JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKTI SAINSTEK, No.0173/C3/DT.05.00/2025
Vol. 8, No. 2, August 2025, pp. 85-93 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v8i2.9865 e-ISSN: 2656-1948

COMPARISON OF NAÏVE BAYES CLASSIFIER AND K-NEAREST NEIGHBOR ALGORITHMS IN SENTIMENT ANALYSIS ON SOCIAL MEDIA X WITH VADER LEXICON

Steven Thiang[1*], Wenripin Chandra[2], Ferawaty[3], Mangasa A. S. Manulang[4]

[1,2,3,4]Universitas Pelita Harapan Kampus Medan
Email: [*1]03082210020@student.uph.edu, [2]wenripin@lecturer.uph.edu[2], ferawaty.fik@uph.edu, [4]mangasa.manullang@uph.edu

(Received: 13 May 2025, Revised: 11 June 2025, Accepted: 22 June 2025)

Abstract

The increasing use of social media as a platform for expressing public opinion has established platform X (formerly Twitter) an important data source for sentiment analysis. However, the ever-growing volume of data and the lack of sentiment labels present significant challenges for manual analysis, which is inefficient and time-consuming. This research addresses the problem of selecting effective algorithms for accurate and efficient sentiment classification on large-scale unlabeled data. The study aims to compare the performance of the Naïve Bayes Classifier and K-Nearest Neighbor (KNN) algorithms in sentiment classification related to the Value Added Tax (VAT) increase on platform X. To support classification accuracy, sentiment labeling is performed automatically using the VADER Lexicon. The research methodology involves data scraping, automatic sentiment labeling, implementation and training of classification models, and performance evaluation using a Confusion Matrix and ROC curve. The results show that the KNN algorithm with k = 1 achieved the best performance with an accuracy of 93.19%, precision of 94.07%, recall of 92.96%, a misclassification error of 6.81%, and an AUC of 0.95. In contrast, the Naïve Bayes Classifier achieved an accuracy of 88.29%, precision of 87.43%, recall of 86.67%, misclassification error of 11.71%, and an AUC of 0.93. Therefore, KNN is proven to be superior in classifying sentiment more accurately and efficiently than the Naïve Bayes Classifier.

Keywords: *Sentiment Analysis, Naïve Bayes Classifier, K-Nearest Neighbor, VADER Lexicon*

**Corresponding Author: Steven Thiang*

1. INTRODUCTION

Social media can serve as an important platform for understanding public opinions and responses to various subjects such as products, events, and more. One of the most widely used social media platforms globally is Platform X (formerly known as Twitter), which enables individuals from various countries to express their opinions and views on different matters [1]. Through Platform X, users share experiences through posts that spread quickly and reach a wide audience, making it a valuable data source for analyzing public opinion trends, understanding societal perceptions, and supporting decision-making in business, policy, and academic research [2]. In practice, manual analysis of the increasing volume of posts is considered inefficient [3]. Therefore, the use of information technology in sentiment analysis becomes an effective solution to understand public opinions and sentiments more quickly and accurately.

Text mining algorithms such as K-Nearest Neighbor (KNN) and Naïve Bayes Classifier can be used for sentiment analysis. The use of the K-Nearest Neighbor (KNN) algorithm in sentiment analysis regarding the Indonesian presidential election showed that using the training set achieved 100% accuracy, precision, recall, and f-measure. The 10-fold cross-validation test resulted in 92.5% accuracy, 100% precision, 91% recall, and 94% f-measure, while the 80% percentage split obtained 88.55% accuracy, 100% precision, 87% recall, and 93.04% f-measure. The KNN method with 80% percentage split proved superior to 10-fold cross-validation in sentiment classification [4]. Sentiment analysis of X application users regarding the free lunch program using the Naïve Bayes Classifier method achieved an accuracy of 80.31%, meaning that 80.31% of the sentiment

predictions matched the actual labels, with a margin of error of around 5.27%. Additionally, the precision results showed that the model classified positive sentiment with an accuracy of 80.69% and negative sentiment with an accuracy of 79.95%. Meanwhile, the recall results indicated that the model was able to detect positive sentiment with a sensitivity of 79.71% and negative sentiment with a sensitivity of 80.93% [2].

These studies used different datasets, making it difficult to directly compare the sentiment analysis results due to varying characteristics, structures, and complexities. Additionally, datasets obtained through the scraping process generally lack sentiment labels, thus requiring annotation before being used in analysis [5]. Conventional dataset labeling is often inefficient and can affect the accuracy of sentiment analysis, especially for large datasets. To address this challenge, this study employs the VADER Lexicon tool for automatic sentiment labeling, which can classify text into negative, positive, and neutral sentiment categories more accurately and efficiently [6]. This integration of VADER as an automatic labeling method serves as a novel contribution by significantly reducing manual labeling effort while maintaining or improving classification accuracy. This approach enhances the preprocessing phase, allowing the classification models to be trained on a well-labeled dataset with less time and resources, which is particularly important when dealing with large and noisy social media data.

In this study, we deliberately selected K-Nearest Neighbor and Naïve Bayes as baseline algorithms due to their simplicity, low computational cost, and established effectiveness in previous sentiment analysis tasks. These models provide a reliable foundation for performance benchmarking and allow researchers to assess sentiment classification feasibility on noisy social media datasets with minimal resource overhead [7]. Although other machine learning methods such as Decision Tree, Support Vector Machine (SVM), Random Forest, and Multi-Layer Perceptron (MLP) are widely recognized for their high classification accuracy, this study limits its scope to KNN and Naïve Bayes to maintain focus and computational efficiency. The use of more advanced models is proposed as part of future research directions to explore potential improvements in classification performance.

2. RESEARCH METHOD

2.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) is an AI field related to understanding and generating human language. With NLP techniques, computers can understand and generate text naturally, enabling applications such as machine translation, chatbots, and sentiment analysis [8]. NLP is an important branch of artificial intelligence that typically involves the interaction between machines/computers and humans using natural language [9].

With the development of Natural Language Processing (NLP), various methods and approaches have been developed to improve accuracy in understanding the meaning and emotions of a text. One of the approaches widely used in sentiment analysis is the use of Lexicon-Based Methods, which rely on lists of words along with their sentiment values. One such method is the Valence Aware Dictionary and sEntiment Reasoner (VADER), which has become a popular tool due to its ability to effectively analyze sentiment in texts, especially those from social media and other informal platforms [10].

2.2 VADER Lexicon

One of the analysis tools from Lexicon-Based methods is VADER (Valence Aware Dictionary and Sentiment Reasoner). VADER Lexicon is used to analyze data based on a Lexicon (dictionary). The analysis results in polarity classes such as positive, neutral, and negative, with an additional compound score or overall score. VADER Lexicon contains 7,500 words, including sentiment-related synonyms, acronyms, and English words [11].

Lexical is a dictionary used as the primary language in the Lexicon-Based method. To detect classification or sentiment, it is done by calculating the compound score using a formula 1 [12].

$$Compound\ Score = \frac{\sum_{i=1}^{n} Si}{\sqrt{(\sum_{i=1}^{n} S_i)^2 \alpha}} \quad (1)$$

Where:
Si is the sentiment score for the *i*-th word in the text.
n is the number of words analyzed.

The sentiment score for each word, Si, is obtained from the VADER Lexicon, which assigns a sentiment value based on the context of the word.

The sentiment classification process can be carried out using the following equation (Equation 2) [13].

$$Sentiment = \begin{cases} positive\ if\ compound > 0{,}05 \\ neutral\ if\ compound - 0{,}05 \leq compound \leq 0{,}05 \\ negative\ if\ compound < -0{,}05 \end{cases} \quad (2)$$

Another important approach in text processing is Term Frequency-Inverse Document Frequency (TF-IDF). Unlike the lexicon-based method, which relies on sentiment dictionaries, TF-IDF is a statistical approach used to evaluate how important a word is within a document relative to a collection of other documents. TF-IDF is often used as the basis for feature representation in various text mining and machine learning applications, including classification and sentiment analysis [13].

2.3 Term Frequency-Inverse Document Frequency *(TF-IDF)*

The TF-IDF method is one of the popular methods for determining the weight of each word.

TF-IDF assigns weights to words in a text document, reflecting the importance of those words within the document. The goal of the TF-IDF method is to obtain labels or sentiment for each word found in the document. Data that has gone through the pre-processing stage is then processed using word weighting with the TF-IDF method [14]. The stage of determining the weight values in the TF-IDF method is as follows [15]:

1. Calculation of Term Frequency (TF) in text documents which are assumed to have an equal level of importance in each document. The TF formula is:

$$\mathrm{TF(t,d)} = \frac{f_{t,d}}{\sum_k f_{k,d}} \quad (3)$$

Where:
$f_{t,d}$ is the frequency of word t in document d
$\sum_k f_{k,d}$ is the total of all words in the document d

2. Furthermore, the less frequently the term appears in the document, the Inverse Document Frequency (IDF) value will increase. The IDF formula is:

$$\mathrm{IDF(t)} = \log\left(\frac{N}{1+n_t}\right) \quad (4)$$

Where:
N is the total number of documents
n_t = number of documents containing the word t
Plus 1 in the denominator to avoid dividing by zero.

3. TF-IDF is the result of multiplying TF and IDF.

$$\mathrm{TF-IDF(t,d)} = \mathrm{TF(t,d)xIDF(t)} \quad (5)$$

In the context of sentiment analysis, the results of word weighting using the TF-IDF method become a numerical representation of the text document that will be used as input to the classification algorithm. One of the effective algorithms in processing this representation is the Naive Bayes Classifier. By utilizing TF-IDF values as features, this algorithm is able to learn the distribution patterns of words in each sentiment class and perform classification efficiently. Therefore, the following discussion will outline the basic concepts and applications of the Naive Bayes Classifier in sentiment analysis.

2.4 Naïve Bayes Classifier

The Naïve Bayes Classifier (NBC) algorithm is an algorithm used to determine the probability or likelihood in predicting chances based on previous data or to enable grouping within a systeml. There are several variants of the Naïve Bayes algorithm, with their application depending on the type of data used. These include [16]:

1. Gaussian Naïve Bayes – Used for continuous (numeric) data, assuming each feature follows a Gaussian (normal) distribution. Probability density is calculated using the mean and variance of the data.
2. Multinomial Naïve Bayes – Suitable for categorical data with multiple categories (e.g., text classification). It assumes features follow a multinomial distribution and is commonly used in tasks like sentiment analysis or spam detection.
3. Bernoulli Naïve Bayes – Similar to the Multinomial variant but designed for binary data (e.g., features with values 0 or 1). It assumes each feature follows a Bernoulli distribution.
4. Categorical Naïve Bayes – Used for categorical features, assuming each feature is generated from a categorical probability distribution. It's useful when dealing with many categorical features and limited numeric data. (2.3)

In this study, because the focus is on sentiment analysis, the Multinomial Naïve Bayes algorithm is used.

2.5 Multinomial Naïve Bayes Classifier

Multinomial Naive Bayes is a variant of the Naive Bayes algorithm specifically used for data in the form of term frequencies, such as the number of word occurrences in a text [17]. The formula for the Multinomial Naive Bayes Classifier is: (2.4)

1. Formula for the Multinomial assumption:
For Multinomial Naive Bayes, the likelihood is $P(x_i|C_k)$ is modeled using a Multinomial distribution with the formula [18]:

$$P(x_i|C_k) = \frac{count(x_i,C_k)+\alpha}{\sum_{i'} count(x_i,C_k)+\alpha V} \quad (6)$$

Where:
$\mathrm{count(x_i, C_k)}$ is the number of occurrences of the feature x_i in class C_k.
α is the smoothing parameter (usually using Laplace smoothing with a value of α=1).
V is the number of unique features (vocabulary size).

2. The formula for the total log probability is:
The sum of the posterior probabilities and all log likelihoods (Multinomial assumption) [19]:

$$log\ P(C_k|x) = log\ P(C_k) + \sum_{i=1}^{n} logP(x_i|C_k) * TF - IDF \quad (7)$$

In addition to probabilistic approaches like those used in Multinomial Naïve Bayes, there are other classification methods that rely on the concept of distance between data, such as K-Nearest Neighbor (KNN). This algorithm does not explicitly build a model, but instead determines the class of a data point based on its proximity to the nearest training data. Therefore, to enhance the understanding of comparing the performance of classification algorithms in sentiment analysis, the following discussion will explain the principles and application of K-Nearest Neighbor.

2.6 K-Nearest Neighbor

K-Nearest Neighbors (KNN) classification is a simple and commonly used non-parametric classification algorithm. The KNN algorithm works by storing all the training data along with their respective labels (classes). When a new data point needs to be classified, KNN does not build a model beforehand, but instead compares the new data with all existing training data based on the distance between them. This distance is typically calculated using the Euclidean Distance formula [20].

$$d(x, y) = \sum_{i=1}^{m} |x_i - y_i| \quad (8)$$

Information:
d(x,y)= distance
xi= training data

2.7 Confusion Matrix

Confusion Matrix is one of the commonly used methods for evaluating models in data mining algorithms by predicting the correctness of an object classification. The values generated through the Confusion Matrix method serve as evaluations as follows [21]:

1. Accuracy**:** The percentage of data records that are correctly classified (predicted) by the algorithm. Formula:

   (TP + TN) / Total data = Accuracy (9)

2. Precision**:** The percentage ratio of correctly predicted positive cases compared to the total predicted positive cases. Formula:

   (TP) / (TP + FP) = Precision (10)

3. Recall**:** The percentage ratio of correctly predicted positive cases compared to the total actual positive data. Formula:

   (TP) / (TP + FN) = Recall (11)

4. Misclassification (Error) Rate**:** The percentage of data records that are incorrectly classified (predicted) by the algorithm. Formula:

   (FP + FN) / Total data = Misclassification Rate (12)

2.8 Receiver Operating Characteristic (ROC) Curve

The Receiver Operating Characteristic (ROC) curve is a method for evaluating the performance of classification models, particularly in machine learning and statistics. This curve is used to describe the model's ability to differentiate between positive and negative classes at various threshold values.The Y-axis of the ROC curve represents TPR, while the X-axis represents FPR. A good model will have a curve bending toward the top-left of the graph, indicating high TPR and low FPR. An important measure in ROC is the Area Under the Curve (AUC), which shows how well the model can distinguish between classes (Nur & Oktora, 2020). An AUC of 1.0 indicates a perfect model, ≥ 0.9 indicates very good performance, between 0.7 and 0.9 indicates good performance, and ≤ 0.5 suggests the model is no better than random guessing [22].

2.9 Research Steps

Figure 1 shows the flow of the research methodology used to achieve the research analysis objectives.

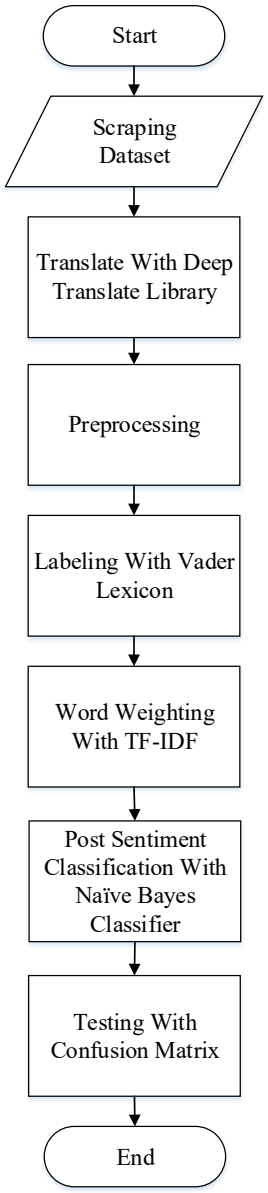

Figure 1. Example using picture

Based on Figure 1, the stages of the analysis process include:

1. Scraping Dataset.

The dataset scraping process uses the Tweepy library, focusing on posts related to the sentiment of the

increase in Value Added Tax (VAT) with the hashtags #TolakPPN12Persen, #PPN12Persen, and #PPN MemperkuatEkonomi.

2. Translate with the Deep Translate Library.
   Next, the dataset is translated into English using the Deep Translate library because, for labeling with the Vader Lexicon, the post dataset must be in English. This is because the Vader Lexicon is a sentiment dictionary specifically designed to analyze text in English.
3. Preprocessing.
   Next, the translated post dataset undergoes pre-processing, which consists of 8 stages: removing mentions, deleting URLs, removing colons, eliminating hashtags, performing case folding (converting text to lowercase), removing punctuation, eliminating extra spaces, normalizing special characters, and removing stopwords.
4. Word Weighting with TF-IDF.
   After sentiment labeling on the post dataset using the Vader Lexicon, word weighting is performed using TF-IDF to enable sentiment classification using the NBC and KNN algorithms.
5. Sentiment Classification of Posts.
   Sentiment classification of posts is performed using two algorithms:

a. Naïve Bayes Classifier.
   Model: Multinomial Naive Bayes classifier, suitable for discrete features such as TF-IDF word counts.
   Input Features: TF-IDF vectorized data, treated as frequency-like features.
   Training Data: TF-IDF vectors generated from the training subset of preprocessed posts.
   Testing Data: TF-IDF vectors generated from the testing subset of preprocessed posts.Assumptions: Features are conditionally independent given the class.
   Prediction: Probability estimation for each class, choosing the class with the highest posterior probability.

b. K-Nearest Neighbor.
   Number of Neighbors (k): 1 until 3
   The K-NN classifier uses the nearest neighbor approach with k=1, meaning the classification decision is based on the single closest training sample in the TF-IDF vector space.
   Input Features: TF-IDF vectorized representation of the preprocessed text data.
   Distance Metric: Default Euclidean distance (used implicitly by scikit-learn's KNeighborsClassifier).
   Training Data: Vectorized posts from the training folds.
   Testing Data: Vectorized posts from the testing folds.
   Prediction: The model predicts the sentiment label based on the nearest neighbor's label.

6. Testing with Confusion Matrix.
   After the sentiment classification of posts is completed, the performance of the classification models (Naïve Bayes & KNN) is tested using the Confusion Matrix.

3. RESULT AND DISCUSSION

3.1 Model Development Results

The model developed uses two algorithms, namely Naïve Bayes Classifier and K-Nearest Neighbor, with the help of the Vader Lexicon as a sentiment analysis tool. The built model was then integrated into a website, allowing users to access and view the sentiment classification results directly. The following shows the display of the algorithm comparison results on the developed website, as shown in Figure 2.

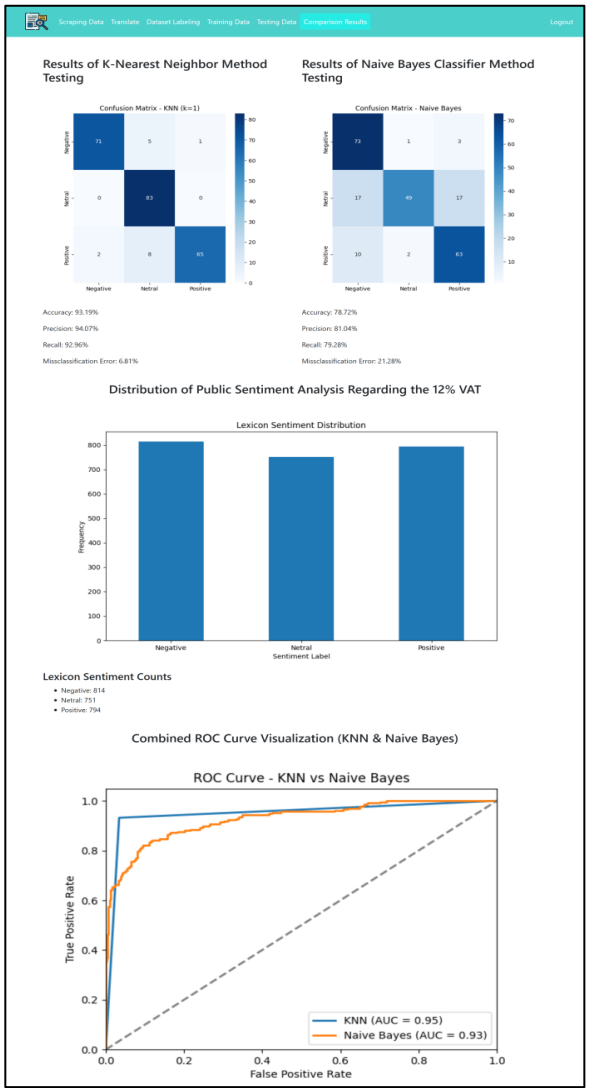

Figure 2. One of the website displays is the Algorithm Comparison Results page.

3.2 Testing Results

The testing was conducted using the Naive Bayes Classifier and K-Nearest Neighbor algorithms for sentiment analysis on social media platform X with the Vader Lexicon. The dataset consists of 2,359

posts, with a 90:10 split, where 90% (2,124 posts) was used for training data and 10% (235 posts) for testing data. The results of the testing are as follows:

1. Confusion Matrix Testing.

   The Confusion Matrix testing aims to evaluate the performance of the classification models (Naïve Bayes & KNN). Figure 3 shows the structure of the Confusion Matrix Plot for the Naïve Bayes Classifier algorithm.

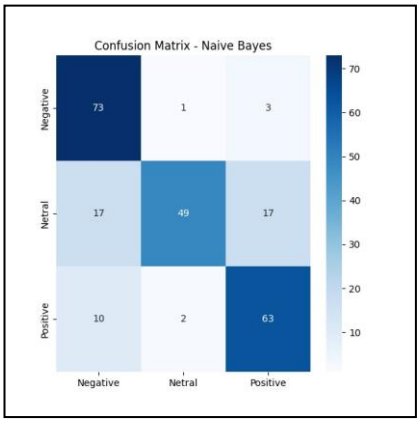

Figure 3. Confusion Matrix Plot of the Naïve Bayes Classifier Algorithm

Based on the Confusion Matrix testing results for the Naïve Bayes Classifier algorithm, an accuracy of 78.72%, precision of 81.04%, recall of 79.28%, and a misclassification error of 21.28% were obtained. Figure 4 shows the structure of the Confusion Matrix Plot for the K-Nearest Neighbor algorithm.

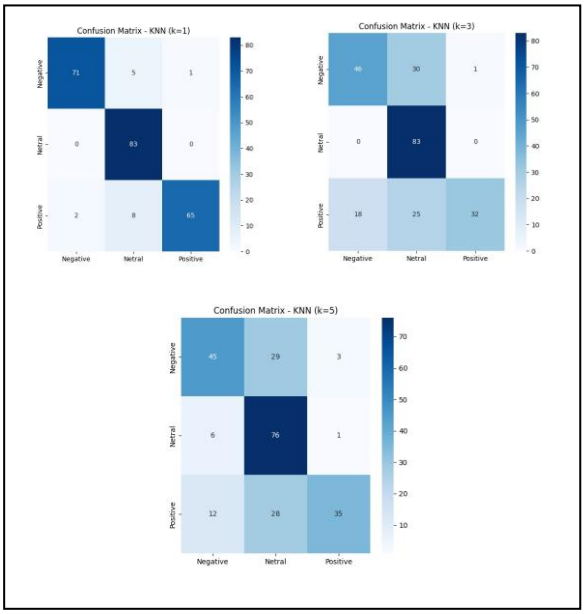

Figure 4. Confusion Matrix Plot of the K-Nearest Neighbor Algorithm

The summary of the performance testing results of the K-Nearest Neighbor algorithm in sentiment analysis on social media platform X is shown in Table 1.

Table 1. Summary of Testing Results for the K-Nearest Neighbor Algorithm

k-value	Accuracy (%)	Precision (%)	Recall (%)	Missclassification Error (%)
1	93.19	94.07	92.96	6.81
3	68.51	76.33	67.47	31.49
5	66.38	72.77	65.56	33.62

In Table 1, it can be seen that the value of k=1 has the best performance and is therefore used as the reference in this study. The significant difference in performance when varying the value of k can be explained by the nature of the K-NN algorithm. A smaller k, such as k=1, considers only the closest neighbor, allowing the model to capture very specific local patterns in the data, which leads to higher accuracy and precision. However, as k increases, the algorithm averages over more neighbors, which can smooth out noise but also causes the model to lose sensitivity to fine-grained distinctions, resulting in a decline in performance. This effect shows that selecting an appropriate k is crucial to balance between sensitivity to local patterns and generalization to the broader data distribution.

2. Sentiment Analysis Distribution Testing Results.

The sentiment analysis distribution testing is presented in the form of a bar chart to provide a clear visualization of the number of data points in each sentiment category, namely negative, neutral, and positive. This bar chart helps in illustrating the proportion of class distribution in the dataset or the results of the model's classification. Figure 5 shows the sentiment analysis distribution bar chart using the Vader Lexicon tools.

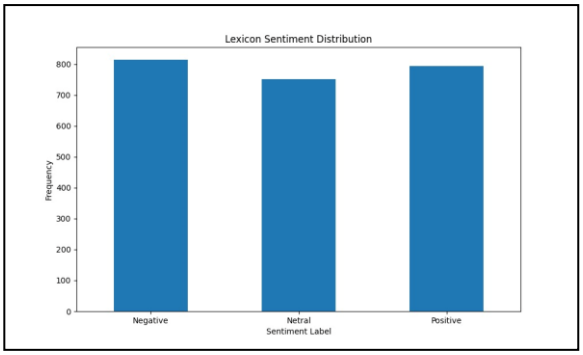

Figure 5. Sentiment Analysis Distribution Testing Results

Based on the bar chart in Figure 5, it can be seen that the number of negative sentiment analyses is 814 posts, neutral is 751 posts, and positive is 794 posts. It can be concluded that the public still holds a negative opinion regarding the sentiment about the increase in Value Added Tax (VAT).

3. ROC Curve Testing Results.

The results of the ROC (Receiver Operating Characteristic) curve testing are presented in the form of a graph that illustrates the performance of the classification model in distinguishing between classes. The ROC curve shows the relationship between True Positive Rate (Recall) and False

Positive Rate for each decision threshold used. Figure 6 shows the results of the ROC curve testing.

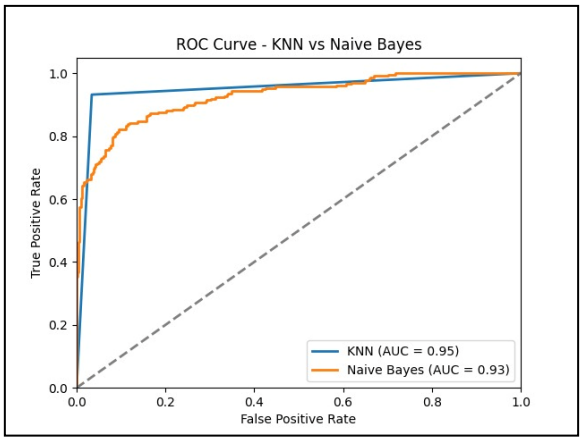

Figure 6. ROC Curve Testing Results

The testing results using the ROC curve in Figure 6 show that the K-Nearest Neighbor (KNN) algorithm has an AUC value of 0.95, while the Naive Bayes Classifier (NBC) algorithm has an AUC value of 0.93. An AUC value close to 1 indicates that both algorithms have very good classification performance in distinguishing between classes. However, with a slightly higher AUC value, it can be concluded that the KNN algorithm has better classification ability than the Naive Bayes Classifier in this test.

3.3 Discussion

The first test conducted was using the Confusion Matrix to measure the performance of the model. Based on the test results, the Naive Bayes Classifier algorithm achieved an accuracy of 78.72%, with an average precision of 81.04%, recall of 79.28%, and a misclassification error of 21.28%. This accuracy indicates that the Naive Bayes Classifier model can classify sentiment data well, although there are some errors in predicting sentiment, especially in the neutral class.

Meanwhile, the K-Nearest Neighbor algorithm with k=1 showed better performance, with an accuracy of 93.19%, precision of 94.07%, recall of 92.96%, and a misclassification error of 6.81%. This shows that KNN with k=1 can predict sentiment very well, even more accurately than the Naive Bayes Classifier. When tested with k=3 and k=5, the performance of KNN started to decline, with accuracies of 68.51% and 66.38%, along with an increased misclassification error rate. Therefore, it can be concluded that k=1 provides the best results in this test.

For the Naive Bayes Classifier algorithm, the highest precision was found in the neutral class, at 94.23%, indicating that the model is very good at classifying neutral sentiment. However, the recall for the neutral class was only 59.04%, meaning the model struggles to identify all neutral data. On the other hand, the negative class has a very high recall (94.81%), but lower precision (73.00%), indicating that there are some errors in classifying negative data as positive or neutral.

K-Nearest Neighbor with k=1, on the other hand, shows a more balanced precision and recall. Precision for the negative, neutral, and positive classes are 97.26%, 86.46%, and 98.48%, respectively. Meanwhile, recall for the negative and positive classes is also quite high (92.21% and 86.67%), but the neutral class has perfect recall (100%) with minimal misclassification.

Based on the results obtained from testing the Naive Bayes Classifier (NBC) and K-Nearest Neighbor (KNN) algorithms, it can be concluded that K-Nearest Neighbor with k=1 has better performance than Naive Bayes Classifier in terms of accuracy, precision, and recall. These results are consistent with the research conducted by Abdillah et al. (2024), which compared the Naïve Bayes and K-Nearest Neighbor methods in sentiment analysis of Zenius app users. The study found that the accuracy of Naive Bayes was 88.41%, while KNN reached 100% [23]. Additionally, these results align with the research conducted by Elfansyah et al. (2024), who compared the K-Nearest Neighbor and Naïve Bayes methods in sentiment analysis of Dana e-wallet app users using TF-IDF feature extraction. The study found that the KNN and Naive Bayes methods showed different accuracy based on data label sources. For the lexicon-labeled data, KNN achieved 78% accuracy and Naive Bayes 74% [24].

The performance difference between K-Nearest Neighbor and Naive Bayes in this sentiment analysis is due to K-NN's ability to capture local patterns and complex dependencies among features [25], while Naive Bayes is limited by the feature independence assumption that is rarely met in text data [26]. Therefore, K-NN with k=1 produces more accurate results in TF-IDF-based sentiment classification compared to Naive Bayes.

The sentiment analysis distribution conducted using Vader Lexicon provides a clear picture of the public's sentiment towards the issue being studied, in this case, the increase in Value Added Tax (VAT). From the sentiment analysis distribution test results, it can be seen that the majority of the public tend to have a negative sentiment (814 posts), followed by positive sentiment (794 posts) and neutral sentiment (751 posts). This indicates that the majority of people are dissatisfied with the VAT policy, although a small number have positive or neutral views on the policy.

Evaluation using the ROC curve provides a deeper perspective on the model's ability to distinguish between sentiment classes. The AUC value of 0.95 for K-Nearest Neighbor indicates that this model is very good at distinguishing between sentiment classes. Meanwhile, Naive Bayes achieved an AUC of 0.93, which also indicates good classification performance, although slightly lower than KNN. Overall, both algorithms perform very

well in sentiment classification, but KNN has a slight edge in distinguishing between sentiment classes.

4. CONCLUSION

The results of the testing indicate that the K-Nearest Neighbor (KNN) algorithm with k=1 outperforms Naïve Bayes in sentiment classification, achieving an accuracy of 93.19%, precision of 94.07%, recall of 92.96%, and a misclassification error of 6.81%. In comparison, Naïve Bayes recorded an accuracy of 88.29%, precision of 87.43%, recall of 86.67%, and a misclassification error of 11.71%. The ROC curve evaluation further supports KNN's superior performance with an AUC of 0.95, slightly higher than Naïve Bayes, which had an AUC of 0.93. Both algorithms showed strong classification abilities, but KNN was more accurate and efficient in distinguishing between sentiment classes. The use of the Vader Lexicon tool for automatic sentiment labeling proved to be effective in speeding up the labeling process and enhancing preprocessing efficiency, without compromising the classification accuracy of either KNN or Naïve Bayes. The sentiment distribution revealed that negative sentiment dominated, with 814 posts labeled as negative, followed by 794 positive posts and 751 neutral posts. This suggests that the majority of the public expresses dissatisfaction with the VAT increase policy.

However, this study has several limitations that should be addressed in future research. First, the data scraping was limited to 100 posts per month due to developer platform constraints, potentially affecting the representativeness of the dataset. Second, only two classification algorithms (KNN and Naïve Bayes) were explored, while incorporating other models like Random Forest or XGBoost may improve performance. Lastly, the analysis was confined to a single topic, so testing the models on different topics or datasets is necessary to assess the generalizability and robustness of the sentiment classification approach.

5. REFERENCE

[1] F. A. Adiyatma, S. Alam, and M. A. Komara, "Analisis Sentimen Masyarakat di Platform X Terhadap Penggunaan Bansos Untuk Memenangkan Salah Satu Capres Tertentu di Pilpres 2024 Menggunakan Metode Naive Bayes Classifier," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 5, pp. 9941–9947, 2024.

[2] L. Nursinggah and T. Mufizar, "Analisis Sentimen Pengguna Aplikasi X Terhadap Program Makan Siang Gratis Dengan Metode Naive Bayes Classifier," *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 12, no. 3, pp. 1615–1622, 2024.

[3] T. Prasetyo, H. Zakaria, and P. Wiliantoro, "Analisis Layanan Pelanggan PT PLN Berdasarkan Media Sosial Twitter Dengan Menggunakan Metode Naïve Bayes Classifier," *OKTAL J. Ilmu Komput. dan Sains*, vol. 1, no. 6, pp. 573–582, 2022, [Online]. Available: https://journal.mediapublikasi.id/index.php/oktal.

[4] T. A. Siddiq and M. Ikhsan, "Analisis Sentimen X Terhadap Pemilihan Presiden Indonesia 2024 dengan Metode K-Nearest Neighbor," *J. Comput. Syst. Informatics*, vol. 5, no. 4, pp. 1064–1078, 2024, doi: 10.47065/josyc.v5i4.5802.

[5] P. A. Prastyo, Berlilana, and I. Tahyudin, "Analisis Sentimen dan Pemodelan Topik pada Ulasan Pengguna Aplikasi myIM3 Menggunakan Support Vector Machine dan Latent Dirichlet Allocation," *Build. Informatics, Technol. Sci.*, vol. 6, no. 3, pp. 1618–1626, 2024, doi: 10.47065/bits.v6i3.6268.

[6] M. I. Maulana, E. Budianita, M. Fikry, and F. Yanto, "Klasifikasi Sentiment Ulasan Aplikasi Sausage Man Menggunakan VADER Lexicon dan Naïve Bayes Classifier," *J. Sist. Komput. dan Inform.*, vol. 4, no. 3, pp. 485–492, 2023, doi: 10.30865/json.v4i3.5854.

[7] M. Iqbal, A. D. Wiranata, R. Suwito, and R. F. Ananda, "Perbandingan Algoritma Naïve Bayes, KNN, dan Decision Tree terhadap Ulasan Aplikasi Threads dan Twitter," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 3, pp. 1799–1807, 2023, doi: 10.30865/klik.v4i3.1402.

[8] Muttaqin *et al.*, *Implementasi Artificial Intelligence (AI) Dalam Kehidupan*. Medan: Yayasan Kita Menulis, 2023.

[9] D. Purnamasari *et al.*, *Pengantar Metode Analisis Sentimen*. Depok: Gunadarma, 2023.

[10] Y. Asri, W. N. Suliyanti, and D. Kuswardani, "Analisis Sentimen Opini Pelanggan Aplikasi Pln Mobile Menggunakan Metode Vader Lexicon & Naive Bayes," in *Prosiding Seminar Nasional Energi, Kelistrikan, Teknik dan Informatika*, 2022, vol. 3.

[11] F. Amaliah and I. K. D. Nuryana, "Perbandingan Akurasi Metode Lexicon Based Dan Naive Bayes Classifier Pada Analisis Sentimen Pendapat Masyarakat Terhadap Aplikasi Investasi Pada Media Twitter," *J. Informatics Comput. Sci.*, vol. 3, no. 3, pp. 384–393, 2022, doi: 10.26740/jinacs.v3n03.p384-393.

[12] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentimen Analysis of Social Media Text," in *Eighth International AAAI Conference on Weblogs and Social Media*, 2014, pp. 216–225, [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[13] V. Nurcahyawati and Z. Mustaffa, "Vader Lexicon and Support Vector Machine Algorithm to Detect Customer Sentiment Orientation," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 1, pp. 108–118, 2023, doi: 10.20473/jisebi.9.1.108-118.

[14] I. S. Wibowo, A. Witanti, and I. Susilawati, "Keyword Extraction Judul Berita Online Di Indonesia Menggunakan Metode TF-IDF," *J. Tek. Inform. dan Sist. Inf.*, vol. 11, no. 1, pp. 99–111, 2024, [Online]. Available: http://jurnal.mdp.ac.id.

[15] M. D. Afandi, A. Homaidi, A. Ghofur, and A. Zubairi, "Penerapan Information Retrieval dalam Sistem Analisis Kemiripan Proposal Skripsi menggunakan Cosine Similarity," *Swabumi*, vol. 12, no. 1, pp. 39–46, 2024.

[16] D. Tuhenay and E. Mailoa, "Perbandingan Klasifikasi Bahasa Menggunakan Metode Naïve Bayes Classifier (NBC) Dan Support Vector Machine (SVM)," *JIKO (Jurnal Inform. dan Komputer)*, vol. 4, no. 2, pp. 105–111, 2021, doi: 10.33387/jiko.v4i2.2958.

[17] A. Aninditya, M. A. Hasibuan, and E. Sutoyo, "Text Mining Approach Using TF-IDF and Naive Bayes for Classification of Exam Questions Based on Cognitive Level of Bloom's Taxonomy," in *Proceedings - 2019 IEEE International Conference on Internet of Things and Intelligence System, IoTaIS 2019*, 2019, no. 11, pp. 112–117, doi: 10.1109/IoTaIS47347.2019.8980428.

[18] Yuyun, N. Hidayah, and S. Sahibu, "Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 820–826, 2021, doi: 10.29207/resti.v5i4.3146.

[19] A. S. Wilindia, M. Dasuki, and N. Q. Fitriyah, "Implementasi Algoritma Multinomial Naïve Bayes Untuk Analisis Sentimen Twitter Terhadap Kebijakan Merdeka Belajar," *J. Smart Teknol.*, vol. 1, no. 1, pp. 100–102, 2021.

[20] P. Putra, A. M. H. Pardede, and S. Syahputra, "Analisis Metode K-Nearest Neighbour (Knn) Dalam Klasifikasi Data Iris Bunga," *J. Tek. Inform. Kaputama*, vol. 6, no. 1, pp. 297–305, 2022.

[21] M. A. Muslim *et al.*, *Data Mining Algoritma C4.5*. 2019.

[22] M. R. F. Nur and S. I. Oktora, "Analisis Kurva Roc Pada Model Logit Dalam Pemodelan Determinan Lansia Bekerja Di Kawasan Timur Indonesia," *Indones. J. Stat. Its Appl.*, vol. 4, no. 1, pp. 116–135, 2020, doi: 10.29244/ijsa.v4i1.524.

[23] T. Abdillah, U. Khaira, and B. F. Hutabarat, "Komparasi Metode Naive Bayes dan K-Nearest Neighbors Terhadap Analisis Sentimen Pengguna Aplikasi Zenius," *J. Process.*, vol. 19, no. 1, pp. 32–44, 2024, doi: 10.33998/processor.2024.19.1.1596.

[24] M. R. Elfansyah, Rudiman, and F. Yulianto, "Perbandingan Metode K–Nearest Neighbor (Knn) Dan Naive Bayes Terhadap Analisis Sentimen Pada Pengguna E-Wallet Aplikasi Dana Menggunakan Fitur Ekstraksi Tf-Idf," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 8, no. 2, pp. 139–159, 2024.

[25] F. R. Irawan, A. Jazuli, and T. Khotimah, "Analisis Sentimen Terhadap Pengguna Gojek Menggunakan Metode K-Nearset Neighbors," *JIKO (Jurnal Inform. dan Komputer)*, vol. 5, no. 1, pp. 62–68, 2022, doi: 10.33387/jiko.

[26] G. H. Kusuma, I. Permana, F. N. Salisah, M. Afdal, M. Jazman, and A. Marsal, "Pendekatan Machine Learning : Analisis Sentimen Masyarakat Terhadap Kendaraan Listrik Pada Sosial Media X," *J. Sist. Inf.*, vol. 9, no. 2, pp. 65–76, 2023.