

Klasifikasi Sentimen *Review* Pengguna Pada Aplikasi Google Meet Menggunakan PSO Terhadap Optimasi Metode SVM

Hilda Harisa¹, Iin Ernawati², Mayanda Mega Santoni³
Informatika

Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak,
Daerah Khusus Ibukota Jakarta 12450

hildaharisa15@gmail.com¹, megasantoni@upnvj.ac.id², iinernawati@upnvj.ac.id³

Abstrak. Semenjak adanya virus corona diberlakukan kebijakan baru di Indonesia. Kebijakan tersebut menganjurkan pelajar dan pekerja untuk berkegiatan dari rumah secara virtual. Aplikasi *video conference* dapat membantu dalam mengatasi masalah tersebut, salah satunya adalah *Google Meet*. Dalam pelayanan yang diberikan aplikasi tersebut tentu belum sempurna, memiliki kelebihan dan kekurangan dari sudut pandang pengguna. Oleh karena itu, penelitian ini akan melakukan analisis sentimen pada aplikasi *Google Meet* untuk memberikan informasi atau evaluasi mengenai tanggapan pengguna melalui ulasan komentar, dengan mengklasifikasikan opini menjadi opini positif dan negatif menggunakan metode *Support Vector Machine* (SVM) dan *Particle Swarm Optimazation* (PSO) sebagai metode seleksi fitur. Data ulasan yang diperoleh akan dilakukan pelabelan dan pembersihan data sebelum dilakukan pengolahan data, kemudian data diberikan bobot pada setiap kata dengan TF-IDF yang akan dijadikan fitur setelah itu dilakukan seleksi fitur dengan PSO, kemudian dilakukan pembagian data menggunakan *10-fold cross validation* dan diklasifikasikan dengan metode SVM. Hasil rata-rata evaluasi *confusion matrix* dimana *accuracy* sebesar 80%, *precision* sebesar 84%; *recall* sebesar 82% dengan menggunakan metode SVM dan *accuracy* sebesar 82%; *precision* sebesar 92%; *recall* sebesar 80% dengan menggunakan metode PSO.

Kata Kunci: Analisis Sentimen, Klasifikasi, Google Meet, *Support Vector Machine* (SVM), *Particle Swarm Optimazation* (PSO).

1. Pendahuluan

Saat ini video conference memiliki peranan yang cukup penting, semenjak adanya wabah virus *Corona* yang melanda hampir seluruh Negara termasuk Indonesia. Untuk mengurangi penyebaran virus *Corona*, maka diberlakukan kebijakan yang membatasi masyarakat dalam berkegiatan untuk selalu menjaga jarak, mengurangi mobilitas sosial dan mendorong seluruh masyarakat untuk tidak beraktifitas atau berkegiatan di luar rumah [1]. Dalam mendukung kebijakan tersebut diperlukan aplikasi berupa video conference yang dapat membantu dalam bekerja dan belajar dari rumah. Beberapa aplikasi yang dapat mendukung kegiatan tersebut kini telah banyak digunakan, salah satunya yaitu aplikasi Google Meet yang dikembangkan oleh Google.

Google Meet merupakan aplikasi yang cukup populer dengan memiliki rating 3,9 dengan unduhan lebih dari 100 juta kali, dan tercatat sebanyak 1,854,937 ulasan pengguna di kolom komentar pada aplikasi Google Meet di Google Play Store. Mengingat banyaknya ulasan yang diberikan oleh pengguna terhadap aplikasi Google Meet di Google Play Store, hal ini dapat digunakan sebagai peluang untuk menggali informasi untuk mengevaluasi atas kinerja aplikasi Google Meet yang dilakukan dengan menggunakan analisis sentimen.

Beberapa penelitian lain yang telah dilakukan, seperti penelitian oleh Handayani (2019) tentang optimasi algoritma SVM menggunakan PSO untuk analisis sentimen pada ulasan produk Tokopedia menghasilkan nilai akurasi sebesar 83.33% dengan nilai AUC sebesar 0,910 untuk klasifikasi yang hanya menerapkan metode SVM, dan menghasilkan nilai akurasi sebesar 88,89% dengan nilai AUC sebesar 0,946 untuk klasifikasi yang menggunakan optimasi algoritma PSO [2]. Penelitian yang dilakukan oleh Herlinawati (2020) tentang analisis sentimen *Zoom Cloud Meetings* menggunakan metode *Naïve Bayes* dan *Support Vector Machine* menghasilkan nilai akurasi dan AUC pada algoritma NB sebesar 74,37% dan 0,659. Sedangkan pada algoritma SVM menghasilkan nilai akurasi dan AUC sebesar 81,22% dan 0,886. Berdasarkan hasil tersebut diketahui bahwa tingkat akurasi yang menggunakan SVM lebih unggul sekitar 6,85% dibandingkan NB [3].

Berdasarkan dari penelitian yang dijelaskan sebelumnya, hasil penelitian menggunakan metode SVM memiliki nilai akurasi yang baik, dan mampu mengidentifikasi antara 2 kelas, namun memiliki kekurangan pada pengolahan

data berskala besar. Oleh karena itu peneliti mengusulkan penelitian analisis sentimen dengan menggunakan metode klasifikasi *Support Vector Machine (SVM)* dan seleksi fitur *Particle Swarm Optimization (PSO)* untuk meningkatkan hasil akurasi terhadap ulasan dari pengguna aplikasi *Google Meet* yang akan diklasifikasikan ke dalam dua kategori yaitu komentar positif dan negatif.

2. Metode Penelitian

2.1. Analisis Sentimen

Analisis Sentimen merupakan salah satu cabang ilmu data mining yang memiliki tujuan untuk mengekstraksi, mengolah data maupun memahami data yang bersifat tekstual, biasanya berupa opini atau pendapat terhadap entitas, seperti produk, servis, individu, organisasi, dan topik tertentu atau kegiatan lainnya [4].

2.2. Text Mining

Text Mining didefinisikan sebagai proses mengekstraksi pengetahuan implisit dari suatu data tekstual. *Text mining* merupakan suatu proses pengolahan data tekstual yang tidak terstruktur menjadi terstruktur untuk mengidentifikasi suatu informasi yang relevan agar dapat memberikan hasil dengan bersifat kualitatif [5].

2.3. Preprocessing Data

Preprocessing data merupakan salah satu tahapan dalam mempersiapkan teks untuk diolah ke tahapan lebih lanjut. Pada penelitian ini tahap *preprocessing* data yang diterapkan adalah *case folding, data cleaning, normalization, stemming, stopword removal, dan tokenizing*.

2.3.1 Case Folding

Case Folding merupakan tahapan untuk mengubah seluruh teks dokumen dengan mengubah seluruh huruf besar (*uppercase*) menjadi huruf kecil (*lowercase*). Dengan tujuan untuk memudahkan dalam pencarian, karena tidak semua teks dokumen konsisten dengan menggunakan huruf capital [7].

2.3.2 Data Cleaning

Data Cleaning merupakan tahapan penghapusan karakter yang ada dalam teks dokumen seperti *hashtag, url, emoticon, username, dan lainnya* [8]. Tujuan tahapan ini untuk menghilangkan *noise* pada teks dokumen agar memudahkan dalam pengolahan data.

2.3.3 Normalization

Normalization merupakan tahapan mengubah kata yang disingkat atau salah pengejaan. Tahapan ini dilakukan untuk memperkecil dimensi kata yang salah eja atau disingkat namun memiliki arti yang sama, maka akan dianggap sebagai kata yang baru jika kata tersebut tidak diubah [9].

2.3.4 Stemming

Stemming merupakan proses memecah berbagai bentuk (*variants*) dari sebuah kata menjadi kata dasar [10]. Tahapan ini juga merupakan penghapusan kata imbuhan pada kata menjadi kata dasar. Tujuan pada tahapan ini untuk membersihkan kata dari pengejaan yang tidak tepat.

2.3.5 Stopword Removal

Stopwords Removal merupakan tahapan penghilangan kata-kata yang tidak digunakan. Kata yang tidak digunakan berpengaruh tidak baik pada pengolahan teks dokumen seperti *atau, dengan, ke, di, dan tetapi* [7].

2.3.6 Tokenizing

Tokenizing merupakan tahapan dalam pemisahan kalimat menjadi potongan-potongan kata yang biasa disebut token. Tujuan pada tahapan ini adalah untuk memudahkan dalam membedakan karakter [11].

2.4. Pembobotan Kata

Setelah tahapan praproses dilakukan, selanjutnya pemberian bobot pada setiap kata-kata untuk membantu klasifikasi data. Pembobotan kata merupakan proses dalam perhitungan bobot pada setiap kata berdasarkan frekuensi kata. Tujuan dilakukannya pembobotan kata untuk mengetahui seberapa besar suatu kata dibutuhkan untuk mewakili sebuah kalimat, sehingga sebuah perhitungan dan pembobotan kata diperlukan untuk menentukan seberapa banyak suatu kata dibutuhkan [8]. Pada pembobotan ini diperlukan nilai tf , df , dan idf serta hasil perkalian tf dan idf yang merupakan hasil akhir pembobotan.

2.4.1 Term Frequency (TF)

Term Frequency (TF) merupakan banyaknya jumlah kata yang muncul pada sebuah dokumen. Banyaknya jumlah kata yang muncul sebanding dengan bobot yang diberikan.

$$w_{tf} = \begin{cases} 1 + \log_{10}(tf_{t,d}), & \text{if } tf_{t,d} > 0 \\ 0, & \text{if } tf_{t,d} = 0 \end{cases} \quad (1)$$

2.4.2 Invers Document Frequency (IDF)

Inverse Document Frequency (IDF) merupakan proses untuk mengukur pentingnya kata-kata dalam sebuah dokumen. Nilai idf berbanding terbalik dengan kemunculan kata, semakin besar nilai idf menandakan suatu kata jarang muncul dalam sebuah dokumen.

$$idf_t = \log_{10} \left(\frac{N}{df_t} \right) \quad (2)$$

2.4.3 Term Frequency – Inverse Document Frequency (TF-IDF)

Merupakan hasil perkalian pembobotan *term frequency* (w_{tf}) dengan *inverse document frequency* (idf_t) dalam suatu term.

$$w_{t,d} = w_{tf} \times idf_t \quad (3)$$

Dimana $w_{t,d}$ adalah bobot kata di setiap dokumen, $tf_{t,d}$ adalah jumlah kemunculan term dalam dokumen, N adalah jumlah keseluruhan dokumen, dan df_t adalah jumlah dokumen yang mengandung term.

2.5 Support Vector Machine

Support Vector Machine merupakan metode *machine learning* yang berprinsip atas *Structural Risk Minimization* (SRM) yang bertujuan untuk menemukan *hyperplane* terbaik untuk memisahkan dua buah *class* pada input *space* [10]. Cara untuk menemukan *hyperplane* atau pemisahan data dibutuhkan pengukuran margin *hyperplane* dan mencari titik maksimalnya dengan mempertimbangkan titik data menggunakan $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\}$ dimana $y_n = 1/-1$ sebuah konstanta yang menunjukkan kelas yang titik x_n miliki dan $n =$ jumlah sampel, dapat didefinisikan persamaan berikut ini [6]:

$$(w \cdot x) + b = 0 \quad (4)$$

Dalam data x , masuk kedalam kelas *negative* yang diinisiasikan $y_n = -1$, sehingga didefinisikan bidang pembatas pertama pada persamaan berikut:

$$(w \cdot x + b) = -1 \quad (5)$$

Dalam data x , masuk kedalam kelas *positif* yang diinisiasikan $y_n = 1$, sehingga didefinisikan bidang pembatas kedua pada persamaan berikut:

$$(w \cdot x + b) = 1 \quad (6)$$

Diketahui w adalah vector bobot, x adalah vector data (inputan) dan b adalah bias. Jika data dipisahkan secara linear kita dapat memaksimalkan nilai jarak antara titik terdekat dengan jaraknya, dengan $\frac{1}{||w||}$ atau dengan meminimalkan $|w|^2$.

Dalam proses klasifikasi menggunakan *Support Vector Machine* biasanya sering dijumpai kondisi dimana *kernel linier* tidak bekerja secara optimal, sehingga mengakibatkan kinerja klasifikasi yang buruk pada data. Untuk mengatasi hal tersebut, maka perlu menggunakan *kernel non-linear* dengan menggunakan *kernel trick*, dimana dilakukan *mapping* data input ke *feature space* yang dimensinya lebih tinggi, dan masukan yang dihasilkan terpisah secara *linear* untuk membentuk *hyperplane* yang optimal. Berikut ini persamaan dari setiap *kernel Support Vector Machine* [8]:

a. *Kernel Linear*

$$K(x_i, x) = x_i^T x \quad (7)$$

b. *Kernel Polynomial*

$$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0 \quad (8)$$

c. *Kernel RBF*

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (9)$$

d. *Kernel Sigmoid*

$$K(x_i, x) = \tanh(\gamma x_i^T x + r) \quad (10)$$

2.6 Particle Swarm Optimization

Particle Swarm Optimization (PSO) merupakan metode optimasi stokastik berbasis populasi berdasarkan perilaku social dari sekawanan burung atau ikan yang dikembangkan oleh Russell Eberhart dan James Kennedy pada tahun 1995. Menurut Basari, PSO ini adalah teknik optimasi yang sangat sederhana untuk menerapkan dan memodifikasi banyak parameter, metode ini memiliki beberapa teknik untuk melakukan optimasi, seperti meningkatkan bobot atribut ke semua atribut atau variabel yang digunakan, pemilihan atribut dan pemilihan fitur [12].

Particle Swarm Optimization (PSO) dimulai dengan sebuah populasi yang terdiri dari sejumlah partikel acak. Kemudian memperbarui posisi dan kecepatan setiap partikel secara berulang untuk menghasilkan yang baru dan akan berhenti ketika solusi optimal telah ditemukan atau kondisi tertentu telah terpenuhi. Pergerakan partikel memiliki persamaan sebagai berikut [13]:

$$V_i(t) = V_i(t-1) + c_1 r_1 (X_i^l - X_i(t-1)) + c_2 r_2 (X^g - X_i(t-1)) \quad (11)$$

$$X_i(t) = V_i(t) + X_i(t-1) \quad (12)$$

Keterangan:

$X_i^l =$ Local best

$X^g =$ Global best

$c_1 c_2 =$ Konstanta Percepatan

$r_1 r_2 =$ Bilangan random

2.7 Evaluasi

Setelah tahapan klasifikasi, selanjutnya masuk ke dalam tahap evaluasi yang digunakan untuk mengetahui seberapa baik model klasifikasi yang telah dibuat agar mampu mengetahui kemampuan model klasifikasi yang digunakan dapat memprediksi data dengan baik. Untuk menghitung evaluasi model klasifikasi diperlukan metode yang disebut *confusion matrix*. Dengan metode ini diperoleh *accuracy*, *precision*, *recall*, dan *F-Measure* sebagai persentase hasil dari evaluasi model [14]. Pada *Confusion Matrix*, hasil dari pengklasifikasian sebuah data direpresentasikan dalam empat kemungkinan yaitu, *true positif*, *true negatif*, *false positif*, dan *false negatif*.

Confusion Matrix merupakan metode yang digunakan untuk mengukur serta menilai tingkat ketepatan pada sebuah model machine learning dalam memprediksi label dengan membandingkan nilai aktual dan nilai prediksi.

Tabel 1 *Confusion Matrix*

<i>Classification</i>	<i>Predicted Positives</i>	<i>Predicted Negatives</i>
<i>Actual Positive Cases</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>

<i>Actual Negative Cases</i>	<i>True Negative(FP)</i>	<i>True Negative(TN)</i>
------------------------------	--------------------------	--------------------------

Dimana TP adalah data yang diprediksi bernilai benar sebagai positif. TN adalah data yang diprediksi benar sebagai negatif. FP adalah data yang diprediksi bernilai salah sebagai positif, dimana data yang diprediksi bernilai positif namun data sebenarnya bernilai negatif. FN adalah data yang diprediksi bernilai salah sebagai negatif, dimana data yang diprediksi bernilai negatif namun data sebenarnya bernilai positif. Berikut ini rumus untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *F-Measure*[11]:

$$accuracy = \frac{TP+TN}{TP + TN + FP + FN} \quad (13)$$

$$precision = \frac{TP}{TP + FP} \quad (14)$$

$$recall = \frac{TP}{TP + FN} \quad (15)$$

$$f - measure = \frac{2 \times (precision \times recall)}{precision + recall} \quad (16)$$

3. Hasil dan Pembahasan

Dataset yang digunakan pada penelitian ini dilakukan melalui proses *scraping* terhadap aplikasi *Google Meet* pada *Google Play* dengan menggunakan *library Google Play Scraper* sebagai akses dalam pengambilan data. Data penelitian ini diambil menggunakan perangkat lunak *Google Colab* dengan Bahasa pemrograman yang digunakan adalah *python*. Sebanyak 1000 ulasan pengguna berbahasa Indonesia yang telah didapatkan dalam rentang waktubulan September sampai November 2021.

Data ulasan yang telah didapatkan akan diberikan label dengan menjadi dua jenis yaitu positif dan negatif. Ulasan yang diberikan label positif mengandung kesenangan serta kepuasan dalam layanan yang diberikan aplikasi. Dan ulasan yang diberikan label negatif mengandung kekecewaan dan keluhan atas layanan yang diberikan aplikasi. Pada penelitian ini dilakukan pelabelan secara manual dengan anotator sebanyak 3 orang, hasil dari pelabelan ketiga anotator akan disepakati dengan pemungutan suara terbanyak atau mayoritas.

Tabel 2 Ulasan dengan label Positif dan Negatif

Positif	Negatif
Menyenangkan sekali dan mudah ketika sekolah online.dengan lewat aplikasi ini jadi tidak ribet	Update menyusahkan pengguna. Kami mau belajar di sekolah, untuk mengakses google meet harus menunggu penyelenggara bergabung, tolong kembalikan fitur lama, terima kasih.

Berdasarkan kesepakatan yang telah ditetapkan seperti pada Tabel 2, pelabelan dilakukan dengan tiga anotator dimana setiap anotator akan melakukan pelabelan data dengan membaca dan memberikan keputusan (label) pada ulasan. Setelah semua anotator memberikan keputusan (label) akan dilakukan *voting* untuk mendapatkan hasil *final*. Tabel 3 merupakan hasil label dari tiga anotator:

Tabel 3 Ulasan dengan label hasil

Ulasan	Anotator 1	Anotator 2	Anotator 3	Label Hasil
Aplikasi nya bagus banget sangat membantu terimakasih.	Positif	Positif	Positif	Positif

Kecewa karena keluar" sendiri pdahal gak perlu update tapi suruh ngupdate	Negatif	Negatif	Negatif	Negatif
Suka putus sendiri,, suara kurang jelas dan videonya burem padahal kuota full	Negatif	Negatif	Negatif	Negatif

Hasil dari ketiga anotator tersebut akan diukur menggunakan *fleiss kappa*, apabila nilai kappa hanya mencapai < 0,60 maka akan didiskusikan kembali terkait Poleh ketiga anotator, karena memiliki kategori yang berbeda, namun apabila nilai kappa mencapai > 0,60 label yang telah disepakati dapat dikatakan baik dan dapat digunakan kedalam proses selanjutnya. Berikut ini merupakan perhitungan kesepakatan menggunakan *fleiss kappa* dengan rumus yang sudah dijelaskann sebelumnya pada persamaan berikut ini:

$$K = \frac{P_0 - P_e}{1 - P_e} \quad (17)$$

Hasil dari nilai *fleiss kappa* pada penelitian ini sebesar 0,7299. Berdasarkan evaluasi hasil pelabelan, perhitungan kesepakatan menggunakan *fleiss kappa* didapatkan hasil sebesar 0.7299, dimana nilai tersebut masuk kedalam kategori *substantial perfect agreement* yaitu kesepakatan yang digunakan dalam pelabelan data sudah baik dan dapat digunakan ke dalam proses selanjutnya.

Setelah didapatkan data ulasan yang telah disepakati, akan dilakukan praproses data untuk membersihkan data menjadi data yang siap untuk diklasifikasi. Berikut ini merupakan hasil dari tahapan praproses data:

Tabel 4 Praproses Data

Ulasan	Sebelum Praproses	Setelah Praproses
D1	Aplikasi nya bagus banget sangat membantu terimakasih.	['aplikasi', 'bagus', 'bantu', 'terima', 'kasih']
D2	Kecewa karena keluar" sendiri pdahal gak perlu update tapi suruh ngupdate	['kecewa', 'keluar', 'sendiri', 'suruh']
D3	Suka putus sendiri,, suara kurang jelas dan videonya burem padahal kuota full	['suka', 'putus', 'sendiri', 'suara', 'jelas', 'video', 'buram', 'kuota', 'penuh']

Data bersih yang telah melalui prose sebelumnya. akan dilakukan pembobotan kata untuk mengetahui seberapa besar diperlukannya suatu kata untuk mewakili sebuah kalimat, diperlukan perhitungan dan pembobotan kata untuk mengetahui tingkat dibutuhkan suatu kata. Pembobotan kata akan dilakukan dengan menghitung TF (*Term Frequency*) menggunakan persamaan rumus (1), IDF (*Invers Document Frecuency*) menggunakan persamaan rumus (2), dan TF-IDF menggunakan persamaan rumus (3). Pada Tabel 5 berikut ini merupakan perhitungan TF-IDF:

Tabel 5 TF-IDF

Token	TF			DF	IDF	TF-IDF		
	D1	D2	D3			D1	D2	D3
aplikasi	1	0	0	1	0.48	0.48	0.00	0.00
bagus	1	0	0	1	0.48	0.48	0.00	0.00
bantu	1	0	0	1	0.48	0.48	0.00	0.00
terima	1	0	0	1	0.48	0.48	0.00	0.00
kasih	1	0	0	1	0.48	0.48	0.00	0.00
kecewa	0	1	0	1	0.48	0.00	0.48	0.00
keluar	0	1	0	1	0.48	0.00	0.48	0.00
sendiri	0	1	1	2	0.18	0.00	0.18	0.18

Token	TF			DF	IDF	TF-IDF		
	D1	D2	D3			D1	D2	D3
suruh	0	1	0	1	0.48	0.00	0.48	0.00
suka	0	0	1	1	0.48	0.00	0.00	0.48
putus	0	0	1	1	0.48	0.00	0.00	0.48
suara	0	0	1	1	0.48	0.00	0.00	0.48
jelas	0	0	1	1	0.48	0.00	0.00	0.48
video	0	0	1	1	0.48	0.00	0.00	0.48
buram	0	0	1	1	0.48	0.00	0.00	0.48
kuota	0	0	1	1	0.48	0.00	0.00	0.48
penuh	0	0	1	1	0.48	0.00	0.00	0.48

Data yang telah dibobotkan akan dilakukan pembagian data dengan menggunakan *10-fold cross validation*. Metode ini akan membagi data latih dan data uji dengan ratio yang jumlahnya hamper sama dari 1000 ulasan, dimana data 1000 dokumen yang akan dibagi menjadi 10 kelompok, setiap kelompok akan terdiri dari 100 dokumen. Pada iterasi pertama kelompok satu menjadi data uji sebanyak 100 dokumen, dan kelompok lainnya menjadi data latih dengan jumlah yang sama yaitu sebanyak 100 dokumen, sehingga data latih berjumlah 900 dokumen pada setiap iterasinya. Setiap iterasi memiliki data uji yang berbeda-beda, sehingga setiap kelompok dapat dijadikan sebagai data uji dan data latih.

Selanjutnya data yang telah dibagi akan dilakukan proses pelatihan pada data latih untuk mendapatkan sebuah model klasifikasi. Model tersebut akan diuji untuk mengetahui sejauh mana model tersebut dapat mengklasifikasikan data uji. Klasifikasi pada penelitian ini menggunakan metode SVM dengan parameter *default* dengan dilakukan percobaan pada beberapa *kernel*, yaitu *kernel Linear*, *Polynomial*, *Radial Basis Function (RBF)*, dan *Sigmoid* dan metode SVM yang terintegrasi dengan menggunakan metode PSO dengan parameter *default*, dengan dilakukan beberapa percobaan terhadap iterasi secara konstan dengan selisih 50 dimana akan dimulai dari iterasi 50 sampai 1000.

Setelah dilakukan pemodelan, maka akan dilakukan menggunakan *confussion matrix* dengan tujuan untuk mengetahui seberapa baik model klasifikasi yang telah dibuat agar mampu mengetahui kemampuan model klasifikasi yang digunakan dapat memprediksi data dengan baik. Berikut adalah Tabel 6 yang merupakan hasil evaluasi dari setiap skenario:

Tabel 6 Hasil Evaluasi

Skenario	Akurasi	Pecision	Recall	F1-Score
Pertama (SVM)	80%	84%	82%	83%
Kedua (SVM+PSO) 50 Iterasi	80%	91%	79%	84%
Ketiga (SVM+PSO) 100 Iterasi	80%	91%	78%	84%
Keempat (SVM+PSO) 150 Iterasi	81%	92%	79%	85%
Kelima (SVM+PSO) 200 Iterasi	81%	92%	78%	85%
Keenam (SVM+PSO) 250 Iterasi	78%	92%	76%	83%
Ketujuh (SVM+PSO) 300 Iterasi	80%	91%	79%	84%
Kedelapan (SVM+PSO) 350 Iterasi	82%	92%	80%	85%
Kesembilan (SVM+PSO) 400 Iterasi	82%	93%	79%	85%

Kesepuluh (SVM+PSO) 450 Iterasi	81%	92%	78%	85%
Kesebelas (SVM+PSO) 500 Iterasi	81%	91%	79%	85%
Kedua belas (SVM+PSO) 550 Iterasi	81%	93%	79%	85%
Ketiga belas (SVM+PSO) 600 Iterasi	80%	91%	79%	84%
Keempat belas (SVM+PSO) 650 Iterasi	81%	92%	79%	85%
Kelima belas (SVM+PSO) 700 Iterasi	81%	92%	79%	85%
Keenam belas (SVM+PSO) 750 Iterasi	81%	92%	79%	85%
Ketujuh belas (SVM+PSO) 800 Iterasi	79%	91%	77%	84%
Kedelapan belas (SVM+PSO) 850 Iterasi	80%	91%	78%	84%
Kesembilan belas (SVM+PSO) 900 Iterasi	81%	93%	78%	85%
Kedua puluh (SVM+PSO) 950 Iterasi	81%	92%	79%	85%
Kedua puluh satu (SVM+PSO) 1000 Iterasi	80%	91%	78%	84%

Tabel 6 menunjukkan hasil evaluasi klasifikasi dengan tanpa menggunakan metode *Particle Swarm Optimization* dan menggunakan *Particle Swarm Optimization*, diperoleh hasil dari skenario pertama yaitu mendapatkan nilai akurasi sebesar 80%, precision sebesar 84%, recall sebesar 82% dan f1-score sebesar 83% dengan menggunakan metode *Support Vector Machine*, dan mendapatkan nilai akurasi terbaik pada skenario kedelapan sebesar 82%, precision sebesar 92%, recall sebesar 80%, dan f1-score sebesar 85%.

Berdasarkan dari hasil yang didapatkan dari setiap skenario, bahwa kenaikan setiap iterasi cukup memberikan pengaruh pada performa yang dihasilkan, dimana membuktikan bahwa dengan mengoptimasi menggunakan PSO mampu menaikkan nilai akurasi 2% dari hasil skenario pertama. Namun, pada saat proses pengujian memiliki perbandingan waktu yang cukup jauh, dimana pada skenario pertama membutuhkan waktu selama 2 menit dan pada skenario ketiga ketika iterasi PSO 400 kali membutuhkan waktu selama 24 menit.

Penelitian ini tidak hanya menampilkan hasil evaluasi, namun menampilkan hasil visualisasi yang merupakan tahapan yang dilakukan untuk menggambarkan hasil dari penelitian yang dilakukan, dimana hasil tersebut berupa informasi yang dianggap penting dan sering dibicarakan pada ulasan aplikasi *Google Meet*.



Gambar 1 Sentimen Positif

Pada Gambar 1 didapatkan berupa informasi dalam Bahasa Indonesia pada aplikasi *Google Meet* yang mewakili ulasan pengguna, dimana kata yang sering muncul pada ulasan yaitu “bagus”, “google”, “meet”, “latar belakang”,

Referensi

- [1] Dewayani, T., 2020. Bekerja dari Rumah (Work From Home) Dari Sudut Pandang Unit Kepatuhan Internal. [Online]
Available at: <https://www.djkn.kemenkeu.go.id/artikel/baca/13014/Bekerjadari-Rumah-Work-From-Home-Dari-Sudut-Pandang-UnitKepatuhan-Internal.html>
[Diakses 5 Oktober 2021].
- [2] Handayani, R. N., Mubarak, A. & Susanti, S., 2019. Optimasi Algoritma Support Vector Machine Menggunakan Particle Swarm Optimization Untuk Analisis Sentimen Pada Ulasan Produk Tokopedia. JURNAL INFORMATIKA, Volume 6, pp. 1-8.
- [3] Herlinawati, N. et al., 2020. ANALISIS SENTIMEN ZOOM CLOUD MEETINGS DI PLAY STORE MENGGUNAKAN NAÏVE BAYES DAN SUPPORT VECTOR MACHINE. CESS (Journal of Computer Engineering System and Science), Volume 5, pp. 293-298.
- [4] Liu, B., 2012. Sentiment Analysis and Opinion Mining. s.l.:Morgan & Claypool Publishers.
- [5] Jo, T., 2019. Text Mining Concepts, Implementation, and Big Data Challenge. Cham: Springer.
- [6] Hulu, A. M. F. & Lhaksana, K. M., 2019. Analisis Sentimen Politik pada Twitter Menggunakan Metode Support Vector Machine (Studi Kasus : Pilpres 2019). e-Proceeding of Engineering, Volume 6, pp. 9726-9735.
- [7] Cahyono, Y., 2017. ANALISIS SENTIMENT PADA SOSIAL MEDIA TWITTER MENGGUNAKAN NAÏVE BAYES CLASSIFIER DENGAN FEATURE SELECTION PARTICLE SWARM OPTIMIZATION DAN TERM FREQUENCY. JURNAL INFORMATIKA UNIVERSITAS PAMULANG, Volume 2, pp. 14-19.
- [8] Husada, H. C. & Paramita, A. S., 2021. Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM). TEKNIKA, Volume 10, pp. 18-26.
- [9] Praptiwi, D. Y., 2018. ANALISIS SENTIMEN ONLINE REVIEW PENGGUNA E-COMMERCE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE DAN MAXIMUM ENTROPY. PROGRAM STUDI STATISTIKA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS ISLAM INDONESIA.
- [10] Rezki, M. et al., 2020. Analisis Review Pengguna Google Meet dan Zoom Cloud Meeting Menggunakan Algoritma Naïve Bayes. Jurnal Infortech, Volume 2, pp. 264-270.
- [11] Simanjuntak, S. T., 2021. ANALISIS SENTIMEN PADA LAYANAN GOJEK INDONESIA MENGGUNAKAN XTREME GRADIENT BOOSTING. PROGRAM STUDI S1 TEKNOLOGI INFORMASI FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA.
- [12] Wardhani, N. K. et al., 2018. SENTIMENT ANALYSIS ARTICLE NEWS COORDINATOR MINISTER OF MARITIME AFFAIRS USING ALGORITHM NAIVE BAYES AND SUPPORT VECTOR MACHINE WITH PARTICLE SWARM OPTIMIZATION. Journal of Theoretical and Applied Information Technology, Volume 96, pp. 8365-8378.
- [13] Kour, V. P. & Arora, S., 2019. Particle Swarm Optimization Based Support Vector Machine (P-SVM) for the Segmentation and Classification of Plants. IEEE Access, Volume 7, pp. 29374-29385.
- [14] Simanjuntak, S. T., 2021. ANALISIS SENTIMEN PADA LAYANAN GOJEK INDONESIA MENGGUNAKAN XTREME GRADIENT BOOSTING. PROGRAM STUDI S1 TEKNOLOGI INFORMASI FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA.