

Pengaruh Seleksi Fitur Particle Swarm Optimization terhadap Sentimen Analisis Aplikasi PeduliLindungi di Twitter dengan Algoritma Support Vector Machine

Irza Ramira Putra¹, Yuni Widiastiwi², Nurul Chamidah³

^{1,2,3} Informatika, Fakultas Ilmu Komputer

^{1,2,3} Universitas Pembangunan Nasional Veteran Jakarta

Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota Jakarta 12450

irzar@upnvj.ac.id¹, widiastiwi@yahoo.com², nurul.chamidah@upnvj.ac.id³

Abstrak. PeduliLindungi merupakan aplikasi yang ditujukan kepada masyarakat guna menangkal serta menangani COVID-19 di Indonesia. Aplikasi ini wajib dimiliki oleh masyarakat Indonesia, sebagai salah satu hal untuk kewajiban yang dibuat oleh pemerintah dari perundangan yang dibuat untuk memasuki fasilitas publik. Tentunya aplikasi ini juga mendatangkan beberapa tanggapan dari masyarakat. Tanggapan tersebut bisa diungkapkan melalui media sosial yang cukup populer seperti twitter. Melalui twitter, mereka bebas mengungkapkan pendapat mereka tentang penggunaan aplikasi tersebut. Penelitian ini bermaksud untuk mendapatkan informasi sentimen terkait opini masyarakat yang berhubungan dengan penggunaan aplikasi PeduliLindungi, dengan mengaplikasikan algoritma *Support Vector Machine* serta kernel *Radial Basis Function* dan algoritma seleksi fitur yaitu *Particle Swarm Optimization* dalam mengklasifikasikan opini masyarakat terhadap aplikasi PeduliLindungi dari data tweet yang sudah diperoleh serta diberi label sentimen bersifat positif dan label sentimen bersifat negatif. Model *Support Vector Machine* menghasilkan akurasi sebesar 76.24%, *recall (sensitivity)* sebesar 82.14%, presisi sebesar 76.67%, dan *specificity* sebesar 68.89%, sedangkan model *Support Vector Machine* ditambahkan seleksi fitur yaitu *Particle Swarm Optimization* mengalami peningkatan akurasi sebesar 88.12%, *recall (sensitivity)* sebesar 96.43%, presisi sebesar 84.36%, dan *specificity* sebesar 77.78%.

Kata Kunci: Analisis Sentimen, PeduliLindungi, Twitter, *Support Vector Machine*, *Particle Swarm Optimization*

1 Pendahuluan

COVID-19 atau *Coronavirus Disease* adalah salah satu penyakit yang sangat mematikan yang menyerang dunia setelah *Influenza* dan SARS. Penelitian tentang COVID-19 juga menemukan bahwa penyakit ini merupakan penyakit yang sangat mengancam lebih khususnya kepada masyarakat lansia. Ditemukan juga bahwa sebuah penelitian di *England* dan *Wales* menemukan bahwa orang yang terjangkit kasus COVID-19 adalah sekitar 11% untuk orang dibawah 45 tahun dan 65% untuk umur 65-74 tahun, 82% untuk yang berumur 75 tahun dan seterusnya. Virus COVID-19 yang berawal penyebarannya di Wuhan, China, dan negara tetangga Indonesia seperti Singapura dan Malaysia yang sudah mulai terlihat kasus penyebaran COVID-19, sementara Indonesia masih dinyatakan zero case yang berarti Indonesia tidak ada kasus COVID-19. Tetapi pada tanggal 2 Maret 2020, tepat di saat itu juga Jokowi menyatakan bahwa ada dua kasus yang terkonfirmasi positif COVID-19, sejak kejadian ini pemerintah memobilisasikan tes massal dan juga restriksi mobilisasi juga diadakan [1].

Dengan adanya restriksi ini yang ada di Indonesia, maka masyarakat Indonesia harus mengubah kegiatan seperti pekerjaan yang di kantor atau kegiatan belajar dan mengajar di sekolah menjadi bekerja di rumah atau *work from home* dan belajar secara daring atau *online learning* [2]. Sementara pemerintah juga meluncurkan sebuah aplikasi yang dibuat oleh kolaborasi antara Kementerian KOMINFO, Kementerian Kesehatan, Kementerian BUMN dan Badan Nasional Penanggulangan Bencana (BPNNB), dimana aplikasi ini berfungsi untuk mencegah dan menanggulangi pandemi COVID-19, khususnya di Indonesia [3]. Aplikasi PeduliLindungi ini menggunakan sistem jaringan konektivitas yang menggunakan *Bluetooth* dan GPS didalam gawai masyarakat. Aplikasi ini juga membantu menerapkan peraturan perundangan dari pemerintah untuk scan *Barcode* atau QR code yang dipasang pada pintu masuk sebelum memasuki fasilitas publik seperti Mall. Tujuannya tentu untuk menjamin bahwa masyarakat yang berhak memasuki fasilitas publik tersebut tidak terjangkit COVID-19 dan sudah divaksinasi. Aplikasi ini juga memang masih memiliki beberapa isu yang bisa dibilang menjadi kekhawatiran masyarakat, contohnya seperti penjaminan akan keamanan terhadap peretasan, selain itu juga jaminan pemerintah jika memang diretas, maka pihak mana yang akan bertanggung jawab [4]. Aplikasi ini tentunya digunakan oleh banyak pengguna, terutama masyarakat Indonesia, dan feedback yang

didapat dari masyarakat tentunya juga banyak, baik melalui aplikasinya sendiri atau di Playstore, bahkan di media sosial.

Berbicara tentang media sosial tentunya juga cukup banyak, seperti Instagram, Facebook, TikTok, dan Twitter. Semua media sosial tersebut merupakan media sosial yang sangat banyak digunakan oleh masyarakat dunia, terutama di Indonesia, salah satunya Twitter yang merupakan media sosial dengan penggunanya sangat banyak dan informasi melalui Twitter dengan mengirim sebuah pesan atau “*tweet*”. Pengguna Twitter di Indonesia juga merupakan salah satu yang terbanyak, di tahun 2012 terbukti bahwa Jakarta, mengeluarkan tweet terbanyak dari kota-kota yang ada di dunia. Twitter, tidak hanya masyarakat biasa tetapi juga politikus, selebriti, bahkan sampai organisasi pun juga memakainya [5]. Bisa disimpulkan juga bahwa Twitter merupakan salah satu media sosial yang penggunanya cukup banyak, bahkan di seluruh dunia. Flow of information atau arus informasi dari media ini juga cukup banyak dan merupakan informasi public yang bisa dilihat banyak pengguna. Selain sebagai tempat untuk berinteraksi untuk satu sama lain pengguna, Twitter juga merupakan tempat untuk bebas berekspresi baik itu ekspresi positif maupun negatif [6]. Kandungan informasi yang terdapat di dalamnya juga penting untuk diketahui apa yang diberikan dari informasi tersebut. Karena kekhawatiran isu yang sudah diberikan melalui feedback melalui sosial media twitter, maka perlu dilakukan penelitian analisis sentimen bertujuan untuk memberikan informasi mengenai opini publik masyarakat Indonesia terhadap penggunaan aplikasi ini.

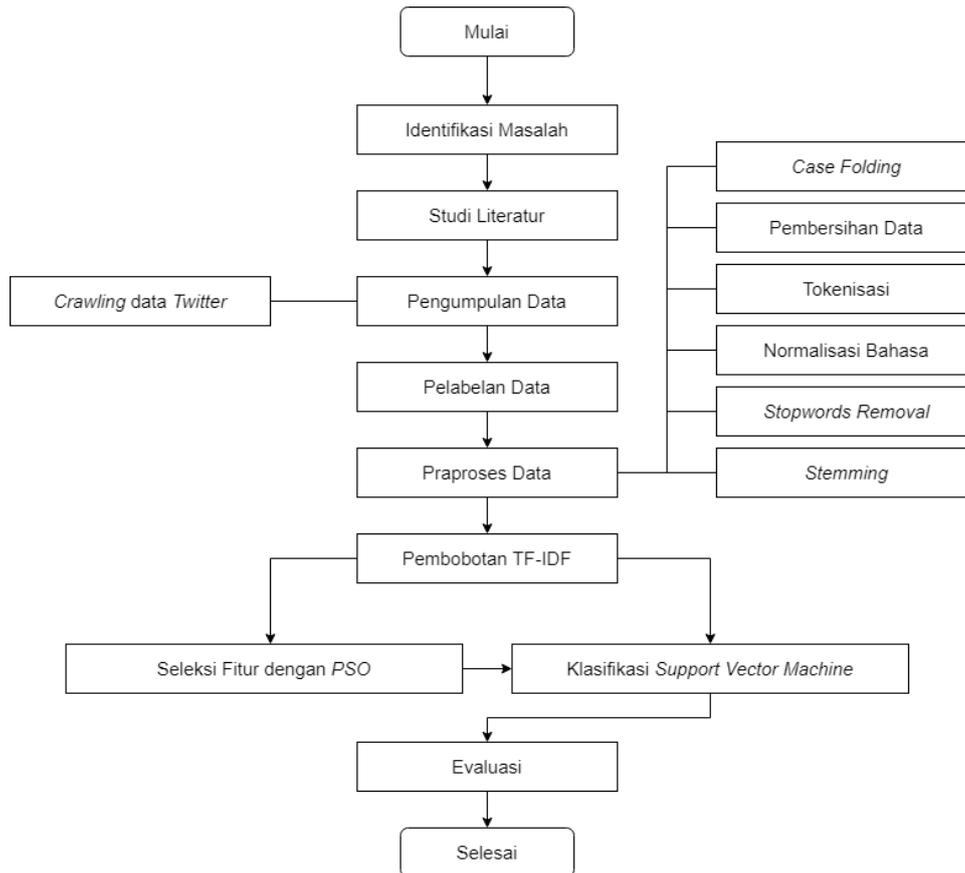
Metode klasifikasi Support Vector Machine merupakan salah satu algoritma yang digunakan untuk mendeteksi sentimen pada tweet, dan penggunaan algoritma Particle Swarm Optimization sebagai seleksi fitur juga meningkatkan performa model klasifikasi. Penelitian dengan topik judul analisis sentimen terhadap layanan Indihome berdasarkan twitter dengan metode klasifikasi Support Vector Machine (SVM) mengenai performa dari algoritma Support Vector Machine yang menghasilkan nilai akurasi sebesar 87%, precision sebesar 86%, recall sebesar 95%, error-rate sebesar 13%, dan f1-score sebesar 90% menyimpulkan bahwa analisis sentimen data Twitter berbahasa Indonesia bisa menggunakan metode SVM [7]. Lalu penelitian yang berjudul Analisis Sentimen Opini Publik Terhadap Penerapan Kebijakan Social Distancing Dalam Pencegahan COVID-19 lain juga membuktikan bahwa penggunaan algoritma Particle Swarm Optimization sebagai seleksi fitur juga dapat meningkatkan performa dari algoritma SVM, dengan hasil akurasi model sebesar SVM+PSO 98,25% dan nilai AUC sebesar 0.744 yang lebih baik daripada hasil akurasi model hanya SVM sebesar 67% dan nilai AUC sebesar 0.999 [8].

Berdasarkan hal yang sudah dibahas, penelitian ini akan menggunakan metode klasifikasi dengan algoritma Support Vector Machine (SVM) dengan seleksi fitur menggunakan algoritma Particle Swarm Optimization (PSO) untuk analisis sentimen terhadap penggunaan aplikasi PeduliLindungi berdasarkan opini masyarakat di sosial media Twitter dengan harapan bisa memberikan informasi sentimen opini masyarakat untuk meningkatkan sistem aplikasi PeduliLindungi, serta juga melihat penggunaan seleksi fitur PSO ini dapat meningkatkan performa algoritma SVM dengan membandingkan performa algoritma SVM ini sebelum dan sesudah menggunakan PSO.

2 Metodologi Penelitian

2.1 Tahap Penelitian

Dalam melakukan penelitian, ada beberapa tahapan penelitian yang dilakukan sebagai berikut:



Gambar. 1. Tahap Penelitian

2.1 Identifikasi Masalah

Tahapan penelitian ini merupakan tahapan untuk mencari dan menemukan suatu masalah untuk memperjelas yang akan diteliti. Permasalahan pada penelitian ini yaitu, mengklasifikasikan data tweet yang mengandung opini publik terhadap aplikasi PeduliLindungi dengan metode klasifikasi *Support Vector Machine* dan pengaruh penggunaan Algoritma *Particle Swarm Optimization* sebagai seleksi fitur pada performa metode klasifikasi.

2.2 Studi Literatur

Tahapan ini melakukan pencarian teori terkait penelitian dari berbagai sumber untuk dijadikan referensi. Sumber yang digunakan berasal dari buku, jurnal, literatur lainnya untuk mencari teori mengenai analisis sentimen, text-mining, algoritma *Support Vector Machine*, *Particle Swarm Optimization*, dan hal-hal lain yang dibahas dalam penelitian ini.

2.3 Pengumpulan Data

Pada tahapan penelitian ini, data tweet dikumpulkan dengan cara crawling menggunakan Application Programming Interface (API) yang disediakan oleh Twitter dan library *rtweet* di RStudio. Untuk pengumpulan data, digunakan kata kunci Peduli Lindungi, hashtag #PeduliLindungi, dan yang menyebutkan username @PeduliLindungi sebanyak 501 tweet. Data hasil crawling disimpan dengan format XLSX (Microsoft Excel Spreadsheet Files)

2.4 Pelabelan Data

Pelabelan data dikerjakan secara manual oleh 3 orang penilai ke dalam 2 kategori, label positif dan negatif. Berikut pada Tabel 1 adalah contoh hasil pelabelan secara manual oleh 3 orang penilai:

Tabel 1. Contoh Pelabelan Data *Tweet*

Data <i>Tweet</i>	Penilai 1	Penilai 2	Penilai 3	Label Akhir
Aplikasi Peduli Lindungi tidak melanggar HAM, justru sangat bermanfaat.	Positif	Positif	Positif	Positif
<p> jelas ya ini, #PeduliLindungi tidak seperti yang dituduhkan melanggar HAM https://t.co/N3B4rvToNX </p>	Positif	Positif	Negatif	Positif
@contactap2 Halo Min, saya sudah booster tapi peduli lindungi saya keterangannya cuma sampai vaksin ke dua, bisa gak saya pakai kartu vaksin aja biar gak usah antigen?	Positif	Negatif	Negatif	Negatif

Dari hasil penilaian label tersebut masih terdapat perbedaan pendapat antara masing-masing penilai dalam mengkategorikan label tweet, maka diperlukan hasil yang menunjukkan persetujuan antar penilai dengan perhitungan *Kappa Value*. Berikut hasil perhitungan *kappa value* untuk data tweet yang sudah diberi nilai oleh penilai:

Rumus Persamaan (1) dengan *Kappa Value*

$$Kappa = \frac{P_0 - P_e}{1 - P_e} \tag{1}$$

Keterangan:

Kappa : Koefisien dari nilai kesepakatan dimana 0 untuk persetujuan secara kebetulan tidak satu penilaian, dan 1 untuk persetujuan yang mutlak.

P₀ : Proporsi frekuensi penilai yang penilaian sama

P_e : Peluang kesepakatan antar penilai yang penilaiannya berbeda.

Dimana persamaan (2) dengan P_e:

$$P_e = P(positif)^2 + P(negatif)^2 \tag{2}$$

Hasil kappa value yang digunakan dapat dikatakan objektif untuk menilai sebuah kesepakatan [9]. Hasil kesepakatan dari 3 orang penilai dengan pengukuran kappa value dapat dikategorikan pada tabel 2 :

Tabel 2. Nilai Kesepakatan *Kappa Value*

Kesepakatan	Nilai k
Rendah (<i>poor</i>)	k < 0.00
Kurang (<i>deficient</i>)	0.00 - 0.20
Lumayan (<i>fair</i>)	0.21 - 0.40
Cukup (<i>moderate</i>)	0.41 - 0.60
Baik (<i>good</i>)	0.61 - 0.80
Sangat Baik (<i>Very Good</i>)	k > 0.81

2.5 Praproses Data

Sebelum data tweet diklasifikasi, perlu dilakukan pra proses terlebih dahulu karena data belum berstruktur dan memiliki banyak noise. Tahapan dari pra proses ini terdiri dari case folding, pembersihan data, tokenisasi, normalisasi bahasa, stopword removal dan stemming.

2.5.1 Case Folding

Tahap pertama pada pra proses data yaitu *Case Folding*, dimana seluruh data tweet yang didapat dari yang memiliki huruf kapital (*uppercase*) dikonversi menjadi huruf non-kapital (*lowercase*), bertujuan untuk mencegah terjadinya case sensitive.

2.5.2 Pembersihan Data

Sebelum dilakukan tokenisasi, perlu dilakukan pembersihan pada data tweet yang tidak memiliki nilai dengan menghapus *username*, *tag @*, *hashtag #*, *link URL*, dan juga *emoji* serta juga menghapus beberapa karakter tanda baca atau whitespace untuk mengurangi noise menggunakan *regular expression*.

2.5.3 Tokenisasi

Setelah pembersihan data, data dilakukan tokenisasi, yaitu metode pemenggalan dokumen menjadi potongan-potongan kata yang disebut token [10] memanfaatkan *library nltk*.

2.5.4 Normalisasi Bahasa

Setelah token-token terbuat, data tweet yang diperoleh perlu dilakukan normalisasi, dimana kata tidak baku diubah menjadi kata baku, dan kata yang sesuai dengan kamus besar Bahasa Indonesia (KBBI), menggunakan Kamus Alay – *Colloquial Indonesian Lexicon*, dan beberapa kata yang dimasukkan secara manual berdasarkan KBBI.

2.5.5 Stopwords Removal

Setelah data sudah di melewati tahap normalisasi bahasa, selanjutnya adalah penghilangan stopword, yaitu penghapusan kata yang berulang kali muncul sehingga tidak penting dan tidak berpengaruh pada performa proses klasifikasi memanfaatkan *library nltk* dan ditambahkan secara manual dari <https://stopwords.net/indonesian-id/>.

2.5.6 Stemming

Tahap terakhir pada pra proses data, data-data tweet akan dilakukan stemming sebagai proses untuk menghilangkan imbuhan pada kata-kata yang dikembalikan ke kata dasar [11].

2.6 Pembobotan Term (TF-IDF)

Setelah pra proses data dilakukan, maka perhitungan bobot term dengan mengubah kata-kata pada data tweet menjadi sebuah angka sehingga kata/term dapat dikenali sebagai fitur untuk klasifikasi nanti. Metode perhitungan Term Frequency (TF) dan Inverse Document Frequency (IDF) menggunakan rumus:

$$W_{t,d} = tf_{t,d} \times idf_t = tf_{t,d} \times \log \log \frac{N}{df_t} \quad (3)$$

Keterangan:

- $W_{t,d}$: Bobot TF-IDF
- $tf_{t,d}$: Jumlah frekuensi kata
- idf_t : Jumlah inverse frekuensi dokumen tiap kata
- df_t : Jumlah frekuensi dokumen tiap kata
- N : Jumlah total dokumen

2.7 Seleksi Fitur dengan Particle Swarm Optimization

Tahap selanjutnya adalah seleksi fitur menggunakan Particle Swarm Optimization, dengan pertama menentukan posisi awal partikel dan kecepatan awal secara random, lalu menentukan Local Best dan Global Best. Local Best pertama, yaitu posisi awal partikel. Untuk menentukan local best selanjutnya adalah dengan membandingkan nilai fitness partikel sekarang dengan local best. Jika nilai fitness pada partikel sekarang lebih kecil dari nilai fitness di local best, maka local best akan diperbarui dengan nilai posisi sekarang. Setelah mendapat local best, ditentukan partikel global best dengan membandingkan nilai fitness dari setiap local best yang memiliki nilai fitness terkecil. Lalu update kecepatan dan posisi menggunakan rumus:

$$v_j = v_j(i - 1) + c_1 r_1 (P_{best,j} - X_j(i - 1)) + c_2 r_2 (G_{best} - X_j(i - 1)) \quad (4)$$

$$x_j(i) = v_j(i) + x_j(i - 1) \quad (5)$$

Keterangan:

- v_j : Kecepatan partikel
- x_j : Posisi partikel
- j : 1, 2, ..., N mempresentasikan jumlah partikel
- $P_{best,j}$: posisi terbaik dari partikel ke j
- G_{best} : posisi terbaik global
- $c1$ dan $c2$: learning factor
- $r1$ dan $r2$: bilangan acak antara 0 sampai 1

Lalu algoritma berhenti jika nilai fitness terbaik sudah tercapai atau iterasi maksimal tercapai.

2.8 Klasifikasi dengan Support Vector Machine

Klasifikasi menggunakan Support Vector Machine akan dilakukan dengan kernel linear dan radial basis function (RBF). Data akan dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%. Proses klasifikasi menggunakan algoritma Support Vector Machine dihitung dengan:

Rumus Persamaan (6) dan (7) Support Vector Machine

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

atau

$$f(x) = \sum_{i=1}^m a_i y_i K(x, x_i) + b \quad (7)$$

Keterangan :

- w : Nilai parameter untuk mencari hyperplane pada garis tegak lurus antara garis hyperplane dengan titik data vector.
- x : Titik data yang dimasukkan ke Support Vector Machine.
- a_i : Bobot nilai dari setiap titik data.
- $K(x, x_i)$: Fungsi pada kernel.
- b : Parameter hyperplane untuk mencari nilai bias.

Rumus Persamaan (8) Kernel Radial Basis Function

$$K(x_t, x_{tn}) = \exp\left(-\frac{\|x_t - x_{tn}\|^2}{2\sigma^2}\right) \quad (8)$$

Keterangan:

- $K(x, y)$: Nilai kernel dari data
- \exp : Nilai eksponensial
- x : Fitur data untuk training
- c : Nilai konstanta
- σ : Standar deviasi kernel radial basis function

2.9 Evaluasi

ada tahap akhir penelitian, model klasifikasi akan melewati tahap evaluasi menggunakan metode confusion matrix yang sudah dijelaskan pada tabel untuk menganalisis hasil performanya. Berikut rumus yang digunakan untuk menghitung evaluasi pada penelitian ini:

Persamaan rumus (9) untuk menghitung nilai akurasi:

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (10)$$

Persamaan rumus (10) untuk menghitung nilai *recall* (*sensitivity*):

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (11)$$

Persamaan rumus (11) untuk menghitung nilai presisi:

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (12)$$

Aplikasi Peduli	aplikasi peduli	aplikasi peduli	['beberapa',	'melanggar',		
Lindungi	lindungi	lindungi	'hari',	'ham']	['aplikasi',	['aplikasi',
membantu	membantu	membantu	'kemudian',	'peduli',	'peduli',	'peduli',
pemerintah	pemerintah	pemerintah	'ada', 'laporan',	'lindungi',	'lindungi',	'lindung',
dalam	dalam	dalam	'media',	'membantu',	'membantu',	'bantu',
pengendalian	pengendalian	pengendalian	'internasional',	'pemerintah',	'pemerintah',	'perintah',
penyebaran	penyebaran	penyebaran	'mengenai',	'dalam',	'pengendalia	'kendali',
covid-19.	covid-19.	covid	'app', 'peduli',	'pengendalian',	n',	'sebar',
			'lindungi',	'penyebaran',	'penyebaran',	'covid']
#VaksinCapai	#vaksincapai		'yang',	'covid']	'covid']	
HerdImmunity	herdImmunity		'berbahaya',			
https://t.co/	https://t.co/		'dan',			
qh6J2bnFuF	qh6j2bnfuf		'melanggar',			
			'ham']			

Kemudian data diatas dilakukan perhitungan pembobotan kata dengan variabel kata atau term yang diperoleh dari hasil praproses data sebanyak 809 kata (term) dari jumlah data tweet sebanyak 501 tweet menggunakan metode perkalian Term Frequency - Inverse Document Frequency (TF-IDF) seperti pada Tabel 4:

Tabel 4. Perhitungan Pembobotan Kata (TF-IDF)

Term	Dokumen		DF	IDF	TF-IDF	
	D1	D2			D1	D2
lapor	1	0	1	0,301	0,301	0,000
media	1	0	1	0,301	0,301	0,000
internasional	1	0	1	0,301	0,301	0,000
aplikasi	1	1	2	0,000	0,000	0,000
peduli	1	1	2	0,000	0,000	0,000
lindung	1	1	2	0,000	0,000	0,000
bahaya	1	0	1	0,301	0,301	0,000
langgar	1	0	1	0,301	0,301	0,000
ham	1	0	1	0,301	0,301	0,000
bantu	0	1	1	0,301	0,000	0,301
perintah	0	1	1	0,301	0,000	0,301
kendali	0	1	1	0,301	0,000	0,301
sebar	0	1	1	0,301	0,000	0,301
covid	0	1	1	0,301	0,000	0,301

Data tweet yang sudah diberi bobot dengan TF-IDF dibagi menjadi data latih (training) yang diambil 80% secara acak, sedangkan data uji (testing) diambil 20% yang berisi hasil sisa dari proses pembagian data latih (training). Perbandingan pembagian data latih (training) dan data uji (testing) secara bebas danimbang seperti pada Tabel 5 berikut:

Tabel 5. Pembagian Data

	Label Positif	Label Negatif	Total
Data Latih (Training)	180	220	400
Data Uji (Testing)	45	56	101
Total	225	276	501

Setelah dilakukan pembagian data, maka masuk ke tahapan klasifikasi. Proses klasifikasi akan diterapkan menggunakan Support Vector Machine (SVM) dengan kernel yang diaplikasikan ialah Radial Basis Function (RBF). Kernel Radial Basis Function (RBF) akan memanfaatkan parameter untuk performa model klasifikasi yang baik yaitu cost (C) dan gamma (γ), dengan nilai cost (C) yang akan diimplementasikan yaitu (C) = [0.1, 1, 10, 100] dan gamma (γ) yang digunakan (γ) = [1, 0.1, 0.01, 0.001]. Hasil percobaan implementasi parameter (C) dan (γ) dilampirkan pada Tabel 6 berikut:

Tabel 6. Hasil Nilai Uji Rata-rata Terbaik Percobaan Cost dan Gamma

Parameter	$\gamma=1$	$\gamma=0.1$	$\gamma=0.01$	$\gamma=0.001$
C=0.1	0.55 0	0.550	0.550	0.550
C=1	0.62 0	0.755	0.550	0.550
C=10	0.63 3	0.753	0.765	0.55
C=100	0.63 3	0.738	0.760	0.75

Dari hasil percobaan cost (C) dan gamma (γ), dan diperoleh hasil nilai uji rata-rata terbaik pada nilai C = 10 dan nilai $\gamma = 0.01$ untuk membuat model klasifikasi. Hasil evaluasi model SVM dengan confusion matrix dapat dilihat pada Tabel 7:

Tabel 7. Confusion Matrix dari Model Klasifikasi SVM

Aktual	Prediksi	
	Positif	Negatif
Positif	46 (TP)	10 (FN)
Negatif	14 (FP)	31 (TN)

Maka dari Tabel 6 Confusion Matrix dapat dihitung hasil evaluasi model SVM menggunakan rumus (10), (11), (12), dan (13) sebagai berikut:

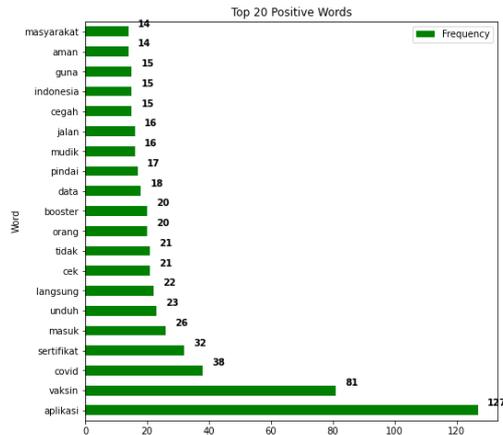
$$\begin{aligned}
 \text{Akurasi} &= \frac{TP+TN}{TP+FN+FP+TN} \times 100\% = \frac{46+31}{46+10+14+31} \times 100\% = 76.24\% \\
 \text{Recall} &= \frac{TP}{TP+FN} \times 100\% = \frac{46}{46+10} \times 100\% = 82.14\% \\
 \text{Presisi} &= \frac{TP}{TP+FP} \times 100\% = \frac{46}{46+14} \times 100\% = 76.67\% \\
 \text{Specificity} &= \frac{TN}{TN+FP} \times 100\% = \frac{31}{31+14} \times 100\% = 68.89\%
 \end{aligned}$$

Selanjutnya dilakukan klasifikasi kedua akan memanfaatkan seleksi fitur dengan algoritma Particle Swarm Optimization (PSO), dengan parameter nilai learning factor (c1 dan c2), inersia (w), jumlah tetangga (k), p-norm Minkowski (p), dan jumlah partikel (n_particles), dimana pada penelitian ini yang digunakan adalah parameter default atau bawaan, yaitu untuk nilai (c1) sebesar 0.5, (c2) sebesar 0.5, (w) sebesar 0.9, (p) sebesar 9, dan (n_particles) sebanyak 30. Proses seleksi fitur dengan algoritma Particle Swarm Optimization (PSO) menghasilkan pengurangan fitur. Hasil pengurangan fitur tersebut akan diaplikasikan kembali pada proses pemodelan klasifikasi dengan algoritma Support Vector Machine (SVM) berkernel Radial Basis Function (RBF) dengan C = 10 dan $\gamma = 0.01$. Maka dilakukan percobaan sebanyak 4 skenario dengan iterasi PSO yang berbeda, yaitu 50, 200, 350 dan 500 dengan parameter yang sama yang dapat dilihat pada Tabel 8 berikut:

Tabel 8. Hasil Akurasi Percobaan 4 Skenario Iterasi PSO

Iterasi PSO	Banyak Fitur yang digunakan	Akurasi SVM+PSO
50	456	85.15%
200	391	87.13%
350	421	86.14%
500	412	88.12%

Dari hasil 4 skenario iterasi PSO dan diperoleh hasil nilai akurasi terbaik pada 500 iterasi dengan jumlah fitur sebanyak 412, yang selanjutnya hasil jumlah fitur tersebut digunakan untuk membuat model klasifikasi SVM. Hasil evaluasi model SVM dengan seleksi fitur PSO dengan confusion matrix dapat dilihat pada Tabel 9



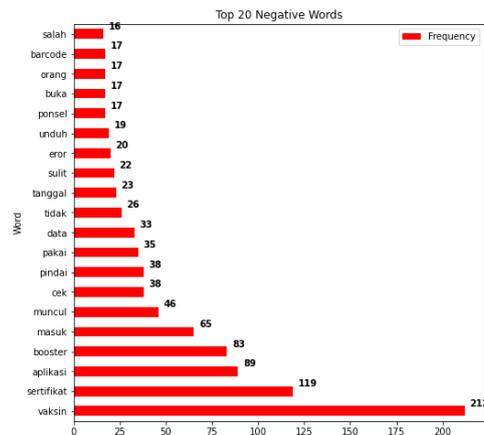
Gambar. 4. Frekuensi Kata Positif

Berdasarkan Gambar 4, kata sering muncul diantaranya ‘aplikasi’ sebanyak 127 kali, ‘vaksin’ sebanyak 81 kali, ‘covid’ sebanyak 38 kali, ‘sertifikat’ sebanyak 32 kali, dan lain-lain.



Gambar. 5. Wordcloud Sentimen Negatif

Berdasarkan Gambar 4, kata yang kemunculannya sering pada sentimen negatif adalah ‘vaksin’, ‘sertifikat’, ‘aplikasi’, ‘booster’, ‘masuk’, ‘muncul’, dan lain-lain. Jumlah kata-kata tersebut dapat dilihat pada Gambar 6 berikut:



Gambar. 6. Frekuensi Kata Negatif

Berdasarkan Gambar 6 tersebut, terlihat bahwa kata ‘vaksin’ sebanyak 212 kali, ‘sertifikat’ sebanyak 119 kali, ‘aplikasi’ sebanyak 89 kali, ‘booster’ sebanyak 83 kali, dan lain-lain.

4 Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan hasil penelitian dapat ditarik kesimpulan yaitu data tweet yang diperoleh pada tanggal tanggal 13 Maret 2022 sampai 11 April 2022 sebanyak 501 data tweet yang diberi label oleh 3 penilai dengan jumlah 225 tweet berlabel positif dan 276 tweet berlabel negatif, sehingga persentase sentimen positif masyarakat terhadap aplikasi PeduliLindungi sebesar 44.9% sedangkan sentimen negatif masyarakat terhadap aplikasi PeduliLindungi berdasarkan data yang diperoleh pada penelitian ini sebesar 55.1%, maka dapat disimpulkan bahwa sentimen opini masyarakat terhadap aplikasi PeduliLindungi bersifat negatif, dengan kata “vaksin” sebanyak 243 kali pada sentimen negatif, sedangkan untuk sentimen positif kata “aplikasi” muncul sebanyak 127 kali. Beberapa kata lain yang sering muncul pada sentimen positif terhadap aplikasi PeduliLindungi dari data yang diperoleh pada penelitian ini adalah “aplikasi”, “vaksin”, “covid”, “sertifikat”, “booster”, “pindai”, “mudik”, “cegah”, dan “indonesia”. Sedangkan beberapa kata yang sering muncul pada sentimen negatif adalah “vaksin”, “aplikasi”, “sertifikat”, “booster”, “cek”, “pindai”, “sulit”, “eror”, “unduh” dan “ponsel”.

Hasil performa model klasifikasi Support Vector Machine menggunakan kernel Radial Basis Function (RBF) dengan *parameter cost* (C) sebesar 10 dan γ sebesar 0.01 dengan kata (*term*) sebanyak 809 dengan pembagian data latih (*training*) 80% dan data uji (*testing*) 20% adalah akurasi sebesar 76.24%, recall (*sensitivity*) sebesar 82.14%, presisi sebesar 76.67%, dan specificity sebesar 68.89%. Hasil performa algoritma Particle Swarm Optimization sebagai seleksi fitur dengan iterasi sebanyak 500 kali yang mengurangi kata (*term*) atau fitur dari 809 menjadi 412 adalah akurasi sebesar 88.12%, recall (*sensitivity*) sebesar 96.43%, presisi sebesar 84.36%, dan specificity sebesar 77.78%, yang berhasil meningkatkan akurasi sebesar 11.88%, recall (*sensitivity*) sebesar 14.29%, presisi sebesar 7.69%, dan specificity sebesar 8.89% daripada model hanya SVM. Pengaruh Algoritma Particle Swarm Optimization terhadap performa Support Vector Machine dalam mengklasifikasikan data tweet adalah algoritma Particle Swarm Optimization menyeleksi fitur berdasarkan posisi terbaik (memiliki bobot TF-IDF terbaik dari setiap kelas) didalam ruang pencarian, sehingga memberikan hasil evaluasi model Support Vector Machine yang lebih baik meliputi akurasi, recall (*sensitivity*), presisi dan specificity.

4.2 Saran

Terdapat beberapa saran dari hasil penelitian untuk pengembangan penelitian selanjutnya agar menjadi lebih baik, yaitu:

- Jumlah data tweet yang digunakan untuk pemodelan klasifikasi lebih banyak sehingga kata (*term*) atau fitur lebih beragam sehingga model yang dibuat menjadi lebih baik dalam memprediksi kata (*term*) dan memiliki performa yang lebih baik.
- Untuk penelitian pada masa yang akan datang, pada pelabelan data diharapkan menggunakan ahli tata Bahasa sehingga tidak terjadi bias pada hasil label masing-masing data tweet, dan tidak memerlukan 3 orang penilai.
- Lebih memperhatikan pada pra proses data tweet agar data yang digunakan lebih bersih sehingga pada proses pemodelan menghasilkan performa yang lebih baik untuk klasifikasi data tweet.
- Untuk penelitian pada masa yang akan datang, dapat menggunakan algoritma seleksi fitur yang lain seperti Chi-Square atau Information Gain.
- Untuk penelitian pada masa yang akan datang, dapat menggunakan algoritma klasifikasi yang lain seperti Naive Bayes.

Referensi

- [1] S. Olivia, J. Gibson, dan R. Nasrudin, “Indonesia in the Time of Covid-19,” *Bull Indones Econ Stud*, vol. 56, no. 2, hal. 13-174, Agustus 2020, doi: <https://doi.org/10.1080/00074918.2020.1798581>.
- [2] P. Putra, F. Y. Liriwati, T. Tahrir, S. Syafrudin, S. Suhono, dan A. Aslan, “The Students Learning from Home Experiences during Covid-19 School Closures Policy in Indonesia,” *jurnal iqra'*, vol. 5, no. 2, hal. 30–42, Sep. 2020.
- [3] A. Fadli, “Mengenal Covid-19 Dan Cegah Penyebarannya Dengan “Peduli Lindungi” Aplikasi Berbasis Andorid,” 2020.
- [4] D. Herdiana, “Aplikasi Peduli Lindungi: Perlindungan Masyarakat Dalam Mengakses Fasilitas Publik di Masa Pemberlakuan Kebijakan PPKM,” *Jurnal Inovasi Penelitian*, vol. 2, no. 6, hal. 1685-1693, 2021.
- [5] V. Alatas, A. G. Chandrasekhar, M. Mobius, B. A. Olken, dan C. Paladines, “When Celebrities Speak: A Nationwide Twitter Experiment Promoting Vaccination in Indonesia,” *Communication & Computational Methods eJournal*, 2019.

- [6] H. S. Arifin, W. Widyowati, dan D. T. Hernawaty, “Freedom of Expression Di Media Sosial Bagi Remaja Secara Kreatif Dan Bertanggung Jawab: Bagi Siswa SMA Al-Ma’soem Rancaekek dan SMA Muhammadiyah Pangandaran,” *Jurnal Pengabdian Kepada Masyarakat*, vol.1, no. 5, hal. 332-337, Oktober 2017.
- [7] R. Tineges, A. Triayudi, dan I. D. Sholihati, “Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM),” *Jurnal Media Informatika Budidarma*, vol. 4, no. 3, Juli 2020, doi: <http://dx.doi.org/10.30865/mib.v4i3.2181>.
- [8] B. Rifai, B. D. Febryanto, F. Yulianto, dan N. Reflianah, “Analisis Sentimen Opini Publik Terhadap Penerapan Kebijakan Social Distancing Dalam Pencegahan Covid-19,” *Paradigma*, vol. 23, no. 1, 2021.
- [9] F. Adams, L. Ernawati, dan N. Chamidah, “Analisis Sentimen Vaksin COVID-19 pada Twitter Menggunakan Algoritma Support Vector Machine,” dalam *Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, vol. 2, no. 2, 2021.
- [10] D. Alita, dan A. Rahman, “Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier,” *Jurnal Komputasi*, vol. 8, no. 2, 2020, doi: <http://dx.doi.org/10.23960%2Fkomputasi.v8i2.2615>.
- [11] D. Sarkar, “Text Analytics with Python,” dalam *Apress*, 2016.