95.0	94.9	93.4	93.4	93.4	93.3
			NB	Multinomial	Alpha	0.5	93.9	93.9	93.9	93.9	91.2	91.3	91.2	91.2
			KNN	Euclidean	K	1	63.1	81.4	63.1	63.9	70.4	83.2	70.4	71.4
		70/20/10	SVM	Linear	C	1	95.3	95.3	95.3	95.3	90.0	90.7	90.0	89.4
			NB	Multinomial	Alpha	0.5	94.3	94.3	94.3	94.3	90.7	90.6	90.7	90.5
			KNN	Euclidean	K	3	60.2	83.0	60.2	60.5	63.8	81.6	63.8	65.2
2021 - 2023	SMOTE Manual	50/25/25	SVM	Linear	C	1	93.3	93.4	93.3	93.3	86.9	86.8	86.9	86.8
			NB	Multinomial	Alpha	10.0	92.7	92.9	92.7	92.8	86.6	86.6	86.6	86.5
			KNN	Euclidean	K	1	46.5	76.2	46.5	43.1	52.0	75.9	52.0	51.0
		60/20/20	SVM	Linear	C	1	95.3	95.3	95.3	95.3	88.6	89.0	88.6	88.8
			NB	Multinomial	Alpha	1.0	93.6	93.5	93.6	93.5	91.2	91.1	91.2	91.1
			KNN	Euclidean	K	1	46.4	75.8	46.4	43.8	56.2	77.6	56.2	56.0
		70/20/10	SVM	Linear	C	0.1	94.3	94.2	94.3	94.3	90.7	90.7	90.7	90.4
			NB	Multinomial	Alpha	1.0	93.9	93.9	93.9	93.9	92.1	92.2	92.1	91.9
			KNN	Euclidean	K	1	48.9	76.7	48.9	47.1	51.7	75.2	51.7	52.2
2021 - 2023	SMOTE Sastrawi & Manual	50/25/25	SVM	Linear	C	1	92.7	92.7	92.7	92.7	89.9	89.8	89.9	89.8
			NB	Multinomial	Alpha	5.0	91.9	92.2	91.9	92.0	89.6	89.6	89.6	89.6
			KNN	Euclidean	K	1	68.2	83.9	68.2	69.0	69.6	80.8	69.6	70.7
		60/20/20	SVM	Linear	C	1	95.3	95.3	95.3	95.3	94.1	94.1	94.1	94.0
			NB	Multinomial	Alpha	0.5	93.9	93.9	93.9	93.9	91.6	91.8	91.6	91.6
			KNN	Euclidean	K	1	66.6	82.4	66.6	67.7	70.4	83.2	70.4	71.4
		70/20/10	SVM	Linear	C	1	96.0	96.1	96.0	96.0	91.4	92.3	91.4	90.9
			NB	Multinomial	Alpha	5.0	94.3	94.4	94.3	94.3	90.7	90.6	90.7	90.6
			KNN	Euclidean	K	3	60.9	83.2	60.9	61.3	60.9	80.8	60.9	62.2
2024	SMOTE Sastrawi	50/25/25	SVM	Linear	C	1	91.9	92.5	91.9	91.7	91.1	91.5	91.1	90.9
			NB	Multinomial	Alpha	10.0	92.8	92.7	92.8	92.7	89.7	89.9	89.7	89.7
			KNN	Euclidean	K	1	70.7	78.3	70.7	71.5	67.7	79.4	67.7	67.2
		60/20/20	SVM	Linear	C	1	93.5	93.8	93.5	93.4	93.9	93.9	93.9	93.9
			NB	Multinomial	Alpha	5.0	95.3	95.3	95.3	95.3	90.4	90.9	90.4	90.5
			KNN	Euclidean	K	3	61.9	82.5	61.9	62.4	59.2	79.3	59.2	57.6
		70/20/10	SVM	Linear	C	1	95.3	95.4	95.3	95.2	96.2	96.2	96.2	96.1
			NB	Multinomial	Alpha	5.0	95.3	95.3	95.3	95.3	93.9	93.9	93.9	93.9
			KNN	Euclidean	K	3	65.1	84.0	65.1	65.8	69.6	84.4	69.6	70.2
2024	SMOTE Manual	50/25/25	SVM	Linear	C	1	91.6	92.2	91.6	91.4	90.8	91.0	90.8	90.7
			NB	Multinomial	Alpha	5.0	93.1	93.1	93.1	93.0	89.1	89.2	89.1	89.1
			KNN	Euclidean	K	1	79.9	83.0	79.9	80.4	75.1	81.5	75.1	75.3
		60/20/20	SVM	Linear	C	1	93.5	93.7	93.5	93.4	91.4	91.5	91.4	91.3
			NB	Multinomial	Alpha	1.0	95.3	95.3	95.3	95.3	91.4	91.6	91.4	91.5
			KNN	Euclidean	K	5	70.4	84.0	70.4	71.5	65.6	81.1	65.6	65.2
		70/20/10	SVM	Linear	C	1	93.5	93.7	93.5	93.4	94.6	94.7	94.6	94.6
			NB	Multinomial	Alpha	1.0	96.0	96.0	96.0	96.0	93.9	93.9	93.9	93.9
			KNN	Euclidean	K	3	72.9	83.7	72.9	74.0	78.0	86.0	78.0	78.7
2024	SMOTE Sastrawi & Manual	50/25/25	SVM	Linear	C	1	91.9	92.5	91.9	91.7	90.2	90.6	90.2	90.0
			NB	Multinomial	Alpha	5.0	93.1	93.1	93.1	93.0	90.8	90.9	90.8	90.9
			KNN	Euclidean	K	1	70.7	78.3	70.7	71.5	69.1	79.9	69.1	68.8
		60/20/20	SVM	Linear	C	1	94.3	94.4	94.3	94.1	93.9	93.9	93.9	93.9
			NB	Multinomial	Alpha	5.0	95.0	94.9	95.0	94.9	91.1	91.4	91.1	91.2
			KNN	Euclidean	K	3	62.9	82.0	62.9	63.6	58.8	78.3	58.8	57.3
		70/20/10	SVM	Linear	C	1	95.7	95.7	95.7	95.6	96.2	96.2	96.2	96.1
			NB	Multinomial	Alpha	5.0	95.3	95.3	95.3	95.3	93.9	93.9	93.9	93.8
			KNN	Euclidean	K	3	64.7	83.2	64.7	65.5	68.9	84.2	68.9	69.4

Acc : Accuracy, Prec : Precision, Rec : Recall, F1 : F1-Score.

SVM is the most stable algorithm across all cross-validation folds, showing superiority in handling complex preprocessing such as Sastrawi & Manual SMOTE. Naive Bayes is better suited for simple preprocessing such as SMOTE Sastrawi. KNN is superior on a larger dataset (2024) with Sastrawi

SMOTE preprocessing. These details can be seen in the Table 4.

Table 4. Balancing by SMOTE (Cross Validation)

Dataset	Stages	Split data	Algorithm	Function	Parameter	Value	Performance			
							Acc	Prec	Rec	F1
2021 - 2023	SMOTE Sastrawi Cross Validation	4	SVM	Linear	C	1	93.1	93.1	93.1	93.1
			NB	Multinomial	Alpha	0.1	91.4	91.5	91.4	91.4
			KNN	Euclidean	K	1	84.9	86.7	84.9	84.7
		5	SVM	Linear	C	1	92.8	92.8	92.8	92.8
			NB	Multinomial	Alpha	0.1	91.3	91.4	91.3	91.3
			KNN	Euclidean	K	1	86.1	87.6	86.1	86.0
		10	SVM	Linear	C	1	93.0	93.1	93.0	93.0
			NB	Multinomial	Alpha	0.01	91.8	91.9	91.8	91.8
			KNN	Euclidean	K	1	88.2	88.8	88.2	88.1
	SMOTE Manual Cross Validation	4	SVM	Linear	C	1	91.3	91.4	91.3	91.3
			NB	Multinomial	Alpha	0.1	89.2	89.3	89.2	89.2
			KNN	Euclidean	K	1	61.2	77.3	61.2	57.4
		5	SVM	Linear	C	1	91.7	91.8	91.7	91.7
			NB	Multinomial	Alpha	0.1	89.2	89.3	89.2	89.2
			KNN	Euclidean	K	1	61.4	76.9	61.4	57.8
		10	SVM	Linear	C	1	91.7	91.8	91.7	91.7
			NB	Multinomial	Alpha	0.1	89.8	89.9	89.8	89.8
			KNN	Euclidean	K	1	61.8	76.9	61.8	58.3
2021 - 2023	SMOTE Sastrawi & Manual Cross Validation	4	SVM	Linear	C	1	92.8	92.8	92.8	92.8
			NB	Multinomial	Alpha	0.1	91.4	91.4	91.4	91.4
			KNN	Euclidean	K	1	85.4	87.1	85.4	85.2
		5	SVM	Linear	C	1	92.5	92.6	92.5	92.5
			NB	Multinomial	Alpha	0.01	91.3	91.3	91.3	91.2
			KNN	Euclidean	K	1	86.6	88.0	86.6	86.5
		10	SVM	Linear	C	1	92.8	92.9	92.8	92.8
			NB	Multinomial	Alpha	0.01	91.8	91.9	91.8	91.8
			KNN	Euclidean	K	1	88.5	89.1	88.5	88.4
2024	SMOTE Sastrawi Cross Validation	4	SVM	Linear	C	1	95.2	95.2	95.2	95.2
			NB	Multinomial	Alpha	0.01	94.2	94.2	94.2	94.2
			KNN	Euclidean	K	1	91.2	91.4	91.2	91.1
		5	SVM	Linear	C	1	95.1	95.1	95.1	95.1
			NB	Multinomial	Alpha	0.1	94.1	94.1	94.1	94.1
			KNN	Euclidean	K	1	91.7	92.0	91.7	91.6
		10	SVM	Linear	C	1	95.0	95.0	95.0	95.0
			NB	Multinomial	Alpha	0.01	94.3	94.3	94.3	94.3
			KNN	Euclidean	K	1	92.1	92.3	92.1	92.0
	SMOTE Manual Cross Validation	4	SVM	Linear	C	1	94.3	94.3	94.3	94.3
			NB	Multinomial	Alpha	0.1	93.4	93.4	93.4	93.4
			KNN	Euclidean	K	3	81.5	85.9	81.5	80.8
		5	SVM	Linear	C	10	94.0	94.1	94.0	94.0
			NB	Multinomial	Alpha	0.01	93.1	93.2	93.1	93.1
			KNN	Euclidean	K	3	81.6	86.4	81.6	80.8
		10	SVM	Linear	C	10	94.3	94.4	94.3	94.3
			NB	Multinomial	Alpha	0.1	93.4	93.5	93.4	93.4
			KNN	Euclidean	K	3	86.6	88.9	86.6	86.3
2024	SMOTE Sastrawi & Manual Cross Validation	4	SVM	Linear	C	1	95.5	95.5	95.5	95.5
			NB	Multinomial	Alpha	0.01	94.3	94.3	94.3	94.3
			KNN	Euclidean	K	1	90.4	90.6	90.4	90.4
		5	SVM	Linear	C	1	95.1	95.1	95.1	95.1
			NB	Multinomial	Alpha	0.01	94.0	94.0	94.0	94.0
			KNN	Euclidean	K	1	91.2	91.4	91.2	91.2
		10	SVM	Linear	C	1	95.1	95.2	95.1	95.1
			NB	Multinomial	Alpha	0.01	94.2	94.2	94.2	94.2
			KNN	Euclidean	K	1	91.3	91.4	91.3	91.2

SVM is the most stable and superior algorithm across all preprocessing stages and datasets, with the highest performance on the 2024 dataset, Sastrawi stage, and 70:20:10 data split. KNN has high potential in complex preprocessing without balancing but is highly dependent on the *k* parameter and sensitive to datasets with less-than-ideal class distributions. Preprocessing, such as Manual, does not provide optimal results for KNN. These details can be seen in the Table 5.

Table 5. Performance Without Balancing

Dataset	Stages	Split data	Algorithm	Function	Parameter	Value	Validation				Testing			
							Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
2021 - 2023	Sastrawi	50/25/25	SVM	Linear	C	1	91.9	91.9	91.9	91.9	89.6	89.5	89.6	89.6
			NB	Multinomial	Alpha	5.0	91.3	91.7	91.3	91.4	89.1	89.1	89.1	89.1
			KNN	Euclidean	K	1	66.2	83.4	66.2	66.9	69.9	81.3	69.9	70.9
		60/20/20	SVM	Linear	C	1	95.0	95.0	95.0	94.9	93.4	93.4	93.4	93.3
			NB	Multinomial	Alpha	0.5	93.9	93.9	93.9	93.9	91.2	91.3	91.2	91.2
			KNN	Euclidean	K	1	63.1	81.4	63.1	63.9	70.4	83.2	70.4	71.4
		70/20/10	SVM	Linear	C	1	95.3	95.3	95.3	95.3	90.0	90.7	90.0	89.4
			NB	Multinomial	Alpha	0.5	94.3	94.3	94.3	94.3	90.7	90.6	90.7	90.5
			KNN	Euclidean	K	3	60.2	83.0	60.2	60.5	63.8	81.6	63.8	65.2
2021 - 2023	Manual	50/25/25	SVM	Linear	C	1	93.3	93.4	93.3	93.3	86.9	86.8	86.9	86.8
			NB	Multinomial	Alpha	10.0	92.7	92.9	92.7	92.8	86.6	86.6	86.6	86.5
			KNN	Euclidean	K	1	46.5	76.2	46.5	43.1	52.0	75.9	52.0	51.0
		60/20/20	SVM	Linear	C	1	95.3	95.3	95.3	95.3	88.6	89.0	88.6	88.8
			NB	Multinomial	Alpha	1.0	93.6	93.5	93.6	93.5	91.2	91.1	91.2	91.1
			KNN	Euclidean	K	1	46.4	75.8	46.4	43.8	56.2	77.6	56.2	56.0
		70/20/10	SVM	Linear	C	0.1	94.3	94.2	94.3	94.3	90.7	90.7	90.7	90.4
			NB	Multinomial	Alpha	1.0	93.9	93.9	93.9	93.9	92.1	92.2	92.1	91.9
			KNN	Euclidean	K	1	48.9	76.7	48.9	47.1	51.7	75.2	51.7	52.2
2021 – 2023	Sastrawi & Manual	50/25/25	SVM	Linear	C	1	92.7	92.7	92.7	92.7	89.9	89.8	89.9	89.8
			NB	Multinomial	Alpha	5.0	91.9	92.2	91.9	92.0	89.6	89.6	89.6	89.6
			KNN	Euclidean	K	1	68.2	83.9	68.2	69.0	69.6	80.8	69.6	70.7
		60/20/20	SVM	Linear	C	1	95.3	95.3	95.3	95.3	94.1	94.1	94.1	94.0
			NB	Multinomial	Alpha	0.5	93.9	93.9	93.9	93.9	91.6	91.8	91.6	91.6
			KNN	Euclidean	K	1	66.6	82.4	66.6	67.7	70.4	83.2	70.4	71.4
		70/20/10	SVM	Linear	C	1	96.0	96.1	96.0	96.0	91.4	92.3	91.4	90.9
			NB	Multinomial	Alpha	5.0	94.3	94.4	94.3	94.3	90.7	90.6	90.7	90.6
			KNN	Euclidean	K	3	60.9	83.2	60.9	61.3	60.9	80.8	60.9	62.2
2024	Sastrawi	50/25/25	SVM	Linear	C	1	91.9	92.5	91.9	91.7	91.1	91.5	91.1	90.9
			NB	Multinomial	Alpha	10.0	92.8	92.7	92.8	92.7	89.7	89.9	89.7	89.7
			KNN	Euclidean	K	1	70.7	78.3	70.7	71.5	67.7	79.4	67.7	67.2
		60/20/20	SVM	Linear	C	1	93.5	93.8	93.5	93.4	93.9	93.9	93.9	93.9
			NB	Multinomial	Alpha	5.0	95.3	95.3	95.3	95.3	90.4	90.9	90.4	90.5
			KNN	Euclidean	K	3	61.9	82.5	61.9	62.4	59.2	79.3	59.2	57.6
		70/20/10	SVM	Linear	C	1	95.3	95.4	95.3	95.2	96.2	96.2	96.2	96.1
			NB	Multinomial	Alpha	5.0	95.3	95.3	95.3	95.3	93.9	93.9	93.9	93.9
			KNN	Euclidean	K	3	65.1	84.0	65.1	65.8	69.6	84.4	69.6	70.2
2024	Manual	50/25/25	SVM	Linear	C	1	91.6	92.2	91.6	91.4	90.8	91.0	90.8	90.7
			NB	Multinomial	Alpha	5.0	93.1	93.1	93.1	93.0	89.1	89.2	89.1	89.1
			KNN	Euclidean	K	1	79.9	83.0	79.9	80.4	75.1	81.5	75.1	75.3
		60/20/20	SVM	Linear	C	1	93.5	93.7	93.5	93.4	91.4	91.5	91.4	91.3
			NB	Multinomial	Alpha	1.0	95.3	95.3	95.3	95.3	91.4	91.6	91.4	91.5
			KNN	Euclidean	K	5	70.4	84.0	70.4	71.5	65.6	81.1	65.6	65.2
		70/20/10	SVM	Linear	C	1	93.5	93.7	93.5	93.4	94.6	94.7	94.6	94.6
			NB	Multinomial	Alpha	1.0	96.0	96.0	96.0	96.0	93.9	93.9	93.9	93.9
			KNN	Euclidean	K	3	72.9	83.7	72.9	74.0	78.0	86.0	78.0	78.7
2024	Sastrawi & Manual	50/25/25	SVM	Linear	C	1	91.9	92.5	91.9	91.7	90.2	90.6	90.2	90.0
			NB	Multinomial	Alpha	5.0	93.1	93.1	93.1	93.0	90.8	90.9	90.8	90.9
			KNN	Euclidean	K	1	70.7	78.3	70.7	71.5	69.1	79.9	69.1	68.8
		60/20/20	SVM	Linear	C	1	94.3	94.4	94.3	94.1	93.9	93.9	93.9	93.9
			NB	Multinomial	Alpha	5.0	95.0	94.9	95.0	94.9	91.1	91.4	91.1	91.2
			KNN	Euclidean	K	3	62.9	82.0	62.9	63.6	58.8	78.3	58.8	57.3
		70/20/10	SVM	Linear	C	1	95.7	95.7	95.7	95.6	96.2	96.2	96.2	96.1
			NB	Multinomial	Alpha	5.0	95.3	95.3	95.3	95.3	93.9	93.9	93.9	93.8
			KNN	Euclidean	K	3	64.7	83.2	64.7	65.5	68.9	84.2	68.9	69.4

SVM is the most stable and superior algorithm, with the best performance at complex preprocessing stages such as Sastrawi & Manual, especially with the 2024 dataset and 10-fold cross-validation. Naive Bayes showed superiority on simple preprocessing, such as Sastrawi, with consistent performance across all cross-validation folds but was less optimal on complex preprocessing. KNN performs better on new datasets (2024) and complex preprocessing but is very sensitive to *k* valuesand shows lower performance on low folds and simple preprocessing. These details can be seen in the Table 6.

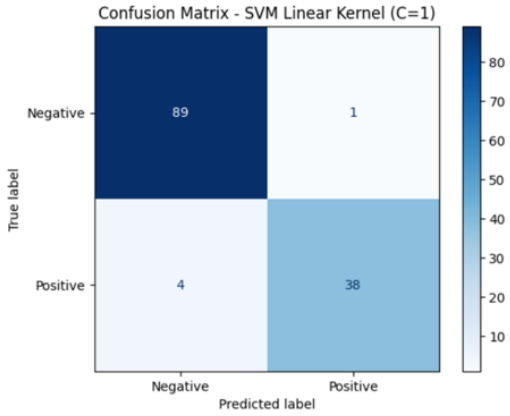
Table 6. Performance Without Balancing (Cross-Validation)

Dataset	Stages	Split data	Algorithm	Function	Parameter	Value	Performance			
							Acc	Prec	Rec	F1
2021 - 2023	Sastrawi Cross Validation	4	SVM	Linear	C	1	90.2	90.1	90.2	90.0
			NB	Multinomial	Alpha	0.1	88.2	88.2	88.2	87.7
			KNN	Euclidean	K	27	87.1	87.0	87.1	86.7
		5	SVM	Linear	C	1	91.1	91.1	91.1	91.0
			NB	Multinomial	Alpha	0.1	88.8	88.7	88.8	88.4
			KNN	Euclidean	K	25	86.7	86.5	86.7	86.4
		10	SVM	Linear	C	1	90.4	90.4	90.4	90.3
			NB	Multinomial	Alpha	0.1	88.6	88.8	88.6	88.1
			KNN	Euclidean	K	29	86.8	86.7	86.8	86.5
	Manual Cross Validation	4	SVM	Linear	C	1	90.6	90.6	90.6	90.4
			NB	Multinomial	Alpha	0.1	87.8	87.8	87.8	87.5
			KNN	Euclidean	K	23	85.6	85.6	85.6	85.5
		5	SVM	Linear	C	1	90.6	90.6	90.6	90.4
			NB	Multinomial	Alpha	0.1	88.9	88.8	88.9	88.6
			KNN	Euclidean	K	17	85.2	85.5	85.2	85.3
		10	SVM	Linear	C	1	90.6	90.6	90.6	90.4
			NB	Multinomial	Alpha	0.1	88.4	88.5	88.4	88.0
			KNN	Euclidean	K	25	85.9	86.0	85.9	85.8
	Sastrawi & Manual Cross Validation	4	SVM	Linear	C	1	90.4	90.3	90.4	90.2
			NB	Multinomial	Alpha	0.1	88.0	88.1	88.0	87.6
			KNN	Euclidean	K	29	86.6	86.5	86.6	86.2
		5	SVM	Linear	C	1	90.9	90.9	90.9	90.8
			NB	Multinomial	Alpha	0.1	88.6	88.6	88.6	88.3
			KNN	Euclidean	K	15	86.5	86.3	86.5	86.2
		10	SVM	Linear	C	1	90.5	90.5	90.5	90.4
			NB	Multinomial	Alpha	0.1	88.5	88.6	88.5	88.0
			KNN	Euclidean	K	11	86.4	86.3	86.4	86.2
2024	Sastrawi Cross Validation	4	SVM	Linear	C	1	92.6	92.6	92.6	92.5
			NB	Multinomial	Alpha	0.1	89.5	89.5	89.5	89.3
			KNN	Euclidean	K	29	88.8	88.7	88.8	88.4
		5	SVM	Linear	C	1	92.3	92.3	92.3	92.2
			NB	Multinomial	Alpha	0.1	90.1	90.0	90.1	89.8
			KNN	Euclidean	K	13	88.6	88.6	88.6	88.4
		10	SVM	Linear	C	1	92.7	92.8	92.7	92.6
			NB	Multinomial	Alpha	0.1	90.2	90.2	90.2	90.0
			KNN	Euclidean	K	13	88.8	88.8	88.8	88.6
	Manual Cross Validation	4	SVM	Linear	C	1	91.5	91.5	91.5	91.4
			NB	Multinomial	Alpha	0.01	89.5	89.4	89.5	89.2
			KNN	Euclidean	K	11	87.3	87.2	87.3	87.2
		5	SVM	Linear	C	1	91.6	91.6	91.6	91.5
			NB	Multinomial	Alpha	0.1	89.9	89.8	89.9	89.7
			KNN	Euclidean	K	11	87.3	87.2	87.3	87.2
		10	SVM	Linear	C	1	92.1	92.1	92.1	92.0
			NB	Multinomial	Alpha	0.1	89.6	89.7	89.6	89.4
			KNN	Euclidean	K	11	87.8	87.7	87.8	87.7
	Sastrawi & Manual Cross Validation	4	SVM	Linear	C	1	92.4	92.5	92.4	92.3
			NB	Multinomial	Alpha	0.1	89.5	89.5	89.5	89.3
			KNN	Euclidean	K	29	88.8	88.8	88.8	88.5
		5	SVM	Linear	C	1	92.3	92.4	92.3	92.1
			NB	Multinomial	Alpha	0.1	88.9	90.0	90.0	89.7
			KNN	Euclidean	K	13	88.9	88.9	88.9	88.7
		10	SVM	Linear	C	1	93.2	93.2	93.2	93.1
			NB	Multinomial	Alpha	0.1	90.0	90.1	90.0	89.8
			KNN	Euclidean	K	13	89.0	89.0	89.0	88.9

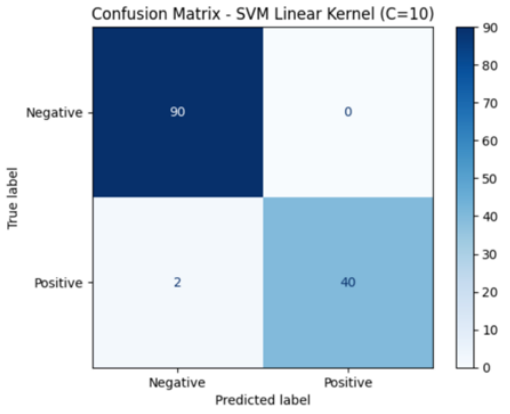
.SVM consistently performed best in all preprocessing stages, especially in Sastrawi & Manual with the 2024 dataset. At a data split of 70:20:10, the highest F1 score was achieved at 96.9% for validation and testing. In Sastrawi, with the 2024 dataset at 70:20:10 data split, the highest F1-score was achieved at 98.4%. SVM remained stable at all cross-validation folds (4-fold, 5-fold, and 10-fold), recording the highest F1-score of 95.5% on dataset 2024 with Sastrawi & Manual SMOTE preprocessing. Naive Bayes is superior at the SMOTE Sastrawi stage, with F1-score testing reaching 93.9% on dataset 2024. Naive Bayes performs optimally on all cross-validation folds, with F1-score stable at around 94% on dataset 2024, with SMOTE Sastrawi. Performance degrades under complex preprocessing such as Sastrawi & Manual on the 2021-2023 dataset, where F1-score testing only reaches 84.7% on a 60:20:20 data split. Highly dependent on

smoothing parameter (α), with optimal results at $\alpha = 0.1$ or $\alpha = 0.01$. KNN showed the best performance, with up to 92.0% accuracy and 91.2% F1 Score on the 2024 dataset, both on standard data splits (50/25/25, 60/20/20, 70/20/10) and cross-validation.

Confusion matrix of the highest accuracy at each stage is shown in Figure 7 while the highest accuracy bloxplot from cross validations is presanted in Figure 8 and 9. In addition, Figure 10, 11 and 12 depict the dominant or frequently occurring word and 10 dominant words are summarized in Table 7.



(a)



(b)

Figure 7. Confusion Matrix (a) Balanced by SMOTE, (b) Without balanced of Linear SVM

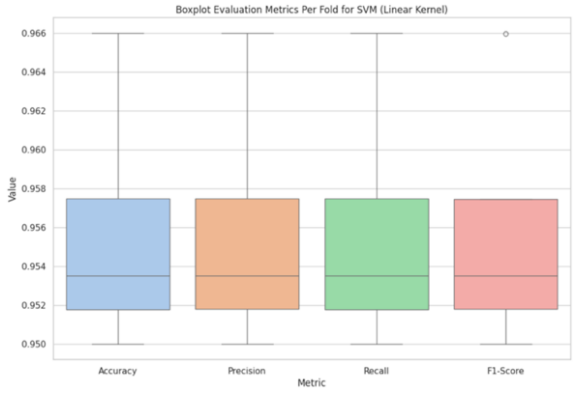


Figure 8. Boxplot balancing by SMOTE Cross Validation

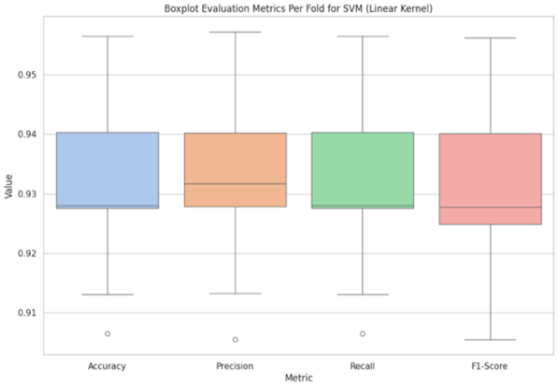


Figure 9. Boxplot Without balancing Cross Validation

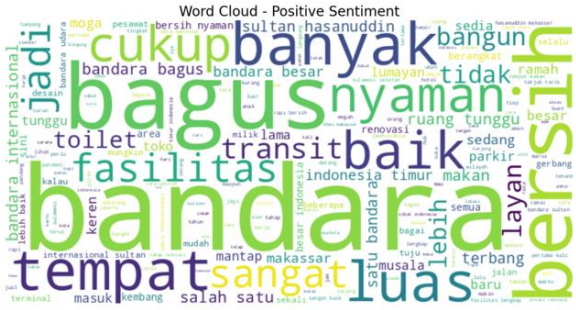


Figure 10. Word Cloud – Positive Sentiment

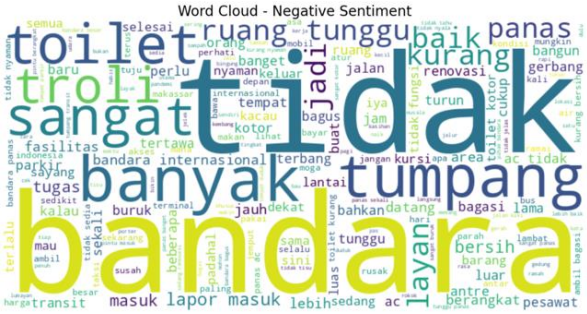


Figure 11. Word Cloud – Negative Sentiment

Figure 12. Word Cloud – All Text

Table 7. Most Frequently Appearing Words

No	Training Data	Validation Data	Testing Data
1	*Bandara*	*Bandara*	*Bandara*
2	*Tidak*	*Tidak*	*Tidak*
3	*Toilet*	*Panas*	*Panas*
4	*Banyak*	*Tunggu*	*Bagus*
5	*Sangat*	*Ruang*	*Ac*
6	*Baik*	*Ac*	*Bersih*
7	*Bersih*	*Sangat*	*Toilet*
8	*Masuk*	*Banyak*	*Kurang*
9	*Tempat*	*Baik*	*Sangat*
10	*Kurang*	*Toilet*	*Tunggu*

3.2 Discussions

SVM performed best across different preprocessing scenarios and data balancing. Naive Bayes performed better on simple datasets using balancing. However, performance degrades on more complex datasets. KNN shows the lowest performance on large datasets with data balancing. This algorithm is more suitable for small datasets without data balancing or high noise. Sastrawi gave the best results in all experiments. On the 2024 dataset, this preprocessing resulted in 98.4% accuracy with SVM. This combination of sastrawi and manual preprocessing gave stable results, especially when used with SMOTE data balancing and cross validation. Accuracy reached 95.5% on the 2024 dataset. On the 2024 dataset with Sastrawi preprocessing without SMOTE, SVM produced the highest accuracy of 98.4%. This shows that data balancing is only sometimes necessary, especially if the dataset is already relatively balanced or when the algorithm can handle class imbalance naturally. This result also indicates that balancing can be beneficial only under certain conditions, such as datasets with significant class imbalance. SMOTE allows algorithms such as Naive Bayes to perform more optimally on datasets with an initially unbalanced class distribution. SMOTE has been shown to help improve model performance on datasets with class imbalance, especially for algorithms such as SVM and Naive Bayes. However, data balancing is only sometimes necessary. In addition, algorithms such as KNN show weaknesses in handling synthetic data, which can compromise performance due to its sensitivity to neighborhood distributions. Thus, the selection of data balancing should be tailored to the dataset's characteristics and the algorithm being used. Cross-validation techniques provide more stable evaluation results than simple dataset division. Stratified K-Fold Validation. With k=4, this technique yielded higher accuracy (95.5%) than k=10 in SMOTE Sastrawi & Manual preprocessing. This shows that smaller folds can provide a more representative evaluation, especially on datasets with more complex class distributions. However, it only sometimes gives better results compared to k=5 and k=10. Based on the evaluation results, the SVM algorithm consistently performs best across various preprocessing, data balancing, and dataset-splitting scenarios. SVM with a linear kernel achieved the highest accuracy of 98.4% on the 2024 dataset with a combination of Sastrawi preprocessing and 70/20/10 data split. This performance shows that SVM excels in capturing complex sentiment patterns, especially on datasets processed with deep text techniques. Naive Bayes is Optimal for simple datasets with data balancing but degrades slightly on complex datasets. KNN has Unstable performance on large datasets with balancing, making it less suitable for this scenario. KNN shows significant performance degradation up to 51.0%. This study's results significantly contribute to Sultan Hasanuddin Airport's management to improve the quality of data-driven services. The study's validity was ensured by ensuring the review data was taken from the Google Maps platform, which is relevant for representing user perceptions of Sultan Hasanuddin Airport services. The text preprocessing techniques, such as filtering and stopword and data balancing using SMOTE, were designed to maximize the model's ability to capture complex sentiment patterns. In addition, the selection of SVM, Naive Bayes, and KNN algorithms was based on previous studies showing their effectiveness in sentiment analysis tasks. The reliability of the results was tested using stratified k-fold cross-validation, which ensures that the model evaluation covers the entire data distribution evenly and consistently. The model was tested on various data-sharing scenarios (50:25:25, 60:20:20, 70:20:10) and preprocessing to ensure stable results under different conditions. This approach allows the results to be reliable and replicated by other researchers. In addition, this research opens up great opportunities for developing more sophisticated sentiment analysis methods in the future, with potential applications in other public service sectors. The combination of in-depth data analysis and the application of modern technology can be a strategic step to improve customer satisfaction and competitiveness of air transport services. This research has several limitations, including limitations on the dataset, which only includes reviews from Google Maps, so it does not represent a broader perception of various platforms. Another challenge lies in text preprocessing, especially in dealing with slang or complex informal expressions, which are sometimes difficult to normalize accurately. In addition, some of the algorithms used showed sensitivity to data balancing, which may affect model performance when data is unbalanced or when certain balancing methods are applied. Future research could enrich the dataset by combining data from different platforms such as TripAdvisor, Twitter, or Instagram to create a more diverse dataset and cover a longer period (e.g., 2018-2020) to analyze the impact of COVID-19 on airport service perceptions. In text preprocessing, it is recommended to use context-based models such as BERT to capture the nuances of the Indonesian language, expand the slang dictionary, and integrate word embeddings such as Word2Vec or FastText. In addition, advanced algorithms such as LSTM or Transformer can improve sentiment analysis. In contrast, KNN optimization through data balancing methods such as ADASYN or weighted nearest neighbors can improve model performance. This research significantly contributes to improving the management of public services in the air transport sector, especially through machine learning-based user review analysis. In addition, this research enriches sentiment analysis methods with insights into the importance of text preprocessing and data balancing. The findings are also relevant to other sectors, opening up opportunities for further research to develop more sophisticated and applicable methods. This research complements previous studies by adding the SVM

algorithm, which achieved 98.4% accuracy, higher than Random Forest (83%) and KNN (82%) in sentiment analysis of airline reviews, and optimizing the SVM linear kernel on Google Maps reviews, surpassing the RBF kernel approach which achieved 84.37% accuracy. This research also improves on previous studies' weaknesses in data balancing using SMOTE, increasing accuracy from 91.4% to 94.0% in SVM and Naive Bayes and applying deep text preprocessing such as slang normalization and Sastraw-manual combination for more stable performance. Significant differences include more varied preprocessing, relevant data balancing approaches for unbalanced datasets, and robust evaluation with stratified k-fold cross-validation to ensure better model generalization.

4. Conclusions

This research aims to analyze public sentiment towards services and facilities at Sultan Hasanuddin Airport using three machine learning algorithms, namely SVM, Naive Bayes Multinomial, and KNN. This research also explores various aspects of sentiment analysis, including data-sharing techniques, model validation, text preprocessing, data balancing using SMOTE, and method parameterization. The results showed that SVM with a linear kernel gave the best performance, achieving an F1-score of 98.4% with Sastrawi preprocessing and was stable with 4-fold cross-validation, resulting in a score of 95.5%. Naive Bayes showed optimal performance on Sastrawi preprocessing with SMOTE, achieving an F1-score of 93.9% and stable on 10-fold cross-validation with a score of 94.3%. Meanwhile, KNN recorded the best F1-score of 92.0% on Sastrawi preprocessing and SMOTE, although its performance highly depended on parameter k and data distribution. The SMOTE technique effectively improved Naive Bayes's performance on unbalanced datasets but had no significant impact on SVM. In contrast, SMOTE improved KNN accuracy on cross-validation but decreased accuracy on data sharing. Literary preprocessing yielded the best overall performance, while manual methods were less effective, although the combination of the two sometimes showed significant improvement in some instances. In addition, 10-fold cross-validation provided a more stable evaluation than other methods, and data sharing, with a ratio of 70:20:10, showed the best results. This study provides practical recommendations for Sultan Hasanuddin Airport managers, such as optimizing air conditioning in hot areas, adding smoke-free zones, tidying up parking areas to improve efficiency, and providing special zones for ride-hailing services to reduce congestion. In addition, maintenance and replacement of damaged trolleys are also recommended to enhance passenger comfort. For future research, it is recommended that the dataset be expanded by integrating data from other platforms such as TripAdvisor, Twitter, and Instagram to create a more diverse dataset. In addition, model-based approaches such as BERT and advanced algorithms such as LSTM or Transformer can be used to improve the accuracy of sentiment analysis, while balancing techniques such as ADASYN can optimize the performance of KNN on synthetic data. This research contributes to the development of machine learning-based sentiment analysis methods. It provides practical insights to improve the management of public services in the air transportation sector more broadly and sustainably.

References

[1] N. Putu, O. Intan, Y. Putri, and S. Widiyanesti, "Sentiment Analysis of West Java International Airport (BIJB) Kertajati on Twitter," *Jurnal Manajemen dan Bisnis*, vol. 4, no. 2, 2020, doi: 10.36555/almana.v4i2.1348.

[2] I. Imelda and Arief Ramdhan Kurnianto, "Naïve Bayes and TF-IDF for Sentiment Analysis of the Covid-19 Booster Vaccine," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 1–6, Jan. 2023, doi: 10.29207/resti.v7i1.4467.

[3] M. S. M. Alanazi, J. Li, and K. W. Jenkins, "Evaluating Airport Service Quality Based on the Statistical and Predictive Analysis of Skytrax Passenger Reviews," *Applied Sciences (Switzerland)*, vol. 14, no. 20, Oct. 2024, doi: 10.3390/app14209472.

[4] A. Nurdina and A. B. I. Puspita, "Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis," *Journal of Information System Exploration and Research*, vol. 1, no. 2, Jul. 2023, doi: 10.52465/joiser.v1i2.167.

[5] L. Li, Y. Mao, Y. Wang, and Z. Ma, "How has airport service quality changed in the context of COVID-19: A data-driven crowdsourcing approach based on sentiment analysis," *J Air Transp Manag*, vol. 105, Oct. 2022, doi: 10.1016/j.jairtraman.2022.102298.

[6] A. Nieuwborg, M. Melles, S. Hiemstra-van Mastrigt, and S. Santema, "How can airports prepare for future public health disruptions? Experiences and lessons learned during the COVID-19 pandemic from a systemic perspective based on expert interviews," *Transp Res Interdiscip Perspect*, vol. 23, Jan. 2024, doi: 10.1016/j.trip.2023.101000.

[7] A. W. Sari, T. I. Hermanto, and M. Defriani, "Sentiment Analysis Of Tourist Reviews Using K-Nearest Neighbors Algorithm And Support Vector Machine," *Sinkron*, vol. 8, no. 3, pp. 1366–1378, Jul. 2023, doi: 10.33395/sinkron.v8i3.12447.

[8] E. Pujo Ariesanto Akhmad, K. Adi, and A. Puji Widodo, "Machine learning approach to customer sentiment analysis in twitter airline reviews," in *E3S Web of Conferences*, EDP Sciences, Nov. 2023. doi: 10.1051/e3sconf/202344802044.

[9] R. Song, W. Shi, W. Qin, X. Xue, and H. Jin, "Exploring Passengers' Emotions and Satisfaction: A Comparative Analysis of Airport and Railway Station through Online Reviews," *Sustainability (Switzerland)* , vol. 16, no. 5, Mar. 2024, doi: 10.3390/su16052108.

[10] S. Szeghalmy and A. Fazekas, "A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042333.

[11] A. M. Sarhan, H. Ayman, M. Wagdi, B. Ali, A. Adel, and R. Osama, "Integrating machine learning and sentiment analysis in movie recommendation systems," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, p. 53, Nov. 2024, doi: 10.1186/s43067-024-00177-7.

[12] R. Budiarto, P. Siber, and S. Negara, "Text Preprocessing for Optimal Accuracy in Indonesian Sentiment Analysis Using a Deep Learning Model with Word Embedding," 2021. doi: 10.1063/5.0126116.

[13] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics (Switzerland)*, vol. 9, no. 3, Mar. 2020, doi: 10.3390/electronics9030483.

[14] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion," in *Proceeding - 6th Information Technology International Seminar, ITIS 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 238–242. doi: 10.1109/ITIS50118.2020.9320958.

[15] L. Cahyaningrum, A. Luthfiarta, and M. Rahayu, "Sentiment Analysis on the Impact of MBKM on Student Organizations Using Supervised Learning with Smote to Handle Data Imbalance," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 9, no. 1, pp. 58–66, Jan. 2024, doi: 10.25139/inform.v9i1.7484.

[16] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, and R. Ocanitra, "Gojek and Grab User Sentiment Analysis on Google Play Using Naive Bayes Algorithm and Support Vector Machine Based Smote Technique," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Nov. 2020. doi: 10.1088/1742-6596/1641/1/012102.

[17] M. Hadwan, M. Al-Sarem, F. Saeed, and M. A. Al-Hagery, "An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique," *Applied Sciences (Switzerland)*, vol. 12, no. 11, Jun. 2022, doi: 10.3390/app12115547.

[18] A. Fadlil, I. Riadi, and F. Andrianto, "Improving Sentiment Analysis in Digital Marketplaces through SVM Kernel Fine-Tuning," *International Journal of Computing and Digital Systems*, vol. 16, no. 1, pp. 159–171, Jul. 2024, doi: 10.12785/ijcds/160113.

[19] A. Bhalla, "Comparative Analysis of Text Classification using SVM, Naïve Bayes, and KNN Models," in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2023, pp. 878–883. doi: 10.1109/ICIRCA57980.2023.10220728.

[20] K. Munawaroh, "Performance Comparison of SVM, Naïve Bayes, and KNN Algorithms for Analysis of Public Opinion Sentiment Against COVID-19 Vaccination on Twitter," *Journal of Advances in Information Systems and Technology*, vol. 4, no. 2, 2022, doi: 10.15294/jaist.v4i2.59493.

[21] V. B. Shtino and M. Muça, "Comparative Study of K-NN, Naive Bayes and SVM for Face Expression Classification Techniques," *Balkan Journal of Interdisciplinary Research*, vol. 9, no. 3, pp. 23–32, Dec. 2023, doi: 10.2478/bjir-2023-0015.

[22] H. Darwis, A. N. P. Pagala, S. Anraeni, T. Amaliah, I. As'ad, and A. U. Tenripada, "Analysis of Public Sentiment about Childfree in Indonesia using Support Vector Machine Methods," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–8. doi: 10.1109/IMCOM64595.2025.10857551.

[23] S. Akter *et al.*, *A Comprehensive Study Of Machine Learning Approaches For Customers Sentiment Analysis In Banking Sector*, vol. 6, no. 10. 2024, pp. 100–111. doi: 10.37547/tajet/Volume06Issue10-11.

[24] N. Umar and M. Adnan Nur, "Application of Naïve Bayes Algorithm Variations On Indonesian General Analysis Dataset for Sentiment Analysis," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 585–590, Aug. 2022, doi: 10.29207/resti.v6i4.4179.

[25] Azwan Triyadi, "Public Sentiment Analysis About Neuralink from Twitter Using Naïve Bayes: Multinomial, Gaussian and Complement," *The Indonesian Journal of Computer Science*, vol. 13, no. 5, Oct. 2024, doi: 10.33022/ijcs.v13i5.4278.

[26] J. Muliawan and E. Dazki, "Sentiment Analysis of Indonesia's Capital City Relocation Using Three Algorithms: Naive Bayes, KNN, and Random Forest," *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 5, pp. 1227–1236, 2023, doi: 10.52436/1.jutif.2023.4.5.347.

[27] F. M. J. M. Shamrat *et al.*, "Sentiment analysis on twitter tweets about COVID-19 vaccines using NLP and supervised KNN classification algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 463–470, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp463-470.

[28] M. F. Fakhrezi, Adian Fatchur Rochim, and Dinar Mutiara Kusomo Nugraheni, "Comparison of Sentiment Analysis Methods Based on Accuracy Value Case Study: Twitter Mentions of Academic Article," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 161–167, Feb. 2023, doi: 10.29207/resti.v7i1.4767.

[29] A. R. Isnain, J. Supriyanto, and M. P. Kharisma, "Implementation of K-Nearest Neighbor (K-NN) Algorithm For Public Sentiment Analysis of Online Learning," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 2, p. 121, Apr. 2021, doi: 10.22146/ijccs.65176.

