Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

Twitter Sentiment Towards 2024 Jakarta Governor Candidates With Naïve Bayes Algorithm

Fikri Abei[1)*], Asep Arwan Sulaeman[2)], Suprapto[3)]
[1,2,3)] Informatics Engineering Study Program, Faculty of Engineering, Pelita Bangsa University, Indonesia
[1)]fikriabei88@gmail.com, [2)]aseparwan@pelitabangsa.ac.id, [3)]suprapto@pelitabangsa.ac.id

ABSTRACT

This study aims to analyze public sentiment towards candidates for the 2024 Governor of DKI Jakarta through the Twitter platform, with a focus on classifying positive and negative sentiment. Along with the rapid development of social media, Twitter has become the main channel for people to voice political opinions. Sentiment analysis was conducted using the Naive Bayes algorithm to classify the sentiment of tweets collected through crawling techniques during the campaign period. The data used includes user tweets, with features such as frequently occurring words, popular hashtags, and discussion topics related to each gubernatorial candidate. The results showed that the Naive Bayes algorithm provided the best performance in classifying sentiment data in the period August 1 to December 26, 2024, with the highest accuracy rate reaching 75% at a data ratio of 90:10. This research also identified challenges in sentiment classification, such as the presence of new terms in test documents that are not recognized by the training model. The findings are expected to provide a clearer picture of public perceptions of gubernatorial candidates and contribute to the analysis of political sentiment on social media.

Keywords: Sentiment Analysis; Twitter; Naïve Bayes Algorithm; Jakarta Governor Election 2024; Public Sentiment; Social Media; Text Classification

INTRODUCTION

The election of the Governor of DKI Jakarta has always been a big concern among the public. As the capital city of the country, Jakarta has a strategic role both in terms of politics and social, so every stage of the regional head election attracts public attention. In today's digital era, social media, especially Twitter, has developed into an interactive platform for people to express their opinions and respond to various issues related to the elections. With its fast, transparent and accessible characteristics, Twitter has become one of the main platforms for sharing views on gubernatorial candidates.

Sentiment analysis of public opinion on social media is increasingly important, especially ahead of political events such as regional elections. The sentiment reflected on Twitter can provide insights into the public's views on each candidate, as well as reveal the issues that are of primary concern to the public (Isnain, 2021). However, the massive amount of data on Twitter every day creates challenges for manual processing and analysis.

Referring to research conducted by (Darwis, 2021), This research aims to utilize the development of Twitter as a platform for users to convey information, criticism, and suggestions related to the services provided by the National BMKG. By applying the Naïve Bayes Classifier (NBC) algorithm, this research develops a sentiment classification system to determine whether a tweet has a positive, neutral or negative sentiment (Peretz, 2024). Data taken from Twitter will go through a text mining process before classification. This research also utilizes the Python 3.74 application as a tool to support data processing and classification processes (Wornow, 2023). In addition, this research aims to make it easier for users to understand public opinion about the BMKG by categorizing tweets into three main categories: positive, neutral and negative. With a Naïve Bayes algorithm accuracy rate of 69.97%, this research presents a sentiment analysis approach that can help in understanding public perceptions of BMKG services.

This research aims to identify people's sentiment towards the 2024 DKI Jakarta Governor candidates

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

on Twitter by using the Naive Bayes algorithm, which is known for its ability to classify text simply and effectively. This research will develop a model to categorize tweets into two main categories, namely positive and negative sentiments. In addition to providing a better understanding of the public's views on the 2024 DKI Jakarta Governor candidates, this research is also expected to contribute to the development of sentiment analysis methods in Indonesia, especially in the political realm. With this research, it is hoped that in-depth insight into the pattern of public sentiment on social media can be obtained as a form of public participation in the democratic process.

LITERATURE REVIEW

This research draws on literature from previous relevant studies, especially those from scientific journals. The following are sources related to this research.

According to (Insan, 2023), this study analyzes user sentiment on the BRImo application on Google Play using web scraping and the Naïve Bayes algorithm. From 1,550 reviews collected between August 2022 and January 2023, 1,012 were classified as positive and 894 as negative, indicating a majority of favorable feedback despite some negative sentiment. The Naïve Bayes algorithm achieved 84.52% accuracy, 82.51% precision, and 87.62% recall.

According to (Lestari, 2021), this study aims to analyze public sentiment in Indonesia towards the Sinovac vaccine through opinions shared on Twitter. By applying the Naive Bayes method and using RapidMiner, this study classifies the sentiment of English tweets about the Sinovac vaccine into two categories: positive and negative. This study aims to uncover the public's views on the Sinovac vaccine, which has become a hot topic on social media, and provide insight into the public's perception of the vaccine.

According to (Febriyani, 2023), this study analyzes the sentiment of public opinion towards the Kampus Merdeka program on Twitter by measuring the accuracy of the method and the percentage of sentiment. Data was collected from tweets with the hashtags #kampusmerdeka and #mbkm between November 2021 and March 2022, using the Naive Bayes Classifier algorithm. The analysis of 501 tweets showed 272 positive and 229 negative opinions, with 60% accuracy, 64% precision, 58% recall, and 58% f1-score. The program was built using Python on Google Colab, and the visualization shows the dominant word in each sentiment.

According to (Azhar, 2022), this study aims to analyze the sentiment of public opinion regarding cryptocurrencies on Twitter, especially with regard to the hashtag #crypto, by using machine learning to automate sentiment labeling. By utilizing Python TextBlob, this research categorizes tweets into positive, neutral, and negative sentiments, and tests the Naïve Bayes algorithm to measure accuracy, precision, recall, and f1-score.

According to (Sulaeman, 2024), this study analyzes public responses to ChatGPT on Twitter and tests the performance of the K-Nearest Neighbors (KNN) method in classifying tweet sentiment. Responses were divided into positive and negative categories, and the KNN model was tested with a variety of k values. The analysis showed that 74.3% of the responses were positive and 25.7% were negative. The KNN test yielded the highest accuracy of 88% at k=5, with adequate precision, recall, and f1-score. In conclusion, sentiment analysis and classification using KNN is effective for understanding public responses to ChatGPT.

According to (Satria, 2020), this research aims to analyze the sentiment of user reviews on Google Play Store to evaluate the consistency between ratings and comments. In addition, this research aims to measure the performance of Naïve Bayes, C4.5, and Levenshtein Distance algorithms in classifying sentiment. Levenshtein Distance is utilized for text normalization, while Naïve Bayes and C4.5 algorithms are applied for classification based on review and rating text attributes. This research also aims to assess the effectiveness of the combination of these methods in improving the accuracy of sentiment classification in mobile applications.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

METHOD

This research focuses on analyzing sentiment on Twitter towards the 2024 DKI Jakarta governor candidates. To achieve this goal, the research will involve a number of stages, including modeling the Naïve Bayes algorithm. This algorithm is proposed as a sentiment classification method to categorize public opinion into positive and negative sentiments based on the collected tweet data.

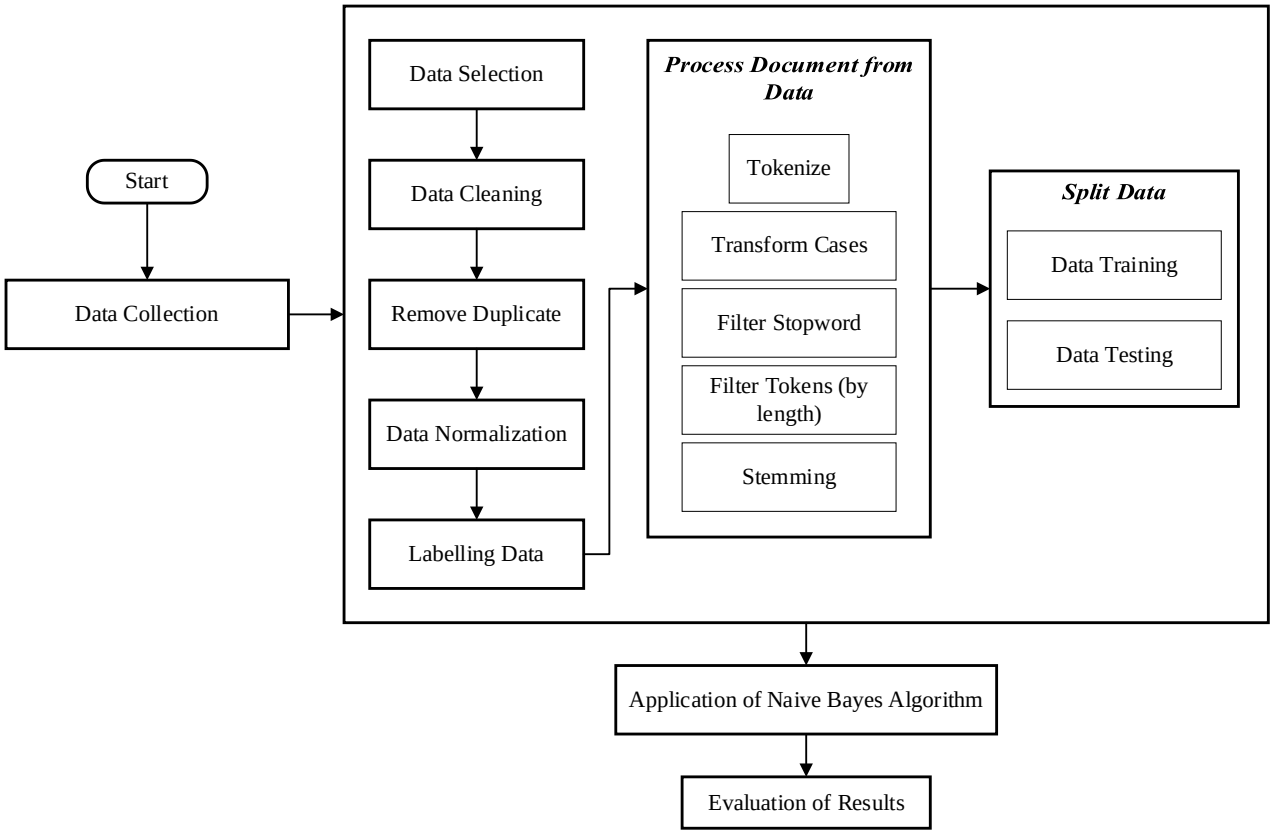

Fig 1 Sentiment Analysis Process

Twitter sentiment analysis related to DKI Jakarta 2024 gubernatorial candidates begins by collecting data using queries such as "pilgub dki jakarta 2024" through certain tools. The data obtained then goes through the stages of text cleaning, selection of relevant features, and format transformation in preparation for analysis. The processed data is used to build sentiment models with the Naïve Bayes algorithm. Model evaluation is done by dividing the data into training set and testing set, and measured using metrics such as accuracy, precision, recall, and F1-score. If the evaluation results are not optimal, the processing and modeling process will be improved until the model can provide accurate and reliable results.

Naïve Bayes

Naïve Bayes is a widely used algorithm in classification tasks, particularly for text analysis. It is based on the Bayes probability principle with the assumption that each feature in the data is mutually independent. Although the method is simple, Naïve Bayes is proven to be efficient and capable of providing accurate results in determining sentiment polarity, even on unbalanced datasets. In addition, it is characterized by high bias and low variance, making it suitable for application to small-sized datasets (Samsir et al., 2021).

Naïve Bayes is a probability-based machine learning algorithm used for classification tasks. It utilizes Bayes' Theorem with the assumption that every feature in the dataset is independent of each other. Its simple

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

and efficient nature makes it widely used in various applications, such as sentiment analysis, spam filtering, and document clustering. Despite its simple concept, Naïve Bayes is able to provide good performance, even on datasets with small sizes or imbalances (Jeng, 2021).

The Naïve Bayes algorithm calculates the probability that a sample with a certain characteristic belongs to class h (posterior) by multiplying the probability of class x by the probability of occurrence of that sample characteristic in class c (likelihood). This relationship is generally explained through the following equation.

$$P(c|x) = \frac{(P(x|c)xP(c))}{P(x)} \quad (1)$$

Description:

x : Data with unknown class.
c : Hypothesis that the data x belongs to a certain class.
$(c|x)$: Posterior probability of the class c (target) based on the given predictor x (attribute).
(c) : Initial probability of the class.
$(x|c)$: Probability of occurrence of the predictor x in the given class c.
(x) : Probability of occurrence of predictor x without considering the class.

RESULT

The purpose of this response data analysis is to provide an overview of the sentiment on Twitter regarding the 2024 DKI Jakarta gubernatorial candidates. After text processing and automatic labeling, the data collected from August 1, 2024 to December 26, 2024 was divided into two categories, positive and negative, to evaluate all positive and negative responses.

In this research, data collection was conducted using Google Colab with the query "pilgub dki jakarta 2024" and a maximum data limit of 2000 with the language "id". The crawling process produced 1,500 data, and the final amount of data obtained was 993.

Table 1.Dataset after Crawling

No	final_Text	username
1	welcome to jakarta s new leader happy mas pram and bang doel to go to jakarta.....	AleenaDi09
2	dki jakarta pilgub vote peacefully support mas pram bang doel stable character commitment.....	aniesbaswedan
…	…..	…..
…	…..	…..
992	amp proker vision i hold price benyamin s clean creat makmur commitment form jakarta work program.....	lensaRTV
993	real close jkt inclusive city shares record teacher welfare level.....	lensaRTV

The next step, after obtaining the data, is to determine its accuracy by going through several stages, including:

Preprocessing Data

Data preprocessing is a series of steps to process raw data to make it ready for use in analysis or modeling. This process includes data cleaning (for example, dealing with missing data and duplication), transformations (such as normalization and coding), and selection or removal of irrelevant features. The goal of preprocessing is to improve data quality so that the resulting analysis or model is more accurate and reliable. This stage is a crucial first step in data mining or machine learning (Fan, 2021).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

Data Selection: The process of selecting relevant subsets of data from a large data set based on certain criteria, such as time, location, or attributes, to ensure more efficient and quality analysis. For example, in the analysis of public sentiment towards air pollution, data selection includes selecting tweets based on a specific region or time range as well as eliminating irrelevant data, which is an important stage in data preprocessing (Gómez, 2021).

```
# Read CSV files with only 'full_text' columns
df = pd.read_csv(file_path, usecols=['username','full_text'])

# Display the first 100 lines in markdown format, or fewer if not enough
print(df.head(100).to_markdown(index=False, numalign="left", stralign="left"))
```

Fig 2 Data Selection

The following are the results of data selection using the Python programming language:

Table 2.Data Selection Result

No	full_text	username
1	Welcome to Jakarta's new leaders and congratulations to Mas Pram and Bang Doel for their work towards Jakarta...	AleenaDi09
…	…..	…..
993	Anyway, inaugurated on February 7, 2025 Pramono Anung and Rano Karno Si Doel immediately...	PaltiWest2024

Data Cleaning: The process of correcting or removing errors and inconsistencies in data to improve its quality, so that it can be optimally used in analysis. This step involves addressing issues such as missing values, duplication, improper formatting or entry errors, with the aim of ensuring the data is accurate, complete and consistent. As an important part of data preprocessing, data cleaning plays a crucial role in preventing the negative impact of low-quality data on analysis results (Athanasios, 2022).

```
# text cleanup with regex compilation
url_pattern = re.compile(r'http\S+|www\.\S+')
mention_hashtag_pattern = re.compile(r'@\w+|#\w+')
punctuation_pattern = re.compile(r'[^\w\s]')
number_pattern = re.compile(r'\d+')

def clean_text(text):
    text = url_pattern.sub('', text)  # Delete URL
    text = mention_hashtag_pattern.sub('', text)  # Remove mentions and hashtags
    text = punctuation_pattern.sub('', text)  # Remove punctuation
    text = number_pattern.sub('', text)  # Delete numbers
    return ' '.join(text.split())  # Remove double spaces
```

Fig 3 Data Cleaning

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

The following are the results of data cleaning using the Python programming language:

Table 3. Data Cleaning Result

No	username	cleaned_tex
1	AleenaDi09	Welcome to Jakarta's new leaders and congratulations to Mas Pram and Bang Doel for their work towards Jakarta...
…	…..	…..
993	Mata_Netizen62	Mas Pram And Bang Doel Are Governor And Deputy Governor Of Jakarta The General Election Commission...

Removing Duplicated: A step to detect and eliminate duplicate data in a dataset to ensure each entry is unique and the data is of good quality. Duplicate data can arise due to input errors, merging of datasets, or uncontrolled collection processes. This process is important to avoid distorting analysis results, reduce redundancy, and improve data processing efficiency (Cassidy, 2022).

```
# Remove duplicates and keep only the first row of each group
df_cleaned = df.drop_duplicates(subset=['cleaned_text'], keep='first')
```

Fig 4 Removing Duplicated

The following are the results of Removing Duplicated using the Python programming language:

Table 4.Removing Duplicated Result

No	username	cleaned_text
1	AleenaDi09	Welcome the new leader of Jakarta, congratulations...
…	…..	…..
993	ch_chotimah2	Will go down in history A comedian just...

Data Normalization: The process of converting data into a uniform scale or format to improve the quality and accuracy of analysis. The goal of normalization is to equalize data values so that they fall within the same range or have a similar distribution, such as converting data to a range of 0 to 1 or converting categorical values to numbers. This process is important to avoid bias in the analysis, especially when using algorithms that are sensitive to differences in data scale or distribution (Huang et al., 2023).

```
# Function to normalize text
def normalize_text(text):
    for word, replacement in normalization_dict.items():
        text = re.sub(r'\b' + re.escape(word) + r'\b', replacement, text, flags=re.IGNORECASE)
    text = re.sub(r'(.)\1{2,}', r'\1', text)  # Remove repeating characters
    return text
```

Fig 5 Normalization Data

The following are the results of Data Normalization using the Python programming language:

Table 5.Data Normalization Result

No	username	tweet	sentiment
1	AleenaDi09	Welcome to Jakarta's new leader happy Mas Pram and Bang Doel...	Positive
2	aniesbaswedan	In this DKI Jakarta Pilgub, I chose to make peace...	Positive
…	…..	…..	…..
992	PaltiWest2024	Anyway, inaugurated February Pramono Anung and Rano Karno...	Positive
993	EenExWokeDocent	Pogromhangover After horrible Jewish Yacht Night In...	Negative

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

Labelling Data: The process of labeling or categorizing data to classify it, so that it is ready for use in analysis, model training, or machine learning. This process is commonly used in supervised learning, where the labeled data is used to train a model to make predictions or classifications on unlabeled data. The label can be a category, a number, or a classification based on certain criteria (Zhang et al., 2021).

```
# Labeling to determine sentiment (no neutrals)
def get_sentiment(final_text):
    analysis = TextBlob(final_text)
    # Jika polarity 0, diubah ke positif untuk menghindari netral
    if analysis.polarity >= 0:
        return 'positif'
    else:
        return 'negatif'
```

Fig 6 Labelling Data

The following are the results of Labelling Data using the Python programming language:

Table 6.Labelling Data Result

No	username	tweet	sentiment
1	AleenaDi09	Welcome to Jakarta's new leader happy Mas Pram...	Positive
2	aniesbaswedan	In this DKI Jakarta Pilgub, I chose to make pe...	Positive
…	…	…..	…..
992	aniesbaswedan	Given the rising of the sun to take place with...	Negative
993	ch_chotimah2	It will be noted that the history of a comedia...	Positive

Process Document from Data

Document processing is an advanced stage in data processing that includes various steps such as changing case, tokenizing, filtering, removing common words (stopwords), and stemming (Bodell, 2022). These stages can be seen below.

Tokenize: The process of breaking down text into small units called tokens, such as words or phrases, for easier analysis and processing in natural language processing (NLP) (Sun et al., 2023).

Transform Case: The process of converting text into a consistent font format, such as lowercase or uppercase, to ensure uniformity in text analysis (Tijan, 2021).

Filter Stopword: The process of removing common words that do not provide important meaning, such as conjunctions or prepositions, to improve the quality of text analysis (Sarica & Luo, 2021).

Filter Tokens (by length): The process of removing short words, such as one or two letters, that are deemed not to make a significant contribution to the text analysis (Kim et al., 2022).

Stemming: The process of changing an inflected word to its base or root form, such as changing “running” to “run”, to simplify the text in analysis (Abidin, 2024).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

The following is a process document from data using the Python programming language.

```
# process document from data
def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()
    # Tokenization by finding words
    tokens = re.findall(r'\b\w+\b', text)
    # Remove stopwords
    tokens_no_stopwords = [word for word in tokens if word not in stop_words]
    # Perform stemming
    tokens_stemmed = [stemmer.stem(word) for word in tokens_no_stopwords]
    # Recombine tokens into a final sentence
    final_text = ' '.join(tokens_stemmed)
    return tokens, tokens_no_stopwords, tokens_stemmed, final_text
```

Fig 7 Process Document from Data

The following are the results of Labelling Data using the Python programming language:

Table 7.Process Document from Data Result

No	username	Tweet	sentiment
1	AleenaDi09	welcome to jakarta's new leader happy mas pram...	Positive
2	aniesbaswedan	Considering that the rising of the sun to occur in...	Negative
…	…..	…..	…..
992	ekowboy2	Bang Doel was asked what was asked until Mr. a...	Positive
993	ch_chotimah2	It should be noted that the history of a comedian ha...	Positive

Split Data: The process of dividing the dataset into subsets, usually for training and testing the model, to avoid overfitting and evaluate the model performance (Nguyen et al., 2021).

```
# Divide the dataset into training data and test data with a ratio of 90:10
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.1, random_state=19)
```

Fig 8 Split Data

The following are the results of Split Data using the Python programming language:

```
Distribusi Sentimen pada Data Training:
Sentiment
negatif    116
positif    777
Name: count, dtype: int64

Distribusi Sentimen pada Data Testing:
Sentiment
negatif    12
positif    88
Name: count, dtype: int64
```

Fig 9 Split Data Result

Application of Naïve Bayes Method

The use of classification algorithms based on Bayes' Theorem assumes independence between features. This algorithm is used to classify text, such as sentiment analysis, by calculating the probability of a feature occurring in each class and classifying the data based on the highest probability (Viet, 2021).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

```
# Initialize and train Naive Bayes model
model = MultinomialNB()
model.fit(X_train, y_train)

# Prediction on test data
y_pred = model.predict(X_test)

# Naive Bayes Model Evaluation
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, output_dict=True)
precision = report['macro avg']['precision']
recall = report['macro avg']['recall']
```

Fig 10 Naïve Bayes Algorithm Modeling Process

The results of modeling the naive bayes algorithm can be seen below:

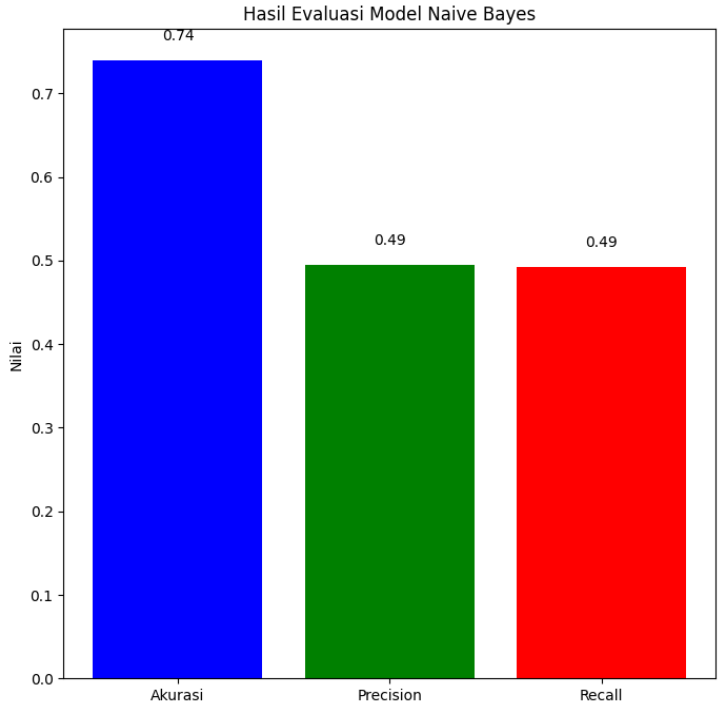

Fig 11 Naïve Bayes Method Application Result

The results of the calculations that have been carried out using Google Collab using the Naïve Bayes algorithm are 74% accuracy, 49% precision, and 49% Recall.

WordCloud

A text visualization that displays words based on their frequency, with more frequently occurring words displayed larger, to identify keywords or major themes (Selvi, 2021).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

Fig 12 Visualization Word Cloud Result

In the wordcloud visualization shown, some of the keywords that often appear are “jakarta”, “choose”, “ridwan kamil”, “pilgub”, and “bang doel”. The occurrence of these words indicates the main topic discussed is related to the DKI Jakarta gubernatorial election.

Fig 13 Visualization Results Based On Words That Appear Frequently

The word frequency diagram provides deep insights into the distribution of words in texts focusing on the Tapera program. In this case, the most frequently occurring words, such as “Jakarta”, “Ridwan Kamil”, “Bang Doel”, “pilgub”, and “win”, illustrate the main topics frequently discussed by Twitter users.

DISCUSSIONS

This research uses a dataset of tweets related to the 2024 DKI Jakarta governor candidates collected through the Twitter API. Data from this dataset is processed through several stages such as preprocessing, feature selection, data splitting, data mining, and evaluation. In this study, researchers used the Naïve Bayes algorithm to classify tweet sentiment into positive and negative. The test results showed that the Naïve Bayes method successfully achieved an average accuracy rate of 75%. In addition, the use of TF-IDF method in feature selection makes it possible to focus on the most relevant words in the tweet, thus improving the algorithm's ability to classify sentiment.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

CONCLUSION

Based on the analysis, the Naïve Bayes method with 10-fold cross-validation technique at a ratio of 90% training data and 10% test data produces an average accuracy of about 75% of a total of 893 training data and 100 test data. This study found that positive sentiments dominate in the analysis of tweets related to the 2024 DKI Jakarta governor candidates. The majority of users show support for candidates who have a good track record, a vision that is relevant to the needs of the community, and work programs that are considered adequate. In contrast, negative sentiments are generally related to political issues, dissatisfaction with the performance of certain candidates, or controversies that accompany them.

This finding reflects how public opinion is formed based on perceptions of candidates' competence, integrity and work programs. It also confirms the importance of understanding public sentiment to formulate more effective political communication strategies.

REFERENCES

Abidin, Z. (2024). Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review. *Journal of Information Systems Engineering and Business Intelligence*, *10*(2), 217–231. https://doi.org/10.20473/jisebi.10.2.217-231

Athanasios. (2022). Ontology-Driven Data Cleaning Towards Lossless Data Compression. *Studies in Health Technology and Informatics*, *294*, 421–422. https://doi.org/10.3233/SHTI220492

Azhar, R. (2022). Analisis Sentimen Terhadap Cryptocurrency Berbasis Python TextBlob Menggunakan Algoritma Naïve Bayes. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *6*(1), 267–281. https://doi.org/10.30645/j-sakti.v6i1.443

Bodell, M. H. (2022). From Documents to Data: A Framework for Total Corpus Quality. *Socius*, *8*, 1–15. https://doi.org/10.1177/23780231221135523

Cassidy, B. (2022). Analysis of the ISIC image datasets: Usage, benchmarks and recommendations. *Medical Image Analysis*, *75*, 102305. https://doi.org/10.1016/j.media.2021.102305

Darwis, D. (2021). Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional. *Jurnal Tekno Kompak*, *15*(1), 131. https://doi.org/10.33365/jtk.v15i1.744

Fan, C. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research*, *9*(March), 1–17. https://doi.org/10.3389/fenrg.2021.652801

Febriyani, E. (2023). Analisis Sentimen Terhadap Program Kampus Merdeka Menggunakan Algoritma Naive Bayes Classifier Di Twitter. *Jurnal Tekno Kompak*, *17*(1), 25. https://doi.org/10.33365/jtk.v17i1.2061

Gómez, C. S. F. K. J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, *181*(2021), 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., & Shao, L. (2023). Normalization Techniques in Training DNNs: Methodology, Analysis and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(8), 10173–10196. https://doi.org/10.1109/TPAMI.2023.3250241

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

Insan, M. K. (2023). ANALISIS SENTIMEN APLIKASI BRIMO PADA ULASAN PENGGUNA DI GOOGLE PLAY MENGGUNAKAN ALGORITMA NAIVE BAYES. *Jurnal Mahasiswa Teknik Informatika*, *7*(1), 478–483. https://doi.org/10.36040/jati.v7i1.6373

Isnain, A. R. (2021). Sentiment Analysis Of Government Policy On Corona Case Using Naive Bayes Algorithm. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *15*(1), 55–64. https://doi.org/10.22146/ijccs.60718

Jeng, C. V. J. J. M. J.-G. H. X. A. I. J.-H. (2021). Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes. *Information (Switzerland)*, *12*(5), 1–16. https://doi.org/10.3390/info12050204

Kim, S., Shen, S., Thorsley, D., Gholami, A., Kwon, W., Hassoun, J., & Keutzer, K. (2022). Learned Token Pruning for Transformers. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *22*, 784–794. https://doi.org/10.1145/3534678.3539260

Lestari, S. (2021). Analisis Sentimen Vaksin Sinovac Pada Twitter Menggunakan Algoritma Naive Bayes. *Seminar Nasional Sistem Informasi Dan Manajemen Informatika*, *7*(1), 163–170. Retrieved from https://sismatik.nusaputra.ac.id/index.php/sismatik/article/view/23

Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., Van Le, H., Tran, V. Q., … Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*, 1–15. https://doi.org/10.1155/2021/4832864

Peretz. (2024). Naive Bayes classifier – An ensemble procedure for recall and precision enrichment. *Engineering Applications of Artificial Intelligence*, *136*(1), 108972. https://doi.org/10.1016/j.engappai.2024.108972

Samsir, Irmayani, D., Edi, F., Harahap, J. M., Jupriaman, Rangkuti, R. K., … Watrianthos, R. (2021). Naives Bayes Algorithm for Twitter Sentiment Analysis. *Journal of Physics: Conference Series*, *1933*(1), 1–6. https://doi.org/10.1088/1742-6596/1933/1/012019

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. *PLoS ONE*, *16*(8), 1–13. https://doi.org/10.1371/journal.pone.0254937

Satria, A. R. (2020). Analisis Sentimen Ulasan Aplikasi Mobile menggunakan Algoritma Gabungan Naive Bayes dan C4.5 berbasis Normalisasi Kata Levenshtein Distance. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *4*(11), 4154–4163. Retrieved from https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8270

Selvi, B. (2021). Word Clouds in Grammar Production. *Turkish Online Journal of English Language Teaching (TOJELT)*, *6*(1), 44–57.

Sulaeman, A. A. (2024). Sentiment Analysis on Social Media X (Twitter) Against ChatGBT Using the K-Nearest Neighbors Algorithm. *Brilliance: Research of Artificial Intelligence*, *4*(1), 265–275. https://doi.org/10.47709/brilliance.v4i1.4105

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 7, Number 1, Januari 2025
https://doi.org/10.47709/cnahpc.v7i1.5358

Submitted : Jan 18, 2025
Accepted : Jan 23, 2025
Published : Jan 25, 2025

Sun, W., Yan, L., Chen, Z., Wang, S., Zhu, H., Ren, P., … Ren, Z. (2023). Learning to Tokenize for Generative Retrieval. *NeurIPS Proceedings*, *36*, 1–17. Retrieved from http://arxiv.org/abs/2304.04171

Tijan, E. (2021). Digital transformation in the maritime transport sector. *Technological Forecasting and Social Change*, *170*, 1–15. https://doi.org/10.1016/j.techfore.2021.120879

Viet, T. N. (2021). The naÏve bayes algorithm for learning data analytics. *Indian Journal of Computer Science and Engineering*, *12*(4), 1038–1043. https://doi.org/10.21817/indjcse/2021/v12i4/211204191

Wornow, M. (2023). APLUS: A Python library for usefulness simulations of machine learning models in healthcare. *Journal of Biomedical Informatics*, *139*(February), 1–12. https://doi.org/10.1016/j.jbi.2023.104319

Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J. F., Barriuso, A., … Fidler, S. (2021). DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 10140–10150. https://doi.org/10.1109/CVPR46437.2021.01001

* Corresponding author