Analisis Sentimen Tanggapan Masyarakat Terhadap Vaksin Covid-19 Menggunakan Algoritma Support Vector Machine (SVM)

by Layla Qodary Zalyhaty

Submission date: 15-Aug-2021 07:51PM (UTC+0700)
Submission ID: 1631566980
File name: Jurnal_17410100190.docx (1.51M)
Word count: 3361
Character count: 21630

JSIKA Vol. 11, No. 1. Tahun 2021 | ISSN 2338-137X

Analisis Sentimen Tanggapan Masyarakat Terhadap Vaksin Covid-19 Menggunakan Algoritma *Support Vector Machine* (SVM)

Layla Qodary Zalyhaty [1)] **Vivine Nurcahyawati** [2)] **Erwin Sutomo** [3)]
Program Studi/Jurusan Sistem Informasi
Universitas Dinamika
Jl. Raya Kedung Baruk 98 Surabaya, 60298
Email : 1)17410100190@dinamika.ac.id, 2)vivine@dinamika.ac.id , 3)sutomo@dinamika.ac.id

Abstract: Covid-19 vaccine is one of the government's efforts to overcome the pandemic. But there are pros and cons in the community. Some support vaccines, some doubt, some even reject it. The study took the responses and public opinion of 283 data from 7 online news. The data of public responses from online news is still unstructured. To solve the problem, this study will conduct sentiment analysis by classifying community responses into positive and negative sentiments using the Support Vector Machine algorithm in order to help the government to know the responses or concerns of the public about COVID-19 vaccine, as well as evaluation materials to determine the next strategy related to education and socialization about the COVID-19 vaccine. The stages are starting from collecting data from online news, manually labelling data, then the pre-processing text, TF-IDF weighting, SVM classification model creation, classification model testing, validation, evaluation and visualization. Out of a total of 283 data on community responses to the COVID-19 vaccine with a ratio of 90:10 with 254 training data and 29 testing data, the result is 66,8% positive sentiment and 33,2% negative sentiment, an average cross validation score is 74.05%, and accuracy score is 82.76%.

Keywords: *Sentiment Analysis, COVID-19 Vaccine, Online News, Support Vector Machine.*

Upaya pemerintah untuk mengatasi pandemi adalah dengan pengadaan vaksin COVID-19. Vaksin terdiri dari agen yang menyerupai virus atau bakteri yang sudah mati atau dilemahkan. Vaksin ini merangsang sistem imun di dalam tubuh untuk mengenalinya sebagai agen asing, menghancurkannya. Misalkan akhirnya tetap sakit, maka sakitnya tidak akan terlalu berat (Febriyani, 2021).

Tujuan diberikannya vaksin adalah untuk mengurangi penularan virus corona, menurunkan angka kematian dan mencapai kekebalan kelompok, supaya masyarakat bisa kembali produktif. Namun ditengah kemunculan vaksin COVID-19, terdapat pro dan kontra di masyarakat. Ada yang mendukung vaksin, ada yang meragukan keampuhan vaksin, dan ada juga yang menolak meskipun vaksin diberikan secara gratis (Putri, 2020).

Masyarakat yang telah maupun yang belum divaksin memberikan respon dan opininya di berbagai media, salah satunya adalah media berita online. Penelitian ini menggunakan 283 data tanggapan masyarakat dari 7 media berita online yaitu Kompas.com, Liputan6.com, Cnn.com, Cnbc.com, Bbc News, Tempo.co, dan IDN Times.

Pemerintah perlu mempertimbangkan berbagai masukan supaya vaksinasi dapat berjalan maksimal, diantaranya adalah dengan melihat bagaimana respon dan opini masyarakat terhadap wacana vaksinasi. Karena bentuk data tanggapan belum terstruktur, opini masyarakat tentang vaksin COVID-19 pada media berita online perlu dikaji dalam pemrosesan teks. Analisis sentimen digunakan untuk mengklasifikasikan opini masyarakat menjadi kelas positif atau kelas negatif. Hasil klasifikasi tersebut bisa digunakan untuk membantu pemerintah untuk mengetahui tanggapan ataupun kekhawatiran masyarakat terhadap vaksin COVID-19, sehingga pemerintah bisa melakukan evaluasi dan menentukan strategi selanjutnya terkait edukasi maupun sosialisasi tentang vaksin COVID-19 kepada masyarakat.

Diperlukan algoritma untuk menunjang klasifikasi sentimen dan pada penelitian ini menggunakan algoritma *Support Vector Machine* (SVM) untuk mengklasifikasikan respon & opini masyarakat Indonesia terhadap vaksin COVID-19 kedalam kelas positif &

negatif. Data yang diambil dari media berita online dipisahkan menjadi data *training* (untuk membuat model klasifikasi dengan algoritma SVM) dan data *testing* (untuk menguji model klasifikasi yang telah dibuat)

Berdasarkan penelitian-penelitian analisis sentimen sebelumnya, algoritma SVM menghasilkan akurasi yang cukup tinggi dalam melakukan klasifikasi sentimen. Penelitian yang dilakukan oleh Nurrun Muchammad Shiddieqy Hadna, Paulus Insap Santosa, dan Wing Wahyu Winarno ditemukan bahwa algoritma SVM yang dikolaborasikan dengan ekstraksi fitur TF-IDF menghasilkan akurasi sebesar 81.5%, lebih unggul daripada algoritma *Naïve Bayes* yang menghasilkan akurasi 80.8% dalam Studi Literatur Tentang Perbandingan Metode Untuk Proses Analisis Sentimen di Twitter.

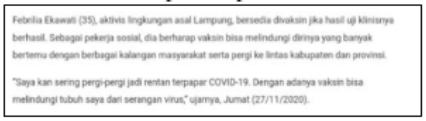
METODE

Untuk melakukan analisis sentimen tanggapan masyarakat terhadap vaksin COVID-19 menggunakan algoritma *Support Vector Machine* pada media berita online akan melalui Tahap Awal dan Analisis Data.

Tahap Awal

Gambar 1 Tahap Awal

Tahap Awal dimulai dengan Studi Literatur yaitu penulis mengulas penelitian terdahulu seperti jurnal dan tugas akhir yang berkaitan dengan studi kasus yang diambil, setelah itu dilanjutkan dengan Pengumpulan Data secara manual yang diambil dari media berita online yang memuat tanggapan masyarakat. Pengumpulan Data, dibagi menjadi tiga tahap yaitu Observasi, Studi Dokumentasi dan Teknik Pengolahan Data. Hasil studi dokumentasi ditampilkan pada Gambar 2.

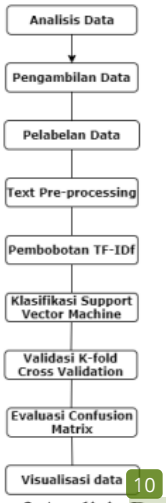

Febrilia Ekawati (35), aktivis lingkungan asal Lampung, bersedia divaksin jika hasil uji klinisnya berhasil. Sebagai pekerja sosial, dia berharap vaksin bisa melindungi dirinya yang banyak bertemu dengan berbagai kalangan masyarakat serta pergi ke lintas kabupaten dan provinsi.

"Saya kan sering pergi-pergi jadi rentan terpapar COVID-19. Dengan adanya vaksin bisa melindungi tubuh saya dari serangan virus," ujarnya, Jumat (27/11/2020).

Gambar 2 Hasil Studi Dokumentasi

Pengolahan data awalnya dilakukan pada *microsoft excel* mulai dari penyimpanan data hingga pelabelan manual. Setelah data memiliki label baru diproses didalam *python* dengan menggunakan media *google collaboratory*.

Analisis Data



Gambar 3 Analisis Data

Pada tahap Analisis Data, semua pengolahan dilakukan dengan bahasa pemrograman *Python* dengan media *google collaboratory*. Tahapan dalam Analisis Data terdiri dari Pengambilan Data, Pelabelan Data, *Text Pre-processing*, Pembobotan TF-IDF, Klasifikasi SVM, Validasi K-*fold Cross Validation*, Evaluasi *Confusion Matrix* dan Visualisasi Data.

Pengambilan Data

Pengambilan data dilakukan secara manual dengan cara mencari berita menggunakan kata kunci "tanggapan masyarakat vaksin covid" pada media berita online. Data tersebut kemudian disimpan kedalam *file Microsoft excel* dalam format .csv.

Pelabelan Data

Pelabelan data dilakukan pada keseluruhan data di *Microsoft excel*. Pelabelan data tanggapan masyarakat dilakukan dengan cara mencocokkan kata dari masing-masing tanggapan dengan *Load Dictionary* yang berisi daftar kata positif dan kata negatif. Dalam satu kalimat, jika suatu kata terdapat pada kamus maka akan diberi nilai 1 dan jika tidak maka akan diberi nilai 0. Setelah pencocokan kata dengan kamus, nilai dari masing-masing kolom

kata positif dan kata negatif dijumlahkan. Jika lebih banyak kata negatif dibanding positif maka tanggapan tersebut dikategorikan sebagai tanggapan negatif, begitu juga sebaliknya. Jika dalam satu kalimat jumlah kata positif dan kata negatifnya seimbang, maka tanggapan tersebut akan dimasukkan kedalam kelas positif karena pada penelitian ini tidak terdapat kelas netral.

Text Pre-processing

Text *Pre-processing* gunanya untuk membersihkan data, menghilangkan dan mengatasi data yang kotor, mengatasi informasi yang hilang, menghapus kata yang tidak digunakan dan mengubahnya kata menjadi kata dasar (Wardani, 2019). *Text pre-processing* terdiri dari *cleansing, case folding, parsing/tokenizing, filtering* dan *stemming*. Langkah-langkah yang harus dilakukan dalam pengolahan teks adalah (Pravina, Cholissodin, & Adikara, 2019) :

1. *Cleansing*

   Proses penghapusan tanda baca seperti .,"!?#@%(), menghapuskan URL atau *link*, dan menghapuskan angka (Pravina, Cholissodin, & Adikara, 2019).
2. *Case folding*

   Proses *case folding* berguna untuk menyeragamkan seluruh huruf menjadi huruf kecil/*lowercase* (Pravina, Cholissodin, & Adikara, 2019).
3. *Tokenizing*

   Proses *tokenizing* berguna untuk membagi teks yang awalnya berupa kalimat menjadi token atau *term* (Ulfah & Anam, 2020).
4. *Stopword Removal* atau *Filtering*

   Proses ini berguna untuk menghapus kata-kata yang tidak memiliki makna seperti *stoplist/stopword*. (Safitri, 2020).
5. *Stemming*

   Proses mengubah kata dalam dokumen menjadi kata dasar (*root word*), menghapus prefiks (imbuhan awalan) dan sufiks (imbuhan akhiran) (Pravina, Cholissodin, & Adikara, 2019).

Pembobotan TF-IDF

Pembobotan *Term Frequency-Inverse Document Frequency* TF-IDF yang gunanya untuk mengubah data teks menjadi data numerik supaya bisa dilakukan perhitungan dan juga untuk menghitung bobot setiap kata, semakin besar bobot suatu kata maka kata tersebut dianggap penting. Nilai TF-IDF dapat ditemukan menggunakan rumus berikut (Pravina, Cholissodin, & Adikara, 2019):

1. Menentukan nilai TF bisa menggunakan TF biner. Jika suatu kata atau term terdapat dalam sebuah dokumen maka diberi nilai 1 (satu), jika tidak ada maka diberi nilai 0 (nol).
2. Menghitung nilai *Inverse Document Frequency* (IDF) untuk melihat seberapa penting kata dalam sebuah dokumen. IDF dirumuskan sebagai berikut:

$$IDF = log \frac{D}{df} \quad (1)$$

3. Setelah itu menghitung TF IDF dengan menggabungkan perhitungan TF dengan IDF sebagai berikut:

$$W_{ij} = tf \text{ x } IDF \quad (2)$$

Keterangan:
Wij : bobot kata dalam setiap dokumen
Tf : jumlah kemunculan kata dalam dokumen
D : jumlah total seluruh dokumen

Klasifikasi Support Vector Machine

Algoritma *Support Vector Machine* termasuk kedalam pembelajaran mesin (*supervised learning*) yang bisa memprediksi atau mengklasifikasi kelas berdasarkan model prediksi yang dibuat dari data *training* untuk mendapatkan pola yang nantinya bisa digunakan untuk memprediksi pada proses pelabelan (Pravina, Cholissodin, & Adikara, 2019).

Tahapan algoritma SVM adalah:

1. Meminimalkan nilai margin dengan rumus:

$$W_i x_i + b = 0 \quad (3)$$

2. Jika nilai *wi* berada di kelas +1 maka masuk kelas positif

$$w_i.x_i + b \geq +1 \quad (4)$$

3. Jika nilai *wi* berada di kelas -1 maka masuk kelas negatif

$$w_i.x_i + b \geq -1 \quad (5)$$

4. Kernel yang umum digunakan adalah kernel Linear dengan rumus:

$$K(x,y) = x.y \quad (6)$$

Keterangan:
$i = 1,2,3, \ldots$
x = data ke-i
w = bobot

Validasi K-fold Cross Validation

Uji validasi dengan *k-fold cross validation* caranya adalah dengan mengulangi sampel data

uji beberapa kali dengan sampel yang berbeda-beda pada setiap pengujian. Prosedur ini merupakan cara umum untuk menguji validasi model klasifikasi (Wardani, 2019).

Evaluasi Confusion Matrix

Dalam evaluasi klasifikasi, ada empat kemungkinan hasil klasifikasi data. Jika data berlabel positif dan diprediksi positif masuk kedalam *true positive*, dan jika data berlabel positif diprediksi negatif masuk kedalam *false negative*. Jika data berlabel negatif diprediksi negatif, itu dihitung sebagai *true negative*, dan jika data berlabel negatif diprediksi positif termasuk *false positive* (Fawcett, 2006).

Tabel 1 *Confusion Matrix*

Class		Aktual	
		Positive	*Negative*
Prediksi	*Positive*	*True Positive* (TP)	*False Positive* (FP)
	Negative	*False Negative* (FN)	*True Negative* (TN)

Berdasarkan *matrix confusion*, maka dapat dihitung nilai *accuracy*, *precision*, dan *recall*. *Accuracy* menggambarkan hasil klasifikasi sentimen positif dan negatif yang benar dari keseluruhan data (Hadna, Santosa, & Winarto, 2016). Dirumuskan sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Precision menggambarkan persentase klasifikasi sentimen positif yang benar-benar bernilai positif (Hadna, Santosa, & Winarto, 2016), dirumuskan :

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall menggambarkan persentase tanggapan yang klasifikasikan positif dari seluruh tanggapan yang bernilai positif (Hadna, Santosa, & Winarto, 2016) dirumuskan sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

F1 adalah rata-rata dari perbandingan presisi dan *recall*. F1 baik untuk digunakan sebagai acuan performansi algoritma jika jumlah *False Negative* dan *False Positive* berbeda jauh. F1 dirumuskan dengan:

$$F1 = \frac{2(Recall\ x\ Presisi)}{Recall + Presisi} \quad (9)$$

Visualisasi Data

Visualisasi dengan *WordCloud* akan memperlihatkan kata yang paling sering muncul/digunakan pada data yang telah dianalisis sebelumnya. Dalam *python* untuk memvisualisasikan *WordCloud* menggunakan *library WordCloud*. Semakin besar ukuran kata menandakan kata tersebut sering muncul/digunakan dalam teks tersebut.

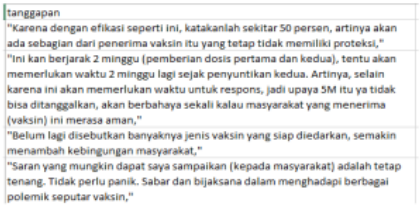
Visualisasi dengan diagram *pie* menunjukkan persentase hasil tanggapan positif dan negatif terhadap vaksin COVID-19 menggunakan *library matplotlib* yang ada di *python*.

HASIL DAN PEMBAHASAN

Bab ini membahas hasil dari penelitian yang telah dilakukan, dimulai dari pengambilan data dari media berita online sampai visualisasi data.

Pengambilan Data

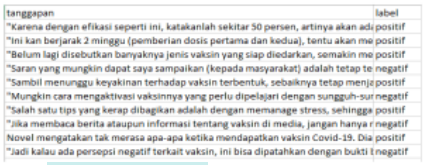
Pengambilan data dilakukan secara manual dengan mencari berita online dengan topik tanggapan masyarakat terhadap vaksin COVID-19 dari tahun 2020-2021. Total data yang ditemukan sejumlah 283 data. Data yang telah ditemukan kemudian disimpan dalam *Microsoft Excel* dalam format .csv. Hasil data yang telah dikumpulkan ditampilkan pada Gambar 4.


tanggapan
"Karena dengan efikasi seperti ini, katakanlah sekitar 50 persen, artinya akan ada sebagian dari penerima vaksin itu yang tetap tidak memiliki proteksi,"
"Ini kan berjarak 2 minggu (pemberian dosis pertama dan kedua), tentu akan memerlukan waktu 2 minggu lagi sejak penyuntikan kedua. Artinya, selain karena ini akan memerlukan waktu untuk respons, jadi upaya 5M itu ya tidak bisa ditanggalkan, akan berbahaya sekali kalau masyarakat yang menerima (vaksin) ini merasa aman,"
"Belum lagi disebutkan banyaknya jenis vaksin yang siap diedarkan, semakin menambah kebingungan masyarakat,"
"Saran yang mungkin dapat saya sampaikan (kepada masyarakat) adalah tetap tenang. Tidak perlu panik. Sabar dan bijaksana dalam menghadapi berbagai polemik seputar vaksin,"

Gambar 4 Tanggapan Masyarakat

Pelabelan Data

Tahap pelabelan ini dilakukan secara manual dalam *Microsoft excel* dengan bantuan 2(dua) sukarelawan. Pelabelan dilakukan pada keseluruhan data. Label yang diberikan berupa positif atau negatif pada masing-masing tanggapan masyarakat pada dengan panduan *Load Dictionary* untuk mencocokkan kata positif dan kata negatif. Hasil pelabelan data ditampilkan pada Gambar 5.



tanggapan	label
"Karena dengan efikasi seperti ini, katakanlah sekitar 50 persen, artinya akan ada	positif
"Ini kan berjarak 2 minggu (pemberian dosis pertama dan kedua), tentu akan me	positif
"Belum lagi disebutkan banyaknya jenis vaksin yang siap diedarkan, semakin me	positif
"Saran yang mungkin dapat saya sampaikan (kepada masyarakat) adalah tetap te	negatif
"Sambil menunggu keyakinan terhadap vaksin terbentuk, sebaiknya tetap menja	positif
"Mungkin cara mengaktivasi vaksinnya yang perlu dipelajari dengan sungguh-sur	negatif
"Salah satu tips yang kerap dibagikan adalah dengan memanage stress, sehingga	positif
"Jika membaca berita ataupun informasi tentang vaksin di media, jangan hanya r	negatif
Novel mengatakan tak merasa apa-apa ketika mendapatkan vaksin Covid-19. Dia	positif
"Jadi kalau ada persepsi negatif terkait vaksin, ini bisa dipatahkan dengan bukti i	negatif

Gambar 5 Hasil Pelabelan Data

Text Preprocessing

Tahap ini melakukan pengolahan terhadap keseluruhan data. Tahap ini terdiri dari *cleansing*, *case folding*, *parsing/tokenizing*, *filtering dan stemming*. Sebelum melakukan *text pre-processing* data di-inputkan terlebih dahulu melalui *python* menggunakan *library pandas*. Sintaks untuk input data ditampilkan pada Gambar 6.

```
import pandas as pd

df_review = pd.read_csv("trainingset.csv")
```

Gambar 6 Sintaks input data

Setelah itu tampilkan isi dari data yang telah diinputkan dengan cara memanggil nama variabelnya. Jika data telah berhasil di-inputkan maka data ditampilkan seperti pada Gambar 7.

	tanggapan	label
0	"Karena dengan efikasi seperti ini, katakanlah...	positif
1	"Ini kan berjarak 2 minggu (pemberian dosis pe...	positif
2	"Belum lagi disebutkan banyaknya jenis vaksin ...	positif
3	"Saran yang mungkin dapat saya sampaikan (kepa...	negatif
4	"Sambil menunggu keyakinan terhadap vaksin ter...	positif

Gambar 7 Data Sebelum *Text Pre-processing*

Sebelum dilakukan *text preprocessing*, kolom label diganti menjadi angka 1 untuk label positif dan 0 untuk label negatif supaya memudahkan proses validasi dan evaluasi. Selanjutnya tahap pertama dalam *text pre-processing* adalah *cleansing* yang bertujuan untuk menghapus tanda baca yang tidak dibutuhkan untuk analisis sentimen. Pada *python* proses *cleansing* dilakukan menggunakan *library string* dan *re*. Sintaks *cleansing* ditampilkan pada Gambar 8.

```
import string, re

def cleansing(data):

    # hapus punctuation
    remove = string.punctuation
    translator = str.maketrans(remove, ' '*len(remove))
    data = data.translate(translator)

    #remove angka
    data = re.sub(r'[0-9]+',r'', data)

    return data
# jalankan cleansing data
review = []
for index, row in df_preprocessed.iterrows():
    review.append(cleansing(row["tanggapan"]))

df_preprocessed["tanggapan"] = review
df_preprocessed
```

Gambar 8 Sintaks *Cleansing* Data

Tampilan data setelah melalui tahap *cleansing* dapat dilihat pada Gambar 9.

	tanggapan	label
0	Karena dengan efikasi seperti ini katakanlah...	1
1	Ini kan berjarak minggu pemberian dosis per...	1
2	Belum lagi disebutkan banyaknya jenis vaksin ...	1
3	Saran yang mungkin dapat saya sampaikan kepa...	0
4	Sambil menunggu keyakinan terhadap vaksin ter...	1

Gambar 9 Setelah Melalui Tahap *Cleansing*

Langkah selanjutnya adalah *case folding* yang bertujuan untuk menyeragamkan seluruh huruf menjadi huruf kecil (*lowercase*). Pada *python* proses *case folding* dilakukan menggunakan fungsi *lower*(). Sintaks untuk proses *case folding* ditampilkan pada Gambar 10.

```
def casefolding(data):
    # lower text
     data = data.lower()

     return data

# jalankan casefolding
review = []
for index, row in df_preprocessed.iterrows():
  review.append(casefolding(row["tanggapan"]))

df_preprocessed["tanggapan"] = review
df_preprocessed
```

Gambar 10 Sintaks *Case Folding*

Data yang telah melalui tahap *case folding* ditampilkan pada Gambar 11.

	tanggapan	label
0	karena dengan efikasi seperti ini katakanlah...	1
1	ini kan berjarak minggu pemberian dosis per...	1
2	belum lagi disebutkan banyaknya jenis vaksin ...	1
3	saran yang mungkin dapat saya sampaikan kepa...	0
4	sambil menunggu keyakinan terhadap vaksin ter...	1

Gambar 11 Setelah Melalui Tahap *Case Folding*

Selanjutnya adalah tahap *parsing/tokenizing* yaitu mengubah kalimat menjadi beberapa kata atau token. Pada *python* proses *parsing/tokenizing* dilakukan menggunakan *library nltk* dengan fungsi *tokenize* untuk mengubah kalimat menjadi token. Sintaks *tokenizing* ditampilkan pada Gambar 12.

```
#tokenizing import
import nltk
from nltk.tokenize import TweetTokenizer

# function tokenize
def tokenizing(data):
  #tokenizing
  tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True, reduce_len=True)
  data = tokenizer.tokenize(data)

  return data

# jalankan tokenizing
review = []
for index, row in df_preprocessed.iterrows():
    review.append(tokenizing(row["tanggapan"]))

df_preprocessed["tanggapan"] = review
df_preprocessed
```

Gambar 12 Sintaks *Tokenizing*

Hasil tahap *parsing/tokenizing* ditampilkan pada Gambar 13.

	tanggapan	label
0	[karena, dengan, efikasi, seperti, ini, kataka...	1
1	[ini, kan, berjarak, minggu, pemberian, dosis,...	1
2	[belum, lagi, disebutkan, banyaknya, jenis, va...	1
3	[saran, yang, mungkin, dapat, saya, sampaikan,...	0
4	[sambil, menunggu, keyakinan, terhadap, vaksin...	1

Gambar 13 Hasil Tahap *Tokenizing*

Selanjutnya adalah tahap *filtering* yang bertujuan untuk menghilangkan kata yang tidak bermakna (*stopword/stoplist*) seperti kata hubung atau kata ganti. Pada *python* proses *filtering* dilakukan menggunakan *library* nltk untuk men-*download* kamus *stopwords* dan *library* Sastrawi dengan fungsi *StopWordRemover* untuk menghapus kata hubung atau kata ganti. Sintaks untuk tahap *filtering* ditampilkan pada Gambar 14.

```
# import library
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

#jalankan pada data kita
review = []
for index, row in df_preprocessed.iterrows():
    review.append(stopword.remove(row["tanggapan"]))

df_preprocessed["tanggapan"] = review
df_preprocessed
```

Gambar 14 Sintaks *Filtering/Stopword Removal*

Data yang telah melalui tahap *filtering* ditunjukkan pada Gambar 15.

	tanggapan	label
0	karena dengan efikasi seperti ini katakanlah s...	1
1	ini kan berjarak minggu pemberian dosis pertam...	1
2	belum lagi disebutkan banyaknya jenis vaksin y...	1
3	saran yang mungkin dapat saya sampaikan kepada...	0
4	sambil menunggu keyakinan terhadap vaksin terb...	1

Gambar 15 Hasil *Filtering/Stopword Removal*

Tahap terakhir adalah *stemming*, yaitu proses pengubahan kata menjadi kata dasar dengan menghilangkan imbuhan. Pada *python* proses *stemming* dilakukan menggunakan *library* Sastrawi dengan modul *Stemmer* untuk mengubah kata menjadi kata dasar. Sintaks untuk proses *stemming* ditampilkan pada Gambar 16.

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

#function stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# jalankan pada data kita
review = []
for index, row in df_preprocessed.iterrows():
    review.append(stemmer.stem(row["tanggapan"]))

df_preprocessed["tanggapan"] = review
df_preprocessed
```

Gambar 16 Sintaks *Stemming*

Data yang telah melalui tahap *stemming* ditampilkan pada Gambar 17.

	tanggapan	label
0	efikasi persen penerima vaksin memiliki proteksi	1
1	berjarak minggu pemberian dosis waktu minggu p...	1
2	banyaknya jenis vaksin diedarkan menambah kebi...	1
3	saran masyarakat tenang panik sabar bijaksana ...	0
4	menunggu keyakinan vaksin terbentuk menjalanka...	1

Gambar 17 Hasil Tahap *Stemming*

Total keseluruhan data yang telah dibersihkan sejumlah 283 tanggapan, kemudian dibagi menjadi dua, data *training* sebanyak 90% yaitu sejumlah 254 dan data *testing* 10% yaitu sejumlah 29 data *testing*. Pembagian data pada *python* menggunakan *library sklearn*. Sintaks untuk pembagian data ditampilkan pada Gambar 18.

```
#membagi data set
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df_preprocessed['tanggapan'], df_preprocessed['label'],
                                                    test_size=0.1, stratify=df_preprocessed['label'], random_state=0)
```

Gambar 18 Sintaks Pembagian Data

Hasil pembagian data *training* dan data *testing* ditampilkan pada Gambar 19 dan Gambar 20.

```
(195    antre tanah abang lansia animo sungguh dukung ...
235    obat vaksin bebas risiko astrazeneca manfaat r...
14     vaksin covid pakai masker cuci tangan jaga jar...
103     gratis terap efek samping takut ragu aman vaksin
22     uji praklinik aman lanjut fase uji klinik bada...
                             ...
263    bijak masyarakat malas vaksinasi gratis malas ...
168    nego negara produsen vaksin mudah mudah mei no...
36     tambah isu vaksin cina masyarakat kualitas vak...
96     baik sehat sih tuju aja rencana perintah dampa...
26     putus vaksin gratis komitmen presiden jokowi d...
Name: tanggapan, Length: 254, dtype: object,
```

Gambar 19 Data *Training*

```
1      jarak minggu beri dosis waktu minggu sunti wak...
228    harap upaya tingkat percaya masyarakat program...
183    prinsip vaksin masyarakat izin badan pom aspek...
151    masyarakat non lansia usia tahun damping lansi...
126    timbang situasi kondisi kait sedia vaksin peng...
180    paham masyarakat keliru kait bebas traveling v...
61     kondisi alamiah psikologis umum kelompok muda ...
49     sosialisasi tangan covid kait vaksinasi protok...
249    senang berita datang vaksin covid bal kalo gra...
172    vaksinasi ganti protokol sehat traveling takut...
170          teliti lindung virus tahan vaksinasi penuh
Name: tanggapan, dtype: object,
```

Gambar 20 Data *Testing*

Pembobotan Term Frequency-Inverse Document Frequency

Data yang telah dibagi menjadi data *training* dan data *testing* dilanjutkan proses pembobotan TF-IDF. Pengolahan TF-IDF di *python* mengunakan *library TfidfVectorizer*. Fungsi dari tahap ini adalah mengubah data teks menjadi data numerik supaya bisa dilakukan perhitungan dan juga untuk menghitung bobot setiap kata, semakin besar bobot suatu kata maka kata tersebut dianggap penting. Sintaks pembobotan TF-IDF ditampilkan pada Gambar 21.

```
#proses TF-IDF (string to float)
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()

# Implementasi pada dokumen kita
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
f_test = vectorizer.transform(f_test)
```

Gambar 21 Sintaks Pembobotan TF-IDF

Hasil pembobotan TF-IDF dapat dilihat pada Gambar 22.

```
(0, 923)      0.08405592590509782
(0, 502)      0.18109085213970988
(0, 212)      0.3499258239258101
(0, 802)      0.38617022723412187
(0, 45)       0.4149390286058649
(0, 457)      0.2768716549997068
(0, 1)        0.36575843782222295
(0, 824)      0.36575843782222295
(0, 53)       0.4149390286058649
(1, 637)      0.3743539125524612
(1, 917)      0.31569963840205295
(1, 490)      0.27807380744487803
(1, 66)       0.31569963840205295
```

Gambar 22 Hasil Pembobotan TF-IDF

Contoh cara membaca data baris pertama pada Gambar 23 adalah kata pertama pada kalimat ke-0 terletak pada urutan index ke-923 dan bobotnya adalah 0.08405592590509782.

Klasifikasi Support Vector Machine

Tahap ini menggunakan algoritma SVM untuk membuat model klasifikasi untuk memprediksi kelas berdasarkan model prediksi yang dibuat dari data *training* untuk mendapatkan pola yang nantinya bisa digunakan untuk memprediksi. Sintaks untuk memanggil fungsi algoritma SVM ditampilkan pada Gambar 23.

```
# algoritma SVM
from sklearn import svm

clf = svm.SVC(kernel="linear")
```

Gambar 23 Memanggil Fungsi Algoritma SVM

Lalu sintaks untuk pembuatan model klasifikasi menggunakan data *training* ditampilkan pada Gambar 24.

```
clf.fit(X_train,y_train)
predict = clf.predict(X_test)
```

Gambar 24 Pembuatan Model Klasifikasi

Model klasifikasi algoritma SVM menggunakan data *training* ditampilkan pada Gambar 25.

```
(0, 145)      0.21114364134752145
(0, 198)      0.45794904756864613
(0, 369)      0.44061050850165573
(0, 550)      0.5450871252650836
(0, 588)      0.3087756782697215
(0, 858)      0.38296776014776035
(0, 868)      0.11152004512143725
(1, 17)       0.2295121383707279
(1, 24)       0.19964439439434314
(1, 56)       0.24698364857805316
(1, 91)       0.4590242767414558
```

Gambar 25 Model Klasifikasi Dengan Algoritma SVM

Model klasifikasi digunakan untuk memprediksi label dari data *testing*. Data *testing* yang telah melalui tahap pembobotan TF-IDF dimasukkan ke dalam model klasifikasi untuk diprediksi labelnya. Data *testing* yang dimasukkan kedalam model klasifikasi hanyalah bagian kolom tanggapannya saja tanpa kolom label karena bagian labelnya yang akan di prediksi oleh model klasifikasi. Hasil klasifikasi dari algoritma SVM dapat dilihat pada Gambar 26.

	tanggapan	label
0	kaji efektivitas vaksin cipta kebal individu t...	positif
1	perintah overclaim vaksin olah sedia solusi sa...	positif
2	jarak minggu beri dosis waktu minggu sunti wak...	positif
3	amelia dapat program vaksinasi covid hasil edu...	positif
4	aneh situasi darurat kian buruk perintah pakai...	positif

Gambar 26 Hasil Klasifikasi Algoritma SVM

Validasi K-fold Cross Validation

Validasi berguna untuk mengetahui seberapa baik model yang telah dibuat dan menggambarkan tingkat akurasi analisis yang

dilakukan. Uji validasi dengan *k-fold cross validation* caranya adalah dengan mengulangi sampel data uji beberapa kali dengan sampel yang berbeda-beda pada setiap pengujian. Pada *python cross validation* menggunakan modul cross_val_score dari *library sklearn*. Penelitian ini menggunakan 5-*fold cross validation* yaitu menggunakan sampel data *testing* sebanyak 20% (dihasilkan dari total 100% dibagi 5 *fold*) sebanyak 5 kali dengan sampel yang berbeda-beda. Sintaks validasi dapat dilihat pada Gambar 27.

```
# validasi model dengan kfold cross validation
from sklearn.model_selection import cross_val_score

scores = cross_val_score(clf, X_train, y_train, cv=5)
scores= list(map('{:.2%}'.format,scores))
print(f'{scores}')
```

Gambar 27 *Cross Validation*

Untuk menghitung rata-rata hasil *cross validation* dapat menggunakan sintaks seperti yang ditunjukkan pada Gambar 28.

```
# rata rata cross val score
print(f"{cross_val_score(clf, X_train, y_train, cv=5).mean():.2%}")
```

Gambar 28 Rata-rata *Cross Validation*

Hasil validasi model klasifikasi dilakukan menggunakan 5-*fold cross validation* ditampilkan pada Tabel 3.

Tabel 2 5-*fold cross validation*

Fold ke	*Cross Validation Score*
1	72.55%
2	72.55%
3	74.51%
4	70.59%
5	72.00%
Rata-rata	72.44%

Evaluasi Confusion Matrix

Evaluasi dengan *confusion matrix* untuk mengetahui hasil klasifikasi *true positive, true negative, false positive dan false negative*. Pada *python* evaluasi dengan *confusion matrix, accuracy, precisión, recall dan* f1 menggunakan *library sklearn*. Sintaks untuk menghitung *confusion Matrix* ditampilkan pada Gambar 29.

```
# confusion matrix
tn, fp, fn, tp = confusion_matrix(y_test, predict).ravel()
```

Gambar 29 Sintaks *Confusion Matrix*

Pada penelitian ini terdapat 19 data berlabel positif yang diprediksi positif, 5 data berlabel negatif yang diprediksi positif, 5 data berlabel negatif yang diprediksi negatif dan 0 data berlabel positif yang diprediksi negatif. Hasil *confusion matrix* ditampilkan pada Tabel 4.

Tabel 3 *Confusion Matrix*

Class		Aktual	
		Positive	*Negative*
Prediksi	*Positive*	*True Positive* (TP) 19	*False Positive* (FP) 5
	Negative	*False Negative* (*FN*) *0*	*True Negative* (TN) 5

Akurasi digunakan untuk mengetahui berapa persen hasil klasifikasi sentimen positif dan negatif yang benar dari keseluruhan data. Presisi digunakan untuk mengetahui berapa persen klasifikasi sentimen positif yang benar-benar bernilai positif. Sedangkan *Recall* untuk mengetahui berapa persen tanggapan yang diprediksi positif dari seluruh tanggapan positif. F1 menunjukkan rata-rata perbandingan dari presisi dan *recall* yang di bobotkan. Sintaks untuk evaluasi ditampilkan pada Gambar 30.

```
# f1_score
print("hasil prediksi score f1 adalah: ")
print(f"{f1_score(y_test, predict):.2%}")

# accuracy score
print("hasil prediksi score accuracy adalah: ")
print(f"{accuracy_score(y_test, predict):.2%}")

# precision score
print("hasil prediksi score precision adalah: ")
print(f"{precision_score(y_test, predict):.2%}")

# recall score
print("hasil prediksi score recall adalah: ")
print(f"{recall_score(y_test, predict):.2%}")
```

Gambar 30 Skor Evaluasi

Hasil dari evaluasi ditampilkan pada Tabel 5.

Tabel 4 Hasil Evaluasi

Validation	*Score*
F1	87.80%
Accuracy	82.76%
Precision	78.26%
Recall	100%

Visualisasi Data Dengan WordCloud

Visualisasi dengan *Wordcloud* pada *python* menggunakan *library wordcloud*, sintaks untuk membuat *wordcloud* ditampilkan pada Gambar 31 dan 32.

Gambar 31 Sintaks *WordCloud* Positif

Gambar 32 Sintaks *WordCloud* Negatif

Pada Gambar 33 diketahui bahwa kata positif yang sering muncul yaitu kata *"aman", "sehat" dan "terima".* Mayoritas sentimen positif masyarakat adalah menerima dan percaya bahwa vaksin bisa melindungi diri dari virus.

Gambar 33 *WordCloud* Positif

Gambar 34 menampilkan kata negatif yang sering muncul yaitu kata "bayar", "tolak" dan "ragu". Mayoritas sentimen negatif masyarakat adalah menolak dan ragu karena khawatir efek samping dari vaksin.

Gambar 34 *WordCloud* Negatif

Visualisasi Data Dengan Diagram Pie

Untuk membuat diagram pie, dibutuhkan data nama label dan jumlah sentimen positif dan negatif. Sintaks untuk mendapatkan nama label dan jumlah sentimen ditampilkan pada Gambar 35.

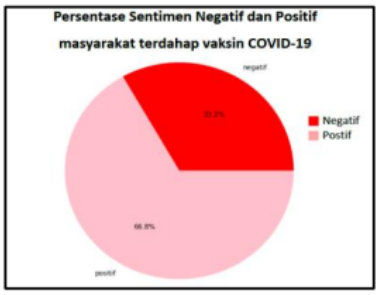
```
# memasukkan label dan jumlah kedalam variabel
type_labels = title_type.tanggapan.sort_values().index
type_counts = title_type.tanggapan.sort_values()
```

Gambar 35 Sintaks Jumlah Label

Selanjutnya sintaks untuk membuat diagram *pie* ditampilkan pada Gambar 36.

Gambar 36 Sintaks Diagram *Pie*

Gambar 37 merupakan diagram *pie* yang menunjukkan hasil persentase sentimen masyarakat berdasarkan klasifikasi SVM. Persentase keseluruhan tanggapan masyarakat terhadap Vaksin Covid-19 adalah sebanyak 66,8% positif dan sebanyak 33,2% negatif.



Gambar 37 Visualisasi Diagram Pie

SIMPULAN

Berdasarkan penelitian analisis sentimen tanggapan masyarakat terhadap vaksin COVID-19 menggunakan algoritma SVM yang telah dilakukan, disimpulkan:

1. Dari total 283 tanggapan masyarakat terhadap vaksin COVID-19 menghasilkan persentase sentimen positif yaitu sebanyak 66,8% dan negatif sebanyak 33,2%. Artinya sosialisasi pemerintah terhadap vaksin COVID-19 bisa dikatakan berhasil karena lebih banyak masyarakat yang setuju daripada yang menolak.
2. Hasil rata-rata validasi menggunakan 5-*fold cross validation* adalah 74,05% yang artinya model klasifikasi masih belum sempurna bisa disebabkan karena terdapat bahasa gaul yang tidak dikenali oleh kamus/sistem, atau terdapat kesalahan dalam pelabelan data secara manual.

3. Hasil evaluasi dengan *confusion matrix* terdapat 5 prediksi yang salah yaitu tanggapan negatif diprediksi positif oleh model klasifikasi. Penyebabnya bisa jadi karena kata hubung yang terhapus dalam proses *filtering*/*stopword removal*.
4. Penggunaan kamus *stopwords* dari *library nltk* kurang tepat karena masih terdapat beberapa kata ganti subjek yang tidak terhapus.

RUJUKAN

Fawcett, T. (2006). In *An Introduction to ROC Analysis*. *Pattern Recognition Letters* (pp. 861–874).

Febriyani. (2021). *Mengenal Vaksin Covid-19*. (Online). (https://ciputrahospital.com/mengenal-vaksin-covid-19/, diakses 20 Maret 2021).

Hadna, N. M., Santosa, P. I., & Winarto, W. W. (2016). Studi Literatur Tentang Perbandingan Metode Untuk Proses Analisis Sentimen di Twitter. *Seminar Nasional Teknologi Informasi dan Komunikasi 2016*.

Putri, G. S. (2020). *Keraguan pada Vaksin Covid-19, Bagaimana Masyarakat Harus Bersikap?*. (Online). (https://www.kompas.com/sains/read/2020/12/23/160000023/keraguan-pada-vaksin-covid-19-bagaimana-masyarakat-harus-bersikap?page=all, diakses 23 Maret 2021).

Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.

Safitri, R. N. (2020). *Analisis Sentimen Review Pelanggan Hotel Menggunakan Metode K-Nearest Neighbor (K-NN) (Studi Kasus: Hotels.com Booking.com, Agoda.com)*. Surabaya: Universitas Dinamika.

Ulfah, A. N., & Anam, M. K. (2020). Analisis Sentimen Hate Speech Pada Portal Berita Online Menggunakan Support Vector Machine (SVM). *Jurnal Teknik Informatika dan Sistem Informasi*, 1-10.

Wardani, F. K. (2019). *Analisis Sentimen Untuk Pemeringkatan Popularitas Situs Belanja Online di Indonesia Menggunakan Metode Naive Bayes (Studi Kasus Data Sekunder)*. Surabaya: Institut Bisnis dan Informatika STIKOM Surabaya.

Analisis Sentimen Tanggapan Masyarakat Terhadap Vaksin Covid-19 Menggunakan Algoritma Support Vector Machine (SVM)

ORIGINALITY REPORT

20%	17%	7%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	6%
2	Submitted to Universitas Brawijaya Student Paper	3%
3	ciputrahospital.com Internet Source	1%
4	ejournal.bsi.ac.id Internet Source	1%
5	repository.dinamika.ac.id Internet Source	1%
6	docplayer.info Internet Source	1%
7	repository.ub.ac.id Internet Source	1%
8	Hendry Cipta Husada, Adi Suryaputra Paramita. "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter	1%

	Menggunakan Algoritma Support Vector Machine (SVM)", Teknika, 2021 Publication	
9	jurnal.dinamika.ac.id Internet Source	1 %
10	repository.uin-suska.ac.id Internet Source	1 %
11	repository.uksw.edu Internet Source	1 %
12	Euis Saraswati, Yuyun Umaidah, Apriade Voutama. "Penerapan Algoritma Artificial Neural Network untuk Klasifikasi Opini Publik Terhadap Covid-19", Generation Journal, 2021 Publication	<1 %
13	www.kompasiana.com Internet Source	<1 %
14	Frizka Fitriana, Ema Utami, Hanif Al Fatta. "Analisis Sentimen Opini Terhadap Vaksin Covid - 19 pada Media Sosial Twitter Menggunakan Support Vector Machine dan Naive Bayes", Jurnal Komtika (Komputasi dan Informatika), 2021 Publication	<1 %
15	Fajar Fathur Rachman, Rani Nooraeni, Lia Yuliana. "Public Opinion of Transportation integrated (Jak Lingko), in DKI Jakarta, Indonesia", Procedia Computer Science, 2021	<1 %

	Publication	
16	Submitted to University of Westminster Student Paper	<1 %
17	core.ac.uk Internet Source	<1 %
18	Submitted to Universitas Negeri Surabaya The State University of Surabaya Student Paper	<1 %
19	id.scribd.com Internet Source	<1 %
20	bambangunesa.wordpress.com Internet Source	<1 %
21	ejurnal.its.ac.id Internet Source	<1 %
22	repository.its.ac.id Internet Source	<1 %
23	www.coursehero.com Internet Source	<1 %
24	mafiadoc.com Internet Source	<1 %
25	Desdwyatma Wahyu Wibawa, Muhammad Nasrun, Casi Setianingsih. "Sentiment Analysis on User Satisfaction Level of Cellular Data Service Using the K-Nearest Neighbor (K-NN) Algorithm", 2018 International Conference on Control,	<1 %

	Electronics, Renewable Energy and Communications (ICCEREC), 2018 Publication	
26	es.scribd.com Internet Source	<1 %
27	id.123dok.com Internet Source	<1 %
28	journal.ikippgriptk.ac.id Internet Source	<1 %
29	jtsiskom.undip.ac.id Internet Source	<1 %
30	jurnal.upnyk.ac.id Internet Source	<1 %
31	repositori.usu.ac.id Internet Source	<1 %
32	www.iaeng.org Internet Source	<1 %
33	doku.pub Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		