

# PEMBANGUNAN MODEL ANALISIS SENTIMEN ULASAN APLIKASI E-COMMERCE PADA PLAYSTORE BERBASIS STOCHASTIC GRADIENT DESCENT

Robert Varian<sup>1</sup>, Jimmy Tjen<sup>2</sup>, Hendro<sup>3</sup>

Informatika<sup>[1,2]</sup>, Sistem Informasi<sup>[3]</sup> Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak  
<sup>1</sup>robertvarian7@gmail.com, <sup>2</sup>jimmy\_tjen@widyadharm.ac.id, <sup>3</sup>hendro@widyadharm.ac.id

## Abstract

*The rapid advancement of digital technology has driven a significant increase in the use of e-commerce platforms as practical and efficient transaction media. The growing number of e-commerce users has resulted in a substantial increase in user-generated reviews on application platforms, particularly on the Playstore. User reviews play a strategic role in reflecting service quality and serve as an important source of information for application developers. However, the large volume of reviews makes classical approach less effective and less efficient. Therefore, a computational-based sentiment analysis approach is required to automatically classify user reviews. This study aims to develop and compare sentiment analysis models using support vector machine (SVM) with and without optimization through stochastic gradient descent (SGD). An experimental research method was employed, utilizing user review data that were transformed into numerical representations using the term frequency-inverse document frequency (TF-IDF) technique. Sentiment classification was conducted across three sentiment classes: positive, negative, and neutral. The experimental results indicate that the SVM model optimized with SGD achieves better performance compared to the standard SVM model, demonstrated by an accuracy improvement of 4% and a reduction in computational time of 3 seconds. Therefore, the combination of TF-IDF and SVM optimized with SGD can be considered a reliable and effective approach for sentiment analysis of e-commerce application reviews.*

**Keywords:** Sentiment analysis, E-commerce reviews, Stochastic Gradient Descent, Support Vector Machine, TF-IDF

## Abstrak

Perkembangan teknologi digital yang semakin pesat telah mendorong peningkatan penggunaan media *e-commerce* sebagai sarana transaksi yang praktis dan efisien. Peningkatan jumlah pengguna *e-commerce* berimplikasi pada bertambahnya volume ulasan yang diberikan oleh pengguna pada platform aplikasi, khususnya Playstore. Ulasan pengguna memiliki peran strategis dalam merefleksikan kualitas layanan serta menjadi sumber informasi penting bagi pengembang aplikasi. Namun, tingginya jumlah ulasan menyebabkan proses analisis secara manual menjadi tidak efektif dan kurang efisien. Oleh karena itu, diperlukan pendekatan analisis sentimen berbasis komputasi untuk mengklasifikasikan ulasan pengguna secara otomatis. Penelitian ini bertujuan untuk mengembangkan dan membandingkan model analisis sentimen menggunakan *Support Vector Machine* (SVM) dengan dan tanpa optimasi *Stochastic Gradient Descent* (SGD). Metode penelitian yang digunakan adalah metode eksperimental dengan data ulasan dan direpresentasikan dalam bentuk numerik menggunakan teknik *Term Frequency – Inverse Document Frequency* (TF-IDF). Proses klasifikasi sentimen dilakukan terhadap tiga kelas sentimen, yaitu positif, negatif, dan netral. Hasil pengujian menunjukkan bahwa model SVM yang dioptimasi menggunakan SGD menghasilkan performa lebih baik dibandingkan dengan SVM tanpa optimasi, baik dari segi akurasi dengan peningkatan sebesar 4% dan efisiensi waktu komputasi lebih cepat 3 detik. Dengan demikian, kombinasi TF-IDF dan SVM yang dioptimasi menggunakan SGD dapat dijadikan sebagai pendekatan yang andal dalam analisis sentimen ulasan aplikasi *e-commerce*.

**Kata kunci:** Analisis sentimen, Ulasan e-commerce, Stochastic Gradient Descent, Support Vector Machine, TF-IDF

## 1. PENDAHULUAN

Internet saat ini telah menjadi elemen yang tidak terpisahkan dari kehidupan masyarakat Indonesia, didorong oleh adopsi teknologi digital yang sangat tinggi. Berdasarkan data survei dari Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), jumlah pengguna internet di Indonesia pada tahun 2024 telah mencapai 221 juta jiwa <sup>[1]</sup>. Selain itu, laporan Digital 2024 mencatat adanya 185,3 juta pengguna aktif dengan jumlah sambungan seluler yang melebihi total populasi, yang mengindikasikan penggunaan perangkat ganda oleh masyarakat <sup>[2]</sup>.

Fenomena ini menjadikan internet sebagai platform vital bagi berbagai aktivitas ekonomi, khususnya *e-commerce* yang didominasi oleh pemain besar seperti Tokopedia, Shopee, dan Lazada [3].

Pertumbuhan sektor *e-commerce* di Indonesia berbanding lurus dengan meningkatnya volume ulasan pengguna yang diberikan pada platform aplikasi seperti *Playstore*. Ulasan tersebut memiliki peran strategis dalam membantu calon konsumen mengambil keputusan serta memberikan masukan penting bagi pengembang untuk meningkatkan kualitas layanan [4]. Namun, besarnya volume data ulasan ini menimbulkan tantangan signifikan dalam proses analisisnya. Analisis ulasan secara manual memerlukan waktu yang lama dan rentan terhadap kesalahan tafsir, sehingga diperlukan sistem otomatisasi yang efisien [6].

Penelitian terdahulu telah berupaya mengatasi kendala ini menggunakan berbagai algoritma. Sebagai contoh, algoritma *Naïve Bayes* telah diterapkan untuk menganalisis sentimen pengguna aplikasi *travel* dengan tingkat akurasi yang baik [7], serta digunakan untuk klasifikasi ulasan produk kosmetik di *marketplace* [8]. Selain itu, *Naïve Bayes* juga diaplikasikan dalam meneliti sentimen masyarakat terkait keamanan transaksi pada *e-commerce* B2C [9]. Di sisi lain, penggunaan teknik pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) telah terbukti efektif dalam merepresentasikan teks secara numerik untuk analisis sentimen [4]. Namun, tantangan utama tetap terletak pada efisiensi komputasi ketika menangani dataset dalam skala besar [10].

Berdasarkan permasalahan tersebut, diperlukan model analisis sentimen yang mampu menangani data berdimensi tinggi dengan waktu pelatihan yang lebih cepat. Penelitian ini mengusulkan pengembangan model menggunakan kombinasi TF-IDF dan algoritma *Support Vector Machine* (SVM) yang dioptimasi dengan *Stochastic Gradient Descent* (SGD). Optimasi SGD dipilih karena ketangguhannya dalam meningkatkan efisiensi proses pelatihan pada dataset besar dibandingkan metode *batch* konvensional [5]. Melalui pendekatan ini, diharapkan model mampu mengklasifikasikan sentimen ulasan *e-commerce* ke dalam kategori positif, netral, dan negatif dengan tingkat akurasi dan efisiensi yang optimal [10].

## 2. METODOLOGI PENELITIAN

### 2.1 Jenis dan Desain Penelitian

Penelitian ini dikategorikan sebagai penelitian eksperimental kuantitatif yang berfokus pada pengujian hipotesis performa model machine learning dalam tugas klasifikasi teks. Desain penelitian mengadopsi pendekatan iteratif *Knowledge Discovery in Databases* (KDD) yang mencakup tahapan selection, preprocessing, transformation, data mining, dan evaluation, sebagaimana digambarkan dalam alur kerja standar KDD untuk analisis data teks berskala besar. Pendekatan ini dipilih karena kemampuannya mengelola siklus pengembangan model dari melakukan pemilihan data hingga menghasilkan informasi yang bermanfaat, dengan perlakuan proses 10 kali *run* percobaan untuk memastikan hasil yang pasti. Waktu pelaksanaan penelitian mencakup periode Oktober hingga Desember 2025, dengan jadwal terperinci: minggu 1-4 Oktober untuk pengumpulan data dan studi literatur awal, November untuk pengolahan dan mode, serta Desember untuk evaluasi dan analisa hasil. Berikut merupakan pemaparan alur kerja metode KDD pada Gambar 1:

- Tahap *Selection*: *selection* berfokus pada penentuan kelas dan pengumpulan data yang relevan dengan menggunakan teknik *web scraping* pada aplikasi *e-commerce* di *Playstore*.
- Tahap *Preprocessing*: *preprocessing data* bertujuan membersihkan dan mempersiapkan data mentah agar dapat diproses pada model dengan melakukan teknik pembersihan data teks seperti *cleaning text*, *case folding*, *slang word normalization*, eliminasi *stopwords*, *stemming*, dan *tokenization*.
- Tahap *Transformation*: *transformation* melibatkan perubahan data teks menjadi data vektor numerik agar dapat diproses oleh model dengan menggunakan teknik TF-IDF dan pembagian data *training* serta *testing* dengan besar 70:30.
- Tahap *Data Mining atau Processing*: *data mining* disini yaitu menjalankan proses *training* dan *testing* model SVM konvensional dan SVM yang dioptimasi dengan SGD.
- Tahap *Evaluation*: *evaluation* melibatkan perbandingan hasil antara kedua model yaitu SVM biasa dan SVM yang dioptimasi dengan SGD. Metrik evaluasi yang menjadi perbandingan adalah akurasi, presisi, *recall*, *F1-score*, dan *runtime*.

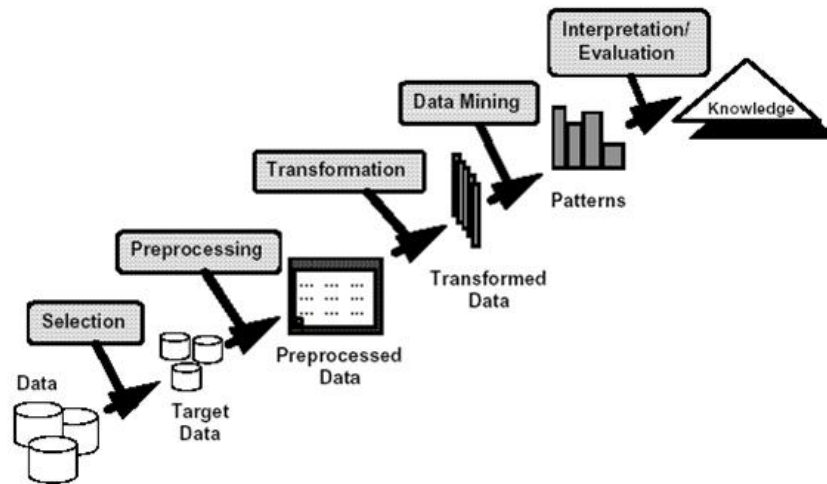
### 2.2 Pengumpulan Data

Pengumpulan data dilakukan melalui teknik web scraping otomatis menggunakan library Python *google-play-scraper* untuk mengakses ulasan pengguna dari *Playstore*. Target dataset terdiri dari 1.000 ulasan terbaru per aplikasi dari lima platform e-commerce dominan, dengan distribusi berimbang per kategori sentimen: 500 ulasan positif (rating bintang 5), 250 ulasan negatif (rating bintang 1), dan 250 ulasan netral (rating bintang 3), menghasilkan total 5.000 ulasan yang merepresentasikan opini pengguna autentik tahun 2025. Batasan pengumpulan diterapkan pada ulasan berbahasa Indonesia dengan menghindari duplikasi melalui pengecekan hash unik.

### 2.3 Preprocessing dan Transformasi Data

Tahap *preprocessing* dalam kerangka KDD dirancang untuk membersihkan dan menstandarisasi data teks mentah menjadi *format* yang siap untuk pemodelan. Langkah awal meliputi *cleaning text* dengan menghapus karakter non-alfanumerik, *mention* (@), *hashtag* (#), URL, dan simbol *punctuation* (!@#\$%^&\*()\_+=[{}];:;<>?), diikuti *case*

*folding* untuk konversi seluruh teks menjadi huruf kecil. Selanjutnya, normalisasi *slang words* dilakukan menggunakan kamus berbasis AI yang mengonversi istilah gaul seperti "mantul" menjadi "mantap", "gk" menjadi "tidak", dan "bgtt" menjadi "banget", dengan akurasi konversi di atas 95% setelah validasi manual sampel. Proses dilanjutkan dengan eliminasi *stopwords* menggunakan daftar kata umum bahasa Indonesia menggunakan pustaka Sastrawi, stemming untuk reduksi kata dasar, dan tokenisasi untuk pemecahan kalimat menjadi *token* individual. Dataset akhir dibagi dengan *train\_test\_split* rasio 70:30 menggunakan *random\_state* beragam untuk setiap run, memastikan pembagian pada data kelas sentimen.



Gambar 1. Alur Kerja Metode KDD

#### 2.4 Evaluasi Model

Performa dievaluasi menggunakan *confusion matrix multiclass* dengan dimensi 3x3 karena data terdiri dari 3 kelas, metrik akurasi, presisi, *recall*, F1-score, dan *runtime*. Contoh *confusion matrix* penelitian ini dapat dilihat pada Gambar 2.

		Predicted		
		C1	C2	C3
Actual	C1	$TP_{(c1,c1)}$	$FN_{(c1,c2)}$	$FN_{(c1,c3)}$
	C2	$FP_{(c2,c1)}$	$TN_{(c2,c2)}$	$TN_{(c2,c3)}$
	C3	$FP_{(c3,c1)}$	$TN_{(c3,c2)}$	$TN_{(c3,c3)}$

Gambar 2. Confusion Matrix

Terdapat empat hasil klasifikasi <sup>[11]</sup>:

- True Positive* (TP) merupakan prediksi positif yang hasil *actual*-nya positif.
- True Negative* (TN) merupakan prediksi negatif yang hasil *actual*-nya negatif.
- False Positive* (FP) merupakan prediksi positif namun hasil *actual*-nya negatif.
- False Negative* (FN) merupakan prediksi negatif namun hasil *actual*-nya positif.

Terdapat beberapa metrik pada *confusion matrix* <sup>[12]</sup>, yaitu:

- Accuracy* meliputi total prediksi yang benar dari seluruh observasi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Recall* atau *sensitivity* meliputi jumlah data *true positive* yang diprediksi terhadap jumlah data aktual positif.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

- c. *Precision* meliputi data positif yang dapat diklasifikasikan dengan nilai *true positive* terhadap jumlah total hasil yang bersifat positif.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

- d. *F1-Score* meliputi perhitungan hubungan antara nilai *precision* dan *recall*.

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengambilan dan Preprocessing Data

Data yang dikumpulkan untuk penelitian ini merupakan hasil *web scraping* ulasan pengguna pada *Playstore* dengan menggunakan bahasa pemrograman python dengan library *google-play-scraper* oleh JoMingyu. Pada Gambar 3 adalah *pseudocode web scraping*, dimulai dengan pendeklarasian variabel seperti *app\_id*, *lang*, *country*, *sort*, *filter\_score\_with*, *count* dan *scrapereview*. *App\_id* memuat id *e-commerce* yang dituju dimana pada penelitian ini yaitu kelima aplikasi *e-commerce*, untuk *lang* memuat bahasa Indonesia, *country* menggunakan negara Indonesia jadi hanya ulasan pengguna Indonesia saja yang diambil, untuk *sort* sebagai *filter* ulasan dimana pada penelitian ini menggunakan *newest* atau data terbaru, *count* untuk seberapa banyak ulasan yang ingin diambil, dan *scrapereview* memuat semua hasil *web scraping* dengan *parameter* berupa variabel yang telah diinisialisasi sebelumnya.

START

DEKLARASI:

status : STRING  
*app\_id*, *lang*, *country* : STRING  
 sort : CONST  
*filter\_score\_with\_count* : INT  
*scrapereview* : ARRAY

ALGORITMA:

```
RUN "pip install google_play_scraper"
RUN "from google_play_scraper import app, sort, reviews_all"
app_id ← "com.tokopedia.tkpd"
lang ← "id"
country ← "id"
sort ← Sort.NEWEST
filter_score_with ← 5
count ← 250
scrapereview ← reviews_all(app_id, lang, country, sort, filter_score_with, count)
```

END

Gambar 3. Alur Pseudocode Web Srapping

Data dikumpulkan dari lima aplikasi *e-commerce* terbesar di Indonesia yaitu Tokopedia, Shopee, Lazada, Blibli, dan Bukalapak. Pada Tabel 1 menjelaskan contoh struktur data yang diambil melalui *web scraping* terhadap *e-commerce*.

Distribusi data dibagi ke dalam 3 kelas sentimen sebagai berikut:

1. 2500 data ulasan positif (50% dari total data),
2. 1250 negatif, (25% dari total data),
3. 1250 netral (25% dari total data).

Tabel 1. Data Hasil Web Scraping

No.	Review	Rating
1.	Samgat mudah dan pastinya ori dong	Positif
2.	mantap sekali	Positif
3.	Okee terimakasih	Positif
4.	mantul aplikasinya ada promo dan tdk perlu nyicil lagi, jadi utk bayar tagihan beli pulsa saya gunakan blibi saja bukan yg lain ...	Positif
⋮	⋮	⋮
5000.	Harga barang nya mahal banget	Netral

Sebelum memasuki tahap pemodelan, data melalui serangkaian *preprocessing data* yang meliputi:

- a. *Cleaning Text: cleaning text* meliputi penghapusan karakter non-alfabet seperti angka, simbol dan tanda baca guna menghilangkan *noise* pada data. Hasil proses *cleaning text* dapat dilihat pada Tabel 2.

Tabel 2. Data Hasil Proses Cleaning Text

No.	Review	Rating
1.	Samgat mudah dan pastinya ori dong	Positif
2.	mantap sekali	Positif
3.	Okee terimakasih	Positif
4.	mantul aplikasinya ada promo dan tdk perlu nyicil lagi jadi utk bayar tagihan beli pulsa saya gunakan blibi saja bukan yg lain	Positif
⋮	⋮	⋮
5000.	Harga barang nya mahal banget	Netral

- b. *Case Folding: case folding* mencakup penyeragaman seluruh huruf menjadi huruf kecil untuk menghindari ambiguitas kata dan duplikasi kata akibat perbedaan format penulisan. Hasil proses *case folding* dapat dilihat pada Tabel 3.

Tabel 3. Data Hasil Proses Case Folding

No.	Review	Rating
1.	sangat mudah dan pastinya ori dong	Positif
2.	mantap sekali	Positif
3.	okee terimakasih	Positif
4.	mantul aplikasinya ada promo dan tdk perlu nyicil lagi jadi utk bayar tagihan beli pulsa saya gunakan blibi saja bukan yg lain	Positif
⋮	⋮	⋮
5000.	harga barang nya mahal banget	Netral

- c. *Slang Word Normalization: slang word normalization* melibatkan konversi bahasa gaul atau singkat menjadi kata baku agar model dapat memahami makna kata secara konsisten. Hasil proses *slang word normalization* dapat dilihat pada Tabel 4.

- d. Eliminasi *Stopword*: eliminasi *stopword* dengan menerapkan penghapusan kata umum yang tidak memiliki signifikansi kontekstual seperti kata sambung dan kata ganti. Hasil proses eliminasi *stopword* dapat dilihat pada Tabel 5.

- e. *Stemming: stemming* mencakup pengembalian kata ke bentuk dasar dengan menggunakan *library Sastrawi* untuk menghilangkan imbuhan. Hasil proses *stemming* dapat dilihat pada Tabel 6.

Tabel 4. Data Hasil Proses Slang Word Normalization

No.	Review	Rating
-----	--------	--------

1.	sangat mudah dan pastinya original	Positif
2.	mantap	Positif
3.	oke terimakasih	Positif
4.	mantap aplikasinya ada promo dan tidak perlu cicil jadi untuk bayar tagihan beli pulsa saya gunakan blibi saja bukan yang lain	Positif
⋮	⋮	⋮
5000.	harga barang nya sangat mahal	Netral

Tabel 5. Data Hasil Proses Eliminasi Stopword

No.	Review	Rating
1.	mudah pastinya original	Positif
2.	mantap	Positif
3.	oke terimakasih	Positif
4.	mantap aplikasinya promo tidak perlu cicil lagi bayar tagihan beli pulsa gunakan blibi bukan lain	Positif
⋮	⋮	⋮
5000.	harga barang sangat mahal	Netral

Tabel 6. Data Hasil Proses Stemming

No.	Review	Rating
1.	mudah pasti original	Positif
2.	mantap	Positif
3.	oke terimakasih	Positif
4.	mantap aplikasi promo tidak perlu cicil lagi bayar tagih beli pulsa guna blibi bukan lain	Positif
⋮	⋮	⋮
5000.	harga barang sangat mahal	Netral

- f. *Tokenization*: *tokenization* melibatkan pemecahan teks yang telah bersih menjadi bagian kecil yang disebut token. Tahap ini menjadi dasar bagi perhitungan bobot pada tahap ekstraksi fitur. Hasil proses *stemming* dapat dilihat pada Tabel 7.

Tabel 7. Data Hasil Proses Tokenization

No.	Review	Rating
1.	‘mudah’, ‘pasti’, ‘original’	Positif
2.	‘mantap’	Positif
3.	‘oke’, ‘terimakasih’	Positif
4.	‘mantap’, ‘aplikasi’, ‘promo’, ‘tidak’, ‘perlu’, ‘cicil’, ‘lagi’, ‘bayar’, ‘tagih’, ‘beli’, ‘pulsa’, ‘guna’, ‘blibi’, ‘bukan’, ‘lain’	Positif
⋮	⋮	⋮
5000.	‘harga’, ‘barang’, ‘sangat’, ‘mahal’	Netral

Data yang telah melalui *preprocessing data* kemudian dikonversi ke dalam bentuk numerik menggunakan teknik TF-IDF. Proses ini dimulai dengan *Bag of Words* (BoW) untuk menghitung kemunculan kata dapat dilihat pada Tabel 8. Setelah hasil kemunculan kata didapatkan maka akan dilanjutkan dengan pembobotan menggunakan TF-IDF.

Kata yang sudah dibobotkan akan dibagi dengan persentase sebesar 70% data *training* dan 30% data *testing*. Kemudian eksperimen dilakukan terhadap 2 model yaitu model SVM konvensional dan SVM yang dioptimasi dengan SGD. *Parameter* yang digunakan pada penelitian ini dapat dilihat pada Tabel 9.

### 3.3 Analisis dan Evaluasi Model

Pada Tabel 10 ditampilkan hasil pelatihan dan pengujian model dengan menggunakan 10 percobaan dengan mengambil *mean* pada tiap kelasnya. Hasil menunjukkan bahwa integrasi optimasi SGD pada algoritma SVM mampu meningkatkan akurasi sebesar 3,66% dari 64,42% hingga 68,08%. Peningkatan ini membuktikan bahwa pendekatan SGD mampu melakukan pembaruan *parameter* model secara lebih dinamis dibandingkan metode SVM konvensional.

Temuan lain dalam penelitian ini adalah efisiensi waktu yang sangat signifikan. Model SVM yang dioptimasi dengan SGD mampu menyelesaikan proses klasifikasi dalam waktu 1,58 detik atau sekitar 2,5 kali lebih cepat

dibandingkan dengan model SVM konvensional yang memperoleh waktu 4,10 detik. Hal ini terjadi dikarenakan SGD tidak memproses keseluruhan dataset dalam satu waktu atau *batch*, melainkan secara iteratif dimana sangat krusial jika model ini akan diimplementasikan untuk menganalisis jutaan ulasan secara real-time di lingkungan aplikasi *e-commerce*.

Ditinjau dari metrik *recall*, kelas positif memberikan hasil yang paling dominan mencapai 87,79% pada model SVM yang dioptimasi dengan SGD. Hal ini disebabkan oleh proporsi data positif yang lebih besar dalam *dataset* yang berupa 50% dari keseluruhan data yang memberikan pola konsisten untuk dipelajari model. Sebaliknya pada kelas netral memiliki tingkat keberhasilan terendah dengan 36,99%. Hal tersebut merupakan tantangan umum dalam *Natural Language Processing* (NLP), dimana ulasan netral sering kali menggunakan bahasa yang sangat objektif atau mengandung campuran antara sentimen positif dan negatif pada suatu kalimat yang pendek sehingga algoritma seringkali salah mengklasifikasi ke dalam kelas positif maupun kelas negatif.

Tabel 8. Data Hasil BoW

Token	DF	N/DF	Rating
mudah	1	4	2.386
pasti	1	4	2.386
⋮	⋮	⋮	⋮
guna	2	2	1.693
⋮	⋮	⋮	⋮
mahal	1	4	2.386

Tabel 9. Parameter Tuning

No.	Parameter	Rating
1.	Learning Rate	0,45
2.	$\lambda$ (Lambda)	0,00013
3.	Iteration Limit	1300
4.	Regularization	Ridge

Temuan mengenai keseimbangan performa sangat penting dan menjadi yang paling representative dimana model SVM yang dioptimasi dengan SGD mampu memperoleh F1-Score yang lebih baik dibandingkan SVM standar dengan kelas positif sebesar 80,60%, 44,18% untuk kelas netral, dan 59,99% pada kelas negatif. Peningkatan ini membuktikan bahwa integrasi SGD memberikan keseimbangan yang lebih baik antara presisi dan *recall*. Model tidak hanya mengejar akurasi tinggi dengan memihak kelas mayoritas, tetapi juga tetap mengklasifikasikan kelas negatif dan netral yang memiliki data lebih sedikit.

START

DEKLARASI:

X, w : ARRAY OF VECTOR

Y : ARRAY OF INTEGER

b,  $\eta$ , C, margin : REAL

E, epoch, i : INTEGER

ALGORITMA:

w  $\leftarrow$  0

b  $\leftarrow$  0

FOR epoch  $\leftarrow$  1 TO E DO

    Shuffle(X,Y)

    FOR i  $\leftarrow$  1 TO length(X) DO

        margin  $\leftarrow$  Y[i] \* (DotProduct(w, X[i]) + b)

    IF margin < 1 THEN

        w  $\leftarrow$  w -  $\eta$  \* (w - C \* Y[i] \* X[i])

        b  $\leftarrow$  b +  $\eta$  \* C \* Y[i]

    ELSE

        w  $\leftarrow$  w -  $\eta$  \* w

    ENDIF

    ENDFOR

ENDFOR

END

Gambar 4. Pseudocode SVM Yang Dioptimasi Dengan SGD

Performa berupa *confusion matrix* juga digunakan sebagai metrik untuk melakukan evaluasi pada model. *Confusion matrix* dapat dilihat pada Gambar 5. Didapatkan bahwa kelas positif mendapati akurasi yang sangat baik dibandingkan kelas lainnya, hal tersebut juga berhubungan karena *dataset* memiliki rasio 2:1:1 dimana kelas positif memiliki jumlah yang lebih banyak dibandingkan kelas lainnya. Dilihat bahwa kelas netral juga kesusahan dalam menentukan kelas karena terdapat ambiguitas pada kelas netral karena dapat masuk ke kelas negatif maupun positif.

Tabel 10. Data Hasil Training dan Testing

Model	Evaluasi	Kelas	Hasil
SVM Konvensional	Akurasi	Positif	64,42%
		Netral	<b>75,98%</b>
		Negatif	45,59%
	Presisi	Positif	56,56%
		Netral	<b>81,35%</b>
		Negatif	<b>41,04%</b>
	Recall	Positif	81,35%
		Netral	<b>41,04%</b>
		Negatif	53,95%
	F1-Score	Positif	78,55%
		Netral	43,03%
		Negatif	55,08%
Runtime			4,10 detik
	Akurasi	Positif	<b>68,08%</b>
		Netral	74,60%
Negatif		<b>56,76%</b>	
Presisi	Positif	<b>87,79%</b>	
	Netral	36,99%	
	Negatif	<b>59,76%</b>	
Recall	Positif	<b>87,79%</b>	
	Netral	36,99%	
	Negatif	<b>59,76%</b>	
F1-Score	Positif	<b>80,60%</b>	
	Netral	<b>44,18%</b>	
	Negatif	<b>59,99%</b>	
Runtime			<b>1,58 detik</b>

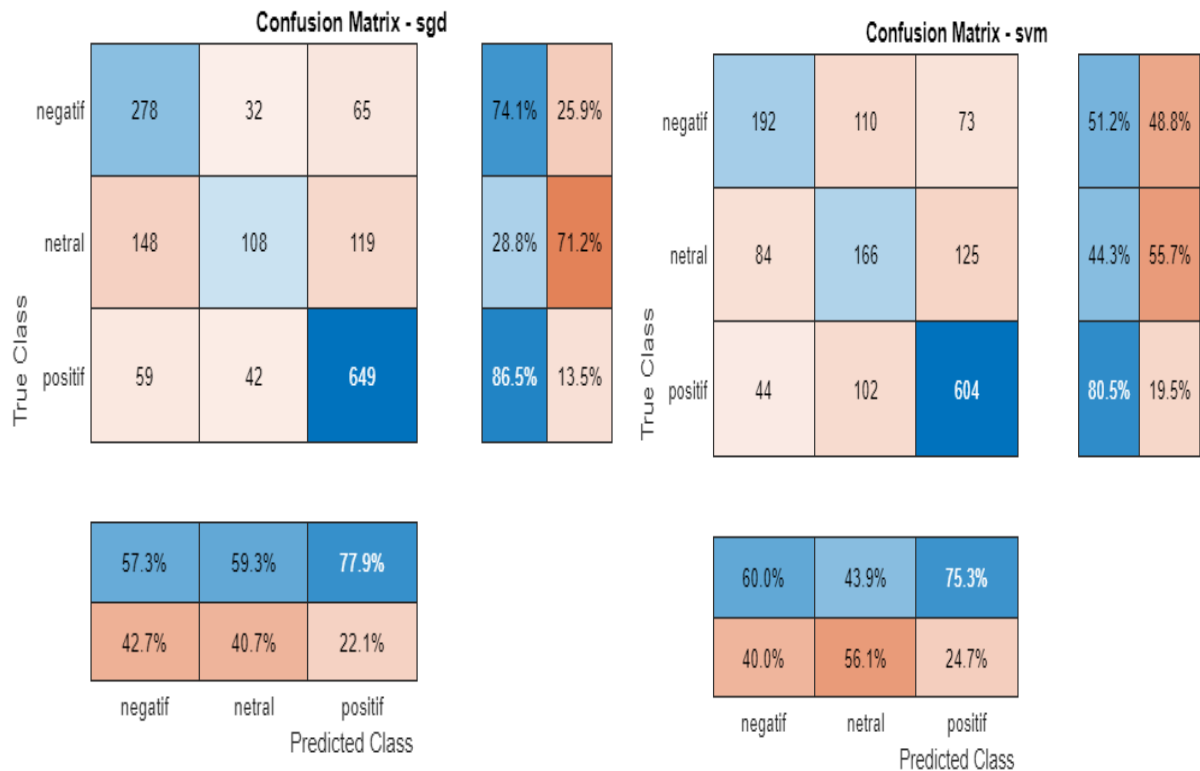
Performa berupa *confusion matrix* juga digunakan sebagai metrik untuk melakukan evaluasi pada model. *Confusion matrix* dapat dilihat pada Gambar 5. Didapatkan bahwa kelas positif mendapati akurasi yang sangat baik dibandingkan kelas lainnya, hal tersebut juga berhubungan karena *dataset* memiliki rasio 2:1:1 dimana kelas positif memiliki jumlah yang lebih banyak dibandingkan kelas lainnya. Dilihat bahwa kelas netral juga kesusahan dalam menentukan kelas karena terdapat ambiguitas pada kelas netral karena dapat masuk ke kelas negatif maupun positif.

Gambar 4 menjelaskan terkait alur algoritma SVM yang dioptimasi dengan SGD. Sebelum menjalankan model perlu diawali dengan inialisasi bobot dan bias. Kemudian dilanjutkan dengan inialisasi jumlah iterasi yang diinginkan dalam menjalankan model tersebut. Pada setiap pengulangan akan dilakukan pengacakan pada data latih yang berfungsi untuk menghindari bias urutan dari data yang ada pada *dataset*. Data pada setiap pengulangan akan diproses dan dilakukan perhitungan nilai *margin* menggunakan SVM untuk mengetahui posisi data terhadap *hyperplane*. Jika nilai *margin* kurang dari satu maka data dianggap salah diklasifikasikan dan bobot serta bias akan di-*update* menggunakan aturan SGD untuk memperbaiki posisi *hyperplane*. Jika nilai *margin* berupa satu maka bobot hanya di-*update* dengan komponen regularisasi. Pada penelitian ini tidak menggunakan parameter  $\gamma$  karena pada penggunaan kode *linearTemplate* pada MATLAB tidak terdapat pengaturan untuk parameter tersebut. Untuk parameter  $c$  pada penelitian ini menggunakan nilai *default* pada MATLAB yaitu berupa 1.

Pada Gambar 5 terlihat bahwa terdapat perbedaan yang mencolok dimana kelas negatif pada model SVM yang dioptimasi dengan SGD memperoleh 278 TP lebih banyak dibandingkan dengan SVM konvensional yang hanya mendapati nilai berupa 192. Namun SVM konvensional lebih baik pada kelas netral dimana memperoleh 166 lebih baik dibandingkan SVM yang dioptimasi dengan SGD. Perbedaan lainnya yang dapat dilihat adalah model SVM yang dioptimasi dengan SGD mampu memperoleh 649 kelas positif lebih baik dibandingkan model SVM konvensional yang hanya mendapati 604.

Namun dapat terlihat bahwa model SVM konvensional lebih baik dalam penentuan kelas netral dibandingkan model yang sudah dioptimasi dengan SGD, hal tersebut menunjukkan SGD kurang mampu dalam menentukan kelas netral dibandingkan model SVM konvensional. Hal lain yang terlihat adalah kelas positif yang cenderung lebih baik dari kedua kelas tersebut, namun karena *dataset* yang tidak berimbang maka hal tersebut tidak bisa menjadi acuan bahwa kelas positif lebih mudah untuk diklasifikasikan dibandingkan kelas netral maupun kelas negatif.

Pada Gambar 6 terlihat bahwa terdapat perbedaan antara model SVM konvensional dengan SVM yang telah dioptimasi dengan SGD. Metrik akurasi yang didapatkan pada kelas SVM konvensional cenderung rendah dengan 64,13% dibandingkan akurasi milik SVM yang telah dioptimasi SGD sebesar 69%. Pada presisi terlihat bahwa model SVM yang telah dioptimasi dengan SGD lebih unggul pada kelas positif dan netral sedangkan model SVM konvensional lebih baik dalam menentukan kelas negatif. Hasil *recall* pada kelas positif dan negatif lebih unggul pada model SVM yang dioptimasi dengan SGD namun kelas netral jauh lebih buruk dibandingkan SVM konvensional yang mendapatkan 44,26% dibandingkan dengan 28,8%. Untuk metrik *F1-Score* yang menghitung secara keseluruhan antara *recall* dan *presisi* dimenangkan oleh model SVM yang dioptimasi dengan SGD pada kelas positif dan negatif namun kalah pada kelas negatif. Hal yang jauh berbeda diantara kedua model tersebut terletak pada *runtime* atau laju komputasi dimana SVM konvensional mendapati hasil 9,88 detik sedangkan SVM yang dioptimasi dengan SGD mampu mendapati 2,99 detik.



Gambar 5. Confusion Matrix SVM SGD dan SVM Konvensional

Akurasi: 0.64133				Akurasi: 0.69			
Presisi per kelas: 0.6	0.43915	0.75312		Presisi per kelas: 0.5732	0.59341	0.77911	
Recall per kelas: 0.512	0.44267	0.80533		Recall per kelas: 0.74133	0.288	0.86533	
F1-score per kelas: 0.55252	0.4409	0.77835		F1-score per kelas: 0.64651	0.38779	0.81996	
Runtime: 9.8856 detik				Runtime: 2.9999 detik			

Gambar 6. Metrik Evaluasi SVM Konvensional dan SVM SGD

Pada Gambar 5 dan Gambar 6 dapat terlihat bahwa terdapat kelebihan dan keterbatasan pada masing model dengan menggunakan *parameter* yang sama dan percobaan yang sama yaitu:

- Kelebihan: model SVM yang dioptimasi dengan SGD lebih baik dalam menentukan kelas positif dan negatif namun masih terbatas pada penentuan kelas netral. Akurasi yang didapatkan oleh model SVM yang dioptimasi dengan SGD juga lebih baik sebesar 69% lebih tinggi dibandingkan SVM konvensional yang hanya mendapati akurasi berupa 64,13%. Terlihat juga pada model SVM yang dioptimasi dengan SGD jauh lebih cepat dengan waktu 3 detik dibandingkan dengan SVM konvensional yang hanya mendapati 9,88 detik. Penggunaan model SVM yang dan dioptimasi dengan SGD juga cenderung lebih ringan dibandingkan SVM konvensional dalam menggunakan memori yang sangat banyak
- Kekurangan: model SVM yang dioptimasi dengan SGD terbatas pada penentuan kelas netral dibandingkan dengan SVM konvensional. Model SVM konvensional juga lebih baik dalam menentukan kelas negatif karena nilai

presisinya melebihi model SVM yang dioptimasi dengan SGD. Parameter yang digunakan harus sesuai dengan topik dan situasi model yang diangkat karena jika parameter tidak sesuai maka model yang bekerja akan kesulitan.

#### 4. KESIMPULAN

Berdasarkan pengujian dan perbandingan hasil kedua model tersebut. Kesimpulan yang dapat diambil sebagai berikut:

- a. Terdapat peningkatan akurasi dan efisiensi komputasi dengan menerapkan model SVM yang dioptimasi dengan SGD karena hasil yang diperoleh sebesar 68,08% melampaui model SVM konvensional yang hanya mencapai 64,42%. Selain dari aspek akurasi, model SVM yang dioptimasi dengan SGD juga menunjukkan efisiensi waktu dengan *runtime* hanya 1,59 detik jauh lebih cepat dibandingkan model dasar yang memerlukan waktu 4,10 detik.
- b. Secara keseluruhan penggunaan SGD dan TF-IDF memberikan dampak positif terhadap hampir keseluruhan metrik evaluasi. Model SVM yang dioptimasi dengan SGD memberikan performa keseluruhan yang lebih stabil dan representative dalam menggambarkan klasifikasi sentimen secara menyeluruh.
- c. Penerapan optimasi SGD pada model SVM mampu menambah laju model dalam melakukan proses analisis sentimen pada data ulasan *e-commerce* yang sangat banyak. Dengan *parameter* yang tepat proses analisis juga akan lebih akurat dan lebih baik dibandingkan SVM konvensional yang memakan banyak proses komputasi namun kurang cepat dibandingkan model SVM yang sudah dioptimasi dengan SGD.

#### UCAPAN TERIMA KASIH

Peneliti banyak mendapatkan bantuan berupa bimbingan, petunjuk, data, serta motivasi dalam melaksanakan penelitian ini. Oleh karena itu, peneliti ingin mengucapkan banyak terima kasih kepada Civitas Akademika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak terutama untuk kedua dosen pembimbing, orang tua, saudara, dan rekan-rekan yang tidak dapat peneliti sebutkan satu per satu.

#### DAFTAR PUSTAKA

- [1] Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), "APJII Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang." Accessed: Aug. 28, 2025. [Online]. Available: <https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>
- [2] "Digital 2024 - We Are Social Indonesia." Accessed: Aug. 28, 2025. [Online]. Available: <https://wearesocial.com/id/blog/2024/01/digital-2024/>
- [3] Tim eDOT, "5 Platform E-Commerce Terbesar di Indonesia 2025!" Accessed: Aug. 28, 2025. [Online]. Available: <https://edot.id/id/articles/5-e-commerce-dengan-pengunjung-terbanyak-di-indonesia-pada-awal-2024-siapa-juaranya>
- [4] C. H. Lin and U. Nuha, "Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy," *J. Big Data*, vol. 10, no. 1, p. 88, Dec. 2023, doi: 10.1186/S40537-023-00782-9.
- [5] V. D. Antonio, S. Efendi, and H. Mawengkang, "Sentiment analysis for covid-19 in Indonesia on Twitter with TF-IDF featured extraction and stochastic gradient descent," *Int. J. Nonlinear Anal. Appl.*, vol. 13, no. 1, pp. 1367–1373, Mar. 2022, doi: 10.22075/IJNAA.2021.5735.
- [6] Kazee Media Intelligence, "Pentingnya Analisis Sentimen bagi E-Commerce." Accessed: Sep. 14, 2025. [Online]. Available: [https://blog.kazee.id/pentingnya-analisis-sentimen-bagi-e-commerce?utm\\_source=chatgpt.com](https://blog.kazee.id/pentingnya-analisis-sentimen-bagi-e-commerce?utm_source=chatgpt.com)
- [7] R. S. Nuraini *et al.*, "ANALISIS SENTIMEN PENGGUNA APLIKASI AGODA DI GOOGLE PLAY Sistem Informasi Universitas Amikom Purwokerto Abstraksi Keywords : Pendahuluan Tinjauan Pustaka Metode Penelitian," vol. 7, no. 1, 2025.
- [8] A. Y. Sain, S. A. S. Mola, A. Y. Huan, and I. R. Nomleni, "Analisis Sentimen Sunscreen Azarine dengan Naïve Bayes di Toko Aneka Kosmetik Kupang pada Marketplace Shopee," *J. SAINTEKOM*, vol. 15, no. 1, pp. 94–105, 2025, doi: 10.33020/saintekom.v15i1.783.
- [9] M. V. Galena, A. S. A. Sadikin, A. Prastyaningrum, R. F. Nugroho, and M. F. F. Mardianto, "Analisis Sentimen Masyarakat terhadap Keamanan Penggunaan E-Commerce B2C Menggunakan Pendekatan Naïve Bayes Berbasis Text Mining untuk Mencegah Penipuan," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 2003–2012, 2024, doi: 10.33379/gtech.v8i3.4846.
- [10] T. P. R. Sanjaya, A. Fauzi, and A. F. N. Masruriyah, "Analisis sentimen ulasan pada e-commerce shopee menggunakan algoritma naive bayes dan support vector machine," *INFOTECH J. Inform. Teknol.*, vol. 4, no. 1, pp. 16–26, 2023, doi: 10.37373/infotech.v4i1.422.
- [11] Nurhayati *et al.*, *Machine Learning*. 2024.
- [12] J. K. Wororomi, S. A. Mandowen, H. E. Manurung, E. H. Pujiarini, H. Yuliana, and R. M. Dinata, *Data Mining (Memahami Pola di Balik Angka)*. Eureka Media Aksara, 2024.