

Machine Learning-Based Sentiment Analysis on Twitter (X): A Case Study of the “Kabur Aja Dulu” Issue Using SVM

Lina Rohmatun ^{1*}, Anna Baita ^{2*}

* Informatika, Universitas Amikom Yogyakarta, Indonesia
linarohmatun8@students.amikom.ac.id ¹, anna@amikom.ac.id ²

Article Info

Article history:

Received 2025-06-28

Revised 2025-07-20

Accepted 2025-08-11

Keyword:

*Sentiment Analysis,
Twitter (X),
Support Vector Machine
(SVM),
Machine Learning,
Kabur Aja Dulu.*

ABSTRACT

This study aims to analyze public sentiment toward the phenomenon of “Kabur Aja Dulu” on Twitter (X) using the Support Vector Machine (SVM) method. The data used consists of 4,768 Indonesian-language tweets collected through web scraping. The pre-processing process includes data cleaning, tokenization, stemming, and translation into English for automatic sentiment labeling using TextBlob. The data is then classified into three sentiment categories: positive, negative, and neutral. To address class imbalance, the SMOTE method is applied to the training data, along with TF-IDF techniques for feature extraction. The model was evaluated using the K-Fold Cross Validation method and Grid Search for hyperparameter tuning. The results of the study show that the SVM model with a linear kernel and parameter $C=10$ provides the best performance with an accuracy value of 85.56%, precision of 84.19%, recall of 85.56%, and F1-score of 85.30%. The main finding of this study is that the linear SVM method is capable of classifying sentiment well, particularly for neutral sentiment data, and has proven effective as an approach to sentiment analysis in the context of social media using the Indonesian language.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Fenomena yang menyebar di kalangan generasi muda saat ini adalah kabur aja dulu. Fenomena ini menjadi trend di berbagai media sosial sebagai bentuk protes dan kritik masyarakat terhadap kondisi ekonomi, sosial dan politik di Indonesia[1]. Fenomena “Kabur Aja Dulu” sebenarnya sudah mulai ada di tahun 2023 dan kembali menjadi trend di awal tahun 2025 dengan masalah sulitnya mencari pekerjaan dengan gaji yang layak di Indonesia[2]. Fenomena ini juga menggambarkan kegagalan negara dalam mengolah sumber daya manusia yang bertalenta, sehingga menyebabkan generasi muda lebih memilih kabur dengan harapan masa depan yang lebih layak dan pasti[3].

Di era digital seperti saat ini, isu yang sedang memanas atau trending menjadi sebuah topik yang sering dibahas di berbagai media sosial terutama di *Twitter (X)*[4]. *Twitter (X)* merupakan sebuah *microblog* yang digunakan penggunaanya untuk berkomunikasi dan mengungkapkan pendapat tentang berbagai hal yang menarik perhatian publik [5]. Media sosial

Twitter (X) juga sering dijadikan sumber data untuk berbagai penelitian isu sosial termasuk analisis sentimen [6].

Analisis sentimen adalah proses mengidentifikasi, mengekstrak dan memperoleh pengetahuan tentang sentimen yang terkandung dalam teks atau dokumen tertentu, sehingga dapat memahami opini yang tersirat dalam publik[7]. Tujuannya untuk mengidentifikasi dan memahami sentimen yang terkandung dalam teks serta mengukur dan mengelompokkan menjadi berbagai kategori, seperti positif, negatif, dan netral[8].

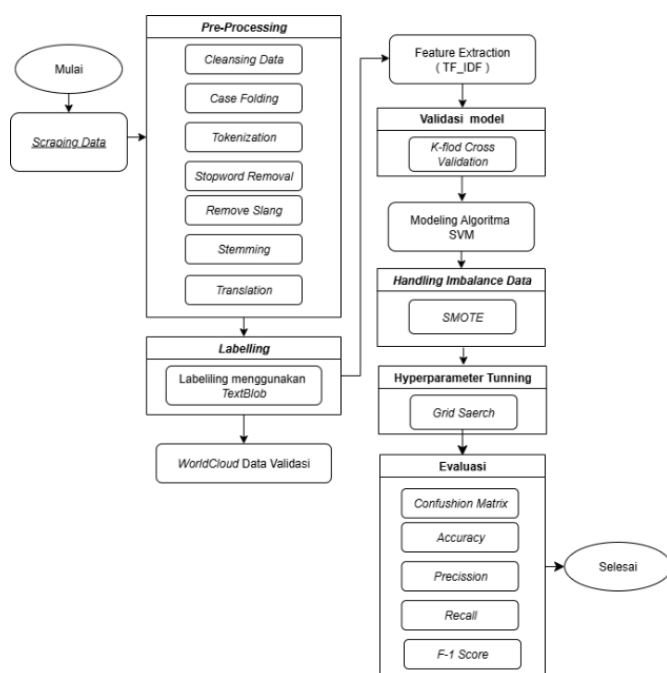
Support Vector Machine (SVM) terdefinisi metode yang baik untuk model klasifikasi [9]. SVM merupakan suatu teknik yang dapat memisahkan dua himpunan data dari dua kelas yang berbeda dengan memaksimalkan *hyperplane* serta dapat melakukan pengklasifikasian dan mengatasi regresi linear maupun non linear[10]. Pemilihan metode SVM didasarkan pada efisiensi dan kesesuaian dataset. Pada penelitian Hidayatunnisa’I membandingkan metode SVM dan *Naive Bayes* menggunakan kernel linear, SVM lebih efektif untuk menangani data dengan hubungan linear maupun non-linear melalui pemilihan kernel yang sesuai, sedangkan *Naive Bayes*

hanya efektif pada data dengan distribusi sederhana dan asumsi independensi antar fitur[11]. Serta metode *Random Forest* lebih cocok untuk data numerik, *Deep Learning* (LSTM, BERT) memang lebih akurat dalam NLP, tetapi membutuhkan dataset besar, komputasi tinggi, dan waktu pelatihan lama[12]. Oleh karena itu metode SVM dipilih sebagai metode yang lebih efisien dan praktis dalam penelitian ini[13].

Penelitian ini bertujuan untuk mengevaluasi akurasi dari metode SVM dalam menganalisis sentiment Masyarakat di media sosial *Twitter* (X). Studi ini juga berfokus pada opini Masyarakat terhadap fenomena kabur aja dulu yang beredar di media sosial.

II. METODE

Penelitian ini menggunakan pendekatan analisis sentimen untuk mengelompokkan opini pengguna *Twitter* (X) terhadap fenomena Kabur Aja Dulu menggunakan metode SVM yang diperoleh melalui proses *crawling data*. Rangkaian proses penelitian mencakup pengumpulan data, tahapan pre-processing teks, pelabelan otomatis menggunakan *TextBlob*, transformasi data ke dalam bentuk numerik menggunakan metode TF-IDF, penanganan ketidak seimbangan data dengan teknik *SMOTE*, pelatihan dan evaluasi model klasifikasi menggunakan algoritma *Support Vector Machine* (SVM), serta validasi performa model menggunakan metode *K-Fold Cross Validation*. Alur lengkap proses penelitian ini ditampilkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

A. Scraping Data

Tahap pertama dalam proses analisis sentimen ini adalah *scraping data* yang akan digunakan, dalam penelitian ini, data

dikumpulkan dari platform *Twitter* (x). Pengambilan data dari *Twitter* terbilang cukup mudah karena platform ini menyediakan API (*Application Programming Interface*) yang ditujukan khusus bagi para pengembang untuk mengakses data secara langsung[14]. Penelitian ini menggunakan 4768 data yang dikumpulkan menggunakan Bahasa Indonesia dan bertagar Kabur Aja Dulu. Kemudian dilakukan seleksi berdasarkan kategori yang relevan.

B. Pre-Processing

Setelah proses pengumpulan data selesai dilakukan, langkah selanjutnya adalah *pre-processing*. Tahapan ini bertujuan untuk mempersiapkan data agar dapat dianalisis secara optimal. *Preprocessing* mencakup beberapa proses penting seperti pembersihan data, *case folding*, *tokenization*, *stopword*, *stemming*, *translation*[15]. Pada tahap ini dilakukan pembersihan data dan standar format data agar siap dianalisis. Sebanyak 4768 data akan diproses seperti berikut.

1. Cleansing Data

Cleansing data merupakan proses untuk menghapus elemen-elemen yang tidak diperlukan dari teks, seperti karakter khusus, tautan, emoji, mention, serta tanda baca yang tidak dibutuhkan. Beberapa contoh tanda baca yang biasanya dihilangkan meliputi tanda seru, tanda tanya, koma, dan titik[16].

2. Tokenization

Tokenization merupakan langkah yang dilakukan setelah proses *case folding*, di mana pada tahap ini, tanda baca, spasi, dan karakter-karakter lain yang tidak relevan akan di hapus atau diabaikan. Proses ini bertujuan untuk memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau frasa, yang disebut token. Setiap token ini nantinya dapat dianalisis atau digunakan dalam pemrosesan lebih lanjut, seperti pencarian atau perhitungan frekuensi kata[17].

3. Stopword Removal

Stopword removal adalah tahapan dalam pengolahan teks yang bertujuan untuk menghilangkan kata-kata yang dianggap tidak memberikan kontribusi signifikan terhadap analisis atau pemahaman makna teks. Kata-kata ini sering kali bersifat fungsional atau struktural, seperti konjungsi, preposisi, dan kata penghubung, yang sering muncul dalam kalimat namun tidak memberikan informasi penting dalam konteks analisis, seperti “yang”, “di”, “dan”, “pada”, “untuk”, dan sebagainya [18].

4. Remove Slang

Remove slang adalah proses membersihkan dan menormalkan kata-kata dalam sebuah kalimat dengan cara mengganti istilah-istilah *slang* atau tidak baku menjadi bentuk kata yang sesuai dengan kaidah bahasa Indonesia, sehingga memudahkan sistem analisis sentimen dalam memahami makna teks secara akurat[19].

5. Stemming

Stemming adalah proses untuk mengubah kata yang memiliki awalan atau akhiran, seperti kata depan, kata sambung, atau kata ganti, menjadi bentuk dasarnya. Misalnya, kata "berlari" diubah menjadi "lari", atau "mempelajari" menjadi "pelajari". Tujuannya adalah agar kita hanya fokus pada bentuk kata yang paling sederhana dan sesuai dengan yang ada di Kamus Besar Bahasa Indonesia (KBBI). Proses ini mempermudah analisis teks karena kata-kata yang berbeda bentuknya akan diperlakukan sebagai satu kata yang sama[20].

6. Translation

Translation merupakan proses penerjemahan otomatis dari teks Bahasa Indonesia ke Bahasa Inggris dengan memanfaatkan library seperti Googletrans, yang bertujuan untuk menyesuaikan data dengan alat pelabelan sentimen berbasis leksikal seperti TextBlob, yang hanya mengenali dan memproses teks berbahasa Inggris[21].

C. Labelling

Dari 4768 data yang sudah melalui proses pre-processing terdapat beberapa data yang dipilih untuk proses *labelling*. *Labeling* adalah proses di mana teks diberi label atau kategori berdasarkan analisis tertentu, seperti senimen, topik, atau entitas yang terkandung dalam teks tersebut. Proses ini sering digunakan dalam berbagai aplikasi, seperti analisis sentimen di media sosial, klasifikasi berita, atau pengenalan entitas nama (NER)[22].

Labeling dapat dilakukan secara otomatis dengan berbagai metode, salah satunya menggunakan *TextBlob*, dalam penelitian ini *TextBlob* digunakan untuk memberikan dua label data yakni positif dan negatif[23]. Proses pelabelan melibatkan proses terjemah Bahasa Indonesia ke Bahasa Inggris[24].

D. WordCloud Data Validation

WordCloud adalah proses visualisasi yang menunjukkan seberapa sering kata-kata muncul dalam sebuah kumpulan teks. Dalam *WordCloud*, setiap kata diwakili oleh sebuah huruf atau frasa, dan ukuran huruf tersebut menggambarkan frekuensi kemunculannya. Semakin besar ukuran huruf, semakin sering kata tersebut muncul dalam teks, sedangkan kata yang jarang muncul akan memiliki ukuran huruf yang lebih kecil[25].

E. Feature Extraction (TF-IDF)

TF-IDF (*Term Frequency - Inverse Document Frequency*) adalah metode yang digunakan dalam pengolahan teks untuk memberikan bobot atau nilai pada kata-kata dalam sebuah dokumen, berdasarkan seberapa sering kata tersebut muncul dalam dokumen dan seberapa unik kata tersebut di seluruh dataset [26].

1. Term Frequency (TF)

TF mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Semakin sering kata muncul, semakin besar bobotnya. Pada perhitungan *augment TF*, digunakan rumus:

$$tf = 0.5 + 0.5 \times \frac{tf}{\max(tf)} \quad (1)$$

Keterangan :

tf : adalah frekuensi kata dalam dokumen. $\max(tf)$: adalah frekuensi kata tertinggi dalam dokumen tersebut.

2. Inverse Document Frequency (IDF)

IDF mengukur seberapa penting suatu kata dalam seluruh koleksi dokumen. Kata yang jarang muncul di banyak dokumen dianggap lebih penting, sementara kata yang sering muncul di hampir semua dokumen akan memiliki bobot IDF yang lebih rendah. Rumus IDF adalah:

$$idf = \ln\left(\frac{N}{df}\right) + 1 \quad (2)$$

Keterangan :

N : adalah jumlah total dokumen.

df : jumlah dokumen yang mengandung kata tersebut.

ln : adalah algoritma natural

F. Validasi model

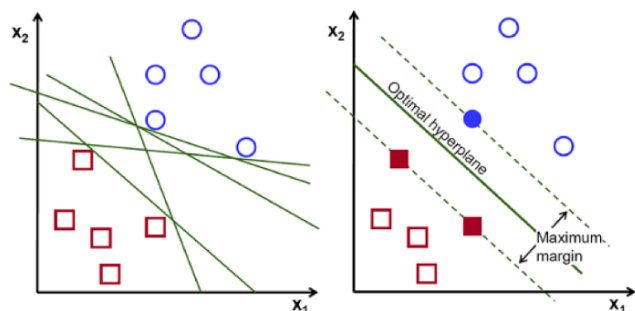
K-fold cross validation merupakan salah satu Teknik yang berfungsi untuk membagi data menjadi data *test* dan data *train*. Teknik ini dilakukan untuk mengurangi bias yang didapat Ketika pengambilan sebuah sampel. *K-fold cross validation* berlaku berulang-ulang membagi data menjadi data *train* dan data *test*, sehingga setiap data mendapat kesempatan untuk menjadi data *test* [27].

G. Handling Imbalance Data (SMOTE)

Untuk menangani ketidakseimbangan kelas dan data, tahap selanjutnya adalah menggunakan SMOTE (Synthetic Minority Over-sampling Technique), Teknik ini menghasilkan sampel sintesis dari kelas minoritas untuk menyeimbangkan data pada kelas mayoritas. Tujuannya untuk mengurangi bias pada model *machine learning*[28]. Selain itu *SMOTE* berguna untuk meningkatkan akurasi dan efektivitas metode klasifikasi dengan mengurangi bias yang sering terjadi akibat ketidakseimbangan kelas[29].

H. Modeling Algoritma Support Vector Machine

Support Vector Machine (SVM) diperkenalkan oleh Vapnik pada tahun 1992 sebagai teknik klasifikasi yang sangat efisien, khususnya untuk masalah yang tidak linier[30]. SVM dikenal sebagai salah satu teknik pembelajaran mesin (*machine learning*) yang paling canggih, yang muncul setelah metode sebelumnya seperti *Neural Network* (NN). Baik SVM maupun NN telah berhasil diterapkan dalam berbagai masalah pengenalan pola. Proses pembelajaran pada SVM dilakukan dengan menggunakan pasangan data input dan output yang berisi sasaran atau label yang diinginkan[26].



Gambar 2. Algoritma SVM

Secara sederhana, SVM bekerja dengan cara mencari hyperplane terbaik yang dapat memisahkan dua kelas yang berbeda dalam ruang input. Tujuan dari SVM adalah untuk menemukan sebuah hyperplane yang memaksimalkan jarak antara dua kelas tersebut, yang disebut dengan margin. Semakin besar margin yang tercipta, semakin baik kemampuan model SVM dalam memisahkan kelas-kelas tersebut[31]. Dengan pendekatan ini, SVM mampu menjamin *generalization* yang tinggi pada data yang belum pernah terlihat sebelumnya, sehingga sangat efektif dalam memprediksi data baru yang akan datang.

I. Hyperparameter Tunning

Hyperparameter tuning merupakan proses sistematis untuk mencari nilai kombinasi yang baik dalam *machine learning* untuk meningkatkan performa model yang digunakan[32]. Salah satu metode populer dalam *hyperparameter* tuning adalah *Grid Search*, yaitu teknik eksplorasi menyeluruh yang menguji semua kombinasi nilai *hyperparameter* dalam sebuah *grid multidimensi*. *Grid Search* menggunakan *cross-validation* untuk mengevaluasi setiap kombinasi parameter, sehingga dapat memilih konfigurasi terbaik berdasarkan hasil validasi yang diperoleh[33].

J. Evaluasi model

Evaluasi model yang dilakukan di system ini menggunakan metode *confusion matrix* untuk pengukuran precision, recall, dan accuracy.

TABEL I.
CONFUSION MATRIX

Kelas	Klasifikasi	
	True Positive (TP)	False Negative (FP)
Positif		
Negatif	False Negative (FN)	True Negative (TN)

Dari tabel 1, dapat dijelaskan bahwa:

1. TP : Prediksi sitem positif dan sesuai target positif.
2. TN : Prediksi sitem negatif dan sesuai target negative.
3. FP : Prediksi sistem positif namun target negative.
4. FN : Prediksi sistem negatif namun target positif.

Hasil evaluasi menggunakan *confussion matrix* menghasilkan nilai:

1. Akurasi yang merupakan presentase suatu kelas terprediksi yang sudah dibuat sesuai model dengan rumus:

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

2. *Precision* untuk menggambarkan akurasi data yang diminta dengan prediksi yang diberikan dengan rumus:

$$Precision = \frac{Tp}{TP + FP} \quad (4)$$

3. *Recall* untuk menggambarkan keberhasilan model dengan rumus:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

4. *F-1 Score* menggambarkan perbandingan rata – rata precision dan recall yang dibobotkan dengan rumus:

$$F - 1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

III. HASIL DAN PEMBAHASAN

A. Pengumpulan data

Pengumpulan data dalam penelitian ini dilakukan dengan menggunakan metode *web scraping* pada media sosial *Twitter* (X) untuk memperoleh data terkait isu “Kabur Aja Dulu” yang menjadi tren di kalangan pengguna media sosial Indonesia. *Scraping* dilakukan dengan bantuan pustaka Python seperti *snsrape*, yang memungkinkan pengambilan data *tweet* tanpa memerlukan akses API *Twitter* resmi. Jumlah data yang terkumpul dari proses pengumpulan data sebanyak 4768.

Data hasil *scraping* disusun dalam bentuk tabel dengan kolom sebagai berikut: *conversation_id_str*, *created_at*, *favorite_count*, *full_text*, *id_str*, *image_url*, *in_reply_to_screen_name*, *lang*, *location*, *quote_count*, *reply_count*, *retweet_count*, *tweet_url*, *user_id_str*, *username*.

Data yang telah dikumpulkan kemudian disimpan dalam format CSV dan digunakan dalam proses *Pre-processing* data sebelum dilakukan analisis sentimen. Data hasil *scraping* ditampilkan di tabel II.

TABEL II.
DATASET

NO	<i>created_at</i>	...	<i>full_text</i>	...
1	Tue Jan 21 02:50:23 +0000 2025	...	Udah ngerasain #KaburAjaDulu trus balik bentar...	
2	Tue Jan 21 11:10:11 +0000 2025	...	https://t.co/mC0g1FJSgk Sambil grinding buat	...

3	Tue Jan 21 02:29:09 +0000 2025	...	@twisterfruity Search hashtag #kaburajadulu	...
4	Tue Jan 21 16:46:13 +0000 2025	...	@cattominyg @tanyakanrl biar bisa make hastag
5	Tue Jan 21 10:45:47 +0000 2025	...	Apakah di sini ada yang butuh info apply visa

B. Pre-processing

Tahap pertama dalam proses *pre-processing* ini adalah dengan menghapus semua kolom data kecuali kolom *full_text*. Kolom yang dihapus berupa kolom, *conversation_id_str*, *'created_at'*, *'favorite_count'*, *'id_str'*, *'image_url'*, *'in_reply_to_screen_name'*, *'lang'*, *'location'*, *'quote_count'*, *'reply_count'*, *'retweet_count'*, *'tweet_url'*, *'user_id_str'*, *'username'*]. Lalu mengubah teks menjadi *lowercase* seperti yang ditampilkan di tabel 2.

TABEL III.
PENGHAPUSAN KOLOM

NO	<i>full_text</i>
1	udah ngerasain #kaburajadulu trus balik bentar...
2	@twisterfruity search hashtag #kaburajadulu
3	@cattominyg @tanyakanrl biar bisa make hastag ...
4	bismillah #kaburajadulu ke jepang kalo betah l...
5	udah ga bisa #kaburajadulu

Selanjutnya proses mencari data duplikasi dan menghapus data duplikasi dilakukan. Dari 4768 data terdapat 184 data duplikat. Jumlah data yang ada setelah proses tersebut adalah 4584 data yang siap di oleh di proses selanjutnya.

Tahapan *pre-processing* yang akan dilaksanakan terdiri dari Langkah utama berikut:

1. Data Cleaning

Tahap selanjutnya ada *data cleaning*, proses tersebut berupa karakter khusus seperti link, emoji, tanda baca, dan simbol-simbol yang tidak relevan, contohnya seperti mengubah kalimat "#KaburAjaDulu" menjadi kalimat biasa "kabur aja dulu" agar teks lebih bersih dan siap dianalisis. Seperti dijelaskan dalam tabel 3.

TABEL IV.
HASIL DATA CLEANING

NO	<i>full_text</i>	<i>clean_text</i>
1	udah ngerasain #kaburajadulu trus balik bentar	udah ngerasain kaburajadulu trus balik bentar nyesel
2	https://t.co/mc0glfjsgk sambil grinding buat persiapan kaburajadulu. all the best for all of you!	sambil grinding buat persiapan kaburajadulu all the best for all of you

3	@twisterfruity search hashtag #kaburajadulu	search kaburajadulu	hashtag
---	--	------------------------	---------

2. Tokenization

Pada tahap ini sistem akan memecah setiap teks *tweet* yang telah dibersihkan menjadi potongan-potongan kata (token), seperti kalimat "*search hashtag kaburajadulu*" menjadi pecahan kata berupa [*'search'*, *'hashtag'*, *'kaburajadulu'*] proses ini memudahkan analisis lebih lanjut seperti penghitungan frekuensi kata atau klasifikasi sentimen.

Tabel dibawah akan menjelaskan hasil dari proses *tokenization*.

TABEL V.
HASIL TOKENIZATION

NO	<i>Clean Text</i>	<i>Token</i>
1	udah ngerasain kaburajadulu trus balik bentar nyesel..	['udah', 'ngerasain', 'kaburajadulu', 'trus', 'balik', 'bentar'...]
2	sambil grinding buat persiapan kaburajadulu all the best for all of you	['sambil', 'grinding', 'buat', 'persiapan', 'kaburajadulu', 'all', 'the', 'best', 'for', 'all', 'of', 'you']
3	search hashtag kaburajadulu	['search', 'hashtag', 'kaburajadulu']

3. Stopword Removal

Dalam penelitian ini dilakukan untuk menghapus kata-kata umum yang tidak memiliki makna penting, seperti "yang", "dan", atau "di", agar analisis fokus pada kata-kata yang lebih relevan terhadap sentiment.

Tabel dibawah akan menjelaskan hasil dari proses *Stopword Removal*.

TABEL VI.
HASIL STOPWORD REMOVAL

NO	<i>Token</i>	<i>Stopword removal</i>
1	['udah', 'ngerasain', 'kaburajadulu', 'trus', 'balik', 'bentar', 'nyesel'...]	['udah', 'ngerasain', 'kaburajadulu', 'trus', 'bentar', 'nyesel',...]
2	['sambil', 'grinding', 'buat', 'persiapan', 'kaburajadulu', 'all', 'the', 'best', 'for', 'all', 'of', 'you']	['grinding', 'persiapan', 'kaburajadulu', 'all', 'the', 'best', 'for', 'all', 'of', 'you']
3	['search', 'hashtag', 'kaburajadulu']	['search', 'hashtag', 'kaburajadulu']

4. Remove Slang

Proses *remove slang* bertujuan untuk mengganti kata-kata tidak baku menjadi bentuk baku menggunakan kamus slang agar teks lebih konsisten dan mudah dipahami oleh model analisis sentimen. Misalnya, kata "udah" diubah menjadi "sudah" dan "gw" menjadi "gue".

Tabel dibawah akan menjelaskan hasil dari proses *Remove Slang*.

TABEL VII.
HASIL REMOVE SLANG

NO	Stopword removal	remove slang
1	['udah', 'ngerasain', 'kaburajadulu', 'trus', 'bentar', 'nyesel', 'bersyukur', 'gw', 'indo', 'sekarat',...]	['sudah', 'merasai', 'kaburajadulu', 'terus', 'bentar', 'menyesal', 'bersyukur', 'gue', 'indonesia', 'sekarat',...]
2	['grinding', 'persiapan', 'kaburajadulu', 'all', 'the', 'best', 'for', 'all', 'of', 'you']	['grinding', 'persiapan', 'kaburajadulu', 'all', 'the', 'best', 'for', 'all', 'of', 'you']
3	['search', 'hashtag', 'kaburajadulu']	['search', 'hashtag', 'kaburajadulu']

5. Stemming

Proses *stemming* dalam penelitian ini dilakukan untuk mengubah kata turunan menjadi bentuk dasarnya (kata dasar) menggunakan library Sastrawi, sehingga memperkecil variasi kata yang memiliki makna sama. Fungsi *optimized_stemming()* digunakan untuk mempercepat proses dengan menyimpan hasil stemming dalam *cache*. Misalnya, kata "bermain" diubah menjadi "main", atau "menyukai" menjadi "suka".

TABEL VIII.
HASIL STEMMING

NO	stopword	stemming
1	[udah, ngerasain, kaburajadulu, trus, bentar, ...]	[udah, ngerasain, kaburajadulu, trus, bentar, ...]
2	[grinding, persiapan, kaburajadulu, all, the, ...]	[grinding, siap, kaburajadulu, all, the, best, ...]
3	[search, hashtag, kaburajadulu]	[biar, make, hastag, kaburajadulu]

6. Translation

Proses *translation* dalam penelitian ini bertujuan untuk menerjemahkan hasil *stemming* dari Bahasa Indonesia ke Bahasa Inggris menggunakan library *GoogleTranslator* dari *deep_translator*. Setiap daftar token yang telah di sistem di gabung menjadi satu kalimat, lalu diterjemahkan secara otomatis ke dalam bahasa Inggris.

Langkah ini penting karena proses pelabelan sentimen menggunakan *textblob*, karena *textblob* hanya mendukung teks berbahasa Inggris. Contohnya, daftar token seperti ['suka', 'tinggal', 'luar', 'negeri'] akan diterjemahkan menjadi kalimat "*like living abroad*", sehingga model dapat melakukan analisis sentimen secara akurat dalam konteks bahasa yang didukung.

TABEL IX.
HASIL TRANSLATE

NO	stemming	translated en
1	['udah', 'ngerasain', 'kaburajadulu', 'trus', 'bentar', 'nyesel', 'syukur', 'gw', 'indo', ...]	I have felt Kawujadulu then for a while, I regret it, I am gratitude,...

2	['grinding', 'siap', 'kaburajadulu', 'all', 'the', 'best', 'for', 'all', 'of', 'you']	grinding siap kaburajadulu all the best for all of you
3	['biar', 'make', 'hastag', 'kaburajadulu']	Let Make Hastag Kajurajadulu

C. Labelling

Setelah melalui proses *pre-processing* selanjutnya adalah proses *labelling*. Proses ini akan dilakukan dengan library *textblob*. Hasil dari kolom *translated_en* akan dihitung nilai polaritasnya yang menunjukkan kecenderungan emosi dalam teks, yaitu positif (> 0), negatif (< 0), atau netral ($= 0$). Proses ini dilakukan dengan menerapkan fungsi *get_sentiment()* pada kolom *translated_en*, kemudian hasil klasifikasi sentimen disimpan dalam kolom baru bernama sentimen.

TABEL X.
HASIL LABELING

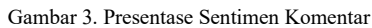
NO	Tweet	Polarity	Sentiment
1	I have felt Kaburajadulu then for a while, I regret it, I am gratitude, I am a dying of the end of the death, not if I get immigration for whatever weird coincidences challenge	-0,5	Negatif
2	grinding siap kaburajadulu all the best for all of you	1	Positif
3	Search hashtag kaburajadulu	0	Netral

Dari tabel diatas dapat dilihat bahwa tweet "I have felt Kaburajadulu then for a while, I regret it, I am gratitude, I am a dying of the end of the death, not if I get immigration for whatever weird coincidences challenge" memiliki polaritas -0,5 yang berarti oleh *textblob* dilabelkan menjadi negatif. Kata "death" yang berarti mati juga jelas menjadi kata negatif. Setelah proses pelabelan selesai, selanjutnya jumlah data sesuai kategori bisa dilihat di tabel bawah:

TABEL XI.
JUMLAH SENTIMEN

NO	Sentiment	Jumlah
1	Positif	1168
2	Negatif	615
3	Netral	2820

Tabel XI menunjukkan distribusi jumlah data sentimen dalam dataset. Dapat dilihat bahwa mayoritas tweet bersentimen netral, yaitu sebanyak 2.820 tweet (sekitar 59,7%). Sementara itu, tweet dengan sentimen positif berjumlah 1.168 (24,7%) dan negatif hanya 615 (13%). Ketidakseimbangan ini berpotensi menyebabkan bias model dalam klasifikasi, terutama terhadap kelas minoritas. Oleh karena itu, pada tahap praproses dilakukan penyeimbangan data menggunakan teknik *oversampling* (SMOTE) untuk memastikan model tidak hanya terlatih pada mayoritas (netral), tetapi juga memiliki kemampuan representatif terhadap sentimen positif dan negatif.



D. WordCloud

[illegible]

Gambar 4. Wordcloud Sentimen Positif

[illegible]

Gambar 5. Wordcloud Sentimen Negatif

WordCloud - Sentimen Netral

A word cloud visualization of neutral sentiment words. The most prominent words are "Indonesia", "Jepang", "Korea", "Negeri", "Jalur", "Jepang", "Korea", "Negeri", "Jalur", "Jepang", "Korea", "Negeri", "Jalur". Other visible words include "Jepang", "Korea", "Negeri", "Jalur", "Jepang", "Korea", "Negeri", "Jalur".

Gambar 6. Wordcloud Sentimen Netral

E. Feature Extraction (TF-IDF)

Pada tahap ini, teks diubah menjadi bentuk numerik menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). Metode ini menghitung seberapa penting suatu kata dalam sebuah dokumen dibandingkan dengan seluruh dokumen lainnya. Proses dilakukan dalam dua langkah: pertama, menghitung frekuensi kemunculan kata menggunakan *CountVectorizer()*, lalu mengubahnya menjadi bobot TF-IDF dengan *TfidfTransformer()*.

Hasilnya adalah matriks TF-IDF berukuran (4584, 9063), yang menunjukkan 4584 dokumen dan 9063 kata unik. Karena sebagian besar nilainya nol, data disimpan dalam format hemat memori (*Compressed Sparse Row*). Nilai-nilai TF-IDF menunjukkan kontribusi kata tertentu terhadap isi

dokumen, yang digunakan sebagai masukan untuk model pembelajaran mesin.

TABEL XII.
HASIL PROSES TF-IDF

TF-IDF Matrix Shape	
Shape	(4584, 9063)
Tipe Matriks	<class 'scipy.sparse._csr.csr_matrix'>
TF-IDF Matrix - Sparse Format (Koordinat & Value)	
Coords	Values
(0, 814)	0.2553
(0, 1331)	0.2253
(0, 1494)	0.2678
(0, 2024)	0.2465
(0, 2275)	0.2553
(0, 2804)	0.1180
(0, 3184)	0.2294
(0, 3613)	0.1192
(0, 3661)	0.2397
(0, 3681)	0.1309

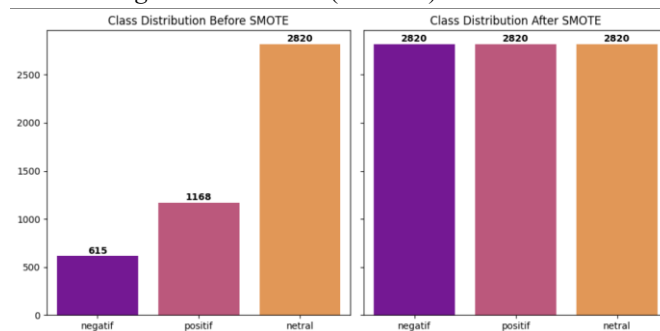
Dari tabel 7 dapat diketahui bahwa setiap baris menunjukkan kombinasi antara indeks dokumen dan indeks kata unik, serta nilai TF-IDF-nya. Nilai TF-IDF yang lebih tinggi menandakan bahwa kata tersebut memiliki kontribusi besar dalam membentuk konteks isi dokumen tersebut. Dengan bobot ini, fitur teks menjadi lebih representatif sehingga dapat digunakan secara optimal dalam algoritma klasifikasi sentimen seperti SVM.

F. Validasi model menggunakan K-Fold

Proses pengujian dengan metode *5-fold Stratified Cross-Validation* untuk mengevaluasi stabilitas performa dan kemampuan generalisasi model SVM dalam klasifikasi sentimen. Proses ini dilakukan dengan cara membagi seluruh data menjadi lima bagian (*fold*) yang proporsional berdasarkan distribusi kelas (*stratified*), di mana setiap *fold* secara bergantian digunakan sebagai data uji, sementara empat *fold* lainnya digunakan sebagai data latih. Dengan demikian, seluruh data memperoleh kesempatan yang adil untuk digunakan sebagai data uji dalam siklus pelatihan dan pengujian model.

Selama proses pelatihan pada tiap *fold*, dilakukan penanganan ketidak seimbangan kelas menggunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*). SMOTE bekerja dengan cara mensintesis data baru untuk kelas minoritas dalam data latih agar distribusi antar kelas menjadi seimbang, sehingga model tidak bias terhadap kelas mayoritas.

G. Handling Imbalance data (SMOTE)



Gambar 7. Hasil Proses SMOTE Sebelum Dan Sesudah Proses.

Gambar 7 menunjukkan distribusi jumlah data pada masing-masing kelas sentimen sebelum dan sesudah penerapan metode *Synthetic Minority Oversampling Technique* (SMOTE). Sebelum dilakukan *balancing data*, data sangat tidak seimbang dengan dominasi kelas netral. Hal ini berpotensi menyebabkan model hanya fokus mengenali pola dari kelas mayoritas dan mengabaikan kelas minoritas.

Setelah penerapan SMOTE, distribusi antar kelas menjadi seimbang, dengan masing-masing kelas (positif, negatif, dan netral) memiliki jumlah data yang sama, yaitu 2820. *Oversampling* ini dilakukan secara eksklusif hanya pada data latih (*training set*) untuk mencegah *data leakage* dan menjaga keobjektifan evaluasi model.

Evaluasi kinerja model sebelum dan sesudah SMOTE menunjukkan peningkatan akurasi dan *F1-score* yang signifikan, khususnya pada kelas minoritas. Peningkatan ini mengindikasikan bahwa SMOTE berperan penting dalam meningkatkan performa model, terutama dalam mengenali data dari kelas positif dan negatif yang sebelumnya *under represented*. Oleh karena itu, dapat disimpulkan bahwa peningkatan performa model sebagian besar didominasi oleh hasil dari *oversampling* yang berhasil menyamakan distribusi data antar kelas.

H. Implementasi Metode Support Vector Machine

TABEL XIII.
HASIL PROSES IMPLEMENTASI SVM

No	Kernel	Spilt	Accuracy
1	Linear	80 : 20	0.852781
2		70 : 30	0.835029
3		60 : 40	0.822792
4	Sigmoid	80 : 20	0.844057
5		70 : 30	0.830669
6	Polynomial	60 : 40	0.815703
7		80 : 20	0.714286
8		70 : 30	0.704215
		60 : 40	0.686478

Hasil evaluasi menunjukkan bahwa SVM dengan kernel *Linear* dan rasio split 80:20 memberikan hasil akurasi terbaik dibandingkan kombinasi lainnya, yaitu sebesar 0.852781. Sementara itu, SVM dengan kernel Sigmoid juga

menunjukkan performa cukup baik dengan akurasi tertinggi pada rasio split yang sama sebesar 0.844057. Sebaliknya, kernel Polynomial menghasilkan akurasi yang relatif lebih rendah di semua rasio split. Berdasarkan hasil tersebut, dapat disimpulkan bahwa SVM *Linear* dengan rasio split 80:20 merupakan pilihan terbaik untuk digunakan pada proses klasifikasi sentimen ini.

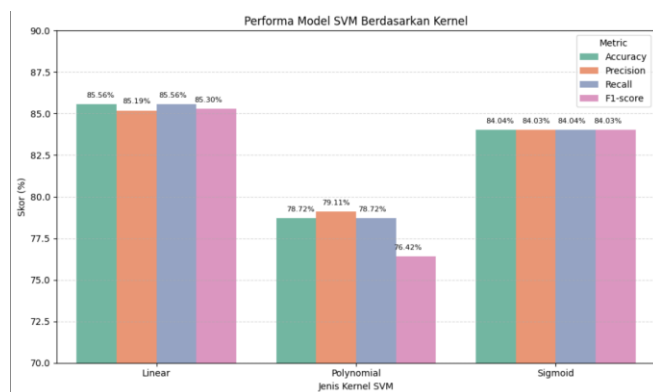
I. Hyperparameter Tuning (Grid Search)

Proses *hyperparameter tuning* menggunakan *Grid Search* dalam penelitian ini bertujuan untuk menemukan kombinasi parameter terbaik bagi masing-masing model SVM dengan kernel *linear*, *polynomial*, dan *sigmoid*. Setiap model diuji dengan beberapa nilai parameter seperti *c*, *degree*, dan *gamma* yang telah didefinisikan dalam *param_grid*. *GridSearchCV* melakukan pencarian menyeluruh terhadap semua kombinasi parameter dengan evaluasi menggunakan *cross-validation* untuk menghindari *overfitting* dan meningkatkan keakuratan.

Setelah *Grid Search* menemukan parameter terbaik berdasarkan *CV accuracy*, model dengan parameter optimal diuji kembali menggunakan data latih dan data uji dengan pembagian 80:20. Hasil pengujian disetiap parameter bisa dilihat di tabel- tabel dibawah.

TABEL XIV.
KERNEL UJI

NO	Kernel	Parameter	Value
1	Linear	C	(0.1, 1, 10, 100)
2	Polynomial	C	(0.1, 1, 10, 100)
		Degree	(2,3)
		Gamma	Scale
3	Sigmoid	C	(0.1, 1, 10, 100)
		Gamma	Scale



Gambar 8. Hasil Performa Antar Kernel

Gambar 8 memperlihatkan perbandingan performa tiga model SVM dengan kernel yang berbeda Linear, Polynomial, dan *Sigmoid* berdasarkan hasil tuning menggunakan metode *Grid Search*. Metrik yang digunakan meliputi *Accuracy*, *Precision*, *Recall*, dan *F1-score*, dengan data yang telah melalui proses *balancing* menggunakan SMOTE.

Hasil menunjukkan bahwa SVM dengan kernel Linear menghasilkan performa terbaik secara konsisten di seluruh

metrik, dengan *accuracy* dan *recall* sebesar 85.56%, serta F1-score mencapai 85.30%. Hal ini mengindikasikan bahwa kernel linear mampu memisahkan data dengan margin optimal secara efisien tanpa *overfitting*, menjadikannya cocok untuk klasifikasi sentimen pada dataset ini yang kemungkinan besar memiliki fitur yang cukup terpisah secara linear.

Kernel *Sigmoid* menempati posisi kedua, dengan metrik yang relatif stabil dan hanya terpaut sekitar 1.5% di bawah linear. Ini menunjukkan bahwa kernel *sigmoid* juga mampu menangkap pola non-linear, namun tidak secara signifikan mengungguli kernel linear dalam kasus ini.

Sebaliknya, kernel Polynomial menghasilkan performa paling rendah dengan F1-score hanya 76.42%. Hal ini kemungkinan disebabkan oleh kompleksitas tinggi dari kernel *polynomial* yang kurang cocok untuk struktur distribusi fitur pada data TF-IDF, sehingga berpotensi menyebabkan *overfitting* atau ketidak efisienan dalam memisahkan kelas.

Dengan mempertimbangkan hasil ini, dapat disimpulkan bahwa kernel linear merupakan pilihan paling optimal untuk klasifikasi sentimen pada kasus ini. Selain performa yang unggul, kernel linear juga lebih cepat dalam proses pelatihan dan interpretasinya lebih sederhana secara matematis dibanding kernel non-linear lainnya.

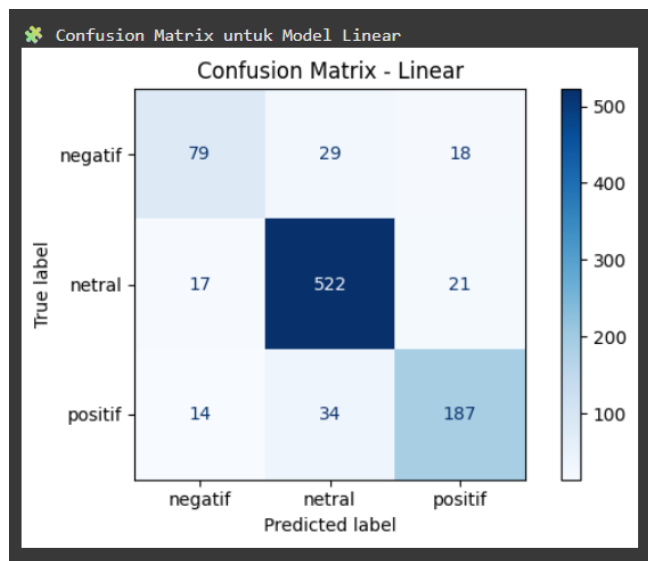
J. Evaluasi Model

Pada tahap ini merupakan ahir dari proses penelitian. Pengujian dibawah merupakan proses pengujian yang paling optimal dibandingkan yang lain. Tabel dibawah menyajikan hasil dari proses penelitian yang dilakukan.

TABEL XV.
HASIL DARI EVALUASI MODEL

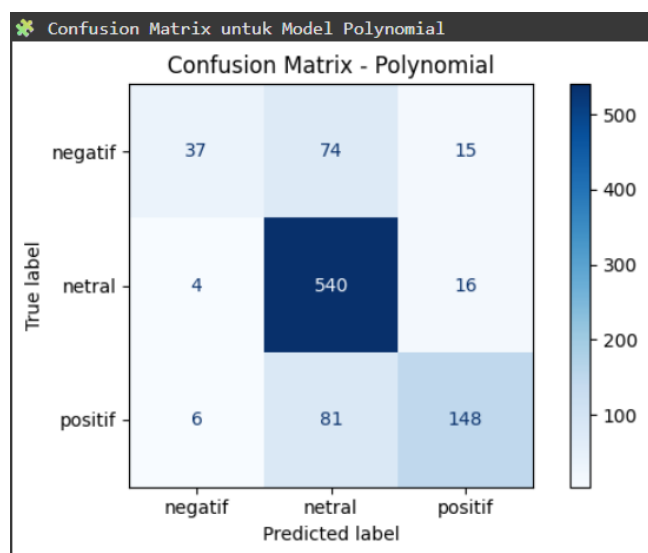
NO	Model	Best Params	Accuracy
1	SVM_Linear	{'C': 10, 'kernel': 'linear'}	0.8556
2	SVM_Polynomial	{'C': 10, 'degree': 2, 'gamma': 'scale', 'kernel': 'poly'}	0.7872
3	SVM_Sigmoid	{'C': 1, 'gamma': 'scale', 'kernel': 'sigmoid'}	0.8404

Setelah melakukan pengujian disemua kernel didapatkan hasil akurasi paling tinggi dari setiap parameter yang sudah di cantumkan di tabel XV diatas. Dari tabel diatas dapat dilihat bahwa model dengan SVM *linear* dengan parameter *c* : 10, menggunakan karnel linear menjadi model yang paling bagus dan mendapatkan akurasi tertinggi yaitu 0. 8556.



Gambar 9. Confusion matrix SVM linear

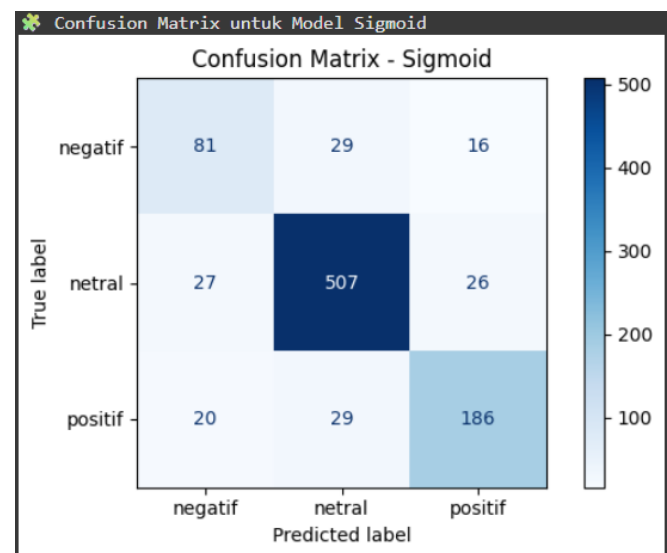
Gambar 9 menunjukkan hasil klasifikasi sentimen menggunakan model *Support Vector Machine* (SVM) dengan kernel linear. Model ini mampu mengklasifikasikan sentimen netral dengan sangat baik, terbukti dari 522 *tweet* yang berhasil diprediksi secara tepat. Untuk sentimen negatif, terdapat 79 *tweet* yang diklasifikasikan dengan benar, sementara untuk sentimen positif terdapat 187 *tweet* yang sesuai. Meski demikian, model masih mengalami kesalahan klasifikasi, seperti 29 *tweet* negatif yang diklasifikasikan sebagai netral, serta 34 *tweet* positif yang juga salah diprediksi sebagai netral.



Gambar 10. Confusin Matrix SVM Polynomial

Gambar 10 menunjukkan hasil klasifikasi sentimen menggunakan model SVM dengan kernel *polynomial*. Model ini mampu mengklasifikasikan 37 *tweet* sebagai sentimen negatif, 540 *tweet* sebagai sentimen netral, dan 148 *tweet* sebagai sentimen positif secara benar. Namun, model juga

melakukan beberapa kesalahan klasifikasi, di antaranya 74 *tweet* negatif salah diklasifikasikan sebagai netral, dan 15 *tweet* negatif salah diklasifikasikan sebagai positif. Untuk kelas netral, terdapat 4 *tweet* yang diklasifikasikan sebagai negatif dan 16 *tweet* sebagai positif. Sedangkan pada kelas positif, sebanyak 6 *tweet* salah diklasifikasikan sebagai negatif dan 81 *tweet* sebagai netral.

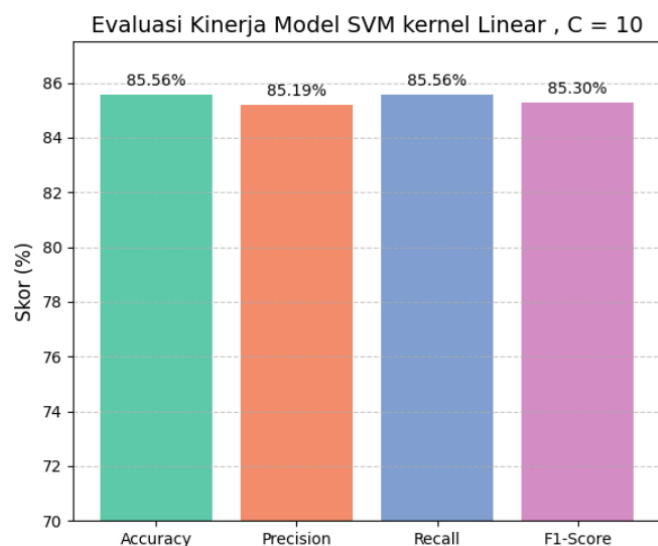


Gambar 11. Confusion Matrix hasil SVM Sigmoid

Gambar 11 menunjukkan hasil klasifikasi sentimen menggunakan model SVM dengan kernel *sigmoid*. Model ini mampu mengklasifikasikan 81 *tweet* sebagai sentimen negatif, 507 *tweet* sebagai netral, dan 186 *tweet* sebagai positif dengan benar. Namun, model juga melakukan sejumlah kesalahan, yaitu 29 *tweet* negatif salah diklasifikasikan sebagai netral dan 16 sebagai positif. Untuk kelas netral, terdapat 27 *tweet* yang salah diklasifikasikan sebagai negatif dan 26 sebagai positif. Sementara itu, pada kelas positif, 20 *tweet* diklasifikasikan keliru sebagai negatif dan 29 sebagai netral.

Berdasarkan hasil *confusion matrix* dari ketiga jenis kernel SVM linear, *polynomial*, dan *sigmoid* terlihat bahwa ketiganya menunjukkan kecenderungan kuat untuk mengklasifikasikan *tweet* ke dalam kelas **netral**. Hal ini ditunjukkan oleh jumlah prediksi benar yang paling tinggi berada pada sentimen netral, serta sebagian besar kesalahan klasifikasi dari kelas negatif dan positif yang dialihkan ke kelas netral. Salah satu alasan utama kecenderungan ini adalah karena *tweet* dengan sentimen netral secara leksikal cenderung menggunakan kosakata umum, informatif, dan tidak memuat ekspresi emosional yang ekstrem. Dalam representasi vektor TF-IDF yang digunakan dalam model, fitur dari *tweet* netral cenderung lebih konsisten dan lebih mudah dipisahkan dibandingkan *tweet* positif atau negatif yang seringkali lebih bervariasi secara kontekstual. Selain itu, tumpang tindih leksikal antara *tweet* netral dengan kelas lain juga menyebabkan model kesulitan dalam membedakan dengan tegas, sehingga model cenderung memilih kelas netral saat

menghadapi ambiguitas. Meskipun proses *balancing* data telah dilakukan dengan SMOTE, pola distribusi kata dan struktur kalimat netral yang lebih “aman” secara semantik menjadikan kelas ini lebih mudah dikenali oleh model dalam berbagai jenis kernel.



Gambar 12. Hasil Evaluasi Model Terbaik

Setelah melalui berbagai proses, hasil terbaik dalam proses penelitian ini adalah SVM dengan karnel linear, $c = 10$ dengan akurasi 85.56%.

IV. KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa metode *Support Vector Machine* (SVM) efektif digunakan untuk analisis sentimen pada isu “Kabur Aja Dulu” di media sosial *Twitter* (X), dengan menerapkan tahapan *pre-processing* data, ekstraksi fitur menggunakan TF-IDF, penanganan data tidak seimbang melalui SMOTE yang diterapkan hanya pada data latih untuk menghindari *data leakage*, serta evaluasi performa model menggunakan *5-Fold Cross Validation* dan *Grid Search* untuk *hyperparameter tuning*. Hasil menunjukkan bahwa model SVM dengan kernel linear dan parameter $C=10$ memberikan performa terbaik dengan akurasi sebesar 85.56%, *precision* 85.19%, *recall* 85.56%, dan *f1-score* 85.30%, dibandingkan kernel *polynomial* dan *sigmoid* yang memiliki performa lebih rendah. Temuan ini menunjukkan bahwa kombinasi antara TF-IDF, SMOTE, dan SVM linear mampu menghasilkan model yang akurat dan stabil dalam mengklasifikasikan sentimen meskipun data bersifat tidak seimbang. Untuk penelitian selanjutnya, disarankan untuk menggunakan metode pelabelan sentimen berbasis *supervised learning* daripada *TextBlob*, agar hasil pelabelan lebih sesuai dengan karakteristik bahasa Indonesia dan konteks sosial yang dianalisis. Selain itu, pengembangan dapat dilakukan dengan memperluas data, menerapkan model *deep learning* seperti LSTM atau BERT, serta melakukan analisis aspek

sentimen agar hasil lebih detail dan relevan terhadap isu yang sedang dibahas.

DAFTAR PUSTAKA

- [1] Putri Sari Margaret Julianty Silaban, Diya Mirza, Nida Nafilah, and Surya Zulfachrinal Tanjung, “Menghadapi Ancaman Nasionalisme Disintegrasi Bangsa di Tengah Trend Kabur Aja Dulu,” *Jurnal Bintang Pendidikan Indonesia*, vol. 3, no. 2, pp. 193–199, Mar. 2025, doi: 10.55606/jubpi.v3i2.3821.
- [2] N. Abelia et al., “Dampak Framing Tagar #Kaburajadulu Terhadap Opini Publik dan Kebijakan Sosial di Indonesia,” *Filosofi Publikasi Ilmu Komunikasi, Desain, Seni Budaya*, vol. 2, pp. 71–77, May 2025, doi: 10.62383/filosofi.v2i2.577.
- [3] Meylia Arifah Salsa et al., “Eksplorasi Dinamika Sosial Melalui Analisis Sentimen Tagar #KaburAjaDulu di Media Sosial Tiktok,” *SABER: Jurnal Teknik Informatika, Sains dan Ilmu Komunikasi*, vol. 3, no. 2, pp. 238–251, May 2025, doi: 10.59841/saber.v3i2.2631.
- [4] D. Irenniza, A. Putri, F. T. Saputra, and R. Hardiyanti, “Pengaruh Penggunaan Media Sosial Twitter Terhadap Pemenuhan Kebutuhan Informasi (Survei Terhadap Pengikut Akun @Habisonontonfilm),” *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 8, pp. 410–418, 2024, doi: 10.5281/zenodo.11107309.
- [5] M. Aziz Assuja, “Analisis Sentimen Tweet Menggunakan Backpropagation Neutral Network,” *Jurnal TEKNOINFO*, vol. 10, no. 2, pp. 1693–1703, 2016.
- [6] R. Puspita and T. N. Suciati, “Mobile Phone dan Media Sosial: Pnggunaan dan Tantangannya pada Jurnalisme Online Indonesia,” *Jurnal Ilmu Komunikasi*, vol. 3, no. 2, pp. 2656–050, 2020, [Online]. Available: <http://ejournal.upnvj.ac.id/index.php/JEP/index>
- [7] S. A. Mahira, I. Sukoco, C. S. Barkah, N. Jamil, A. Novel, and J. A. Bisnis, “Teknologi Artificial Intelegence Dalam Analisis Sentimen: Studi Literatur Pada Perusahaan Kata.ai,” *Bulan Agustus Tahun*, vol. 6, no. 2, pp. 139–148, Aug. 2023.
- [8] Irma Surya Kumala Idris, Yasin Aril Mustofa, and Irvan Abraham Salihi, “Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)boards,” *Jambura Journal of Electrical and Electronics Engineering*, vol. 5, no. 1, pp. 823–848, Jan. 2023, Accessed: Apr. 29, 2025. [Online]. Available: <https://ejournal.ung.ac.id/index.php/jjee/article/view/16830/5678>
- [9] A. Mudya Yolanda and R. Tri Mulya, “Implementasi Metode Support Vector Machine untuk Analisis Sentimen pada Ulasan Aplikasi Sayurbox di Google Play Store,” *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, vol. 6, no. 2, pp. 76–83, 2024, doi: 10.35580/variansiunm258.
- [10] R. Damasela, B. P. Tomasouw, and Z. A. Leleury, “Penerapan Metode Support Machine Learning (SVM) Untuk Mendeteksi Penyalahgunaan Narkoba,” *PARAMETER: Jurnal Matematika, Statistika dan Terapannya*, vol. 1, no. 2, pp. 111–122, Oct. 2022, doi: 10.30598/parameterv1i2pp111-122.
- [11] Hidayatunnisa, Kusri, and Kusnawi, “Perbandingan Kinerja Metode Naive Bayes dan Support Vector Machine dalam Analisis Soal,” *Agustus*, vol. 13, no. 2, pp. 173–180.
- [12] D. Haliza and M. Ikhsan, “Sentiment Analysis on Public Perception of the Nusantara Capital on Social Media X Using Support Vector Machine (SVM) and K-Nearest Neighbor (K-NN) Methods,” 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [13] I. Kurniawan et al., “Perbandingan Algoritma Naive Bayes Dan SVM Dalam Sentimen Analisis Marketplace Pada Twitter,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, no. 1, 2023, [Online]. Available: <http://jurnal.mdp.ac.id>
- [14] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, “Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine,” vol. 6, no. 3, 2021, [Online]. Available: <http://situs.com>

- [15] N. Muchammad Shiddieqy Hadna, P. Insap Santosa, and W. Wahyu Winarno, "Studi Literatur Tentang Perbandingan Metode Untuk Proses Analisa Sentimen Di Twitter," Mar. 2016.
- [16] M. I. Fikri, T. S. Sabrila, Y. Azhar, and U. M. Malang, "Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter," vol. 10, p. 2, Dec. 2020.
- [17] F. Abdusyukur, "Penerapan Algoritma Support Vector Machine (SVM) Untuk Klasifikasi Pencemaran Nama Baik di Media Sosial Twitter," KOMPUTA: Jurnal Ilmiah Komputer dan Informatika, vol. 12, no. 1, 2023.
- [18] A. Amelia, L. N. Hayati, and H. Darwis, "Analisis Sentimen Masyarakat Terhadap Sistem Pembayaran MyPertamina dengan Metode Random Forest, SVM, dan Naïve Bayes," Literatur Informatika & Komputer, vol. 1, no. 1, pp. 28–44, 2024, doi: 10.33096/linier.v1i1.2269.
- [19] N. A. Lestari et al., "Metode Naive Bayes Classifier Dengan Textblob Untun Analisis Sentimen Terhadap Pelayanan Indihome dan First Media," Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK), vol. 4, no. 1, Sep. 2020, [Online]. Available: <https://t.co/Ws2wOyU5kz>
- [20] R. Parlita, S. Ilham Pradika, A. M. Hakim, and R. N. M. Kholilul, "Analisis Sentimen Twitter Terhadap Bitcoin dan Cryptocurrency Berbasis Python TextBlob," 2020. [Online]. Available: <https://t.co/QaUW3P2TKc>
- [21] A. Erlangga, Y. P. Astuti, E. Kartikadarma, S. Rakasiwi, and E. R. Subhiyakto, "Penggunaan Algoritma Naïve Bayes dengan Polarity Textblob untuk Analisis Sentimen pada Acara ASEAN CUP 2024 U-16 di Media Sosial Twitter," Jurnal Sains dan Teknologi Informasi, vol. 3, no. 1, pp. 177–189, 2025, doi: 10.62951/switch.v3i1.357.
- [22] A. Fariz Zulhilmil, R. Setya Perdana, U. Brawijaya, and P. Korespondensi, "Pengenalan Entitas Bernama Menggunakan BI-LSTM Pada Chatbot Bahasa Indonesia Named Entity Recognition Using BI-LSTM in Indonesian Language Chatbot," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), vol. 11, no. 5, 2024, doi: 10.25126/jtiik.2024117968.
- [23] G. R. Putri, M. A. Maulana, and S. Bahri, "Perbandingan Algoritma Naïve Bayes dan TextBlob Untuk Mendapatkan Analisis Sentimen Masyarakat Pada Sosial Media," Teknika, vol. 13, no. 2, pp. 213–218, Jun. 2024, doi: 10.34148/teknika.v13i2.815.
- [24] M. H. Al-Areef and K. Saputra, "Analisis Sentimen Pengguna Twitter Mengenai Calon Presiden Indonesia Tahun 2024 Menggunakan Algoritma LSTM," Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer), vol. 22, pp. 270–279, Aug. 2023, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jis/index>
- [25] P. Agusia, M. Uli, A. Manurung, V. Calista, and V. C. Mawardi, "Pemanfaatan Word Cloud Pada Analisis Sentimen Dalam Menggali Persepsi Publik," Seminar Nasional Corisindo, Jun. 2024.
- [26] Z. Alhaq, A. Mustopa, and J. D. Santoso, "Penerapan Metode Support Machine Learning Untuk Analisis Sentimen Pengguna Twitter," Jurnal Of Information System Management, vol. 3, no. 3, 2021.
- [27] T. Ridwansyah, "KLIK: Kajian Ilmiah Informatika dan Komputer Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier," Media Online, vol. 2, no. 5, pp. 178–185, 2022, [Online]. Available: <https://djournals.com/klik>
- [28] M. Rafly Gusmansyah and H. Hendrawan, "Peningkatan Kinerja Analisis Sentimen pada Ulasan Aplikasi Identitas Kependudukan Digital (IKD) di Indonesia Menggunakan Algoritma Support Vector Machine (SVM) dan SMOTE under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)," vol. 10, no. 1, pp. 2541–1179, 2025, [Online]. Available: <https://journal.uin-alauddin.ac.id/index.php/instek>
- [29] N. Fajriyah, N. T. Lapatta, D. W. Nugraha, and R. Laila, "Implementasi SVM dan SMOTE Pada Analisis Sentimen Pada Media Sosial X Terhadap Pelantikan Agus Harimurti Yudhoyono," JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), vol. 10, no. 2, pp. 1359–1370, Mar. 2025, doi: 10.29100/jipi.v10i2.6246.
- [30] J. T. Kumalasari and I. Puspitorini, "Perbandingan Metode Klasifikasi dan SMOTE Terhadap Analisa Sentimen Mobil Listrik Indonesia," Jurnal Minfo Polgan, vol. 13, no. 2, pp. 2257–2268, Jan. 2025, doi: 10.33395/jmp.v13i2.14428.
- [31] A. Pradhan and W. : Wwv, "Support vector machine-A survey International Journal of Emerging Technology and Advanced Engineering Support Vector Machine-A Survey," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 8, Jul. 2012, [Online]. Available: www.ijetae.com
- [32] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search," Sistemasi: Jurnal Sistem Informasi, vol. 11, no. 2, pp. 2540–9719, May 2022, [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [33] Anugerah Simanjuntak et al., "Research and Analysis of IndoBERT Hyperparameter Tuning in Fake News Detection," Jurnal Nasional Teknik Elektro dan Teknologi Informasi, vol. 13, no. 1, pp. 60–67, Feb. 2024, doi: 10.22146/jnteti.v13i1.8532.