

Penerapan Algoritma *Support Vector Machine* Pada Review Aplikasi TikTok

Nahdah Arlisa Irdianto^{1*}, Raissa Amanda Putri²

Ilmu Komputer, UINSU, Deli Serdang, Indonesia

*Email Corresponding Author: nahdahemail@gmail.com

Abstract

This research aims to evaluate the application of the Support Vector Machine (SVM) algorithm in classifying user reviews of the TikTok application. The research process consists of three main stages, namely: data crawling to collect user reviews, data preprocessing to clean and prepare the data, and a classification process using the SVM algorithm. In the crawling stage, TikTok user reviews are collected from various sources. Then, in the preprocessing stage, the raw data is cleaned from irrelevant elements and converted into a format that can be processed by the SVM algorithm. Finally, in the classification stage, the SVM algorithm is applied to classify the reviews. The research results show that from a total of 399 test data, the SVM algorithm succeeded in classifying 299 data correctly, resulting in an accuracy rate of 75%. This shows that the SVM algorithm has good capabilities in classifying user reviews of the TikTok application, although there is still room for improvement. This research contributes to the understanding of the application of SVM algorithms for sentiment analysis and text classification in the context of social media application reviews.

Keywords: *Support vector machine; Classification; TikTok; Sentiment analysis*

Abstrak

Penelitian ini bertujuan untuk mengevaluasi penerapan algoritma *Support Vector Machine* (SVM) dalam klasifikasi ulasan pengguna aplikasi *TikTok*. Proses penelitian terdiri dari tiga tahap utama, yaitu: crawling data untuk mengumpulkan ulasan pengguna, preprocessing data untuk membersihkan dan mempersiapkan data, serta proses klasifikasi menggunakan algoritma SVM. Pada tahap crawling, ulasan pengguna *TikTok* dikumpulkan dari berbagai sumber. Kemudian, pada tahap preprocessing, data mentah dibersihkan dari elemen-elemen yang tidak relevan dan diubah menjadi format yang dapat diproses oleh algoritma SVM. Terakhir, pada tahap klasifikasi, algoritma SVM diterapkan untuk mengklasifikasikan ulasan-ulasan tersebut. Hasil penelitian menunjukkan bahwa dari total 399 data uji, algoritma SVM berhasil mengklasifikasikan 299 data dengan benar, menghasilkan tingkat akurasi sebesar 75%. Hal ini menunjukkan bahwa algoritma SVM memiliki kemampuan yang baik dalam melakukan klasifikasi ulasan pengguna aplikasi *TikTok*, meskipun masih terdapat ruang untuk perbaikan. Penelitian ini memberikan kontribusi dalam pemahaman tentang penerapan algoritma SVM untuk analisis sentimen dan klasifikasi teks dalam konteks ulasan aplikasi media sosial.

Kata kunci: *Support vector machine; Klasifikasi; TikTok; Analisis sentimen*

1. Pendahuluan

Media sosial telah mengalami perkembangan pesat dalam dekade terakhir, membentuk cara kita berkomunikasi, berinteraksi, dan berbagi informasi secara fundamental. *TikTok*, salah satu platform media sosial terkini, telah mendapatkan perhatian besar dari pengguna di seluruh dunia. Dengan lebih dari satu miliar pengguna aktif setiap bulan, *TikTok* telah menjadi kekuatan besar dalam ekosistem media sosial global. Menurut Statista dalam publikasinya pada statista.com, Indonesia menduduki peringkat ke-2 dengan pengguna terbanyak di dunia dengan total pengguna 112 juta per tahun 2023.

TikTok memungkinkan penggunaannya untuk membuat, membagikan, dan mengeksplorasi video pendek dengan cara yang intuitif dan menghibur. Aplikasi ini menawarkan beragam konten, mulai dari tarian, komedi, hingga konten pendidikan. Namun,

seperti halnya platform media sosial lainnya, TikTok juga menghadapi berbagai pandangan dan perasaan dari penggunanya. Review dan komentar di *platform Play Store* seringkali menjadi saluran di mana pengguna menyuarakan pendapat, pengalaman, dan sentimen mereka terhadap aplikasi *TikTok* [1]

Di Indonesia, TikTok telah mengalami pertumbuhan pesat. Berbagai peristiwa dan tren di dalam dan di luar platform ini telah memengaruhi bagaimana pengguna di Indonesia merespons dan merasakan *TikTok*. Isu-isu seperti privasi data, peraturan pemerintah, serta tren dan tantangan yang bermunculan memengaruhi pandangan pengguna terhadap aplikasi ini. Salah satu fitur TikTok yang populer adalah *TikTok Shop*. *TikTok Shop* adalah fitur yang memungkinkan pengguna TikTok untuk berbelanja langsung dari aplikasi. Pengguna dapat menemukan berbagai produk yang ditawarkan oleh berbagai penjual, seringkali berupa barang-barang yang menjadi tren di platform ini. Namun saat ini permasalahan dan berita mengenai potensi penutupan *TikTok Shop* muncul mulai muncul di Indonesia. Penutupan *TikTok Shop* menjadi isu yang kontroversial karena hal ini dapat berdampak pada ekosistem penjual dan bisnis kecil yang menggunakan platform ini sebagai salah satu saluran penjualan mereka [2].

Analisis sentimen adalah salah satu proses yang memanfaatkan teknik data mining. *Data mining*, secara sederhana merupakan suatu langkah ekstraksi untuk mendapatkan informasi penting yang sifatnya implisit dan belum diketahui [3]. Analisis sentimen merupakan teknik untuk menggali data opini, dan secara otomatis dapat memahami dan memproses data teks untuk melihat sentimen yang terkandung dalam opini tersebut, baik itu positif maupun negatif. Dalam menganalisis opini, terdapat banyak metode yang dapat digunakan, salah satunya dengan teknik klasifikasi text yaitu metode *Support Vector Machine*.

Algoritma *Support Vector Machine* merupakan algoritma yang umum digunakan dalam proses klasifikasi, tidak terkecuali dalam penelitian analisis sentiment [4]. Algoritma SVM akan diintegrasikan ke dalam bahasa pemrograman Python untuk melakukan analisis sentimen karena bahasa pemrograman *Python* memiliki banyak *library* yang bisa digunakan untuk keperluan data mining, beberapa diantaranya seperti *pandas*, *numpy*, *matplotlib*, *scikit-learn*, dll [5]. Kode python *Google Colaboratory* atau *google colab* adalah sebuah IDE untuk pemrograman *Python* dimana pemrosesan akan dilakukan oleh *server Google* yang memiliki perangkat keras dengan performa yang tinggi hon akan dioleh menggunakan teks editor *google colaboratory* atau *google colab*. Dalam penelitian ini pustaka yang dibutuhkan adalah *Keras*, *TensorFlow*, *NumPy*, *Pandas*, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat *Matplotlib*. Dari sisi perangkat keras, *Google Colab* menyediakan layanan berupa media penyimpanan yang terintegrasi dengan *Google Drive*, prosesor yang berupa *CPU*, *GPU*, dan *TPU*, serta *RAM* [6]

Menurut penelitian [7] mengenai Perbandingan *Naïve Bayes*, SVM, dan k-NN untuk Analisis Sentimen *Gadget* Berbasis Aspek, dengan membandingkan algoritma NB, SVM, dan k-NN, terbukti bahwa model klasifikasi SVM menunjukkan hasil terbaik. Rata-rata *accuracy* SVM sebesar 96.43% dari empat aspek, yaitu aspek desain sebesar 94.40%, aspek harga sebesar 97.44%, aspek spesifikasi sebesar 96.22%, dan aspek citra merk sebesar 97.63%. Kemudian penelitian yang dilakukan oleh Rahman, A., & Dinata, M [8] yang berjudul Penerapan Algoritma *Support Vector Machine* Dan *Particle Swarm Optimization* Untuk Pembuatan Rating Aplikasi *Google Playstore* memperoleh hasil bahwa Nilai Akurasi Dari Algoritma *Support Vector Machine* Cukup Tinggi Rata-Rata Berkisar 90% Untuk Semua Aplikasi. Setelah itu pada penelitian Suryawan, I. W. B., Utami, N. W., & Fredlina, K. Q [9] yang berjudul Analisis Sentimen Review Wisatawan Pada Objek Wisata Ubud Menggunakan Algoritma *Support Vector Machine* memperoleh kesimpulan bahwa Hasil Pengujian Menggunakan *Confusion Matrix* Pada Metode Klasifikasi *Support Vector Machine* Mendapatkan Hasil *Accuracy* Sebesar 84,01%, *Recall* sebesar 89,83%, *Precision* Sebesar 90,40% Dan *F1-Score* Sebesar 90,11%. Sedangkan peneltian dari Chrisdiyanti, I. N., Fa'rifah, R. Y., & Pratiwi, O. N [10] dengan judul Klasifikasi Review Customer Di E-Commerce Bukalapak Menggunakan Metode *Support Vector Machine* (SVM) memperoleh kesimpulan bahwa Hasil Evaluasi Yang Telah Dilakukkberdasarkan Proses Training Dengan Akurasi Tertinggi Yaitu Pada Rasio 60:40, Didapatkan Akurasi Sebesar 85% Yang Masuk Ke Dalam Kategori Good Classification, *Recall* Sebesar 79%, *Precision* Sebesar 89% Dan *F1-Score* Sebesar 84%. Dan pada penelitian dari Radiena, G., & Nugroho, A [11] (2023) dengan judul Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Kai Access Menggunakan Metode *Support Vector Machine* memperoleh hasil bahwa Berdasarkan Pengujian Dan Nilai Dari Akurasi Yang Dihasilkan Dari Aspek *Learn*

Ability Sebesar 94,73%, Aspek *Efficiency* Sebesar 94,38%, Aspek *Error* Sebesar 85,13%, Dan Aspek *Satisfaction* Sebesar 87,26% .

Pada penelitian ini akan dilakukan klasifikasi ulasan pengguna TikTok pada aplikasi *PlaysStore* ke dalam dua kelas, yaitu positif dan negatif. Pengklasifikasian akan dilakukan dengan metode *Support Vector Machine* (SVM). SVM adalah algoritma yang sudah mendapatkan pengakuan luas untuk klasifikasi dengan akurasi yang baik. Umumnya, masalah dalam dunia nyata jarang yang bersifat linear separable. Untuk menyelesaikan problem nonlinear, SVM dimodifikasi dengan memasukkan fungsi kernel, yang bisa membuat klasifikasi menggunakan SVM dapat beroperasi dalam ruang dimensi tinggi [12]

2. Metodologi

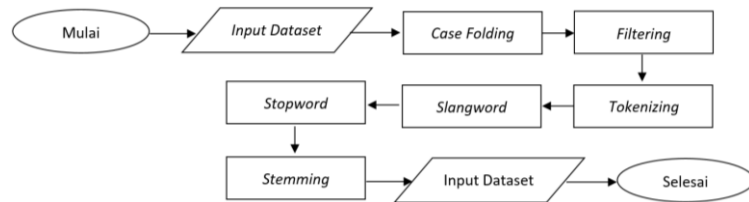
Deskripsi Dalam hal ini yang menjadi subjek kajian adalah informasi dari ulasan aplikasi TikTok di Playstore. Perencanaan diperlukan dalam pelaksanaan penelitian ini agar prosedur dapat berjalan sesuai dengan yang diharapkan. Proses penelitian ini akan diimplementasikan menggunakan algoritma *Support Vector Machine* (SVM) untuk menganalisis sentimen dari ulasan pengguna terhadap aplikasi TikTok. Penelitian ini bertujuan untuk memahami persepsi pengguna terkait fitur-fitur, performa, dan pengalaman penggunaan aplikasi TikTok, serta untuk mengidentifikasi pola-pola sentimen yang dominan.

2.1. Pengumpulan Data

Library *BeautifulSoup* dan *Selenium* dalam bahasa pemrograman Python digunakan untuk *scraping* ulasan aplikasi TikTok di Playstore selama langkah pengumpulan data. Ulasan pengguna terbaru per januari 2024 yang mencakup berbagai aspek ulasan, *username* dan *rating*. Untuk diproses lebih lanjut, dataset yang telah dikumpulkan disimpan dalam format *.csv*.

2.2. Preprocessing Data

Dataset tersebut ditambahkan ke kategori data tidak terstruktur mengikuti prosedur pengumpulan data. Kumpulan data menjalani prapemrosesan teks sebelum analisis lebih lanjut untuk menghilangkan dan mengatasi data yang berisik serta memberikan hasil komputasi terbaik. *case folding*, *filtering*, *tokenizing*, dan *stopword removal* semuanya disertakan dalam alur proses persiapan teks. Pelabelan data dilakukan secara manual setelah dibersihkan. Diagram alir *figure 1*. untuk persiapan teks disajikan di bawah ini [13].



Gambar 1. *Preprocessing Data* [14]

2.3. Vektorisasi TF-IDF

Untuk memberikan kata (*term*) berdasarkan nilai suatu dokumen atau tweet, penelitian ini menggunakan teknik pembobotan kata yang disebut TF-IDF (*Term Frequency-Inverse Document Frequency*) yang berguna untuk menimbang (Putra, 2022). Persamaan (1) memberikan rumus algoritma TF-IDF sebagai berikut [15] :

$$w_{ij} = tf_{ij} \times \log \left(\frac{D + 1}{df_i + 1} \right) + 1 \dots\dots\dots (1)$$

Keterangan :

- w_{ij} = bobot kata t terhadap dokumen
- tf_{ij} = jumlah kemunculan kata t_i
- D = jumlah dokumen
- df_{ij} = jumlah kemunculan kata pada D_j

2.4. Support Vector Machine

Pada tahap ini, data dibagi menjadi dua kelompok: data pelatihan (80%) dan data pengujian (20%). Data pengujian digunakan sebagai tolok ukur dan untuk mengevaluasi kinerja algoritma atau model yang telah diajarkan sebelumnya, sedangkan data pelatihan digunakan untuk melatih algoritma dan membangun pemodelan. Prosedur pemodelan SVM adalah langkah selanjutnya.

Dengan memanfaatkan fungsi kernel untuk melakukan modifikasi matematis pada ruang pembelajaran, pendekatan SVM memungkinkan komputasi untuk masalah linier. Ide utama dalam pendekatan SVM untuk mengkategorikan data adalah memilih hyperplane optimal untuk memberikan pemisahan atau jarak antara dua kelas yang telah ditetapkan. Menemukan *hyperplane* ideal yang memberikan pemisahan atau jarak antara dua kelas adalah cara SVM beroperasi. *hyperplane* dengan margin terbesar [16].

Tahapan dalam metode *Support Vector Machine* adalah sebagai berikut:

- 1) Menentukan kata yang sering muncul dari tiap dokumen atau *review* yang digunakan.
- 2) Menentukan inisialisasi awal untuk nilai $\alpha=0$, $C=1$, $\lambda=0.5$ dan $\gamma=0.1$
- 3) Menghitung matriks dengan rumus (2):

$$D_{ij} = y_i y_j (K(x_i \rightarrow \cdot x_j \rightarrow) + \lambda^2) \dots\dots\dots (2)$$

Keterangan:

- D_{ij} = elemen matriks data ke-ij
- y_i = label data ke-i
- y_j = label data ke-j
- $K(x_i \rightarrow \cdot x_j \rightarrow)$ = nilai kernel linear
- λ = *lambda*

d) Untuk data ke n = 1,2,3, .. n gunakan persamaan (3),(4),(5) berikut.

$$E_i \sum_{j=1}^n \alpha_j D_{ij} \dots\dots\dots (3)$$

$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i, -\alpha_i], C - \alpha_i]\} \dots\dots\dots (4)$$

$$\alpha_i = \alpha_i + \delta \alpha_i \dots\dots\dots (5)$$

Keterangan:

- E = Nilai *error*
- α_j = *alfa* ke-j
- D_{ij} = matriks Hessian
- $\delta \alpha_i$ = *delta alfa* ke-i
- γ = *learning rate*
- α_i = *alfa* ke-i
- C = *variable slack*
- $\delta \alpha$ = *delta alpha*

e) Mencari nilai *bias* (b) dengan menggunakan persamaan (6)

$$b = -\frac{1}{2} [w \cdot x^+ + w \cdot x^-] \dots\dots\dots (6)$$

f) Perhitungan Keputusan

$$h(x) = \begin{cases} +1 & \text{jika } w \cdot x + b \geq 0 \\ -1 & \text{jika } w \cdot x + b < 0 \end{cases}$$

Jika hasil perhitungan keputusan lebih dari sama dengan 0 maka nilai adalah +1, maka termasuk kelas positif dan jika hasil perhitungan keputusan kurang dari 0 maka nilai nilai -1, maka termasuk kelas negatif. Perhitungan keputusan dengan menggunakan persamaan (7).

$$h(x) = w \cdot x + b \quad \dots\dots\dots (7)$$

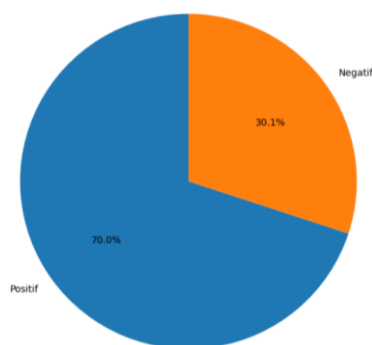
2.5. Evaluasi

Fase evaluasi digunakan untuk menilai kebenaran pemodelan setelah diterapkan pada kumpulan data pelatihan. Menerapkan *classification report* untuk menentukan *precision*, *recall*, *f1-score*, dan *accuracy* akan memungkinkan Anda membandingkan hasil dari dua kumpulan data terpisah.

3. Hasil dan Pembahasan

3.1. Persiapan Data

Secara Data ulasan yang disimpan dalam format .csv dan diolah menggunakan bahasa pemrograman Python. Jumlah total data ulasan yang berhasil dikumpulkan untuk analisis ini adalah 2000 ulasan. Dimana setelah melakukan proses pelabelan maka diketahui sebanyak 70% ulasan bersifat positif dan 30% ulasan bersifat negatif.



Gambar 2. Persebaran Data

3.2. Text preprocessing

Persiapan teks dilakukan pada awalnya setelah pengumpulan data Twitter selesai. Dataset yang digunakan berisi data yang tidak terstruktur, oleh karena itu dilakukan hal ini. Daftar berikut menjelaskan berbagai langkah persiapan teks yang digunakan dalam penelitian ini:

- 1) *Case Folding* adalah tahap pengubahan semua huruf campuran baik itu *uppercase*, atau *lowercase*, menjadi *lowercase* (huruf kecil) semua.
- 2) *Filtering* adalah tahap pembersihan data dari tanda baca, simbol atau elemen yang tidak dibutuhkan seperti URL.
- 3) *Tokenizing* adalah tahap pemecahan teks menjadi kata-kata, seperti "saya pergi ke sekolah" menjadi "saya", "pergi", "ke", "sekolah".
- 4) *Stopword Removal* adalah tahap penghapusan kata sambung seperti "ke", "di", "dan", "dia", "kami", "aku", "saya".

Pada table 1. dibawah ini disajikan contoh data uji setelah melewati tahap *preprocessing* sehingga diperoleh data uji yang siap dianalisis

Table 1. Hasil *text preprocessing*

Model	Akurasi
Input	sat saya membuka aplikasi ini saya gabisa daftar tapi masih sedikit keren bisa offline
Output	['sat', 'buka', 'enggak bisa', 'daftar', 'keren', 'offline']

3.3. Vektorisasi TF-IDF

Metode vektorisasi memiliki keunggulan dalam memberikan nilai pada kata-kata dalam dokumen atau tweet, sehingga memungkinkan dilakukannya pemrosesan dan pencarian pola ketika dokumen atau ulasan tersebut memasuki tahap klasifikasi. Dengan memeriksa terlebih dahulu sampel data tweet kemudian melakukan perhitungan berdasarkan rumus, metode vektorisasi menggunakan teknik perhitungan pembobotan kata atau TF-IDF.

Contoh data latih:

D1 = ['lumayan', 'mudah', 'aplikasi']

D2 = ['update', 'kecewa', 'komentar', 'malas', 'susah']

D3 = ['posting', 'komentar']

D4 = ['buka', 'sandil', 'kadang', 'versi', 'akses', 'ikut']

D5 = ['tambah', 'lagu', 'bebas', 'nyaman', 'mudah', 'lagu', 'enak']

Setelah tahap *preprocessing* langkah yang dilakukan selanjutnya adalah pembobotan TF-IDF. Adapun hasil pembobotan yang dilakukan dapat dilihat pada tabel berikut.

Table 2. Hasil vektorisasi TF-IDF

Term	TF-IDF				
	D1	D2	D3	D4	D5
lumayan	0.206	0	0	0	0
mudah	0.166	0	0	0	0.166
aplikasi	0.206	0	0	0	0
update	0	0.206	0	0	0
kecewa	0	0.206	0	0	0
komentar	0	0.166	0.166	0	0
malas	0	0.206	0	0	0
susah	0	0.206	0	0	0
posting	0	0	0.206	0	0
buka	0	0	0	0.206	0
sandi	0	0	0	0.206	0
kadang	0	0	0	0.206	0
versi	0	0	0	0.206	0
akses	0	0	0	0.206	0
ikut	0	0	0	0.206	0
tambah	0	0	0	0	0.206
bebas	0	0	0	0	0.206
nyaman	0	0	0	0	0.206
lagu	0	0	0	0	0.412
enak	0	0	0	0	0.206

3.4. Proses Klasifikasi *Support Vector Machine*

Setelah data dibersihkan dan distrukturkan, langkah selanjutnya adalah melakukan klasifikasi dengan menggunakan algoritma *Support Vector Machine*. Tahap pertama dari proses klasifikasi adalah membagi data ke data latih dan data uji. Pada penelitian ini menggunakan rasio 8:2 untuk perbandingan data latih dan data uji. Data latih digunakan untuk mempelajari karakteristik dan perbedaan kedua kelas positif dan negatif sedangkan data uji digunakan untuk melihat persentase keberhasilan ketika mengklasifikasikan dengan benar.

Pada klasifikasi yang dilakukan, jenis kernel yang digunakan adalah kernel RBF karena data yang dimasukkan merupakan data non-linear. Berikut adalah persamaan yang digunakan untuk menghitung nilai kernel RBF.

$$K(x_i, x_j) = \exp(-\gamma * |x_i - x_j|) \dots\dots\dots (8)$$

Adapun hasil dari perhitungan kernel RBF dengan nilai *gamma* = 1 dari data sampel yang dimiliki sebagaimana yang disajikan pada tabel dibawah ini

Table 2. Hasil perhitungan kernel RBF

No.	1	2	3	..	19	20
1	1	0.971	1	..	0.809	0.919
2	0.971	1	0.971	..	0.916	0.971
3	1	0.971	1	..	0.809	0.919
4	0.919	0.907	0.919	.	0.809	0.919
5	0.919	0.907	0.919	..	0.809	0.919
6	0.907	0.896	0.907	..	0.799	0.907
7	0.919	0.907	0.919	..	0.809	0.919
8	0.919	0.907	0.919	..	0.809	0.919
9	0.919	0.907	0.919	..	0.809	0.919
10	0.919	0.907	0.919	..	0.809	0.919
11	0.919	0.907	0.919	..	0.809	0.919
12	0.919	0.907	0.919	..	0.809	0.919
13	0.919	0.907	0.919	..	0.809	0.919
14	0.919	0.907	0.919	..	0.809	0.919
15	0.919	0.907	0.919	..	0.809	0.919
16	0.919	0.971	0.919	..	0.959	1
17	0.919	0.971	0.919	..	0.959	1
18	0.919	0.971	0.919	..	0.959	1
19	0.809	0.916	0.809	..	1	0.959
20	0.919	0.971	0.919	..	0.959	1

Perhitungan untuk data pertama

$$X = [0.206, 0, 0, 0, 0]$$

Hitung Jarak Kuadrat Euclidean

$$\begin{aligned} \|x-x\|^2 &= (0.206-0.206)^2 + (0.000-0.000)^2 + (0.0-0.0)^2 + (0.0-0.0)^2 + (0.000-0.000)^2 \\ &= 0 + 0 + 0 + 0 = 0 \end{aligned}$$

Terapkan persamaan kernel RBF

$$K(x_i, x_j) = \exp(-\gamma * |x_i - x_j|) = \exp(-1 * 0) = \exp(0) = 1$$

Setelah nilai kernel diketahui, langkah selanjutnya adalah perhitungan matrix Hessian. Sebelum melakukan perhitungan matrix Hessian akan ditetapkan beberapa parameter yang antara lain adalah α , C , γ , λ dan iterasi maksimum. Berikut uraian dari parameter yang akan digunakan pada tahap perhitungan matrix Hessian. Langkah perhitungan matriks Hessian dimulai dengan menginisiasi nilai $\alpha = 0$

Table 3. Hasil perhitungan matriks hessian

No.	1	2	3	..	19	20
1	1.250	1.221	1.250	..	1.059	1.169
2	1.221	1.250	1.221	..	1.166	1.221
3	1.250	1.221	1.250	..	1.059	1.169
4	1.169	1.157	1.169	.	1.059	1.169
5	1.169	1.157	1.169	..	1.059	1.169
6	-	-	-	..	-	-
6	1.157	1.146	1.157	..	1.049	1.157
7	-	-	-	..	-	-
7	1.169	1.157	1.169	..	1.059	1.169
8	1.169	1.157	1.169	..	1.059	1.169
9	1.169	1.157	1.169	..	1.059	1.169
10	1.169	1.157	1.169	..	1.059	1.169
11	-	-	-	..	-	-
11	1.169	1.157	1.169	..	1.059	1.169
12	1.169	1.157	1.169	..	1.059	1.169
13	-	-	-	..	-	-
13	1.169	1.157	1.169	..	1.059	1.169

No.	1	2	3	..	19	20
14	1.169	1.157	1.169	..	1.059	1.169
15	1.169	1.157	1.169	..	1.059	1.169
16	1.169	1.221	1.169	..	1.209	1.250
17	1.169	1.221	1.169	..	1.209	1.250
18	1.169	1.221	1.169	..	1.209	1.250
19	1.059	1.166	1.059	..	1.250	1.209
20	1.169	1.221	1.169	..	1.209	1.250

Perhitungan untuk data pertama :

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) = 1 * 1 * 1 + 0.5^2 = 1 + 0.25 = 1.25$$

Langkah selanjutnya setelah perhitungan nilai matriks hessian adalah melakukan perhitungan *sequential training*. Pada perhitungan awal, iterasi dimulaidari iterasi 0. Karena nilai α awal masih bernilai 0. Maka hasil perhitungan *sequential training* adalah sebagaimana yang ditampilkan pada tabel dibawah ini

Table 4. Nilai *error* awal

No.	E_i
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

Nilai error awal ditunjukkan pada tabel sebelumnya kemudian digunakan pada persamaan (4) untuk mendapatkan nilai delta alpha.

Table 5. Nilai *delta alpha* awal

No.	$\delta\alpha$
1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
10	0.1
11	0.1

No.	$\delta\alpha$
12	0.1
13	0.1
14	0.1
15	0.1
16	0.1
17	0.1
18	0.1
19	0.1
20	0.1

Contoh perhitungan pada data pertama.

$$\begin{aligned}\delta\alpha_i &= \min\{\max[\gamma(1 - E_i, -\alpha_i), C - \alpha_i]\} \\ &= \min\{\max[0.1(1-0), -0], 1-0\} \\ &= \min\{0.1, 1\} \\ &= 0.1\end{aligned}$$

Kemudian dari hasil $\delta\alpha$ yang didapatkan dilakukan perhitungan untuk nilai α_i dengan persamaan (5).

Table 6. Nilai *alpha* baru

No.	$\delta\alpha$
1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
10	0.1
11	0.1
12	0.1
13	0.1
14	0.1
15	0.1
16	0.1
17	0.1
18	0.1
19	0.1
20	0.1

Contoh perhitungan untuk data pertama.

$$\begin{aligned}\alpha_i &= \alpha_i + \delta\alpha_i \\ &= 0 + 0.1 \\ &= 0.1\end{aligned}$$

Proses *sequential training* akan terus berulang sampai nilai error yang diinginkan memenuhi target. Pada contoh ini, iterasi maksimal pada adalah 3, oleh karena itu proses diulangi sampai 3 kali iterasi untuk mendapat nilai α_i yang diperlukan untuk mencari *support vector*. Tabel dibawah ini menampilkan hasil perhitungan α_i pada iterasi ke-3

Table 7. Nilai α setelah iterasi 3

No.	$\delta\alpha$
1	0.257893
2	0.255541
3	0.257893
4	0.256946
5	0.256946
6	0.256688
7	0.256946
8	0.256946
9	0.258367
10	0.343225
11	0.259225
12	0.259225
13	0.259225
14	0.343225
15	0.343225
16	0.25599
17	0.25599
18	0.25599
19	0.252738
20	0.25599

Setelah proses perhitungan yang diuraikan sebelumnya, dilakukan perhitungan untuk mencari *support vector* masing-masing dokumen. Dari nilai α terbaru. Diambil nilai terbesar dari masing-masing kelasnya.

Table 8. Penentuan hyperplane

No.	1	2	..	20	α	Kelas
1	1.250	1.221	..	1.169	0.2578	1
2	1.221	1.250	..	1.221	0.2555	1
3	1.250	1.221	..	1.169	0.2578	1
4	1.169	1.157	..	1.169	0.2569	1
5	1.169	1.157	..	1.169	0.2569	1
6	-1.157	-	..	-	0.2566	-1
7	-1.169	-	..	-	0.2569	-1
8	1.169	1.157	..	1.169	0.2569	1
9	1.169	1.157	..	1.169	0.2583	1
10	1.169	1.157	..	1.169	0.3432	1
11	-1.169	-	..	-	0.2592	-1
12	1.169	1.157	..	1.169	0.2592	1
13	-1.169	-	..	-	0.2592	-1
14	1.169	1.157	..	1.169	0.3432	1
15	1.169	1.157	..	1.169	0.3432	1

No.	1	2	..	20	α	Kelas
16	1.169	1.221	..	1.250	0.2559	1
17	1.169	1.221	..	1.250	0.2559	1
18	1.169	1.221	..	1.250	0.2559	1
19	1.059	1.166	..	1.209	0.2527	1
20	1.169	1.221	..	1.250	0.2559	1

Langkah selanjutnya adalah menghitung fungsi kernel masing-masing kelas menggunakan nilai α tertinggi pada setiap kelas. Perhitungan manual untuk $K(x_i, x^+)$ didapatkan berdasarkan nilai α_i terbesar dari kelas positif yaitu 0.343225, sedangkan nilai $K(x_i, x^-)$ didapatkan berdasarkan nilai α_i terbesar dari kelas negatif yaitu 0.343225. Berdasarkan nilai tersebut dilihat dari matriks hessian kolom ke 11 dan 10. Sehingga akan diperoleh nilai kernel dari kelas positif sebesar 7.34175 dan kernel dari kelas negatif sebesar -5.55154. Dengan diketahui nilai kernel dari masing-masing kelas, maka nilai tersebut digunakan untuk mencari nilai bias dengan menjumlah nilai dari kernel dari kedua kelas kemudian mengalikan dengan -0.5 sehingga diperoleh nilai bias sebesar -0.895.

Setelah nilai bias diperoleh, kemudian masuk ketahap pengujian. Langkah pertama pada tahap pengujian adalah melakukan perhitungan nilai TF-IDF pada data uji.

Table 9. Nilai TF-IDF data uji

No.	<i>Term</i>	<i>TF-IDF</i>
1	punya	0.2673
2	kendala	0.2673
3	update	0.4019
4	posting	0.4019
5	komentar	0.3243
6	live	0.2673
7	putar	0.2673
8	sulit	0.2673

Setelah *term* ternormalisasi data uji di asosiasikan dengan model data latih, maka diperoleh vektor data uji 1 sebagai berikut :

$$V_{uji} = [0, 0, 0, 0.4019, 0.2673, 0.3243, 0, 0, 0.4019, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

Sedangkan vektor dukungan yang digunakan sesuai dengan vektor pada nilai alpha tertinggi untuk setiap kelas.

$$V_1 = [1.168, 1.157, 1.168, 1.168, 1.168, -1.157, -1.168, 1.168, 1.168, 1.25, -1.25, 1.25, -1.25, 1.25, 1.25, 1.168, 1.168, 1.168, 1.059, 1.168]$$

$$V_2 = [-1.168, -1.157, -1.168, -1.168, -1.168, 1.157, 1.168, -1.168, -1.168, -1.25, 1.25, -1.25, 1.25, -1.25, -1.25, -1.168, -1.168, -1.168, -1.059, -1.168]$$

$$\begin{aligned} |V_{uji}V_1|^2 &= [(0-1.168)^2 + (0-1.157)^2 + (0-1.168)^2 + (0.4019-1.168)^2 + (0.2673-1.168)^2 + (0.3243 - (-1.157))^2 + (0-(-1.168))^2 + (0-1.168)^2 + (0.4019-1.168)^2 + (0-1.25)^2 + (0-(-1.25))^2 + (0-1.25)^2 + (0-(-1.25))^2 + (0-1.25)^2 + (0-1.25)^2 + (0-1.168)^2 + (0-1.168)^2 + (0-1.168)^2 + (0-1.059)^2 + (0-1.168)^2] \\ &= 25.14 \end{aligned}$$

Nilai kernel $V_{uji}V_1$:

$$K(V_{uji}, V_1) = \exp(-0.1 * 25.14) = 0.081$$

$$|V_{uji}V_2|^2 = [(0 - (-1.168))^2 + (0 - (-1.157))^2 + (0 - (-1.168))^2 + (0.4019 - (-1.168))^2 + (0.2673 - (-1.168))^2 + (0.3243 - (-1.157))^2 + (0 - 1.168)^2 + (0 - (-1.168))^2 + (0.4019 - (-1.168))^2 + (0 - (-1.25))^2 + (0 - 1.25)^2 + (0 - (-1.25))^2 + (0 - 1.25)^2 + (0 - (-1.25))^2 + (0 - (-1.25))^2 + (0 - (-1.168))^2 + (0 - (-1.168))^2 + (0 - (-1.168))^2 + (0 - (-1.059))^2 + (0 - (-1.168))^2]$$

$$= 29.14$$

Nilai kernel $V_{uji}V_2$:

$$K(V_{uji}, V_2) = \exp(-0.1 * 29.14) = 0.054$$

Langkah terakhir, dengan memanfaatkan nilai bias yang telah diperoleh dari proses pelatihan, cari nilai $f(x)$ untuk data uji dengan menggunakan persamaan dibawah ini. Apabila data tersebut menghasilkan nilai positif, maka data tersebut digolongkan menjadi kelas positif, apabila data tersebut bernilai negatif, maka data tersebut digolongkan menjadi kelas negatif sedangkan apabila hasil bernilai 0 maka digolongkan menjadi kelas netral:

$$f(x) = \text{sign}(\sum K(V_{uji}, V_i) + b)$$

$$f(x) = \text{sign}(0.081 + 0.054 + (-0.895))$$

$$f(x) = \text{sign}(-0.76)$$

Setelah melakukan proses pengujian pada data uji 1, diketahui bahwa pada fungsi klasifikasi mendapatkan nilai -0.76 sehingga data uji 1 diklasifikasikan sebagai kategori kelas -1 dimana kelas -1 ada kelas negatif.

3.5. Evaluasi

Setelah selesai melakukan proses pengujian pada algoritma *Support Vector Machine* maka akan diperoleh hasil. Dimana hasil yang dimaksud adalah label dari data uji yang diperoleh dari model pada saat proses pelatihan. Hasil klasifikasi data uji yang berupa kelas ulasan yang didapatkan dari program akan dibandingkan dengan data kelas sebenarnya sehingga akan diketahui nilai *accuracy*, *precision*, *recall* dan *f1-score* dari model yang digunakan terhadap dataset. Berikut adalah confusion matrix yang akan menjadi acuan dari pencarian nilai *accuracy*, *precision*, *recall* dan *f1-score*

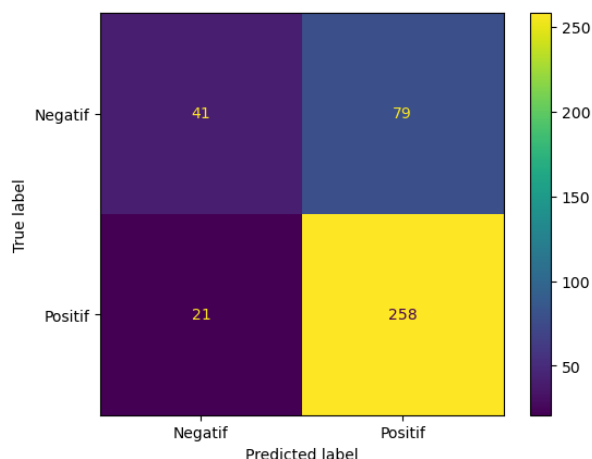


Figure 3. Confusion matrix

Melalui figure 3, maka dapat dihitung nilai dari *accuracy*, *precision*, *recall* dan *f1-score* dengan menggunakan persamaan dibawah ini.

$$\begin{aligned}
 \text{Accuracy} &= \frac{41+258}{41+258+79+21} \times 100\% = 75\% \\
 \text{Precision} &= \frac{258}{71+258} \times 100\% = 77\% \\
 \text{Recal} &= \frac{21+258}{2 \times 77 \times 92} \times 100\% = 92\% \\
 \text{F1-Score} &= \frac{258}{77+92} \times 100\% = 84\%
 \end{aligned}$$

4. Simpulan

Algoritma *Support Vector Machine* dapat digunakan untuk klasifikasi ulasan pengguna aplikasi TikTok melalui beberapa tahapan yaitu *crawling* data, preprocessing data, dan proses klasifikasi SVM. Berdasarkan hasil penelitian, klasifikasi ulasan mengenai aplikasi TikTok menggunakan algoritma *Support Vector Machine* menunjukkan bahwa dari total 399 data uji, sebanyak 299 data berhasil diklasifikasikan dengan benar sesuai dengan data sebenarnya. Dengan demikian, nilai akurasi dari klasifikasi ulasan mengenai aplikasi TikTok menggunakan algoritma *Support Vector Machine* (SVM) yang diperoleh adalah sebesar 75%.

Daftar Referensi

- [1] G. T. Pramono and C. Hasanudin, "Pemanfaatan Aplikasi Tiktok Sebagai Media Pembelajaran Matematika," *Pros. Semin. Nas. Daring*, vol. 06, no. 27, pp. 979–985, 2023, [Online]. Available: <https://pixabay.com/id/vectors/tiktok-tiktok->
- [2] W. Ikke, S. Lestari, D. Suluh, and K. Dewi, "Penutupan Tiktok Shop di Indonesia Analysis of Public Opinion on the Regulation of the Minister of Trade of the Republic of Indonesia Number 31 of 2023 concerning the Closure of Tiktok Shop in Indonesia," vol. 13, no. 3, pp. 617–627, 2024, doi: 10.31289/perspektif.v13i3.11054.
- [3] T. G. Santoso, "Analisis Sentimen Pada Tweet Dengan Tagar #Bpsrasarentenir Menggunakan Metode Support Vectore Machine (Svm)," (*Doctoral Diss. Univ. Islam Riau*), pp. 12–13, 2021.
- [4] S. Ailiyya, "Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Tokopedia Menggunakan Metode Support Vector Machine," vol. 21, no. 1, pp. 1–9, 2020.
- [5] A. Ma'arif, "Buku Ajar Pemrograman Lanjut Bahasa Pemrograman Python Oleh : Alfian Ma ' Arif," *Univ. Ahmad Dahlan*, p. 62, 2020, [Online]. Available: http://eprints.uad.ac.id/32743/1/buku_python.pdf
- [6] R. Gelar Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 1, pp. 55–60, 2023, doi: 10.47233/jteksis.v5i1.750.
- [7] J. W. Iskandar and Y. Nataliani, "Perbandingan Naïve Bayes, SVM, dan k-NN untuk Analisis Sentimen Gadget Berbasis Aspek," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1120–1126, 2021, doi: 10.29207/resti.v5i6.3588.
- [8] A. Rahman and M. I. Dinata, "Penerapan Algoritma Support Vector Machine dan Particle Swarm Optimization Untuk Pembuatan Rating Aplikasi Google Playstore," 2023, [Online]. Available: <http://journal.ummat.ac.id/index.php/jintens/article/view/13579%0Ahttp://journal.ummat.ac.id/index.php/jintens/article/download/13579/pdf>
- [9] I. W. B. Suryawan, N. W. Utami, and K. Q. Fredlina, "Analisis Sentimen Review Wisatawan pada Objek Wisata Ubud Menggunakan Algoritma Support Vector Machine," *J. Inform. Teknol. dan Sains*, vol. 5, no. 1, pp. 133–140, 2023.
- [10] I. N. Chrisdiyanti *et al.*, "Klasifikasi Review Customer Di E-Commerce Bukalapak Menggunakan Metode Support Vector Machine (SVM)," vol. 10, no. 3, pp. 3200–3206, 2023.
- [11] G. Radiana *et al.*, "Aplikasi Kai Access Menggunakan Metode Support Vector Machine," no. April, pp. 1–10, 2023.
- [12] S. Fide, S. Suparti, and S. Sudarno, "Analisis Sentimen Ulasan Aplikasi Tiktok Di Google Play Menggunakan Metode Support Vector Machine (Svm) Dan Asosiasi," *J. Gaussian*, vol. 10, no. 3, pp. 346–358, 2021, doi: 10.14710/j.gauss.v10i3.32786.
- [13] A. R. Iqbal and Y. Miftahuddin, "Implementasi SVM Untuk Deteksi Komentar Negatif Berbahasa Indonesia di Twitter," *Fti*, vol. X, no. X, 2022, [Online]. Available: <https://eproceeding.itenas.ac.id/index.php/fti/article/view/966%0Ahttps://eproceeding.itenas.ac.id/index.php/fti/article/download/966/942>

- [14] R. A. Nandini, Y. A. Sari, and P. P. Adikara, "Analisis Sentimen Impor Beras 2018 Pada Twitter Menggunakan Metode Support Vector Machine dan Pembobotan Jumlah Retweet," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 4, pp. 3396–3406, 2019.
- [15] A. Handayani and I. Zufria, "Analisis Sentimen Terhadap Bakal Capres RI 2024 di Twitter Menggunakan Algoritma SVM," *J. Inf. Syst. Res.*, vol. 5, no. 1, pp. 53–63, 2023, doi: 10.47065/josh.v5i1.4379.
- [16] N. Meilani, Mhd. Furqan, and Suhardi, "Analisis sentimen pengguna aplikasi BSI mobile akibat ransomware menggunakan algoritma support vector machine," *INFOTECH J. Inform. Teknol.*, vol. 5, no. 1, pp. 42–51, 2024, doi: 10.37373/infotech.v5i1.1102.