Public Sentiment Analysis on the 2024 North Sumatra Governor Candidates on YouTube: A Comparison of Naïve Bayes and TextBlob Algorithms

Suvriadi Panggabean[1], Dedy Kiswanto[2], Nurul Maulida Surbakti[3]

{suvriadi@unimed.ac.id[1], dedykiswanto@unimed.ac.id[2], nurulmaulida@unimed.ac.id[3]}

[1,2,3]Mathematics Department, Faculty of Mathematics and Natural Sciences, Medan State University

Abstract. The 2024 North Sumatra Governor Election has captured public attention, especially on social media platforms such as YouTube. This study aims to analyze public sentiment toward two gubernatorial candidates, Boby Nasution and Edy Rahmayadi, based on comments posted on YouTube. Two sentiment analysis methods were employed in this research: Naïve Bayes and TextBlob. The analysis results reveal that negative sentiment dominates public opinion toward both candidates, with Edy Rahmayadi receiving more criticism compared to Boby Nasution. Although there are some positive and neutral comments, the dominance of negative sentiment is the primary focus. These findings highlight the importance for both candidates to respond effectively to public criticism through improved campaign strategies. The visualization of the analysis results provides clear insights into the distribution of public sentiment, which can serve as a guide for the candidates in formulating their strategic steps moving forward.

Keywords: Sentiment Analysis, Naïve Bayes, TextBlob, Governor Election, YouTube, Public Perception.

1. Introduction

The 2024 North Sumatra governor election has drawn significant public attention, particularly on social media platforms like YouTube. Comments on YouTube can be analyzed to understand the overall public sentiment. With the increasing use of social media, sentiment analysis has become an important approach to categorizing opinions as positive, negative, or neutral [1]. This study utilizes two methods: the Naïve Bayes Classifier, which is effective for probabilistic-based sentiment analysis [2], and TextBlob, a Python library that leverages natural language processing [12]. By employing these two methods, this study aims to analyze public sentiment toward the 2024 North Sumatra governor candidates based on comments on YouTube.

ICIESC 2024, September 17, Medan, Indonesia

DOI 10.4108/eai.17-9-2024.2353095

2. Literature Review

2.1 Sentiment Analysis

Sentiment analysis is a technique for extracting and classifying emotional information in text, identifying whether the sentiment contained is positive, negative, or neutral. In this context, sentiment analysis is used to understand public reactions to the 2024 North Sumatra gubernatorial candidates. This technique has been applied in various fields, such as product reviews and political opinions, with Pang and Lee (2008) emphasizing the importance of sentiment analysis in understanding public views from online texts [1].

2.2 Naïve Bayes Classifier

Naïve Bayes is a classification algorithm based on Bayes' theorem, which assumes independence between features. This algorithm is known for its simplicity, speed, and effectiveness in sentiment analysis, especially on social media [2]. Wibowo and Hasan (2022) demonstrated that the Naïve Bayes Classifier could produce accurate results in various cases, including product reviews and political opinions, due to its ability to handle large volumes of text data [3].

2.3 TextBlob

TextBlob is a Python library used for processing text, including in sentiment analysis. This library enables tasks such as tagging, parsing, translation, and sentiment analysis. Loria (2018) stated that TextBlob is effective in analyzing English-language texts, particularly in determining sentiment polarity [12]. With the support of the NLTK library, TextBlob becomes a practical and efficient tool for sentiment analysis, especially for texts translated from other languages.

2.4 Using Social Media as a Sentiment Data Source

Social media platforms like YouTube, Twitter, and Facebook are rich data sources for sentiment analysis. Various studies have shown that comments on social media can be used to analyze public preferences and opinions in political contexts. Ahmad et al. (2019) noted that social media plays a crucial role in public opinion analysis, particularly ahead of political elections [7]. Afandi et al. (2022) also found that sentiment analysis from social media can provide accurate insights into public views on political issues [8].

2.5 Data Crawling with Python

Web scraping or crawling is an automated process for collecting data from websites, including social media. Python libraries like BeautifulSoup and snscrape are often used because of their flexibility and capability to retrieve large amounts of data. Wati and Ernawati (2021) stated that web scraping is an effective method for obtaining datasets from social media without requiring complex API access [9]. Hasan and Dwijayanti (2021) also demonstrated that this process allows researchers to access information not limited by the regulations of certain platforms [2].

3. Research Methods

The following figure illustrates the research stages carried out by the researchers:

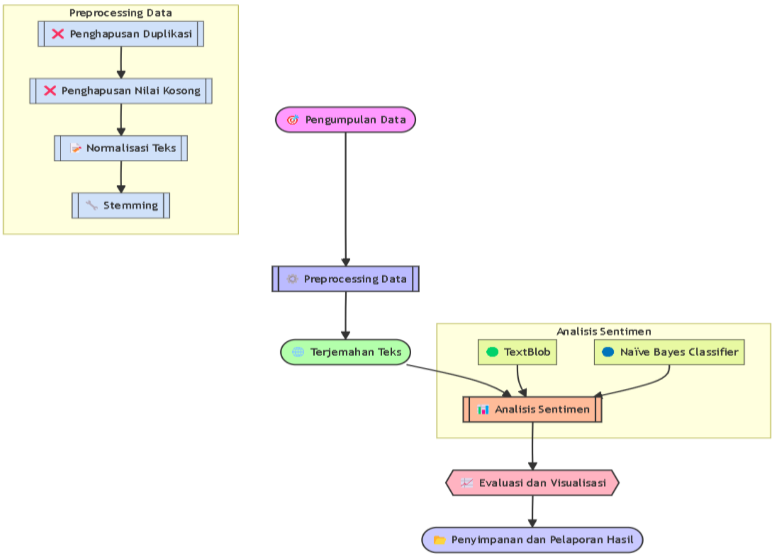


Fig.1. Research Stages

The research method used in this study to analyze public sentiment toward the 2024 North Sumatra gubernatorial candidates through YouTube comments was carefully designed to ensure the accuracy and validity of the results. Each stage in this method plays a crucial role in producing reliable findings.

1. Data Collection
   Data was collected through the YouTube API, enabling automatic retrieval of comments from relevant videos. This method is often used in sentiment analysis research on social media due to its ability to access data in real-time and in large quantities[10].
2. Data Preprocessing
   Preprocessing includes removing duplicates, normalizing text, and stemming to clean the data and ensure analysis is conducted on relevant and clean data. Good preprocessing can improve the accuracy of sentiment analysis models[11].
3. Text Translation
   The processed text is then translated from Indonesian to English using Google Translator to leverage TextBlob's strengths in analyzing English text[12].
4. Sentiment Analysis
   Sentiment analysis is performed using two methods: TextBlob and Naïve Bayes Classifier. Naïve Bayes is a frequently used algorithm in text classification and has proven effective in various studies, including in the context of sentiment analysis [13].

5. Evaluation and Visualization
   The analysis results are evaluated by comparing the two methods and visualized using word clouds and bar charts to present the sentiment distribution in a more intuitive and easily understood manner[14].

4. Results and Discussion

This study aims to analyze public sentiment toward the 2024 North Sumatra gubernatorial candidates through YouTube comments. Two analysis methods, Naïve Bayes Classifier and TextBlob, were used to obtain a clearer picture of public perception and to identify the strengths and weaknesses of each method.

4.1 Data Collection

Data was collected using the YouTube API from various channels discussing the North Sumatra gubernatorial election 2024, such as Kompas TV, TVOne, and MetroTV. The use of APIs allows real-time and large-scale data collection, which is crucial to ensuring that the data analyzed truly reflects the current public sentiment. Additionally, using APIs helps reduce bias that might occur if data were collected manually. a code snippet used to collect comment data from relevant videos on YouTube. This process serves as the main foundation of this research, as the quality and quantity of the data collected are critical in determining the accuracy of the sentiment analysis to be performed.

Table 1. Example of Crawling Results

	publish	authorDisplayName	textDisplay	likeCount
0	2024-06-17T04:09:18Z	@BangnopriBae-bs7yg	PDIP partai politik yang tegas dan jelas bukan...	0
1	2024-05-27T17:16:52Z	@cerymomosidharta3760	Kemarin jelas dukung prabowo dan bukan lagi ka...	0
2	2024-05-23T15:52:19Z	@junaidi7245	kita bisa maju tampa PDIP... maju terus GOLK...	0
3	2024-05-23T14:13:56Z	@user-mj9mo3rh9o	Biasanya orang yang baik dan tersakiti oleh pd...	0
4	2024-05-22T11:39:11Z	@Theorganicgal	PDIP menolak Golkar menugaskan Gerindra ...	0

Table 1 displays the results of the data crawling process, which were then saved in CSV format for further analysis. This data is crucial in determining the accuracy of the sentiment analysis.

4.2 Removing Duplicates

After data collection, the first step is to remove duplicates in the dataset. Duplicates can occur if the same comment appears more than once, which could bias the analysis. To avoid this, a Python script was used to identify and delete duplicate comments, ensuring that the resulting dataset is clean and unique, ready for preprocessing.

```
df = df.drop_duplicates(subset=['textDisplay'])
```

Fig.2. Code Snippet for Data Removal

4.3 Preprocessing

The preprocessing process involves several important steps to ensure that the data used in the analysis has optimal quality. These steps include:

1. Cleaning

The cleaning process aims to remove irrelevant characters such as symbols, URLs, and mentions. The result is cleaner text ready for analysis, improving the accuracy of sentiment analysis.

```
Show hidden output

[15] df = df.dropna()

[16] df.isnull().sum()

Show hidden output

[31] # Fungsi untuk membersihkan teks dari simbol, angka, dan karakter yang tidak diperlukan
     def clean_text(text):
         text = re.sub(r'[^A-Za-z0-9]+', ' ', text)  # Menghapus simbol dan karakter yang tidak diperlukan
         text = re.sub(r'\b\w{1,2}\b', '', text)  # Menghapus kata dengan 1 atau 2 karakter
         text = re.sub(r'RT[\s]+', '', text)  # Menghapus kata 'RT'
         text = re.sub(r'https?:\/\/\S+', '', text)  # Menghapus link
         text = re.sub(r'[^A-Za-z0-9 ]+', '', text)  # Menghapus simbol yang tersisa
         text = re.sub(r'\s+', ' ', text).strip()  # Menghapus spasi berlebihan dan strip teks
         return text

     # Menerapkan fungsi pembersihan teks pada kolom 'textDisplay'
     df['textDisplay'] = df['textDisplay'].apply(clean_text)

     # Mengubah teks menjadi huruf kecil (lowercase)
     df['textDisplay'] = df['textDisplay'].str.lower()
```

Fig.3. Example of Results

Figure 4 shows the code used to perform the cleaning process, resulting in text that is free from irrelevant elements.

2. Normalization

The normalization process involves converting informal words to formal ones and standardizing all letters to lowercase. This normalization ensures consistency in the dataset, enabling more accurate sentiment analysis. Figure 5 shows how normalization is carried out to improve the accuracy of the analysis by making the data more consistent.

```
# Normalisasi dengan tambahan singkatan
norm = {
    " yg ": " yang ",
    " gak ": " tidak ",
    "gk" :"tidak",
    "ga":"tidak",

    " boboy ": "  ",
    "bpk":"bapak",
}



def normalisasi(str_text):
    for i in norm:
        str_text = str_text.replace(i, norm[i])
    return str_text

df['textDisplay'] = df['textDisplay'].apply(lambda x: normalisasi(x))
```

Fig.4. Code for Normalization

3. Tokenization

```
tokenized = df['textDisplay'].apply(lambda x: x.split())
tokenized
```

	textDisplay
0	[Kami, rakyat, medan, mendukung, bapak, edi, e...
1	[Betul, gus.anda, lebih, cocok, jadi, Gubernur...
2	[Menurut, saya,, edy, edy, rahmayadi, edy, edy...
3	[Yg, pasti, GUBSU, bukan, orang, sombong]
4	[Cagubpun, ntidakk, laku, lagi, edi]
...	...
642	[Partai, yang, selalu, teguh, pada, UUD, 45......
643	[Ia,, maklumlah, Bang,, partai, haus, kekuasaa...
644	[Apa, lagi, pdpi, p..partai, pingin, menang, s...
645	[Tenggelam, kan, PDIP...partai, sombong...part...
646	[Semotidak, menang, ya, bapak, Edi...., Menyal...

Fig.5. Code Snippet and Results from Tokenization

The tokenization process breaks the text into smaller words or phrases. This is important because sentiment analysis algorithms process text at the word level, and the results of tokenization affect how the algorithm identifies sentiment in comments.

4. Steaming

```
# Stemming
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

def stemming(text_cleaning):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in text_cleaning:
        dt = stemmer.stem(w)
        do.append(dt)
    d_clean = []
    d_clean = " ".join(do)
    print(d_clean)
    return d_clean
```

Fig.6. Code snappet steaming

Stemming simplifies sentiment analysis by reducing the variation of words that essentially have the same meaning. This helps to improve the accuracy of the Naïve Bayes model in classifying sentiment.

4.4 Translate

```
df['textDisplay'] = df['textDisplay'].astype(str)
from deep_translator import GoogleTranslator
def convert_eng(tweet):
    if isinstance(tweet, str):
        translator = GoogleTranslator(source='auto', target='en')
        return translator.translate(tweet)
    return tweet

df['tweet_english'] = df['textDisplay'].apply(convert_eng)

# Simpan hasil terjemahan ke file CSV
df_sample = df.sample(n=500, random_state=42)
df_sample.to_csv("/content/terjemahan_sample.csv", index=False)
```

```
[47] print(df.columns)
     df.info()
```

```
Index(['textDisplay', 'tweet_english'], dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 904 entries, 0 to 903
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   textDisplay    904 non-null    object
 1   tweet_english  900 non-null    object
dtypes: object(2)
memory usage: 14.2+ KB
```

Fig.7. Code for Translation

	textDisplay	tweet_english
0	['Saya', 'yakin', 'calon', 'Gubernur', 'dari',...	['I', 'sure', 'candidate', 'Governor', 'from',...

Fig.8. Example of Results

Figures 8 and 9 show the process and results of translating comment texts from Indonesian to English using Python libraries such as Google Translator. This step is important to ensure that sentiment analysis using TextBlob, which works better with English texts, is consistent with the results from the Naïve Bayes algorithm.

4.5 Implementation of the Naïve Bayes Algorithm

The Naïve Bayes algorithm was used for sentiment analysis by training the model on a labeled dataset. The classification results show the distribution of positive, negative, and neutral sentiments in the dataset. Model evaluation was performed based on precision, recall, F1-score, and support metrics, providing an overview of the model's accuracy and balance in classifying sentiment.

```
Evaluasi Model Naive Bayes:

              precision    recall  f1-score   support
Positif          0.58       0.76      0.66        29
Negatif          0.65       0.76      0.21        50
Netral           0.86       0.76      0.81        66

accuracy                              0.72       100
macro avg        0.48       0.51      0.49       100
weighted avg     0.74       0.72      0.72       100

Akurasi Model: 0.72
```

Fig.9. Evaluation of the Naive Bayes model.

1. Boby Nasution
    a. Evaluation details:
- Positif**:** Precision 0.58, Recall 0.76, F1-Score 0.66, Support 29.
- Negatif**:** Precision 0.65, Recall 0.76, F1-Score 0.21, Support 50.
- Netral**:** Precision 0.86, Recall 0.76, F1-Score 0.81, Support 66.

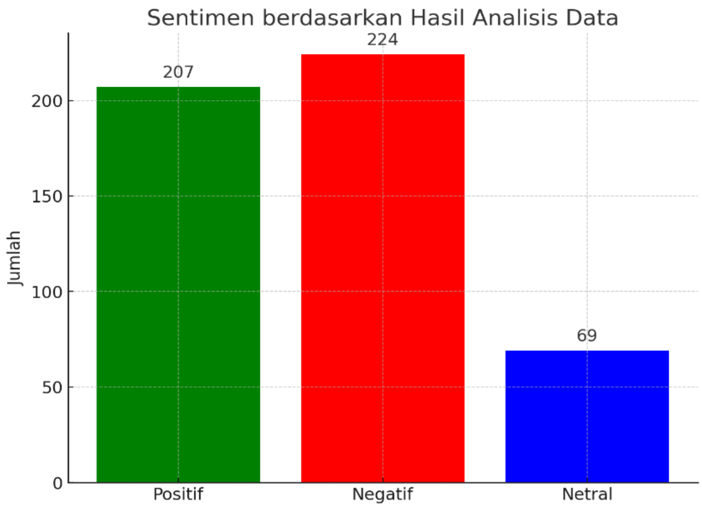


Fig.10. Sentiment based on data analysis results.

The model has an overall accuracy of 72%, with the highest precision in the neutral category. However, the F1-Score for the negative category is low, indicating that the model struggles to capture all truly negative instances. The bar chart shows that negative sentiment dominates, followed by positive, while neutral has the least amount.

b. Analysis and Explanation:

Positive: High recall, lower precision, indicating many false positives.

Negative: Better precision, but a low F1-Score, indicating difficulty in accurately detecting negative sentiment.

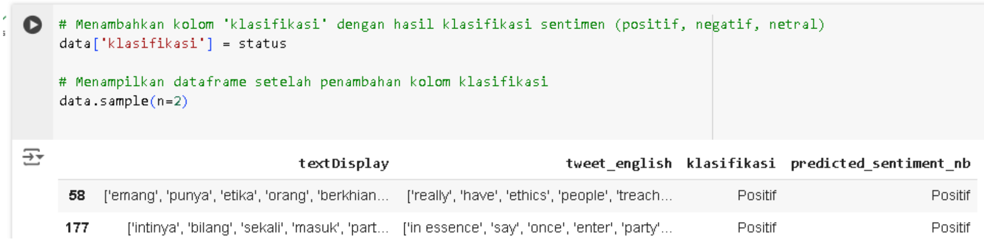
Neutral: Best performance in precision, recall, and F1-Score, indicating the model's reliability in detecting neutral sentiment.

c. Mathematics Behind the Evaluation:

- The F1-Score for Positive is calculated as:

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} = 2 \times \frac{(0.58 \times 0.76)}{(0.58 + 0.76)} \approx 0.66$$

- The weighted average for the F1-Score is 0.72, indicating a fairly good overall performance of the model. However, for more general results, further evaluation with a larger and more diverse dataset is required.



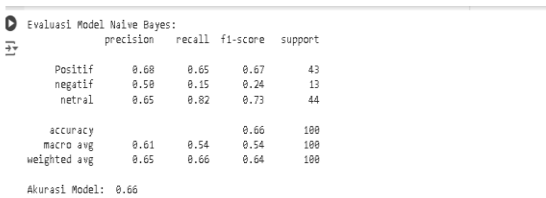
```
# Menambahkan kolom 'klasifikasi' dengan hasil klasifikasi sentimen (positif, negatif, netral)
data['klasifikasi'] = status

# Menampilkan dataframe setelah penambahan kolom klasifikasi
data.sample(n=2)
```

	textDisplay	tweet_english	klasifikasi	predicted_sentiment_nb
58	['emang', 'punya', 'etika', 'orang', 'berkhian...	['really', 'have', 'ethics', 'people', 'treach...	Positif	Positif
177	['intinya', 'bilang', 'sekali', 'masuk', 'part...	['in essence', 'say', 'once', 'enter', 'party'...	Positif	Positif

Fig.11. Example of Sentiment Classification Results

2. Edy Rahmayadi



```
Evaluasi Model Naive Bayes:
              precision    recall  f1-score   support

     Positif       0.68      0.65      0.67        43
     negatif       0.50      0.15      0.24        13
      netral       0.65      0.82      0.73        44

    accuracy                           0.66       100
   macro avg       0.61      0.54      0.54       100
weighted avg       0.65      0.66      0.64       100

Akurasi Model:  0.66
```

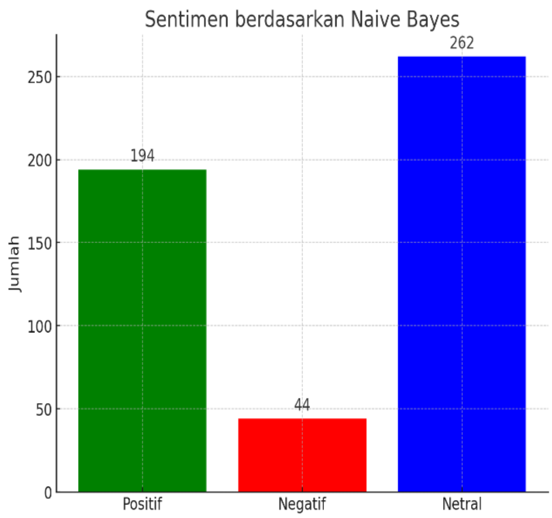


Fig.11. Bar chart

a. Evaluation details:

- Positive: Precision 0.68, Recall 0.76, F1-Score 0.67, Support 43.
- Negative: Precision 0.50, Recall 0.15, F1-Score 0.24, Support 14.
- Neutral: Precision 0.65, Recall 0.81, F1-Score 0.73, Support 44.

The model's overall accuracy is 66%. The bar chart shows that the majority of the data is classified as neutral, followed by positive, with negative being the least. This is consistent with the evaluation, which indicates low recall for the negative category. The overall performance shows that the model is better at classifying neutral and positive sentiments but still has weaknesses in detecting negative sentiment.

```
# Menambahkan kolom 'nb' dengan hasil klasifikasi sentimen (positif, negatif, netral)
data['predicted_sentimen_nb'] = status

# Menampilkan dataframe setelah penambahan kolom klasifikasi
data.sample(n=3)
```

	textDisplay	tweet_english	klasifikasi	predicted_sentiment_nb
472	['mantap', 'ayah', 'edi', 'edy', 'rahmayadi', ...	['steady', 'father', 'edi', 'edy', 'rahmayadi'...	Positif	Positif
305	['acara', 'ini', 'hanya', 'mentidakdu', 'kerja...	['event', 'this', 'only', 'not to worry', 'wor...	Positif	Positif
352	['lanjutkan']	['continue']	netral	Positif

Fig.13. Example of sentiment classification results

4.6 Implementation of TextBlob

In addition to Naïve Bayes, TextBlob was also used to analyze sentiment from texts translated into English. TextBlob provides sentiment polarity as positive, neutral, or negative. These results are then compared with those from Naïve Bayes to evaluate the accuracy and effectiveness of the method.

1. Boby Nasution

```
# Menambahkan kolom 'klasifikasi' dengan hasil klasifikasi sentimen (positif, negatif, netral)
data['klasifikasi'] = status

# Menampilkan dataframe setelah penambahan kolom klasifikasi
data.sample(n=5)
```

	textDisplay	tweet_english	klasifikasi
179	['wajarlah', 'kalau', 'Partai', 'Demokrasi', '...	['natural', 'if', 'Party', 'Democracy', 'Indon...	Positif
217	['bagus', 'lah', 'istiqomah', 'orang', 'mbegru...	['good', 'lah', 'istiqomah', 'people', 'mbegru...	Positif
102	['wajah', 'penuh', 'duka', 'lara']	['face', 'full', 'sorrow', 'pain']	Positif
59	['apakah', 'golkar', 'tidak', 'punya', 'kader'...	['whether', 'golkar', 'not', 'have', 'cadre', ...	netral
381	['pdip', 'sakit', 'hati', 'hancur', 'rakyat', ...	['pdip', 'sick', 'heart', 'destroyed', 'people...	negatif

Fig.14. Example of Random Results

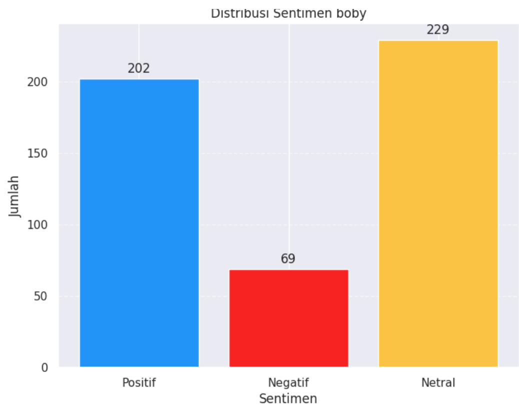


Fig.15. Bar Chart of Sentiment Analysis for Boby

The bar chart shows the sentiment distribution toward Boby Nasution analyzed using TextBlob, where most comments are neutral and positive, indicating a less negative public perception. This visualization helps in understanding public perception more intuitively.

Figure 16, on the other hand, shows an example of random sentiment classification results, providing a more detailed view of various public opinions about this candidate. Meanwhile, this figure shows the words that most frequently appear in the analyzed comments. These words give a general picture of the dominant topics and themes in public opinion about the gubernatorial candidate studied.

2. Edy Rahmayadi

```
# Menambahkan kolom 'klasifikasi' dengan hasil klasifikasi sentimen (positif, negatif, net
data['klasifikasi'] = status

# Menampilkan dataframe setelah penambahan kolom klasifikasi
data.sample(n=3)
```

	textDisplay	tweet_english	klasifikasi
435	['simak', 'donk']	['listen', 'donk']	netral
410	['kader', 'Partai', 'Demokrasi', 'Indonesia', ...	['cadre', 'Party', 'Democracy', 'Indonesia', '...	Positif
380	['hahh', 'tros', 'mau', 'gimana']	['hahh', 'tros', 'want', 'how']	netral

Fig.16. Example of Random Results

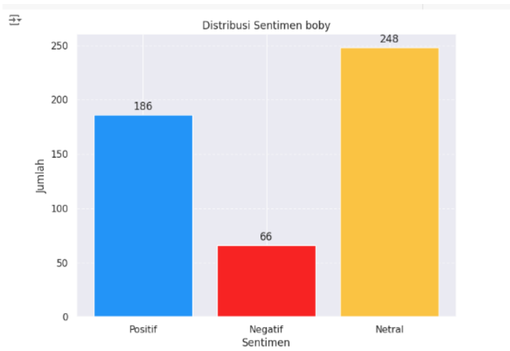


Fig.17. Bar Chart of Sentiment Analysis for Edy

The bar chart for Edy Rahmayadi also shows the dominance of negative sentiment. This analysis indicates strong public criticism of Edy Rahmayadi. The data shows how public perception of Edy Rahmayadi varies on YouTube channels, with visualization helping to identify the main issues discussed in public opinion about this candidate. Figure 19 shows an example of his sentiment classification results.

Meanwhile, the image beside shows the words that most frequently appear in the comments analyzed related to Edy Rahmayadi. These words provide insights into the main issues discussed in public opinion about this candidate.

5. Conclusion

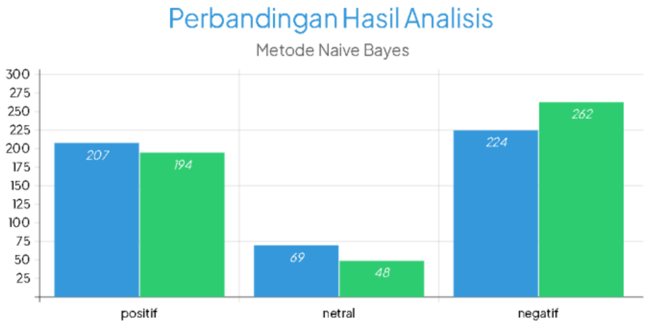


Fig.18. Results of Analysis Using the Naïve Bayes Method

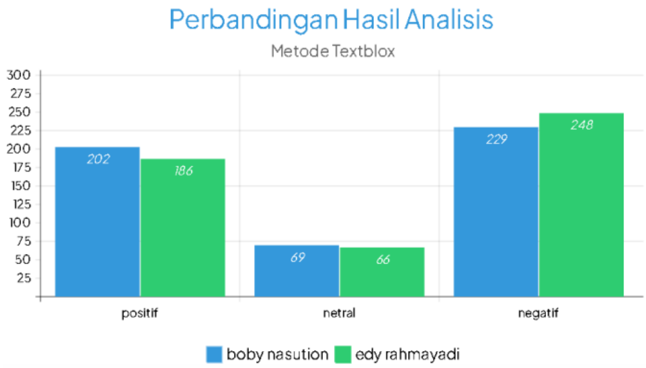


Fig.19. Results of Analysis Using the TextBlob Method

In this study, we analyzed public sentiment toward the 2024 North Sumatra gubernatorial candidates, Boby Nasution and Edy Rahmayadi, based on YouTube comments. We used two analysis methods, Naïve Bayes and TextBlob.

The results from Naïve Bayes show that Boby Nasution received 207 positive comments, 69 neutral, and 224 negative, while Edy Rahmayadi received 194 positive, 48 neutral, and 262 negative. With TextBlob, Boby received 202 positive comments, 69 neutral, and 229 negative, while Edy received 186 positive, 66 neutral, and 248 negative.

Overall, both methods show the dominance of negative sentiment, especially toward Edy Rahmayadi. Although Boby slightly outperforms in positive and neutral sentiment, significant public criticism of both candidates highlights the need for better campaign strategies.

References

[1] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135.

[2] Hasan, F. N., & Dwijayanti, M. (2021). Analisis Sentimen Ulasan Pelanggan Terhadap Layanan Grab Indonesia Menggunakan Multinominal Naïve Bayes Classifier. Jurnal Sistem Informasi, 4(1), 239-248.

[3] Aldisa, R. T., & Maulana, P. (2022). Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi COVID-19 Dengan Perbandingan Metode Naïve Bayes, Decision Tree dan SVM. Jurnal Technologia, 4(1), 106-109.

[4] Hardhika, R. E. B., & Huda, A. M. (2021). Pengalaman Pengguna Paylater Mahasiswa di Surabaya. Jurnal Bisnis dan Investasi, 7(1), 44-57.

[5] Afandi, I. R., Hanif, I. F., Hasan, F. N., & Sinduningrum, E. (2022). Analisis Sentimen Opini Masyarakat Terkait Sistem Elektronik Menggunakan Metode Logistic Regression. Jurnal Teknologi dan Komputer, 5(2), 77-84.

[6] Wibowo, A., & Noor Hasan, F. (2022). Analisis Sentimen Pembelajaran Daring Selama Pandemi COVID-19 Menggunakan Naïve Bayes Classifier. Jurnal Ilmiah Rekayasa Inovatif, 4(1), 239-248.

[7] Ahmad, T., Rahman, M., & Khan, S. (2019). Impact of Social Media on Political Campaigning in Malaysia: An Analysis of the 14th General Election. Asian Journal of Social Science, 47(2), 127-145.

[8] Afandi, I. R., Hanif, I. F., Hasan, F. N., & Sinduningrum, E. (2022). Analisis Sentimen Opini Masyarakat Terkait Sistem Elektronik Menggunakan Metode Logistic Regression. Jurnal Teknologi dan Komputer, 5(2), 77-84.

[9] Wati, R., & Ernawati, S. (2021). Analisis Sentimen Persepsi Publik Mengenai PPKM Pada Twitter Berbasis SVM Menggunakan Python. Jurnal Teknologi dan Komputer, 3(1), 45-54.

[10] AlRyalat, S. A. S., Malkawi, L. W., & Momani, S. M. (2017). Comparing Bibliometric Analysis Using PubMed, Scopus, and Web of Science Databases. Journal of Visualized Experiments, (118), 1-14. doi:10.3791/58494.

[11] Koto, F., Rahimi, A., Louvan, S., & Baldwin, T. (2020). IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Preprocessing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 1074-1086. doi:10.18653/v1/2020.findings-emnlp.96.

[12] Loria, S. (2018). TextBlob: Simplified Text Processing. TextBlob Documentation. Retrieved from https://textblob.readthedocs.io/en/dev/

[13] Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135. doi:10.1561/1500000011.

[14] Heimerl, F., Lohmann, S., Lange, S., & Ertl, T. (2014). Word Cloud Explorer: Text Analytics Based on Word Clouds. Proceedings of the 47th Hawaii International Conference on System Sciences, 1833-1842. doi:10.1109/HICSS.2014.231.