Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Public Sentiment Analysis on Facebook Posts About Shin Tae-Yong's Dismissal Using the K-Nearest Neighbors Algorithm

Syahidan[1]*, Relita Buaton[2], Muammar Khadapi[3]

[1,2,3] *STMIK KAPUTAMA BINJAI*

10syahidan@gmail.com[1], bbcbuaton@gmail.com[2], khdafi5@gmail.com[3]*

Abstract

The sacking of Indonesian national team coach Shin Tae-yong by PSSI on January 6, 2025 sparked massive public attention on various social media platforms, one of which was Facebook. The large volume of unstructured opinions required analysis to accurately understand public perception. This study aims to classify public sentiment toward the news of Shin Tae-yong's dismissal using the K-Nearest Neighbors (K-NN) machine learning method. The data used consists of public comments from Facebook, processed through a series of text preprocessing steps and word weighting using TF-IDF. The K-NN model was tested with a value of k = 80. The results show that the classification model achieved an overall accuracy rate of 76%. While the model performed well for positive and negative sentiment classes, its performance was very weak in identifying neutral sentiment (recall 0.02). The sentiment distribution results indicate that public opinion is dominated by positive sentiment at 56.5%, followed by negative sentiment (29.3%), and neutral sentiment (14.2%). The main finding of this study, which contradicts common assumptions, is that the public response on Facebook to this policy is predominantly positive.

Keywords: *Sentiment Analysis, K-Nearest Neighbors, Shin Tae-yong, PSSI, Facebook*

1. Introduction

n the digital age, social media platforms such as Facebook have become the main arena for the public to voice their opinions on a massive scale. These opinions are closely related to the attitudes of individuals and groups [1], often arise in response to policies or important events that are in the spotlight. One event that sparked widespread public discourse in Indonesia was the decision by the Indonesian Football Association (PSSI) to dismiss Shin Tae-yong (STY) from his position as head coach of the Indonesian national team on January 6, 2025 [2].The policy announced at the press conference, based on internal conflicts and evaluation of match results, suddenly became a topic that was built up and spread by the media [3]. The Facebook platform was flooded with various reactions from supporters and the general public, who were polarized between pros and cons. This wave of comments generated a large volume of opinion data that reflected the true public perception, requiring an accurate analysis process to understand the essence of these sentiments [4].

To process this unstructured textual data, this study applied sentiment analysis, a field of study that analyzes public opinion, sentiment, and emotions towards an entity or event [5]. The main objective of sentiment analysis is to classify texts into positive, negative, or neutral sentiments [6]. In this study, the classification method implemented is the K-Nearest Neighbors (K-NN) machine learning algorithm..
The K-NN method was chosen for its effectiveness in grouping objects based on the proximity or similarity of training data [7]. Using the Python programming language and text mining libraries, this study aims to classify public sentiment on Facebook regarding Shin Tae-yong's dismissal. The classification results will map in-depth insights into public perceptions and opinion trends on this issue.

Previous studies have shown that the K-Nearest Neighbors (K-NN) method is an effective and reliable algorithm for sentiment analysis on social media data in Indonesia. For example, a study by Rizqi Irawan (2022) analyzing sentiment towards Gojek services on Twitter achieved an accuracy of 79.43% using K-NN [8]. Similar success was also demonstrated by Furqan and colleagues (2022) in analyzing public sentiment towards the New Normal policy on Twitter, which achieved a higher accuracy of 94.50% [9]. In another study, namely reviews of e-commerce applications on Google Play Store, Kusuma & Cahyono (2023) also proved the effectiveness of K-NN in classifying sentiment towards Shopee with an accuracy of 82% [10]. All three studies applied standard text preprocessing steps and TF-IDF word weighting, concluding that K-NN is a solid method for text classification.

Based on these findings, this study adopts a proven methodology by applying the K-NN algorithm for sentiment classification. However, the hope of this study lies in the object and context of the analysis. While previous studies have focused on the commercial service sector (Gojek, Shopee) and large-scale government policies (New Normal), this study will fill the gap by analyzing public sentiment towards the dismissal of a public figure in the world of sports, namely the coach of the Indonesian national team, Shin Tae-yong. Thus, this study aims

to test the effectiveness of the K-NN method in a more specific and dynamic domain, involving the emotional sentiments of supporters towards managerial issues in national sports organizations.

2. Metedelogi Penelitian

2.1. Cross Industry Standard for Data mining CRIPS-DM

The approach used in this study employs the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. CRISP-DM is one of the most popular comprehensive approaches in data science and data mining projects since its introduction about two decades ago. The use of this methodology is highly relevant because of its focus on a deep understanding of business processes and data characteristics, resulting in a more accurate interpretation of the problems [11]. The main objective of CRISP-DM is to recommend specific solutions that can optimize an organization's performance [12].

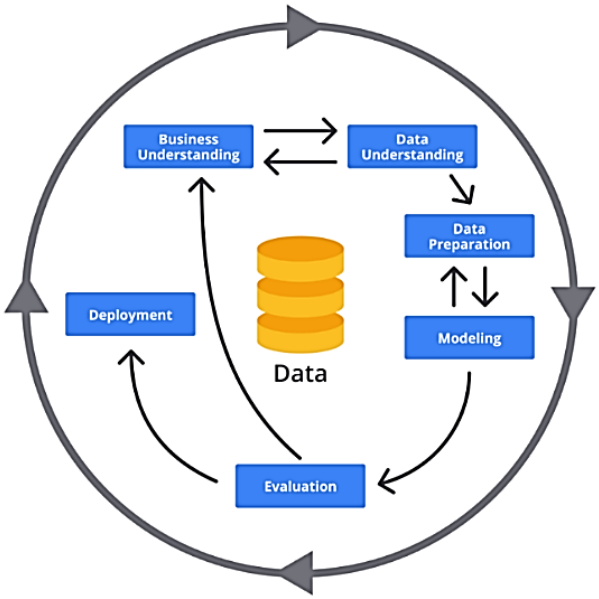

Fig. 1: CRIPS-DM

This methodology consists of six main sequential stages [13]. The process begins with the Business Understanding stage, where business objectives and problem definitions are established to develop an initial project plan. This stage is followed by Data Understanding, which includes collecting, exploring, and examining data quality from its source. Once the data is understood, the Data Preparation stage is carried out, which includes cleaning and selecting data according to criteria relevant to modeling.
The core phase of this process is Modeling, where data mining techniques are selected and applied to build predictive models based on the prepared data. The built models are then tested in the Evaluation stage to check the suitability of the results with the business objectives set at the beginning. The final stage is Deployment, which includes the implementation of the final project results in the form of reports or software components, as well as planning for ongoing monitoring and maintenance.

2.2. Research Method

This study applies sentiment analysis, which is one of the processes in text mining for computational studies of user opinions, sentiments, and emotions, with the aim of classifying textual information into positive or negative categories [14]. Sentiment analysis is a crucial part of Natural Language Processing (NLP) for understanding public opinion developing on social media [15]. NLP itself is a field of artificial intelligence that enables computers to process and understand complex and ambiguous human language [16]. This process falls within the scope of Text Mining, a method for exploring and analyzing unstructured text data to identify previously unknown patterns or valuable information [17].
Data collection in this study was conducted using web scraping techniques, which is the process of automatically extracting data from web pages to obtain large-scale structured data [18][19]. The web scraping process consists of three main phases: data retrieval using the HTTP protocol, information extraction from HTML documents, and data transformation into structured formats such as CSV or JSON [20].
The collected text data then went through a text pre-processing stage to make it more structured and of higher quality before further processing [21]. This stage was necessary to address issues such as data inaccuracy and redundancy [22]. The process involved the following steps:

1. Cleaning (removing non-alphabetic characters)
2. Case Folding (converting text to lowercase)
3. Normalization (handling non-standard spelling)
4. Tokenizing (breaking sentences into words)
5. Stopword Removal (eliminating common words)
6. Stemming (converting words to their base form).

Examples of text pre-processing:

Table 1: Text Pre-Processing

Step	Result
Comment source	G tau terimakasih nih,,, STY sdh bawa nama baik Indo,, d kancah internasional, Sdh susah payah,,, malah d pecat,,, Manusia kacang lupa kulit nih mmg,,, ada apa Erik Tohir,,,,????
Cleaning	G tau terimakasih nih STY sdh bawa nama baik Indo d kancah internasional Sdh susah payah malah d pecat Manusia kacang lupa kulit nih mmg ada apa Erik Tohir
SCase Folding	g tau terimakasih nih sty sdh bawa nama baik indo d kancah internasional sdh susah payah malah d pecat manusia kacang lupa kulit nih mmg ada apa erik tohir
Normalization	Tidak tahu terima kasih nih sty sudah bawa baik indonesia di kancah internasional sudah susah payah malah di pecat manusia kacang lupa kulit nih memang ada apa erik tohir
Tokenizing	["tidak", "tahu", "terima", "kasih", "nih", "sty", "sudah", "bawa", "baik", "indonesia", "di", "kancah", "internasional", "sudah", "susah", "payah", "malah", "di", "pecat", "manusia", "kacang", "lupa", "kulit", "nih", "memang", "ada", "apa", "erik", "tohir"]
Stopword Removal	["tidak", "tahu", "terima", "kasih","sty", "bawa", "baik", "indonesia", "kancah", "internasional", "susah", "payah", "malah", "pecat", "manusia", "kacang", "lupa", "kulit", "memang", "erik", "tohir"]
Stemming	["tidak", "tahu", "terima", "kasih","sty", "bawa", "baik", "indonesia", "kancah", "internasional", "susah", "payah", "malah", "pecat", "manusia", "kacang", "lupa", "kulit", "memang", "erik", "tohir"]

After cleaning the data, textual data was transformed into numerical format using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This technique weights each word based on its frequency of occurrence in a document relative to its frequency in the entire document collection [23]. If the occurrence of a word has a high frequency, then that word has the potential to become a keyword [24].
The next stage of this research is classification, a machine learning technique that aims to predict the class or label of data based on its features [25]. In this process, a model is built by analyzing training data to map each feature to a predetermined class label [26][27]. The classification algorithm used is K-Nearest Neighbors (K-NN), a non-parametric supervised learning method that groups new data based on its closest distance to the data in the training set [28]. The measurement of distance between data is done using one of the metrics such as Euclidean distance [29].
To evaluate the performance of the K-NN classification model, a Confusion Matrix is used. This matrix is the basis for calculating performance metrics such as precision, recall, accuracy, and F1-Score to draw conclusions from the research results.

3. Results and Discussion

3.1. Import Library

The first step is to import all the necessary libraries. These libraries have functions for performing text preprocessing, processing data, running machine learning algorithms, and creating visualizations. That way, the sentiment analysis process from start to finish can run efficiently and in a structured manner.

```
import re
import nltk
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

nltk.download('punkt_tab')
nltk.download('stopwords')
!pip install Sastrawi
!pip install textblob

from tqdm.notebook import tqdm
from textblob import TextBlob
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Collecting Sastrawi
  Downloading Sastrawi-1.0.1-py2.py3-none-any.whl.metadata (909 bytes)
Downloading Sastrawi-1.0.1-py2.py3-none-any.whl (209 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.7/209.7 kB 4.1 MB/s eta 0:00:00
Installing collected packages: Sastrawi
Successfully installed Sastrawi-1.0.1
Requirement already satisfied: textblob in /usr/local/lib/python3.12/dist-packages (0.19.0)
Requirement already satisfied: nltk>=3.9 in /usr/local/lib/python3.12/dist-packages (from textblob) (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob) (1.5.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob) (4.67.1)
```

Fig. 2: Source code Import Library

3.2. Entering data

The first step is to import all the necessary libraries. These libraries have functions for performing text preprocessing, processing data, and running algorithms.

```
df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Skripsi aku/dokumen-dan-kamus/dataset-komentar-facebook.xlsx')
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2362 entries, 0 to 2361
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   username  2362 non-null   object
 1   comment   2362 non-null   object
 2   time      2310 non-null   object
dtypes: object(3)
memory usage: 55.5+ KB
```

	username	comment	time
0	Galuh Panjalu	Minimal selsaikan dulu lah rencana sty di roun...	25w
1	Feri Peye Andrian	Yuk rame2 bikin demo online\nBikin hastag apat...	25w
2	Suro Tamping Gramang	Tunggu timnas Garuda kalah Erik Tohir dan pung...	25w
3	Kamal CeNat CenOt	Udah bagus2 sty melatih timnas.. Eh kok di pecat	25w
4	Dilar Bahtiar	Kecewa ama pssi blum juga timnas beres dah dignti	25w

Fig. 3: Source code and results of dataset calls

From the image above, it can be seen that the amount of data from the dataset that was successfully retrieved or imported amounted to 2362 rows.

3.3. Data Pre-processing

The initial stage of analysis is data cleaning. This process aims to ensure the validity of each comment as a unique entry, while minimizing potential bias in the classification results. Next, the data will go through a series of text preprocessing steps, including:

1. Cleaning

The process of deleting text data that does not contain letters, such as punctuation marks, special characters, hashtags, symbols, and emojis.

```
def cleaning(text):
    # Hapus URL
    text = re.sub(r'http\S+|www.\S+', '', text)
    # Hapus mention dan hashtag
    text = re.sub(r'[@#]\w+', '', text)
    # Hapus angka
    text = re.sub(r'\d+', '', text)
    # Hapus karakter non-huruf (tanda baca, simbol, emoji)
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    # Hapus spasi berlebih
    text = re.sub(r'\s+', ' ', text).strip()
    return text

df['cleaning'] = df['comment'].apply(cleaning)

# Menampilkan beberapa hasil
df[['comment', 'cleaning']].head()
```

	comment	cleaning
0	Minimal selsaikan dulu lah rencana sty di roun...	Minimal selsaikan dulu lah rencana sty di roun...
1	Yuk rame2 bikin demo online\nBikin hastag apat...	Yuk rame bikin demo online Bikin hastag apatah...
2	Tunggu timnas Garuda kalah Erik Tohir dan pung...	Tunggu timnas Garuda kalah Erik Tohir dan pung...
3	Udah bagus2 sty melatih timnas.. Eh kok di pecat	Udah bagus sty melatih timnas Eh kok di pecat
4	Kecewa ama pssi blum juga timnas beres dah dignti	Kecewa ama pssi blum juga timnas beres dah dignti

Fig. 4: Source code and results of cleaning

2. Case Folding

All words and letters (alphabets) containing capital letters in the data are converted to lowercase letters..

```python
def case_folding(text):
    # Ubah semua huruf menjadi lowercase
    text = text.lower()
    return text

df['case_folding'] = df['cleaning'].apply(case_folding)

# Menampilkan beberapa hasil
df[['cleaning', 'case_folding']].head()
```

	cleaning	case_folding
0	Minimal selsaikan dulu lah rencana sty di roun...	minimal selsaikan dulu lah rencana sty di roun...
1	Yuk rame bikin demo online Bikin hastag apatah...	yuk rame bikin demo online bikin hastag apatah...
2	Tunggu timnas Garuda kalah Erik Tohir dan pung...	tunggu timnas garuda kalah erik tohir dan pung...
3	Udah bagus sty melatih timnas Eh kok di pecat	udah bagus sty melatih timnas eh kok di pecat
4	Kecewa ama pssi blum juga timnas beres dah dignti	kecewa ama pssi blum juga timnas beres dah dignti

Fig. 5: Source code and results of Case Folding

3. Tokenizing

The process of dividing text into smaller parts, from sentences into individual words or tokens.

```python
def tokenizing(text):
    tokens = word_tokenize(text)
    return tokens

df['tokenizing'] = df['case_folding'].apply(tokenizing)

# Menampilkan beberapa hasil
df[['case_folding', 'tokenizing']].head()
```

	case_folding	tokenizing
0	minimal selsaikan dulu lah rencana sty di roun...	[minimal, selsaikan, dulu, lah, rencana, sty, ...
1	yuk rame bikin demo online bikin hastag apatah...	[yuk, rame, bikin, demo, online, bikin, hastag...
2	tunggu timnas garuda kalah erik tohir dan pung...	[tunggu, timnas, garuda, kalah, erik, tohir, d...
3	udah bagus sty melatih timnas eh kok di pecat	[udah, bagus, sty, melatih, timnas, eh, kok, d...
4	kecewa ama pssi blum juga timnas beres dah dignti	[kecewa, ama, pssi, blum, juga, timnas, beres,...

Fig. 6: Source code and results of Tokenizing

4. Normalization

A form of word normalization that maps data into intervals to handle non-standard spelling, abbreviations, and numbers.

```python
# Membaca file kamus normalisasi
kamus_normalisasi = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Skripsi aku/dokumen-dan-kamus/kamuskatabaku.xlsx')

# Ubah ke dictionary untuk mempercepat lookup
normalisasi_dict = dict(zip(kamus_normalisasi['tidak_baku'], kamus_normalisasi['kata_baku']))

def normalisasi(tokens):
    hasil_normalisasi = [normalisasi_dict.get(token, token) for token in tokens]
    return hasil_normalisasi

df['normalisasi'] = df['tokenizing'].apply(normalisasi)

# Menampilkan beberapa hasil
df[['tokenizing', 'normalisasi']].head()
```

	tokenizing	normalisasi
0	[minimal, selsaikan, dulu, lah, rencana, sty, ...	[minimal, selesaikan, dulu, lah, rencana, sty,...
1	[yuk, rame, bikin, demo, online, bikin, hastag...	[yuk, ramai, buat, demo, online, buat, hastag,...
2	[tunggu, timnas, garuda, kalah, erik, tohir, d...	[tunggu, timnas, garuda, kalah, erick, thohir,...
3	[udah, bagus, sty, melatih, timnas, eh, kok, d...	[sudah, bagus, sty, melatih, timnas, eh, kok, ...
4	[kecewa, ama, pssi, blum, juga, timnas, beres,...	[kecewa, sama, pssi, belum, juga, timnas, bere...

Fig. 7: Source code and results of Normalization

5. Stopword Removal

The process of dividing text into smaller parts, from sentences into individual words or tokens.

```
[9] # Membaca file stopword
with open('/content/drive/MyDrive/Colab Notebooks/Skripsi aku/dokumen-dan-kamus/stopwords.txt', 'r') as f:
    stopwords = f.read().splitlines()

def stopword_text(hasil_normalisasi):
    cleaned_tokens = []
    for token in hasil_normalisasi:
        if token not in stopwords:
            cleaned_tokens.append(token)
    return cleaned_tokens

df['stopword_removal'] = df['normalisasi'].apply(stopword_text)
df.head()
```

	username	comment	time	cleaning	case_folding	tokenizing	normalisasi	stopword_removal
0	Galuh Panjalu	Minimal selsaikan dulu lah rencana sty di roun...	25w	Minimal selsaikan dulu lah rencana sty di roun...	minimal selsaikan dulu lah rencana sty di roun...	[minimal, selsaikan, dulu, lah, rencana, sty, ...	[minimal, selesaikan, dulu, lah, rencana, sty,...	[minimal, selesaikan, rencana, sty, rounde, pi...
1	Feri Peye Andrian	Yuk rame2 bikin demo online\nBikin hastag apat...	25w	Yuk rame bikin demo online Bikin hastag apatah...	yuk rame bikin demo online bikin hastag apatah...	[yuk, rame, bikin, demo, online, bikin, hastag...	[yuk, ramai, buat, demo, online, buat, hastag,...	[ramai, demo, online, hastag, pssi, apalahmasa...
2	Suro Tamping Gramang	Tunggu timnas Garuda kalah Erik Tohir dan pung...	25w	Tunggu timnas Garuda kalah Erik Tohir dan pung...	tunggu timnas garuda kalah erik tohir dan pung...	[tunggu, timnas, garuda, kalah, erik, tohir, d...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...
3	Kamal CeNat CenOt	Udah bagus2 sty melatih timnas.. Eh kok di pecat	25w	Udah bagus sty melatih timnas Eh kok di pecat	udah bagus sty melatih timnas eh kok di pecat	[udah, bagus, sty, melatih, timnas, eh, kok, d...	[sudah, bagus, sty, melatih, timnas, eh, kok, ...	[bagus, sty, melatih, timnas, pecat]
4	Dilar Bahtiar	Kecewa ama pssi blum juga timnas beres dah dignti	25w	Kecewa ama pssi blum juga timnas beres dah dignti	kecewa ama pssi blum juga timnas beres dah dignti	[kecewa, ama, pssi, blum, juga, timnas, beres,...	[kecewa, sama, pssi, belum, juga, timnas, bere...	[kecewa, pssi, belum, timnas, beres, dignti]

Fig. 8: Source code and results of stopword removal

6. Stemming

The process of eliminating words that are less relevant to the sentiment object, such as conjunctions, for example “tetapi, dengan, untuk, yang, adapun”, and other connecting words.

```
tqdm.pandas()

# Membuat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Fungsi stemming
def stemming(cleaned_tokens):
    return [stemmer.stem(token) for token in cleaned_tokens]

df['stemming'] = df['stopword_removal'].progress_apply(stemming)
df[['stopword_removal', 'stemming']].head()
```

```
100% 2355/2355 [05:59<00:00, 16.26it/s]
```

	stopword_removal	stemming
0	[minimal, selesaikan, rencana, sty, rounde, pi...	[minimal, selesai, rencana, sty, rounde, pildu...
1	[ramai, demo, online, hastag, pssi, apalahmasa...	[ramai, demo, online, hastag, pssi, apalahmasa...
2	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...
3	[bagus, sty, melatih, timnas, pecat]	[bagus, sty, latih, timnas, pecat]
4	[kecewa, pssi, belum, timnas, beres, dignti]	[kecewa, pssi, belum, timnas, beres, dignti]

Fig. 9: Source code and results of Stemming

7. Dictionary Filter (based on a dictionary)

The dictionary filter process is carried out to eliminate irrelevant words based on the available language dictionary and scopus. This stage is an additional option for writers and is important to implement as a word filtering technique. This technique is the opposite of the commonly used stopword removal method. If stopword removal functions as a blacklist to remove common words, then the dictionary filter functions as a whitelist to only allow words that are considered valid.

This process works by matching each token from the cleaned text with a reference dictionary. This dictionary contains a collection of Indonesian and English vocabulary, as well as names of figures relevant to the scope of the research. Words found in this dictionary will be retained, while words not found in it will be removed.

```
[12] with open('/content/drive/MyDrive/Colab Notebooks/Skripsi aku/dokumen-dan-kamus/korpus-indo-inggris-saya.txt', 'r', encoding='utf-8') as f:
        kamus_corpus = list(f.read().splitlines())

    def filtering_text(stemmer):
        filtering = []
        for token in stemmer:
            if token in kamus_corpus:
                filtering.append(token)
        return filtering

    df['filter_text'] = df['stemming'].apply(filtering_text)
    df.head()
```

	username	comment	time	cleaning	case_folding	tokenizing	normalisasi	stopword_removal	stemming	filter_text
0	Galuh Panjalu	Minimal selsaikan dulu lah rencana sty di roun...	25w	Minimal selsaikan dulu lah rencana sty di roun...	minimal selsaikan dulu lah rencana sty di roun...	[minimal, selsaikan, dulu, lah, rencana, sty, ...	[minimal, selesaikan, dulu, lah, rencana, sty,...	[minimal, selesaikan, rencana, sty, rounde, pi...	[minimal, selesai, rencana, sty, rounde, pildu...	[minimal, selesai, rencana, sty, pildun, gagal...
1	Feri Peye Andrian	Yuk rame2 bikin demo online\nBikin hastag apat...	25w	Yuk rame bikin demo online Bikin hastag apatah...	yuk rame bikin demo online bikin hastag apatah...	[yuk, rame, bikin, demo, online, bikin, hastag...	[yuk, ramai, buat, demo, online, buat, hastag,...	[ramai, demo, online, hastag, pssi, apalahmasa...	[ramai, demo, online, hastag, pssi, apalahmasa...	[ramai, demo, pssi, benci, coach, sty, menang,...
2	Suro Tamping Gramang	Tunggu timnas Garuda kalah Erik Tohir dan pung...	25w	Tunggu timnas Garuda kalah Erik Tohir dan pung...	tunggu timnas garuda kalah erik tohir dan pung...	[tunggu, timnas, garuda, kalah, erik, tohir, d...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...
3	Kamal CeNat CenOt	Udah bagus2 sty melatih timnas.. Eh kok di pecat	25w	Udah bagus sty melatih timnas Eh kok di pecat	udah bagus sty melatih timnas eh kok di pecat	[udah, bagus, sty, melatih, timnas, eh, kok, d...	[sudah, bagus, sty, melatih, timnas, eh, kok, ...	[bagus, sty, melatih, timnas, pecat]	[bagus, sty, latih, timnas, pecat]	[bagus, sty, latih, timnas, pecat]
4	Dilar Bahtiar	Kecewa ama pssi blum juga timnas beres dah dignti	25w	Kecewa ama pssi blum juga timnas beres dah dignti	kecewa ama pssi blum juga timnas beres dah dignti	[kecewa, ama, pssi, blum, juga, timnas, beres,...	[kecewa, sama, pssi, belum, juga, timnas, bere...	[kecewa, pssi, belum, timnas, beres, dignti]	[kecewa, pssi, belum, timnas, beres, dignti]	[kecewa, pssi, belum, timnas, beres]

Fig. 10: Source code and results dictionary filter

3.4. Deleting empty rows

The process of deleting empty rows is useful for maintaining data quality and preventing errors during the analysis process in order to maintain data integrity and consistency and avoid biased data results. The source code for this process is as follows:

```
[13] # Menghapus baris di mana kolom 'filter_text' kosong atau tidak ada isi
     df_filtered = df[df['filter_text'].apply(lambda x: len(x) > 0 if isinstance(x, list) else False)].copy()

     print("Jumlah baris sebelum filtering:", len(df))
     print("Jumlah baris setelah filtering:", len(df_filtered))

     # Mengganti df dengan df_filtered untuk langkah selanjutnya
     df = df_filtered

     df.head()

Jumlah baris sebelum filtering: 2355
Jumlah baris setelah filtering: 2319
```

	username	comment	time	cleaning	case_folding	tokenizing	normalisasi	stopword_removal	stemming	filter_text
0	Galuh Panjalu	Minimal selsaikan dulu lah rencana sty di roun...	25w	Minimal selsaikan dulu lah rencana sty di roun...	minimal selsaikan dulu lah rencana sty di roun...	[minimal, selsaikan, dulu, lah, rencana, sty, ...	[minimal, selesaikan, dulu, lah, rencana, sty,...	[minimal, selesaikan, rencana, sty, rounde, pi...	[minimal, selesai, rencana, sty, rounde, pildu...	[minimal, selesai, rencana, sty, pildun, gagal...
1	Feri Peye Andrian	Yuk rame2 bikin demo online\nBikin hastag apat...	25w	Yuk rame bikin demo online Bikin hastag apatah...	yuk rame bikin demo online bikin hastag apatah...	[yuk, rame, bikin, demo, online, bikin, hastag...	[yuk, ramai, buat, demo, online, buat, hastag,...	[ramai, demo, online, hastag, pssi, apalahmasa...	[ramai, demo, online, hastag, pssi, apalahmasa...	[ramai, demo, pssi, benci, coach, sty, menang,...
2	Suro Tamping Gramang	Tunggu timnas Garuda kalah Erik Tohir dan pung...	25w	Tunggu timnas Garuda kalah Erik Tohir dan pung...	tunggu timnas garuda kalah erik tohir dan pung...	[tunggu, timnas, garuda, kalah, erik, tohir, d...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...	[tunggu, timnas, garuda, kalah, erick, thohir,...
3	Kamal CeNat CenOt	Udah bagus2 sty melatih timnas.. Eh kok di pecat	25w	Udah bagus sty melatih timnas Eh kok di pecat	udah bagus sty melatih timnas eh kok di pecat	[udah, bagus, sty, melatih, timnas, eh, kok, d...	[sudah, bagus, sty, melatih, timnas, eh, kok, ...	[bagus, sty, melatih, timnas, pecat]	[bagus, sty, latih, timnas, pecat]	[bagus, sty, latih, timnas, pecat]
4	Dilar Bahtiar	Kecewa ama pssi blum juga timnas beres dah dignti	25w	Kecewa ama pssi blum juga timnas beres dah dignti	kecewa ama pssi blum juga timnas beres dah dignti	[kecewa, ama, pssi, blum, juga, timnas, beres,...	[kecewa, sama, pssi, belum, juga, timnas, bere...	[kecewa, pssi, belum, timnas, beres, dignti]	[kecewa, pssi, belum, timnas, beres, dignti]	[kecewa, pssi, belum, timnas, beres]

Fig. 11: Source code and results of deleting empty rows

3.5. Labeling

The sentiment labeling process classifies each comment into positive, negative, or neutral categories according to its contextual meaning. This process aims to enable the model to recognize word patterns from each sentiment category based on the given labels. The required source code is:

```
# Memuat kamus leksikon positif dan negatif
lexicon_pos = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Skripsi aku/dokumen-dan-kamus/lexicon_positive.csv', header=None, names=['word', 'weight'])
positive_lexicon = dict(zip(lexicon_pos['word'], lexicon_pos['weight']))

lexicon_neg = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Skripsi aku/dokumen-dan-kamus/lexicon_negative.csv', header=None, names=['word', 'weight'])
negative_lexicon = dict(zip(lexicon_neg['word'], lexicon_neg['weight']))

# Fungsi untuk pelabelan sentimen
def sentiment_labeling(filter_text):
    score = 0
    if not isinstance(filter_text, list):
        return 'netral'

    for token in filter_text:
        if token in positive_lexicon:
            score += positive_lexicon.get(token, 0)
        elif token in negative_lexicon:
            score += negative_lexicon.get(token, 0)

    if score > 0:
        return 'positif'
    elif score < 0:
        return 'negatif'
    else:
        return 'netral'

# Menerapkan fungsi pelabelan pada kolom 'filter_text'
df['sentiment'] = df['filter_text'].apply(sentiment_labeling)

# --- Kode Tambahan untuk Menghitung Jumlah Sentimen ---
print("\nJumlah setiap kelas sentimen:")
print(df['sentiment'].value_counts())

# Menampilkan 5 baris pertama dari hasil akhir dengan format .head()
print("Hasil Pelabelan Sentimen:")
print(df[['comment', 'sentiment']].head())
```

Fig. 12: Source code of labeling

```
Jumlah setiap kelas sentimen:
sentiment
positif    1267
negatif     705
netral      345
Name: count, dtype: int64
Hasil Pelabelan Sentimen:
                                             comment sentiment
0  Minimal selsaikan dulu lah rencana sty di roun...   positif
1  Yuk rame2 bikin demo online\nBikin hastag apat...   positif
2  Tunggu timnas Garuda kalah Erik Tohir dan pung...   positif
3   Udah bagus2 sty melatih timnas.. Eh kok di pecat   positif
4  Kecewa ama pssi blum juga timnas beres dah dignti   negatif
```

Fig. 13: Results of labeling

From the labeling results above, we can see the total number of each sentiment class, where positive sentiments number 1,267, negative sentiments number 705, and neutral sentiments number 345. This comparison shows that the majority of public comments tend to be positive. Each line of comments in the data has been labeled with a sentiment class.

3.6. Split Data

The data split stage is performed to separate training and test data to measure the model's ability on new data. The required source code is:

```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['filter_text'], df['sentiment'], test_size=0.2, random_state=42)

print("Training set size:", len(X_train))
print("Testing set size:", len(X_test))
```

```
Training set size: 1853
Testing set size: 464
```

Fig. 14: Source code and results of split data

From the results above, we can see the distribution of training data and testing data. The amount of training data used is 1,853 comments, while the testing data consists of 464 comments. The purpose of this distribution is so that the K-Nearest Neighbors model can recognize word patterns optimally through training data and obtain accurate evaluations by utilizing testing data.

3.7. TF-IDF

The TF-IDF vectorization stage aims to convert textual data (strings) into numerical feature matrices. Each word is given a weight that reflects its significance, calculated from its frequency of occurrence in a document relative to its frequency in the entire data corpus. The required source code is:

```
# 1. Menggabungkan token hasil filter_text menjadi string
#    Data diambil dari kolom 'filter_text' pada DataFrame df
#    Menggunakan indeks X_train dan X_test agar sesuai dengan data hasil split
X_train_joined = df.loc[X_train.index, 'filter_text'].apply(
    lambda tokens: ' '.join(tokens) if isinstance(tokens, list) else '')
X_test_joined = df.loc[X_test.index, 'filter_text'].apply(
    lambda tokens: ' '.join(tokens) if isinstance(tokens, list) else '')

# 2. Inisialisasi TF-IDF Vectorizer
#    max_features=1000 digunakan untuk membatasi jumlah kata yang dipakai
#    agar proses komputasi lebih efisien dan mengurangi noise
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(max_features=3000)

# 3. Melatih TF-IDF pada data latih dan mentransformasi data latih & data uji
X_train_tfidf = vectorizer.fit_transform(X_train_joined)
X_test_tfidf = vectorizer.transform(X_test_joined)

# 4. Mengubah hasil matriks TF-IDF data latih menjadi DataFrame
#    Kolom DataFrame berisi kata-kata (fitur) yang digunakan
tfidf_df = pd.DataFrame(
    X_train_tfidf.toarray(),
    columns=vectorizer.get_feature_names_out())

# 5. Menampilkan 5 baris pertama matriks TF-IDF untuk data latih
print("\nMatriks TF-IDF untuk data latih (5 baris pertama):")
print(tfidf_df.head(10))

# 6. Menampilkan dimensi matriks TF-IDF
#    Bentuk: (jumlah dokumen, jumlah fitur)
print("\nDimensi matriks TF-IDF data latih:", X_train_tfidf.shape)
print("Dimensi matriks TF-IDF data uji:", X_test_tfidf.shape)
```

Fig. 15: Source code of TF-IDF

```
Matriks TF-IDF untuk data latih (5 baris pertama):
   abal    abang  abroad  academy  achieve  aco  acuh  adaptasi  adik  adil  \
0   0.0  0.82391     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
1   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
2   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
3   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
4   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
5   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
6   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
7   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
8   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0
9   0.0  0.00000     0.0      0.0      0.0  0.0   0.0       0.0   0.0   0.0

   ...  welcome  why  wilayah  wing  wni  world  wujud  young  zaman  zona
0  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
1  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
2  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
3  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
4  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
5  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
6  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
7  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
8  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0
9  ...      0.0  0.0      0.0   0.0  0.0    0.0    0.0    0.0    0.0   0.0

[10 rows x 1941 columns]

Dimensi matriks TF-IDF data latih: (1853, 1941)
Dimensi matriks TF-IDF data uji: (464, 1941)
```

Fig. 16: Result of TF-IDF

From the TF-IDF results above, the training data matrix, each column represents a unique word (term) in the training data, while each row describes one comment. The number in each cell indicates the TF-IDF weight of a particular word in that comment. For example, the word “abang” has a weight of 0.82391, which indicates that this word plays a significant role in distinguishing this comment from others. This matrix is used as input by the K-Nearest Neighbors algorithm to learn the word patterns found in the training data.

3.8. Train the K-NN Model

Next is the process of training the K-Nearest Neighbors model, where the model will learn from the patterns in the training data. After that, the trained model will be evaluated using the test data. The code for performing these two processes is as follows:

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Inisialisasi model K-NN
knn_model = KNeighborsClassifier(n_neighbors=80)

# Melatih model K-NN menggunakan data latih TF-IDF
knn_model.fit(X_train_tfidf, y_train)

# Melakukan prediksi pada data uji TF-IDF
y_pred_knn = knn_model.predict(X_test_tfidf)

# Evaluasi model
accuracy_knn = accuracy_score(y_test, y_pred_knn)
classification_report_knn = classification_report(y_test, y_pred_knn)
confusion_matrix_knn = confusion_matrix(y_test, y_pred_knn)

print("Akurasi K-NN:", accuracy_knn)
print("\nClassification Report K-NN:\n", classification_report_knn)

# Menyimpan confusion matrix untuk visualisasi
cm_knn = confusion_matrix_knn
```

Fig. 17: K-NN Model source code and its accuracy

The training process using the K-NN model was carried out by setting k = 80, then evaluation was performed by measuring performance using evaluation matrices such as accuracy, precision, recall, and f1-score.

3.9. Accuracy Results

After training with the K-Nearest Neighbors model, proceed to the accuracy measurement stage using an evaluation matrix that will display the accuracy, precision, recall, and f1-score results with the following source code:

```
Akurasi K-NN: 0.7737068965517241

Classification Report K-NN:
              precision    recall  f1-score   support

     negatif       0.72      0.81      0.76       148
      netral       0.60      0.06      0.11        50
     positif       0.81      0.89      0.85       266

    accuracy                           0.77       464
   macro avg       0.71      0.59      0.57       464
weighted avg       0.76      0.77      0.74       464
```

Fig. 18: K-NN model results and accuracy

The results of the model evaluation above show an accuracy value of 0.77 or 77%, which means that the model is able to make correct predictions on most of the testing data. The positive sentiment category showed the best performance with a precision value of 0.81, recall of 0.89, and f1-score of 0.85, reflecting a high ability to recognize positive comments. Similarly, the model's performance on the negative class obtained a precision value of 0.72, recall of 0.81, and f1-score of 0.76. Meanwhile, comments with neutral sentiment tended to be identified with low performance, with precision values of 0.6, recall of 0.06, and f1-score of 0.11. These findings indicate that the model tends to be more accurate in classifying positive and negative sentiments, but is very weak in the neutral class.

3.10. Visualization results

Visualization results aim to present sentiment analysis results in the form of images and graphs so that they are easier to understand and analyze. Visualization serves as a means of translating complex data into simpler and more communicative visual representations.

3.10.1. Pie Chart

To show the distribution of the overall sentiment (positive, negative, neutral) in a dataset with a visual display in the form of a pie chart. This can be seen with the following source code:

```
[35] import matplotlib.pyplot as plt

     # Hitung jumlah setiap kelas sentimen
     sentiment_counts = df['sentiment'].value_counts()

     # Buat pie chart
     plt.figure(figsize=(5, 5))
     plt.pie(sentiment_counts.values, labels=sentiment_counts.index, autopct='%1.1f%%', startangle=140, colors=['#66b3da','#ff9999','#99ff99'])
     plt.title('Distribusi Label Sentiment')
     plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
     plt.show()
```

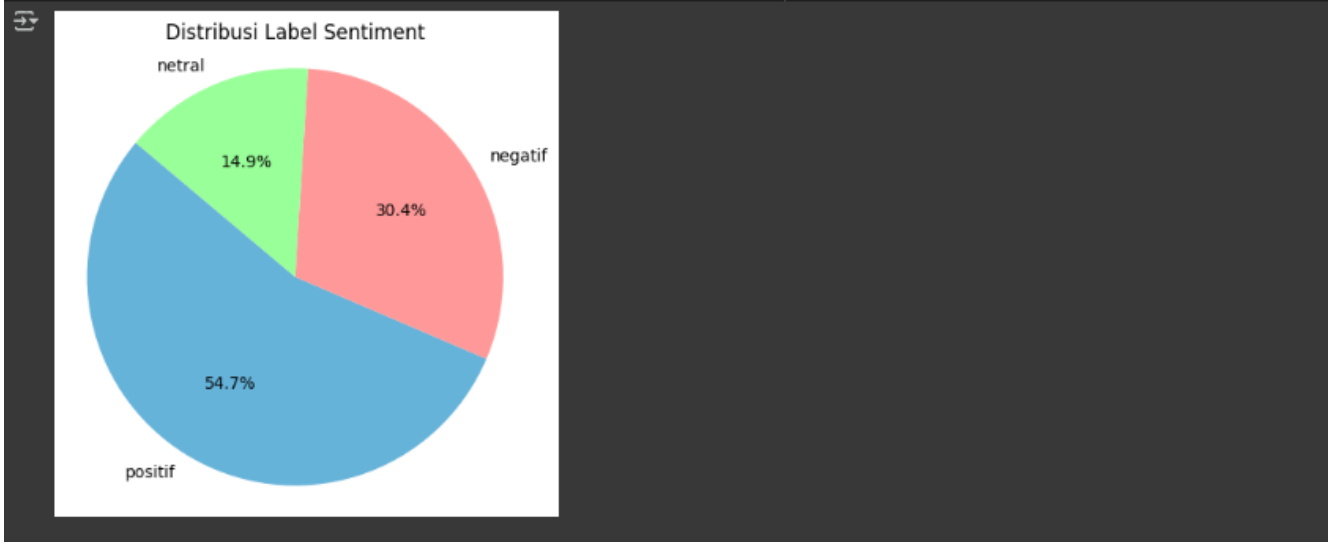

Fig. 19: Source code and results of Pie Chart

The pie chart above shows the distribution of sentiment labels in the dataset that has undergone a series of data cleaning, numerical representation, data split, and labeling processes. The pie chart shows that positive data dominates with 54.7% of the total data. Next, negative sentiment ranks second with a percentage of 30.4%, while neutral sentiment is the least with a percentage of only 14.9%.

3.10.2. Bar Chart

To compare the number or proportion of sentiments in various categories with a visual display in the form of a rectangular bar chart. This can be seen with the following source code:

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Hitung jumlah setiap kelas sentimen dan ubah ke persentase
sentiment_counts = df['sentiment'].value_counts()
sentiment_percentages = sentiment_counts / len(df) * 100

# Urutkan berdasarkan index (label sentimen) agar urutan bar konsisten
sentiment_counts = sentiment_counts.sort_index()
sentiment_percentages = sentiment_percentages.sort_index()

# Buat bar chart dengan warna kustom
plt.figure(figsize=(8, 6))
# Define custom colors based on sentiment labels (sorted by index)
colors = ['red', 'gray', 'green'] # Assuming the order is negatif, netral, positif based on sorted index
ax = sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values, palette=colors)
plt.title('Distribusi Label Sentiment')
plt.xlabel('Sentimen')
plt.ylabel('Jumlah')

# Tambahkan persentase dan jumlah aktual di atas bar
for i, p in enumerate(ax.patches):
    percentage = f'{sentiment_percentages.iloc[i]:.1f}%'
    count = sentiment_counts.iloc[i]
    annotation_text = f'{count} ({percentage})'
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    ax.annotate(annotation_text, (x, y), ha='center', va='bottom')


plt.show()
```

Fig. 20: Source code and results of Bar Chart

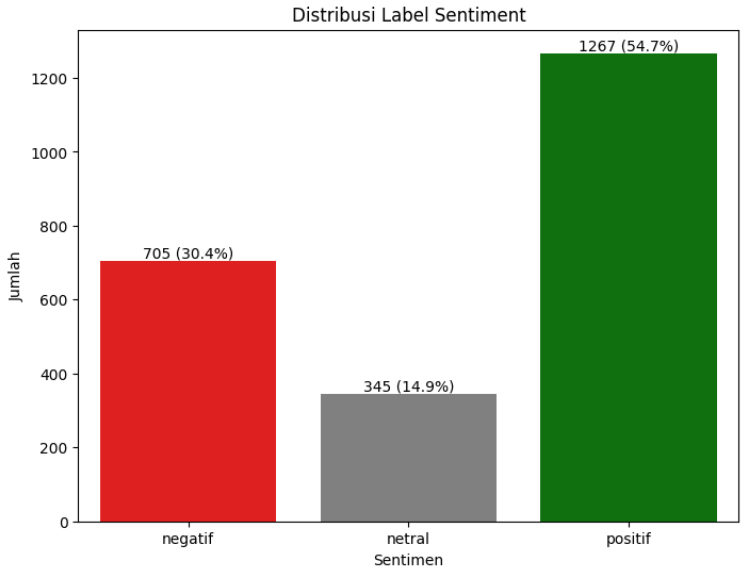

Fig. 21: Source code and results of Bar Chart

The bar chart above shows the distribution of data based on sentiment labels in the form of a bar chart. It can be concluded that the positive class has the largest amount of data, namely more than 1,267 data points. Next, the negative class is in second place with 705 data points, while the neutral class has the least amount of data with around 345 data points.

3.10.3. World Cloud

To display the dominant words in each category, which are visual representations of a collection of words in a text or dataset. This can be seen in the following source code:

Fig. 22: Source code and results of World Cloud

The visualization shows that words such as “tidak”, “pssi”, “sty”, “latih”, “timnas”, and “main” appear in the largest font size, meaning that these words dominate public conversations or comments.

3.10.4. Confusion matrix

Confusion matrix visualization aims to display the results of classification model evaluation in the form of tables or graphs that show the number of correct and incorrect predictions for each class. This can be seen in the following source code:

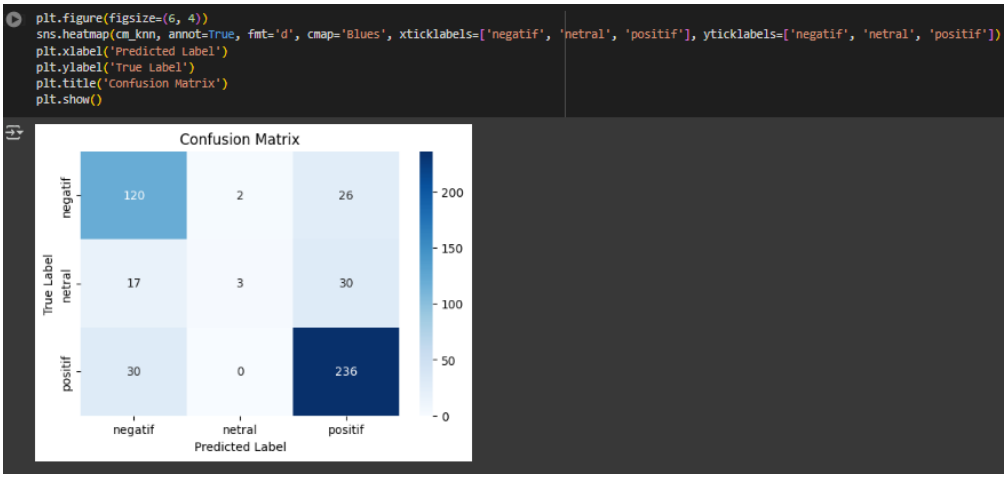

Fig. 23: Source code and results of confusion matrix

Based on the test results using the K-Nearest Neighbors algorithm, a confusion matrix with k = 80 was obtained in the image above. This confusion matrix provides an overview of the number of correct and incorrect predictions in each sentiment class, namely negative, neutral,

and positive. In the neutral class, only 3 data points were correctly predicted, while 17 neutral data points were incorrectly predicted as negative and 30 neutral data points were incorrectly predicted as positive. This shows that the model has difficulty recognizing neutral sentiment. In the positive class, 236 data points were correctly predicted, while 30 positive data points were classified as negative.

4. Conclusions and Suggestions

4.1. Conclusions

Based on the results of sentiment analysis of public comments on Facebook regarding the news of Shin Tae-Yong's dismissal, it can be concluded that the classification model built using the K-Nearest Neighbors (K-NN) method with k = 80 was able to classify public sentiment with an overall accuracy rate of 77%. The model's performance shows good results for positive and negative sentiment classes, but is very weak in identifying neutral sentiment classes. This is evident from the recall value of the neutral class, which is only 0.02, where most of the neutral data is misclassified as positive or negative. The distribution of sentiment in the analyzed dataset shows that public opinion is dominated by positive sentiment at 54.7% (1267 comments). Negative sentiment ranks second with a percentage of 30.4% (705 comments), and neutral sentiment is the least with a percentage of 14.9% (345 comments). Although the dismissal of an accomplished coach could be assumed to trigger negative responses, the findings of this study show that public perception on Facebook is mostly positive.

4.2. Suggestions

Based on the research described above, the author provides several suggestions and recommendations that may be considered for future research, namely, it is recommended to perform a more sophisticated data preprocessing model with a model base that may be more reliable for handling ambiguous or neutral text data, given the very low performance in the neutral class. Expanding the scope of data knowledge by collecting comments from other social media platforms such as X (Twitter) and YouTube to obtain a more comprehensive and diverse picture of public perception.

References

[1] A. Setiadarma, "Kaitan Public Opinion dan Public Relation," *Ilmu Komunikasi*, vol. XXVI, no. 3, pp. 216–219, 2021.

[2] N. M. Sania, N. Baitillah, M. H. Indriani, F. Fernanda, and T. Aditya, "Survei Kepuasan Opini Publik terhadap Kebijakan Naturalisasi Pemain PSSI : dalam Upaya Meningkatkan Prestasi Timnas Indonesia," no. 2, pp. 1–19, 2025.

[3] Idah Nufajriya Awwalin, Latif Syaipudin, and Ahmad Luthfi, "Analisis Respon Publik Melalui Sosial Media Facebook terhadap Wacana Kenaikan Pajak Pasca Pilpres 2024 pada Media Pemberitaan," *AKSAYA: Jurnal Rumpun Akuntansi Publik*, vol. 1, no. 1, pp. 01–08, 2025.

[4] D. Jacarria Pangestu and A. Kodar, "Implementasi Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Terhadap Pelayanan Perusahaan Otobus Menggunakan Data Facebook (Studi Kasus: Grup Facebook Murni Jaya Lovers)," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 7, no. 3, pp. 156–160, 2022.

[5] A. (Asrumi) Asrumi, D. (Didik) Suharijadi, A. D. (Agustina) Setiari, and D. P. (Diah) Wulanda, *Analisis Sentimen dan Penggalian Opini*, no. 1. Purbalingga: CV. Eureka Media Aksara, 2023. [Online]. Available: https://repository.penerbiteureka.com/publications/567221/

[6] R. N. Handayani, "Optimasi Algoritma Support Vector Machine untuk Analisis Sentimen pada Ulasan Produk Tokopedia Menggunakan PSO," *Media Informatika*, vol. 20, no. 2, pp. 97–108, 2021, doi: 10.37595/mediainfo.v20i2.59.

[7] Y. Putra Dinata, M. Fikry, F. Yanto, and E. Pandu Cynthia, "Analisis Sentimen Terhadap Sebuah Figur Publik di Twitter Menggunakan Metode K-Nearest Neighbor," *Media Online)*, vol. 4, no. 6, pp. 2822–2829, 2024, doi: 10.30865/klik.v4i6.1904.

[8] F. Rizqi Irawan, "Analisis Sentimen Terhadap Pengguna Gojek Menggunakan Metode K-Nearset Neighbors," *JIKO (Jurnal Informatika dan Komputer)*, vol. 5, no. 1, pp. 62–68, 2022, doi: 10.33387/jiko.v5i1.4267.

[9] M. Furqan, S. Sriani, and S. M. Sari, "Analisis Sentimen Menggunakan K-Nearest Neighbor Terhadap New Normal Masa Covid-19 Di Indonesia," *Techno.Com*, vol. 21, no. 1, pp. 51–60, 2022, doi: 10.33633/tc.v21i1.5446.

[10] I. H. Kusuma and N. Cahyono, "Analisis Sentimen Masyarakat Terhadap Penggunaan E-Commerce Menggunakan Algoritma K-Nearest Neighbor," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 3, pp. 302–307, 2023, doi: 10.30591/jpit.v8i3.5734.

[11] A. Rianti, N. W. A. Majid, and A. Fauzi, "CRISP-DM: Metodologi Proyek Data Science," *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, pp. 107–114, 2023, [Online]. Available: https://ojs.udb.ac.id/index.php/Senatib/article/view/3015

[12] Y. A. Singgalen, "Penerapan Metode CRISP-DM dalam Klasifikasi Data Ulasan Pengunjung Destinasi Danau Toba Menggunakan Algoritma Naïve Bayes Classifier (NBC) dan Decision Tree (DT)," *Jurnal Media Informatika Budidarma*, vol. 7, no. 3, p. 1551, 2023, doi: 10.30865/mib.v7i3.6461.

[13] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Comput Sci*, vol. 181, no. 2019, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.

[14] M. N. Muttaqin and I. Kharisudin, "Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor," *UNNES Journal of Mathematics*, vol. 10, no. 2, pp. 22–27, 2021, [Online]. Available: http://journal.unnes.ac.id/sju/index.php/ujm

[15] I. P. A. E. D. Udayana, I. G. I. Sudipa, and R. Risaldi, "Sentimen Analisis Inisiatif Vaksin Nasional Menggunakan Naïve Bayes dan Laplacian Smoothing Pada Komentar Video Youtube," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 5, no. 2, pp. 116–126, 2022, doi: 10.31598/jurnalresistor.v5i2.1108.

[16] R. C. Rivaldi, T. D. Wismarini, J. T. Lomba, and J. Semarang, "Analisis Sentimen Pada Ulasan Produk Dengan Metode Natural Language Processing (NLP) (Studi Kasus Zalika Store 88 Shopee)," *Jurnal Ilmiah Elektronika Dan Komputer*, vol. 17, no. 1, pp. 120–128, 2024.

[17] M. Afdal and L. R. Elita, "Penerapan Text Mining Pada Aplikasi Tokopedia Menggunakan Algoritma K-Nearest Neighbor," *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, vol. 8, no. 1, p. 78, 2022, doi: 10.24014/rmsi.v8i1.16595.

[18] D. Chrisinta and J. E. Simarmata, "Eksplorasi Teknik Web Scraping pada Data Mining: Pendekatan Pencarian Data Berbasis Python," *Faktor Exacta*, vol. 17, no. 1, pp. 58–68, 2024, doi: 10.30998/faktorexacta.v17i1.22393.

[19] I. Naing, S. T. Aung, K. H. Wai, and N. Funabiki, "A Reference Paper Collection System Using Web Scraping," *Electronics (Switzerland)*, vol. 13, no. 14, 2024, doi: 10.3390/electronics13142700.

[20] N. K. Kahlon and W. Singh, "Comparative Analysis of Web Scraping Tools for Low-Resource Language Text," *International Journal of Engineering Trends and Technology*, vol. 72, no. 1, pp. 284–299, 2024, doi: 10.14445/22315381/IJETT-V72I1P128.

[21] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *Jurnal Media Informatika Budidarma*, vol. 5, no. 2, p. 406, 2021, doi: 10.30865/mib.v5i2.2835.

[22] D. Wardhani, R. Astuti, and D. D. Saputra, "Optimasi Feature Selection Text Mining: Stemming dan Stopword," *INNOVATIVE: Journal Of Social Science Research*, vol. 4, no. 1, pp. 7537–7548, 2024.

[23] N. Silalahi and Guidio Leonarde Ginting, "Rekomendasi Berita Berkaitan dengan Menerapkan Algoritma Text Mining dan TF-IDF," *Bulletin of Computer Science Research*, vol. 3, no. 4, pp. 276–282, 2023, doi: 10.47065/bulletincsr.v3i4.266.

[24] I. S. Wibowo, A. Witanti, and I. Susilawati, "Keyword Extraction Judul Berita Online Di Indonesia Menggunakan Metode TF-IDF," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 11, no. 1, pp. 99–111, 2024, [Online]. Available: http://jurnal.mdp.ac.id

[25] Tegar Arifin Prasetyo, *Buku Ajar: Data Science Dan Machine Learning Project Mindmap Untuk Kalangan Mahasiswa, Peneliti, Dan Pebisnis*. Deepublish, 2023.

[26] Heliyanti Susana, "Penerapan Model Klasifikasi Metode Naive Bayes Terhadap Penggunaan Akses Internet," *Jurnal Riset Sistem Informasi dan Teknologi Informasi (JURSISTEKNI)*, vol. 4, no. 1, pp. 1–8, 2022, doi: 10.52005/jursistekni.v4i1.96.

[27] T. Harlina and E. Handayani, "Klasifikasi Motif Batik Banyuwangi Menggunakan Metode K-Nearest Neighbor (K-NN) Berbasis Android," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 1, pp. 82–96, 2022, doi: 10.29100/jipi.v7i1.2411.

[28] N. R. P. Wardoyo, J. Santoso, and E. I. Setiawan, "Klasifikasi Micro-Expression Menggunakan K-Nearest Neighbors Menggunakan Fitur CAS dan HOG," *Journal of Intelligent System and Computation*, vol. 5, no. 2, pp. 96–103, 2023, doi: 10.52985/insyst.v5i2.346.

[29] T. N. Halim, R. Martin, R. Kesia, B. Hutasoit, and S. Aisyah, "Kepuasan Pelanggan Terhadap Platform E-Commerce Metode K-NN(Tio Novian Halim) |512 Klasifikasi Kepuasan Pelanggan Terhadap Platform E-Commerce dengan Metode K-Nearest Neighbor (K-NN)," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. 8, no. 2, pp. 512–523, 2023, [Online]. Available: https://tunasbangsa.ac.id/ejurnal/index.php/jurasik