ANALISIS SENTIMEN MENGENAI *SANDWICH GENERATION* PADA MEDIA SOSIAL *X/TWITTER* DENGAN MENERAPKAN ALGORTIMA SUPPORT VECTOR MACHINE

Yoga Pramudita; Dimas Aryo Anggoro S.Kom., M.Sc.
Program Studi Teknik Informatika, Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta

Abstrak

Para Sandwich generation memiliki tuntutan harus menyeimbangkan tanggung jawab menjaga orang tua dan anak, mereka mengalami tingkat stress yang tinggi .Tingkat stress yang dialami sandwich generation tidak hanya memengaruhi hubungannya dengan orang yang di sayangi, anak dan keluarga, tetapi juga memengaruhi kesejahteraan dirinya sendiri. Para sandwich generation melampiaskan dengan cara curhat secara langsung dengan orang lain atau curhat melalui sosial media. Algoritma *Support Vector Machine* (SVM) merupakan metode yang memiliki tingkat nilai akurasi yang paling tinggi dibuktikan dengan penelitian sebelumnya yang terkait dalam penelitian ini. Algoritma *Support Vector Machine* (SVM) dalam penelitian ini digunakan untuk analisis sentiment pengguna *Twitter* terhadap topik *Sandwich Generation* dengan cara menghitung nilai dari data tweet yang menghasilkan nilai akurasi sebesar 88%, nilai presisi dan recall yang baik untuk kedua kelas. Sedangkan untuk hasil sentiment terhadap topik *Sandwich Generation* pada media sosial *Twitter* tergolong negatif. Hasil penelitian ini yang dilakukan dengan menerapkan algoritma *Support Vector Machine* (SVM) menunjukkan bahwa hasil analisis sentimen mengenai *Sandwich Generation* yang ada di media sosial *Twitter* tergolong negatif.

Kata Kunci: analisis sentimen, *sandwich generation*, *support vector machine*.

Abstract

Sandwich generation has to balance the responsibility of taking care of parents and children, they experience high levels of stress. The level of stress experienced by sandwich generation not only affects their relationship with loved ones, children and family, but also affects their own well-being. The sandwich generation vent by venturing directly with others or vent through social media. The Support Vector Machine (SVM) algorithm is a method that has the highest level of accuracy as evidenced by previous research related to this study. In this study, the sentiment of Twitter users regarding Sandwich Generation was analyzed using the Support Vector Machine (SVM) algorithm. The method calculated the value of tweet data, yielding an accuracy value of 88%, a high precision value, and strong recall for both classes. As for the sentiment results on the topic of Sandwich Generation on Twitter social media is classified as negative. The results of this research conducted by applying the Support Vector Machine (SVM) algorithm show that the results of sentiment analysis on Sandwich Generation on Twitter social media are negative.

Keywords: *sentiment analysis*, *sandwich generation*, *support vector machine*.

1. PENDAHULUAN

Hampir di semua negara, fenomena populasi penduduk yang produktif menanggung tanggung jawab untuk kebutuhan 3 (tiga) generasi yang akan datang terjadi (Suharyono & Hadiningrat, n.d.-a). Ini termasuk di Indonesia, hal ini dikenal dengan istilah *Sandwich Generation* (Statistik Penduduk Lanjut Usia, 2022). Generasi ini merujuk pada generasi produktif yang dianalogikan seperti *sandwich*, dimana bagian atas roti menunjukkan generasi di atasnya (orang tua), sedangkan daging menunjukkan dirinya sendiri, dan roti bagian paling bawah menunjukkan generasi dibawah mereka (anak). *Sandwich generation* sendiri biasanya terdiri dari orang dewasa yang berusia 40-50 tahun yang sedang mengurus anak-anak dan orang tua mereka (Suharyono & Hadiningrat, n.d.-b). Para *sandwich generation* mengalami tingkat stress yang lebih tinggi karena mereka harus menyeimbangkan tanggung jawab mereka untuk menjaga anak dan orang tua mereka (Nurmila et al., n.d.). Tingkat stress yang dialami *sandwich generation* tidak hanya memengaruhi hubungannya dengan orang yang di sayangi, anak dan keluarga, tetapi juga memengaruhi kesejahteraan dirinya sendiri. *Sandwich generation* biasanya mendapati problematika kesehatan mental seperti halnya depresi dan kecemasan terhadap suatu hal, kelelahan fisik dan mental, perasaan bersalah, insomnia, merasa khawatir terus menerus, dan kehilangan minat pada aktivitas yang mereka sukai. Para *sandwich generation* melampiaskan perasaannya dengan cara berbicara ke orang lain secara langsung atau melalui media sosial kepada orang-orang yang memiliki kesamaan situasi untuk mendapatkan saran dan nasihat. Hal ini dapat membantu mereka merasa lebih terhubung dan tidak sendirian dalam menghadapi beban sebagai *sandwich generation*(*Heavy Burden of the 'Sandwich Generation' - Kompas.Id*, n.d.).

Dalam era digital, *Twitter* termasuk salah satu *social media* populer yang sering dipakai warga untuk berbagi informasi juga pendapat. Indonesia memiliki 24 juta pengguna *Twitter*, menempati posisi ke-5, menurut data Statista pada Januari 2023 (Statista, 2023). Media sosial ini sudah menjadi platform utama bagi masyarakat yang berfungsi untuk berbagi pendapat, pengalaman, dan berinteraksi dengan topik-topik yang relevan dalam rutinitas sehari-hari. Salah satu contoh topik yang semakin mendapat perhatian di media sosial, termasuk di media sosial *Twitter* adalah *Sandwich Generation*. Pengguna *Twitter* sering menggunakan platform ini untuk berbagi pendapat dan perasaan mereka tentang peran mereka sebagai *sandwich generation*. Mereka membahas tentang kesulitan yang mereka hadapi, kebahagiaan, atau kecemasan yang mereka alami selama

menjadi *Sandwich generation*. Dari opini masyarakat tentang topik *Sandwich Generation* pada *Twitter* dapat dijadikan sebagai subjek penelitian dan menjadi dataset untuk dilakukan *sentiment analysis*. *Sentiment Analysis*, yaitu proses yang bertujuan untuk mengekstrak, memahami data, dan mengolah data secara otomatis oleh sebuah sistem untuk mengumpulkan informasi untuk menentukan apakah pendapat tentang suatu topik cenderung ke sentiment positif, negatif, atau netral (Amrustian et al., 2022).

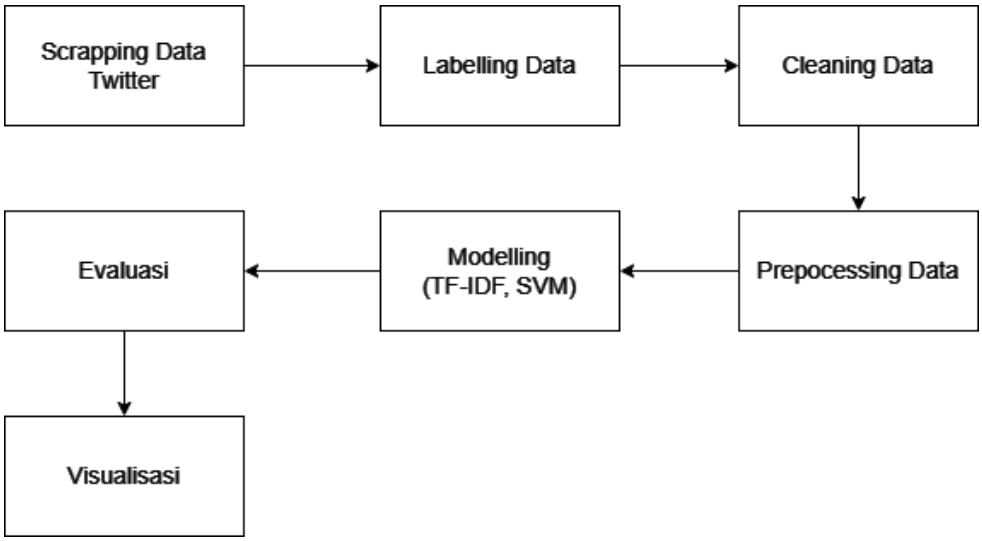
Penelitian sebelumnya yang dipublikasikan oleh Wayan Rangga Pinastawa dan Afifah Arifuddin tentang membandingkan algoritma SVM dan NB *Classifier* diperoleh kesimpulan bahwa metode SVM lebih unggul daripada algoritma NB *Classifier*. SVM menghasilkan tingkat akurasi tertinggi dengan nilai 96%, sedangkan *Naïve Bayes* memiliki nilai 92% yang hanya terpaut 4%. (Wayan Rangga Pinastawa & Afifah Arifuddin, n.d.). Selanjutnya, penelitian tentang analisis data komentar mengenai sistem E-Tilang yang dilakukan di *social media Twitter* dengan menerapkan algoritma SVM telah berhasil mengklasifikasikan sentimen dimana jumlah datanya adalah 2999 data, didalamnya terdapat 426 opini positif, 591 opini negative, dan 1910 opini netral . Penelitian ini berhasil mengklasifikasikan 3 kelas sentimen. Pada penelitian ini, algoritma SVM menghasilkan akurasi dengan nilai 74.20%, nilai presisi sebesar 83.33% dan recall sebesar 5.28% (Oktavia & Ramadahan, 2023a). Kemudian untuk penelitian yang meneliti tentang penerapan algoritma SVM tetapi dengan menggunakan TF-IDF N-GRAM untuk *text classification* menunjukkan kesimpulan bahwa dengan TF-IDF dinilai dapat menjadikan nilai akurasi pada saat proses klasifikasi artikel ilmiah *Syntax Journal of Informatics* lebih unggul.(Arifin et al., n.d.).

Algoritma SVM merupakan algoritma klasifikasi yang menghasilkan tingkat akurasi tertinggi dibuktikan dengan beberapa hasil dari penelitian yang berkaitan dalam penelitian ini. Menilai opini masyarakat tentang topik *Sandwich Generation* pada media sosial *Twitter* apakah opini tersebut bernilai positif, negatif, atau netral merupakan tujuan dari adanya penelitian ini. Untuk manfaat dari penelitian ini adalah analisis sentiment memberikan informasi dan mengukur bagaimana orang-orang merespons situasi tersebut. Apakah mereka memberikan dukungan dan pemahaman, atau apakah mereka merasa marah, frustasi, atau kecewa. Dengan mengetahui sentiment ini, organisasi atau pemerintah dapat menilai sejauh mana masalah tersebut dapat memengaruhi reputasi mereka. Sedangkan untuk luaran untuk penelitian ini adalah skor sentiment yaitu nilai numerik yang menggambarkan sentiment positif, negative, atau netral dari data komentar

Twitter yang telah dianalisis. Visualisasi data yang berupa grafik, *word cloud*, dan *heatmap* emosi.

2. METODE

Penelitian ini menerapkan metode saintifik pada data mining dan metode Knowledge Discovery in Database. Tujuan penelitian ini adalah untuk mengetahui bagaimana algoritma untuk memperlajari data bekerja, membuat model, dan menentukan sebuah pola yang sebelumnya belum diketahui saat mengumpulkan data yang belum diketahui dari sebuah basis data yang sangat besar (Imaduddin et al., n.d.). Algoritma *Support Vector Machine* (SVM) digunakan dalam tahap klasifikasi sebelum melakukan proses evaluasi. Pengumpulan data, labeling, preprocessing, pembagian, TF-IDF, Support Vector Machine, Evaluasi (Confusion Matrix), dan Visualisasi adalah tahapan metode penelitian ini. Alur tahapan penelitian secara detail ditunjukkan oleh gambar 1.



Gambar 1. *Tahap Penelitian*

2.1 *Scrapping Data*

Scrapping merupakan sebuah proses untuk mengumpulkan data, menyimpan data, dan memvalidasi data sehingga dapat digunakan sebagai informasi yang relevan. Scrapping juga dapat dilakukan dengan berbagai cara, termasuk mengubah data non-struktur menjadi data terstruktur sehingga dapat disimpan dalam sebuah database.

2.2 *Labelling Data*

Dalam proses ini, nantinya data akan dikategorikan menjadi tiga label diantaranya label netral, label negatif, dan label positif. label positif merupakan komentar yang mengandung kata yang bersifat memberikan masukan positif atau memberikan semangat, untuk label negatif diberikan pada komentar yang bersifat hujatan atau mengandung emosi negatif, sedangkan untuk label netral diberikan pada komentar yang tidak mengandung keduanya.

Pengkategorian dilakukan secara mandiri dan subjektif (Ferdiana et al., 2019).

2.3 *Cleaning Data*

Proses Clening memulai dengan menghapus elemen seperti url, username, tanda pagar(#), symbol(@), dan karakter unik lainnya dari tweet yang diunduh. Ini menghasilkan data yang bebas dari simbol dan url. (Dewi & Mailoa, 2023).

2.4 *Prepocessing Data*

Proses mempersiapkan data analisis sebelum digunakan dikenal sebagai *data prepocessing.* Tujuan dari proses ini adalah untuk membuat data lebih baik, sehingga data akan berkualitas tinggi dan siap untuk digunakan dalam proses berikutnya (Dyah Fritama et al., 2023). *Data prepocessing* dilakukan menggunakan bahasa pemrograman python melalui *Google Colab* dan library yang disediakan oleh python untuk melakukan *case folding, data cleaning, normalization, tokenization, stemming, dan stopword removal*.

2.2.1 *Case Folding*

Teks yang mengalami ketidakkonsistenan dalam penggunaan huruf, seperti komentar dengan huruf yang tidak konsisten, diubah selama tahap ini. Tujuan dari proses case folding adalah untuk membuat huruf dalam teks koemntar yang dibersihkan menjadi huruf kecil yang sesuai dengan format standar.

2.2.2 *Tokenizing*

Langkah penting dalam pemrosesan teks adalah tokonezing, yang menggabungkan beberapa karakter dalam teks menjadi unit kata. Dalam langkah-langkah ini, karakter pembatas, angka, dan tanda baca dapat dihapus (Satria Buana et al., 2023).

2.2.3 *Stopword Removal*

Proses stopwords dalam pemrosesan teks dengan penghilangan kata-kata yang tidak dibutuhkan tetapi kata tersebut sering muncul. Ini termasuk kata-kata seperti "kok, lah, dll." yang tidak relevan untuk situasi (Oktavia & Ramadahan, 2023b).

2.2.4 *Stemming*

Mengubah kata berimbuhan imbuhan menjadi kata dasar dalam teks dikenal

sebagai stemming. Fungsi dari proses stemming adalah untuk mengelompokkan kata yang mempunyai definisi dasar yang sama (Oktavia & Ramadahan, 2023b).

2.5 *TF – IDF*

Pembobotab kata TF-IDF atau yang dikenal dengan *Term Frequency-Inversse Document Frequency* adalah permodelan yang menilai setiap kata yang berfungsi memberikan pembobotan pada setiap kata untuk mengidentifikasi hubungan antara kata dan data. Tujuan dari pembobotan kata adalah untuk menenmpatkan fitur kata pada tingkat yang lebih tinggi berdasarkan frekuensi kemunculannya. Tujuan dari pembobotan kata TF-IDF adalah untuk menentukan apakah kata kunci yang hendak dicari sudah sesuai dengan keinginan atau tidak, namun kata kunci yang sering digunakan tidak memiliki pengaruh signifikan terhadap kaitannya dengan kata kunci dalam dokumen. (Rofiqoh et al., 2017).

Pembobotan kata TF-IDF dirumuskan dengan persamaan dibawah (Salton & Buckley, 1988) :

$$Wj,i = \frac{\mathbf{nj,i}}{\sum \mathbf{k\ nk,i}} . \log 2 \frac{\mathbf{D}}{\mathbf{dj}}$$

keterangan:

Wj,i = Bobot nilai kata TF-IDF term ke-j pada dokumen ke-i.

nj,i = Jumlah term ke-j dalam dokumen ke-I yang muncul.

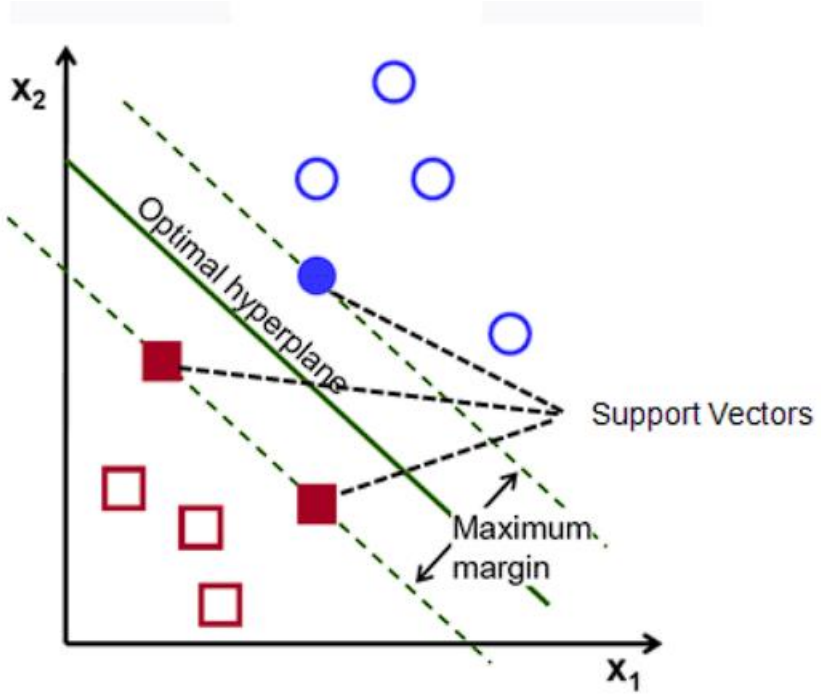
$\sum k\ nk,i$ = Jumlah keseluruhan term dalam dokumen ke-I.

D = Jumlah dokumen

dj = Jumlah dokumen yang mengandung kata ke-j

2.6 *Support Vector Machine (SVM)*

SVM merupakan algoritma yang biasanya mencari support untuk vector dan kemudian memisahkan dua kelas dengan cara mencari garis pembatas dengan margin.(Firdausi, n.d.), dengan ilustrasi sebagai berikut :



Gambar 2. *Klasifikasi SVM*

Dan dirumuskan sebagai berikut :

$$f(x) = \sum_{i=1}^{\infty} ai\ yi\ K(x.x') + b$$

ai = alfa ke-i

yi = kelas data latih yang ke-i

M = total data

K(x.x') = fungsi kernel yang menghitung jarak antara dua vektor fitur

x = data uji

xi = data latih ke-i

b = bias

2.7 *Confusion Matrix*

Nilai keberhasilan sistem yang dibangun dapat dinilai melalui evaluasi performasi (Firmanto & Sarno, n.d.). Dalam proses evaluasi ini menggunakan metode *Confusion Matrix*. Dalam hal ini menghitung nilai presisi, f1-score, akurasi, dan nilai recall, *Confusion Matrix* dalam metode evaluasi ini digunakan. Berikut istilah yang ada pada proses evaluasi *confusion matrix* (Anggoro, 2020):

True positive (TP) : Data positif diidentifikasi dengan benar.

True negative (TN) : Data negatif diidentifikasi dengan benar.

False positif (FP) : Data negatif diidentifikasi sebagai data positif.

False negative (FN) : Data positif diidentifikasi sebagai data negatif.

2.7.1 Akurasi

Pengukuran tingkat keakuratan dalam hal mengklasifikasikan dengan benar disebut dengan nilai akurasi. Dirumuskan dengan sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN}$$

2.7.2 Presisi

Presisi didefinisikan sebagai tingkat kebenaran antara permintaan informasi dan tanggapan sistem. Didefinisikan dengan rumus :

$$Presisi = \frac{TP}{TP + FP}$$

2.7.3 Recall

Pengukuran tingkat keberhasilan dalam menemukan kembali data disebut dengan recall. Didefinisikan dengan rumus :

$$Recall = \frac{TP}{TP + FN}$$

2.7.4 F1-Score

F1-Score ialah perhitungan harmonic mean dari nilai precision dan nilai recall. Didefinisikan dengan rumus :

$$F1 - Score = 2 \times \frac{Recall \times Presisi}{Recall + Presisi}$$

2.8 Visualisasi

Untuk menyederhanakan data dan membuatnya lebih mudah dipahami bagi pembaca, maka dilakukan visualisasi data yang menghasilkan *wordcloud* dari seluruh sentimen, *Pie Chart*, dan *Clustered Column Chart*.

3. HASIL dan PEMBAHASAN

3.1 Pengumpulan Data

Proses ini melakukan pengumpulan data yang ada di *Twitter* dengan mengambil komentar bahasa Indonesia dengan menggunakan kata kunci “sandwich generation” dengan menggunakan library *tweet harvest*. Diambil dari tahun 2022 sampai dengan tahun 2023.

```
!pip install pandas
!curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
!sudo apt-get install -y nodejs
```

ıpilkan output tersembunyi

```
# Crawl Data

filename = 'datasetsand.csv'
search_keyword = 'sandwich generation until:2022-03-28 since:2020-01-01 lang:id'
limit = 1400

!npx --yes tweet-harvest@latest -o "{filename}" -s "{search_keyword}" -l {limit} --token ""
```

Gambar 3. *Source Code Scrapping*

Data hasil scrapping berjumlah 1384 komentar, dengan kolom tanggal dan tweet. Yang dapat dilihat pada gambar 3.

	tanggal	tweet
0	27/03/23	@worksfess Aku dan suami wfh dari 2020 sampai ...
1	27/03/23	@Askrlfess Samaaaa apalagi w sandwich generati...
2	27/03/23	Ternyata fenomena sandwich generation itu ga h...
3	27/03/23	@txtdrpemerintah Anak2nya pasti uda nunggu dap...
4	27/03/23	Tapi privilege kecil itu tetep kalah sama kead...
5	27/03/23	bukan sandwich generation. supportive parents....
6	27/03/23	@hrdbacot Dikasi merantau ke ibukota dan tidak...
7	27/03/23	Waktu pertamakali kerja, mereka ga ngelarang a...
8	27/03/23	@vyndagnir Eee maap ga maksud nyangkal kamu T....
9	27/03/23	Tidak menjadi sandwich generation adalah sebua...

Gambar 4. *Hasil Scrapping*

3.2 Labelling Data

Secara manual dengan menggunakan 4 label yang masing masing diisi sesuai dengan sudut pandang pelabel dan mengambil modus dari 4 label tersebut, pelabelan dilakukan dengan membaca komentar satu per satu dan memeriksa kata-katanya. Untuk label positif merupakan komentar yang mengandung kata yang bersifat memberikan masukan positif atau memberikan semangat untuk para *Sandwich Generations*, untuk label negative diberikan pada komentar yang bersifat hujatan atau mengandung emosi negatif, sedangkan untuk label netral diberikan pada komentar yang tidak memiliki makna. Hasil dari labelling data pada tabel 1.

Tabel 1. Contoh Labelling Data

Tanggal	Tweet	label
09/03/23	buka tl muncul anak pejabat flexing mulu...ini gue yg ga	negatif

	berprivilege + jadi sandwich generation cuma bisa nelen promag biar ga muntah	
27/03/23	jadi buat kalian semua sandwich generation, percaya deh nggak selamanya kalian di fase bawah dan sulit. nanti pasti ada waktunya kalian ngerasa 'lebih plong'. semangat, terus berdoa, ikhlas, jangan ngerasa keluarga kalian beban. God bless youuu	positif
14/03/23	Tips bertahan hidup buat sandwich generation dong	netral

3.3 Data Cleaning

Tahap cleaning pertama adalah melakukan penghapusan username yang ada pada komentar, gambar 5 dibawah ini merupakan source code dari *data cleaning*.

Hapus Username

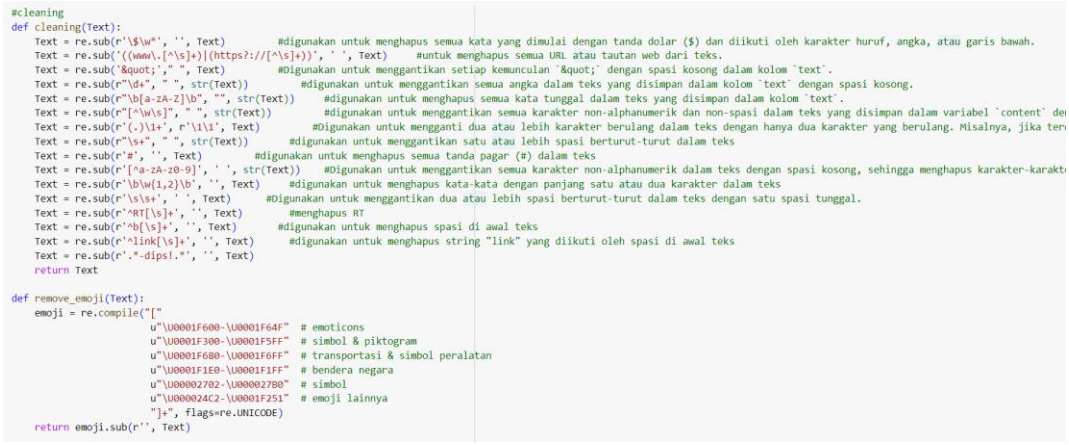
```
[11] import re

     # Fungsi untuk menghilangkan username
     def remove_usernames(text):
         username_pattern = r'@\w+'
         clean_text = re.sub(username_pattern, '', text)
         return clean_text

     # Mengaplikasikan fungsi remove_usernames ke kolom komentar
     df['remove_user'] = df['tweet'].apply(remove_usernames)
```

Gambar 5. *Source Code Hapus Username*

Tahap selanjutnya adalah melakukan cleaning lebih lengkap, seperti melakukan penghapusan emoji, menghapus url, huruf berulang, dan kata-kata dan simbol yang tidak dibutuhkan dalam proses klasifikasi nantinya. Tahap ini dilakukan dengan menggunakan bahasa python. Source code *data cleaning* ditunjukkan pada gambar 6.



```
#cleaning
def cleaning(Text):
    Text = re.sub(r'\$\w*', '', Text)         #digunakan untuk menghapus semua kata yang dimulai dengan tanda dolar ($) dan diikuti oleh karakter huruf, angka, atau garis bawah.
    Text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', ' ', Text)     #untuk menghapus semua URL atau tautan web dari teks.
    Text = re.sub('&quot;'," ", Text)         #Digunakan untuk menggantikan setiap kemunculan `&quot;` dengan spasi kosong dalam kolom `text`.
    Text = re.sub(r"\d+", " ", str(Text))         #digunakan untuk menggantikan semua angka dalam teks yang disimpan dalam kolom `text` dengan spasi kosong.
    Text = re.sub(r"\b[a-zA-Z]\b", "", str(Text))     #digunakan untuk menghapus semua kata tunggal dalam teks yang disimpan dalam kolom `text`.
    Text = re.sub(r"[^\w\s]", " ", str(Text))         #digunakan untuk menggantikan semua karakter non-alphanumerik dan non-spasi dalam teks yang disimpan dalam variabel `content` de
    Text = re.sub(r'(.)\1+', r'\1\1', Text)         #Digunakan untuk mengganti dua atau lebih karakter berulang dalam teks dengan hanya dua karakter yang berulang. Misalnya, jika ter
    Text = re.sub(r"\s+", " ", str(Text))       #digunakan untuk menggantikan satu atau lebih spasi berturut-turut dalam teks
    Text = re.sub(r'#', '', Text)         #digunakan untuk menghapus semua tanda pagar (#) dalam teks
    Text = re.sub(r'[^a-zA-z0-9]', ' ', str(Text))    #Digunakan untuk menggantikan semua karakter non-alphanumerik dalam teks dengan spasi kosong, sehingga menghapus karakter-karakt
    Text = re.sub(r'\b\w{1,2}\b', '', Text)     #digunakan untuk menghapus kata-kata dengan panjang satu atau dua karakter dalam teks
    Text = re.sub(r'\s\s+', ' ', Text)      #Digunakan untuk menggantikan dua atau lebih spasi berturut-turut dalam teks dengan satu spasi tunggal.
    Text = re.sub(r'^RT[\s]+', '', Text)        #menghapus RT
    Text = re.sub(r'^b[\s]+', '', Text)       #digunakan untuk menghapus spasi di awal teks
    Text = re.sub(r'^link[\s]+', '', Text)      #digunakan untuk menghapus string "link" yang diikuti oleh spasi di awal teks
    Text = re.sub(r'.*-dips!.*', '', Text)
    return Text

def remove_emoji(Text):
    emoji = re.compile("["
                       u"\U0001F600-\U0001F64F"  # emoticons
                       u"\U0001F300-\U0001F5FF"  # simbol & piktogram
                       u"\U0001F680-\U0001F6FF"  # transportasi & simbol peralatan
                       u"\U0001F1E0-\U0001F1FF"  # bendera negara
                       u"\U00002702-\U000027B0"  # simbol
                       u"\U000024C2-\U0001F251"  # emoji lainnya
                       "]+", flags=re.UNICODE)
    return emoji.sub(r'', Text)
```

Gambar 6. *Source Code Data Cleaning*

Tabel 2. Contoh Cleaning Data

Sebelum	Sesudah
@hrdbacot Dikasi merantau ke ibukota dan tidak menjadi sandwich generation yg punya tanggunganðŸ¥º	dikasi merantau ke ibukota dan tidak menjadi sandwich generation yg punya tanggungan

3.4 Prepocessing Text

Proses ini terdiri dari proses *transformation, tokenization,* dan *filtering.* Berikut merupakan tahapan dalam prepocessing text :

a. Transformation

Pada tahap transformasi, dilakukan case folding yaitu kata yang terdapat huruf kapital diubah menjadi non-kapital

```
# Fungsi untuk mengubah huruf besar menjadi huruf kecil pada teks komentar
def lowercase_text(text):
    return text.lower()

# Mengaplikasikan fungsi lowercase_text ke kolom komentar
df['case_folding'] = df['3/'].apply(lowercase_text)

df.head()
```

Gambar 7. *Source Code Transformation*

Tabel 3 dibawah merupakan hasil dari transformasi.

Tabel 3. Hasil Transformasi

Sebelum	Sesudah
Semangat para sandwich generation	semangat para sandwich generation

b. Tokenization

Tokenisasi yaitu pembagian kalimat menjadi beberapa kata atau bagian kecil yang dilakukan setelah proses transformasi.

Tokenizing

```
[ ]  import nltk
     from nltk.tokenize import word_tokenize
     def tokenizingText(ulasan):
       ulasan = word_tokenize(ulasan)
       return ulasan

     df['Tokenizing']= df['case_folding'].apply(tokenizingText)
```

Gambar 8. *Source Code Tokenizing*

Tabel dibawah ini merupakan hasil dari *tokenizing.*

Tabel 4. Hasil Tokenisasi

Sebelum	Sesudah
Semangat para sandwich generation	“semangat” “para” “sandwich” “generation”

c. Filtering

Proses filtering melibatkan pengurangan kata atau penyaringan kalimat dalam tweet. Dalam penelitian ini, proses *filtering* menggunakan proses *stopword*, normalisasi dan *stemming*.

1. **Stopword**

   Proses ini bertujuan untuk menghapus kata yang tidak penting tetapi tidak memengaruhi kalimat, contoh kata yang tidak relevan yaitu “yang” atau “ada” tahapan ini disebut dingen stopword.

```
#Disini tidak memerlukan kamus kembali karena sudah otomatis dan akan mengikuti data
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()

def stopwords_text(tokens):
  clened_tokens = []
  for token in tokens:
      if token not in stopwords:
          clened_tokens.append(token)
  return clened_tokens

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

Gambar 9. *Source Code Stopword*

Tabel 5 dibawah merupakan hasil dari stopword.

Tabel 5. Hasil Stopword Removal

Sebelum	Sesudah
Bukan sandwich generation dan punya orang tua ga menuntut untuk kerja kantoran.	['bukan', 'sandwich', 'generation', 'punya', 'orang', 'tua', 'menuntut', 'kerja', 'kantoran']

2. **Normalisasi**

Pada tahap ini dilakukan penormalisasian terhadap kata yang non formal sesuai dengan kamus *colloquial-indonesian-lexicon.*

```
[15] #Normalisasi-menormalisasikan kata yang non formal menjadi formal sesuai dengan kamus colloquial-indonesian-lexicon
     def normalization (Tweets):
       tweets_slang = pd.read_csv('colloquial-indonesian-lexicon.csv')
       dict_slang ={}
       for i in range(tweets_slang.shape[0]):
         dict_slang[tweets_slang["slang"][i]]=tweets_slang["formal"][i]

       drop_slang = []
       for teks in Tweets:
         normalisasi_teks = [dict_slang[word] if word in dict_slang.keys() else word for word in teks]
         drop_slang.append(normalisasi_teks)

       return drop_slang

     df['normalisasi'] = normalization(df['Tokenizing'])
```

Gambar 10. Source Code Normalisasi

Hasil normalisasi ditunjukkan pada tabel 4.

Tabel 6. Hasil normalisasi

Sebelum	Sesudah
salah satunya adalah dengan tidak menjadi sandwich generation ðŸ™□ðŸ□»	['salah', 'satunya', 'adalah', 'dengan', 'tidak', 'menjadi', 'sandwich', 'generation']

3. **Stemming**

Kata yang memiliki preposisi, konjungsi, dan pronominal dipisahkan dan diubah menjadi kata dasar melalui proses stemming, yang menghilangkan awalan atau akhiran.

```
[ ]  #factory = StemmerFactory()
     #stemmer = factory.create_stemmer()

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

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

     term_dict = {}

     for document in df['stopword_removal']:
         for term in document:
             if term not in term_dict:
                 term_dict[term] = stemmed_wrapper(term)

     # Check if 'K' is in term_dict, if not, assign a default value or apply stemming
     if 'K' not in term_dict:
         term_dict['K'] = 'default_value'  # Replace 'default_value' with an appropriate default value or term stemming

     def stemmingText(document):
         return [term_dict.get(term, 'default_value') for term in document]  # Replace 'default_value' with the same default value or term stemming

     df['Stemming'] = df['stopword_removal'].apply(stemmingText)
```

Gambar 11. *Source Code Stemming*

Hasil Stemming ditunjukkan pada tabel 7.

Tabel 7. Hasil Stemming

Sebelum	Sesudah
bukan sandwich generation dan selalu diberi kebebasan dalam mengambil apapun keputusan dalam hidup	['bukan', 'sandwich', 'generation', 'selalu', 'beri', 'bebas', 'ambil', 'apa', 'putus', 'hidup']

3.5 Klasifikasi

Setelah tahap prepocessing adalah tahap klasifikasi, tahap ini dari tahap pembobotan kata TF-IDF dan penerapan model SVM, dibawah ini merupakan tahapannya :

a. Pembobotan Kata TF-IDF

Pada penelitian dengan topik *Sandwich Generations*, terdapat jumlah kata sebanyak 7607 kata. Gambar 12 merupakan jumlah *feature words*.

```
No. of feature_words:  7607
```

Gambar 12. *Jumlah Kata*

Tahap berikutnya adalah tahap pembobotan kata dengan *Term Frequency*, hasil dari pembobotan kata TF diapat dilihat di gambar 13.

```
Tabel Term Frequency (TF):
                Fitur  TF Nilai
0         abai memang  0.169572
1           abai meni  0.429895
2         abang biasa  0.943728
3          abang kamu  0.204233
4         abang kerja  0.195909
...               ...       ...
7602        zero meni  0.404468
7603      zona enggak  0.425682
7604        zona moga  0.425682
7605      zona nyaman  0.170873
7606  zwestika daniel  0.186510
```

Gambar 13. *Hasil TF*

Setelah berhasil mendapatkan bobot TF, Langkah berikutnya adalah menghitung *Inverse Document Frequency* (IDF). Hasil IDF ditunjukkan pada gambar 14.

```
Tabel Inverse Document Frequency (IDF):
                Fitur  IDF Nilai
0         abai memang   7.041444
1           abai meni   6.635979
2         abang biasa   6.348297
3          abang kamu   7.041444
4         abang kerja   7.041444
...               ...        ...
7602        zero meni   6.635979
7603      zona enggak   6.635979
7604        zona moga   6.635979
7605      zona nyaman   7.041444
7606  zwestika daniel   7.041444
```

Gambar 14. *Hasil IDF*

Nilai dari TF-IDF dihitung dengan mengalikan nilai bobot setiap kata dari nilai TF keseluruhan dengan nilai IDF setiap kata. Gambar 5 dibawah merupakan hasil dari penghitungan nilai TF-IDF. .

```
Tabel TF-IDF:
     abai memang  abai meni  abang biasa  abang kamu  abang kerja  \
0            0.0        0.0          0.0         0.0          0.0
1            0.0        0.0          0.0         0.0          0.0
2            0.0        0.0          0.0         0.0          0.0
3            0.0        0.0          0.0         0.0          0.0
4            0.0        0.0          0.0         0.0          0.0
..           ...        ...          ...         ...          ...
835          0.0        0.0          0.0         0.0          0.0
836          0.0        0.0          0.0         0.0          0.0
837          0.0        0.0          0.0         0.0          0.0
838          0.0        0.0          0.0         0.0          0.0
839          0.0        0.0          0.0         0.0          0.0
```

Gambar 15. *Hasil TF-IDF*

b. SVM

Tahap selanjutnya setelah melakukan pembobotan kata dengan TD-IDF adalah implementasi model. Tahap pertama adalah import library yang berkaitan dengan evaluasi model. Ditunjukkan pada gambar 16.

sadasd

```
# mempersiapkan semua library, untuk kali ini saya menggunakna sklearn
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report
```

Gambar 16. *Library SVM*

Tahap selanjutnya adalah pemisahan data antara data training dan data testing. Gambar 17 merupakan hasil dari tahap *splitting data*.

```
X = df.final_clean_text
y = df.label #kolom target
```

```
# akan membagi data menjadi 80% untuk training data dan 20% untuk testing data
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.2, random_state =26105111)
```

Gambar 17. *Splitting Data*

Berdasarkan gambar 17 data yang digunakan yaitu 20% untuk data pengujian dan 80% untuk data pelatihan

Proses berikutnya adalah memasukkan training model, penelitian ini menggunakan model *Support Vector Machine* (SVM). Ditunjukkan pada gambar 18.

```
#fungsi yang digunakan untuk mengevaluasi model
def model_Evaluate(model):
  # nilai prediksi untuk data test
  y_pred = model.predict(X_test)
  # mencetak hasil evaluasi metriks dari dataset
  print(classification_report(y_test, y_pred))

  # menghitung dan menampilkan hasil dari Confusion matrix
  cf_matrix = confusion_matrix(y_test, y_pred)
  categories = ['Negative','Positive']
  group_names = ['True Neg','False Pos', 'False Neg','True Pos']
  group_percentages = ['{0:.2%}'.format(value) for value in cf_matrix.flatten() / np.sum(cf_matrix)]
  labels = [f'{v1}n{v2}' for v1, v2 in zip(group_names,group_percentages)]
  labels = np.asarray(labels).reshape(2,2)
  sns.heatmap(cf_matrix, annot = labels, cmap = 'Blues',fmt = '',
  xticklabels = categories, yticklabels = categories)
  plt.xlabel("Predicted values", fontdict = {'size':14}, labelpad = 10)
  plt.ylabel("Actual values" , fontdict = {'size':14}, labelpad = 10)
  plt.title ("Confusion Matrix", fontdict = {'size':18}, pad = 20)
```

```
# model SVM (Support Vector Machine)
SVCmodel = LinearSVC()
SVCmodel.fit(X_train, y_train)
model_Evaluate(SVCmodel)
y_pred2 = SVCmodel.predict(X_test)
```

Gambar 18. *Source code model*

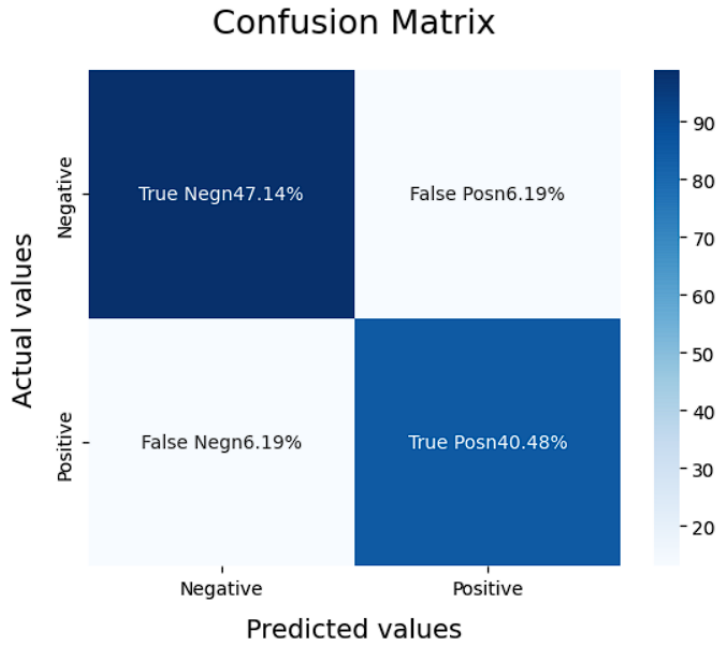
3.6 Evaluasi

Setelah tahap evaluasi matrix selesai, algoritma SVM digunakan pada data tweet untuk membagi 80% data pelatihan dan 20% data pengujian. Ini menghasilkan nilai akurasi sebesar 88%, gambar 19 dibawah merupakan hasil dari evaluasi.

	precision	recall	f1-score	support
0	0.88	0.88	0.88	112
1	0.87	0.87	0.87	98
accuracy			0.88	210
macro avg	0.88	0.88	0.88	210
weighted avg	0.88	0.88	0.88	210

Gambar 19. *Hasil evaluasi*

Selanjutnya, hasil heatmap confusion matrix dapat dilihat pada gambar 20.



Gambar 20. *heatmap*

Berdasarkan gambar 19, hasil pengujian menunjukkan nilai hasil Precision Negatif (0) 0.88, nilai recall 0.88, nilai f1-score 0.88, dan nilai support 112. Sedangkan Precision Positif (1) 0.87, nilai recall 0.87, nilai f1-score 0.87, dan nilai support 98. Sedangkan pada gambar 20, menunjukkan hasil dari True Negative (TN) 47.14%, False Negative (FN) 6.19%, True Positive (TP) 40.48%, dan False Positive (6.19%).

3.7 Visualisasi

Hasil penelitian dapat divisualisasikan berdasarkan hasil dari tahapan yang telah dilakukan. Gambar 20 menunjukkan visualisasi data dalam penelitian ini.

Gambar 20. *Dashboard Visualisasi*

Berdasarkan pada gambar 20, menunjukkan total sentiment per tahun yang menggambarkan sentiment terbanyak adaalah sentiment negatif di tiap tahunnya. Visualisasi Pie Chart total sentiment yang menunjukkan sentiment terbanyak adalah sentiment negative dengan total 43,2%, sentiment positif 38,8%, dan sentiment netral 17,9%. Wordcloud keseluruhan sentimen yang menunjukkan kata yang sering muncul pada keseluruhan sentiment. Dan Visualisasi data jumlah tweet pertahun yang menunjukkan jumlah tweet di tahun 2022 sebanyak 583 dan jumlah tweet di tahun 2023 sebanyak 771.

4. PENUTUP

Analisis sentiment yang dilakukan dengan menerapkan Algoritma Support Vector Machine (SVM) telah berhasil mengkategorikan persepsi warga mengenai topik *Sandwich Generation*. Berdasarkan proses pengumpulan data diperoleh data berjumlah 1384 data dari tahun 2022 – 2023, dan setelah proses labelling data terdapat 585 sentimen negative, 526 sentimen positif dan 243 sentimen netral. Berdasarkan data tersebut dapat disimpulkan tanggapan masyarakat terkait dengan topik *Sandwich Generation* cenderung negatif, banyak masyarakat yang mengeluh karena mereka terbebani dengan *sandwich generation*. Dalam peneletian ini, algoritma *Support Vector Machine* (SVM) menghasilkan nilai akurasi mencapai 88%, presisi dan nilai recall yang baik untuk kedua kelas.

Daftar Pustaka

Amrustian, M. A., Widayat, W., & Wirawan, A. M. (2022). Analisis Sentimen Evaluasi Terhadap Pengajaran Dosen di Perguruan Tinggi Menggunakan Metode LSTM. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *6*(1), 535. https://doi.org/10.30865/mib.v6i1.3527

Anggoro, D. A. (2020). Comparison of Accuracy Level of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) Algorithms in Predicting Heart Disease. *International Journal of Emerging Trends in Engineering Research*, *8*(5), 1689–1694. https://doi.org/10.30534/ijeter/2020/32852020

Arifin, N., Enri, U., & Sulistiyowati, N. (n.d.). *STRING (Satuan Tulisan Riset dan Inovasi Teknologi) PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DENGAN TF-IDF N-GRAM UNTUK TEXT CLASSIFICATION*.

Countries with most X/Twitter users 2023 | Statista. (2023). https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

Dewi, T. A., & Mailoa, E. (2023). PERBANDINGAN IMPLEMENTASI METODE SMOTE PADA ALGORITMA SUPPORT VECTOR MACHINE (SVM) DALAM ANALISIS SENTIMEN OPINI MASYARAKAT TENTANG MIXUE. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, *4*(3), 849–855. https://doi.org/10.35870/jimik.v4i3.289

Dyah Fritama, S., Raymond Ramadhan, Y., & Andayani Komara, M. (2023). Analisis Sentimen Review Produk Acne Spot Treatment di Female Daily Menggunakan Algoritma K-Nearest Neighbor. *Media Online)*, *4*(1), 134–143. https://doi.org/10.30865/klik.v4i1.1070

Ferdiana, R., Jatmiko, F., Purwanti, D. D., Sekar, A., Ayu, T., & Dicka, W. F. (2019). Dataset Indonesia untuk Analisis Sentimen. In *JNTETI* (Vol. 8, Issue 4).

Firdausi, A. J. (n.d.). *Perbandingan Algoritma Klasifikasi SVM dan Naive Bayes Dalam Analisis Sentimen Pembelajaran Daring di Masa Pandemi COVID-19 di Twitter*. https://t.co/uq2luzmnkh

Firmanto, A., & Sarno, R. (n.d.). *Prediction of Movie Sentiment based on Reviews and Score on Rotten Tomatoes using SentiWordnet*.

Heavy Burden of the 'Sandwich Generation' - Kompas.id. (n.d.). Retrieved October 1, 2023, from https://www.kompas.id/baca/english/2022/09/08/heavy-burden-of-the-sandwich-generation

Nurmila, I., Azizah, A., & Awaludin, R. (n.d.). *Hak Asuh Anak Akibat Perceraian dalam Pandangan Ulama Pedesaan*. https://riset-iaid.net/index.php/istinbath

Oktavia, D., & Ramadahan, Y. R. (2023a). Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM). *Media Online)*, *4*(1), 407–417. https://doi.org/10.30865/klik.v4i1.1040

Oktavia, D., & Ramadahan, Y. R. (2023b). Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM). *Media Online)*, *4*(1), 407–417. https://doi.org/10.30865/klik.v4i1.1040

Rofiqoh, U., Setya Perdana, R., & Fauzi, M. A. (2017). *Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features* (Vol. 1, Issue 12). http://j-ptiik.ub.ac.id

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

Satria Buana, R., Gata, W., Zevana, A., Widodo, P., Setiawan, H., & Hilyati, K. (2023). *) 2023 1,2,3,5 Fakultas Teknologi Informasi*. *7*(2). https://doi.org/10.35870/jti

Statistik Penduduk Lanjut Usia 2022. (n.d.).

Suharyono, K. P., & Hadiningrat, S. (n.d.-a). *DAMPAK GENERASI ROTI APIT TERHADAP PELUANG BONUS DEMOGRAFI DI INDONESIA (THE IMPACT OF THE SANDWICH GENERATION ON THE DEMOGRAPHY BONUS OPPORTUNITIES IN INDONESIA)*. www.suara.com/bisnis

Suharyono, K. P., & Hadiningrat, S. (n.d.-b). *DAMPAK GENERASI ROTI APIT TERHADAP PELUANG BONUS DEMOGRAFI DI INDONESIA (THE IMPACT OF THE SANDWICH GENERATION ON THE DEMOGRAPHY BONUS OPPORTUNITIES IN INDONESIA)*.

www.suara.com/bisnis
Wayan Rangga Pinastawa, I., & Afifah Arifuddin, N. (n.d.). *Komparasi Naïve Bayes dan Support Vector Machine dalam Klasifikasi Jenis Citrus Comparison of Naïve Bayes and Support Vector Machine in Citrus Species Classification* (Vol. 22, Issue 2).