International Journal of Information Technology and Computer Science Applications (IJITCSA)
p-ISSN: 2964-3139 e-ISSN: 2985-5330
Vol. 04, No. 01, page 11 - 20
Submitted 15/7/2025; Accepted 21/2/2026; Published 25/2/2026

Public Sentiment Analysis on the Service Quality of PT PLN on X Using Naïve Bayes and K-Nearest Neighbor Algorithms.

[1]Nurul Zahra [2]Wowon Priatna[2] Tyastuti Sri Lestari
[1]Informatics Universitas Bhyangkara Jakarta Raya, Bekasi, INDONESIA
[2]Information Departement, Universitas Bhyangkara Jakarta Raya, Bekasi, INDONESIA
e-mail : [1]202110715022@mhs.ubharajaya.ac.id, [2]wowon.periatna@dsn.ubharajaya.ac.id
[2]tyas@ ubharajaya.ac.id

Corresponding Autor: Wowon Priatna

Abstract
Quality services since electricity is a primary public need. However, numerous complaints still highlight PLN's lack of responsiveness, especially on the X platform (formerly Twitter). This study aims to analyze public sentiment toward PLN's service quality expressed on X and compare the performance of the Naïve Bayes and K-Nearest Neighbor (KNN) algorithms in classifying sentiments into positive, negative, and neutral categories. The research employs the Knowledge Discovery in Databases (KDD) approach, involving data collection through tweet scraping using Tweet-Harvest, preprocessing (case folding, tokenizing, filtering, stemming), transformation with TF-IDF weighting, and data mining using Naïve Bayes and KNN. Evaluation through a confusion matrix shows that Naïve Bayes achieved an accuracy of 87%, outperforming KNN with an accuracy of 86%. These findings provide insights for PLN to better understand public perception and serve as a reference for future sentiment analysis research using machine learning.

Keywords— Sentiment Analysis, Naïve Bayes, K-Nearest Neighbor, PLN Service Quality.

1 Introduction

PT PLN (Persero), as a state-owned enterprise (SOE) in the electricity sector that serves communities throughout the Indonesian archipelago, is committed to providing the best electric power services that meet internationally accepted standards, striving to achieve this by relying on the capabilities of all its personnel. The quality of services provided by PT PLN often attracts public attention, given that electricity is a basic need that significantly impacts the comfort and efficiency of people's daily activities. However, many people continue to complain about the service, citing issues such as a lack of responsiveness, inadequate handling of feedback on the X platform, and insufficient efficiency in addressing public comments, criticisms, and suggestions.

To ensure that customer satisfaction remains well maintained, it is essential to understand public opinion regarding the services provided by PT PLN. Such opinions may take the form of complaints (negative sentiment), support (positive sentiment), or reports and suggestions from the community (neutral sentiment)[1]. The social media platform X (formerly Twitter) serves as an effective medium to collect public opinion directly and in real-time, thus becoming a relevant data source for sentiment analysis[2]. This study focuses on analyzing public sentiment related to the quality of PT PLN's services as expressed on the X (Twitter) platform[1]. In this research, two text classification methods are utilized, namely Naïve Bayes and K-Nearest Neighbors (KNN), to categorize public opinions into positive, negative, or neutral sentiments[3]. Naïve Bayes is known for its efficiency in text analysis, as it can process large datasets quickly by calculating the probability that a review carries a certain sentiment based on the distribution of words found in the text[4]. However, this method has a limitation in assuming feature independence, which does not always reflect the actual conditions of the data. Meanwhile, KNN offers advantages in its flexibility to process non-linear data without requiring assumptions of feature independence. Nonetheless, KNN tends to demand more computational resources and is more susceptible to issues arising from high-dimensional data[4].

Prior to this research, several studies addressed similar themes with different approaches. For example, a study on public sentiment toward COVID-19 on Twitter employed the Naïve Bayes and KNN algorithms. Out of 1,098 collected tweets, the results indicated that Naïve Bayes achieved a higher accuracy (63.21%) compared to KNN (58.10%)[5]. Another study focusing on reviews of the PLN Mobile application also applied Naïve Bayes Classifier (NBC) and K-Nearest Neighbor (KNN) for sentiment analysis, showing that NBC was more accurate (77.69%) than KNN, with most users expressing positive experiences with the application despite facing some issues in purchasing electricity tokens[4].

Although previous studies have discussed sentiment analysis in various contexts, including in the energy sector through the PLN Mobile application, there has been no research specifically exploring public opinions on the overall quality of PLN's services. In fact, a broader understanding of public perceptions regarding electricity services in Indonesia is crucial for PLN to enhance service quality and design policies that are more responsive to customer needs. Therefore, this research aims to fill this gap by analyzing public opinions related to PLN's services using data from the X platform, while also comparing the effectiveness of the Naïve Bayes and KNN methods in sentiment classification.

2 Research methods

The process in this research involves the application of the knowledge Discovery in Databases (KDD) method. Knowledge Discovery in Databases (KDD) is a comprehensive process aimed at uncovering valuable information from data sets, where Data Mining serves as a crucial stage within this process [6]. KDD is a structured analytical process designed to extract valid, novel, and relevant information, as well as to identify hidden patterns from large and complex data sets [7].

2.1 *Data Selection*

Data collection was carried out through a crawling process on Twitter to extract comments related to the quality of PT PLN's services on the X social media platform. After the data was gathered, sampling was conducted to ensure the relevance and quality of the data to be analyzed. In this study, two text classification methods, namely Naive Bayes and K-Nearest Neighbors (KNN), were employed to categorize public opinions into positive, negative, or neutral sentiments[3].

2.2 *Preprocessing*

Meanwhile, another definition states that text preprocessing is an initial process in which the system filters or selects relevant data from each document before proceeding to further analysis [8]. The preprocessing stage consists of several processes:

- Case folding: Converting all characters to lowercase.
- Cleaning: Removing special characters, numbers, emojis, and irrelevant symbols.
- Tokenization: Splitting sentences into individual words (tokens).
- Normalization: Transforming informal words, slang, or abbreviations into their standard forms.
- Stopword removal: Eliminating common words that carry little importance for classification.
- Stemming: Reducing words to their root forms to simplify the data.

2.3 *Transformation*

The TF-IDF weighting is determined by the frequency of a word appearing within a sentence (TF) and its frequency across all documents (IDF). A term will have a high weight if it appears frequently within a single document, but its weight decreases if the term is commonly found across many other documents [9]. The TF-IDF is formulated by the following equation.

$$\mathrm{TF-IDF}\ (\mathrm{t_k}) = \log \frac{\mathrm{N}}{\mathrm{df(t_k)}} \tag{1}$$

TF-IDF is used to assess the degree of similarity between documents in a given context by calculating the weight of each word based on how frequently it appears in a particular document and how rarely it is found across the entire collection of documents.

2.4 *Data Mining*

Data Mining is the process of analyzing a collection of data, particularly large-scale data, to identify significant patterns or relationships and to draw conclusions that were previously unknown. Its primary goal is to generate new information that is relevant, comprehensible, and beneficial to data owners by employing state-of-the-art methods [10]. Data Mining comprises a set of techniques that are automatically used to explore and uncover complex hidden relationships within large datasets. Such data are generally structured in tables and are often implemented in relational data management technologies [11].

2.4.1 *Naïve Bayes*

Naïve Bayes is a simple classification method that involves calculating the count of data and the combination of values within a dataset to determine the probability of that data. This method is recognized as an efficient approach in machine learning and is frequently used to evaluate the performance of a classification model [12].

$$P(Y/X) = \frac{P(Y/X) \times P(Y)}{P(X)} \tag{2}$$

In this equation, P(HIX) represents the probability of hypothesis H (sentiment class) based on data X (word features). The term P(XIH) refers to the probability of observing X if H is true, P(H) indicates the prior probability of H, and P(X) is the overall probability of X. This calculation helps determine the sentiment class with the highest probability based on the word patterns found in the comments.

2.4.2 *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) is an algorithm used to classify an object based on the training data that has the highest similarity or proximity to that object. Although considered relatively simple, the KNN algorithm has proven to be quite effective for clustering, especially with text-based data.

$$\text{Dist(xi,x2)}= \sqrt{\textstyle\sum_{i=1}^{n}(X_{1i} - X_{2i})} \tag{3}$$

The formula is used to calculate the Euclidean distance, which is the shortest distance between two points in an n-dimensional space by summing the squared differences of each attribute and then taking the square root.

2.5 *Evaliation*

At the evaluation stage, the researcher conducted testing and assessment of the models to determine the performance of the Naive Bayes and K-Nearest Neighbor algorithms. The results of the classification process are presented in the form of a confusion matrix. A detailed explanation of the evaluation for each algorithm will be provided in the following subsections.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

In this equation, accuracy measures the proportion of all predictions that are correct. Precision is calculated using the formula:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{5}$$

This metric indicates how many of the predictions labeled as positive are truly positive.

Recall is computed as:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{6}$$

Recall measures how many actual positive cases are correctly identified by the model.

The F1-Score, which balances precision and recall, is determined by the following formula:

$$F = \frac{2\ (\text{precision x Recall})}{\text{precision + Recall}} \tag{7}$$

Precision Recall F1-Score 2 x Precision+Recall.

This score provides a single metric that reflects the model's overall effectiveness in handling both precision and recall.

2.6 *Knowlage Presentation*

At this stage, the data will be visualized using a word cloud and percentage charts for positive, negative, and neutral opinions to make it easier for readers to understand the research findings. The most frequently occurring words in the text will be displayed visually through the word cloud.

3 Results and Discussion

3.1 *Data Selection*

The initial stage in the Knowledge Discovery in Databases (KDD) process for analyzing user sentiment on the Twitter platform is data selection. At this stage, data is collected (crawled) from Twitter using the Python programming language. Once the data has been successfully obtained, the next step is to label the data in order to identify the sentiment categories contained within it. In general, the purpose of this data selection stage is to select and acquire data relevant to the focus of the research so that it can serve as a foundation for further analysis.

3.2 *Processing*

At this stage, the labeled data in processed through several text preprocessing steps, which are shown in Table 1.

Tabel 1Case Folding Results Data

Review Before Case Folding	**Review After Case Folding**
PLN sangat cepat tanggap layananya ok	pln sangat cepat tanggap layananya ok
Kerjaan PLN cuma bisa minta maaf saja Tanpa mau memperbaiki diri dan meningkatkan mutu pelayanan	kerjaan pln cuma bisa minta maaf saja tanpa mau memperbaiki diri dan meningkatkan mutu pelayanan
Gini amat ya pelayanan PLN sekarang sumpah kesel banget	gini amat ya pelayanan pln sekarang sumpah kesel banget

All characters are converted to lowercase to standardize the data and avoid differences caused only by capitalization. After this step, cleansing is performed as shown in the table 2.

Tabel 2 Tokenizing

Dokumen data uji	**Tahapan Tokenizing**
Input	output
pln sangat cepat tanggap layananya ok	["pln", "sangat", "cepat", "tanggap", "layananya", "ok"]
kerjaan pln cuma bisa minta maaf saja tanpa mau memperbaiki diri dan meningkatkan mutu pelayanan	["kerjaan", "pln", "cuma", "bisa", "minta", "maaf", "saja", "tanpa", "mau", "memperbaiki", "diri", "dan", "meningkatkan", "mutu", "pelayanan"]
gini amat ya pelayanan pln sekarang sumpah kesel banget	["gini", "amat", "ya", "pelayanan", "pln", "sekarang", "sumpah", "kesel", "banget"]

Tokenizing splits the text into separate words (tokens) to make it easier for further analysis. After separating

Tabel 3 Filtering Result Data

Dokumen data uji	**Tahapan case filtering**
Input	**output**
["pln", "sangat", "cepat", "tanggap", "layananya", "ok"]	['pln', 'cepat', 'tanggap', 'layananya', 'ok'],
["kerjaan", "pln", "cuma", "bisa", "minta", "maaf", "saja", "tanpa", "mau", "memperbaiki", "diri", "dan", "meningkatkan", "mutu", "pelayanan"]	['kerjaan', 'pln', 'minta', 'maaf', 'memperbaiki', 'diri', 'meningkatkan', 'mutu', 'pelayanan'],

Dokumen data uji	Tahapan case filtering
Input	**output**
["gini", "amat", "ya", "pelayanan", "pln", "sekarang", "sumpah", "kesel", "banget"]	['gini', 'pelayanan', 'pln', 'sekarang', 'sumpah', 'kesel', 'banget']

Stopword removal eliminates common words that carry little significance in analysis, such as "untuk "di," or "ya." Finally, stemming is performed as displayed in the table

Tabel 4 Stemmig Result Data

Dokumen data uji	Tahapan Stemming
Input	**output**
['pln', 'cepat', 'tanggap', 'layananya', 'ok'],	pln cepat tanggap layan ok
['kerjaan', 'pln', 'minta', 'maaf', 'memperbaiki', 'diri', 'meningkatkan', 'mutu', 'pelayanan'],	kerja pln minta maaf perbaiki diri tingkat mutu layan
['gini', 'pelayanan', 'pln', 'sekarang', 'sumpah', 'kesel', 'banget']	gini layan pln sekarang sumpah kesal banget

Stemming reduces words to their root forms by removing affixes, making the word forms more consistent.

3.3 *Transformation*

In the data mining stage, the dataset was divided into training and testing sets with an 80:20 ratio, resulting in 3271 training data points and 653 testing data points. This partitioning process was carried out using the train_test_split module from the sklearn.model_selection library. After the data split, TF-IDF calculations were performed to transform the textual data into numerical vectors for subsequent sentiment classification, as shown in Equation (1) and presented in the table 5.

Tabel 5 TF-IDF

Term	TF_1	TF_2	TF_3	TF_4	TF_5	TF_6	TF_7	TF_8	TF_9	TF_{10}	DF
lampu	1	1	0	0	0	0	0	0	0	0	2
cuci	1	0	0	0	0	0	0	0	0	0	1
sebal	1	0	0	0	0	0	0	0	0	0	1
puasa	0	1	0	0	0	0	0	0	0	0	1
mati	0	1	0	0	0	0	0	0	0	0	1
jam	0	1	0	0	0	0	0	0	0	0	1
kerja	0	1	0	0	0	0	0	0	0	0	1
indonesia	0	0	1	0	0	0	0	0	0	0	1
pajak	0	0	1	0	0	0	0	0	0	0	1

3.4 *Data Mining*

The subsequent stage involves applying the Naive Bayes and KNN algorithms. Samples of the training and testing data are presented in Table 6 and 7.

Tabel 6 Sample Testing Data

D	Komentar	Kategori
D1	lampu cuci sebal	negatif
D2	meningkat pelayanan	positif
D3	lapor kawasan taman sidoarjo alami gangguan listrik	netral

Tabel 7 Sample Data Test

D	Komentar	Kategori
D1	Puasa mati lampu jam kerja	negatif
D2	Meningkat pelayanan	positif

3.4.1 *Naïve Bayes*

He first stage in Naive Bayes calculations is to determine the prior probabilities for each sentiment class. These calculations can be performed using Equation (2) and are summarized in Table

Tabel 8 Prior Probability

	Positif	Netral	Negatif
Prior Probability	$\frac{7}{35} = 0.2$	$\frac{13}{35} = 0.37$	$\frac{15}{35} = 0.42$

After calculating the prior probabilities, the next step is to determine the conditional probabilities for cach word in the vocabulary with respect to every sentiment class. For instance, the probability of the word.

Probablilty of the word 'puasa'

$$P(a_{puasa}|V_{positif}) = \frac{0+1}{7+35} = \frac{1}{42} = 0.02380$$
$$P(a_{puasa}|V_{netral}) = \frac{0+1}{13+35} = \frac{1}{48} = 0.0208$$
$$P(a_{puasa}|V_{negatif}) = \frac{1+1}{15+35} = \frac{2}{50} = 0.04$$

Probablilty of the word 'mati'

$$P(a_{mati}|V_{positif}) = \frac{0+1}{7+35} = \frac{1}{42} = 0.0238$$
$$P(a_{mati}|V_{netral}) = \frac{0+1}{13+35} = \frac{1}{48} = 0.0208$$
$$P(a_{mati}|V_{negatif}) = \frac{1+1}{15+35} = \frac{2}{50} = 0.04$$

Probablilty of the word 'lampu'

$$P(a_{lampu}|V_{positif}) = \frac{0+1}{7+35} = \frac{1}{42} = 0.0238$$
$$P(a_{lampu}|V_{netral}) = \frac{0+1}{13+35} = \frac{1}{48} = 0.0208$$
$$P(a_{lampu}|V_{negatif}) = \frac{2+1}{15+35} = \frac{3}{50} = 0.06$$

Probablilty of the word 'jam'

$$P(a_{jam}|V_{positif}) = \frac{0+1}{7+35} = \frac{1}{42} = 0.0238$$
$$P(a_{jam}|V_{netral}) = \frac{0+1}{13+35} = \frac{1}{48} = 0.0208$$
$$P(a_{jam}|V_{negatif}) = \frac{1+1}{15+35} = \frac{2}{50} = 0.04$$

Probablilty of the word 'kerja'

$$P(a_{kerja}|V_{positif}) = \frac{0+1}{7+35} = \frac{1}{42} = 0.0238$$
$$P(a_{kerja}|V_{netral}) = \frac{0+1}{13+35} = \frac{1}{48} = 0.0208$$
$$P(a_{kerja}|V_{negatif}) = \frac{1+1}{15+35} = \frac{2}{50} = 0.04$$

Probablilty of the word 'meningkat"

$$P(a_{meningkat}|V_{positif}) = \frac{1+1}{7+35} = \frac{2}{42} = 0.0476$$

$$P(a_{meningkat}|V_{netral}) = \frac{0+1}{13+35} = \frac{1}{48} = 0.0208$$

$$P(a_{meningkat}|V_{negatif}) = \frac{1+1}{15+35} = \frac{2}{50} = 0.04$$

Probablilty of the word 'pelayanan"

$$P(a_{pelayanan}|V_{positif}) = \frac{1+1}{7+35} = \frac{2}{42} = 0.0476$$

$$P(a_{pelayanan}|V_{netral}) = \frac{1+1}{13+35} = \frac{1}{48} = 0.0208$$

$$P(a_{pelayanan}|V_{negatif}) = \frac{0+1}{15+35} = \frac{1}{50} = 0.02$$

After obtaining the probability likelihood values, the next step is to classify the test data. This involves performing manual calculations to classify the test data, along with the steps for calculating the classification of the test data samples.

$P(uji \mid positif)$
$= p(positif) \times P(puasa \mid positif) \times P(mati \mid positif) \times P(lampu \mid positif) \times P(jam \mid positif) \times P(kerja \mid positif)$
$= 0.2 \times 0.0238 \times 0.0238 \times 0.0238 \times 0.0238 \times 0.0238$
$= 0.00000000153$
$P(uji \mid netral)$
$= p(netral) \times \text{P(puasa | netral)} \times \text{P(mati | netral)} \times \text{P(lampu |netral)} \times \text{P(jam |netral)} \times \text{P(kerja |netral))}$
$= 0.37 \times 0.0238 \times 0.0208 \times 0.0208 \times 0.0208 \times 0.0208$
$= 0.000000001648$
P(uji | negatif)
$= \text{p(negatif)} \times \text{P(puasa | negatif)} \times \text{P(mati | negatif)} \times \text{P(lampu |negatif)} \times \text{P(jam |negatif)} \times \text{P(kerja |negatif))}$
$= 0.42 \times 0.04 \times 0.04 \times 0.06 \times 0.04 \times 0.04$
$= 0{,}0000016128$

	precision	recall	f1-score	support
negatif	0.64	0.84	0.73	320
netral	0.00	0.00	0.00	51
positif	0.70	0.58	0.63	279
accuracy			0.66	650
macro avg	0.45	0.47	0.45	650
weighted avg	0.62	0.66	0.63	650

Figure 1 Confusion Matrix Algoritma Naïve Bayes

```
from sklearn.metrics import confusion_matrix
from mlxtend.plotting import plot_confusion_matrix
import matplotlib.pyplot as plt

# Hitung confusion matrix
cm_nb = confusion_matrix(y_test, y_pred_nb)

# Plot dengan mlxtend
fig, ax = plot_confusion_matrix(
    conf_mat=cm_nb,
    figsize=(6,6),
    class_names=best_nb.classes_,
    show_normed=True  # untuk persentase
)
plt.title('Confusion Matrix - Naive Bayes')
plt.show()
```

Figure 2 Script Confusion Matrix Algoritma Naïve Bayes

3.4.2 K-Nearest Neighbor

In K-Nearest Neighbor, the initial step involves computing the Euclidean distances from the test document to each training document, as shown in Equation (3) and summarized in Table

Tabel 9 KNN

No	Komentar	Kategori	Meghitung Jarak
d1	lampu cuci sebal	Negatif	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2} = 4$
d2	indonesia pajak layanan bumn kacau parah	Negatif	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2} = 5$
d3	meningkat pelayanan	Positif	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2} = 5$
d4	baik tingkat mutu	Positif	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2} = 5$
d5	cepat layan	Positif	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2} = 5$
d6	tanggap pelayanan	Netral	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2} = 5$
d7	bulan bayar mahal rugi	Netral	$d = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2} = 5$

Using K-Nearest Neighbor with k=1, the test sentence "lucu warna iphone plus" was compared to all training documents. The closest neighbor was found to be document D1, which belongs to the Positive category.

	precision	recall	f1-score	support
negatif	0.80	0.32	0.46	25
netral	0.87	0.99	0.92	297
positif	0.78	0.38	0.51	48
accuracy			0.86	370
macro avg	0.82	0.56	0.63	370
weighted avg	0.85	0.86	0.84	370

Figure 3 Confusion Matrix Algoritma K-Nearest Neighbor

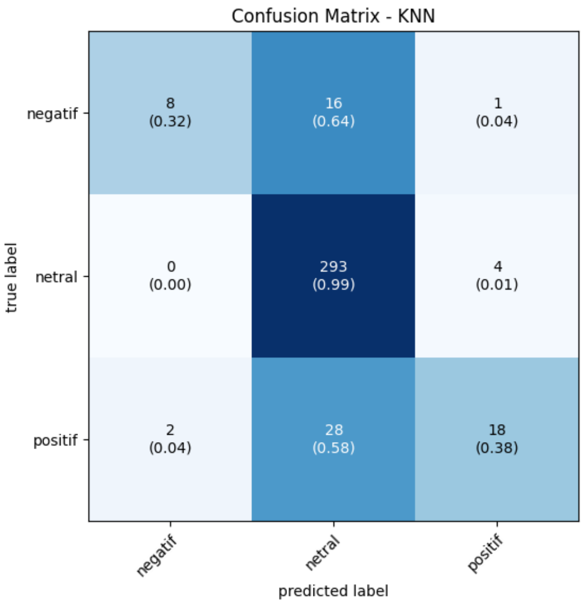

Figure 4 Plot Confunsion Matrix

```
from sklearn.metrics import confusion_matrix
from mlxtend.plotting import plot_confusion_matrix
import matplotlib.pyplot as plt

# Hitung confusion matrix
cm_knn = confusion_matrix(y_test, y_pred_knn)

# Plot confusion matrix
fig, ax = plot_confusion_matrix(
    conf_mat=cm_knn,
    figsize=(6,6),
    class_names=best_knn.classes_,
    show_normed=True  # menampilkan persentase
)
plt.title("Confusion Matrix - KNN")
plt.show()
```

Figure 5 Script Confunsion Matrix KNN

3.5 *Evaluation*

Following the completion of the modelling stage, the next step is model evaluation, where a confusion matrix is used to assess the accuracy of the constructed model. The structure of the confusion matrix can be seen in

4 Conclusion

This research successfully categorized public sentiments toward PLN's services into positive, negative, and neutral using Naïve Bayes and KNN algorithms. Naïve Bayes proved more efficient for large-scale data, whereas KNN captured complex patterns better but required more processing time.

5 Suggestion

Future studies may integrate additional algorithms such as SVM or Random Forest to improve classification accuracy and explore multi-label sentiment analysis to identify multiple sentiments within a single opinion.

6 Acknowledgments

The authors would like to thank their academic supervisors and all parties who provided support during the completion of this research..

BIBLIOGRAPHY

[1] M. S. Alrajak, I. Ernawati, and I. Nurlaili, "Analisis sentimen terhadap Pelayanan PT PLN di Jakarta pada Twitter dengan Algoritma K- Nearest Neighbor (K-NN)," Semin. Nas. Mhs. Ilmu Komput. dan Apl., vol. 1, no. 2, pp. 110–122, 2020.

[2] F. Herlando, A. R. Dzikrillah, F. Nufairi, E. Sinduningrum, and M. Sholeh, "Analisis Perbandingan Sentiment Dan Perbincangan Netizen Terhadap Twitter Pasca Pergantian Nama," JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 9, no. 1, pp. 360–367, 2024, doi: 10.29100/jipi.v9i1.4934.

[3] A. Fitri, N. Azizah, and V. P. Ramadhan, "Komparasi Naïve Bayes dan K-NN Dalam Analisis Sentimen di Twitter Terhadap Kemenangan Paslon 02," no. 204, pp. 228–237, 2024.

[4] S. Syafrizal, M. Afdal, and R. Novita, "Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor," MALCOM Indones. J. Mach. Learn. Comput. Sci., vol. 4, no. 1, pp. 10–19, 2023, doi: 10.57152/malcom.v4i1.983.

[5] M. Syarifuddinn, "Analisis Sentimen Opini Publik Mengenai Covid-19 Pada Twitter Menggunakan Metode Naïve Bayes Dan Knn," INTI Nusa Mandiri, vol. 15, no. 1, pp. 23–28, 2020, doi: 10.33480/inti.v15i1.1347.

[6] D. Jollyta, W. Ramdhan, and M. Zarlis, KONSEP DATA MINING DAN PENERAPAN. 2020.

[7] A. Zanuardi and H. Suprayitno, "Analisa Karakteristik Kecelakaan Lalu Lintas di Jalan Ahmad Yani Surabaya melalui Pendekatan Knowledge Discovery in Database," J. Manejemen Aset Infrastruktur Fasilitas, vol. 2, no. 1, pp. 45–55, 2018, doi: 10.12962/j26151847.v2i1.3767.

[8] WowonPriatna, EkaNurA'ini, JoniWarta, AgusHidayat, TyastutiSriLestari, and Rasim, "Sentiment Analysis of Bjorka Hacker Using the Naive Bayes and C.45 Algorithms," 2023.

[9] Gading Viewieanti Eka Nur Fazriah, "Analisis Sentimen Pengguna Twitter Terhadap Pemakai Gadget Pada

Anak Usia Dini Menggunakan Metode Support Vector Machine (SVM) dan Naive Bayes," 2024.

[10] Trias Handayanto and H. Herlawati, DATA MINING DAN MACHINE LEARNING MENGGUNAKAN MATLAB DAN PYTHON. 2020.

[11] M. K. Amril Mutoi Siregar, S.Kom. and M. K. Adam Puspabhuana, S.Kom., DATA MINING. CV Kekata Group, 2017.

[12] Endrik, A. Nugroho, and A. T. Zy, "Penerapan Algoritma Naive Bayes dan PSO pada Analisis Sentimen Kandidat Calon Presiden 2024," Remik Ris. dan E-Jurnal Manaj. Inform. Komput., vol. 7, no. 3, pp. 1367–1380, 2023.