

# Perbandingan Algoritma SVM dan RF pada Analisis Sentimen menggunakan Pendekatan Machine Learning

Tristanto Ariaaji <sup>\*1</sup>, Sri Ngudi Wahyuni<sup>2</sup>, Muhammad Ikshan <sup>3</sup>

<sup>\*1</sup> Universitas Amikom Yogyakarta; tristanto.a@amikom.ac.id

<sup>2</sup> Universitas Amikom Yogyakarta; yuni@amikom.ac.id

<sup>3</sup> Universitas Amikom Yogyakarta; muhammad.1904@students.amikom.ac.id

**Abstrak:** Analisis sentimen tentang kelangkaan Bahan Bakar Minyak (BBM) di Indonesia merupakan salah satu cara untuk mengetahui opini masyarakat tentang kelangkaan BBM. Analisis sentimen digunakan sebagai dasar pengambilan keputusan oleh pihak berwenang sebagai upaya penyelesaian masalah, sehingga prediksi sentimen perlu dilakukan. Tujuan penelitian ini adalah melakukan perbandingan akurasi algoritma *Support Vector Machine* (SVM), dan *Random Forest* (RF) untuk analisis sentimen. Kontribusi pada penelitian ini adalah penentuan algoritma yang efektif dalam analisis sentimen Bahan Bakar Minyak di Indonesia. Adapun *Tools* olah data menggunakan Google Colab, dengan bahasa pemrograman Python dan pendekatan *Machine Learning*. Data eksperimen menggunakan data Twitter, diambil pada tanggal 1 -30 Juli 2022 dan terkumpul 6602 data dalam bahasa inggris. Hasil eksperimen menunjukkan bahwa hasil uji SVM untuk nilai *Precision*, *F1-Score* dan *support* sebesar 0.98 lalu 0,97, kemudian 0.98 dan 67, sehingga nilai akurasi secara keseluruhan SVM adalah 0.98. Sedangkan RF memiliki hasil uji nilai *Precision*, *Recall*, *F1-Score* dan *support* sebesar 0,86 kemudian 0,99 lalu 0,92 dan 67. Sedangkan nilai akurasi secara keseluruhan RF adalah 0.90. sehingga secara keseluruhan model SVM lebih direkomendasikan untuk pemodelan prediksi khususnya analisis sentimen pada kasus kelangkaan BBM melalui data Twitter.

**Keywords:** SVM;Random Forest; analisis sentiment;Machine Learning;twitter

DOI: 10.47134/jacis.v5i1.107

\*Correspondensi: Tristanto Ariaaji

Email: tristanto.a@amikom.ac.id

Receive: 9 Mei 2025

Accepted: 17 Mei 2025

Published: 26 Mei 2025



**Copyright:** © 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

**Abstrak:** Sentiment analysis about fuel scarcity in Indonesia is one way to find out public opinion about fuel scarcity. Sentiment analysis is used as a basis for decision-making by the authorities as an effort to solve problems, so sentiment prediction needs to be carried out. The purpose of this study was to compare the accuracy of *Support Vector Machine* (SVM) and *Random Forest* (RF) algorithms for sentiment analysis. The contribution to this study is the determination of an effective algorithm in the analysis of Fuel Oil sentiment in Indonesia. The *data processing tools* use Google Colab, with the Python programming language and a *Machine Learning* approach. The experimental data using Twitter data was taken from July 1 to 30, 2022 and collected 6602 data points in English. The results of the experiment showed that the SVM test results for the *Precision*, *F1-Score*, and *support* values were 0.98, then 0.97, then 0.98, and 67, so that the overall accuracy value of SVM was 0.98. Meanwhile, RF has the results of the *Precision value test*, *Recall*, *F1-score* and *support* are 0.86, then 0.99, then 0.92, and 67. While the overall accuracy value of RF is 0.90. so that overall the

SVM model is more recommended for prediction modeling, especially sentiment analysis in the case of fuel scarcity through Twitter data.

**Keywords:** SVM; Random Forest; Sentiment Analysis; Machine Learning; twitter

---

## PENDAHULUAN

Pada awal tahun 2022, Indonesia mengalami kelangkaan Bahan Bakar Minyak (BBM) sehingga menjadi isu krusial yang perlu diselesaikan. Kelangkaan ini menjadi masalah besar karena tingginya kebutuhan bahan bakar transportasi baik darat, laut, dan udara di Indonesia, sehingga kelangkaan bahan bakar berdampak luas pada aktivitas ekonomi dan mobilitas masyarakat. Dampak lain adalah pada sektor strategis dimana kelangkaan BBM menghambat aktivitas di sektor-sektor penting seperti industri, pertanian, dan kelistrikan, yang dapat mengganggu ketahanan ekonomi nasional. Hal ini menyebabkan beban subsidi yang besar oleh Pemerintah yang harus menanggung beban subsidi BBM akibat kenaikan harga minyak mentah global, sehingga mengalokasikan anggaran yang besar untuk hal ini. Hal ini memicu reaksi masyarakat terhadap kelangkaan BBM di Indonesia dengan menyampaikan aspirasi dan opini mereka melalui berbagai forum terutama media sosial salah satunya Twitter. Sehingga pemerintah diharapkan melihat sentimen masyarakat terhadap kenaikan harga BBM sebagai dasar pengambilan keputusan dimasa yang akan datang.

Salah satu cara melakukan analisis sentimen adalah dengan menggunakan pendekatan machine learning. Beberapa model machine learning yang dapat digunakan untuk melakukan analisis sentimen adalah, Support Vector Machine (SVM) [1], Naive Bayes (NB)[2], Random Forest (RF)[3], Logistic Regression (LR) [4], K-Nearest Neighbor (KNN)[5], [6](DT), Gradient Boosting[7]. Guna mendapatkan akurasi yang baik, perlu dibandingkan beberapa algoritma analisis sentimen, untuk melakukan sentimen analisis kasus BBM di Indonesia. Pada penelitian ini akan dibandingkan algoritma SVM dan RF, kedua algoritma ini memiliki akurasi paling tinggi dibanding algoritma lainnya dalam kasus analisis sentimen [8], [9], [10].

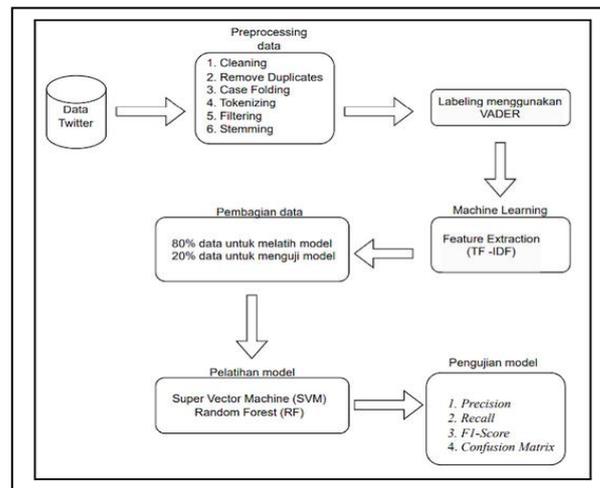
Beberapa penelitian mengungkapkan bahwa SVM digunakan untuk analisis sentimen dalam analisis sentimen pada aplikasi PLUANG [11]. Pada penelitian ini algoritma SVM memiliki akurasi baik yaitu sekitar 99,50%. Selanjutnya digunakan untuk analisis sentimen konflik antara palestina dan Israel, dan akurasi algoritma SVM pada penelitian ini memiliki akurasi baik sekitar 92,93% [12]. Sedangkan algoritma Random Forest memiliki akurasi tinggi juga dalam analisis sentimen, antara lain untuk kasus deteksi depresi pada mahasiswa [13], dan analisis sentimen untuk pengguna aplikasi GOJEK [9]. Berdasarkan uraian tersebut, maka kontribusi penelitian ini adalah menentukan algoritma yang efektif dalam analisis sentimen kasus kelangkaan Bahan Bakar Minyak (BBM) di Indonesia.

Sedangkan tujuan penelitian ini adalah melakukan perbandingan algoritma analisis sentimen menggunakan pendekatan *Machine Learning* khususnya algoritma *Support Vector Machine* (SVM), dan *Random Forest* (RF) untuk analisis sentimen BBM di Indonesia. Hasil penelitian akan disajikan sebagai bahan rekomendasi pihak berwenang dalam menentukan keputusan tentang kenaikan harga BBM di Indonesia.

## METODE

### Desain Penelitian

Tahapan eksperimen dimulai dengan pengumpulan data yang dilakukan melalui Twitter, Selanjutnya dilakukan *preprocessing* data, *labeling*, *feature extraction*, pembagian data, pelatihan model dan pengujian model. Adapun alat yang digunakan adalah Google Collab dengan bahasa pemrograman Python. Adapun tahapan eksperimen yang dilakukan pada penelitian ini disajikan gambar 1.



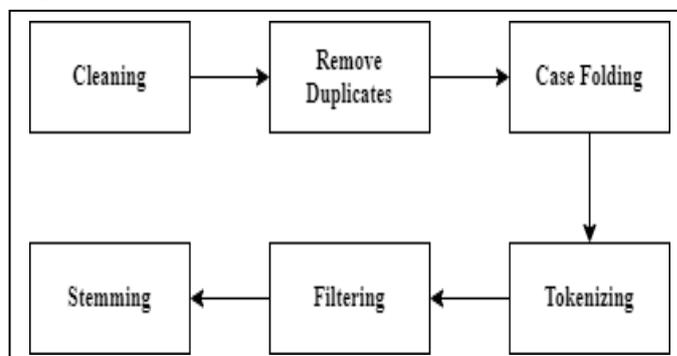
Gambar 1. Tahapan eksperimen

### Pengumpulan data

Data eksperimen ini di ambil dari data Twitter menggunakan snsrape antara 1-30 Juli 2022 terkumpul sebanyak 6602 data berbahasa inggris. Snsrape adalah scraper untuk *Social Networking Services* (SNS), *Scraper* dapat mengikis hal-hal seperti profil pengguna, tagar, atau pencarian dan mengembalikan item yang ditemukan, misalnya postingan yang relevan. Pengambilan data dari Twitter ini menggunakan keyword *#fuelpricehike*, *#petrolprice*, *#fuelprice*, *#fuelprices*, *#fuelcrisis*.

### Preprocessing Data

Proses ini adalah proses pembersihan data, dimana data yang dikumpulkan adalah data tidak terstruktur yang terdiri kata singkatan, simbol, angka, emoji, pengulangan kata serta karakter khusus. *Preprocessing* data disajikan pada Gambar 2.



Gambar 2. Alur *Preprocessing Data Cleaning*

Berikut adalah tahapan *preprocessing* data [11] yaitu:

- a. *Cleaning*  
Tahap ini adalah proses eliminasi komponen-komponen yang tidak dibutuhkan dengan tujuan untuk menurunkan *noise*. Data yang disisakan adalah alfabet.
- b. *Remove Duplicates*  
Adalah proses penghapusan data duplikat setelah melalui proses *cleaning*.
- c. *Case Folding*  
Adalah tahap mengubah teks menjadi *lowercase* atau huruf kecil semua.
- d. *Tokenizing*  
Adalah tahap pemecah kalimat menjadi kata, yang disebut *term* atau *token*.
- e. *Filtering*  
Adalah tahap pengambilan kata-kata penting atau kata-kata yang memiliki makna dari hasil *tokenizing*. Adapun kata yang dihapus adalah kata penghubung seperti *and*, *to*, *from*, *or*.
- f. *Stemming*  
*Stemming* adalah proses mengubah kata-kata yang memiliki imbuhan (awalan atau akhiran) menjadi bentuk kata dasarnya. Aturan-aturan bahasa Inggris seperti tata bahasa dan tenses dihilangkan agar kata tersebut kembali ke bentuk dasar. Contohnya, kata-kata bahasa Inggris '*cleaning*', '*cleaned*', dan '*cleaner*' akan diubah menjadi '*clean*' melalui proses *stemming*.
- g. *Labeling*  
Tahap ini adalah proses pelabelan data menggunakan *Valence Aware Dictionary and sEntiment Reasoner* (VADER). VADER mengkategorikan sentimen menjadi dua kategori, yaitu negatif dan positif. VADER merupakan leksikon berbasis aturan dan alat analisis sentimen yang dirancang khusus untuk menganalisis sentimen yang diekspresikan di media sosial [14]. VADER menggunakan leksikon sentimen, yang merupakan daftar fitur leksikal yang umumnya diberi label berdasarkan orientasi semantiknya sebagai positif atau negatif positif atau negatif sentimen itu) emosi, dan disesuaikan dengan konten jejaring sosial yang umumnya menggunakan tulisan informal (beberapa tanda baca, akronim, emotikon, bahasa gaul).

### ***Feature Extraction***

Adalah proses TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan teknik dalam NLP (*Natural Language Processing*) yang digunakan untuk mengekstraksi kata kunci penting dalam sebuah dokumen. TF-IDF memiliki dua langkah kerja (*Term Frequency* dan *Inverse Document Frequency*). *Term Frequency* bertugas menghitung frekuensi suatu kata berdasarkan berapa kali kata tersebut muncul dalam teks, semakin sering kata tersebut muncul, semakin tinggi nilai kata tersebut. *Inverse Document Frequency* bertugas menghitung kata-kata unik yang ada pada sebuah teks, dengan cara memisahkan kata yang muncul hanya pada satu teks dengan kata yang muncul pada setiap teks [3][12].

### ***Splitting Data***

*Splitting data* merupakan proses pembagian data menjadi dua bagian yaitu data latih (*train data*) dan data uji (*test data*). Pembagian data dilakukan pada kategori sentimen (*positive* dan *negative*) [15].

## Model analisis sentimen

### a. Support Vector Machine (SVM)

*Support Vector Machine* (SVM) merupakan algoritma yang populer dengan kualitas dan pencapaian akurasi yang baik [16]. SVM mengatasi permasalahan secara linier maupun permasalahan non-linier SVM mengidentifikasi batas keputusan yang jauh dari titik data terdekat dari semua kelas dengan mengkonversi data dan menggambar *hyperplane* menggunakan fungsi matematika yang disebut *kernel*. Ada 4 jenis *kernel*. *Kernel* linier yang digunakan untuk pemisahan linier, *kernel* RBF (*Radial Basis Function*) untuk pemisahan non-linier dengan data yang tidak diketahui, serta *kernel* sigmoid dan polynomial [3]. Pada eksperimen ini digunakan *kernel linier* karena jumlah fitur lebih dari data yang disajikan pada persamaan 1 [5].

$$K(x_i, x) = x_i^T x \quad (1)$$

Dimana  $K$  adalah  $x_i$  adalah vektor fitur untuk data ke- $i$ ,  $x$  adalah vektor fitur untuk data uji,  $x_i^T \cdot x$  adalah perkalian transpose vektor  $x_i$  dengan vektor  $x$ . Ini berarti bahwa nilai kernel antara dua vektor  $x_i$  dan  $x$  adalah sama dengan perkalian titik antara kedua vektor tersebut. Konsep dari metode SVM dalam mengklasifikasi data yaitu mencari *hyperplane* terbaik untuk memisahkan antara dua kelas yang telah ditentukan .

### b. Random Forest (RF)

*Random Forest* adalah estimator meta yang sesuai dengan sejumlah pohon keputusan (*Decision Tree*) pada berbagai sub-sampel dari kumpulan data. *Random Forest* menggunakan pendekatan ansambel, menggabungkan prediktor pohon (*tree predictors*) dimana setiap pohon bergantung pada nilai vektor acak dengan distribusi seragam untuk semua RF. Algoritma *Random Forest* memberikan klasifikasi yang efisien dan diskriminatif, sehingga dianggap sebagai *classifier* yang menarik [17].

## Pengujian model

Pengujian akurasi yang dilakukan pada penelitian ini menggunakan beberapa model yaitu Precision, Recall, F1-Score yang disajikan pada formula pengujian pada Tabel 1.

Tabel.1 Formula pengujian

Evaluation Measure	Formula
Accuracy	$(TN + TP) / (TN + FN + TP + FP)$
Precision	$TP / (TP + FP)$
Recall	$TP / (TP + FN)$
F1-Score	$2 * (Precision * Recall) / (Precision + Recall)$

Dimana, TP adalah *True Positive* merupakan data yang sebenarnya positif dan diprediksi dengan benar sebagai positif. Sedangkan TN adalah *True Negative* adalah data yang sebenarnya negatif dan diprediksi dengan benar sebagai negatif. FP adalah *False Positive* dimana data yang sebenarnya negatif namun diprediksi sebagai positif. FN adalah *False Negative* merupakan data yang sebenarnya positif namun diprediksi sebagai negatif [18]. Selanjutnya, hasil pengujian *Confusion Matrix* disajikan pada Tabel 2.

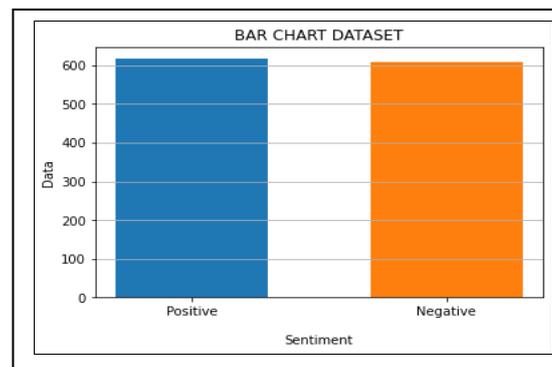
Tabel 2. *Confusion Matrix*

	Prediction		
	Class	Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

## HASIL DAN PEMBAHASAN

### Hasil prapemrosesan data

Berdasarkan pengumpulan data lapangan dengan menggunakan teknik scraping di Twitter, selanjutnya dilakukan prapemrosesan data dan pelabelan data menggunakan VADER. Hasil pada proses ini adalah dataset sentimen positive sebanyak 617 dan dataset negative 608. Sehingga seluruhnya dataset berjumlah 1225 data dengan berjumlah data dan sentimen. Dari dataset tersebut akan di latih dan di uji untuk masing-masing model. Adapun hasil prapemrosesan data disajikan pada Gambar 3.



Gambar 3. Hasil prapemrosesan data

### Hasil cleaning

Langkah pertama pada prapemrosesan data adalah *Cleaning*, berikut adalah *source code* untuk *cleaning data*.

```
#cleaning function
def cleaning(text):
    #replace RT tag
    text = re.sub(r'RT\s', '', text)
    ...
    ...
    text = re.sub(r'^[a-zA-Z]', ' ', text)
    text = text.strip()
    return text
```

Gambar 4. *Sourcode data cleaning*

Gambar 4 menjelaskan Gambar tersebut menunjukkan potongan kode Python yang berfungsi untuk membersihkan teks. Fungsi `cleaning(text)` menggunakan modul *re* (regular expression) untuk menghapus tag "RT" (retweet) dari awal kalimat dan menghilangkan semua karakter selain huruf alfabet (a-z dan A-Z). Selain itu, fungsi ini juga menghapus spasi di awal dan akhir teks dengan `strip()`. Bagian yang diberi simbol titik-titik (...) menandakan kemungkinan adanya proses pembersihan tambahan yang tidak ditampilkan. Fungsi ini umum digunakan

dalam preprocessing data teks seperti tweet agar data menjadi bersih dan siap untuk analisis lebih lanjut seperti klasifikasi atau sentiment analysis.

### Hasil Tokenizing

Selanjutnya adalah proses *Tokenizing* yang disajikan pada *source code* berikut:

```
#tokenizing
df['text']=df['text'].apply(tokenizing)
df.head()
```

**Gambar 5.** *Sourcode Tokenizing*

Gambar 5 menjelaskan tentang proses tokenisasi pada kolom 'text' di DataFrame df. Fungsi `tokenizing` diterapkan ke setiap baris teks menggunakan `.apply()`, lalu hasilnya disimpan kembali ke kolom yang sama. Perintah `df.head()` digunakan untuk menampilkan lima baris pertama hasil tokenisasi. Adapun hasil dari *Tokenizing* adalah sebagai berikut

	datetime	username	text
0	2022-07-30 21:50:30+00:00	_troublemonkey	[chatting, to, my, next, door, neighbour, her,...
1	2022-07-30 21:01:03+00:00	tm_3009	[well, done, it, is, about, time, something, i...
2	2022-07-30 20:50:04+00:00	JonnyKeefe	[esso, near, nottingham, today, cheapest, i, h...
3	2022-07-30 20:48:51+00:00	Ratnesh14223503	[another, master, stroke, from, is, already, a...
4	2022-07-30 20:36:17+00:00	brendamahoro2	[worrying, about, the, it, does, not, make, se...

**Gambar 6.** Hasil *Tokenizing*

Gambar 6 menampilkan isi dari sebuah DataFrame hasil preprocessing data teks, kemungkinan dari media sosial seperti Twitter. Tabel ini memiliki tiga kolom: `datetime`, `username`, dan `text`. Kolom `datetime` menunjukkan waktu unggahan, `username` menampilkan nama pengguna, dan `text` berisi daftar token (kata-kata) hasil dari proses tokenisasi. Setiap entri dalam kolom `text` telah dipisah menjadi kata-kata individual dalam format list Python, menandakan bahwa teks tersebut telah melalui tahapan pembersihan dan tokenisasi. Format seperti ini sangat berguna untuk analisis lanjutan seperti klasifikasi teks, sentiment analysis, atau topic modeling.

### Hasil Filtering

Proses selanjutnya adalah *Filtering* yang disajikan pada *source code* berikut:

```
#filtering/stopword
df['text']=df['text'].apply(stopword)
df.head()
```

**Gambar 7.** *Sourcode Filtering*

Gambar 7 menjelaskan bahwa Gambar tersebut menunjukkan proses menghapus stopwords dari kolom 'text' dalam DataFrame df. Fungsi `stopword` diterapkan ke setiap entri menggunakan `.apply()`, dan hasilnya disimpan kembali. Perintah `df.head()` digunakan untuk menampilkan lima baris pertama hasil filtering teks.

### Hasil filtering

Adapun hasil proses *filtering* disajikan pada Gambar 8.

	datetime	username	text
0	2022-07-30 21:50:30+00:00	_troublemonkey	[chatting, next, door, neighbour, fixed, rate,...
1	2022-07-30 21:01:03+00:00	tm_3009	[well, done, time, something, done, clapping, ...
2	2022-07-30 20:50:04+00:00	JonnyKeefe	[esso, near, nottingham, today, cheapest, seen...
3	2022-07-30 20:48:51+00:00	Ratnesh14223503	[another, master, stroke, already, time, high,...
4	2022-07-30 20:36:17+00:00	brendamahoro2	[worrying, make, sense, tomorrow, take, care, ...

Gambar 8. Hasil *Filtering*

Gambar 8 menampilkan DataFrame yang berisi hasil teks yang telah melalui proses filtering stopwords. Terdapat tiga kolom: *datetime*, *username*, dan *text*. Kolom *text* berisi daftar token (kata-kata) yang telah dibersihkan dari kata-kata umum yang tidak membawa makna penting, seperti "the", "is", "and", dan sebagainya. Hal ini dilakukan untuk meningkatkan kualitas data dalam analisis lanjutan seperti klasifikasi teks atau analisis sentimen. Data ini kemungkinan berasal dari media sosial, dan telah diproses melalui tahapan tokenisasi dan pembersihan, sehingga hanya menyisakan kata-kata bermakna yang relevan untuk dianalisis lebih lanjut. Kemudian proses selanjutnya adalah *Stemming* yaitu proses untuk menghasilkan kata dasar, disajikan pada *source code* berikut ini :

```
#stemming
df['text'] = df['text'].apply(stemming)
df.head()
```

Gambar 9. Sourcode *Stemming*

Gambar 9 menunjukkan proses stemming pada kolom 'text' di DataFrame *df*. Fungsi *stemming* diterapkan untuk mengubah setiap kata menjadi bentuk dasarnya. Hasil stemming disimpan kembali di kolom yang sama, lalu *df.head()* digunakan untuk menampilkan lima baris pertama hasilnya.

### Hasil Stemming

Adapun hasil *stemming* disajikan pada data berikut

	datetime	username	text
0	2022-07-30 21:50:30+00:00	_troublemonkey	chat next door neighbo fix rat gas contract en...
1	2022-07-30 21:01:03+00:00	tm_3009	wel don tim someth don clap hand med dark skin...
2	2022-07-30 20:50:04+00:00	JonnyKeefe	esso near nottingham today cheapest seen long tim
3	2022-07-30 20:48:51+00:00	Ratnesh14223503	anoth mast stroke already tim high nee suff et...
4	2022-07-30 20:36:17+00:00	brendamahoro2	worry mak sens tomorrow tak car self

Gambar 9. Hasil *Stemming*

Gambar 9 menjelaskan hasil proses stemming, proses ini merupakan upaya untuk menormalkan kata-kata ke bentuk dasar, namun terdapat beberapa ketidakkonsistenan atau

kesalahan, hal ini disebabkan adanya kata tidak baku, singkatan, atau bahasa harian di sosial media khususnya Twitter.

### Hasil Labeling

Selanjutnya adalah proses *Labeling* data untuk mendapatkan data positif negatif dan netral, adapun *source code* nya seperti gambar 10

```
def vaderSentimentAnalysis(review):
    vs = sia.polarity_scores(review)
    return vs['compound']
label =
pd.DataFrame(final_processing[['text']
])
label['Polarity'] =
label['text'].apply(vaderSentimentAnal
ysis)
label.head()
```

**Gambar 10.** Sourcode *Labeling*

Gambar 10 merupakan Fungsi `vaderSentimentAnalysis` yang mengambil parameter 'review' dan mengembalikan nilai 'compound' dari hasil analisis polaritas menggunakan `sia.polarity_scores(review)`. Selanjutnya, '`label['Polarity'] = label['text'].apply(vaderSentimentAnalysis)`' menunjukkan bahwa kolom 'Polarity' diisi dengan hasil penerapan fungsi `vaderSentimentAnalysis` pada kolom 'text'. Terakhir, `label.head()` untuk menampilkan beberapa baris DataFrame. Adapun hasil pelabelan data disajikan pada Gambar 11.

	text	Polarity	Sentiment
0	chat next door neighbo fix rat gas contract en...	-0.7717	Negative
1	wel don tim someth don clap hand med dark skin...	0.7096	Positive
2	esso near nottingham today cheapest seen long tim	0.0000	Neutral
3	anoth mast stroke already tim high nee suff et...	0.0000	Neutral
4	worry mak sens tomorrow tak car self	-0.4404	Neutral

**Gambar 11.** Hasil *Labeling*

Gambar 11 adalah hasil pelabelan yang merupakan output analisis sentimen menggunakan VADER yang menampilkan tiga komponen: teks asli, skor *compound* (dari -1 hingga +1), dan label kategori (Negative, Positive, Neutral). Skor negatif (misal: -0.7717) menunjukkan sentimen negatif. Skor positif (0.7096) pada baris kedua mencerminkan pesan dengan nada optimis atau positif. Sementara itu, skor 0.0000 atau mendekati netral (seperti -0.4404 pada baris terakhir) diberi label "Neutral". Secara keseluruhan, hasil pelabelan data pada eksperimen ini adalah Netral 870, Positif 617 dan Negatif 608 data.

### Hasil pengujian performa model *Support Vector Machine (SVM)* dan *Random Forest (RF)*

Berdasarkan model pengujian yang digunakan pada Tabel 1, berikut adalah hasil pengujian SVM, disajikan pada Tabel 3.

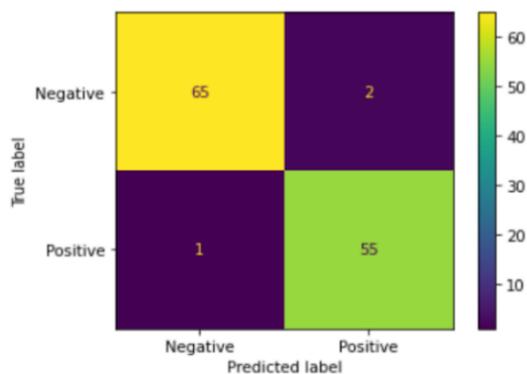
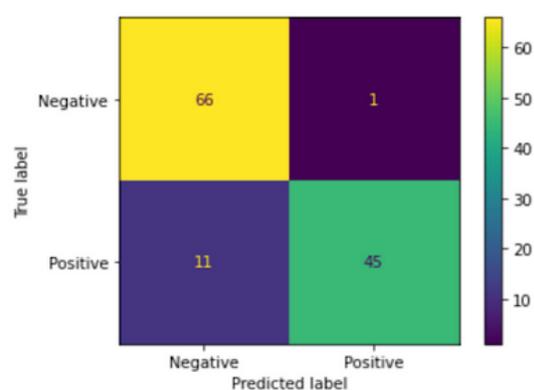
**Tabel 3.** Hasil pengujian performa model SVM dan RF

Model		Precision	Recall	F-Score
SVM	Negative	0,98	0,97	0,98
	Positive	0,96	0,98	0,97
RF	Negative	0,98	0,99	0,92
	Positive	0,96	0,80	0,88

Berdasarkan nilai Presisi model SVM pada Tabel 3, SVM mampu menunjukkan kinerja baik dalam mengidentifikasi kelas dengan akurat dan tweet negatif dan positif. Nilai *Recall* menunjukkan kinerja SVM antara 97-98% dalam mendeteksi tweet negatif dan positif. Adapun nilai *F-Score* menunjukkan rata-rata harmonik dari *precision* dan *recall* untuk kelas negatif, menunjukkan keseimbangan yang sangat baik antara kedua metrik, sedangkan untuk tweet *Positive* 0,97 adalah nilai rata-rata harmonik dari *precision* dan *recall* untuk kelas positif, juga menunjukkan keseimbangan yang sangat baik. Sehingga dapat disimpulkan secara keseluruhan model memiliki F1-score yang sangat baik, menunjukkan kinerja yang konsisten dan seimbang antara *precision* dan *recall*.

Berdasarkan nilai Presisi model RF pada Table 3 mampu menunjukkan kinerja baik dalam mengidentifikasi kelas dengan akurat dan tweet negatif dan positif. Nilai *recall* pada model RF menunjukkan kinerja RF untuk mendeteksi tweet positif adalah 80% dalam mendeteksi sedangkan untuk mendeteksi kelas negatif adalah 99%. Adapun nilai *F-Score* menunjukkan rata-rata harmonik dari *precision* dan *recall* untuk kelas negatif, menunjukkan keseimbangan yang sangat baik antara kedua metrik, sedangkan untuk tweet positive dan negative adalah 0,88 atau 88% dan 0,92 atau 92%. Secara keseluruhan model memiliki *f-score* baik, menunjukkan kinerja yang konsisten dan seimbang antara *precision* dan *recall*. Sedangkan nilai akurasi keseluruhan model adalah 0.90 atau 90%. Ini berarti model dapat mengklasifikasikan 90% dari total tweet adalah benar. Berdasarkan hasil eksperimen ini disimpulkan bahwa SVM memiliki nilai akurasi lebih baik dibandingkan.

Berdasarkan perbandingan performa pada Tabel 3, secara keseluruhan SVM lebih baik dan lebih akurat, terutama untuk kelas positif, dengan nilai *recall* dan *F-Score* yang lebih tinggi. Meskipun RF menunjukkan performa yang baik untuk kelas negatif, kinerjanya untuk kelas positif kurang memuaskan. Adapun metric hasil pengujian SVM disajikan pada Gambar 11.

**Gambar 11 (a).** Hasil pengujian metric Support Vector Machine**Gambar 11 (b).** Hasil pengujian metric Random Fores

Menggunakan model pengujian pada Tabel 1, hasil pengujian metric SVM menunjukkan bahwa distribusi hasil sentimen negatif terdiri dari 65 tweet negatif dan 2 tweet positif sedangkan distribusi hasil sentimen positif terdiri dari 1 hasil tweet positif dan 55 hasil tweet negatif. Sehingga berdasarkan hasil pengujian pada Gambar 11 (a), hasil sentimen negatif terhadap kenaikan harga BBM di Indonesia lebih tinggi dibandingkan hasil sentimen positif. Sedangkan nilai akurasi keseluruhan model adalah 0.98 atau 98%. Ini berarti model dapat mengklasifikasikan dengan benar 98% dari total tweet yang ada. Berdasarkan analisa pada ini disimpulkan bahwa SVM memiliki akurasi sangat baik yaitu 98% dibanding RF, hal ini menunjukkan model yang kuat untuk tugas analisis sentimen. Sedangkan hasil pengujian metric RF berdasarkan Gambar 11 (b) , menunjukkan bahwa pada distribusi hasil sentimen negatif terdiri dari 66 tweet negatif dan 1 tweet positif sedangkan distribusi hasil sentimen positif terdiri dari 11 hasil tweet positif dan 45 hasil tweet negatif. Sehingga berdasarkan gambar di atas, hasil sentimen negatif lebih tinggi dibandingkan hasil sentimen positif. Sehingga Secara keseluruhan hasil analisa menyimpulkan bahwa model SVM lebih efektif dalam mengidentifikasi sentimen terkait kenaikan harga BBM dibandingkan dengan model RF. Berdasarkan Tabel 2, maka Adapun hasil confusion matrix kedua model disajikan pada Tabel 4.

**Tabel 4.** Hasil perbandingan Confussional metric model SVM dan RF.

Actual Model	Prediction		
	Class	Negative	Positive
Actual SVM	Negative	65	2
	Positive	1	55
Actual RF	Negative	66	1
	Positive	11	45

Berdasarkan Tabel 4, hasil prediksi dua model SVM dan RF, menunjukkan bahwa untuk SVM, dari 67 tweet negatif, 65 teridentifikasi dengan benar dan 2 salah menjadi positif. Di sisi lain, dari 56 tweet positif, 55 teridentifikasi dengan benar dan 1 salah menjadi negatif. Untuk RF, dari 67 tweet negatif, 66 terprediksi benar, sedangkan 1 salah menjadi positif. Dari 56 tweet positif, 11 diklasifikasikan salah sebagai negatif dan 45 benar. Sehingga kesimpulannya adalah SVM memiliki kinerja lebih baik untuk dibandingkan RF.

## KESIMPULAN

Berdasarkan hasil eksperimen perbandingan model SVM dan RF menunjukkan bahwa kedua model memiliki nilai akurasi tinggi. Tetapi, pada kasus analisis sentimen kelangkaan BBM ini, akurasi SVM lebih baik dibanding RF Hal ini terlihat pada hasil uji nilai *Pressision*, *F1-Score* dan *support* sebesar 0.98 lalu 0,97, kemudian 0.98 dan 67. Sedangkan nilai akurasi secara keseluruhan SVM adalah 0.98. Sedangkan RF memiliki hasil uji nilai *Pressision*, *Recall*. *F1-Score* dan *support* sebesar 0,86 kemudian 0,99 lalu 0,92 dan 67. Sedangkan nilai akurasi secara keseluruhan RF adalah 0.90. sehingga secara keseluruhan SVM lebih direkomendasikan untuk pemodela prediksi khususnya analisis sentimen pada kasus kelangkaan BBM menggunakan data *Twitter*. Pada penelitian yang akan datang, akan dilakukan perbandingan model analisis sentimen menggunakan pendekatan *Deep Learning*, untuk mendapatkan hasil akurasi lebih baik dibanding hasil eksperimen penelitian ini.

## DAFTAR PUSTAKA

- [1] Fransiscus and A. S. Girsang, "Sentiment Analysis of COVID-19 Public Activity Restriction (PPKM) Impact using BERT Method," *Int. J. Eng. Trends Technol.*, vol. 70, no. 12, pp. 281–288, Dec. 2022, doi: 10.14445/22315381/IJETT-V70I12P226.
- [2] A. S. Neogi, K. A. Garg, R. K. Mishra, and Y. K. Dwivedi, "Sentiment analysis and classification of Indian farmers' protest using twitter data," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, Nov. 2021, doi: 10.1016/j.jjime.2021.100019.
- [3] M. Demircan, A. Seller, F. Abut, and M. F. Akay, "Developing Turkish sentiment analysis models using machine learning and e-commerce data," *Int. J. Cogn. Comput. Eng.*, vol. 2, no. October, pp. 202–207, 2021, doi: 10.1016/j.ijcce.2021.11.003.
- [4] S. R. Shah, A. Kaushik, S. Sharma, and J. Shah, "Opinion-mining on marglish and devanagari comments of youtube cookery channels using parametric and non-parametric learning models," *Big Data Cogn. Comput.*, vol. 4, no. 1, pp. 1–19, Mar. 2020, doi: 10.3390/bdcc4010003.
- [5] S. S. Aljameel et al., "A sentiment analysis approach to predict an individual's awareness of the precautionary procedures to prevent covid-19 outbreaks in Saudi Arabia," *Int. J. Environ. Res. Public Health*, vol. 18, no. 1, pp. 1–12, Jan. 2021, doi: 10.3390/ijerph18010218.
- [6] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, Dec. 2021, doi: 10.3390/informatics8040079.
- [7] P. T. Krishnan, A. N. Joseph Raj, and V. Rajangam, "Emotion classification from speech signal based on empirical mode decomposition and non-linear features: Speech emotion recognition," *Complex Intell. Syst.*, vol. 7, no. 4, pp. 1919–1934, Aug. 2021, doi: 10.1007/s40747-021-00295-z.
- [8] J. Artikel, "Form Hasil Review (Faizah)".
- [9] G. Kanugrahan, V. Hafizh, C. Putra, and Y. Ramdhani, "Analisis Sentimen Aplikasi Gojek Menggunakan SVM, Random Forest dan Decision Tree," vol. 6, no. 2, 2024.
- [10] T. Ahmed Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. Mohd Su'ud, "Sentiment Analysis using Support Vector Machine and Random Forest," *J. Informatics Web Eng.*, vol. 3, no. 1, pp. 67–75, 2024, doi: 10.33093/jiwe.2024.3.1.5.
- [11] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, "Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 375–384, 2024, doi: 10.57152/malcom.v4i2.1206.
- [12] F. M. Carina, Admi Salma, Dony Permana, and Zamahsary Martha, "Sentiment Analysis of X Application Users on the Conflict Between Israel and Palestine Using Support Vector Machine Algorithm," *UNP J. Stat. Data Sci.*, vol. 2, no. 2, pp. 204–212, 2024, doi: 10.24036/ujsds/vol2-iss2/170.
- [13] P. Sankar, N. Palanichamy, and K. Ng, "Sentiment Analysis on Twitter Data for Depression Detection," *J. Logist. Informatics Serv. Sci.*, vol. 11, no. 3, pp. 21–36, 2024, doi: 10.33168/jliss.2024.0302.
- [14] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, "A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis," *PLoS One*, vol. 16, no. 2, Feb. 2021, doi: 10.1371/journal.pone.0245909.
- [15] Ferdi and Vina Ayumi, "Analisa Sentimen Mengenai Kenaikan Harga Bbm Menggunakan Metode Naïve Bayes Dan Support Vector Machine," *JSAI (Journal Sci. Appl. Informatics)*, vol. 6, no. 1, pp. 1–10, 2023, doi: 10.36085/jsai.v6i1.4628.
- [16] R. Alfred and J. H. Obit, "The roles of machine learning methods in limiting the spread of deadly diseases: A systematic review," *Heliyon*, vol. 7, no. 6, Elsevier Ltd, Jun. 01, 2021. doi: 10.1016/j.heliyon.2021.e07371.
- [17] A. Al-Hashedi et al., "Ensemble Classifiers for Arabic Sentiment Analysis of Social Network (Twitter Data) towards COVID-19-Related Conspiracy Theories," *Appl. Comput. Intell. Soft Comput.*, vol. 2022, 2022, doi: 10.1155/2022/6614730.

- 
- [18] M. Maemunah, M. Istighosah, S. N. Wahyuni, and Suyatmi, "The implementation of sentiment analysis on Indonesian sexual violation bill using Naïve Bayes algorithm," AIP Conf. Proc., vol. 2508, 2023, doi: 10.1063/5.0117726.