

**ANALISIS SENTIMEN PADA ULASAN APLIKASI TIKTOK TRAGEDI STADION KANJURUHAN
MENGUNAKAN METODE LEXICON**

***SENTIMENT ANALYSIS ON TIKTOK APPLICATION REVIEWS KANJURUHAN STADIUM
TRAGEDY USING THE LEXICON METHOD***

Faruq Amrulloh¹ Siti Madinah Ladjamuddin²

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Informasi
Institut Sains dan Teknologi Nasional
Jl. Moh Kahfi II, Bhumi Srengseng Indah, Jagakarsa, Jakarta Selatan 12640
Telp. (021) 7874647, Fax. (021) 7866955
faruq.amrulloh8586@gmail.com¹, citymadinah07@istn.ac.id²

ABSTRAKSI

Tiktok merupakan sebuah jaringan sosial media dan platform video musik Tiongkok yang diluncurkan sejak September 2016. Salah satu contoh video yang beredar luas dan sempat menghebohkan dunia adalah video mengenai kerusuhan yang terjadi di stadion Kanjuruhan Malang. Dalam penelitian ini bertujuan untuk mengetahui sentimen pada kasus video tragedi stadion kanjuruhan Malang menggunakan metode lexicon. Metode lexicon merupakan metode pengklasifikasi sentimen kedalam tiga kelas, yaitu positif, negatif dan netral. Penelitian ini menggunakan data dari kolom komentar video tiktok yang diunggah pada tanggal 11 Oktober 2022 tentang tragedi stadion kanjuruhan Malang, dari 524 baris data hanya terpilih 519 baris data setelah melakukan tahapan text preprocessing. Penelitian ini menggunakan software Visual Studio Code dengan beberapa library tambahan. Hasil dari penelitian ini bersifat sentimen negatif dengan jumlah data sebanyak 300 baris dengan presentase 57,80%, sedangkan sentimen positif hanya 96 baris dengan presentase 18,49% dan sentimen netral mendapatkan data sebanyak 123 baris dengan presentase 23,69%.

Kata Kunci : Tiktok, Kanjuruhan, Analisis Sentimen, Lexicon .

ABSTRACT

Tiktok is a Chinese social media network and music video platform launched in September 2016. One example of a video that has become widely circulated and has shocked the world is a video about the riot that occurred at the Kanjuruhan stadium in Malang. This study aims to determine sentiment in the video case of the Malang Kanjuruhan Stadium tragedy using the lexicon method. The lexicon method is a method of classifying sentiment into three classes, namely positive, negative and neutral. This research uses data from the comments column of the tiktok video which was uploaded on October 11 2022 about the Malang kanjuruhan stadium tragedy, out of 524 data lines only 519 data lines were selected after carrying out the text preprocessing stage. This research uses Visual Studio Code software with several additional libraries. The results of this study are negative sentiment with a total of 300 lines of data with a percentage of 57.80%, while positive sentiment is only 96 lines with a percentage of 18.49% and neutral sentiment gets 123 lines of data with a percentage of 23.69%.

Keywords : Tiktok, Kanjuruhan, Analysis Sentimen, Lexicon

1. PENDAHULUAN

Media sosial saat ini sedang mengalami perkembangan yang sangat pesat dikalangan masyarakat Indonesia, salah satu media sosial yang meningkat dan berasal dari China yaitu Tiktok. Aplikasi Tiktok adalah sebuah media sosial untuk membuat dan menyebarkan berbagai macam video pendek. Tiktok tidak hanya untuk membuat dan menyebarkan video pendek, Tiktok juga bisa mengirim kan pesan kepada seseorang yang sudah saling mengikuti di aplikasi tersebut. Tiktok juga menjadi salah satu sarana pemberitaan dan berbagi informasi yang sedang hangat di kalangan masyarakat

seperti isu-isu politik, sosial, budaya, film hingga pemberitaan internasional. Salah satu kasus yang sedang hangat di bahas masyarakat Indonesia adalah kerusuhan yang terjadi di stadion Kanjuruhan Malang pada tanggal 1 Oktober 2022. Banyak sekali pendapat masyarakat mengenai kejadian yang sedang terjadi, kejadian ini membuat banyak pro dan kontra di kalangan masyarakat Indonesia, dan masyarakat banyak menyuarakan pendapatnya di kolom komentar salah satu video yang di unggah ke sosial media tiktok. Ada banyak komentar di video tersebut, kita bisa melihat presentase komentar positif, negatif dan netral dengan menggunakan analisis sentimen untuk

mengetahui sentimen pengguna tiktok dengan menggunakan metode lexicon.

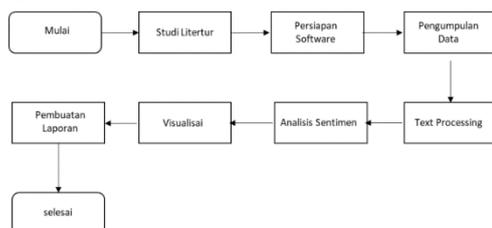
Dengan banyaknya komentar pengguna tiktok yang berpendapat mengenai kasus tersebut, bisa dilihat presentase komentar positif, negatif dan netral dengan analisis sentimen yang akan membagi kelas sentimen menjadi positif, negatif dan netral menggunakan metode lexicon terhadap komentar tiktok. Dengan adanya penelitian ini diharapkan pemerintah bisa mengambil langkah tegas akan kejadian yang terjadi di stadion kanjuruhan malang.

Pada penelitian yang dilakukan oleh (Rifiana Arief & Karel Imanuel 2019) yang berjudul Analisis Sentimen Topik Viral Desa Penari Pada Media Sosial Twitter Dengan Metode Lexicon Based. Hasil yang di dapatkan dari 1000 data tweet ini dengan menggunakan metode lexicon adalah 33 tweet positif (3,3%), 767 tweet netral (76,7%) dan 200 tweet negatif (20%). Ada juga penelitian yang dilakukan oleh (Adi Pandu Rahmat Nababan, Arie S. M. Lumenta, Yaulie D. Y. Rindengan, Fransisca J. Pontoh, Yuri V. Akay 2020) yang berjudul Analisis Sentimen Twitter Pasca Pengumuman Hasil Pilpres 2019 Menggunakan Metode Lexicon Analysis. Hasil yang diperoleh sentimen positif 54%, sentimen negatif 51%, sentimen netral 21%. Dan ada lagi penelitian yang dilakukan oleh (Syifa Khairunnisa, Adiwijaya, Said Al Faraby 2021) dengan judul Pengaruh Text Preprocessing Terhadap Analisis Sentimen Komentar Masyarakat Pada Media Sosial Twitter (Studi Kasus Pandemi Covid-19). Dan hasilnya adalah Ketika menggunakan kombinasi preprocessing cleaning, stemming dan kombinasi normalisasi kata cleaning dan stemming yang menghasilkan akurasi yang sama yaitu 77,77%.

2. METODOLOGI PENELITIAN

Tahapan Penelitian

Pada sub bab tahapan penelitian ini penulis akan menjelaskan alur metode penelitian yang digunakan.



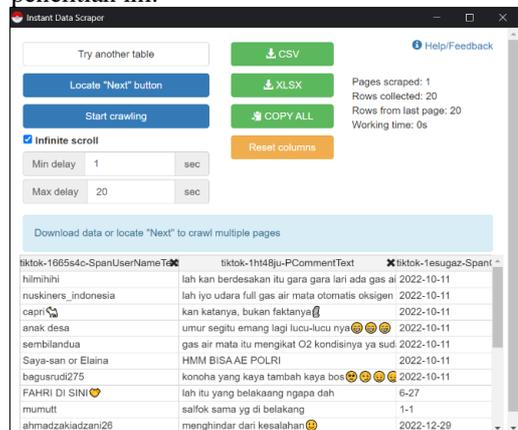
Gambar Tahapan Penelitian
(sumber : rancangan 2023)

Studi Literatur

Pada tahap ini akan dilakukan pengumpulan informasi dan referensi yang berhubungan dengan penelitian ini melalui jurnal ilmiah, e-book dan literature lainnya mengenai sentiment analisis menggunakan metode lexicon. Tahapan ini bertujuan untuk memahami teori yang terkait. Hasil dari tahap ini dapat dilihat pada bagian tinjauan pustaka.

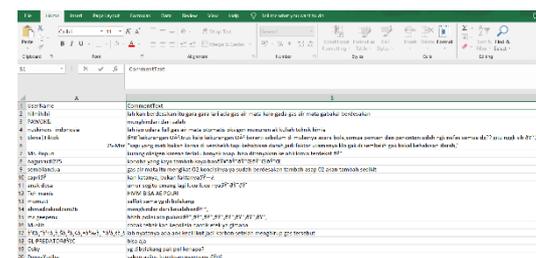
Pengumpulan Data

Pada penelitian ini proses pengumpulan data menggunakan metode web scrapping data otomatis dengan menggunakan extension chrome pada komentar postingan salah satu video akun tiktok yang bernama Bisnis.com. Tujuan dari pengumpulan data ini agar peneliti dapat menghasilkan sebuah informasi dan mengolah informasi tersebut lebih lanjut agar dapat menjawab rumusan masalah pada penelitian ini.



Gambar instant data scraper
(sumber: output program 2023)

Hasil dari scrape data menghasilkan beberapa kolom, yaitu kolom 1665s4c yang merupakan kolom nama pengguna tiktok, kolom 1ht48ju merupakan kolom yang menunjukkan komentar para pengguna tiktok dan kolom 1esugaz yang menunjukkan tanggal berapa komentar itu di posting. Namun penelitian ini hanya menggunakan kolom 1665s4c dan 1ht48ju. Setelah itu download hasil data tersebut dengan jenis file CSV (Comma Sparated Value) kemudian buka file tiktok.csv seperti yang ditunjukkan pada gambar di bawah ini



Gambar Hasil Data Scrape

(sumber : output program 2023)

Setelah data di download, selanjutnya upload data ke dalam Jupyter Notebook. Lalu import data ke dalam program seperti pada Pseudo Code di bawah ini

```
# load data

import pandas as pd

df = pd.read_csv('tiktokk.csv')
df.head()
```

Pseudo Code Sintaks Read Data
(sumber : code program 2023)

Pada Pseudo Code diatas dilakukan import library pandas untuk menganalisis data. Setelah itu data dibaca menggunakan fungsi read_csv. Dan untuk menampilkan data tersebut menggunakan fungsi head() maka akan menampilkan lima data teratas.

Text Preprocessing

Data yang sudah di dapatkan dari hasil scraping, selanjutnya akan dilakukan proses text preprocessing agar menjadi data yang dapat di proses dan diolah dalam bahasa pemrograman seperti yang ditampilkan pada gambar Pseudo Code sintaks Text Preprocessing yang merupakan sintaks untuk text preprocessing, dan library yang dibutuhkan untuk di import ke program adalah Pandas, Numpy, String, Regex, Matplotlib, Sastrawi, dan Natural Language Toolkit.

```
# preprocessing

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
import re
import string
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

import warnings
warnings.filterwarnings('ignore')
```

Pseude Code sintaks Text Preprocessing
(sumber : code program 2023)

Pada penelitian ini data yang diambil hanya data komentar atau opini masyarakat saja, dan untuk mengambil data komentar tersebut menggunakan sintaks pada Pseudo Code di bawah ini

```
# mengambil data comment

df_data = df[['tiktok-q9aj5z-PCommentText']]
df_data
```

Pseude Code sintaks Data Komentar
(sumber : code program 2023)

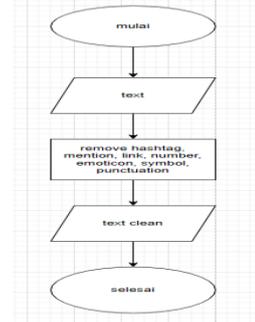
Sintaks df[['tiktok-q9aj5z']] merupakan sintaks untuk mengambil data komentar dari

tiktok-q9aj5z. Dan untuk menampilkan hasilnya menggunakan sintaks df_data.

Selanjutnya melakukan proses text preprocessing yang memiliki lima tahapan yaitu Cleaning text, Case folding, Tokenizing, Filtering, Stemming.

Cleaning text

Cleaning text merupakan text penghapusan yang berpotensi untuk mengurangi akurasi dari proses analisis. Alur kerja pada proses cleaning text ditunjukkan pada gambar



Gambar Flowchart Cleaning text
(sumber : rancangan 2023)

Pada gambar Flowchart Cleaning text dijelaskan bagaimana alur kerja dari cleaning text, diawali dengan memasukkan text yang akan dilakukan cleaning, untuk hal ini yaitu dengan memasukkan komentar dari tiktok, kemudian proses cleaning dilakukan dengan menghapus URL, mention, hashtag, number, emoticon, symbol, punctuation, dan akan menghasilkan teks yang bersih. Untuk code cleaning text ditunjukkan pada Pseudo Code sintaks Text Preprocessing

```
# tahap cleaning text

def cleaningText(text):
    text = re.sub(r'#[a-z0-9]+', '', text) # remove mentions
    text = re.sub(r'http[s]?://', '', text) # remove hashtag
    text = re.sub(r'https?://', '', text) # remove link
    text = re.sub(r'\d+', '', text) # remove numbers
    emoji_patterns = re.compile("[
        \U0001F600-\U0001F64F # emoticons
        \U0001F680-\U0001F6FF # symbols & pictographs
        \U0001F700-\U0001F7FF # transport & map symbols
        \U0001F1E0-\U0001F1FF # flags (iOS)
    ]+", flags=re.UNICODE)
    text = emoji_patterns.sub('', text)
    text = text.replace('\n', ' ') # replace new line into space
    text = text.translate(str.maketrans('', '', string.punctuation)) # remove all punctuations
    text = text.strip(' ') # remove characters space from both left and right text

    return text

# proses penerapannya fungsi cleaning text
df_data['tiktok-q9aj5z-PCommentText'] = df_data['tiktok-q9aj5z-PCommentText'].astype(str)
df_data['comment_clean'] = df_data['tiktok-q9aj5z-PCommentText'].apply(cleaningText)
df_data.head()
```

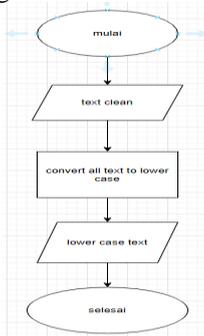
Pseudo Code Tahap Cleaning Text
(sumber : code program 2023)

Dalam tahapan cleaning text terdapat def cleaningtext() yang mendefinisikan fungsi cleaning text. Selanjutnya menggunakan fungsi re.sub untuk me-replace karakter-karakter yang dipilih dengan white space, karakter-karakter yang dipilih adalah mention, hashtag, link dan number. Kemudian ada fungsi re.compile() untuk mengembalikan objek regex, yang kemudian objek tersebut disimpan dalam variabel untuk penggunaan

selanjutnya. Dan selanjutnya ada fungsi `text.replace()` yang di gunakan untuk mengganti atau me-replace garis baru menjadi spasi. Selanjutnya fungsi `text.translate()` untuk menghapus semua tanda baca. Fungsi `text.strip()` untuk menghapus karakter spasi dari kanan atau kiri teks. Kemudian fungsi `astype(str)` untuk mengubah data menjadi string. Dan yang terakhir untuk pengaplikasian data menggunakan `apply()` dan untuk menampilkan data menggunakan `head()`.

Case Folding

Case Folding merupakan proses untuk mengubah seluruh bentuk huruf kapital menjadi huruf kecil atau lower case. Alur kerja pada proses case folding ditunjukkan pada gambar di bawah ini



Gambar Flowchart Case Folding
(sumber : rancangan 2023)

Pada gambar Flowchart Case Folding dijelaskan bahwa alur kerja untuk proses case folding, dimulai dengan memasukkan teks hasil dari cleaning text. Selanjutnya dilakukan proses pengubahan huruf kapital menjadi huruf kecil, dan menghasilkan teks yang berisi huruf kecil semua. Untuk code case folding berada pada Pseudo Code Tahap Case Folding

```

# case folding
def casefoldingText(text): # converting all the characters in a text into lower case
    text = text.lower()
    return text

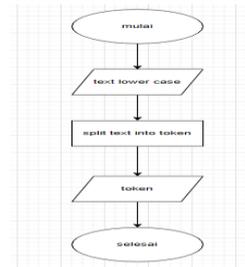
# proses pengaplikasian text folding
df_data['comment_clean'] = df_data['comment_clean'].apply(casefoldingText)
df_data.head(10)
  
```

Pseudo Code Tahap Case Folding
(sumber : program 2023)

Pada Pseudo Code Tahap Case Folding menjelaskan tentang tahap case folding. Terdapat `def casefoldingText()` untuk mendefinisikan case folding. Dan fungsi untuk mengubah huruf kapital menjadi huruf kecil dengan menggunakan `lower()`. Dan untuk pengaplikasian menggunakan `apply()`. Selanjutnya untuk menampilkan data menggunakan `head()`

Tokenizing

Tokenizing merupakan proses pemisahan teks dari bentuk kalimat menjadi bentuk perkata yang disebut token. Alur kerja pada proses tokenizing berada pada gambar Flowchart Tokenizing



Gambar Flowchart Tokenizing
(sumber : rancangan 2023)

Pada gambar Flowchart Tokenizing dijelaskan alur kerja tokenizing, dimulai dengan memasukkan hasil teks case folding lalu dilakukan pemisahan dari bentuk kalimat ke bentuk token. Untuk code tokenizing bisa dilihat pada Pseudo Code Tokenizing

```

# tokenizing
from nltk.tokenize import word_tokenize

def tokenizingText(text): # tokenizing or splitting a string, text into a list tokens
    text = word_tokenize(text)
    return text

# proses pengaplikasian fungsi tokenize
df_data['comment_preprocessing'] = df_data['comment_clean'].apply(tokenizingText)
df_data.head()
  
```

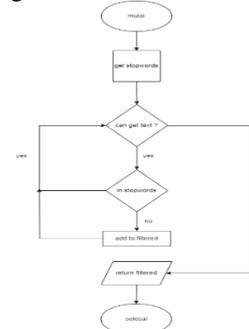
Pseudo Code Tokenizing

(sumber : code program 2023)

Untuk melakukan tokenizing terdapat `def tokenizingText()` untuk mendefinisikan fungsi tokenizing. Selanjutnya menggunakan `word_tokenize()` dari library Natural Language Toolkit (NLTK) untuk memisahkan kalimat menjadi kata perkata. Kemudian untuk pengaplikasiannya menggunakan `apply()` dan untuk menampilkan hasil dari data tersebut menggunakan `head()`.

Filtering

Filtering merupakan proses menghapus atau menyaring kata-kata yang tidak memiliki nilai. Alur kerja pada proses filtering terdapat pada gambar Flowchart filtering



Gambar Flowchart filtering
(sumber : rancangan 2023)

Pada gambar Flowchart filtering dijelaskan alur kerja filtering, dari teks hasil tokenizing dilakukan filtering untuk menyaring atau menghapus apakah kata tersebut terdapat stopword atau tidak, jika tidak ada maka ditambahkan, jika sudah ada maka akan mengembalikan hasil filtering. Penelitian ini menggunakan fungsi StopWordRemoverFactory() dari library Sastrawi yang ditunjukkan pada Pseudo Code Tahap filtering

```
# filtering (stopword / menghapus kata yang tidak penting)
def filteringText(text): # remove stopwords in a text
    listStopwords = StopWordRemoverFactory().get_stop_words()
    filtered = []
    for text in text:
        if text not in listStopwords:
            filtered.append(text)
    text = filtered
    return text

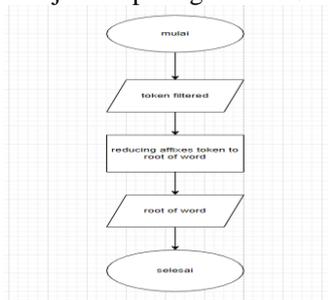
# proses pengaplikasian fungsi filtering
df_data['comment_preprocessing'] = df_data['comment_preprocessing'].apply(filteringText)
df_data.head()
```

Pseudo Code Tahap filtering
(sumber : code program 2023)

Untuk melakukan filtering terdapat def filteringText() untuk mendefinisikan fungsi filtering, selanjutnya menggunakan fungsi StopWordRemoverFactory() untuk menghapus kata yang tidak penting. Kemudian membuat variable kosong “filtered”, dan lakukan looping dari setiap baris dari teks, jika dalam suatu baris tidak terdapat listStopword, maka akan ditambahkan di variabel “filtered”. Kemudian hasilnya diletakkan di variabel “filtered”. Dan untuk pengaplikasiannya menggunakan fungsi apply() untuk menampilkan datanya menggunakan fungsi head().

Stemming

Stemming merupakan proses penghapusan kata yang memiliki imbuhan menjadi kata dasar. Alur kerja pada proses stemming ditunjukkan pada gambar 3.9



Gambar flowchart stemming
(sumber : rancangan 2023)

Pada gambar flowchart stemming dijelaskan alur kerja yang dimulai dari memasukan token hasil filtering, selanjutnya dilakukan proses penghapusan kata imbuhan pada token dan akan menghasilkan kata dasar. Proses ini bisa dilakukan dengan menggunakan Pseudo Code Tahap Stemming

```
# stemming (pembentukan kata dasar)
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

# reducing a word to its word stem that affixes to suffixes and prefixes or to the root of word
def stemmingText(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    text = (stemmer.stemword) for word in text
    return text

# proses pengaplikasian fungsi stemming
df_data['comment_preprocessing'] = df_data['comment_preprocessing'].apply(stemmingText)
df_data.drop_duplicates(subset = 'comment_clean', inplace = True)
df_data.head()
```

Pseudo Code Tahap Stemming
(sumber : code program 2023)

Pada gambar Pseudo Code Tahap Stemming bisa di lihat bahwa tahap stemming menggunakan library dari sastrawi import StemmerFactory dan StopWordRemoverFactory. Kemudian fungsi def stemmingText() untuk mendefinisikan fungsi stemming. Selanjutnya create stemmer dengan menggunakan fungsi StemmerFactory() dan Create_Stemmer() untuk menghapus imbuhan kata menjadi kata dasar yang ada dalam teks. Untuk pengaplikasiannya menggunakan apply(). Selanjutnya menghapus baris duplikat pada kolom “comment_clean” dan menggunakan kolom yang sama pada hasilnya menggunakan fungsi drop_duplicates(). Dan untuk menampilkan hasilnya menggunakan fungsi head(). Untuk melihat hasil dari preprocessing bisa dilihat dari gambar Pseudo Code Hasil Preprocessing

```
# cek hasil preprocessing
df_data.to_csv('tiktok.csv')
```

Pseudo Code Hasil Preprocessing
(sumber : code program 2023)

Pada gambar Pseudo Code Hasil Preprocessing dapat dilihat dengan menggunakan fungsi to_csv(), maka hasilnya akan tersimpan pada komputer.

Analisis sentimen

Tahapan analisis merupakan tahapan yang sangat penting pada penelitian ini karena berfungsi untuk mengelompokkan data komentar tiktok yang telah diambil menggunakan metode lexicon. Proses dari lexicon ini bertujuan untuk menghitung nilai polarity. Lexicon adalah sebuah kamus yang terdiri dari kata dan nilai polarity per kata. Lexicon dibagi menjadi dua yaitu lexicon positive dan lexicon negative. Selanjutnya dari data yang sudah dilakukan proses preprocessing akan dilakukan pengecekan satu per satu agar bisa mengetahui nilai polarity. Apabila ada sebuah kata yang tercantum dalam lexicon maka nilai polarity akan ditambahkan sesuai dengan nilai polarity yang termasuk ke dalam lexicon. Untuk melihat proses analisis

sentimen ini, ditunjukkan pada gambar Pseudo Code Proses Analisis Sentimen

```
# proses analisis sentimen dengan lexicon
lexicon_positive = dict()
import csv
with open('positive.tsv', 'r') as csvfile:
    reader = csv.reader(csvfile, delimiter='\t')
    for row in reader:
        if "word" not in row and "weight" not in row:
            lexicon_positive[row[0]] = int(row[1])

lexicon_negative = dict()
import csv
with open('negative.tsv', 'r') as csvfile:
    reader = csv.reader(csvfile, delimiter='\t')
    for row in reader:
        if "word" not in row and "weight" not in row:
            lexicon_negative[row[0]] = int(row[1])

def analisis_sentimen(text):
    #for word in text:
    score = 0
    for word in text:
        if word in lexicon_positive:
            score = score + lexicon_positive[word]
        for word in text:
            if word in lexicon_negative:
                score = score + lexicon_negative[word]
    polarity=""
    if (score > 0):
        polarity = 'positive'
    elif (score < 0):
        polarity = 'negative'
    else:
        polarity = 'neutral'
    return score, polarity
```

Pseudo Code 3.10 Proses Analisis Sentimen (sumber : code program 2023)

Pada gambar pseudo code 3.10 bisa dilihat proses untuk melakukan analisis sentimen menggunakan metode lexicon. Langkah pertama yang harus dilakukan adalah memasukan file lexicon positive dan lexicon negative dalam hal ini yaitu positive.tsv dan negative.tsv. Untuk lexicon positive, hal yang harus dilakukan adalah membuat variabel untuk kamus lexicon dengan menggunakan fungsi dict(), setelah itu import csv, selanjutnya buka file positive.tsv sebagai file csv menggunakan fungsi open() dan baca file tersebut menggunakan fungsi csv.reader(). Kemudian melakukan looping pada setiap baris file positive.tsv, jika “word” tidak terdapat pada baris dan “weight” tidak terdapat pada baris, maka “word” akan masuk ke row[0] dan “weight” masuk ke row[1] dengan “word” sebagai key dan “weight” sebagai value. Sama hal nya seperti lexicon negative. Langkah selanjutnya melakukan analisis untuk menghitung polarity score, lalu membuat variabel score dengan nilai 0. Selanjutnya dilakukan looping untuk kata pada teks, jika terdapat kata yang berada pada lexicon positive, maka score akan ditambahkan dengan kata lexicon positive. Dan sama hal nya seperti lexicon negative. Dan yang terakhir menghitung nilai polarity, jika score lebih besar dari 0 maka nilai polarity-nya positive, jika score kurang dari 0 maka nilai polarity-nya negative, dan selain dari kedua itu netral. Untuk implementasi analisis sentimen ditunjukkan pada gambar Pseudo Code

```
import pandas
# implementasi fungsi analisis sentimen dan penggabungan data dari setiap analisis sentimen
results = df_data['comment_preprocessing'].apply(analisis_sentimen)
results = list(zip(results))
df_data['polarity_score'] = results[0]
df_data['polarity'] = results[1]
print(df_data['polarity'].value_counts())
df_data
df_data.to_csv("HasilTiktok.csv")
```

Pseudo Code Implementasi Fungsi Analisis Sentimen (sumber : code program 2023)

3. HASIL DAN PEMBAHASAN

Hasil dari scrape data menghasilkan beberapa kolom, yaitu kolom 1665s4c yang merupakan kolom nama pengguna tiktok, kolom 1ht48ju merupakan kolom yang menunjukkan komentar para pengguna tiktok dan kolom 1esugaz yang menunjukkan tanggal berapa komentar itu di posting. Namun penelitian ini hanya menggunakan kolom 1665s4c dan 1ht48ju. Setelah itu download hasil data tersebut dengan jenis file CSV (Comma Sparated Value) kemudian buka file tiktok.csv

Text Preprocessing

Data yang sudah dikumpulkan melalui hasil scraping, kemudian memasuki proses text preprocessing agar menjadi data yang terstruktur dan dapat diolah oleh bahasa pemrograman. Pada tahap text preprocessing ini setidaknya ada lima tahapan utama yaitu cleaning text, case folding, tokenizing, filtering, dan stemming.

Tabel Data Terpilih Opini Pengguna Tiktok

No	Ulasan Aplikasi
1	lah kan berdesakan itu gara gara lari ada gas air mata kalo gada gas air mata gabakal berdesakan.
2	menghindari dari salah.
3	lah iyo, udara full gas air mata otomatis oksigen menurun ak kuliah teknik kimia.

(sumber: output program 2023)

Pada Tabel diatas menunjukkan tiga data terpilih 524 komentar yang most relevan. selanjutnya data ulasan tersebut melewati tahap text preprocessing yang meliputi cleaning text, case folding, tokenizing, filtering, dan stemming.

Cleaning Text

Cleaning text, dimulai dari memasukkan teks yang akan dilakukan cleaning dalam hal ini yaitu opini pada postingan user twitter, kemudian dilakukan proses dengan menghapus link, mention, hashtag, nomor, emoticon, symbol, dan semua punctuation. Kemudian akan menghasilkan text yang bersih. ditampilkan output dari tahap cleaning text. Dimana pada kolom opini terdapat perubahan pada angka dan tanda baca seperti titik dan koma ditandai dengan laber warna biru. Semua karakter dihilangkan dan hasilnya dapat dilihat pada tabel dibawah.

Tabel Hasil Cleaning Text

No	Ulasan Sebelum Perubahan	Ulasan Setelah Perubahan
1	lah kan berdesakan itu gara gara lari ada gas air mata kalo gada gas air mata gabakal berdesakan	lah kan berdesakan itu gara gara lari ada gas air mata kalo gada gas air mata gabakal berdesakan
2	menghindari dari salah	menghindari dari salah
3	lah iyo udara full gas air mata otomatis oksigen menurun ak kuliah teknik kimia	lah iyo udara full gas air mata otomatis oksigen menurun ak kuliah teknik kimia

(sumber: output program 2023)

Case Folding

Case folding, dimulai dengan memasukkan teks hasil dari proses sebelumnya yaitu cleaning text, kemudian diberlakukan proses merubah huruf menjadi kapital menjadi huruf kecil, akan menghasilkan teks yang berisi huruf kecil semua. Dimana pada tabel opini ditampilkan perbandingan beberapa huruf kapital yang diubah menjadi huruf kecil di label dengan warna hijau.

Tabel Hasil Case Folding

No	Ulasan Sebelum Perubahan	Ulasan Setelah Perubahan
1	lah kan berdesakan itu gara gara lari ada gas air mata kalo gada gas air mata gabakal berdesakan	lah kan berdesakan itu gara gara lari ada gas air mata kalo gada gas air mata gabakal berdesakan
2	menghindari dari salah	menghindari dari salah
3	lah iyo udara full gas air mata otomatis oksigen menurun ak kuliah teknik kimia	lah iyo udara full gas air mata otomatis oksigen menurun ak kuliah teknik kimia

(sumber: output program 2023)

Tokenizing

Tokenizing, dimulai dengan memasukkan teks hasil proses case folding, lalu dilakukan proses pemisahan bentuk kalimat menjadi bentuk token.

Tabel Hasil Tokenizing

No	Ulasan Sebelum Perubahan	Ulasan Setelah Perubahan
1	lah kan berdesakan itu gara gara lari ada gas air mata kalo gada gas air mata gabakal berdesakan	[lah, kan, berdesakan, itu, gara, gara, lari, ada, gas, air, mata, kalo, gada, gas, air, mata, gabakal, berdesakan]

2	menghindari dari salah	[menghindari, dari, salah]
3	lah iyo udara full gas air mata otomatis oksigen menurun ak kuliah teknik kimia	[lah, iyo, udara, full, gas, air, mata, otomatis, oksigen, menurun, ak, kuliah, teknik, kimia]

(sumber: output program 2023)

Filtering

filtering, dari teks hasil proses tokenizing dilakukan filtering untuk menyaring apakah sebuah kata terdapat stopword atau tidak, jika tidak maka 36 Institut Sains Dan Teknologi Nasional ditambahkan, jika ada maka akan mengembalikan hasil filtering. Dimana pada kolom opini ketika sebelum diubah, terdapat kata yang tidak memiliki makna seperti lagi, akan, bisa, di, itu, ada, juga, dan kata yang ditandai oleh label warna kuning.

Tabel Hasil Filtering

No	Ulasan Sebelum Perubahan	Ulasan Setelah Perubahan
1	[lah, kan, berdesakan, itu, gara, gara, lari, ada, gas, air, mata, kalo, gada, gas, air, mata, gabakal, berdesakan]	[lah,kan,berdesakan, gara, gara, lari, gas, air, mata, kalo, gada, gas, air, mata, gabakal, berdesakan]
2	[menghindari, dari, salah]	[menghindari, salah]
3	[lah, iyo, udara, full, gas, air, mata, otomatis, oksigen, menurun, ak, kuliah, teknik, kimia]	[lah, iyo, udara, full, gas, air, mata, otomatis, oksigen, menurun, kuliah, teknik, kimia]

(sumber: output program 2023)

Stemming

Stemming adalah proses menghilangkan kata yang memiliki imbuhan menjadi kata dasar. Dimana pada kolom komentar sebelum perubahan banyak kata yang memiliki kata imbuhan misalnya -an, ter-, se-, ber-, nya-, dan lain sebagainya. Kata imbuhan tersebut di tandai dengan label warna ungu. Maka kata imbuhan akan dihapus dan membentuk kata dasar, hasilnya dapat dilihat pada tabel berikut.

Tabel Hasil Stemming

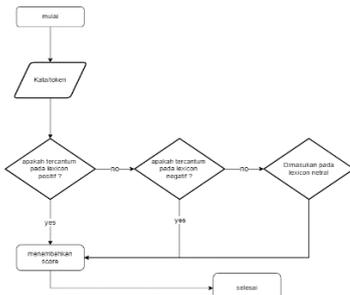
No	Ulasan Sebelum Perubahan	Ulasan Setelah Perubahan
1	[lah, kan, berdesakan, gara, gara, lari, gas, air, mata, kalo, gada, gas, air, mata, gabakal, berdesakan]	[lah, kan, gara, gara, lari, gas, air, mata, kalo, gada, gas, air, mata, gabakal]
2	[menghindari, salah]	[hingar, salah]

3	[lah, iyo, udara, full, gas, air, mata, otomatis, oksigen, menurun , kuliah, teknik, kimia]	[lah, iyo, udara, full, gas, air, mata, otomatis, oksigen, nurun, kuliah, teknik, kimia]
---	--	--

(sumber: output program 2023)

Analisis Sentimen Menggunakan Metode Lexicon

Tahapan mengelompokan data opini kolom komentar pada salah satu video tiktok yang telah di data menggunakan metode lexicon. Proses ini bertujuan untuk menghitung nilai polarity. Lexicon merupakan sebuah kamus yang terdiri dari kata dan nilai polarity per kata. Lexicon juga terdiri dari lexicon positive dan lexicon negative. Kemudian dari data yang sudah dilakukan proses preprocessing, selanjutnya akan dilakukan pengecekan satu per satu untuk mengetahui nilai polarity. Apabila sebuah kata tercantum dalam kamus lexicon maka nilai polarity akan ditambahkan sesuai dengan nilai polarity yang terkandung dalam lexicon.



Gambar Alur Perhitungan Polarity score (sumber:rancangan2023)

Salah satu contoh dari ulasan sebagai berikut : [Jelek, Bagus, Cakep, Mantap]

polarity score > 0, maka sentimen positif
 polarity score < 0, maka sentimen negatif
 polarity score = 0, maka sentimen netral

Tabel Perhitungan Polarity Score

Kata (Teks)	Lexicon Positif	Lexicon Negatif	Skor
Jelek	-	Tercantum (-5)	-5
Bagus	Tercantum (2)	Tercantum (-4)	-2
Cakep	Tercantum (5)	Tercantum (-1)	4
Mantap	Tercantum (5)	Tercantum (-3)	2
Total Polarity Score			-1 (Sentimen Negatif)

(sumber: output program 2023)

Pada Tabel diatas memperlihatkan hasil contoh ulasan diatas setelah melewati proses perhitungan polarity score menggunakan metode lexicon dengan total skor sebesar (-1) maka ulasan tersebut masuk ke dalam kategori negatif. Apabila pada ulasan memiliki jumlah polarity score yang bernilai positif akan menjadikan data ulasan tergolong ke dalam sentimen positif, begitu pula sebaliknya jika polarity score bernilai negatif atau minus maka ulasan termasuk ke dalam sentimen negatif. Jika nilai polarity score sama dengan 0 maka ulasan tersebut masuk ke dalam kelas netral.

Visualisasi

Setelah tahap klasifikasi data menggunakan metode lexicon, selanjutnya memasuki tahap akhir yaitu memvisualisasikan hasil dari analisis sentiment dalam bentuk diagram batang, diagram lingkaran, dan wordcloud. Pada bentuk diagram bisa dilihat seberapa banyak data yang tercantum pada sentiment positif, negative, dan netral. Untuk diagram lingkaran bisa dilihat persentase sentiment positif, negative, dan netral. Untuk wordcloud dapat memperlihatkan kata-kata yang muncul pada hasil sentiment positif, negative, dan netral.

Instrumen Penelitian

Adapun alat-alat penelitian yang digunakan dalam proses penelitian ini, baik perangkat keras ataupun perangkat lunak sebagai berikut:

Perangkat keras yang digunakan dalam penelitian ini adalah laptop ASUS VivoBook 14 dengan spesifikasi sebagai berikut :

- Processor Intel(R) Core(TM) i3-8145U CPU @ 2.10GHz (4 CPUs), ~2.3GHz
- RAM 12 GB

Perangkat Lunak

- Python 3 merupakan bahasa pemrograman yang digunakan.
- Jupyter Notebook merupakan aplikasi yang penting dikuasai oleh data scientist.
- Microsoft Word merupakan media untuk menuliskan hasil skripsi peneliti.

Pada tabel ini ditunjukkan bahwa sentimen negatif mempunyai presentase terbesar yaitu 57,80% yang berarti sebanyak 300 komentar menunjukkan respon yang tidak baik pada kasus kerusuhan yang terjadi di stadion kanjuruhan Malang. Dan untuk sentimen positif mempunyai presentase 18,49% yang berarti sebanyak 96 komentar menunjukkan respon yang baik, bisa diartikan bahwa sentimen positif dalam kasus ini mempunyai presentase terkecil. Sedangkan untuk sentimen netral

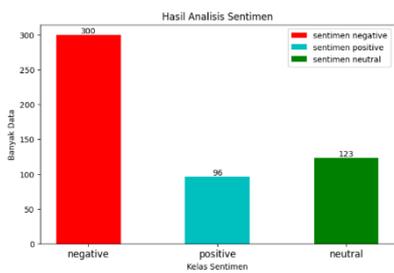
mempunyai presentase 23,69% yang berarti sebanyak 123 komentar menunjukkan respon yang biasa-biasa saja dalam kasus kerusuhan yang terjadi di stadion kanjuruhan Malang.

Tabel Hasil Analisis Sentimen

No	Kelas sentimen	Banyak data	Dalam bentuk persentase
1	Negatif	300	57,80%
2	Positif	96	18,49%
3	Netral	123	23,69%

(sumber: output program 2023)

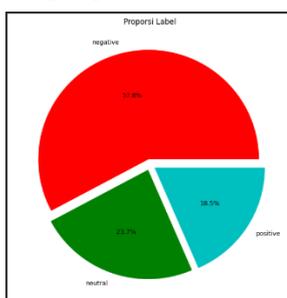
Maka akan terlihat hasil dari visualisasi dalam bentuk diagram batang. Di dalam diagram batang tersebut dapat kita lihat banyaknya komentar negatif yang ditunjukkan dengan warna merah sebanyak 300, dan warna cyan menunjukkan komentar positif sebanyak 96, dan yang terakhir dengan warna hijau menunjukkan komentar netral sebanyak 123. Sehingga dapat kita simpulkan komentar masyarakat Indonesia yang menggunakan aplikasi tiktok mengenai peristiwa kerusuhan di stadion kanjuruhan Malang adalah negatif.



Gambar Tampilan Visualisasi Diagram Batang

(sumber: output program 2023)

Pada Gambar dibawah akan ditunjukkan hasil dari visualisasi dalam bentuk lingkaran. Pada diagram tersebut dapat dilihat hasil dari komentar tiktok tentang peristiwa kerusuhan yang terjadi di stadion kanjuruhan Malang. Dalam diagram ini komentar negatif mendapatkan presentase yang paling besar yaitu 57,80%, sementara komentar positif 18,49% dan yang terakhir komentar netral dengan presentase 23,69%.



Gambar Tampilan Visualisasi Diagram Piechart

(sumber: output program 2023)

Pada Gambar dibawah akan ditampilkan hasil dari visualisasi dalam bentuk wordcloud. Pada gambar tersebut dapat dilihat komentar yang sering muncul dalam video tentang peristiwa kerusuhan di stadion kanjuruhan Malang. Kata-kata yang sering muncul adalah “yang”, “gas”, “air”..



Gambar Tampilan Visualisasi Diagram Wordcloud

(sumber: output program 2023)

4. SIMPULAN

Simpulan yang dapat diambil berdasarkan penelitian analisis sentimen mengenai tragedi kerusuhan yang terjadi di stadion kanjuruhan Malang yang menghasilkan kesimpulan sebagai berikut :

1. Analisis sentimen pada ulasan aplikasi tiktok tragedi stadion kanjuruhan menggunakan metode lexicon mendapatkan hasil sesuai yang di harapkan yaitu dapat mengelompokkan kelas-kelas sentimen seperti negatif, positif dan netral
2. Dari 524 baris data yang diambil dari kolom komentar video tiktok, terpilih 519 baris setelah melakukan tahapan text preprocessing. Selanjutnya melakukan analisis sentimen yang menghasilkan kelas sentimen negatif sebanyak 300 kata dengan presentase 57,80%, sentimen positif 96 kata dengan presentase 18,49%, sentimen netral 123 kata dengan presentase 23,69%
3. Dari hasil analisis sentimen ini dapat dilihat masih adanya masyarakat yang berkomentar positif tentang tragedi kanjuruhan ini. Dan dari hasil ini kita dapat melihat bahwa banyak masyarakat yang tidak suka akan kejadian ini, hal ini dibuktikan dengan banyaknya masyarakat yang berkomentar relative negatif sebanyak 57,80%

Saran

Dari penelitian yang sudah dilakukan, maka saran untuk penelitian selanjutnya yaitu :

1. Masih banyak kata yang tidak baku dan ini akan mempengaruhi polarity score dalam menentukan kelas sentimen. Dan diharapkan untuk bisa membuat sebuah

proses yang dapat merubah kata yang tidak baku menjadi baku berdasarkan KBBI.

2. Penelitian ini hanya menggunakan metode Lexicon, diharapkan pada penelitian selanjutnya bisa menggunakan metode yang lain untuk proses analisis sentimen.

5. DAFTAR PUSTAKA

- AArief, R., & Imanuel, K. (2019). ANALISIS SENTIMEN TOPIK VIRAL DESA PENARI PADA MEDIA SOSIAL TWITTER DENGAN METODE LEXICON BASED Universitas Gunadarma 1, 2 Jalan Margonda Raya No 100 Depok Jawa Barat 16424 Sur-el : rifiana@staff.gunadarma.ac.id 1 , karel4404@gmail.com 2. *Jurnal Ilmiah Matrik*, 21(3).
- BBuana, T., & Maharani, D. (2020). PENGGUNAAN APLIKASI TIK TOK (VERSI TERBARU) DAN KREATIVITAS ANAK. *Jurnal Inovasi*, 14(1).
- IrIrmayani, W. (2021). VISUALISASI DATA PADA DATA MINING MENGGUNAKAN METODE KLASIFIKASI NAÏVE BAYES. www.bsi.ac.id
- KKhairunnisa, S., Adiwijaya, A., & Faraby, S. Al. (2021). Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19). *Jurnal Media Informatika Budidarma*, 5(2), 406. <https://doi.org/10.30865/mib.v5i2.2835>
- Mahendrajaya, R., Buntoro, G. A., & Setyawan, M. B. (2019). ANALISIS SENTIMEN PENGGUNA GOPAY MENGGUNAKAN METODE LEXICON BASED DAN SUPPORT VECTOR MACHINE. <http://studentjournal.umpo.ac.id/index.php/komputek>
- Acai Sudirman, dkk; *BISNIS INFORMASI*, ISBN : 9786235002224 Widina Media Utama, 2024
- Mai, P., Tarigan, S., Tata Hardinata, J., Qurniawan, H., Safii, M., Winanjaya, R., Studi, P., Informasi, S., Tunas, S., & Pematangsiantar, B. (2022). IMPLEMENTASI DATA MINING MENGGUNAKAN ALGORITMA APRIORI DALAM MENENTUKAN PERSEDIAAN BARANG (STUDI KASUS : TOKO SINAR HARAHAP (Vol. 12, Issue 2). <https://jurnal.umj.ac.id/index.php/just-it/index>
- WA Rahmat, SM Ladjamuddin, DT Awaludin, Perbandingan Algoritma Decision Tree, Random Forest dan Naive Bayes pada Prediksi Penilaian Kepuasan Penumpang Maskapai Pesawat Menggunakan Dataset Kaggle, Vol 12 No 2 (2023): *Jurnal Rekayasa Informasi*, ISSN: 2252-7354, <https://ejournal.istn.ac.id/index.php/rekayasainformasi/article/view/1726>
- Romzi, M., & Kurniawan, B. (2020). PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA (Issue 2).
- M Firdaus, SM Ladjamuddin; Design And Build Dual Link Internet Service Provider Using Load Balancing Method With Round Robin Algorithm; Vol 11 No 2 (2022): *Jurnal Incomtech*, ISSN 2337-6805; <https://ejournal.istn.ac.id/index.php/incomtech/article/view/1481>
- Undap, M. G., Rantung, V. P., & Rompas, P. T. D. (n.d.). Analisis Sentimen Situs Pembajak Artikel Penelitian Menggunakan Metode Lexicon-Based. In *JOINTER-JOURNAL OF INFORMATICS ENGINEERING* (Vol. 02, Issue 02).
- Muhammad Reza Pahlevi, Siti Madinah Ladjamuddin, Blockchain Technology Implementation On E-Vote Website Using Python Programming Language, Vol 10 No 1 (2021), *Jurnal Incomtech*, ISSN 2337-6805, <https://ejournal.istn.ac.id/index.php/incomtech/article/view/1065>
- Wahyuni, R. D., & Utomo, A. N. (2022). PENGGUNAAN METODE LEXICON UNTUK ANALISIS SENTIMEN PADA ULASAN APLIKASI KAI ACCESS DI GOOGLE PLAY STORE USING THE LEXICON METHOD FOR ANALYSIS SENTIMENTS ON KAI ACCESS APPLICATION REVIEWS ON GOOGLE PLAY STORE. *Jurnal Rekayasa Informasi*, 11(2).
- Neny Rosmawarni, dkk; *E-COMMERCE*, Editor: Ingrid Yanuar Risca Pratiwi, S.S.T., M.Tr.T., ISBN: 978-623-88989-5-4, Penamuda Media, 2024
- Fajri. (2019). InSet (Indonesia Sentiment Lexicon). <https://github.com/fajri91/InSet>