



Analisis Sentimen Pengguna X Terkait Danantara Menggunakan Metode Support Vector Machine

Adie Fito Njurumana¹, Fajar Hariadi², Novem Berlian Uly³

^{1,2,3} Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Kristen Wira Wacana Sumba
adiefito@gmail.com, fajar@unkriswina.ac.id, novemuly@unkriswina.ac.id

Abstract

Daya Anagata Nusantara (DANANTARA) is an Indonesian government-owned investment body inaugurated by President Prabowo Subianto on 24 February 2025. With assets totaling USD900 billion (Rp14,000 trillion), it aims to optimize national investments, Badan Usaha Milik Negara (BUMN), and funding sources. Its establishment sparked diverse public responses on the social media platform X, with some perceiving the institution as immature and high-risk. This study analyzes public sentiment toward DANANTARA using 1,000 tweets collected during February–March 2025. The data were lexically labeled into Positive, Negative, and Neutral categories, then split into 700 training and 300 testing samples. A Support Vector Machine (SVM) model was employed for classification, yielding an accuracy of 68.7%. The Negative class showed the highest performance (recall 0.944; f1-score 0.740), followed by Positive (0.309; 0.436), and Neutral (0.152; 0.237). The macro-averaged precision, recall, and f1-score were 0.630, 0.468, and 0.471, respectively. Negative sentiment dominated (55.6%), followed by Positive (29.1%) and Neutral (15.3%), indicating a predominantly critical public response toward DANANTARA.

Keywords: *DANANTARA, Lexicon, Sentiment Analysis, Support Vector Machine*

Abstrak

Daya Anagata Nusantara (DANANTARA) adalah badan investasi milik pemerintah Indonesia yang diresmikan Presiden Prabowo Subianto pada 24 Februari 2025. Dengan aset sebesar USD900 miliar (Rp14.000 triliun), lembaga ini bertujuan mengoptimalkan investasi, BUMN, dan sumber dana nasional. Kehadirannya memicu respons publik beragam di media sosial X, sebagian menilai lembaga ini belum siap dan berisiko. Penelitian ini menganalisis sentimen publik terhadap DANANTARA menggunakan 1.000 tweet periode Februari–Maret 2025. Data dilabeli secara leksikal menjadi Positif, Negatif, dan Netral, lalu dibagi menjadi 700 data latih dan 300 data uji. Model Support Vector Machine digunakan untuk klasifikasi dan menghasilkan akurasi 68,7%. Kelas Negatif memiliki performa tertinggi (recall 0,944; f1-score 0,740), diikuti Positif (0,309; 0,436), dan Netral (0,152; 0,237). Rata-rata makro precision, recall, dan f1-score masing-masing 0,630; 0,468; dan 0,471. Sentimen Negatif mendominasi (55,6%), disusul Positif (29,1%) dan Netral (15,3%), mencerminkan respons publik yang cenderung kritis terhadap DANANTARA.

Kata kunci: Analisis Sentimen DANANTARA, Lexicon, *Support Vector Machine*.

1. Pendahuluan

Daya Anagata Nusantara (DANANTARA) merupakan badan investasi milik pemerintah Indonesia yang diresmikan oleh Presiden Prabowo Subianto pada 24 Februari 2025. Lembaga ini dibentuk untuk mengoptimalkan investasi nasional, meningkatkan kinerja Badan Usaha Milik Negara (BUMN), serta mengelola berbagai sumber pendanaan strategis. Modal awal DANANTARA berasal dari penyertaan modal negara dan sumber dana sah lainnya, sebagaimana diatur dalam Undang-Undang Nomor 1 Tahun 2025 tentang Perubahan Ketiga atas Undang-Undang Nomor 19

Tahun 2003, serta Peraturan Pemerintah Nomor 10 Tahun 2025 tentang Organisasi dan Tata Kelola Badan Pengelola Investasi DANANTARA [1]. Total aset yang dikelola mencapai USD900 miliar atau sekitar Rp 14.000 triliun.

Media sosial X, yang sebelumnya dikenal sebagai Twitter, menjadi salah satu platform utama bagi masyarakat Indonesia dalam mengekspresikan opini publik. Dengan jumlah pengguna mencapai 24,69 juta [2]. X kerap digunakan untuk menyampaikan pendapat, membagikan informasi, serta memperdebatkan isu-isu aktual. Salah satu topik yang menjadi sorotan pada akhir



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

Februari 2025 adalah peresmian DANANTARA, yang memicu diskusi luas dan beragam reaksi dari para pengguna. Meskipun telah memiliki dasar hukum yang kuat, sebagian masyarakat menilai lembaga ini belum dirancang secara matang dan berisiko mengalami kegagalan [3].

Memahami sentimen publik terhadap kebijakan pemerintah menjadi penting, karena opini masyarakat dapat menjadi indikator awal atas penerimaan atau penolakan terhadap suatu kebijakan [4]. Analisis sentimen melalui media sosial memungkinkan pemerintah merespons aspirasi publik secara *real-time*. Platform seperti X sering menjadi wadah ekspresi spontan dan autentik, sehingga reaksi negatif yang dibiarkan berlarut-larut berpotensi memicu polarisasi sosial dan penurunan kepercayaan terhadap pemerintah [5][6].

Untuk mengungkap pola dari opini publik secara sistematis, pendekatan data mining digunakan. Teknik ini membantu mengekstraksi pengetahuan tersembunyi dari kumpulan data besar melalui algoritma yang mampu mendeteksi tren, pola, atau anomali yang sulit terlihat secara manual [7].

Salah satu metode data mining yang banyak digunakan adalah klasifikasi, yang bertujuan menyusun model untuk mengelompokkan data berdasarkan karakteristik tertentu [8]. Beberapa algoritma populer dalam klasifikasi meliputi *decision tree*, *naïve Bayes*, *neural network*, *genetic algorithm*, *k-nearest neighbor*, dan *Support Vector Machine (SVM)* [8].

Penelitian ini menggunakan SVM karena kemampuannya mengklasifikasikan data berdimensi tinggi dan non-linier melalui fungsi kernel [9]. Selain itu, SVM telah terbukti efektif dalam tugas klasifikasi teks, terutama pada konteks analisis sentimen. Sebuah studi yang menganalisis opini publik terhadap isu pemindahan ibu kota mencatat akurasi sebesar 96,68%, dengan precision 95,82% dan recall 94,04%. Capaian ini menunjukkan bahwa SVM mampu mengenali pola sentimen secara akurat. Tingginya performa ini mencerminkan keunggulan SVM dalam menangani data berdimensi tinggi dan kompleksitas konteks bahasa alami yang kerap muncul dalam media sosial. [11].

Pada Penelitian ini akan dilakukan klasifikasi terkait dengan respons pengguna media sosial X terkait dengan Kehadiran DANANTARA yang diluncurkan pemerintah dengan menggunakan metode klasifikasi *support vector machine* dan metode pelabelan *lexicon*. Data yang akan digunakan adalah kumpulan *tweets* terkait dengan DANANTARA mulai dari Februari 2025 sampai dengan Maret 2025. Data yang ada kemudian di klasifikasi ke dalam 3 kelas sentimen yaitu negatif, netral, dan positif dan dihitung akurasi, presisi, *Recall*,

dan *F1-Score* dan *macro-average* untuk mengetahui seberapa baik model yang dibuat.

2. Metode Penelitian

Jenis dan Sumber Data

Dalam penelitian ini, data yang digunakan berasal dari *tweets* yang ada pada Aplikasi X. Jenis penelitian adalah penelitian kuantitatif.

Alur Penelitian

Tahapan pada penelitian ini dimulai dari Mengumpulkan data, *Pre-processing* Data, Mengelompokkan Data, Klasifikasi SVM, dan Penyajian dapat dilihat pada Gambar 1.



Gambar 1 Alur Penelitian

Di bawah ini adalah penjelasan mengenai alur pada penelitian ini

1. Mengumpulkan Data

Pada Proses pertama ini dilakukan pengumpulan data dan informasi yang bisa digunakan dalam penelitian ini. Sumber data *Crawling* sendiri adalah media sosial X data yang diambil adalah tweets terkait dengan menggunakan kata kunci “DANANTARA” yang di posting pengguna media sosial X, dari bulan Februari 2025 sampai dengan bulan Maret 2025 dengan jumlah 1000 data, dan di *crawling* dengan menggunakan platform google colab.

2. Pre-processing Data

Pada tahapan ini data-data yang sudah dikumpulkan akan masuk ke dalam proses pembersihan data menggunakan bahasa Python sebelum data diolah data harus di Transformasi, data yang semula tidak terstruktur, menjadi data yang terorganisir dengan baik sehingga proses analisis menjadi lebih efisien. Beberapa tahap dalam data *pre-processing* yang digunakan adalah *letter casing*, *cleaning tokenize*, *filtering*, dan *Stemming*.

- Letter casing*: Proses mengubah seluruh teks dalam *tweet* menjadi format standar, yaitu huruf kecil (*lowercase*)
- Cleaning*: adalah proses penghapusan elemen-elemen tertentu dalam *tweet*, seperti *link URL (Uniform Resource Locator)*, *username*, *RT (Retweet)*, karakter *HTML*, *whitespace* dan *hashtag*.
- Tokenizing*: Proses mengubah *tweet* menjadi token, di mana token adalah kata-kata yang dipisahkan oleh spasi dalam teks.
- Stopword removal/filtering*: Proses memilih kata-kata yang relevan dan menghapus kata-kata yang tidak memiliki arti penting dalam *tweet (stopwords)*, seperti kata penghubung seperti "dan", "atau", "kemudian", dan

sebagainya, atau kata-kata yang tidak berpengaruh pada proses klasifikasi.

- e. *Stemming*: Pada tahap ini, dilakukan proses pengurangan kata yang memiliki awalan atau akhiran menjadi bentuk dasarnya.

3. Mengelompokkan Data

Data yang sudah melalui proses pembersihan atau *pre-processing* kemudian akan dibagi menjadi dua jenis data yaitu data latih dan juga data uji, Pada tahap ini juga dilakukan pelabelan pada data menggunakan metode *Lexicon*. Setiap kata yang ada pada kalimat akan diperiksa satu persatu di kamus dan jika kata ada di dalam kamus maka kata akan diberikan skor jika tidak maka skor kata secara *default* adalah 0.

$$StSc(x) \begin{cases} 1 & \text{kata positif} > \text{Negatif} \\ -1 & \text{kata positif} < \text{Negatif} \\ 0 & \text{selainnya} \end{cases} \dots\dots\dots(1)$$

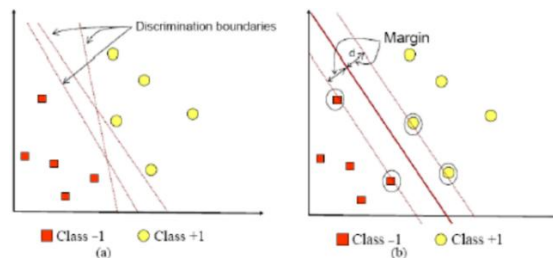
Pada Persamaan (1) ditunjukkan cara kerja metode *lexicon*. Di dalam kamus *lexicon* telah tersedia kumpulan kata-kata yang mewakili masing-masing sentimen, baik positif maupun negatif. Selanjutnya, setiap kata dalam teks dibandingkan dengan kamus tersebut dan diberikan skor sesuai polaritasnya [10]. Skor-skor tersebut kemudian dijumlahkan untuk menentukan skor sentimen keseluruhan dari teks

Setelah melakukan pelabelan dataset kemudian 1000 data akan dibagi menjadi data latih dan data uji dengan perbandingan 70% data latih dan 30% data uji

- a. Data Latih adalah data yang akan digunakan sebagai alat untuk basis pengetahuan model.
- b. Data Uji adalah data yang akan digunakan untuk menguji model yang sudah dilatih menggunakan data latih.

4. Klasifikasi SVM

Proses Klasifikasi menggunakan Support Vector Machine (SVM) pada penelitian ini diterapkan dengan mencari garis pemisah optimal (*hyperplane*) yang mampu membedakan tweet berdasarkan label sentimen, yaitu positif, negatif, dan netral. *Hyperplane* ini ditentukan melalui proses pelatihan, dengan cara mengoptimalkan pembobotan vektor fitur dan margin antar kelas. Pembobotan ini menghasilkan sebuah model yang merepresentasikan pola distribusi data. Pola yang terbentuk dari data latih digunakan untuk mengklasifikasikan data uji, dengan cara memetakan data ke posisi relatif terhadap *hyperplane*, sehingga dapat diberikan label sentimen yang sesuai.



Gambar 2. Proses Kerja SVM

Pada Gambar 1. digambarkan konsep klasifikasi menggunakan SVM. Pada bagian grafik kiri, terdapat beberapa titik data dengan lingkaran mewakili kelas -1 dan kotak merepresentasikan kelas +1. Gambar ini juga menunjukkan berbagai kemungkinan *hyperplane* yang dapat memisahkan kumpulan data tersebut. Bagian grafik kanan menampilkan *hyperplane* optimal yang memiliki margin terbesar. Proses penentuan *hyperplane* ini dilakukan dengan memaksimalkan jarak margin terhadap data paling dekat dari setiap kelas.

5. Penyajian Hasil

Hasil Dari Proses Klasifikasi akan di tunjukan melalui Diagram yang akan menunjukkan persentase dari ketiga label sentimen baik itu sentimen negatif, netral, maupun positif. Hasil ini akan menunjukkan label mana yang paling dominan dari total *dataset* dan perhitungan akurasi, presisi, *Recall*, *F1-Score* dan *macro-average* dari model Akan menunjukan seberapa baik akurasi SVM dalam melakukan klasifikasi sentimen terkait topik DANANTARA.

3. Hasil dan Pembahasan

Pengumpulan Data

Proses pengumpulan data, data didapatkan dari hasil *crawling tweet X* terkait topik DANANTARA dari rentang bulan februari 2025 sampai dengan bulan Maret 2025. data yang diperoleh berjumlah 1000 data, proses *crawling tweet* menggunakan kode *pyton* dan *node.js* dengan menggunakan *tool opensource tweet harvest*. Pada gambar 2 adalah kode program yang digunakan.

```
twitter_auth_token = 'jdsdsuud3443s776d6s7ds8ds'

!pip install pandas
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key \
| sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
!NODE_MAJOR=20 && \
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] \
https://deb.nodesource.com/node_\\$NODE_MAJOR.x \
nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list

!sudo apt-get update
!sudo apt-get install nodejs -y
!node -v
filename = 'ANTM(Oktober).csv'
search_keyword = 'ANTM since:2024-10-30 until:2024-11-01 lang:id'
limit = 300

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

import pandas as pd
file_path = f"tweets-data/{filename}"
df = pd.read_csv(file_path, delimiter=",")
display(df)
num_tweets = len(df)
print(f"Jumlah tweet dalam dataframe adalah {num_tweets}.")
```

Gambar 3. Kode Crawling Data

Pada kode terdapat variabel yang digunakan untuk menyimpan token autentikasi *Twitter*. Token ini wajib diganti dengan token pribadi akun pengguna agar proses pengambilan data dapat diakses, potongan kode tersebut dapat di lihat pada Gambar 3.

```
twitter_auth_token = 'jdsdsuud3443s776d6s7ds8ds'
```

Gambar 4. Bagian Kode Variabel Token

Selanjutnya, dilakukan instalasi pustaka *Python* yang dibutuhkan, yaitu *pandas*, yang digunakan untuk membaca dan mengolah data hasil *crawling* dalam bentuk *DataFrame*, potongan kode tersebut dapat dilihat pada Gambar 4.

```
!pip install pandas
```

Gambar 5. Instalasi Pustaka Python Pandas

karena *tweet-harvest* dibangun menggunakan *Node.js*, maka dilakukan serangkaian instalasi *Node.js* terlebih dahulu. Proses ini melibatkan pembaruan repositori sistem (*apt-get update*), penginstalan dependensi sistem (*ca-certificates*, *curl*, dan *gnupg*), serta penambahan kunci GPG dari *NodeSource* dan penambahan daftar repositori *Node.js* versi 20, potongan kode tersebut dilihat pada Gambar 6.

```
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key \
| sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
!NODE_MAJOR=20 && \
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] \
https://deb.nodesource.com/node_\\$NODE_MAJOR.x \
nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list
!sudo apt-get update
```

Gambar 6. Kode Instalasi Dependensi Node.js

Setelah proses instalasi selesai, akan dilakukan pengecekan versi *Node.js* untuk memastikan bahwa instalasi berhasil potongan kode tersebut dapat dilihat pada Gambar 7.:

```
!node -v
```

Gambar 7. Pengecekan Versi Node.js

Kemudian, bagian berikutnya bertujuan melakukan proses *crawling* data dari *Twitter*. Tiga variabel disiapkan terlebih dahulu, yaitu *filename* untuk menentukan nama *file output CSV*, *search_keyword* untuk menentukan kata kunci pencarian *tweet* beserta filter tanggal dan bahasa, serta limit untuk menetapkan jumlah maksimal *tweet* yang akan diambil. Kode tersebut bisa dilihat pada Gambar 8:

```
filename = 'ANTM(Oktober).csv'
search_keyword = 'ANTM since:2024-10-30 until:2024-11-01 lang:id'
limit = 300
```

Gambar 8. Inisialisasi Variabel Data Crawling

Proses *crawling* akan dilakukan menggunakan *tweet-harvest* melalui perintah *npx*. Perintah ini mengeksekusi modul *tweet-harvest* versi 2.6.1 dengan parameter *output file (-o)*, string pencarian (*-s*), tab yang diakses ("*LATEST*"), limit data (*-l*), dan token autentikasi (*--token*), Kode tersebut dapat dilihat pada Gambar 9.

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

Gambar 9. Menjalankan Proses Crawling

Setelah data berhasil disimpan ke dalam file *CSV*, data tersebut dibaca kembali menggunakan *pandas*. Pertama-tama, ditentukan lokasi *file CSV* melalui *file_path*, kemudian *file* dibaca menjadi *DataFrame*, Kode tersebut dapat dilihat pada Gambar 10.

```
import pandas as pd
file_path = f"tweets-data/{filename}"
df = pd.read_csv(file_path, delimiter=",")
```

Gambar 10. Membaca Data Hasil Crawling

Data yang telah dimuat dalam *DataFrame* kemudian ditampilkan menggunakan *display(df)*, dan jumlah total *tweet* yang berhasil dimuat dihitung dan ditampilkan ke layar menggunakan *print()*. Kode tersebut dapat dilihat pada Gambar 11.

```
display(df)
num_tweets = len(df)
print(f"Jumlah tweet dalam dataframe adalah {num_tweets}.")
```

Gambar 11. Menampilkan Data dan Jumlah Data

Data tweet yang berhasil dikumpulkan dapat dilihat pada Tabel 1.

Tabel 1. Data Hasil Crawling

No.	Tweet
1.	BUMN Holding Industri Pertambangan Indonesia MIND ID meyakini pembentukan Badan Pengelola Investasi Daya Anagata Nusantara (Danantara) merupakan langkah strategis untuk memperkuat investasi hilirisasi dan industrialisasi nasional. https://t.co/Q7IbwuKjcG
2.	@FatkhRoza97280 @bitxt @SBYudhoyono Danantara adalah ladang KORUPSI untuk para penjahat politik!!! TIDAK ada bukti bahwa Danantara dapat diawasi oleh Rakyat Indonesia dari kejahatan politik!!!
3.	@inilahdotcom BUMN yg begini yg disertakan ke DANANTARA?
4.	Mari Kita dukung Pemerintah Indonesia yang membantu BUMN dengan hadirnya Danantara #DanantaraBangunNegeri https://t.co/V8JBgPybpZ
5.	BUMN akan semakin hebat jika bergabung dengan Danantara #DanantaraBangunNegeri https://t.co/Bt3k51kZHR
...	...
996.	@Strategi_Bisnis Tidak ada yang rugi Negara lain iri dengan kita NdasMu Masih ada danantara nanti kita suntik pakai itu Duit nya mau saham nya di bilang Judi Oke gas oke gass All in Tabrak
997.	@tham878 Fakta yang tidak bisa di bantah orang asing menjabat di Danantara kalau pemakan bangkai pasti tidak akan bersuara
998.	Dikawatirkan pgeluaran Danantara utk biaya gaji direksi komisaris pakar & pengawas dari mantan pjabat DN/LN akan mengurus dana pmeegang saham sdgkan proyek yg didanainya belum tentu menguntungkan kecuali direksinya profesional dipilih dari BUMN perbankan yg jujur & berakhlak mulia.
999.	@rongguuur Maafkan diri ini bila belum setransparan pembuatan RUU TNI Masih banyak salah seperti pendirian Danantara Acap berkata kasar dan mengucapkan Ndasmu Teriring salam dn kehangatan yng dpat memenuhi raga seperti Makan Bergizi Gratis Oke Gas menuju kemenangan mohon maaf lahir batin
1000.	@aewin86 Halah bacot. Danantara yg jelas2 ad org ASENG eks KORUPTOR aja lu mingkem wkww. Cuih

Pre-Processing Data

Dalam tahapan ini data kemudian akan dipersiapkan sebelum digunakan dalam proses klasifikasi dengan metode SVM adapun proses pre-processing yang digunakan dalam penelitian ini adalah *letter casing*, *cleaning tokenize*, *filtering*, dan *Stemming*.

a. *Letter casing*: Pada Proses ini seluruh text akan di transformasi ke huruf kecil (*lowercase*). Di bawah

ini adalah rangkaian kode dan hasil implementasi kode pada dataset.

Sebelum memulai proses pre-processing pada awalnya data dibaca dari *file* CSV dataset dan di masukan ke variabel bernama *review* dengan menggunakan pandas dengan kode di Gambar 12.

```
import pandas as pd
df = pd.read_csv \
("/kaggle/input/hasil-pre-process/hasil_pre_process.csv")
```

Gambar 12. Membaca File CSV

Kemudian ditambahkan modul-modul *pre-processing* teks atau NLP (*Natural Language Processing*), khususnya untuk teks berbahasa Indonesia, *string* digunakan untuk memanipulasi atau menghapus tanda baca (*string punctuation* dan untuk ekspresi reguler, misalnya untuk menghapus pola-pola seperti tautan, mention, angka, simbol, dll. Kode tersebut dapat dilihat pada Gambar 13.

```
import sys
sys.path.append("/kaggle/input/nlp-id/nlp-id-master/nlp_id")
import string
import re
```

Gambar 13. Mempersiapkan Modul NLP-Id

Setelah sudah mempersiapkan modul yang diperlukan untuk pre-processing maka akan di tambahkan fungsi untuk proses *Letter Casing* potongan kodenya dapat di lihat pada Gambar 14.

```
def my_tokenizer(doc):
    doc = doc.lower()
    return doc
```

Gambar 14. Kode Fungsi Letter Casing

Fungsi ini berfungsi untuk mengubah teks menjadi huruf kecil (*Letter Casing*) setelah itu fungsi tersebut akan di terapkan ke seluruh data teks dalam kolom data *tweet* dari *DataFrame* *review* dengan menggunakan kode pada Gambar 15.

```
review['preprocessing'] = review['full_text']\
.apply(my_tokenizer)
```

Gambar 15. Implementasi Fungsi Letter Casing

hasil implementasi *Letter casing* pada dataset dapat dilihat pada Tabel 2.

Tabel 2. Hasil Letter Casing

No.	Sesudah <i>Letter Casing</i>
1.	bumn holding industri pertambangan indonesia mind id meyakini pembentukan badan pengelola investasi daya anagata nusantara (danantara) merupakan langkah

	strategis untuk memperkuat investasi hilirisasi dan industrialisasi nasional. https://t.co/q7ibwukjcg
2.	@fatkhuroza97280 @bitxt @sbyudhoyono danantara adalah ladang korupsi untuk para penjahat politik!!! tidak ada bukti bahwa danantara dapat diawasi oleh rakyat indonesia dari kejahatan politik!!!
3.	@inilahdotcom bumn yg begini yg disertakan ke danantara?
4.	mari kita dukung pemerintah indonesia yang membantu bumn dengan hadirnya danantara #danantarabangunnegeer https://t.co/v8jbgpybpz
5.	bumn akan semakin hebat jika bergabung dengan danantara #danantarabangunnegeer https://t.co/bt3k51kzhr
...	...
996.	@strategi_bisnis tidak ada yang rugi negara lain iri dengan kita ndasmu masih ada danantara nanti kita suntik pakai itu duit nya mau saham nya di bilang judi oke gas oke gass all in tabrak
997.	@tham878 fakta yang tidak bisa di bantah orang asing menjabat di danantara kalau pemakan bangkai pasti tidak akan bersuara
998.	dikawatirkan pgeluaran danantara utk biaya gaji direksi komisaris pakar & pengawas dari mantan pjabat dn/ln akan menguras dana pmegang saham sdgkan proyek yg didananya belum tentu menguntungkan kecuali direksinya profesional dipilih dari bumn perbankan yg jujur&berakhlak mulia.
999.	@rongguur maafkan diri ini bila belum setransparan pembuatan ruu tni masih banyak salah seperti pendirian danantara acap berkata kasar dan mengucapkan ndasmu teriring salam dn kehangatan yng dpat memenuhi raga seperti makan bergizi gratis oke gas menuju kemenangan mohon maaf lahir batin
1000.	@aewin86 halah bacot. danantara yg jelas2 ad org aseng eks koruptor aja lu mingkem wkkww. cuih

b. *Cleaning*: Pada proses ini elemen-elemen tertentu dalam *tweet*, seperti *link URL*, *username*, *Retweet*, karakter *HTML*, *whitespace* dan *hashtag* akan dihilangkan. di bawah ini adalah Kode dan hasil implementasinya:

Untuk melakukan *cleaning* pada dataset terlebih dahulu dipersiapkan fungsinya kode fungsi *cleaning* dapat dilihat pada Gambar 16.

```
def my_tokenizer(doc):
    doc = re.sub(r'@[A-Za-z0-9]+', '', doc)
    doc = re.sub(r'#[A-Za-z0-9]+', '', doc)
    doc = re.sub(r'RT|\\s', '', doc)
    doc = re.sub(r'http\\S+', '', doc)
    doc = re.sub(r'[0-9]+', '', doc)
    doc = re.sub(r'\\.\\1+', r'\\1', doc)
    doc = re.sub(r'[\\?\\.\\!]+(?=[\\?\\.\\!])', '', doc)
    doc = re.sub(r'[a-zA-Z]', '', doc)
    doc = re.sub(r'\\b(\\w+)(\\1b)+', r'\\1', doc)
    doc = doc.replace('\\n', '')
    doc = doc.translate(str.maketrans('', '', string.punctuation))
    doc = doc.strip(' ')
    #Punctuation Removal+Menghapus Angka
    doc = doc.translate(str.maketrans('', '', string.punctuation + string.digits))
    #Whitespace Removal
    doc = doc.strip()
    return doc
```

Gambar 16. Kode Fungsi *Cleaning*

Fungsi ini akan membersihkan teks dari berbagai unsur yang dianggap tidak relevan atau mengganggu analisis, seperti *mention*, *hashtag*, tautan, angka, karakter berulang, tanda baca, hingga duplikasi kata. Hasil implementasi *Cleaning* pada dataset dapat dilihat pada Tabel 3.

Tabel 3. Hasil *Cleaning*

No.	Sesudah <i>Cleaning</i>
1.	bumn holding industri pertambangan indonesia mind id meyakini pembentukan badan pengelola investasi daya anagata nusantara danantara merupakan langkah strategis untuk memperkuat investasi hilirisasi dan industrialisasi nasional
2.	danantara adalah ladang korupsi untuk para penjahat politik tidak ada bukti bahwa danantara dapat diawasi oleh rakyat indonesia dari kejahatan politik
3.	bumn yang begini yang disertakan ke danantara
4.	mari kita dukung pemerintah indonesia yang membantu bumn dengan hadirnya danantara
5.	bumn akan semakin hebat jika bergabung dengan danantara
...	...
996.	bisnis tidak ada yang rugi negara lain iri dengan kita ndasmu masih ada danantara nanti kita suntik pakai itu duit nya mau saham nya di bilang judi oke gas oke gass all in tabrak
997.	fakta yang tidak bisa di bantah orang asing menjabat di danantara kalau pemakan bangkai pasti tidak akan bersuara
998.	dikawatirkan pgeluaran danantara untuk biaya gaji direksi komisaris pakar amp pengawas dari mantan pjabat dn ln akan menguras dana pmegang saham sdgkan proyek yang didananya belum tentu menguntungkan kecuali direksinya profesional dipilih dari bumn perbankan yang jujur amp berakhlak mulia
999.	maafkan diri ini bila belum setransparan pembuatan ruu tni masih banyak salah seperti pendirian danantara acap berkata kasar dan mengucapkan ndasmu teriring salam dn kehangatan yng dpat memenuhi raga seperti makan bergizi gratis oke gas menuju kemenangan mohon maaf lahir batin
1000.	halah banyak bicara danantara yang jelas ada org aseng eks koruptor saja kamu mingkem wkkww cuih

c. *Tokenizing*: Pada proses ini *tweet* akan ditrasformasi menjadi token. Untuk melakukan *Tokenizing* pada dataset terlebih dahulu dilakukan inialisasi komponen *tokenizer* dari modul *nlp-id*. Kode dapat dilihat pada Gambar 17.

```
tokenizer = Tokenizer()
```

Gambar 17. Inialisasi Komponen *Tokenizing*

Setelah melakukan inialisasi objek *tokenizer* kemudian dibuat fungsi untuk memanggil objek tersebut. Pada Gambar 18 adalah kode fungsi *Tokenizing*:

```
def my_tokenizer(doc):
    doc = tokenizer.tokenize(doc)
    doc_token1 = [word for word in doc]
    return doc_token1
```

Gambar 18. Kode Fungsi Tokenizing

Fungsi ini akan memanggil objek tokenizer (yang sebelumnya didefinisikan dari modul nlp-id) untuk melakukan tokenisasi, yaitu memecah string atau kata pada dataset menjadi daftar kata. Hasil implementasi *Tokenizing* pada dataset dapat dilihat pada Tabel 4.

Tabel 4. Hasil Tokenizing

No.	Sesudah <i>Tokenizing</i>
1.	['bumn', 'holding', 'industri', 'pertambangan', 'indonesia', 'mind', 'id', 'meyakini', 'pembentukan', 'badan', 'pengelola', 'investasi', 'daya', 'anagata', 'nusantara', 'danantara', 'merupakan', 'langkah', 'strategis', 'untuk', 'memperkuat', 'investasi', 'hilirisasi', 'dan', 'industrialisasi', 'nasional']
2.	['danantara', 'adalah', 'ladang', 'korupsi', 'untuk', 'para', 'penjahat', 'politik', 'tidak', 'ada', 'bukti', 'bahwa', 'danantara', 'dapat', 'diawasi', 'oleh', 'rakyat', 'indonesia', 'dari', 'kejahatan', 'politik']
3.	['bumn', 'yang', 'begini', 'yang', 'disertakan', 'ke', 'danantara']
4.	['mari', 'kita', 'dukung', 'pemerintah', 'indonesia', 'yang', 'membantu', 'bumn', 'dengan', 'hadir', 'nya', 'danantara']
5.	['bumn', 'akan', 'semakin', 'hebat', 'jika', 'bergabung', 'dengan', 'danantara']
...	...
996.	['bisnis', 'tidak', 'ada', 'yang', 'rugi', 'negara', 'lain', 'iri', 'dengan', 'kita', 'ndasmu', 'masih', 'ada', 'danantara', 'nanti', 'kita', 'suntik', 'pakai', 'itu', 'duit', 'nya', 'mau', 'saham', 'nya', 'di', 'bilang', 'judi', 'oke', 'gas', 'oke', 'gass', 'all', 'in', 'tabrak', 'tabrak']
997.	['fakta', 'yang', 'tidak', 'bisa', 'di', 'bantah', 'orang', 'asing', 'menjabat', 'di', 'danantara', 'kalau', 'pemakan', 'bangkai', 'pasti', 'tidak', 'akan', 'bersuara']
998.	['dikawatirkan', 'pgeluaran', 'danantara', 'untuk', 'biaya', 'gaji', 'direksi', 'komisaris', 'pakar', 'amp', 'pengawas', 'dari', 'mantan', 'pjabat', 'dn', 'ln', 'akan', 'menguras', 'dana', 'pmegang', 'saham', 'sdgkan', 'proyek', 'yang', 'didanai', 'nya', 'belum', 'tentu', 'menguntungkan', 'kecuali', 'direksi', 'nya', 'profesional', 'dipilih', 'dari', 'bumn', 'perbankan', 'yang', 'jujur', 'amp', 'berakhlak', 'mulia']
999.	['maafkan', 'diri', 'ini', 'bila', 'belum', 'setransparan', 'pembuatan', 'ruu', 'tni', 'masih', 'banyak', 'salah', 'seperti', 'pendirian', 'danantara', 'acap', 'berkata', 'kasar', 'dan', 'mengucapkan', 'ndasmu', 'teriring', 'salam', 'dn', 'kehangatan', 'yng', 'dpat', 'memenuhi', 'raga', 'seperti', 'makan', 'bergizi', 'gratis', 'oke', 'gas', 'menuju', 'kemenangan', 'mohon', 'maaf', 'lahir', 'batin']
1000.	['halah', 'banyak', 'bicara', 'danantara', 'yang', 'jelas', 'ada', 'org', 'aseng', 'eks', 'koruptor', 'saja', 'kamu', 'mingkem', 'wkkww', 'cuih']

d. *Stopword removal/filtering*: : pada proses ini kata-kata yang tidak relevan, dan kata penghubung akan

di hapus. Di bawah ini adalah Kode *Tokenizing* dan hasil implementasinya:

Untuk melakukan *Stopword removal/filtering* pada dataset terlebih dahulu meng-inisialisasi komponen *Stopword removal/filtering* dari modul nlp-id. Kode tersebut dapat dilihat pada Gambar 19.

```
stopword = StopWord()
```

Gambar 19. Inisialisasi Stopword Removal

Setelah melakukan inisialisasi objek *Stopword removal* kemudian dibuat fungsi untuk memanggil objek tersebut. Kode tersebut dapat dilihat pada Gambar 20.

```
def my_tokenizer(doc):
    doc_token2 = [word for word in doc_token1 \
                  if word not in stopwords.get_stopword()]
    return doc_token2
```

Gambar 20. Kode Fungsi Stopword Removal

fungsi ini akan menggunakan variabel `doc_token1` (hasil *Tokenizing*) untuk dibandingkan isinya dengan daftar *stopword* dari objek *stopword* yang telah dibuat sebelumnya dengan `stopword = StopWord()`. Fungsi ini akan mengembalikan list kata-kata yang perlu dihapus. lalu menyaring kata-kata dalam `doc_token1`, dan hanya menyimpan kata-kata yang tidak ada dalam daftar *stopword*. Hasil implementasi *Stopword removal* pada dataset dapat dilihat pada Tabel 5.

Tabel 5. Hasil Stopword removal

No.	Sesudah <i>Stopword removal</i>
1.	['bumn', 'holding', 'industri', 'pertambangan', 'indonesia', 'pembentukan', 'badan', 'pengelola', 'investasi', 'daya', 'anagata', 'nusantara', 'danantara', 'langkah', 'strategis', 'memperkuat', 'investasi', 'hilirisasi', 'industrialisasi', 'nasional']
2.	['danantara', 'ladang', 'korupsi', 'penjahat', 'politik', 'bukti', 'danantara', 'diawasi', 'rakyat', 'indonesia', 'kejahatan', 'politik']
3.	['bumn', 'disertakan', 'danantara']
4.	['mari', 'dukung', 'pemerintah', 'indonesia', 'membantu', 'bumn', 'hadir', 'danantara']
5.	['bumn', 'hebat', 'bergabung', 'danantara']
...	...
996.	['bisnis', 'rugi', 'negara', 'iri', 'danantara', 'suntik', 'pakai', 'duit', 'saham', 'bilang', 'judi', 'gas', 'gass', 'all', 'in', 'tabrak', 'tabrak']
997.	['fakta', 'bantah', 'orang', 'asing', 'menjabat', 'danantara', 'pemakan', 'bangkai', 'bersuara']
998.	['dikawatirkan', 'pgeluaran', 'danantara', 'biaya', 'gaji', 'direksi', 'komisaris', 'pakar', 'amp', 'pengawas', 'mantan', 'pjabat', 'ln', 'menguras', 'dana', 'pmegang', 'saham', 'proyek', 'didanai', 'menguntungkan', 'direksi']

	'profesional', 'dipilih', 'bumn', 'perbankan', 'jujur', 'berakhlak', 'mulia']
999.	['maafkan', 'setransparan', 'pembuatan', 'ruu', 'tni', 'salah', 'pendirian', 'danantara', 'acap', 'kasar', 'teriring', 'salam', 'kehangatan', 'yng', 'dpat', 'memenuhi', 'raga', 'makan', 'bergizi', 'gratis', 'gas', 'kemenangan', 'mohon', 'maaf', 'lahir', 'batin']
1000.	['bicara', 'danantara', 'org', 'aseng', 'eks', 'koruptor', 'mingkem', 'wkkww', 'cuih']

e. *Stemming*: Pada tahap ini, dilakukan proses pengurangan kata yang memiliki awalan atau akhiran menjadi bentuk dasarnya. di bawah ini adalah Kode dan hasil implementasinya:

Pada tahap *stemming* terlebih dahulu terlebih dahulu ditambahkan modul sastrawi. Kode tersebut dapat dilihat pada Gambar 21.

```
import sys
sys.path.append
("/kaggle/input/py-sastrawi/PySastrawi-master/src")
```

Gambar 21. Kode Menambahkan Modul Sastrawi

Setelah modul di tambahkan dilanjutkan dengan mengimpor komponen dari pustaka Sastrawi, yaitu StemmerFactory. Kode tersebut dapat dilihat pada Gambar 22.

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory
```

Gambar 22. Kode Inisialisasi Komponen Stemmer

Setelah menambahkan modul *stemmer* dibuat fungsi untuk melakukan *stemming*. fungsi tersebut dapat dilihat pada Gambar 23.

```
def stemming(ulasan) :
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in ulasan:
        dt = stemmer.stem(w)
        do.append(dt)
    d_clean = []
    d_clean = " ".join(do)
    return d_clean
review['stemming_ulasan'] = review['preprocessing'].apply(stemming)
```

Gambar 23. Kode Fungsi Stemming

Fungsi *stemming()* didefinisikan untuk menerima parameter ulasan yang berupa daftar *token* (kata-kata). Di dalam fungsi ini, objek *StemmerFactory* dibuat dan digunakan untuk menghasilkan *stemmer*. Selanjutnya, dilakukan iterasi terhadap setiap kata dalam ulasan, kemudian setiap kata tersebut di *stemming* menggunakan *stemmer.stem(w)* dan dimasukkan ke dalam *list do*. Setelah seluruh kata selesai diproses, hasil *stemming* digabung kembali menjadi *string* dengan menggunakan " ".*join(do)* dan hasil akhir dikembalikan sebagai *output*.

Hasil implementasi *stemming* dapat dilihat pada Tabel 6.

Tabel 6. Hasil Stemming

No.	Sesudah <i>Stopword removal</i>
1.	bumn holding industri tambang indonesia mind id bentuk badan kelola investasi daya anagata nusantara danantara langkah strategis kuat investasi hilir industrialisasi nasional
2.	danantara ladang korupsi jahat politik bukti danantara awas rakyat indonesia jahat politik
3.	bumn serta danantara
4.	mari dukung perintah indonesia bantu bumh hadir danantara
5.	bumn hebat gabung danantara
...	...
996.	bisnis rugi negara iri ndasmu danantara suntik pakai duit saham bilang judi gas gass all in tabrak
997.	fakta bantah orang asing jabat danantara makan bangkai suara
998.	dikawatirkan danantara biaya gaji direksi komisaris pakar awas mantan pejabat dn ln uras dana pemegang saham sdgkan proyek dana untung direksi profesional pilih bumh perban jujur akhlak mulia
999.	maaf transparan buat ruu tni salah diri danantara acap kasar iring salam hangat yng dpat penuh raga makan gizi gratis gas menang mohon maaf lahir batin
1000.	bicara danantara org asing eks koruptor mingkem wkkww cuih

Mengelompokkan Data

Pada proses ini akan dilakukan pelabelan pada dataset menggunakan metode lexicon, di mana setiap data akan ditentukan masuk ke kategori sentimen positif, negatif ataupun netral, Di bawah ini adalah Kode dan hasil pelabelan pada dataset:

Untuk melakukan pelabelan dengan metode lexicon pada dataset terlebih dahulu dilakukan inisialisasi variabel yang berisi lexicon atau kamus kata baik itu kata positif maupun negatif dan skor masing-masing kata. kode inisialisasi variabel dapat dilihat pada Gambar 24.

```
lexicon_positive = pd.read_excel
('/kaggle/input/kamus-sentimen-lexicon/InSet/positive.xlsx')
lexicon_negative = pd.read_excel
('/kaggle/input/kamus-sentimen-lexicon/InSet/negative.xlsx')
```

Gambar 24. Kode Inisialisasi Kamus Lexicon

Setelah proses inisialisasi variabel yang berisi kamus kemudian Kedua file ini kemudian diubah menjadi struktur data dictionary agar lebih cepat dicari ketika proses analisis, melalui iterasi setiap baris pada dataframe. Penerapan untuk kata positif dapat dilihat pada Gambar 25.

```
for index, row in lexicon_negative.iterrows():
    if row[0] not in lexicon_negative_dict:
        lexicon_negative_dict[row[0]] = row[1]
```

Gambar 25. Iterasi Baris Kata Negatif

Kemudian itu di buat fungsi untuk menghitung skor sentimen dari setiap ulasan. Fungsi ini bekerja dengan menjumlahkan nilai dari kata-kata dalam ulasan yang terdapat dalam `lexicon_positive_dict` dan `lexicon_negative_dict`. kode fungsi tersebut dapat dilihat pada Gambar 26.

```
def sentiment_analysis_lexicon_indonesia(ulasan):
    score = 0
    for word in ulasan:
        if (word in lexicon_positive_dict):
            score = score + lexicon_positive_dict[word]
    for word in ulasan:
        if (word in lexicon_negative_dict):
            score = score + lexicon_negative_dict[word]
    sentimen=''
    if (score > 0):
        sentimen = 'positif'
    elif (score < 0):
        sentimen = 'negatif'
    else:
        sentimen = 'netral'
    return score, sentimen
```

Gambar 26. Fungsi Menghitung skor sentimen

Setelah skor dihitung, fungsi akan menentukan apakah sentimen dari ulasan tersebut adalah positif, negatif, atau netral berdasarkan nilai skor, jika nilai skor lebih besar dari 0 maka sentimen positif jika sentimen lebih kecil dari 0 maka sentimen negatif tetapi jika tidak maka sentimen adalah netral. Kemudian mendapatkan hasil fungsi mengembalikan dua *output*: *score* dan *sentimen*. Kemudian fungsi ini digunakan ke setiap baris data dalam kolom *preprocessing* dari *dataframe review* menggunakan *method .apply()*. kode untuk proses implementasi dapat dilihat pada Gambar 27.

```
results = review['preprocessing']\
.apply(sentiment_analysis_lexicon_indonesia)
```

Gambar 27. Kode Implementasi Fungsi Lexicon

Dan Karena fungsi mengembalikan tipe data tuple (score, sentimen), hasilnya dipisah menjadi dua list menggunakan `zip(*)` Kemudian hasil Nilai label atau sentimen masing-masing disimpan ke dalam kolom baru. kode pemisahan dapat dilihat pada Gambar 28.

```
results = list(zip(*results))
review['label'] = results[0]
```

Gambar 28. Kode Pemisahan Output Fungsi Lexicon

Kemudian Data hasil akhir kemudian disimpan ke dalam *file CSV*. Kode dapat dilihat pada Gambar 29.

```
data_inset = review
data_inset[['full_text', 'preprocessing', 'label']]
data_inset.to_csv\
('/kaggle/working/Pelabelan_Lexicon.csv', index=False)
```

Gambar 29. Menyimpan Hasil Pelabelan Lexicon

hasil pelabelan dengan metode lexicon pada dataset dapat dilihat pada Tabel 7.

Tabel 7. Hasil Pelabelan Lexicon

No.	tweet	Sentimen
1.	bumn holding industri tambang indonesia mind id bentuk badan kelola investasi daya anagata nusantara danantara langkah strategis kuat investasi hilir industrialisasi nasional	positif
2.	danantara ladang korupsi jahat politik bukti danantara awas rakyat indonesia jahat politik	negatif
3.	bumn serta danantara	positif
4.	mari dukung perintah indonesia bantu bumn hadir danantara	positif
5.	bumn hebat gabung danantara	netral
...	...	
996.	bisnis rugi negara iri ndasmu danantara suntik pakai duit saham bilang judi gas gass all in tabrak	negatif
997.	fakta bantah orang asing jabat danantara makan bangkai suara	negatif
998.	dikawatirkan danantara biaya gaji direksi komisaris pakar awas mantan pejabat dn ln uras dana pemegang saham sdgkan proyek dana untung direksi profesional pilih bumn perban jujur akhlak mulia	positif
999.	maaf transparan buat ruu tni salah diri danantara acap kasar iring salam hangat yng dpat penuh raga makan gizi gratis gas menang mohon maaf lahir batin	netral
1000.	bicara danantara org asing eks koruptor mingkem wkkww cuih	negatif

Setelah dilakukan pelabelan dari total dataset yang berjumlah 1000 terdapat 556 data dengan label negatif, 291 data dengan label positif, dan 153 data dengan label netral, dataset yang sudah dilabeli ini selanjutnya akan dibagi menjadi 2 bagian yaitu data uji dan data latih. Setelah melewati proses pembagian ini dataset

kemudian akan digunakan dalam proses klasifikasi dengan SVM.

Klasifikasi SVM

Pada bagian ini dataset yang sudah melalui proses *pre-processing* dan pengelompokan kemudian akan digunakan untuk mengembangkan model dengan mencari garis pemisah atau *hyperplane* untuk memisahkan setiap kelas baik itu positif, negatif maupun netral adapun pembagian data latih dan data uji adalah 70% data uji dan 30% data latih atau dari 1000 data 700 data akan digunakan sebagai data latih dan 300 data akan digunakan sebagai data uji. Di bawah ini adalah Kode dan hasil implementasinya:

Proses diawali dengan mengimpor pustaka yang dibutuhkan seperti *pandas* dan *matplotlib*, kemudian membaca file CSV hasil *preprocessing*. kode dapat dilihat pada Gambar 30.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv(
    "/kaggle/input/hasil-pre-process/hasil_pre_process.csv")
```

Gambar 30. Mempersiapkan Library Untuk Program SVM

Selanjutnya Label sentimen seperti 'positif', 'negatif', dan 'netral' akan dikonversi ke angka menggunakan *mapping* agar bisa digunakan dalam model *machine learning*. *Mapping* ini dilakukan dengan *.map()* pada kolom label. Kode dapat dilihat pada Gambar 31.

```
sentimen_mapping = {"negatif": -1, "netral": 0, "positif": 1}
df["sentimen_encoded"] = df["label"].map(sentimen_mapping)
df["sentimen_encoded"]
```

Gambar 31. Kode Konversi label ke angka

Kemudian dicari representasi vektor dari dataset menggunakan Program TF-IDF Vectorizer dari *sklearn* untuk mengubah teks pada kolom 'full_text' atau tweet menjadi representasi numerik Kode dapat dilihat pada Gambar 32.

```
from sklearn.feature_extraction.text import TfidfVectorizer
COL = 'full_text'
tfidf = TfidfVectorizer(min_df=3, max_df=0.5, ngram_range=(1, 1))
features = tfidf.fit_transform(df[COL].values)
dfuni = pd.DataFrame(features.todense(), columns=tfidf.get_feature_names_out())
```

Gambar 32. Kode Mencari TF-IDF

Setelah dilakukan pencarian TF-IDF selanjutnya pelatihan model SVM akan dilakukan dengan menggunakan kode dapat dilihat pada Gambar 33.

```
from sklearn import svm

clf = svm.SVC(
    max_iter=-1,
    C=1000,
    kernel='rbf'
)
clf.fit(X_unitrain, y_unitrain.values.ravel())
```

Gambar 33. Kode Melatih Model SVM

Setelah membuat model kita mencoba model tersebut. kode untuk menggunakan 1 data latih untuk klasifikasi dapat dilihat pada Gambar 34.

```
index = 299
hasil_klasifikasi = pd.DataFrame({
    'Label Asli': [y_unitest.iloc[index]],
    'Klasifikasi Model': [y_unipred[index]]
})

print("\nIsi tweet yang diklasifikasikan:")
print(X_text_test_asli[index])

print("\nTabel Hasil Klasifikasi:")
print(hasil_klasifikasi.to_markdown(index=False, tablefmt="grid"))
```

Gambar 34. Menampilkan Hasil Klasifikasi 1 Data Uji

Bagian kode di atas menampilkan hasil klasifikasi satu data uji, prosesnya dimulai dengan inisialisasi variabel *index = 299*, yang menunjuk pada data ke-299 dari *X_unitest* atau data uji. Selanjutnya, dibuat sebuah *Data frame* bernama *hasil_klasifikasi* yang menampilkan label asli (*y_unitest.iloc[index]*) dan hasil klasifikasi model (*y_unipred[index]*) untuk data tersebut. Isi tweet asli yang diuji kemudian ditampilkan melalui *X_text_test_asli[index]*, yaitu *list* berisi teks asli *tweet* yang sebelumnya disusun agar urutannya sesuai dengan *X_unitest*. Terakhir, hasil klasifikasi tersebut dicetak dalam bentuk tabel menggunakan *to_markdown()* agar tampil rapi di terminal.

Contoh hasil klasifikasi menggunakan model yang sudah di bentuk dapat dilihat pada Gambar 35.

Isi Tweet yang diklasifikasikan:
 @aewin86 Halah bacot. Danantara yg jelas2 ad org ASENG eks KORUPTOR aja lu mingkem wkkww. Cuih

Tabel Hasil Klasifikasi:

Label Asli	Klasifikasi Model
negatif	negatif

Gambar 35. Hasil Klasifikasi Dengan 1 Data

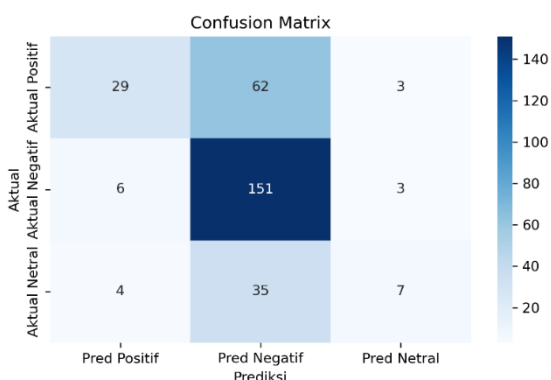
hasil klasifikasi dengan 1 data tersebut menunjukkan model berhasil mengklasifikasi label asli dengan tepat.

Untuk mendapatkan *Confusion matrix* hasil klasifikasi model terhadap 300 data uji digunakan kode pada Gambar 36.

```
from sklearn import metrics
import matplotlib.pyplot as plt
cm = metrics.confusion_matrix\
(y_unittest, y_unipred, labels=['positif', 'negatif', 'netral'])
cm_df = pd.DataFrame(cm,
    index=['Aktual Positif', 'Aktual Negatif', 'Aktual Netral'],
    columns=['Pred Positif', 'Pred Negatif', 'Pred Netral'])
plt.figure(figsize=(6, 4))
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.ylabel('Aktual')
plt.xlabel('Prediksi')
plt.tight_layout()
plt.savefig("confusion_matrix.png", dpi=300)
```

Gambar 36. Kode confusion Matrix

Pada baris kode di atas, bagian `from sklearn import metrics` dan `import matplotlib.pyplot as plt` berperan penting dalam mengevaluasi serta menampilkan hasil model, khususnya membuat dan menyimpan *confusion matrix*. Modul `metrics` menghitung perbandingan antara jawaban sebenarnya (`y_unittest`) dan hasil prediksi model (`y_unipred`), dengan nama kelas ditentukan langsung agar urutannya tetap. *Matriks* ini diubah menjadi *DataFrame* `cm_df`. Hasilnya ditampilkan dalam grafik berwarna biru menggunakan `seaborn.heatmap()`. Grafik ini dilengkapi judul, nama sumbu, angka, dan ditata rapi dengan `plt.tight_layout()`, lalu disimpan sebagai *file* gambar melalui `plt.savefig()`. Hasil *confusion matrix* 300 data uji dapat dilihat pada Gambar 37.



Gambar 37. Hasil Confusion Matrix

Confusion matrix memberikan gambaran menyeluruh tentang kinerja model klasifikasi dalam mengklasifikasi data.

Pada data yang sebenarnya berlabel Positif, terdapat 29 data yang berhasil diklasifikasikan dengan benar oleh model sebagai Positif. Namun, model pada 62 data yang seharusnya Positif diprediksi sebagai Negatif, dan 3 data lainnya diprediksi sebagai Netral. Untuk data dengan label Negatif, model berhasil mengklasifikasikan 151 data dengan benar sebagai Negatif, tetapi masih terdapat 6 data yang diklasifikasikan sebagai Positif dan 3 data lainnya sebagai Netral. Sedangkan pada data dengan label Netral, model hanya berhasil mengklasifikasikan 7 data secara benar sebagai Netral, terdapat 35 data yang salah sebagai Negatif dan 4 data lainnya sebagai Positif.

Dari total 300 data uji, sebanyak 187 data diklasifikasikan dengan benar.

Dan Untuk mendapatkan presisi, akurasi, *recall*, *f1-score*, dan *macro-average* Model ditambahkan pada Gambar 38.

```
import numpy as np
import pandas as pd
from sklearn import metrics
import seaborn as sns
import matplotlib.pyplot as plt

report = metrics.classification_report\
(y_unittest, y_unipred, output_dict=True, digits=3)
report_df = pd.DataFrame(report).transpose()
rows = ['positif', 'negatif', 'netral', 'accuracy', 'macro avg']
cols = ['precision', 'recall', 'f1-score']

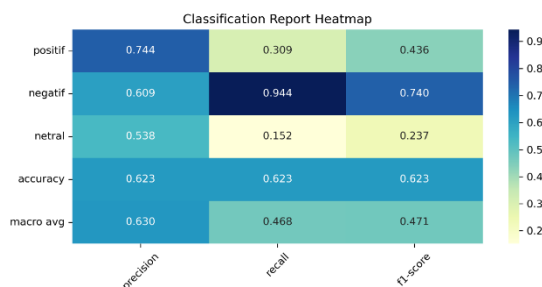
eval_matrix = report_df.loc[rows, cols]
if 'accuracy' in eval_matrix.index:
    acc_value = report_df.loc['accuracy']['precision']
    eval_matrix.loc['accuracy'] = [acc_value] * 3

plt.figure(figsize=(8, 4))
sns.heatmap(eval_matrix.astype(float), annot=True, cmap='YlGnBu', fmt='.3f')
plt.title("Classification Report Heatmap")
plt.yticks(rotation=0)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("classification_report_matrix.png", dpi=300)
plt.show()
```

Gambar 38. Kode Classification Report

kode ini dimulai dengan mengimpor *library* yang dibutuhkan untuk evaluasi model, seperti `numpy`, `pandas`, `seaborn`, `matplotlib`, dan `sklearn.metrics`. Selanjutnya, *classification report* dihitung menggunakan `metrics.classification_report()` dengan `output_dict=True` agar hasilnya berupa *dictionary*, lalu dikonversi ke *DataFrame* untuk difilter dan divisualisasikan. *Output* dari kode ini akan menampilkan metrik penting seperti *precision*, *recall*, dan *f1-score* untuk kelas Positif, Negatif, dan Netral, serta akurasi dan rata-rata makro atau *Macro average*. Kemudian data divisualisasikan menggunakan `seaborn.heatmap()`, lalu disimpan sebagai gambar `classification_report_matrix.png`.

Pada Gambar 39 di tunjukan hasil presisi, akurasi, *recall*, *f1-score*, dan *macro-average* dari Model:



Gambar 39. Hasil Evaluasi Model

Hasil evaluasi model menunjukkan bahwa secara keseluruhan model memiliki akurasi sebesar 62,3% pada data uji sebanyak 300 data. Pada kelas Negatif, nilai

recall 0.944, *precision*-nya 0.609, dan *f1-score* untuk kelas ini adalah 0.740. Untuk kelas Netral, *precision*-nya di angka 0.538, *recall* 0.152, dengan *f1-score* 0.237. Sementara pada kelas Positif, *precision* 0.744, nilai *recall* 0.309, dan nilai *f1-score*-nya adalah 0.436. Secara keseluruhan, nilai rata-rata makro (*macro average*) dari *precision*, *recall*, dan *f1-score* masing-masing adalah 0.630, 0.468, dan 0.471.

Penyajian Hasil

Hasil dari analisis sentimen DANANTARA menggunakan Metode *Support Vector Machine* pada media sosial X di visualisasikan menggunakan *pie diagram*. Kode untuk visualisasi diagram dan hasil dapat dilihat pada Gambar 40.

```
import pandas as pd
import matplotlib.pyplot as plt

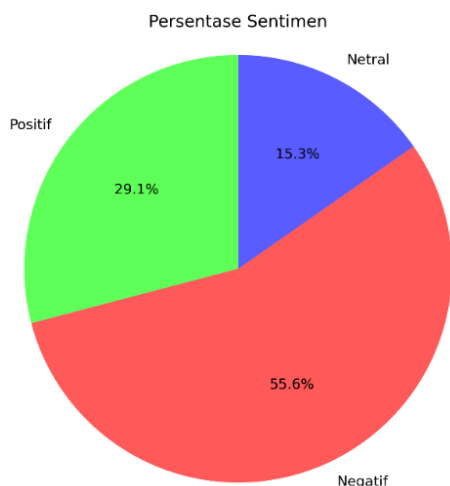
positif_count = df[df['label'] == 'positif'].shape[0]
negatif_count = df[df['label'] == 'negatif'].shape[0]
netral_count = df[df['label'] == 'netral'].shape[0]

sentimen = [positif_count, negatif_count, netral_count]
labels = ['Positif', 'Negatif', 'Netral']
colors = ['#5FFF59', '#FF5959', '#595CFF']

plt.figure(figsize=(6, 6))
plt.pie(sentimen, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
plt.title('Persentase Sentimen')
plt.axis('equal')
plt.savefig('/kaggle/working/persentase_sentimen.png', dpi=300)
```

Gambar 40. Kode Visualisasi Diagram

Kode ini memulai proses visualisasi dengan mengimpor *pandas* dan *matplotlib.pyplot*. *DataFrame* *df* berisi data tweet dengan kolom label sentimen: positif, negatif, dan netral. Jumlah tweet per kategori dihitung menggunakan filter kondisi dan *shape[0]*, lalu disusun dalam *list* bersama label dan warnanya. Diagram pie dibuat dengan *plt.pie()* untuk menunjukkan persentase sentimen, diputar 90 derajat agar rapi, dan *plt.axis('equal')* digunakan agar grafik bulat. Hasil visualisasi disimpan sebagai 'persentase_sentimen.png'. Diagram ditampilkan pada Gambar 41.



Gambar 41. Persentase Label Sentimen

Dari grafik tersebut dapat diketahui bahwa sentimen dengan label negatif pada topik DANANTARA masih mendominasi proporsi sebesar 55,6% atau 556 *tweet* dari total data. Sentimen Positif menempati posisi kedua dengan proporsi 29,1% atau 291 *tweet*, sedangkan sentimen Netral hanya sebesar 15,3% atau 153 *tweet*.

Untuk mendapatkan visualisasi *Wordcloud* atau kata yang paling sering muncul di ketiga label, digunakan kode pada Gambar 42.

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

df['text_cleaning'] = df['stemming_ulasan'].fillna('')
df_netral = df[df['label'] == 'netral']
df_negatif = df[df['label'] == 'negatif']
df_positif = df[df['label'] == 'positif']
text_netral = ' '.join(df_netral['text_cleaning'])
text_negatif = ' '.join(df_negatif['text_cleaning'])
text_positif = ' '.join(df_positif['text_cleaning'])

wordclouds = {
    'netral': WordCloud(width=800, height=400).generate(text_netral),
    'negatif': WordCloud(width=800, height=400).generate(text_negatif),
    'positif': WordCloud(width=800, height=400).generate(text_positif)
}

for label, wc in wordclouds.items():
    # Simpan sebagai file gambar PNG
    filename = f'wordcloud_{label}.png'
    wc.to_file(filename)
```

Gambar 42. Kode Visualisasi Wordcloud

Kode di atas diawali dengan mengimpor library *WordCloud* dan *matplotlib.pyplot* untuk pembuatan serta visualisasi *wordcloud*. Kolom *stemming_ulasan* (berisi hasil pre-processing teks tweet) dibersihkan dari nilai kosong menggunakan *fillna("")*. Data kemudian dipisahkan menjadi tiga subset berdasarkan label sentimen: netral, negatif, dan positif. Masing-masing subset digabungkan menjadi satu string teks dengan *' '.join(...)*, lalu digunakan untuk menghasilkan objek *WordCloud* berukuran 800x400 piksel. *WordCloud* tiap sentimen disimpan dalam *dictionary wordclouds*, dan melalui perulangan, tiap gambar disimpan sebagai file PNG dengan nama sesuai label kategorinya. *Wordcloud* dari ketiga label (positif, negatif, dan netral) dapat dilihat di bawah ini:

a. *Wordcloud* Sentimen Positif



Gambar 43. *Wordcloud* Sentimen Positif

- [11] “Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 1, no. 1, pp. 24–33, 2021, doi: 10.57152/malcom.v1i1.20.
- D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, “Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia,” *Edutic - Sci. J. Informatics Educ.*, vol. 7, no. 1, pp. 1–11, 2020, doi: 10.21107/edutic.v7i1.8779.