



## Analisis Sentimen Pemilihan Calon Presiden 2024 Pada Sosial Media Twitter Menggunakan Metode *Support Vector Machine*

Yuliana Dwi Lestari<sup>1</sup>, Saiful Nur Budiman<sup>2</sup>, Sabitul Kirom<sup>3</sup>

Fakultas Teknologi Informasi, Universitas Islam Balitar<sup>1,2,3</sup>

|  |               |  |
|--|---------------|--|
|  |               | <b>Abstract</b>  |
| Received:  | 20 April 2025 | Pemilihan presiden merupakan bagian penting dari sistem politik di banyak negara yang memiliki sistem pemerintahan demokratis. Warga negara dapat memilih presiden untuk periode tertentu melalui proses ini. Para kandidat memberikan visi, program, dan rencana mereka kepada pemilih selama kampanye presiden. Twitter telah berkembang menjadi sumber data yang dapat digunakan untuk mengumpulkan pendapat dan pandangan masyarakat tentang calon presiden serta masalah yang terkait dengan pemilu. Melalui postingan, komentar, dan berbagi konten visual, pengguna twitter secara terbuka menyuarakan dukungan, kritik, atau ketidaksetujuan mereka terhadap kandidat dan aturan. Analisis sentimen yang dilakukan di twitter selama kampanye pemilihan dapat memberikan gambaran sentiment yang berbeda seperti sentiment positif, negatif, atau netral. Metode penelitian menggunakan <i>support vector machine</i> dengan sumber data melalui twitter berupa teks atau komentar. Dilakukan pengumpulan data dari rentang waktu tertentu sebesar 308 data. Pada tahap penelitian dilakukan <i>preprocessing text</i> untuk membersihkan data. Kemudian dilakukan dengan pengujian terhadap metode <i>Support Vector Machine</i> (SVM) dan dievaluasi dengan pengujian <i>confusion matrix</i> . Dari hasil pengujian untuk kasus pada penelitian ini didapatkan bahwa SVM dapat diimplementasikan dengan nilai akurasi mencapai 79%. |
| Revised:   | 27 April 2025 |  |
| Accepted:  | 01 Mei 2025   |  |
| <b>Keywords:</b>   |               | Analisis Sentimen, Calon Presiden 2024, Twitter, <i>Support Vector Machine</i>   |
| (*) Corresponding Author:  |               | <a href="mailto:yulianaadl@gmail.com">yulianaadl@gmail.com</a> , <a href="mailto:sync.saifulnb@gmail.com">sync.saifulnb@gmail.com</a> , <a href="mailto:sabitulkirom@gmail.com">sabitulkirom@gmail.com</a>   |
| <p><b>How to Cite:</b> Lestari, Y., Budiman, S., &amp; Kirom, S. (2025). Analisis Sentimen Pemilihan Calon Presiden 2024 Pada Sosial Media Twitter Menggunakan Metode Support Vector Machine. <i>Jurnal Ilmiah Wahana Pendidikan</i>, 11(5.A), 315-330. Retrieved from <a href="https://jurnal.peneliti.net/index.php/JIWP/article/view/11908">https://jurnal.peneliti.net/index.php/JIWP/article/view/11908</a></p> |               |  |

### PENDAHULUAN

Pemilihan Presiden merupakan bagian penting dari sistem politik di banyak negara yang menganut sistem pemerintahan demokratis. Proses ini memberikan warga negara hak suara untuk memilih seorang individu yang akan menjabat sebagai presiden untuk periode tertentu. Selama kampanye pemilihan presiden, kandidat-kandidat yang bersaing mempresentasikan visi, program, dan rencana- rencana mereka kepada pemilih. Warga negara yang memiliki hak suara kemudian memberikan suara mereka pada kandidat pilihan mereka. Calon presiden memengaruhi banyak aspek kehidupan negara, penelitian ini memilih calon presiden sebagai fokusnya. Kebijakan pemerintah, stabilitas politik, dan situasi ekonomi suatu negara dapat dipengaruhi oleh kepemimpinan seorang calon presiden. Selain itu, minat publik terhadap pemilihan presiden sangat besar. Pemangku kepentingan seperti masyarakat, media, dan lainnya aktif mengikuti dan memberikan pandangan mereka tentang calon presiden. Selain itu, kontroversi dan konflik sering mengganggu pemilihan presiden. Oleh karena itu, sangat penting untuk melihat

bagaimana opini publik berubah tentang calon presiden. Narasi dan persepsi calon presiden juga dibentuk oleh pengaruh media sosial.

Sosial media memainkan peran penting yang melibatkan masyarakat dalam proses pemilihan presiden. Hal ini disebabkan oleh fakta bahwa mereka membentuk politik yang dinamis. Twitter menjadi sumber data yang potensial untuk menggali pandangan dan opini masyarakat terkait calon presiden dan isu-isu pemilu. Melalui postingan, komentar, dan berbagi konten visual, pengguna Twitter secara terbuka menyuarakan dukungan, kritik, atau ketidaksetujuan mereka terhadap calon dan kebijakan. Dalam penelitian ini, pemilihan Twitter sebagai platform sosial media menurut **Damar Nugraha dan Puji Astuti, (2023)** Pertama, platform ini adalah salah satu platform sosial media yang paling populer dan digunakan secara luas di seluruh dunia yang memungkinkan untuk menganalisis sentimen tidak hafalan melalui konten visual seperti foto dan video pendek serta teks singkat seperti caption. Kedua, Twitter menarik pengguna dari berbagai usia, latar belakang, dan preferensi, sehingga hasil analisis sentimen dapat mencerminkan perspektif dari berbagai segmen masyarakat.

Analisis sentimen di Twitter dapat memberikan gambaran yang variatif dan *real-time* tentang suasana hati masyarakat seiring berjalannya kampanye pemilihan. Dengan menggunakan metode seperti *Support Vector Machine (SVM)*, dapat mengolah data teks dan visual dari Twitter untuk mengidentifikasi sentimen yang terkandung di dalamnya. Namun, kekuatan Twitter tidak hanya terbatas pada hal-hal positif. Banyak hal negatif yang juga dapat dengan mudah menyebar memicu tantangan baru dalam upaya memastikan informasi yang akurat dan sehat bagi demokrasi.

Analisis sentimen juga membantu dalam memahami perasaan, pandangan, dan preferensi masyarakat terhadap calon presiden. Ini memberikan gambaran yang lebih jelas tentang bagaimana pemilih merespon kampanye, pidato, dan kebijakan yang diajukan oleh para calon presiden. Dengan menganalisis sentimen, dapat diidentifikasi isu-isu yang paling memengaruhi pemilih. Ini membantu calon untuk fokus pada masalah-masalah yang benar-benar relevan dan penting bagi masyarakat. Dalam politik, untuk mengelola keanekaragaman pendapat masyarakat yang beragam, penggunaan berbagai bahasa yang digunakan untuk menyampaikan pendapat, dan dinamika perubahan sikap adalah semua tantangan yang dapat dipengaruhi oleh berbagai faktor politik. Oleh karena itu, pendekatan yang cermat dan efisien diperlukan untuk mengolah data yang sangat besar terkait dengan pemilihan presiden.

Metode-metode yang digunakan dalam analisis sentimen telah banyak digunakan dalam berbagai konteks, namun pengaplikasiannya dalam pemilihan umum presiden masih memerlukan penyelidikan lebih lanjut. Salah satu pendekatan yang bisa digunakan dalam penelitian ini adalah menggunakan Metode SVM, sebuah metode pembelajaran mesin yang telah terbukti efektif dalam mengatasi masalah klasifikasi, termasuk dalam analisis sentimen. SVM dapat memberikan keakuratan yang tinggi dalam mengidentifikasi polaritas sentimen untuk memahami dinamika dukungan dan kritik terhadap calon presiden, Menurut penelitian **Nurirwan Saputra dan Teguh Bharata Adji, (2015)** dalam hal studi kasus analisis sentimen terhadap kinerja presiden Bapak Jokowi, akurasi yang terbaik dalam penelitian ini adalah dengan dilakukan normalisasi dan *stemming* pada data sebesar 89,2655% menggunakan metode SVM, dan kemudian data yang dinormalisasi saja sebesar 88,7006%. Dengan menerapkan metode SVM ini untuk mengklasifikasikan komentar yang positif, negatif, dan netral, sehingga dapat diketahui

komentar yang benar –benar negatif, komentar yang benar –benar positif, dan komentar yang netral.

Dengan memfokuskan penelitian pada pemilihan presiden tahun 2024, dapat dirinci dan digambarkan perubahan sikap masyarakat yang mungkin terjadi seiring waktu, serta mengidentifikasi faktor-faktor yang mempengaruhi sentimen terhadap calon presiden tertentu. Berdasarkan penelitian sebelumnya yang telah dipaparkan di atas beberapa pertimbangan dalam menghadapi fenomena yang sedang terjadi diharapkan penelitian ini memberikan hasil yang objektif dan transparan dalam kebebasan berpendapat mengenai pemilu 2024 dengan metode yang digunakan yaitu SVM. Berdasarkan latar belakang di atas penulis memutuskan untuk melakukan penelitian dengan menggunakan judul “*Analisis Sentimen Pemilihan Calon Presiden 2024 pada Sosial Media Twitter Menggunakan Metode Support Vector Machine*”.

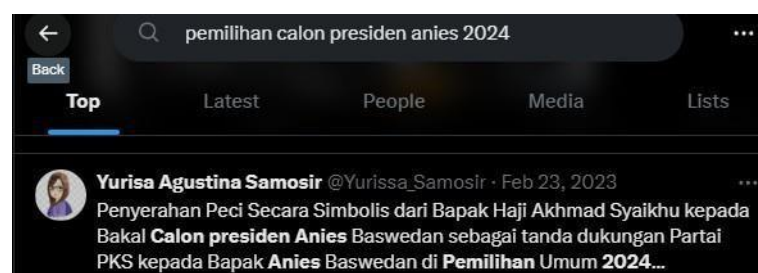
## METODE PENELITIAN

Waktu dan lokasi penelitian dengan judul Analisis Sentimen Pemilihan Calon Presiden 2024 pada Sosial Media Twitter Menggunakan Metode SVM yang akan dilakukan pada bulan November 2023 sampai bulan Juni 2024. Metode pengumpulan data dalam penelitian ini yaitu observasi dan studi. Dimulai dengan identifikasi masalah, perumusan masalah, penelitian literatur, penyusunan kuesioner, pengumpulan data, analisis data, dan akhirnya, kesimpulan dan rekomendasi.

## HASIL DAN PEMBAHASAN

### 1. Pengumpulan Data

Proses pengumpulan data merupakan langkah krusial yang mendasari analisis sentimen yang akan dilakukan. Langkah pertama adalah menemukan sumber data, seperti Twitter, dan berkonsentrasi pada postingan yang berkaitan dengan pemilihan presiden 2024. Kata kunci yang relevan, seperti nama calon presiden, partai politik, dan hashtag yang terkait dengan pemilihan presiden 2024, diidentifikasi untuk membantu dalam pengumpulan data seperti pada gambar 1 dibawah ini.



Gambar 1. Data twitter

API Twitter digunakan untuk mengumpulkan data postingan berdasarkan kata kunci. Data yang dikumpulkan termasuk user pengguna, postingan, hashtag, dan teks. Jika API Twitter tidak dapat memberikan akses yang memadai, *scraping* data dapat digunakan sebagai alternatif. Proses ini kemudian disimpan dalam format terstruktur di database, dengan informasi seperti *username* pengguna, postingan, hashtag, dan teks. Proses pengumpulan data ini menggunakan teknik *crawling* data untuk memperoleh informasi yang relevan dari media sosial Twitter.

```
# Crawl Data

filename      =      'zonasi.csv'
search_keyword = 'zonasi lang:id'
limit = 190

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s
"{search_keyword}" --tab "LATEST" -l {limit} --token
{twitter_auth_token}
```

**Gambar 2. Source Code Python Scraper**

Dalam penelitian ini telah mengumpulkan data sebanyak 308 data yang merupakan data mentah yang diperoleh dari web scraping. Hasil dari *crawling* data pada twitter dapat dilihat pada tabel 1.

**Tabel 1. Data Hasil Scraping**

| No | User            | Text  |
|----|-----------------|---|
| 1  | @BudiBukanIntel | Bowo itu kan gemes ke anies... Mas anies... Mas anies..   |
| 2  | @geloraco       | Janji Prabowo -1 poin Janji Anies +1 poin   |
| 3  | @irwndfry       | wkwk mas anies mas anies segitunya demi jabatan 11 dari 100 buat mas anies  |
| 4  | @CNNIndonesia   | Cukup sudahi kebodohan ini. Pak Anies dan Kaesang itu bumi dan langit kecerdasannya dlm memimpin. Tolonglah rakyat ini dah teramat sgt berjuang dg hidupnya trs mau makin dibuat gila dg menjadikan kang pisang ini sbg pe(mimpi)n wilayah yg g k |
| 5  | @momon_garuk44  | Kenapa rerata pembenci Anies pada serem ya ?  |

Data tabel 1 diatas terdiri dari kolom nama pengguna dan komentar. Nama pengguna berisi *username* pengguna twitter dan komentar berisi kata/pendapat dari pengguna yang dikumpulkan sesuai topik. Data di atas bukanlah keseluruhan data, melainkan hanya sampel. Data keseluruhan akan dilampirkan pada lampiran.

## 2. Hasil Pelabelan Data Frame Manual

Pelabelan dataset dilakukan secara manual terhadap pelabelan data untuk mengklasifikasikan setiap komentar ke dalam kategori sentimen: positif, negatif, atau netral. Pelabelan dilakukan dengan bantuan ahli Bahasa Indonesia.



**Gambar 3. Hasil Pelabelan Sentimen**

- Jika jumlah kata kunci positif dan negatif sama, komentar diberi label 0 (netral) maka didapat sentiment netral sebanyak 117 dengan persentasi 38%.
- Jika jumlah kata kunci negatif lebih banyak, komentar diberi label - 1 (negatif) maka didapat sentiment netral sebanyak 111 dengan persentasi 36%.
- Jika jumlah kata kunci positif lebih banyak, komentar diberi label 1 (positif) maka didapat sentiment netral sebanyak 80 dengan persentasi 26%.

### 3. Hasil Case Folding

Proses *case folding* dimulai dengan mengambil teks dari setiap postingan atau komentar yang ada dalam *dataset*. Kemudian, menggunakan fungsi bawaan dalam bahasa pemrograman yang digunakan dan semua huruf dalam teks diubah menjadi huruf kecil. Selanjutnya, teks yang telah diubah menjadi huruf kecil disimpan kembali ke dalam dataset yang terstruktur. Tahap ini akan dibantu dengan bantuan *library* RegEx.

```
# Mengubah teks menjadi huruf
kecil def casefoldingText(ulasan):
    ulasan = ulasan.lower()
    return ulasan
```

**Gambar 4. Source Code Case Folding**

Kode ini mendefinisikan sebuah fungsi yang bertujuan untuk mengubah seluruh teks menjadi huruf kecil, yang merupakan salah satu langkah praproses dalam analisis teks. Di dalam fungsi ini, teks ulasan diubah menjadi huruf kecil menggunakan metode `.lower`, yang merupakan metode bawaan *Python* untuk string. Langkah ini penting untuk memastikan konsistensi dalam analisis teks, karena perbedaan antara huruf besar dan kecil dapat mempengaruhi hasil analisis. Untuk hasil *case folding* dapat dilihat pada tabel dibawah ini.

**Tabel 2. Hasil Case Folding**

| Sebelum   | Sesudah                               |
|---|---------------------------------------|
| Bowo itu kan gemes ke anies... Mas anies... Mas anies.. | bowo itu kan gemes ke anies mas anies |

### 4. Hasil Tokenizing

Dari data yang sudah dibersihkan sebelumnya, akan dilanjutkan ke dalam tahap *tokenizing*. Proses ini memungkinkan analisis yang lebih mendalam dan terstruktur dengan membagi teks menjadi bagian yang lebih kecil.

```
# Fungsi tokenisasi
def tokenizingText(ulasan): ulasan
    = word_tokenize(ulasan) return
    ulasan

# Menerapkan tokenisasi pada data yang sudah melalui tahap
case folding
df['Tokenizing'] = df['CaseFolding'].apply(tokenizingText)
```

**Gambar 5. Source Code Tokenizing**

Kode ini bertujuan untuk melakukan praproses pada teks, fungsi `tokenizingText` digunakan untuk metokenisasi teks dengan metode `word_tokenize` yang digunakan untuk memecah teks menjadi kata-kata terpisah. Proses ini dilakukan pada kolom "`CaseFolding`" dataframe `df`, dan hasilnya disimpan di kolom baru yang disebut "`Tokenizing`" yang dimana hasil *tokenizing* dapat dilihat pada tabel dibawah ini.

**Tabel 3. Hasil Tokenizing**

| Sebelum   | Sesudah                               |
|---|---------------------------------------|
| Bowo itu kan gemes ke anies... Mas anies... Mas anies.. | bowo itu kan gemes ke anies mas anies |

## 5. Hasil Normalisasi

Tahap normalisasi dilakukan untuk pengubah penggunaan kata tidak baku menjadi baku sesuai dengan KBBI. Proses ini menggunakan file dataset *slangwords* yang berisi kata *slang* yang nantinya akan diubah menjadi baku. Tahap ini akan dibantu dengan *library* RegEx.

```
# Menerapkan konversi slang ke kata baku
df['Normalisasi'] =
df['Tokenizing'].apply(convertToSlangword)

# Menampilkan hasil
print(df[['Tokenizing', 'Normalisasi',
label_column]].head())
```

**Gambar 6. Source Code Normalisasi**

Kode ini melanjutkan proses praproses teks dengan fokus ke kata baku dan menampilkan hasil akhirnya. Pertama, dataframe *df* memiliki kolom *tokenizing* yang berisi kata-kata yang telah dipecah menjadi token. Dengan menggunakan fungsi *apply*, setiap daftar kata dalam kolom *tokenizing* diterapkan fungsi *vertToSlangword*. Fungsi ini menggantikan kata-kata slang dengan kata baku berdasarkan kamus yang telah diunggah dari file *slangwords.txt*. Hasil dari proses konversi ini disimpan dalam kolom baru yang diberi nama 'Normalisasi'. Setelah proses konversi selesai, kode ini menampilkan lima baris pertama dari dataframe untuk kolom normalisasi dan kolom label yang relevan.

**Tabel 4. Hasil Normalisasi**

| Sebelum  | Sesudah |
|--|---------|
| bowo, itu, kan g e m e s , ke, anies, mas, bowo, itu, kan gemes, ke, anies, mas, anies | anies   |

## 6. Hasil Stop Removal

Selanjutnya, pada langkah ini, kata-kata yang tidak berguna akan dihilangkan dengan menggunakan kamus *stopwords* dan ditambahkan ke dalam daftar *stopwords* yang tidak ada. Tahap ini dibantu dengan *library* NLTK.

```
# Menerapkan penghapusan stopwords pada data yang sudah
dinormalisasikan
df['Stopword_Removal'] =
df['Normalisasi'].apply(stopwordText)
```

**Gambar 7. Source Code Stop Removal**

Tujuan dari kode ini adalah untuk menghilangkan *stopwords* dari teks yang telah dinormalisasikan. Pertama, *stopwords* dikeluarkan dari daftar kata dengan menggunakan fungsi *stopwordText*. Daftar *stopword*, yang berisi *stopwords* bahasa Indonesia yang diambil dari NLTK dan ditambahkan secara manual, mengandung kata-kata yang tidak ada dalam fungsi ini. Selanjutnya, metode *apply* digunakan untuk menerapkan fungsi *stopwordText* pada kolom "Normalisasi" dataframe *df*. Teks yang sudah bebas dari *stopwords* disimpan dalam kolom baru yang disebut "*Stopword\_Removal*" sebagai hasil dari penggunaan fungsi ini. Karena *stopwords* yang tidak bermakna telah dihapus, teks yang tersisa hanya mengandung kata-kata yang dianggap penting untuk analisis lebih lanjut.

**Tabel 5. Hasil Stop Removal**

| Sebelum | Sesudah |
|---------|---------|
|---------|---------|

---

bowo, itu, kan gemes, ke, anies, mas,    bowo, gemes, anies, mas, anies  
 anies

---

## 7. Hasil Stemming

Proses selanjutnya yaitu *stemming* dengan mengurangi kata-kata ke bentuk dasarnya untuk membantu dalam menyederhanakan teks. Ini mengurangi variasi kata yang tidak perlu dan memungkinkan fokus pada arti utama teks. Tahap ini dibantu dengan *library* Sastrawi dan Swifter.

```

# Import library untuk stemming
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

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

# Fungsi untuk melakukan
stemming                                     def
stemmed_wrapper(term) :
    return stemmer.stem(term)

```

**Gambar 8. Source Code Stemming**

Tujuan kode ini adalah untuk melakukan stemming pada teks dengan menggunakan library Sastrawi, alat stemming bahasa Indonesia. Perpustakaan yang dibutuhkan diimpor, termasuk *StemmerFactory* dari Sastrawi. *Stemmer* dan *Swifter* digunakan untuk mempercepat implementasi fungsi pada dataframe. Selanjutnya, metode *create\_stemmer* digunakan untuk membuat objek *stemmer* dan menggunakan fungsi *stemmed\_wrapper* untuk melakukan *stemming* pada sebuah term atau kata. Fungsi ini menggunakan objek *stemmer* untuk mengubah kata yang diberikan menjadi bentuk aslinya. Mengurangi kata ke bentuk dasarnya adalah proses yang membantu menjaga teks konsisten dan seragam untuk analisis lebih lanjut.

**Tabel 6. Hasil Stemming**

| Sebelum  | Sesudah                        |
|--|--------------------------------|
| bowo, itu, kan gemes, ke, anies, mas,<br>anies | bowo, gemes, anies, mas, anies |

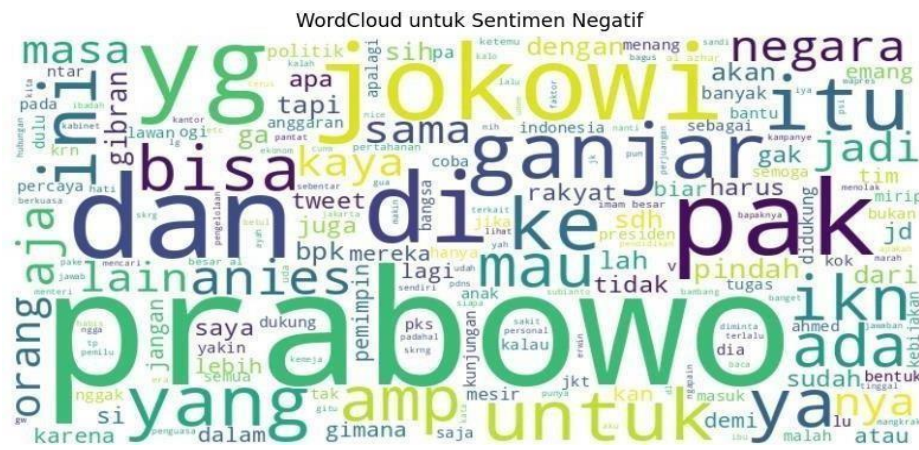
---

Setelah melakukan *text preprocessing* pada data, maka diperoleh visualisasi data yang merepresentasikan masing-masing kelas pada data. visualisasi data pada kelas netral dapat dilihat pada gambar berikut ini :



Pada kelas positif, frekuensi kemunculan kata ditunjukkan oleh gambar yaitu sebagai berikut :

1. Dominasi Kata "ganjar": Kata "ganjar" jelas mendominasi dengan frekuensi yang sangat tinggi, yang menunjukkan bahwa kata ini sangat relevan dan sering muncul dalam konteks positif dalam dataset. Ini bisa menandakan bahwa ganjar (sebagai calon presiden 2024 nomor urut 3) mungkin sering dikaitkan dengan pandangan positif, terutama jika dataset berkaitan dengan pemilu 2024.
2. Frekuensi Kata Lain: Kata-kata seperti "pak" dan "anies" juga memiliki frekuensi yang sering muncul. Hal ini bisa menunjukkan topik lain yang sering mendapat respons positif.



**Gambar 11. Visualisasi Data pada Kelas Negatif**

Pada kelas negatif, frekuensi kemunculan kata ditunjukkan oleh gambar yaitu sebagai berikut :

1. Dominasi Kata "prabowo": Kata "prabowo" jelas mendominasi dengan frekuensi yang sangat tinggi yang menunjukkan bahwa kata ini sangat relevan dan sering muncul dalam konteks negatif dalam dataset. Ini bisa menandakan bahwa prabowo (sebagai calon presiden nomor urut 2) mungkin sering dikaitkan dengan pandangan negatif, terutama jika dataset berkaitan dengan pemilu 2024.
2. Frekuensi Kata Lain: Kata-kata seperti "jokowi" juga memiliki frekuensi yang sering muncul. Hal ini bisa menunjukkan topik lain yang sering mendapat respons negatif.

## 8. Hasil Pembobotan Kata TF-IDF

Setelah melakukan pelabelan data, langkah selanjutnya adalah pembobotan kata menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*). TF-IDF adalah teknik pembobotan yang digunakan untuk mengevaluasi pentingnya sebuah kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya. Tahap ini dibantu dengan *library sklearn*.

```
# Membagi dataset menjadi training dan testing set
x_train, x_test, y_train, y_test =
model_selection.train_test_split(X, Y, test_size=0.2,
random state=42)
```

### Gambar 12. Source Code TF-IDF

Pada gambar 12 bertujuan untuk membagi dataset menjadi dua subset yaitu *training set* dan *testing set*. Hal ini dilakukan menggunakan fungsi *train\_test\_split* dari modul *model\_selection* dalam *library sklearn*. Variabel X berisi fitur-fitur dari dataset, sedangkan Y berisi label atau target yang ingin diprediksi. Dalam kode ini, fungsi *train\_test\_split* dipanggil dengan parameter *test\_size=0.2*, yang berarti 20% dari data akan digunakan sebagai *testing set* dan 80% sisanya akan digunakan sebagai *training set*. Parameter *random\_state=42* digunakan untuk memastikan bahwa pembagian dataset ini dapat direproduksi. Dengan menetapkan nilai *random state*, setiap kali kode dijalankan, data akan dibagi dengan cara yang sama, memastikan konsistensi dalam hasil evaluasi model. Output dari fungsi ini adalah empat variable yaitu *x\_train*, *x\_test*, *y\_train*, dan *y\_test*.

```

x_train shape: (246, 1204)
x_test shape: (62, 1204)
Fitur TF-IDF:
['aamiin' 'aaw' 'abah' 'acara' 'adem' 'adik' 'adil' 'aduh' 'ae' 'agaam']

```

### Gambar 13. Hasil TF-IDF

Setelah melakukan pelabelan data, langkah selanjutnya adalah pembobotan kata menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*). TF-IDF adalah teknik pembobotan yang digunakan untuk mengevaluasi pentingnya sebuah kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya. Tahap ini dibantu dengan *library sklearn*.

```

# Membagi dataset menjadi training dan testing set
x_train, x_test, y_train, y_test =
model_selection.train_test_split(X, Y, test_size=0.2,
random state=42)

```

### Gambar 14. Source Code TF-IDF

Pada gambar 14 bertujuan untuk membagi dataset menjadi dua subset yaitu *training set* dan *testing set*. Hal ini dilakukan menggunakan fungsi *train\_test\_split* dari modul *model\_selection* dalam *library sklearn*. Variabel X berisi fitur-fitur dari dataset, sedangkan Y berisi label atau target yang ingin diprediksi. Dalam kode ini, fungsi *train\_test\_split* dipanggil dengan parameter *test\_size=0.2*, yang berarti 20% dari data akan digunakan sebagai *testing set* dan 80% sisanya akan digunakan sebagai *training set*. Parameter *random\_state=42* digunakan untuk memastikan bahwa pembagian dataset ini dapat direproduksi. Dengan menetapkan nilai *random state*, setiap kali kode dijalankan, data akan dibagi dengan cara yang sama, memastikan konsistensi dalam hasil evaluasi model. Output dari fungsi ini adalah empat variable yaitu *x\_train*, *x\_test*, *y\_train*, dan *y\_test*.

```

x_train shape: (246, 1204)
x_test shape: (62, 1204)
Fitur TF-IDF:
['aamiin' 'aaw' 'abah' 'acara' 'adem' 'adik' 'adil' 'aduh' 'ae' 'agaam']

```

**Gambar 15. Hasil TF-IDF****9. Hasil Klasifikasi *Support Vector Machine* (SVM)**

Setelah melakukan pembobotan kata dengan TF-IDF, langkah berikutnya adalah mengklasifikasikan data menggunakan algoritma *Support Vector Machine* (SVM). SVM adalah salah satu algoritma pembelajaran mesin yang kuat untuk tugas klasifikasi dan regresi. Dalam kasus ini, peneliti akan menggunakan SVM untuk mengklasifikasikan sentimen pada komentar yang akan dibantu dengan

*library Scikit-Learn.*

```
# Import libraries untuk klasifikasi dan evaluasi
from sklearn.model_selection import cross_val_score
from sklearn import svm

# Menggunakan kernel RBF untuk SVM
svm_rbf = svm.SVC(kernel='rbf')
svm_rbf.fit(x_train, y_train)
```

**Gambar 16. Source Code SVM**

Pada gambar 16 kode ini mengimpor library yang diperlukan untuk klasifikasi dan evaluasi menggunakan *Support Vector Machine*. Library *cross\_val\_score* dari *sklearn.model\_selection* digunakan untuk melakukan *cross-validation*, yaitu metode untuk mengevaluasi kinerja model dengan membagi data menjadi beberapa subset, melatih model pada beberapa subset, dan menguji pada subset yang tersisa. Library SVM dari *sklearn* digunakan untuk mengimplementasikan model SVM. Dalam kode ini, model SVM dengan kernel RBF dibuat menggunakan SVM. Kernel RBF dipilih karena mampu menangani hubungan non-linear antara fitur-fitur dalam data. Setelah model dibuat, ia dilatih menggunakan data pelatihan dengan memanggil metode *fit*. Proses pelatihan ini memungkinkan model untuk belajar dari data pelatihan dan mengidentifikasi pola-pola yang berguna untuk melakukan klasifikasi pada data baru.

Proses klasifikasi menggunakan nilai data latih dan data uji sebesar 80% : 20% yang dilakukan percobaan sebanyak 5 kali. Metode SVM memiliki beberapa kernel dengan parameter yang mewakilinya. Salah satu metode yang digunakan untuk mengetahui kombinasi pengaturan terbaik (*hyperparameter tuning*) adalah *gridsearch*.

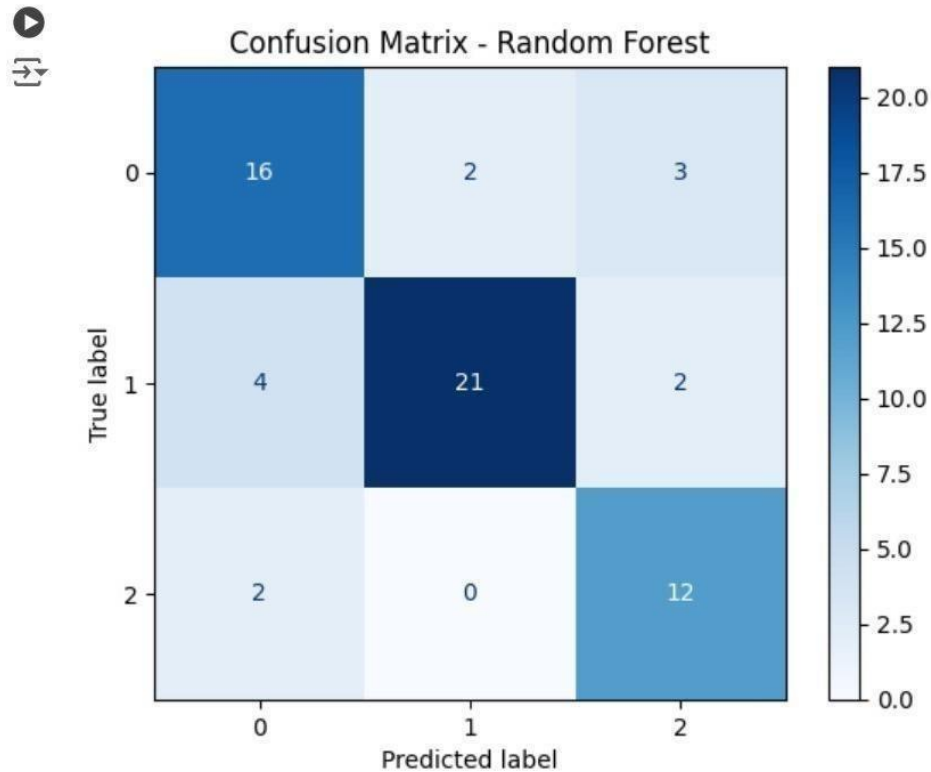
```
Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'linear'}
```

**Gambar 17. Hasil Evaluasi Model SVM**

Metode ini akan mengkombinasikan nilai dan mengevaluasi model setiap kombinasi. Pada penelitian ini akan menggunakan kernel linier dan rbf sebagai perbandingan dengan nilai C yaitu [0.1, 1, 10, 100] dan gamma yaitu [1, 0.1, 0.01, 0.001]. Nilai C akan mewakili kernel linier dan gamma akan khusus kernel rbf. Kemudian akan

dilakukan pengujian parameter sebanyak 5 kali menggunakan *cross validation*. Hasil dari *gridsearch* menunjukkan bahwa parameter terbaik adalah kernel linear dengan nilai C sebesar 0.1.

Berdasarkan hasil klasifikasi sentimen menggunakan metode SVM, diperoleh *confusion matrix* yang dapat dilihat pada gambar di bawah ini.



**Gambar 18. Visualisasi *Confusion Matrix***

1. Kelas 0 (netral)
  - a) TP (True Positive) : 16 : Jumlah data yang sebenarnya kelas netral dan diprediksi sebagai kelas netral
  - b) TN (True Negative): 35. Jumlah data yang sebenarnya bukan kelas netral dan diprediksi sebagai bukan kelas netral.
  - c) FP (False Positive): 6. Jumlah data yang sebenarnya bukan kelas netral tetapi diprediksi sebagai kelas netral.
  - d) FN (False Negative): 5. Jumlah data yang sebenarnya kelas netral tetapi diprediksi sebagai bukan kelas netral.
2. Kelas 1 (Positif)
  - a) TP (True Positive): 21. Jumlah data yang sebenarnya kelas positif dan diprediksi sebagai kelas positif.
  - b) TN (True Negative): 33. Jumlah data yang sebenarnya bukan kelas positif dan diprediksi sebagai bukan kelas positif.
  - c) FP (False Positive): 2. Jumlah data yang sebenarnya bukan kelas positif tetapi diprediksi sebagai kelas positif.
  - d) FN (False Negative): 6. Jumlah data yang sebenarnya kelas positif tetapi diprediksi sebagai bukan kelas positif.
3. Kelas 2 (Negatif)

- a) TP (True Positive): 12. Jumlah data yang sebenarnya kelas negatif dan diprediksi sebagai kelas negatif.
- b) TN (True Negative): 43 Jumlah data yang sebenarnya bukan kelas negatif dan diprediksi sebagai bukan kelas negatif .
- c) FP (False Positive): 5. Jumlah data yang sebenarnya bukan kelas negatif tetapi diprediksi sebagai kelas negatif .FN (False Negative): 2. Jumlah data yang sebenarnya kelas negatif tetapi diprediksi sebagai bukan kelas negatif

### **Pembahasan**

Berdasarkan penelitian yang telah dilakukan dapat diketahui bahwa proses pengumpulan data dengan menggunakan *scraping* diperoleh 308 data komentar pada twitter dengan menggunakan tagar pemilu 2024. Analisis sentimen pada penelitian ini menggunakan konsep *text mining* dengan memanfaatkan *google collab* dan bahasa pemrograman *python*. Tahap yang dilakukan antara lain pelabelan manual, *text preprocessing*, pembobotan kata dengan TF-IDF, dan klasifikasi dengan metode SVM serta pengujian dengan *confusion matrix*.

Dari hasil pengujian pada *confusion matrix* dapat diketahui bahwa semua data *sentiment* terklasifikasi dengan benar yang ditunjukkan dengan nilai TP (*True Positive*) sentimen netral adalah 16, nilai TP untuk sentimen positif adalah 21, dan nilai TP untuk sentimen negatif adalah 12. Sedangkan untuk nilai TN (*True Negative*) sentimen netral adalah 35, sentimen positif adalah 33, dan sentimen negatif adalah 43. FP (*False Positive*) sentimen netral adalah 6, sentimen positif adalah 2, dan sentimen negatif adalah 5, dan FN (*False Negative*) sentimen netral adalah 5, sentimen positif adalah 6, dan sentimen negatif adalah 2. Dari prediksi tersebut, mengakibatkan model memiliki nilai akurasi 79%

### **KESIMPULAN**

1. Penelitian ini menggunakan SVM untuk mengklasifikasikan sentimen pada komentar, yang dibantu oleh *library SciKit-Learn*. Proses klasifikasi akan menggunakan 80% nilai data uji dan latih, dengan 20% nilai yang dilakukan percobaan lima kali. Hasil dari prediksi tersebut menunjukkan bahwa model memiliki nilai akurasi 79% yang berarti model dapat memprediksi dengan benar sekitar 79% sentimen masyarakat terkait pemilihan calon presiden 2024.
2. Hasil pengujian pada *confusion matrix* menunjukkan bahwa semua data sentimen diklasifikasikan dengan benar. Ini ditunjukkan oleh nilai TP (*True Positive*) sentimen netral adalah 16, nilai TP untuk sentimen positif adalah 21, dan nilai TP untuk sentimen negatif adalah 12. Sementara nilai TN (*True Negative*) sentimen netral adalah 35, sentimen positif adalah 33, dan sentimen negatif adalah 43. Nilai FP (*False Positive*) sentimen netral adalah 6, dan sentimen positif yang salah adalah 1. Hasil ini menunjukkan bahwa model memiliki kinerja yang baik dalam mengklasifikasikan sentimen. Tingginya nilai TP dan TN serta rendahnya nilai FP menunjukkan bahwa model dapat membedakan dengan baik antara sampel-sampel dari berbagai sentimen mengurangi kesalahan dalam klasifikasi dan meningkatkan keakuratannya. Ini juga menunjukkan bahwa model lebih baik dalam memprediksi sentimen positif dibandingkan sentimen negatif.

### **DAFTAR PUSTAKA**

Agustina, W., Furqon, M. T., & Rahayudi, B. (2018). Implementasi Metode Support

- Vector Machine (SVM) Untuk Klasifikasi Rumah Layak Huni (Studi Kasus: Desa Kidal Kecamatan Tumpang Kabupaten Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(10), 3366–3372. <http://j-ptiik.ub.ac.id>
- Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *INTEGER: Journal of Information Technology*, 2(1).
- Darmawan, I. P. D. W., Pradnyana, G. A., & Pascima, I. B. N. (2023). Optimasi Parameter Support Vector Machine Dengan Algoritma Genetika Untuk Analisis Sentimen Pada Media Sosial Instagram. *SINTECH (Science and Information Technology) Journal*, 6(1), 58–67. <https://doi.org/10.31598/sintechjournal.v6i1.1245>
- Darwis, D., Pratiwi, E. S., & Pasaribu, A. F. O. (2020). Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia. *EduTic - Scientific Journal of Informatics Education*, 7(1), 1–11. <https://doi.org/10.21107/edutic.v7i1.8779>
- Deviyanto, A., & Wahyudi, M. D. R. (2018). Penerapan analisis sentimen pada pengguna twitter menggunakan metode K-Nearest Neighbor. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 3(1), 1-13.
- Fikri, M. I., Sabrila, T. S., & Azhar, Y. (2020). Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter. *SMATIKA Jurnal: STIKI Informatika Jurnal*, 10(2), 71–76.
- Giovani, A. P., Ardiansyah, A., Haryanti, T., Kurniawati, L., & Gata, W. (2020). Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi. *Jurnal Teknoinfo*, 14(2), 115-123.
- Hidayatullah, A. F., & Azhari, A. S. (2015, July). Analisis sentimen dan klasifikasi kategori terhadap tokoh publik pada twitter. In *Seminar Nasional Informatika (SEMNASIF) (Vol. 1, No. 1)*.
- Luqyana, W., Cholissodin, I., & Perdana, R. S. (2018). Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(11), 4704–4713. <http://j-ptiik.ub.ac.id>
- Maulana, M. A., Setyanto, A., & Kurniawan, M. P. (2018). Analisis Sentimen media Sosial Universitas Amikom Yogyakarta Sebagai Sarana Penyebaran Informasi Menggunakan Algoritma Klasifikasi Svm. *Seminar Nasional Teknologi Informasi Dan Multimedia 2018 Universitas Amikom Yogyakarta, 10 Februari 2018*, 7–12.
- Musfiroh, D., Khaira, U., Utomo, P. E. P., & Suratno, T. (2021). Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 1(1), 24–33. <https://doi.org/10.57152/malcom.v1i1.20>
- Novantirani, A., Sabariah, M. K., & Effendy, V. (2015). Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine. *E-Proceeding of Engineering*, 2(1), Agustina, W., Furqon, M. T., & Rahayudi, B. (2018). Implementasi Metode Support Vector Machine (SVM) Untuk Klasifikasi Rumah Layak Huni (Studi Kasus: Desa Kidal Kecamatan Tumpang Kabupaten Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(10), 3366–3372. <http://j-ptiik.ub.ac.id>
- Sarimole, F. M., & Septian, W. (2023). Analisis Sentimen Masyarakat Terhadap Isu *Penundaan* Pemilu 2024 Pada Twitter Dengan Metode Naive Bayes Dan Support

Vector Machine. *Jurnal Sains Dan Teknologi*, 5(3), 890–899.  
<http://ejournal.sisfokomtek.org/index.php/saintek/article/view/1359>

Utami, D. S., & Erfina, A. (2021, September). Analisis Sentimen Pinjaman Online di Twitter Menggunakan Algoritma Support Vector Machine (SVM). In *Prosiding Seminar Nasional Sistem Informasi dan Manajemen Informatika Universitas Nusa Putra* (Vol. 1, No. 01, pp. 299-30