

## Analisis Sentimen Ulasan Pengguna Pada Aplikasi *Google Classroom* Menggunakan Metode SVM Dan Seleksi Fitur PSO

Ghaisa Amany Mursianto<sup>1</sup>, Didit Widiyanto<sup>2</sup>, Bambang Tri Wahyono<sup>3</sup>

<sup>1,2,3</sup> Informatika, Fakultas Ilmu Komputer

<sup>1,2,3</sup> Universitas Pembangunan Nasional Veteran Jakarta

<sup>1,2,3</sup> Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, DKI Jakarta 12450  
ghaisaam@upnvj.ac.id<sup>1</sup>, didit.widiyanto@upnvj.ac.id<sup>2</sup>, bambang.triwahyono@upnvj.ac.id<sup>3</sup>

**Abstrak.** Kebijakan PSBB mengharuskan penerapan kegiatan pembelajaran jarak jauh secara *online* menggunakan aplikasi berbasis *daring* seperti *Google Classroom*. Dengan menggunakan *Google Classroom* memfasilitasi penggunaannya untuk mendistribusikan, mengumpulkan, dan memberikan penilaian terhadap tugas-tugas yang diberikan kepada mahasiswa maupun siswa di seluruh negeri. Penelitian ini bertujuan untuk mengetahui sentimen opini publik terhadap aplikasi *Google Classroom*. Dalam melakukan analisis sentimen penelitian ini menggunakan metode *Support Vector Machine* serta *Particle Swarm Optimization* sebagai seleksi fitur. Data yang sudah didapatkan akan dilakukan pelabelan data, dilanjut dengan *pre-processing* data, lalu data yg sudah melewati proses itu akan diberikan bobot kata dengan TF-IDF, kemudian akan masuk seleksi fitur dengan PSO, kemudian dilakukan pembagian data sebesar 80:20, lalu pengklasifikasian dengan metode SVM. Hasil rata-rata evaluasi menggunakan *confusion matrix* dimana *accuracy* sebesar 79%, *precision* sebesar 78%, *recall* sebesar 67% dengan menggunakan metode SVM dan dengan menggunakan seleksi fitur PSO mendapatkan hasil *accuracy* sebesar 83%, *precision* sebesar 86%, *recall* sebesar 67%.

**Kata Kunci:** Analisis Sentimen, Klasifikasi, *Google Classroom*, *pre-processing*, TF-IDF, *Support Vector Machine* (SVM), *Particle Swarm Optimazation* (PSO).

### 1. Pendahuluan

Saat ini sistem pendidikan menerapkan pembelajaran secara daring, semenjak adanya Virus Covid-19 pemerintah mengeluarkan kebijakan *Sosial Distancing* atau yang lebih umum disebut sebagai Pembatasan Sosial Berskala Besar (PSBB) untuk mencegah penyebaran virus. Oleh karena itu sistem pendidikan saat ini menerapkan kegiatan pembelajaran tatap muka secara *online* yang dinamakan *Learning Management System* (LMS) atau yang lebih biasa disebut adalah *Study From Home* (SFH) untuk para pelajar, dalam kegiatan pembelajaran secara daring menggunakan aplikasi pembelajaran daring, untuk aplikasi yang digunakan dalam daring sendiri sangat beragam salah satunya aplikasi pendukung tersebut adalah *Google Classroom* yang sudah banyak digunakan dalam berbagai macam kegiatan pembelajaran daring, *Google Classroom* sendiri berada dibawah naungan *Google*, dan dikembangkan oleh *Google*. Aplikasi *Google Classroom* digunakan untuk membuat sebuah ruang belajar daring, *google classroom* dirancang juga untuk menjadi sarana mendistribusikan sebuah tugas, mengumpulkan atau melakukan submit, untuk menilai tugas-tugas yang sudah dikumpulkan, serta untuk mempermudah interaksi antara pelajar, dan guru dalam melakukan pembelajaran daring.

Berdasarkan pada penelitian sebelumnya menggunakan algoritma *Support Vector Machine* (SVM) dengan menggunakan *cross validation* menghasilkan nilai rerata akurasi sebesar 83,33% untuk Ruang Guru, 82,67% untuk Zenius, 82,00% untuk Kelas Pintar, 74,67% untuk Edmodo, dan 64,33% untuk *Google Classroom*. (Erfina, et al., 2020) [1]. Kemudian pada penelitian mengenai optimasi algoritma SVM menggunakan *particle swarm optimization* menghasilkan rata-rata nilai akurasi sebesar 83,33% untuk klasifikasi yang hanya menerapkan metode SVM, dan menghasilkan rata-rata nilai akurasi sebesar 88,89% untuk klasifikasi yang menggunakan optimasi algoritma PSO (Handayani, et al., 2019) [2].

Berdasarkan dari penelitian yang sudah dijelaskan sebelumnya, hasil penelitian dengan menggunakan metode SVM memiliki nilai akurasi yang baik, dan mampu dalam mengidentifikasi antara 2 kelas, namun memiliki kekurangan pada pemilihan parameter dan pengolahan data yang berskala besar. Oleh sebab 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 pengguna aplikasi *Google Classroom*, diklasifikasikan dalam dua kategori komentar yaitu komentar positif dan komentar negatif.

## 2. Metode Penelitian

### 2.1. Analisis Sentimen

Suatu metode analisa yang digunakan dalam *Data Mining, Natural Language Processing (NLP)*, dan *Machine Learning* yang menitikberatkan pada ekstraksi data sentiment isi kalimat [3].

### 2.2. Text Mining

*Text analytic* proses pendeskripsian suatu teks data baik yang semi-terstruktur maupun yang tidak terstruktur, alasan tersebut lah yang menjadi pembeda antara data mining dengan *text mining* [4].

### 2.3. Text Preprocessing

Merupakan sekumpulan data yang akan dipergunakan untuk melakukan analisis sentimen, yang nantinya data yang sudah diperoleh akan diproses melalui beberapa tahapan, untuk diambil informasi yang terkandung dalam data tersebut, biasanya hasil kandungan yang ada dalam data yang tersebut berupa hal sentimen seperti kalimat positif atau negatif, untuk memudahkan dalam pemrosesan data, analisis dilakukan dengan manual dengan membaca kalimat yang akan diproses, lalu diberikan pelabelan yang nantinya akan menentukan kalimat tersebut merupakan sentimen negatif atau positif, *Text Preprocessing* digunakan untuk membersihkan *noise* atau ciri kata-kata yang tidak digunakan pada proses selanjutnya [5]. Pada tahapan ini berguna untuk melakukan pembersihan data sehingga pada saat proses pengklasifikasian data yang digunakan sudah bersih dari *noise* yang ada, dan untuk proses tersebut pada penelitian ini ada 6 tahap *preprocessing* data yang diterapkan adalah *case folding, data cleaning, normalization, stemming, stopword removal, dan tokenizing*.

- a. **Case Folding** adalah salah satu tahapan yang digunakan dalam merubah semua huruf yang berada dalam satu dokumen menjadi sebuah huruf kecil yang pada awalnya menggunakan huruf besar. Huruf yang diubah juga dimulai dari “a”, dan sampai huruf “z” [6].
- b. **Data Cleaning** merupakan suatu proses pembersihan data yang digunakan dalam mengurus suatu data dalam dokumen yang tidak lengkap, *data cleaning* sendiri digunakan dalam memeriksa penduplikasian sebuah data, memeriksa data yang tidak konsisten, serta memperbaiki kesalahan dalam sebuah data. *Data Cleaning* sendiri sangat penting untuk membuat data menjadi terstruktur, relevan, dan mencegah data supaya tidak terjadi korup [7].
- c. **Normalization** atau normalisasi merupakan suatu tahapan proses dalam melakukan mendeteksi suatu kesalahan dalam kata seperti kata yang duplikat dalam artian memiliki suatu makna yang sama atau disingkat, kata-kata yang salah tersebut akan dilakukan suatu perbaikan kata sehingga menghasilkan data yang lebih terstruktur dan sederhana [8].
- d. **Stemming** adalah suatu tahapan dalam preprocessing untuk menemukan kata dasar dengan menghilangkan imbuhan yang ada dalam kata tersebut. Tujuan dalam proses ini menghilangkan sufiks, konfiks, infiks, dan prefiks yang ada dalam kata [9].
- e. **Stopword Removal** merupakan suatu proses menghilangkan kata yang tidak memiliki nilai atau tidak memiliki makna [10].
- f. **Tokenizing** atau *tokenization* merupakan suatu tahapan untuk memecah suatu kumpulan teks menjadi sebuah kata. Tokenizing digunakan untuk menghapus tanda baca seperti koma (,), titik (.), spasi, serta angka yang ada dalam teks tersebut [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. Pembobotan kata merupakan suatu proses yang bertujuan dalam pembobotan kata yang nantinya akan memberikan suatu bobot pada fitur berdasarkan frekuensi kemunculan data semakin banyak dokumen yang diproses semakin banyak juga fitur yang muncul yang disebut dengan TF-IDF (*Term Frequency - Inversed Document Frequency*) [12]. 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* atau  $tf$  adalah banyaknya kemunculan kata pada sebuah dokumen. Sedangkan  $w_{tf}$  merupakan jumlah bobot yang berasal dari  $tf$  yang sudah dihitung.

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

### 2.4.2 Invers Document Frequency (IDF)

*Document Frequency* atau  $df$  adalah banyaknya frekuensi atau jumlah dalam dokumen yang mengandung suatu kata. Sedangkan *Inverse Document Frequency* atau  $idf$  merupakan bobot kebalikan dari merupakan kebalikan dari  $df$  yang artinya jika semakin jarang sebuah kata muncul dalam dokumen semakin besar bobot *Inverse Document Frequency*.

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

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

*Term Frequency Inverse Document Frequency* atau  $tf-idf$  merupakan hasil perkalian dari  $tf$  dan  $idf$  dari suatu kata atau *term*.

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

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

Keterangan:

- $N$  = Jumlah keseluruhan teks dokumen
- $w_{tf_t}$  = Bobot kata
- $tf_{t,d}$  = Jumlah kata atau *term* yang muncul dalam dokumen
- $df_t$  = Jumlah dokumen mengandung suatu kata atau *term*
- $idf_t$  = Bobot inverse
- $w_{t,d}$  = Pembobotan TF-IDF

## 2.5. Klasifikasi Data

Setelah dilakukan pembobotan data, maka proses selanjutnya melakukan klasifikasi sentimen dengan dua metode, yaitu metode klasifikasi SVM tanpa PSO dan metode klasifikasi SVM dengan PSO.

### 2.5.1 Support Vector Machine

*Support Vector Machine* (SVM) adalah algoritma klasifikasi dan regresi yang sudah banyak digunakan dalam pemecahan masalah linear, dan nonlinear. Cara pemisahannya sendiri menerapkan pemisahan linear, yang dimana nanti input data non linear berdimensi tinggi diperoleh melalui fungsi *kernel* [13]. SVM merupakan metode pembelajaran dalam *machine learning* (*supervised learning*) yang memiliki fungsi untuk melakukan suatu prediksi kelas berdasarkan hasil dari *data train*. Dengan menggunakan pelatihan data numerik, hasil proses ekstraksi fitur dapat membentuk pola yang akan digunakan dalam proses pelabelan [14] yang dinamakan *hyperplane* yang berfungsi memisahkan dua kelas. Seperti pada gambar berikut [15]. Untuk mendapatkan *hyperplane* atau pemisahan data tersebut dibutuhkan perhitungan *margin hyperplane* dan menemukan titik max. Dalam pencarian tersebut digunakan persamaan, seperti yang berikut ini:

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

Dalam data  $x_i$ , masuk kedalam kelas *negative* yang diinisialisasikan -1 yang didefinisikan bidang pembatas pertama dalam persamaan berikut:

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

Dalam data  $x_i$ , masuk kedalam kelas positif yang diinisialisasikan 1 yang didefinisikan bidang pembatas kedua dalam persamaan berikut:

$$(w \cdot x + b) = 1 \tag{6}$$

Pada support vector machine, jika data dipisahkan secara linear nilai margin antara bidang pembatas dimaksimalkan dengan  $\frac{2}{|w|}$  atau dengan meminimalkan  $|w|^2$ . Maka pencarian hyperplane terbaik dengan memaksimalkan kedua bidang pembatas pada (5) dan (6) dapat dipresentasikan dalam bentuk persamaan sebagai berikut:

$$y_i(x_i w + b) \geq 1 \tag{7}$$

Pada persamaan berikut merupakan hasil prediksi untuk membuat *hyperplane* sebagai fungsi keputusan melakukan proses klasifikasi dari dua buah kelas [16]:

$$f(\phi(x)) = \text{sign}(w\phi(x) + b) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \phi(x_i) T_{\phi(x)} + b\right) \tag{8}$$

Pada *hyperplane* fungsi tunggal dibuat sesuai dengan nilai  $w$  dan  $b$  yang diperoleh, setelah itu  $x_i$  merupakan sub-himpunan data pembelajaran akan mencari nilai matrik  $K$  berdasarkan persamaan dibawah ini:

$$K(x_i, x) = \phi(x_i) \tag{9}$$

Untuk mencari tahu nilai dari  $w$  dan  $b$  berkaitan dengan margin, dapat menggunakan persamaan berikut ini [17]:

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-) \tag{10}$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i \tag{11}$$

Keterangan:

- $w$  = bobot vektor
- $w \cdot x^+$  = nilai *weight* untuk kelas positif
- $w \cdot x^-$  = nilai *weight* untuk kelas negatif
- $\alpha_i$  = nilai bobot data
- $x_i$  = nilai atribut
- $y_i$  = kelas data
- $b$  = scalar yang digunakan sebagai nilai bias/konstanta

Dalam pemrosesan menggunakan SVM, *kernel linear* tidak berfungsi secara maksimal sehingga pengklasifikasian data menjadi kurang baik. Oleh sebab itu jika terjadi permasalahan tersebut dapat menggunakan *kernel trick* yang dalam prosesnya menghasilkan input yang terpisah secara linier, sehingga membentuk *hyperplane* yang optimal. *Kernel* sendiri memiliki beberapa persamaan antara lain, *Kernel linear*, *polynomial*, *RBF*, dan *sigmoid* [18]:

a. *Kernel Linear*

$$K(x_i, x) = x_i^T x \tag{12}$$

b. *Kernel Polynomial*

$$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0 \tag{13}$$

c. *Kernel RBF*

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \tag{14}$$

d. Kernel Sigmoid

$$K(x_i, x) = \tanh(\gamma x_i^T x + r) \quad (15)$$

2.5.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) menggunakan kecepatan dan model pencarian posisi, di dalam sejumlah partikel, posisi masing-masing partikel digunakan untuk mewakili kandidat solusi dari ruang dan kecepatan setiap partikel untuk memperbarui posisi partikel [19]. Particle Swarm Optimization (PSO) juga merupakan salah satu *feature selection* untuk memecahkan permasalahan dalam optimalisasi, seperti dalam peningkatan bobot sebuah fitur terhadap semua atribut atau variabel yang digunakan. Algoritmanya sangat sederhana dalam menerapkan dan melakukan pemodifikasian beberapa parameter [20].

Populasi dalam PSO terdiri dari sejumlah partikel acak yang kemudian nantinya akan memperbarui posisi dan kecepatan masing-masing partikel dengan terbang secara iteratif untuk menghasilkan partikel yang baru. Proses pada PSO akan berhenti ketika solusi optimal telah ditemukan atau kondisi tertentu telah tercapai. Pergerakan partikel memiliki persamaan dengan persamaan sebagai berikut [21]:

$$V_i(t) = V_i(t - 1) + c_1 r_1 (X_i^l - X_i(t - 1)) + c_2 r_2 (X^G - X_i(t - 1)) \quad (16)$$

$$X_i(t) = V_i(t) + X_i(t - 1) \quad (17)$$

Keterangan:

- Vs(t) : Kecepatan partikel s saat iterasi t
- Xs(t) : Posisi partikel s saat iterasi t
- Ps = solusi terbaik dari partikel
- W = berat inerti
- G = solusi terbaik dari partikel yang ditemukan
- c<sub>1</sub> dan c<sub>2</sub> = konstanta percepatan
- r<sub>1</sub> dan r<sub>2</sub> = fungsi acak dalam kisaran [0,1]

2.6 Evaluasi

Dalam sebuah model klasifikasi yang sudah dibuat dalam melakukan prediksi suatu data baru. Dalam melakukan suatu prediksi data, tidak semua data yang dimiliki dapat dilakukan klasifikasi dengan akurat, oleh sebab itu dalam suatu pemodelan diperlukanlah sebuah evaluasi pemodelan. Evaluasi model klasifikasi merupakan suatu evaluasi kinerja dari suatu model klasifikasi. Dalam menghitung evaluasi digunakan sebuah metode, metode itu disebut dengan *confusion matrix*. Metode ini bertujuan untuk melakukan evaluasi yang bertujuan untuk mengukur tingkat akurasi dari hasil klasifikasi [22].

Tabel 1. Confusion Matrix

		True Class	
		Positive	Negative
Predicted Class	Positive	TP : True Positive	FP : False Positive
	Negative	FN : False Negative	TN : True Negative

Pada tabel *confusion matrix* tersebut memiliki penjelasan bahwa TP (*True Positif*) adalah *record* data positif suatu kelas yang memiliki hasil positif saat dilakukan prediksi maupun pada kenyataannya. TN (*True Negatif*) adalah suatu *record* data negatif kelas yang memiliki hasil negatif saat diprediksi, dan kenyataannya negatif. FP (*False Positif*) adalah suatu *record* data suatu kelas yang diprediksi memiliki hasil positif, tetapi kenyataannya negatif. FN (*False Negatif*) adalah suatu *record* data suatu kelas yang diprediksi memiliki hasil negatif, tetapi kenyataannya positif. Berikut ini persamaan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *F-Measure*:

$$accuracy = \frac{TP+TN}{TP + TN + FP + FN} \quad (18)$$

$$precision = \frac{TP}{TP + FP} \quad (19)$$

$$recall = \frac{TP}{TP + FN} \quad (20)$$

$$f - measure = \frac{2 \times (precision \times recall)}{precision + recall} \quad (21)$$

### 3. Hasil dan Pembahasan

Pengumpulan data pada penelitian ini dilakukan dengan menggunakan cara teknik *scraping* dengan modul *google-play-scraper*. Aplikasi *google play store* telah menyediakan API yang memudahkan akses dalam mengambil data *google play store*. Pengambilan datanya dilakukan menggunakan perangkat lunak yaitu *google colab* dengan menggunakan bahasa pemrograman *python* dan teknik *scraping*. Pengumpulan data ini memiliki dilakukan dalam rentang waktu November 2021 sampai Desember 2021 dengan menggunakan kata kunci “relevan”. Data yang sudah dikumpulkan tersebut akan dilakukan seleksi ulasan sehingga didapatkan sebanyak 950 data. Data ulasan yang telah didapatkan akan diberikan label akan dikategorikan menjadi dua kategori yaitu komentar positif dan komentar negatif. Sebelum dilakukannya pelabelan data. Pelabelan data *google classroom* dikerjakan oleh tiga orang anotorator/penilai secara manual berdasarkan dengan sudut pandang dari penilai. Setelah semua anotorator memberikan keputusan (label) akan dilakukan *voting* untuk mendapatkan hasil *final*. Tabel 2 merupakan hasil label dari tiga anotorator:

**Tabel 2.** Ulasan dengan label hasil

Ulasan	Anatorator 1	Anatorator 2	Anatorator 3	Hasil
Tolong hapus aplikasi ini karena tugas saya banyak gara gara aplikasi ini	Negatif	Negatif	Negatif	Negatif
Semakin lama semakin lemot aplikasinya.. Muter2 terus.. Payah.	Negatif	Negatif	Negatif	Negatif
Sangat membanru dalam dalam persiapan kegiatan daring. Terima kasih	Positif	Positif	Positif	Positif

Hasil dari pelabelan data oleh anotorator tersebut disebut dengan kesepakatan dari tiga anotorator, dengan hasil kesepakatan tersebut penting untuk mengetahui apakah kesepakatan tersebut konsisten atau tidak. Kesepakatan yang sudah dilakukan akan diukur dengan *fleiss kappa*. Jika nilai *kappa* hanya mencapai <0,60 maka label data yang memiliki kategori yang berbeda akan dilakukan diskusi Kembali oleh tiga anotorator, namun jika nilai *kappa* sudah mencapai >0,60 maka kesepakatan yang telah dibuat sudah dapat dikatakan baik dan memiliki konsisten. Berikut ini merupakan perhitungan kesepakatan menggunakan *fleiss kappa* dengan rumus yang sudah dijelaskann sebelumnya pada persamaan berikut ini:

$$K = \frac{P_0 - P_e}{1 - P_e} \quad (22)$$

Hasil dari nilai *fleiss kappa* pada penelitian ini sebesar 0.72094, 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 *pre-processing* data untuk membersihkan data menjadi data yang siap untuk diklasifikasi. Berikut ini merupakan hasil dari tahapan *pre-processing* data:

**Tabel 3.** *Pre-processing* Data

Sebelum Praproses	Setelah Praproses
hapus aplikasi tugas aplikasi	['hapus', 'aplikasi', 'tugas', 'aplikasi']
lambat aplikasi mutar payah	['lambat', 'aplikasi', 'mutar', 'payah']
bantu giat daring terima kasih	['bantu', 'giat', 'daring', 'terima', 'kasih']

Data bersih yang telah melalui prose sebelumnya. akan dilakukan pembobotan kata untuk mengetahui seberapa besar diperlukannya suatu kata untuk mewakili sebuah kalimat. Pembobotan kata akan dilakukan dengan menghitung TF (*Term Frequency*) menggunakan persamaan (1), IDF (*Invers Document Frequency*)

menggunakan persamaan (2), dan TF-IDF menggunakan persamaan (3). Perhitungan bobot tiap kata menggunakan TF-IDF ditunjukkan dalam Tabel 4.

Tabel 4. TF-IDF

Token	TF			DF	IDF	TF-IDF		
	D1	D2	D3			D1	D2	D3
aplikasi	2	1	0	2	aplikasi	2	1	0
daring	0	0	1	1	daring	0	0	1
giat	0	0	1	1	giat	0	0	1
hapus	1	0	0	1	hapus	1	0	0
kasih	0	0	1	1	kasih	0	0	1
lambat	0	1	0	1	lambat	0	1	0
bantu	0	0	1	1	bantu	0	0	1
mutar	0	1	0	1	mutar	0	1	0
payah	0	1	0	1	payah	0	1	0
terima	0	0	1	1	terima	0	0	1
tugas	1	0	0	1	tugas	1	0	0

Data yang telah dibobotkan akan dilakukan pembagian data menjadi 80:20 yaitu 80% data *training* dan 20% data *testing*. Selanjutnya data yang telah dibagi akan dilakukan proses pelatihan untuk mendapatkan sebuah model klasifikasi. Model tersebut akan diuji untuk mengetahui sejauh mana model dapat mengklasifikasikan data uji. Percobaan ini menggunakan Metode Klasifikasi SVM dengan beberapa *kernel*, yaitu *kernel Linear*, *Polynomial*, *Radial Basis Function (RBF)*, dan *Sigmoid*. Metode Klasifikasi SVM ini menggunakan seleksi fitur PSO dengan parameter *default*, dengan dilakukan percobaan terhadap beberapa iterasi dimulai dari iterasi 50, 100, 250, 500, dan terakhir 950.

Setelah dilakukan proses pemodelan, maka akan dilakukan evaluasi menggunakan *confussion matrix* dengan tujuan mendapatkan model klasifikasi yang lebih akurat yang memiliki kemampuan untuk memprediksi data dengan baik seperti ditunjukkan pada Tabel 5 merupakan hasil evaluasi dari setiap skenario:

Tabel 5. Hasil Evaluasi

Skenario	Akurasi	Precision	Recall	F1-Score
Pertama (SVM)	79%	78%	67%	72%
Kedua (SVM+PSO, 50 Iterasi)	82%	85%	66%	74%
Ketiga (SVM+PSO, 100 Iterasi)	79%	79%	64%	71%
Keempat (SVM+PSO, 250 Iterasi)	81%	85%	62%	72%
Skenario	Akurasi	Precision	Recall	F1-Score
Kelima (SVM+PSO, 500 Iterasi)	81%	90%	58%	70%
Keenam (SVM+PSO, 950 Iterasi)	83%	86%	67%	76%

Tabel 5 menunjukkan hasil evaluasi klasifikasi dengan dua skenario, yaitu menggunakan metode *Support Vector Machine (SVM)* tanpa metode *Particle Swarm Optimization (PSO)* dan skenario kedua menggunakan gabungan dua metode SVM dan PSO. Skenario pertama diperoleh hasil nilai akurasi sebesar 79%, precision sebesar 78%, recall sebesar 67% dan f1-score sebesar 72% dengan menggunakan metode SVM dan PSO, mendapatkan nilai akurasi terbaik pada skenario keenam sebesar 83%, precision sebesar 86%, recall sebesar 67%, dan f1-score sebesar 76%.

Berdasarkan dari hasil yang didapatkan dari setiap skenario, kenaikan dari setiap iterasi memberikan pengaruh pada performa yang dihasilkan, seperti pada skenario keenam dengan iterasi 950 kali mampu menaikkan nilai akurasi 4% dari hasil skenario pertama yang sebelumnya 79% menjadi 83%.

Penelitian ini juga menampilkan hasil visualisasi dalam bentuk *wordcloud* yang digunakan untuk menggambarkan hasil penelitian, dimana hasil tersebut berupa informasi yang dianggap penting dan sering dibicarakan pada ulasan aplikasi *Google Classroom*.



Gambar. 1. Sentimen Positif

Pada Gambar 1 Dari *wordcloud* tersebut dapat dilihat bahwa kata yang sering terlihat pada sentimen positif mengenai *google classroom* adalah “bantu”, “bagus”, ”tugas”, ”aplikasi”, “notifikasi”, “ajar”, “kiriman” dan lain-lain. Dari kata-kata yang muncul pada sentimen positif ini, dapat diambil kesimpulan bahwa aplikasi *google classroom* dapat mempermudah penggunaannya dalam mengumpulkan tugas, dan melakukan pengajaran tanpa perlu keluar dari rumah, apalagi di masa pandemi sekarang ini kita diharuskan untuk menjaga jarak serta meminimalisir berkontak langsung dengan orang lain.



Gambar. 2. Sentimen Negatif

Pada Gambar 2 *wordcloud* tersebut dapat dilihat bahwa kata yang sering terlihat pada sentimen negatif mengenai *google classroom* adalah ”tugas”, “aplikasi”, “kiriman”, “eror”, “notifikasi”, “buka”, ”kelas”, “video” dan lain-lain. Dari rangkaian kata yang muncul pada sentimen negatif, dapat disimpulkan bahwa pengguna aplikasi *google classroom* memnerikan kritikterhadap *error* dalam beberapa fiturunya, susah dalam mengumpulkan tugas, dan sering terlambat muncul notifikasi.

## 4. Kesimpulan dan Saran

### 4.1 Kesimpulan

Klasifikasi ulasan *google classroom* dengan SVM dan seleksi fitur PSO, dilakukan dengan pengumpulan data ulasan dari pengguna *Google Play Store* pada aplikasi *Google Classroom* pada tanggal 20 November 2021 sampai dengan 25 Desember 2021 yang berjumlah 950 data. Data yang dikumpulkan tersebut akan dilakukan pelabelan manual oleh 3 orang anator. Setelah dilakukan pelabelan oleh ketiga anator tersebut, data akan dilanjutkan ke proses pembersihan data atau *pre-processing* dalam proses ini akan dilakukan 6 proses yaitu, *case folding*, *data cleaning*, *normalization*, *stopword removal*, *stemming*, dan *tokenizing*, kemudian data yang telah melalui proses tersebut, setiap kata akan diberikan bobot menggunakan proses *Term Frequency-Invers Document Frequency* (TF-IDF) yang nantinya akan menjadi fitur, lalu fitur-fitur tersebut akan dilakukan seleksi lagi menggunakan *Particle Swarm Optimization*, kemudian fitur yang telah melewati tahap seleksi membagi data menjadi data latih dan data uji, kemudian dibuat menjadi model menggunakan metode SVM, kemudian hasil dari pemodelan akan dilakukan evaluasi dengan menggunakan *confussion matrix*, hasil dari evaluasi akan berupa nilai akurasi, presisi, *recall*, dan *f1-score* yang akan menunjukkan hasil keberhasilan dalam pembuatan model. Dari hasil pemodelan klasifikasi menggunakan metode SVM dan metode PSO untuk melakukan seleksi fitur, klasifikasi ulasan yang berkaitan dengan *Google Classroom* berjalan dengan baik dan mendapatkan nilai akurasi sebesar 79% dengan metode SVM tanpa PSO, setelah digunakannya PSO hasil akurasi meningkat menjadi 83%. Dengan begitu PSO dapat menjadi seleksi fitur yang baik untuk metode SVM

dalam klasifikasi sentimen dengan dibuktikan pada penelitian ini yang mampu meningkatkan nilai akurasi sebesar 4%. Menunjukkan bahwa PSO mampu dalam meningkatkan hasil akurasi.

#### 4.2 Saran

Dari hasil penelitian ini, dapat dikembangkan menjadi beberapa topik penelitian yang lain, yaitu:

- a. melakukan pengklasifikasikan ulasan lebih dari 950 ulasan.
- b. melakukan pre-processing yang lebih baik lagi, sehingga hasil performa klasifikasi dapat meningkat.
- c. label yang digunakan tidak hanya positif dan negatif saja, tetapi menambahkan label netral.
- d. menggunakan metode seleksi fitur lainnya untuk membandingkan hasil performa yang didapatkan.

#### Referensi

- [1] A. Erfina, E. S. Basryah, A. Saepulrohman, dan D. Lestari, "Analisis Sentimen Aplikasi Pembelajaran Online Di Play Store Pada Masa Pandemi Covid-19 Menggunakan Algoritma Support Vector Machine (SVM)," dalam *Seminar Nasional Informatika (SEMNASIF)*, vol. 1, no. 1, 2020, hal. 145-152.
- [2] R. N Handayani, A. Mubarak, dan S. Susanti, "Optimasi Algoritma Support Vector Machine Menggunakan Particle Swarm Optimization Untuk Analisis Sentimen Pada Ulasan Produk Tokopedia," *JURNAL INFORMATIKA*, vol. 6, hal. 1-8, 2019.
- [3] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, dan M. N. Al-Kabi, "A comprehensive survey of arabic sentiment analysis," *Information Processing & Management*, vol. 56, no. 2, hal. 320–342, 2019.
- [4] N. Khotimah, "Analisis Sentimen Terhadap Review E-Commerce Dengan Metode Stochastic Gradient Descent," Muhammadiyah University, Semarang, 2019.
- [5] M. Rivki, dan A. M. Bachtiar, "Implementasi algoritma K-Nearest Neighbor dalam pengklasifikasian follower twitter yang menggunakan Bahasa Indonesia," *Jurnal Sistem Informatika*, vol. 13, no. 1, hal. 31–37, 2017.
- [6] M. Z. Naf'an, A. Burhanuddin, dan A. Riyani, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," *Jurnal Linguistik Komputasional*, vol. 2, no. 1, hal. 23–27, 2019.
- [7] N. P. A. Widiari, I. M. A. D. Suarjaya, dan D. P. Githa, "Teknik Data Cleaning Menggunakan Snowflake untuk Studi Kasus Objek Pariwisata di Bali," *JURNAL ILMIAH MERPAT*, vol. 8, no. 2, Agustus 2020.
- [8] Z. Efendy, "Normalisasi dalam desain database," *Jurnal coreIT*, vol. 4, no. 1, hal. 34–43, Juni 2018.
- [9] J. A. Septian, T. M. Fachrudin, dan A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *Journal of Intelligent System and Computation*, vol. 1, no. 1, hal. 43–49, 2019.
- [10] N. K. Fitriyani, dan A. D. Hartanto, "Analisis Sentimen Terhadap Tokoh Publik Menggunakan Support Vector Machine," *MEANS (Media Informasi Analisa Dan Sistem)*, vol. 5, no. 1, hal. 8–12, 2020.
- [11] N. C. Siregar, R. R. A. Siregar, dan M. Y. D. Sudirman, "Implementasi Metode Naive Bayes Classifier (NBC) Pada Komentar Warga Sekolah Mengenai Pelaksanaan Pembelajaran Jarak Jauh (PJJ)," *JURNAL TEKNOLOGIA*, vol. 3, no. 1, 2020.
- [12] A. M. Pravina, I. Cholissodin, dan P. P. Adikara, "Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 3, no. 3, hal. 2789-2797, 2019.
- [13] K. Setiawan, B. Rahmatullah, B. Burhanuddin, A. B. Paryanti, dan F. Fauzi, "Komparasi Metode Naive Bayes Dan Support Vector Machine Menggunakan Particle Swarm Optimization Untuk Analisis Sentimen Mobil Esemka," *Journal of Information System, Applied, Management, Accounting and Research*, vol. 4, no. 3, hal. 102–111, 2020.
- [14] H. C. Husada, dan A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, hal. 18–26, 2021.
- [15] S. P. H. S. MZ, dan M. Adami, "Algoritma SVM pada Data Mining Tingkat Pemahaman Mata Kuliah (Studi Kasus pada Mata kuliah Rekamata Perangkat Lunak). dalam *Seminar Nasional Informatika (SEMNASIF)*, vol. 1, no. 1, hal. 291-300, Desember 2020.
- [16] R. Noviana, dan D. Wahyuni, "Metode Naïve Bayes Classifier Dalam Analisis Sentimen Pada Komentar Tweet," 2020
- [17] I. M. Parapat, "Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak," Doctoral dissertation, Universitas Brawijaya 2018.
- [18] H. C. Husada, dan A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, hal. 18–26, 2021.
- [19] Y. Ding, W. Zhang, L. Yu, dan K. Lu, "The accuracy and efficiency of GA and PSO optimization schemes on estimating reaction kinetic parameters of biomass pyrolysis," *Energy*, vol. 176, hal. 582–588, 2019.
- [20] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, dan W. Gata, "Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi," *Jurnal Teknoinfo*, vol. 14, no. 2, hal. 115–123, 2020.
- [21] V. P. Kour, dan S. Arora, "Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants," *IEEE Access*, vol. 7, hal. 29374–29385, 2019.
- [22] B. Santoso, dan A. I. S. Azis, "Machine Learning & Reasoning Fuzzy Logic Algoritma, Manual, Matlab, & Rapid Miner," *Deepublish*, 2020.