 ISSN:2302-8149
e-ISSN:2540-9719

Analisis Sentimen menggunakan Algoritma Support Vector Machine pada Covid_19

Sentiment Analysis using the Support Vector Machine Algorithm on Covid_19

[1]Adytyo Wahyu Nugroho, [2] Norhikmah*
[1]Informatika, Fakultas Komputer, Universitas Amikom Yogyakarta
[2]Sistem Informasi, Ilmu Komputer, Universitas AMIKOM Yogyakarta
Jln.Ring Road Utara, Condong Catur,Sleman,Yogyakarta,Indonesia
*e-mail:*hikmah@amikom.ac.id*

(***received*:** 18 December 2023, ***revised*:** 24 July 2024, ***accepted*:** 27 July 2024)

Abstrak

Perkembangan teknologi informasi yang masif ini mempermudah dalam kehidupan masyarakat dalam berbagai bidang, salah satunya adalah sosial media, Sosial media yang masyarakat gunakan banyak mendapatkan informasi mengenai berita atau peristiwa yang sedang terjadi di Indonesia, salah satunya adalah sosial media *Twitter* yang banyak memberikan informasi untuk masyarakat Indonesia salah satunya adalah informasi mengenai *Covid-19* yang sedang marak terjadi di wilayah Indonesia. Analisis sentimen merupakan salah satu cabang dari *Natural Language Processing* (NLP) yang dapat membantu mengetahui sentimen yang terjadi pada masyarakat. Penelitian ini menggunakan data berupa *tweet* untuk melakukan sentimen analisis yang didapat pada sosial media *Twitter*. Penelitian ini memanfaatkan salah satu algoritme dari *Supervised Learning* yaitu *Support Vector Machine* dalam dalam penelitian ini menggunakan tiga (3) kernel untuk *Support Vector Machine* masing masing adalah Linear, *Radial basis function* dan *Polynomial* untuk mencari pada kernel berapa menghasilkan nilai akurasi tertinggi. Dari percobaan yang dilakukan dengan menggunakan pembagian data untuk *training* sebanyak 70% dan untuk data *testing* sebanyak 30% dari total data 6000 data, dihasilkan nilai akurasi untuk metode *Support Vector Machine* pada kernel Linear menghasilkan nilai akurasi sebesar *89%* dan untuk kernel *Radial basis function* akurasi sebesar *90%* dan untuk kernel *Polynomial* menghasilkan akurasi sebesar *88%*. Sehingga disimpulkan untuk ke-tiga (3) kernel untuk pengujian metode *Support Vector Machine* pada kernel *Radial basis function* menghasilkan nilai akurasi terbaik.
Kata kunci: sentimen analisis, *support vector machine* (SVM), *twitter, covid-19*

Abstract

This massive development of information technology makes it easier for people's lives in various fields, one of them is social media, social media that people use a lot to get information about news or events that are happening in Indonesia, one of which is social media Twitter which provides a lot of information for the people of Indonesia, one of which is information about Covid-19 which is currently rife in the territory of Indonesia Sentiment analysis is a branch of Natural Language Processing (NLP) which can help determine the sentiments that occur in society. This study uses data in the form of tweets to carry out sentiment analysis obtained on Twitter social media.This research utilizes one of the Supervised Learning algorithms, namely Support Vector Machine. In this study, three (3) kernels are used for the Support Vector Machine, each of which is Linear, Radial basis function and Polynomial, to find which kernel produces the highest accuracy value. From the experiments carried out using data sharing for training as much as 70% and for testing data as much as 30% of the total data of 6000 data, the resulting accuracy value for the Support Vector Machine method on the Linear kernel produces an

 ISSN:2302-8149
e-ISSN:2540-9719

accuracy value of 89% and for the Radial kernel base function accuracy by 90% and for the Polynomial kernel it produces an accuracy of 88%. So it is concluded for the three (3) kernels for testing the Support Vector Machine method on the Radial basis function kernel to produce the best accuracy value.
Keywords: *sentiment analysis, support vector machine (SVM), twitter, covid-19*

1 Pendahuluan

World Health Organization (WHO) mendeklarasikan virus COVID-19 sebagai pandemi global pada 11 Maret 2020 WHO melaporkan lebih 52 juta orang terkonfirmasi positif Covid-19 dan 1,2 juta orang meninggal dunia pada minggu kedua bulan November 2020. Sementara Indonesia mencatat 463 ribu orang terkonfirmasi positif dengan korban meninggal telah mencapai 15.148 orang. Penyebaran virus secara cepat melalui kontak fisik memaksa semua negara menerapkan social distancing dan physical distancing untuk mengurangi interaksi. Faktor utama penularan melalui droplets yang dikeluarkan selama berbicara, batuk, atau bersin menjadikan penerapan pembatasan sosial merupakan strategi yang paling banyak diadopsi disaat belum ada vaksin [1] Data tersebut kemudian di proses dengan melakukan *Text Mining* lalu kemudian di lakukan klasifikasi Tweet berdasarkan pembagian sentimen yaitu bermuatan Positif dan Negatif.

Klasifikasi ini menggunakan algoritme *Support Vector Machine*. Klasifikasi di gunakan untuk memberikan kemudahan dalam melihat opini yang bernilai Positif dan Negatif Tingkat akurasi dari algoritme akan memberikan pengaruh pada hasil klasifikasi[2]Metode *Support Vector Machine* (SVM) digunakan dalam penelitian ini untuk mengklasifikasi teks sentimen. Dalam komputasi metode ini akan memakan waktu yang lama apabila fitur yang digunakan cukup banyak, dan digunakan fitur fitur untuk menyeleksi yang relevan, menghindari overfitting. Seleksi fitur yang di gunakan adalah seleksi fitur Weight by SVM yang di harapkan bisa meningkatkan nilai akurasi yang signifikan [3]**.**

Algoritme *Support Vector Machine* dapat di gunakan untuk mengklasifikasi opini-opini tersebut. Berdasarkan penelitian yang di lakukan sebelum nya digunakan algoritme *Support Vector Machine* menghasilkan tingkat akurasi yang tinggi dalam melakukan sentimen analisis Penelitian yang di lakukan ditemukan algoritme *Support Vector Machine* disandingkan dengan fitur TF-IDF menghasil tingkat akurasi sebesar 86% lebih unggul dibandingkan dengan metode *Naïve Bayes* yang juga dikolaborasikan dengan TF-IDF ketika melakukan analisis opini masyarakat mengenai Gojek pada *Twitter* yang dibagi menjadi dua kelas yaitu positif dan negatif. TF-IDF digunakan agar dapat membantu dalam meningkatkan akurasi apabila disandingkan dengan metode *Support Vector Machine* dibanding ekstrasi fitur seperti *Ratio dan N-Gram* [4]**.**

2 Tinjauan Literatur

Di Indonesia virus tersebut sudah menjangkit 34 provinsi dan 415 kabupaten / kota. Pada saat tulisan ini dibuat data pada Worldmeter yang bersumber dari data *World Health Organization* (WHO) sudah terdata sekitar 3.015.411 tersebar di sekitar 213 negara dan menyebabkan kematian sebesar 207.968 korban jiwa[5].

Algoritme *Support Vector Mahcine*(SVM) dan seleksi fitur juga telah dilakukan pada [6], dengan menggunakan Information Gain (IG) sebagai seleksi fitur mendapatkan nilai akurasi terbaik yaitu 76,66%. Kekurangan dari seleksi fitur *informasi gain* yaitu mempunyai nilai yang rendah jika fitur tersebut muncul didalam semua kelas Seleksi fitur lain juga digunakan untuk meningkatkan akurasi seperti pada penelitian [7]. Dalam permasalahan untuk analisis sentimen menggunakan salah satu algoritma *Support Vector Machine*(SVM),karena dapat memberikan tingkat hasil yang lebih baik jika dibandingkan oleh salah satu metode klasifikasi sejenis seperti *Artificial Neural Network*(ANN) terutama pada menyelesaikan solusi dikarenakan *Support Vector Machine* dapat menemukan solusi yang lebih optimal[9].

Penerapan algoritme dari *Support vector machine* dalam melakukan klasifikasi buku yang terdapat didalam perpustakaan berbasis digital *on-line*, buku-buku yang cocok yang sesuai dengan keilmuan yang dipakai untuk melakukan bahan analisis dalam memberikan rekomendasi judul untuk mahasiswa dan sesuai refrensi buku yang diingikan.[10].Dalam penelitian yang menyangkut komparasi Algoritme Klasifikasi *Machine Learning* dan *feature selection* pada Analisis Sentimen pada Review Film, dari penelitian yang dilakukan tersebut diperoleh hasil bahwa algoritme *Support Vector Machine*

 ISSN:2302-8149
e-ISSN:2540-9719

mempunyai akurasi yang tinggi dibanding algoritme klasifikasi yang lainnya serta seleksi fitur dapat menghasilkan akurasi yang tinggi dibanding metode *Support Vector Machine.*[11].

Algoritme klasifikasi yang saat ini mempunyai tingkat akurasi tingkat akurasi sebesar 80.77%. pada peneltian Sentimen Analisis Operasi Tangkap Tangan KPK Menurut Masyarakat[12]. *Support Vector Machine* dapat melakukan klasifikasi dari data opini yang beredar di sosial media *Twitter* dengan hasil dari tingkat akurasi sebesar 91,67% [13]. Salah satu dari penelitian yang melakukan pemanfaatan data yang berasal dari sosial media *twitter* adalah data yang mengenai politik yang terkait dengan sentimen calon presiden Jokowi dan Prabowo untuk melakukan hasil dari sentimen menggunakan algoritme *Support Vector Machine* menghasilkan tingkat akurasi yang diperoleh dari melakukan klasifikasi adalah sebesar 86%[14]. Analisis Sentimen Berita Artis Dengan menggunakan Algoritma *Support Vector Machine.* Pada penelitian tersebut memberikan akurasi yang ditampilkan dalam *confusion matrix* dan kurva *ROC* dengan tingkat akurasi sebesar 73.33% dan AUC sebesar 0,774 dalam hal ini algoritma *Support Vector Machine* dapat melakukan optimasi yang berdampak pada peningkatan nilai akurasi [15]. Dari jumlah data yang didapat dari situs *A World of Tweets dot com* negara Indonesia mendapatkan posisi ketiga didunia sebagai negara yang banyak menggunakan media sosial *Twitter* dan menulis (*tweet)* yakni sebesar 11,39% diperoleh dari data pada tahun 2010 sampai dengan tahun 2012 dengan pengguna sebanyak 383 juta profil[16]. *Support Vector Machine* (SVM) adalah salah satu metode klasifikasi yang mampu menyelesaikan permasalahan yang bersifat nonlinier. SVM bekerja sesuai dengan proses yang dilakukan dengan mencari fungsi atau proses pemisah (*hyperplane)* dengan cara memaksimalkan jarak antara kelas pada setiap sentimennya[17].

Pada penelitian memiliki kontribusi yaitu berdasarkan uji perbandingan pada algoritma SVM menggunakan 3 kernel yaitu *Linear, Radial basis function* dan *Polynomial* untuk mendapatkan pada kernel apa menghasilkan nilai akurasi tertinggi dengan menggunakan metode evaluasi yaitu confusion matrix, dengan menggunakan data penyebaran covid_19.

3 Metode Penelitian

Dalam penelitian ini dijelaskan metode yang dipakai dalam penelitian terdiri dari beberapa tahapan meliputi Mengumpulkan Dataset menggunakan aplikasi Webharvy, Labeling Data, Preprocessing, Transformation TF-IDF, Training Model Klasifikasi (*Support Vector Machine*) dan Hasil Klasifikasi. Alur dari penelitian dapat di jelaskan pada alur penelitian sebagai berikut:

1. Pengumpulan data
   Dataset yang akan di gunakan dalam penelitian ini didapat melalui pengumpulan menggunakan WebHarvy yang mana di Webharvy sudah tersedia data berupa *tweet* dengan banyak data sejumlah 8577 data, data yang dikumpulkan dilakukan dari bulan Januari 2020 sampai dengan bulan Juli 2021 data yang sudah dikumpulkan berupada data teks tidak terstruktur dan harus diolah untuk melakukan penelitian. Dari sebanyak 8577 data yang sudah diolah kemudian dipisahkan data yang bernilai positif dan bernilai negatif yang mana dilakukan untuk mempermudah dalam melakukan penelitian dengan menggunakan data yang sudah diolah.
2. Pelabelan Data
   Pelabelan data yang akan di gunakan penelitian di lakukan menggunakan metode textblob dan label yang di gunakan adalah label negatif dan positif yang mana untuk menentukan jenis sentimen pada tweet apakah bersentimen negatif atau positif. Hasil akhir dari labeling dataset diperoleh 50,8% data sentimen positif dan 49,2% data sentimen negatif, dilakukan labeling data menjadi positif negatif adalah supaya mempermudah untuk melakukan proses pengujian data.
3. Proses *Pre-processing*
   Pada tahapan selanjutnya di lakukan *Pre-Processing* pada dataset yang telah di kolektif, *Pre-Processing* pada penelitian ini melalui beberapa tahapan sebagai berikut:
   1. *Case Folding*
      Pada tahapan *case folding* di lakukan perubahan huruf kapital menjadi huruf kecil semua atau yang di sebut *lower case*
   2. *Tokenizing*
      Pada tahapan *tokenizing* dilakukan untuk memotong kata dalam dataset yang akan di gunakan penelitian.
   3. *Stopword Removal*

 ISSN:2302-8149
e-ISSN:2540-9719

Tahapan dalam *stopword removal* adalah membuang atau menyaring kata kata yang sering muncul dalam dataset yang akan digunakan untuk pemrosesan dataset nanti nya

4. *Normalisasi*
   Tahapan dalam *normalisasi* adalah untuk menghapus karakter diluar huruf alfabet (a-z) dan juga termasuk tanda baca itu sendiri
5. *Stemming*
   Tahapan *Stemming* dilakukan yaitu merubah kata-kata menjadi sebuah kata dasar untuk digunakan melakukan penelitian

4. Pembobotan kata menggunakan *term frequency-invers document frequency (tf-idf)*
   Didalam penelitian ini peneliti menggunakan *term frequency-invers document frequency (tf-idf)* untuk melakukan perubahan pada teks menjadi nilai matrik(atau nilai vektor), Nilai dari *tf-idf* akan semakin bagus dengan dilakukannya berapa kali dari sebuah kata yang muncul didalam dokumen dan akan turun sebanding dengan banyaknya data yang digunakan. *TF-IDF* merupakan hasil dari *tf* dan *idf* atau yang diformulakan melalui persamaan(1) ,*tf (t, d)* didalam persamaan (2) , dan *idf (t, D)* didalam persamaan 3.
   $$tf - idf\ (t, d, D) = tf(t, d).idf(t, D) \quad (1)$$

   Yang dimana:

   $$tf(t, d) = log(1 + freq(t, d) \quad (2)$$

   $$idf(t, D) = log\left(\frac{N}{count(d \in D) : t \in d}\right) \quad (3)$$

   Tf(t,d) adalah sebuah frekuensi kata atau (*term*) yang akan menentukan seberapa banyak dari suatu kata yang muncul pada seluruh isi dokumen dan, idf(t,D) adalah *inverse document frequency* dari kata atau (t) dan N adalah jumlah dari dokumen yang ada didalam data dan df adalah dokumen yang mengandung kata atau term (t) [18]
5. Implementasi *Support Vector Machine*
   Support Vector Machine berfungsi untuk melakukan tugas fungsi pemisah yang terbaik di antara fungsi yang ada dalam melakukan pemisahan dua objek.

4 Hasil dan Pembahasan

Pada proses pengumpulan data dilakukan menggunakan *webharvy* diperoleh data sebanyak 8577 data yang diperoleh, data penelitian dikumpulkan pada bulan Januari 2020 sampai dengan Juli 2021. Jumlah data yang didapat kemudian dilakukan pembagian untuk data sebelumnya dilakukan *balancing* data sebanyak 3000 untuk data yang bermuatan positif dan 3000 data untuk bermuatan positif, setelah dilakukan *balancing* data kemudian data dibagi untuk 70% data training dan 30% untuk data testing, kemudian untuk melakukan klasifikasi menggunakan algoritme *support vector machine* dengan ditampilkan akurasi menggunakan *confusion matrix*.

4.1 Data

Data yang didapatkan melalui aplikasi *webharvy* yang didapat dari sosial media *Twitter* dengan keyword #covid19indonesia dan #coronaindonesia, dari data yang dicari didapat data sebanyak 8577 data, contoh data dapat dilihat pada tabel 1 berikut.

Tabel 1. Contoh data

Si covid-19 dari yg awalnya jauh di luar sana, sampe akhirnya dalam enam bulan ini banyak ring satu saya yg kena. Bahkan	Ulasan positif

 ISSN:2302-8149
e-ISSN:2540-9719

org yg taat protokol kesehatanpun mulai terpapar. Semoga semuanya selalu sehat dan pandemi ini segera berakhir #COVID19 #COVID19indonesia	
rumah sakit mana saja yg malprateknya tdk ada (atau minim) selama menghandling pasien corona di #indonesia ?	Ulasan negatif

4.2 Pre-Processing

Tahap pada *Preprocessing* digunakan untuk membersihkan data dari *noise* atau gangguan yang dapat menghambat proses klasifikasi untuk menjadi maksimal. Dalam penelitian ini *preprocessing* digunakan untuk mengubah data mentah menjadi data yang siap digunakan untuk pengujian, berikut beberapa tahapan *Preprocessing*

4.2.1 Case-Folding

Tahap selanjutnya adalah *case folding* pada tahapan ini dilakukan untuk mengubah huruf menjadi huruf kecil semua yang digunakan untuk mempermudah dalam dalam proses selanjutnya.

```
#------ Case Folding --------
# gunakan fungsi Series.str.lower() pada Pandas
TWEET_DATA['tweet'] = TWEET_DATA['tweet'].str.lower()


print('Case Folding Result : \n')
print(TWEET_DATA['tweet'].head(5))
print('\n\n\n')
```

Gambar 1. Kode *case-folding*

Hasil dari case-folding adalah mengubah teks yang terdapat huruf kapital di dalam menjadi huruf kecil untuk memudahkan dalam proses pengujian menggunakan dataset tersebut

Tabel 2. *Case-folding*

Sebelum Case Folding	Sesudah Case Folding
Kamu gak bosen kah begini terus ? Tolong cepatlah pergi, #COVID19 #Covid #COVID19indonesia	kamu gak bosen kah begini terus tolong cepatlah pergi
Heran ya sama org2 yg mengelu2kan angka kesembuhan COVID hari ini tinggi, tapi kalau angka infeksi barunya jauh lebih tinggi lagi, ya ga ada guna oi! #COVIDIOTS #COVID19 #covid19Indonesia	heran ya sama org yg mengelukan angka kesembuhan covid hari ini tinggi tapi kalau angka infeksi barunya jauh lebih tinggi lagi ya ga ada guna oi
Semenjak pernah terpapar covid-19 saya jadi berhenti merokok dan saya mulai menabung dan uangnya saya gunakan untuk keperluan swabb antigen dll.	semenjak pernah terpapar covid saya jadi berhenti merokok dan saya mulai menabung dan uangnya saya gunakan untuk keperluan swabb antigen dll keren bukan

Output dari hasil *case folding* adalah mengubah huruf kapital menjadi huruf kecil pada dataset.

 ISSN:2302-8149
e-ISSN:2540-9719

4.2.2 Tokenizing

Proses melakukan pemotongan kata ini didalam *Preprocessing* disebut dengan *tokenizing*. Tokenizing dengan kata lain bisa disebut untuk menghilangkan sebagian tanda baca seperti nomor,tanda kutip,simbol baca dan megubah kata kata ke huruf yang kecil

```
# ------ Tokenizing ---------

def remove_tweet_special(text):
    # remove tab, new line, ans back slice
    text = text.replace('\\t'," ").replace('\\n'," ").replace('\\u'," ").replace('\\',"")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")

TWEET_DATA['tweet'] = TWEET_DATA['tweet'].apply(remove_tweet_special)

#remove number
def remove_number(text):
    return  re.sub(r"\d+", "", text)

TWEET_DATA['tweet'] = TWEET_DATA['tweet'].apply(remove_number)

#remove punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans("","",string.punctuation))

TWEET_DATA['tweet'] = TWEET_DATA['tweet'].apply(remove_punctuation)

#remove whitespace leading & trailing
def remove_whitespace_LT(text):
    return text.strip()

TWEET_DATA['tweet'] = TWEET_DATA['tweet'].apply(remove_whitespace_LT)

#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+',' ',text)

TWEET_DATA['tweet'] = TWEET_DATA['tweet'].apply(remove_whitespace_multiple)
```

Gambar 2. Kode *Tokenizing*

Pada kode tokenizing diatas digunakan untuk menghapus tanda baca, emoji, link/url, hastag yang terdapat didalam dataset yang digunakan untuk mempermudah dalam Pemrosesan pengujian dataset yang akan digunakan. Tokenizing adalah sebuah proses untuk memecah kalimat ke dalam kata, dan kata yang sudah dihasilkan dari proses Tokenizing disebut Token, didalam kode tersebut terdapat fungsi seperti def remove_number(text) yang berguna untuk menghapus setiap nomor yang terdapat didalam dataset kemudian def remove_puctuation(text) untuk menghapus tanda baca yang terdapat dalam dataset selanjutnya def remove_whitespace(text) menghapus penggunaan dua spasi menjadi satu spasi dalam dataset,kemudian untuk baris kode def remove_singl_char(text) berfungsi untuk menghapus huruf yang terduplikat atau ganda yang ada terdapat didalam dataset, setelah melakukan Tokenizing makan dilakukanlah untuk melihat seberapa banyak kata yang muncul.

Tabel 3. *Tokenizing*

Sebelum Tokenizing	Sesudah Tokenizing
Jadi begini rasanya terkena COVID-19 dengan gejala-gejala yang ada. Bagaimana denganmu? #COVID19 #COVID19indonesia	jadi begini rasanya terkena covid dengan gejala gejala yang ada bagaimana denganmu covid covid indonesia
Tetap jaga jarak kalau gak mau TERSAKITI. #Covid_19 #COVID19 #COVID19indonesia	tetap jaga jarak kalau gak mau tersakiti covid covid covid indonesia

 ISSN:2302-8149
e-ISSN:2540-9719

Covid19 itu beneran apa bohongan sih? , Yg punya argumen soal cocid tolong di jelasin donk :) #COVID19 #COVID19indonesia #AstraZeneca #COVIDIOTS #COVISHIELD #Covid_19	covid itu beneran apa bohongan sih yg punya argumen soal cocid tolong di jelasin donk covid covid indonesia astrazeneca covidiots covishield covid
Apa hasil #ppkm atau #psbb selama ini? Koq bisa rekor baru malah terukir di tengah penggiatan PPKM? #inisihmakinparah #COVID19 #COVID19indonesia	apa hasil ppkm atau psbb selama ini koq bisa rekor baru malah terukir di tengah penggiatan ppkm inisihmakinparah covid covid indonesia

4.2.1 Stopword removal

```
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')
# ---------------------- get stopword from NLTK stopword -----------------------
# get stopword indonesia
list_stopwords = stopwords.words('indonesian')
print(len(list_stopwords))

# ---------------------------- manualy add stopword  ---------------------------
# append additional stopword
list_stopwords.extend(["yg", "dg", "rt", "dgn", "ny", "d", 'klo',
                       'kalo', 'amp', 'biar', 'bikin', 'bilang',
                       'gak', 'ga', 'krn', 'nya', 'nih', 'sih',
                       'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
                       'jd', 'jgn', 'sdh', 'aja', 'n', 't',
                       'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',
                       '&amp', 'yah'])
len(list_stopwords)
# ----------------------- add stopword from txt file ------------------------
# read txt stopword using pandas
txt_stopword = pd.read_csv("stopwordbahasa.csv", names= ["stopwords"], header = None)

# convert stopword string to list & append additional stopword
list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
len(list_stopwords)
# ---------------------------------------------------------------------------

# convert list to dictionary
list_stopwords = set(list_stopwords)


#remove stopword pada list token
def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

TWEET_DATA['tweet_tokens_WSW'] = TWEET_DATA['tweet_tokens'].apply(stopwords_removal)


print(TWEET_DATA['tweet_tokens_WSW'].head())
```

Gambar 3. Kode *stopword-removal*

Tahapan selanjutnya adalah melakukan *stopword removal* yaitu melakukan penyaringan terhadap kata-kata yang sering muncul atau yang jarang muncul, pada tahapan ini dari beberapa kata yang muncul tidak akan digunakan karena untuk membuat sistem yang optimal.

Tahapan selanjutnya adalah melakukan *stopword removal* yaitu melakukan penyaringan terhadap kata-kata yang sering muncul atau yang jarang muncul, pada tahapan ini dari beberapa kata yang muncul tidak akan digunakan karena untuk membuat sistem yang optimal. Stopword dijelaskan bahwa untuk menyaring kata-kata yang muncul dalam sebuah dataset, didalam codingan ini dijelaskan bahwa stopword menggunakan bahasa Indonesia, kemudian menambahkan kata stopword yang akan digunakan seperti “yg, dg, rt, dgn, ny,d ll” setelah itu menambahkan stopword yang sudah ada didalam text file yang berbentuk csv, text file ini diperoleh dari kata-kata yang sering digunakan dalam sosial media twitter kemudian setelah melakukan penambahan stopword dari file csv tadi kita lakukan merubah stopword yang berbentuk string menjadi list dan kemudian di konversikan atau rubah menjadi sebuah kamus, setelah melakukan konversi tadi maka dilakukan menghilangkan stopword atau kata kata yang sering muncul didalam token atau kata-kata yang sudah di tokenizing tadi kemudian ditampilkan hasil dari stopword yang sudah dilakukan tadi, bisa dilihat hasil dari stopword yang dilakukan oleh kode diatas seperti hasil dibawah ini.

Tabel 4. *Stopword-removal*

 ISSN:2302-8149
e-ISSN:2540-9719

Sebelum Stopword	Sesudah Stopword
berita tahun penuh dengan terpapar penyakit tahun kenaikan kasus terpapar virus melonjak angka kematian terus naik tahun tahun tahun	['berita', 'penuh', 'terpapar', 'penyakit', 'kenaikan', 'terpapar', 'virus', 'melonjak', 'angka', 'kematian']
kalo ppkm diperpanjang apakah cicilan kpr kkb kartu kredit itu boleh ditunda tanpa denda bunga dan surat peringatan ya ppkm darurat lanjut lalu rakyatnya mau makan apa tdk semua orang tidak bekerja mendptkan penghasilan	['ppkm', 'diperpanjang', 'cicilan', 'kpr', 'kkb', 'kartu', 'kredit', 'ditunda', 'denda', 'bunga', 'surat', 'peringatan', 'ppkm', 'darurat', 'rakyatnya', 'makan', 'mendptkan', 'penghasilan']
heran ya sama org yg mengelukan angka kesembuhan covid hari ini tinggi tapi kalau angka infeksi barunya jauh lebih tinggi lagi ya ga ada guna oi	['heran', 'mengelukan', 'angka', 'kesembuhan', 'covid', 'angka', 'infeksi', 'barunya']

4.2.2 Stemming

Tahapan yang dilakukan selanjutnya adalah *Stemming* pada tahapan ini dilakukan untuk mengubah kalimat menjadi kata dasar suatu kalimat. Proses *stemming* pada dokumen Bahasa Indonesia cukup kompleks, karena harus dilakukan penghilangan seluruh imbuhan pada kata-kata yang terdapat pada *tweets*. Library yang di gunakan merupakan library sastrawi yang digunakan untuk melakukan *Stemming* dalam bahasa Indonesia

```
# import Sastrawi package
!pip install Sastrawi
!pip install swifter
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in TWEET_DATA['tweet_normalized']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("------------------------")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":" ,term_dict[term])

print(term_dict)
print("------------------------")


# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

TWEET_DATA['tweet_tokens_stemmed'] = TWEET_DATA['tweet_normalized'].swifter.apply(get_stemmed_term)
print(TWEET_DATA['tweet_tokens_stemmed'])
```

Gambar 4. Kode *stemming*

Stemming pada kode diatas menggunakan library *Sastrawi* yang dimana library *sastrawi* adalah library yang digunakan untuk melakukan stemming berbahasa Indonesia, untuk melihat hasil dari stemming ditampilkan pada tabel berikut:

Tabel 5. *Stemming*

Sebelum proses stemming	Sesudah proses stemming

ISSN:2302-8149
e-ISSN:2540-9719

waspada covid melandai waspada tukang jualan vaksin antigen dan test pcr masih berusaha main drama lagi	['waspada', 'covid', 'landai', 'waspada', 'tukang', 'jual', 'vaksin', 'antigen', 'test', 'pcr', 'usaha', 'main', 'drama']
akhir ini saya lihat pemerintah pusat dgn tenaga dlm lebih memilih tangani papua operasi militer daripada melawan	lihat perintah pusat dgn tenaga dlm pilih tangan papua operasi militer lawan covid covid indonesia
liat status di sosmed hampir setiap hari isinya kabar duka setiap harinya ada aja yg wafat dari orang yg kenal sampe gak kenal bahkan karena corona atau bukan kayaknya kematian sangat dekat banget kayak tinggal tunggu giliran aja gitu	['lihat', 'status', 'sosmed', 'isi', 'kabar', 'duka', 'hari', 'wafat', 'kenal', 'sampai', 'kenal', 'corona', 'kayak', 'mati', 'banget', 'tinggal', 'tunggu', 'gilir']
semoga keadaan bumi ini lekas membaik	['moga', 'bumi', 'lekas', 'baik']

 ISSN:2302-8149
e-ISSN:2540-9719

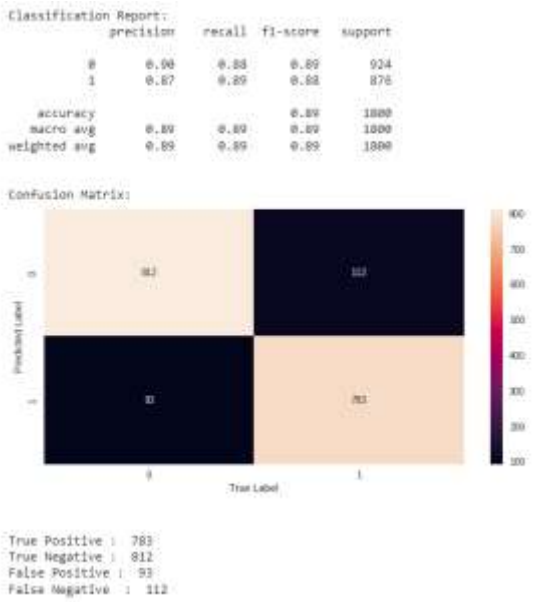
4.3 Analisis sentimen menggunakan *Support vector machine* untuk kernel *linear*

Dalam tahap ini, untuk melakukan pengujian model *Support vector machine* kernel *linear* menggunakan bahasa pemrograman *python* dan dibuat pada google colab, untuk mengetahui hasil dari klasifikasi menggunakan algoritme *Support vector machine* yang menggunakan data sebanyak 70% untuk data training dan 30% untuk data testing ditampilkan dibawah.

```
[ ]  1 #coding svm linear
     2 linear = SVC(kernel='linear',probability=True)
     3 linear.fit(data_train_x,y_train)
     4 linear1=linear.predict(data_test_x)
     5 #buat history time
     6 classifier_linear = SVC(verbose=1)
     7 time_0 = time.time()
     8 history = linear.fit(data_train_x, y_train)
     9 time_1 = time.time()
    10 prediction_linear = linear.predict(data_train_x)
    11 time_2 = time.time()
    12 time_linear_train = time_1-time_0
    13 time_linear_predict = time_2-time_1
    14 y_train_hat_linear=linear.predict(data_train_x)
    15 y_test_hat_linear=linear.predict(data_test_x)
    16 #tampilkan hasil test dan training score
    17 y_train_hat_linear=linear.predict(data_train_x)
    18 y_test_hat_linear=linear.predict(data_test_x)
    19 print ( "train accuracy= "  ,np.mean(y_train_hat_linear == y_train)*100)
    20 print ("test accuracy= "  ,np.mean(y_test_hat_linear == y_test)*100)
    21 #tampilkan hasil dari metode svm kernel linear
    22 linear.score(data_test_x, y_test)
```

Gambar 5. kode untuk *Support vector machine* kernel *linear*

dari hasil proses kode diatas dihasilkan untuk tingkat akurasi menggunakan algoritme *Support vector machine* dengan data training 70% dan data test 30% dihasilkan akurasi sebesar 88,61% dan untuk melihat hasil dari *confusion matrix.*



Gambar 6. Hasil dari *confusion matrix*

Untuk mengetahui hasil dari *confusion matrix* dilakukan perhitungan manual untuk mengetahui dari *accuracy,precision,recall,f1-score* dilakukan perhitungan dibawah.

 ISSN:2302-8149
e-ISSN:2540-9719

$$Akurasi = \frac{783 + 812}{783 + 812 + 93 + 112} X\ 100\ =\ 0.8861\ =\ 88{,}61\%$$

$$Presisi = \frac{783}{783+93}\ = 89\%$$

$$Recall = \frac{783}{783 + 112}\ =\ 87\%$$

$$f1 - score = \frac{2x0{,}89\ X\ 0{,}87}{0{,}89\ +\ 0{,}87}\ =\ 89\%$$

Berdasarkan perhitungan manual di atas nilai akurasi adalah 88,61 % sedangkan untuk presisi/*precision* sebesar 89% dan *recall* 87% dan juga F1 *score* sebesar 89% yang mana untuk *Support vector machine* kernel Linear ini lumayan cukup baik dalam melakukan prediksi sentimen kalimat yang terdapat dalam dataset.

4.4 Analisis sentimen menggunakan *Support vector machine* kernel *radial basis function*

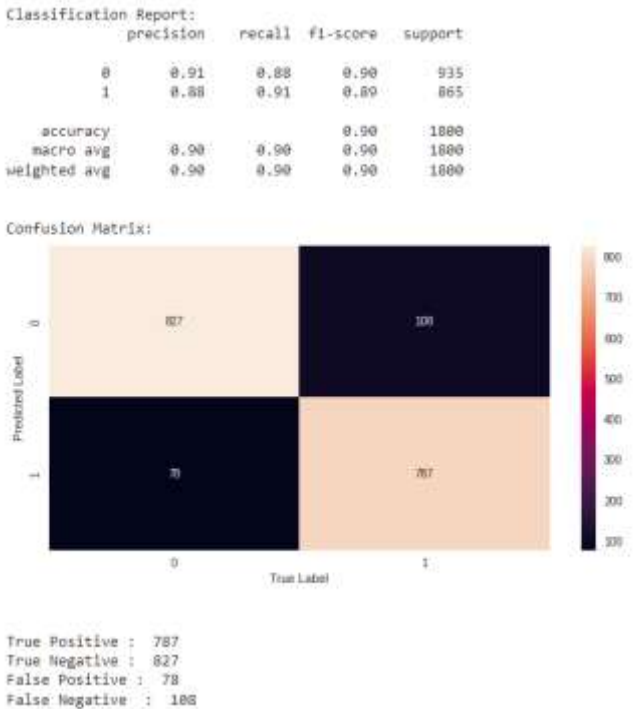
Untuk pengujian kernel *radial basis function* hampir sama dengan kernel *linear*, pada pengujian menggunakan *radial basis function* menggunakan data training 70% dan data test 30%, untuk hasil dari *radial basis function* dilihat pada kode pada gambar 3.

```
# Process of making models Klasifikasi SVM RBF
rbf = SVC(kernel='rbf')
rbf.fit(data_train_x, y_train)
rbf1 = rbf.predict(data_test_x)
#history time
classifier_rbf = SVC(verbose=1)
time_rbf0 = time.time()
history = rbf.fit(data_train_x, y_train)
time_rbf1 = time.time()
prediction_rbf = rbf.predict(data_train_x)
time_rbf2 = time.time()
time_rbf_train = time_rbf1-time_rbf0
time_rbf_predict = time_rbf2-time_rbf1
y_train_hat_rbf = rbf.predict(data_train_x)
y_test_hat_rbf = rbf.predict(data_test_x)
#hasil tampil data test dan training
y_train_hat_rbf = rbf.predict(data_train_x)
y_test_hat_rbf = rbf.predict(data_test_x)
print ( "train accuracy= "  ,np.mean(y_train_hat_rbf == y_train)*100)
print ("test accuracy= "  ,np.mean(y_test_hat_rbf == y_test)*100)
#tampilkan hasil dari metode svm kernel rbf
rbf.score(data_test_x, y_test)
```

Gambar 7. Kode untuk s*upport vector machine* kernel *radial basis*

Hasil dari pengujian menggunakan kode diatas untuk menentukan hasil dari klasifikasi pada algoritme *Support vector machine* kernel *radial basis function* menunjukkan tingkat akurasi sebesar 89,66% dan untuk melihat hasil dari *confusion matrix* untuk kernel *radial basis function* dapat dilihat pada gambar 4.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 8. *Confusion matrix* untuk kernel *radial basis function*

Untuk mengetahui hasil dari *confusion matrix* dilakukan perhitungan manual untuk mengetahui dari *accuracy,precision,recall,f1-score* dilakukan perhitungan dibawah.

$$Akurasi = \frac{787 + 827}{787 + 827 + 78 + 108} X\ 100\ =\ 0{,}8966\ =\ 89\%$$

$$Presisi = \frac{787}{787 + 78}\ = 90\%$$

$$Recall = \frac{787}{787 + 108}\ =\ 88\%$$

$$f1 - score = \frac{2\ X\ 0{,}90\ X\ 0{,}88}{0{,}90\ +\ 0{,}88}\ =\ 89\%$$

Berdasarkan perhitungan manual di atas nilai akurasi adalah 89,66 % sedangkan untuk presisi/*precision* sebesar 90% dan *recall* 88% dan juga F1 *score* sebesar 89% yang mana untuk *Support vector machine* kernel *radial basis function* ini hampir sama dengan kernel Linear dalam melakukan prediksi sentimen kalimat yang terdapat dalam dataset.

 ISSN:2302-8149
e-ISSN:2540-9719

4.5 Analisis sentimen menggunakan *Support vector machine* kernel *Polynomial*

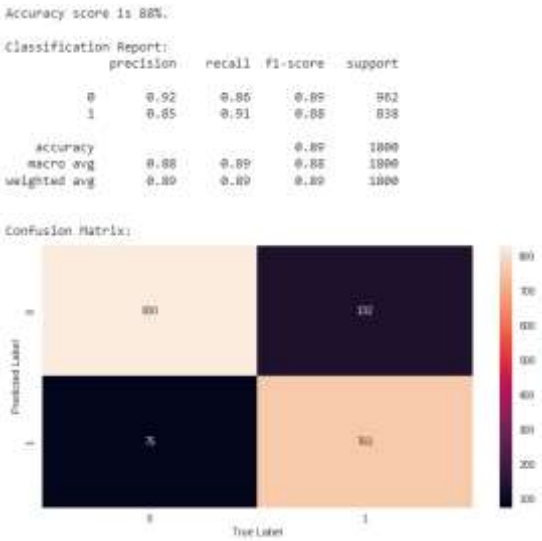
Pada pengujian menggunakan kernel Polynomial tidak berbeda jauh dengan pengujian kernel sebelumnya yaitu menggunakan kode yang hampir sama tetapi pada kernel Polynomial ini menggunakan derajat/degree untuk mengetahui akurasi nya, dan untuk mengetahui nilai akurasi dari kernel Polynomial dan untuk mengetahui nilai akurasi dari kernel Polynomial dapat dilihat pada gambar yang ada dibawah ini:

```
poly = SVC(kernel='poly',)
poly.fit(data_train_x, y_train)
poly1 = poly.predict(data_test_x)
#history time
classifier_poly = SVC(verbose=1)
time_poly0 = time.time()
history = poly.fit(data_train_x, y_train)
time_poly1 = time.time()
prediction_poly = poly.predict(data_train_x)
time_poly = time.time()
time_poly_train = time_poly-time_poly
time_poly_predict = time_poly-time_poly
y_train_hat_poly = poly.predict(data_train_x)
y_test_hat_poly = poly.predict(data_test_x)
#hasil tampil data test dan training
y_train_hat_poly = poly.predict(data_train_x)
y_test_hat_poly = poly.predict(data_test_x)
print ( "train accuracy= "  ,np.mean(y_train_hat_poly == y_train)*100)
print ("test accuracy= "  ,np.mean(y_test_hat_poly == y_test)*100)
#tampilkan hasil dari metode svm kernel rbf
poly.score(data_test_x, y_test)
```

```
train accuracy=  99.92857142857143
test accuracy=  88.5
0.885
```

Gambar 9. *Kode* untuk kernel *Polynomial*

Pada hasil dari *confusion matrix* hasil dari akurasi model yang dihasilkan dalam melakukan prediksi sebesar 88% untuk metode *Support vector machine* kernel *radial basis function* dan untuk classification report dapat dilihat nilai *precision*, *recall*, dan F1 *score* untuk pelabelan 0 dan 1 menghasilkan nilai *precision* 92% untuk label 0 dan *precision 85%* untuk label 1, nilai *recall* menghasilkan 86% untuk label 0 dan *recall* menghasilkan 91% untuk label 1, F1 memiliki nilai 89% untuk label 0 dan 88% untuk label 1, selanjutnya output matrix untuk *confusion matrix* untuk label 0 sebanyak 830 kalimat dari seharusnya 962 kalimat sedangkan untuk label 1 sebanyak 763 kalimat dari yang seharus nya 838 kalimat.



Gambar 10. *Confusion matrix* untuk kernel *Polynomial*

 ISSN:2302-8149
e-ISSN:2540-9719

Kemudian untuk melakukan hasil perhitungan manual untuk hasil akurasi dari kernel *Polynomial* menggunakan rumus atau metode dibawah.

$$Akurasi = \frac{763 + 830}{763 + 830 + 75 + 132} X\ 100\ =\ 0{,}88{,}55\ =\ 88\%$$

$$Presisi = \frac{763}{763 + 75}\ = 91\%$$

$$Recall = \frac{763}{763 + 132}\ =\ 85\%$$

$$f1 - score = \frac{2\ X\ 0{,}91\ X\ 0{,}85}{0{,}91\ +\ 0{,}85}\ =\ 87\%$$

Berdasarkan perhitungan manual di atas nilai akurasi adalah 88 % sedangkan untuk presisi/*precision* sebesar 91% dan *recall* 85% dan juga F1 *score* sebesar 87% yang mana untuk *Support vector machine* kernel *radial basis function* ini hampir sama dengan kernel Polynomial dalam melakukan prediksi sentimen kalimat yang terdapat dalam dataset. Kemudian kita melakukan perhitungan akurasi dan menampilkan hasil visual untuk metode *Support vector machine* kernel *Polynomial.*

5 Kesimpulan

Pada penelitian terhadap sentimen analisis mengenai Covid-19 Indonesia pada sosial media Twitter dengan menggunakan *Support Vector Machine* yang menggunakan tiga (3) kernel yaitu Linear, *Radial Basis Function, Polynomial* menghasilkan tingkat akurasi untuk kernel Linear sebesar 89% kemudian untuk kernel *Radial Basis Function* sebesar 90% dan untuk kernel terakhir yaitu *Polynomial* sebesar 88%. Dari tingkat akurasi yang didapat setelah melalui proses *Pre-Processing* yang dilalui oleh dataset. Dari tingkat akurasi yang didapat disimpulkan bahwa untuk metode *Support Vector Machine* menghasilkan akurasi terbaik pada kernel *Radial Basis Function* memiliki akurasi yang baik dalam melakukan Sentimen Analisis, sehingga dapat dikatakan mampu memberikan prediksi sentimen pada teks dengan baik dan akurat.

Referensi

[1] Samsir, Ambiyar, U. Verawardina, F. Edi, and R. Watrianthos, "Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 menggunakan Metode Naïve Bayes," *Jurnal Media Informatika Budidarma*, vol. 5, no. 1, pp. 157–163, Jan. 2021, doi: 10.30865/mib.v5i1.2604.

[2] F. Rahutomo, Y. Saputra, and A. Fidyawan, "Implementasi Twitter Sentiment Analisis untuk Review Film menggunakan Algoritma *Support Vector Machine*," *Jurnal Informatika Polinema*, vol. 4, no. 2, pp. 93–99, 2018.

[3] A. Raudya Wibowo, N. Nidya, A. Firdausi Rahma, and Agussalim, "Analisis Sentimen Hastag 'Dirumahaja' Saat Pandemi Covid-19 Di Indonesia menggunakan NLP," 2020.

[4] A. M. Pravina, I. Cholissodin, and P. P. Adikara, "Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter menggunakan *Algoritme Support Vector Machine* (SVM)," 2019. [Online]. Available: http://j-ptiik.ub.ac.id

[5] D. Agustono, D. Sianturi, A. Taufik, and W. Gata, "Analisis Sentimen Terhadap Warga China Saat Pandemi Dengan Algoritma *Term Frequency-Inverse Document Frequency* dan *Support Vector Machine*," 2020. [Online]. Available: http://e-journal.stmiklombok.ac.id/index.php/jire

[6] A. Utami, "Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Algoritma *Support Vector Machine* dan *K-Nearest Neighbor* Berbasis *Particle Swarm Optimization*," *Jurnal Pilar Nusa Mandiri*, vol. 13, no. 1, 2017, [Online]. Available: www.tribunnews.com

[7] I. Santoso, W. Gata, and Paryanti Budi Atik, "Penggunaan *Feature Selection* di Algoritma *Support Vector Machine* untuk Sentimen Analisis Komisi Pemilihan Umum," *Jurnal Resti*, vol. 3, no. 3, pp. 364–370, 2017.

 ISSN:2302-8149
e-ISSN:2540-9719

[8] Erlin, J. Sianturi, A. Hajjah, and Agustin, "Analisis Sentimen Prosesor AMD Ryzen menggunakan Metode *Support Vector Machine*," *SATIN-Sains dan Teknologi Informasi*, vol. 7, no. 2, pp. 129–141, 2021, doi: 10.33372/stn.v7i2.804.

[9] S. Azza Amira, S. Utama, and dan Muhammad Hanif Fahmi, "Penerapan Metode *Support Vector Machine* untuk Analisis Sentimen pada Review Pelanggan Hotel," *Edu Komputika*, vol. 7, no. 2, 2020, [Online]. Available: http://journal.unnes.ac.id/sju/index.php/edukom

[10] A. Handayanto, K. Latifa, N. D. Saputro, and R. R. Waliyansyah, "Analisis dan Penerapan Algoritma *Support Vector Machine* (SVM) dalam Data Mining untuk Menunjang Strategi Promosi," 2019.

[11] N. Made Gita Dwi Purnamasari, M. Ali Fauzi, and L. Shinta Dewi, "Identifikasi *Tweet Cyberbullying* pada Aplikasi Twitter menggunakan Metode *Support Vector Machine* (SVM) dan Information Gain (IG) sebagai Seleksi Fitur," 2018. [Online]. Available: http://j-ptiik.ub.ac.id

[12] R. Pebrianto, T. Rivanie, R. Nurfalah, W. Gata, M. Fahmi Julianto, and S. Nusa Mandiri, "Adopsi *Algorithm Support Vector Machine* untuk Analisis Sentimen Larangan Mudik Lebaran 2020 pada Twitter," vol. VI, no. 2, 2020, doi: 10.31294/jtk.v4i2.

[13] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, "Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter dengan Metode *Support Vector Machine*," *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, vol. 6, no. 3, 2021, [Online]. Available: http://situs.com

[14] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia menggunakan Algoritma *Support Vector Machine* (SVM)," vol. 8, no. 1, pp. 147–156, 2021, doi: 10.25126/jtiik.202183944.

[15] D. S. Utami and A. Erfina, "Analisis Sentimen Pinjaman Online Di Twitter menggunakan Algoritma *Support Vector Machine* (SVM)," 2021.

[16] N. Dwi Putranti and E. Winarko, "Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan *Maximum Entropy dan Support Vector Machine*," *IJCCS*, vol. 8, no. 1, pp. 91–100, 2014.

[17] B. Pamungkas, E. Purbaya, and D. Januarita, "Analisis Sentimen Twitter menggunakan Metode *Support Vector Machine* (SVM) pada Kasus Benih Lobster 2020," *J.OF INISTA*, vol. 3, pp. 10–20, 2021.