

Analisis Sentimen Terhadap Aplikasi Peduli Lindungi pada Jejaring Sosial Twitter Menggunakan Algoritma *Naïve Bayes* dan Seleksi Fitur *Particle Swarm Optimization*

Muhamad Hanif Razka¹, Theresiawati², Nurul Chamidah³

Informatika /Fakultas Ilmu Komputer

Universitas Pembangunan Nasional Veteran Jakarta

Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

hanifr@upnvj.ac.id¹, theresiawati@upnvj.ac.id², nurul.chamidah@upnvj.ac.id³

Abstrak. Aplikasi PeduliLindungi merupakan sebuah aplikasi resmi dari Kominfo yang bekerjasama dengan beberapa kementerian lainnya. Aplikasi ini bertujuan untuk membantu mencegah penyebaran virus COVID-19 karena selalu terhubung dengan penggunaannya berdasarkan lokasi. Penelitian ini menggunakan data opini publik terhadap penggunaan aplikasi PeduliLindungi dari hasil *tweets* masyarakat menggunakan kata kunci seperti Peduli Lindungi, *hashtag* #PeduliLindungi dan pengguna yang menyebutkan *username* @PLindungi. Pengambilan data dilakukan pada tanggal 13 Maret hingga 11 April 2022. Dalam penelitian ini memiliki tujuan untuk melakukan proses mengkategorikan sebuah data *tweet* menjadi sentimen bersifat positif dan negatif dan menggunakan algoritma *Naïve Bayes* untuk proses klasifikasinya kemudian menerapkan penggunaan seleksi fitur *Particle Swarm Optimization* untuk selanjutnya masuk ke dalam tahap evaluasi dengan *confusion matrix* guna melihat perbandingan akurasi penggunaan seleksi fitur bagi algoritma klasifikasi tersebut. Dan dari hasil pengujian menggunakan algoritma klasifikasi *Naïve Bayes* mendapatkan nilai akurasi sebesar 76.23%, Recall sebesar 76.78%, serta *Precision* sebesar 79.62%. Sementara penggunaan seleksi fitur *Particle Swarm Optimization* pada algoritma *Naïve Bayes* mendapatkan hasil terbaik pada proses iterasi PSO sebanyak 250 kali dengan peningkatan nilai akurasi menjadi 80.19% kemudian nilai *recall* menjadi 85.71% serta terdapat peningkatan pada *precision* menjadi 80%.

Kata Kunci: PeduliLindungi, Analisis Sentimen, Twitter, *Naïve Bayes*, *Particle Swarm Optimization*.

1 Pendahuluan

Aplikasi PeduliLindungi digunakan dalam rangka melakukan penelusuran aktivitas yang dilakukan masyarakat di fasilitas umum, dalam penggunaannya aplikasi ini memberikan batasan bagi masyarakat di zona atau wilayah tertentu yang sudah melakukan vaksinasi pertama maupun kedua yang dapat memasuki fasilitas umum [1]. Dengan penggunaannya yang berkaitan dengan penelusuran aktivitas masyarakat di fasilitas umum pastinya aplikasi ini memiliki intensitas penggunaan di kalangan masyarakat dalam berkegiatan sehari-hari akan sangat tinggi dari penggunaan tersebut tentunya akan sangat banyak feedback dari masyarakat baik melalui ulasan aplikasi maupun media sosial.

Melihat animo masyarakat Indonesia terhadap penggunaan media sosial untuk berekspresi serta memberikan pendapat atas hal-hal yang memang dirasakan dalam kehidupan sehari-hari. Membuat salah satu media sosial yang menjadi tempat bagi masyarakat memberikan pendapat mereka maupun berkeluh kesah adalah twitter. Dengan aktifnya masyarakat dan kebebasan setiap individu di negeri ini untuk mengakses hingga memberikan pendapat dalam media sosial, baik pendapat negatif maupun positif. Tentunya akan sangat penting untuk dapat mengetahui kandungan nilai apa yang ada dalam informasi yang diberikan di media sosial tersebut dari penggunaannya [2].

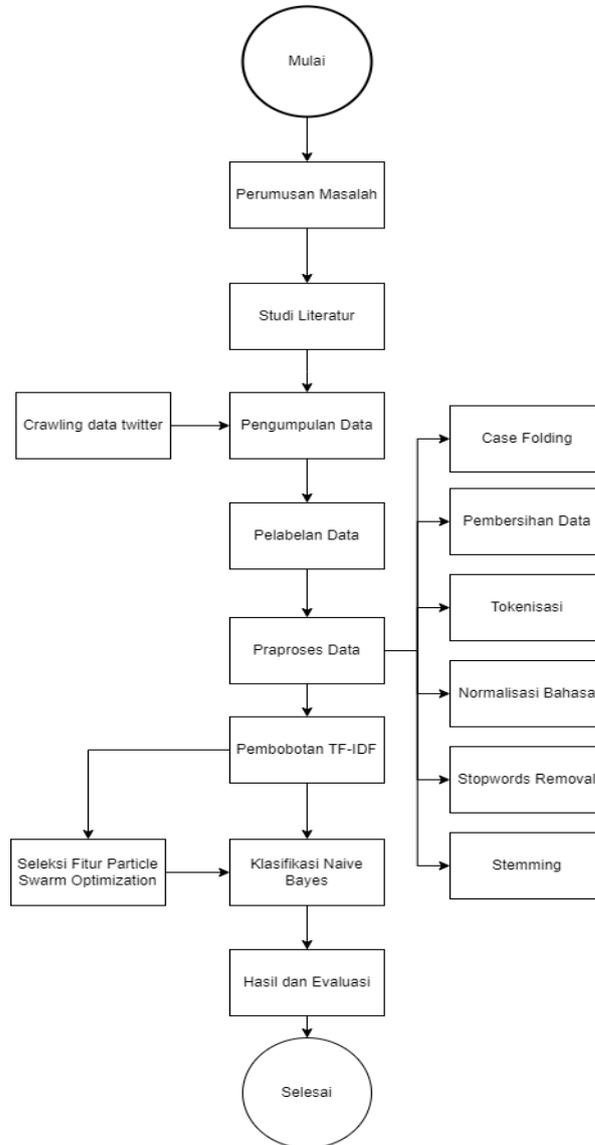
Kemudian dengan menilik pada penelitian terdahulu mengenai tingkat akurasi dari algoritma *Naïve Bayes* yang memiliki hasil evaluasi akurasi yang lebih baik dibandingkan menggunakan algoritma KNN dalam penelitian mengenai analisis sentimen [3]. Serta pada penelitian lainnya terbukti bahwa penggunaan *Particle Swarm Optimization* mampu untuk meningkatkan akurasi dari algoritma yang digunakan pada penelitian dengan *Support Vector Machine* dan pada penelitian tersebut juga menghasilkan tingkat akurasi SVM yang lebih baik setelah menggunakan algoritma PSO [4].

Sehingga berdasarkan pada hal-hal tersebut maka penelitian ini akan meneliti analisis sentimen dari masyarakat mengenai tanggapan serta pengalaman terhadap penggunaan aplikasi PeduliLindungi berdasarkan opini yang dituangkan oleh masyarakat di jejaring sosial media twitter menggunakan algoritma klasifikasi *Naïve Bayes*. kemudian akan melihat efektifitas penggunaan fitur seleksi *Particle Swarm Optimization* yang akan di

simulasikan penggunaannya dalam meningkatkan akurasi dari algoritma klasifikasi tersebut baik sesudah maupun sebelum penggunaan Particle Swarm Optimization.

2 Metodologi Penelitian

Pada penelitian ini, analisis sentimen akan berjalan mengikuti alur rancangan yang telah dibuat. Berikut ini terdapat beberapa tahapan proses penelitian yang akan dilaksanakan seperti Gambar 1 sebagai berikut.



Gambar. 1. Tahapan Penelitian

2.1 Pengumpulan Data

Pada tahapan pengumpulan data, data tweet berasal dari media sosial twitter yang didapatkan melalui metode crawling memanfaatkan keberadaan Application Programming Interface (API) yang sudah dimiliki sebagai fasilitas dari aplikasi twitter yang memang digunakan untuk proses klasifikasi serta menggunakan library rtweet di dalam RStudio. Pada tahap ini pengumpulan dilakukan dengan menggunakan kata kunci Peduli Lindungi, hastag #PeduliLindungi dan pengguna yang menyebutkan username @PLindungi yang dilakukan pada tanggal 13 Maret hingga 11 April 2022. Proses ini memperoleh data dari user twitter yang menuliskan tweet yang sama sesuai kata kunci yang digunakan dan mendapatkan sebanyak 501 data tweet.

2.2 Pelabelan Data

Tahapan selanjutnya adalah pelabelan data secara manual kedalam dua kelas yaitu positif dan negatif berdasarkan data *tweets* yang sudah dikumpulkan. Proses ini dilakukan secara manual dengan penilaian dari 3 orang penilai. Dari penggunaan 3 penilai memungkinkan terjadinya pemberian nilai dalam penentuan pelabelan data sehingga akan memberikan nilai mutlak positif atau negatif dalam setiap dokumen. Namun tetap diperlukan suatu kriteria untuk menggambarkan hasil persetujuan antar penilai (*annotators*) melalui perhitungan *kappa value* yang memiliki persamaan sebagai berikut:

$$Kappa = \left[\frac{P(a)-P(e)}{1-P(e)} \right] \quad (1)$$

Keterangan:

Kappa : Koefisien dari nilai persetujuan, dimana 0 untuk persetujuan dengan hasil kebetulan dan 1 untuk persetujuan total setuju

P(a) : Proporsi frekuensi penilai yang memiliki penilaian sama/ setuju

P(e) : Persetujuan yang memiliki hasil kebetulan

Adapun nilai P(e) yaitu:

$$P(e) = P(positif)^2 + P(negatif)^2 \quad (2)$$

Sementara persamaan untuk penilaian P(a), yaitu:

$$P(a) = \frac{\text{Jumlah Hasil Persetujuan Sama}}{\text{Keseluruhan Data}} \quad (3)$$

Tabel 1. Interpretasi Kata

Kesepakatan	Nilai K
Rendah (poor)	$k < 0.00$
Kurang (slight)	0.00 - 0.20
Adil (fair)	0.21 – 0.40
Cukup (moderate)	0.41 – 0.60
Bagus (substantial)	0.61 – 0.80
Sangat baik (almost perfect)	$k > 0.81$

2.3 Praproses Data

2.3.1 Case Folding

Pada tahapan praproses data yaitu *case folding*, seluruh data yang sudah di dapatkan akan dilakukan perubahan yaitu yang sebelumnya memiliki huruf kapital (*uppercase*) akan dijadikan menjadi huruf kecil (*lowercase*) seluruhnya data *tweets* tersebut. Tujuan hal ini untuk mencegah terjadinya *case sensitive*.

2.3.2 Pembersihan Kata

Proses selanjutnya yaitu membersihkan data *tweets* yang ada dari beberapa hal yang memang tidak memiliki kegunaan, seperti *username*, @, angka, hastag, URL, emoji yang bertujuan untuk mengurangi *noise* serta saat pembersihan kata pun akan menghapus beberapa tanda baca atau *whitespace*.

2.3.3 Tokenisasi

Kemudian tahapan tokenisasi adalah metode untuk semua tweet yang dilakukan untuk membuat kalimat yang berisikan kata-kata diubah menjadi kata tunggal.

2.3.4 Normalisasi Bahasa

Proses dalam normalisasi Bahasa yang berguna untuk mengubah kata yang ada dalam *tweets* untuk diubah apabila terdapat data atau kata yang tidak baku sesuai dengan kaidah kamus besar Bahasa Indonesia (KBBI). Proses ini dilakukan dengan menggunakan *tweet* tersebut kemudian mencocokkan dengan data yang akan di import yaitu Kamus Alay – *Colloquial Indonesian Lexicon.csv*.

2.3.5 Stopword Removal

Proses stopword removal merupakan proses dimana melakukan penghilangan term yang ada pada data yang memang tidak memiliki pengaruh dan kepentingan kepada performa dari proses klasifikasi nantinya.

2.3.6 Stemming

Pada tahap terakhir ini kata yang ada pada data tweets akan dilakukan perubahan dengan menghapus imbuhan baik prefiks, sufiks maupun konfiks hingga membuatnya menjadi ke bentuk kata dasar kembali.

2.4 Pembobotan Kata TF-IDF

Pembobotan Kata (Term Weighting) adalah proses pemberian bobot pada setiap kata sehingga dapat meningkatkan kemampuan dalam melakukan analisis sentimen dalam proses *text mining* [4].

TF-IDF (Term Frequency – Inverse Document Frequency) merupakan sebuah statistic numerik berguna dalam memberikan petunjuk seberapa penting sebuah kata yang ada dalam sebuah dokumen [6].

$$TF - IDF_{t,d,D} = TF_{t,d} \times IDF_{t,D} . \quad (4)$$

Keterangan dari rumus berikut adalah:

$TF_{t,d}$ = Merupakan frekuensi kata t muncul pada dokumen d

$IDF_{t,D}$ = Merupakan banyak dokumen dalam kumpulan dokumen D yang mengandung kata t

D = Merupakan total seluruh dokumen yang tersedia [6].

2.5 Seleksi Fitur *Particle Swarm Optimization*

Particle Swarm Optimization merupakan proses mengoptimalkan sesuatu yang sangat sederhana dalam memodifikasi beberapa parameter, untuk menemukan solusi dalam PSO dari populasi tertentu berdasarkan pada beberapa partikel. Populasi pun diberikan nilai secara acak dan mempunyai batasan nilai terkecil dan terbesar, partikel melacak solusi melewati ruang pencarian kemudian beradaptasi dengan posisi terbaik sendiri (local best) dan beradaptasi dengan letak partikel terbaik dari seluruh kawanan (global best) selama di ruang pencarian [4].

Berikut Rumus Particle Swarm Optimization:

$$v_i(t) = v_i(t-1) + c1r1[Xpbest_i - X_i(t)] + c2r2[Xgbest - X_i(t)] . \quad (5)$$

$$X_i(t) = X_i(t-1) + v_i(t) \quad (3)$$

Keterangan:

$v_i(t)$ = Kecepatan partikel i saat iterasi t

$X_i(t)$ = Posisi partikel i saat iterasi t

- c_1 dan c_2 = Learning rates untuk kemampuan individu dan pengaruh sosial
- r_1 dan r_2 = Bilangan random yang berdistribusi uniformal dalam interval 0 dan 1
- X_{pbest} = Posisi terbaik partikel i
- X_{gbest} = Posisi terbaik global

2.6 Klasifikasi *naïve bayes*

Klasifikasi menggunakan *naïve bayes* merupakan konsep dari probabilitas dapat difungsikan untuk menentukan kelompok kelas sebuah dokumen teks dan juga bisa untuk mengolah data yang berjumlah tidak sedikit namun tetap memiliki akurasi yang tinggi [7].

Dapat dilihat bahwa *Naïve Bayes* memiliki akurasi yang sangat baik sehingga *naïve bayes* menjadi algoritma yang dipakai dalam melakukan pengelompokan atau klasifikasi data.

$$\begin{aligned}
 \text{Posterior} &= \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \\
 P(c|d) &= \frac{P(c) \times P(d|c)}{P(d)}. \quad (6)
 \end{aligned}$$

Keterangan rumus antara lain:

- c : Hipotesis dokumen yang merupakan suatu kelas spesifik
- d : Dokumen dengan kelas yang belum diketahui.
- $P(c|d)$: *Posterior*, probabilitas hipotesis c berdasarkan kondisi d (*posterior probability*).
- $P(c)$: *Prior*, probabilitas hipotesis c (*prior probability*)
- $P(d|c)$: *Likelihood*, probabilitas d berdasarkan kondisi pada hipotesis c .
- $P(d)$: *Evidence*, probabilitas d (Ruhjana, 2019).

Untuk Persamaan rumus dalam mengukur nilai probabilitas yang ada di masing masing kelas, menggunakan persamaan dibawah ini:

$$P(c) = \frac{|doc\ c|}{|document|}. \quad (7)$$

Keterangan rumus antara lain:

- $P(c)$: Probabilitas kemunculan suatu data yang memiliki kelas c .
- $doc\ c$: Total dari data untuk tiap kelas c .
- $|document|$: Total data dari setiap kelas (Ruhjana, 2019).

Rumus untuk dapat mengukur peluang setiap kata yang berasal dokumen yang ada berdasarkan kategori menggunakan persamaan rumus berikut ini.

$$P(W_i|c) = \frac{\text{Count}(W_i,c)+1}{|c|+|V|}. \quad (8)$$

Keterangan:

- $P(W_i|c)$: Peluang kata W_i pada *class* c
- $\text{Count}(W_i, c)$: Total kemunculan kata W_i pada *class* c .
- $|c|$: Total keseluruhan kata pada *class* c
- $|V|$: Total keseluruhan *term* (kata).

Dan untuk melakukan proses klasifikasi dari data uji menggunakan persamaan sebagai berikut.

$$c_{MAP} = \underset{c \in V}{\text{arg max}} P(c) \prod_i P(W_i|c). \quad (9)$$

Keterangan:

- $P(c)$: Peluang kemunculan suatu dokumen yang memiliki *class* c .
- $P(W_i|c)$: Peluang kata W_i pada *class* c .

2.7 Evaluasi

Pada tahap ini melakukan evaluasi menggunakan Confusion Matrix, Confusion matrix adalah proses untuk melakukan metode penghitungan akurasi dalam data mining, Confusion Matrix memiliki perhitungan dengan 4 output antara lain *recall*, *precision*, *accuracy*, dan *error rate*. Berikut merupakan tabel dari confusion matrix yang terdiri dari 4 sel antara lain true positif, false positif, true negative, false negative [8].

Tabel 2. Confusion Matrix

		AKTUAL	
		Positif	Negatif
PREDIKSI	Positif	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	Negatif	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Keterangan tabel diatas:

1. True Positive (TP), *tweet* dengan aktual positif dan diprediksi positif
2. True Negative (TN), *tweet* dengan aktual negatif dan diprediksi negatif
3. False Positive (FP), *tweet* dengan aktual negatif namun diprediksi positif.
4. False Negative (FN), *tweet* dengan aktual positif namun diprediksi negatif.

Persamaan untuk menghitung nilai akurasi:

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

Persamaan untuk menghitung nilai *recall*:

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

Persamaan untuk menghitung nilai *Precision*:

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

2.8 Visualisasi

Dari Hasil klasifikasi dapat memberikan informasi mengenai opini masyarakat terhadap aplikasi peduli lindungi dengan data yang di dapatkan dari media sosial twitter dengan jangka waktu tertentu. Visualisasi akan menampilkan wordcloud serta bar chart berisikan informasi mengenai data yang ada dengan harapan dapat memberikan kemudahan untuk memahami data opini masyarakat terhadap aplikasi peduli lindungi.

3 Hasil dan Pembahasan

Proses pengambilan data ini dilakukan dalam rentang waktu 13 Maret 2022 hingga 11 April 2022 dan mendapatkan 501 Data yang berisikan sentimen opini publik terhadap aplikasi peduli lindungi. Data tweet akan diberikan label dengan 2 kategori, yaitu positif dan negatif. Berikut tabel 4.1 perbandingan hasil pelabelan data dari 3 orang *annotator*.

Tabel 3. Hasil Pelabelan Data

Annotator 1	Annotator 2	Annotator 3	Jumlah Dokumen
Positif	Positif	Positif	211
Positif	Positif	Negatif	9

Positif	Negatif	Negatif	8
Positif	Negatif	Positif	2
Negatif	Positif	Positif	3
Negatif	Negatif	Positif	6
Negatif	Positif	Negatif	2
Negatif	Negatif	Negatif	260
Total Tweet			501

Proses selanjutnya yaitu untuk menghitung seberapa baik tingkat persetujuan antar *annotator* maka dilakukan perhitungan menggunakan metode *kappa value* Dari persamaan *kappa value* (1),(2),(3).

Tabel 4. Hasil Perhitungan Kappa

$$P(a) = \frac{211 + 260}{501} = 0,940$$

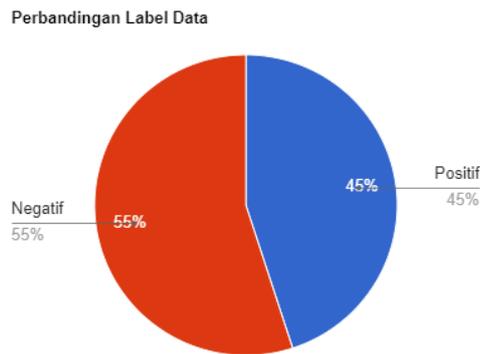
$$P(Negatif) = \frac{9 + 8 + 8 + 2 + 3 + 6 + 6 + 2 + 2 + 260 + 260 + 260}{1503} = 0,549$$

$$P(Positif) = \frac{211 + 211 + 211 + 9 + 9 + 8 + 2 + 2 + 3 + 3 + 6 + 2}{1503} = 0,450$$

$$P(e) = (0,549)^2 + (0,450)^2 = 0.503$$

$$Kappa\ Value = \frac{(0,940 - 0.503)}{(1 - 0.503)} = 0.879$$

Berdasarkan hasil perhitungan dengan kappa value menghasilkan nilai 0,879, dengan hasil tersebut maka nilai kappa >0.80 mendapatkan hasil penilaian kesepakatan sangat baik. Sehingga hasil akhir dari 501 data *tweet* yang dilakukan pengambilan data dari 13 Maret hingga 11 April 2022 dapat dipresentasikan secara diagram pie melalui gambar 2. berikut ini.



Gambar. 2. Presentase perbandingan label data

Dari gambar 4.1 diatas, terlihat bahwa perbandingan sentimen opini publik terhadap aplikasi Peduli Lindungi didapati hasil label positif sebanyak 45% yaitu 225 *tweet* sementara untuk label negatif sebanyak 55% yaitu 276 *tweet* dari keseluruhan total 501 data *tweets*.

Pada tahap selanjutnya yaitu praproses data setelah data selesai dilakukan proses labelling, maka data akan masuk ke dalam tahap praproses yang melalui beberapa tahapan seperti *Case Folding*, Pembersihan Data, Tokenisasi, Normalisasi Bahasa, *Stopwords Removal*, *Stemming*. Proses selanjutnya yaitu pembobotan kata

adalah proses yang harus dilakukan sebelum data nantinya masuk ke dalam proses modelling klasifikasi. Metode yang akan digunakan dalam pemberian bobot pada setiap kata menggunakan metode Term Frequency – Inverse Document Frequency (TF-IDF). Dari hasil pemberian bobot TF-IDF maka didapatkan hasil sebanyak 809 variabel kata dari keseluruhan data sebanyak 501 *tweet*. Berikut tabel 5 adalah gambaran implementasi TF-IDF berdasarkan 4 data *tweet* dari data latih.

Tabel 5. Sampel Data Latih (*Tweet*)

Sampel Data Latih		
	Data Latih	Label
D1	['booster', 'sertifikat', 'muncul', 'aplikasi']	Negatif
D2	['masuk', 'sulit', 'kudu', 'pindai', 'peduli', 'lindung', 'eror']	Negatif
D3	['aplikasi', 'peduli', 'lindung', 'hasil', 'bantu', 'tekan', 'sebar', 'covid']	Positif
D4	['aplikasi', 'mudah', 'bantu', 'tekan', 'sebar', 'covid', 'peduli', 'lindung']	Positif

Berdasarkan sampel data latih diatas, maka selanjutnya akan dilakukan perhitungan dengan TF-IDF (*term frequency – inverse document frequency*), sebagai contoh berikut adalah perhitungan menggunakan TF-IDF yang dilakukan pada tabel 6.:

Tabel 6. Sampel Perhitungan (TF-IDF)

Term	Dokumen				DF	IDF	TF- IDF			
	D1	D2	D3	D4			D1	D2	D3	D4
booster	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
sertifikat	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
muncul	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
aplikasi	1	0	1	1	3	0,125	0,125	0,000	0,125	0,125
peduli	0	1	1	1	3	0,125	0,000	0,125	0,125	0,125
lindung	0	1	1	1	3	0,125	0,000	0,125	0,125	0,125
masuk	0	1	0	0	1	0,602	0,000	0,602	0,000	0,000
sulit	0	1	0	0	1	0,602	0,000	0,602	0,000	0,000
kudu	0	1	0	0	1	0,602	0,000	0,602	0,000	0,000
pindai	0	1	0	0	1	0,602	0,000	0,602	0,000	0,000
eror	0	1	0	0	1	0,602	0,000	0,602	0,000	0,000
hasil	0	0	1	0	1	0,602	0,000	0,000	0,602	0,000
bantu	0	0	1	1	2	0,301	0,000	0,000	0,301	0,301
tekan	0	0	1	1	2	0,301	0,000	0,000	0,301	0,301
sebar	0	0	1	1	2	0,301	0,000	0,000	0,301	0,301
covid	0	0	1	1	2	0,301	0,000	0,000	0,301	0,301
mudah	0	0	0	1	1	0,602	0,000	0,000	0,000	0,602

Tahapan klasifikasi dari data yang sudah disiapkan sebelumnya, dengan menggunakan metode algoritma klasifikasi *Naïve Bayes* serta melakukan percobaan dengan menggunakan seleksi fitur dari algoritma *Particle Swarm Optimization*. Proses ini melakukan pembagian data menjadi 2 yaitu data latih (*training*) dan data uji (*testing*) dari keseluruhan sebanyak 501 data *tweet* dan 809 kata (*term*) atau fitur yang ada pada data. Untuk pembagiannya sendiri data latih (*training*) diambil secara acak sebanyak 80% dari keseluruhan 501 data *tweet* yang sudah memiliki label positif dan negatif, kemudian data uji (*testing*) diambil sebanyak 20% dari sisa

pembagian dengan data latih (*training*). Sebagai contoh berikut perbandingan diantara data latih (*training*) dan data uji (*testing*) pada tabel 7 berikut ini.

Tabel 7. Perbandingan Data Latih dan Data Uji

	Positif	Negatif	Total
Latih (<i>Training</i>)	180	220	400
Uji (<i>Testing</i>)	45	56	101
Total	225	276	501

Proses pemodelan menggunakan Algoritma *Naïve Bayes* maka diperoleh hasil dari proses pembobotan pada tabel 7 yang akan dijadikan sebagai data latih untuk proses modelling klasifikasi, dengan mencari nilai probabilitas kategori serta probabilitas dari kata yang ada pada data latih. Kemudian dalam tahapan selanjutnya yaitu pengujian terhadap data yang sudah melalui proses *training*. Untuk selanjutnya tahapan pengujian dilakukan melalui penggunaan data uji yang berpatokan pada hasil pemodelan *naïve bayes* yang sudah dilakukan pembentukan pada proses latih (*training*). Pada tahapan klasifikasi data uji yang dilakukan adalah data uji akan melalui proses perhitungan probabilitas dari tiap kata yang ada menggunakan acuan nilai TF-IDF untuk kata yang terdapat pada data latih kemudian akan dilakukan proses perhitungan nilai probabilitas dari setiap kata berdasarkan kategorinya. Hasil dari perhitungan tersebut menghasilkan perbandingan nilai probabilitas dari dua kategori yaitu positif dan negatif selanjutnya dari hasil perbandingan tersebut akan didapatkan nilai probabilitas kategori tertinggi yang merupakan hasil klasifikasi dari data uji tersebut.

Proses selanjutnya pada penelitian ini akan dilakukan proses seleksi fitur terlebih dahulu terhadap data *tweet* menggunakan algoritma *Particle Swarm Optimization* Sehingga untuk mendapatkan hasil terbaik dilakukan beberapa scenario percobaan dengan iterasi PSO yang berbeda jumlah iterasinya, dalam penelitian ini yaitu 50, 100, 150, 200, 250, dan 300 dengan parameter *c1* dan *c2* yaitu learning factor dengan nilai 0.5, kemudian nilai inersia (*w*) diberikan nilai 0.9, *p*-norm Minkowski dengan simbol (*p*) memiliki nilai parameter 9, dan parameter terakhir yaitu untuk menentukan jumlah partikel yang disebut (*n_particles*) memiliki nilai 30.

Tabel 8. Hasil Akurasi Percobaan 5 Skenario Iterasi PSO

Iterasi PSO	Banyak Fitur Baru	Runtime NB dan PSO	Akurasi
50	451	14,8 detik	76,23%
100	418	34.4 detik	76,23%
150	459	1 menit 15.5 detik	78,21%
200	405	1 menit 19.5 detik	79,02%
250	406	1 menit 58.8 detik	80.19%
300	393	1 menit 66.6 detik	72.27%

Berdasarkan hasil percobaan dengan beberapa skenario yang dijelaskan pada Tabel 8. Sehingga dapat dikatakan bahwa untuk proses klasifikasi data *tweet* menggunakan algoritma *Naïve Bayes* serta seleksi fitur *Particle Swarm Optimization* (PSO) mendapatkan 250 iterasi sebagai hasil terbaik untuk meningkatkan performa klasifikasi *Naïve Bayes* dan menghasilkan akurasi sebesar 80,19% dan Runtime 1 menit 58.8 detik.

Kemudian hasil dari evaluasi terhadap model berguna sebagai panduan dalam mengukur hasil kinerja *modelling* ke dalam sebuah confusion matrix sesuai dengan jumlah data hasil testing, Berikut adalah tabel 9 hasil evaluasi dari pemodelan *Naïve Bayes*.

Tabel 9. Hasil Evaluasi Klasifikasi *Naïve Bayes*

		AKTUAL	
		Positif	Negatif
PREDI KSI	Positif	43 (TP)	11 (FP)
	Negatif	13 (FN)	34 (TN)

Berdasarkan Tabel 9 performa model *Naïve Bayes* dapat dihitung dengan rumus persamaan poin (10), (11), (12) seperti berikut.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} = \frac{43 + 34}{43 + 13 + 11 + 34} = \frac{77}{101} = 76,23 \%$$

$$recall = \frac{TP}{TP + FN} = \frac{43}{43 + 13} = \frac{43}{56} = 76,78\%$$

$$Precision = \frac{TP}{TP + FP} = \frac{43}{43 + 11} = \frac{43}{54} = 79,62\%$$

Hasil evaluasi dari proses *modelling* menggunakan seleksi fitur *Particle Swarm Optimization* dengan Algoritma Klasifikasi *Naïve Bayes* dengan 250 iterasi menghasilkan *Confusion Matrix* seperti tabel 10 Berikut ini.

Tabel 10. Hasil *Confusion Matrix* Model Klasifikasi *Naïve Bayes* dan Seleksi Fitur PSO

		AKTUAL	
		Positif	Negatif
PREDIKSI	Positif	48 (TP)	12 (FP)
	Negatif	8 (FN)	33 (TN)

Dari hasil yang terdapat pada Tabel 10 *Confusion Matrix* maka dapat dilakukan perhitungan akurasi, *recall*, dan *precision* berdasarkan *modelling* klasifikasi *Naïve Bayes* dan Seleksi Fitur *Particle Swarm Optimization*. Berikut adalah hasil perhitungan tersebut.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} = \frac{48 + 33}{48 + 8 + 12 + 33} = \frac{81}{101} = 80,19 \%$$

$$recall = \frac{TP}{TP + FN} = \frac{48}{48 + 8} = \frac{48}{56} = 85,71\%$$

$$Precision = \frac{TP}{TP + FP} = \frac{48}{48 + 12} = \frac{48}{60} = 80\%$$

Berikut adalah perbandingan antara kedua model pada tabel 11 ini.

Tabel 11. Hasil Perbandingan Performa Model *Naïve Bayes* dan *Naïve Bayes + PSO*

	Akurasi	Recall	Precision
NB	76,23%	76,78%	79,62%
NB + PSO (250 ITERASI)	80,19%	85,71%	80%

Berdasarkan hasil pada Tabel 11 dapat diketahui dengan hasil percobaan yang sudah dilakukan bahwa performa model terbaik untuk klasifikasi sentimen analisis dengan menggunakan data *tweet* terhadap aplikasi PeduliLindungi adalah model klasifikasi menggunakan algoritma klasifikasi *Naïve Bayes* (NB) dengan seleksi fitur menggunakan *Particle Swarm Optimization* (PSO) sebanyak 250 iterasi yang menghasilkan pengurangan

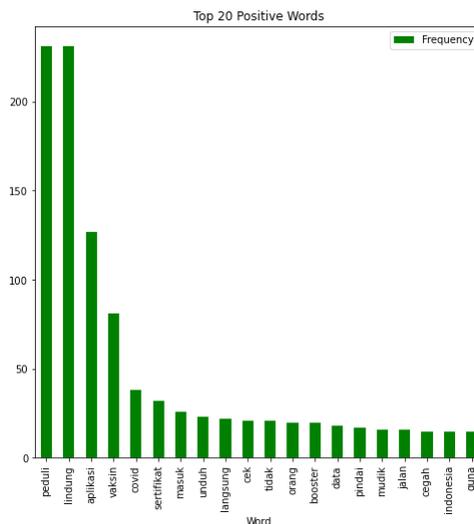
fitur dari 809 fitur menjadi menjadi 406 fitur dengan peningkatan akurasi sebesar 3,96%, *recall* sebesar 8,93% dan *precision* sebesar 0,38%.

Pada penelitian ini dapat divisualisasikan bahwa berdasarkan dengan jumlah kata yang terdapat pada *tweet* serta tingkat kemunculannya, diperoleh hasil kata yang sering muncul untuk sentimen positif sesuai dengan gambar 3 berikut.



Gambar. 3. Wordcloud Sentimen Positif

Berdasarkan gambar tersebut dapat dilihat bahwa pada sentimen positif kata yang sering muncul meliputi ‘Peduli’, ‘Lindung’, ‘Sertifikat’, ‘Aman’, ‘Langsung’, ‘Guna’, Dan lain-lain. Berikutnya adalah visualisasi dari frekuensi dari kata pada sentimen positif terhadap Aplikasi PeduliLindungi.



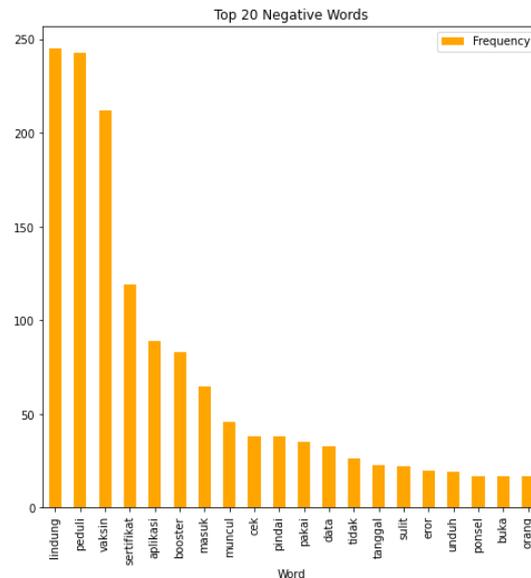
Gambar. 4. Frekuensi 20 Kata Terbanyak pada Sentimen Positif

Berdasarkan Gambar 4 dapat diketahui bahwa kata yang paling sering keluar dari sentimen positif publik terhadap Aplikasi PeduliLindungi dengan data *tweet* yang didapatkan bahwa 3 kata terbanyak diluar aplikasi PeduliLindungi itu sendiri adalah kata aplikasi, kemudian disusul kata vaksin, dan covid, dan serta kata lainnya dengan 20 kata terbanyak berdasarkan kemunculannya pada sentimen positif mengenai Aplikasi PeduliLindungi. Kemudian dapat divisualisasikan pula berdasarkan dengan jumlah kata yang terdapat pada *tweet* serta tingkat kemunculannya, diperoleh hasil kata yang sering muncul untuk sentimen negatif sesuai melalui gambar 5 berikut.



Gambar. 5. WordCloud Sentimen Negatif

Sementara pada sentimen negatif didapatkan hasil bahwa berdasarkan gambar tersebut pada sentimen negatif kata yang memiliki frekuensi kemunculan terbanyak adalah ‘Peduli’, ‘Lindung’, ‘Sertifikat’, ‘Aplikasi’, ‘Booster’, ‘Vaksin’, ‘Pindai’, Dan lain sebagainya. Selanjutnya adalah visualisasi dari frekuensi dari kata pada sentimen negatif terhadap Aplikasi PeduliLindungi. Sesuai Gambar 6 berikut ini.



Gambar. 6. Frekuensi 20 Kata Terbanyak pada Sentimen Negatif

Dari hasil visualisasi pada gambar 6 mendapatkan informasi bahwa 20 kata terbanyak untuk muncul di dalam data *tweet* yang berisikan sentimen negatif terhadap Aplikasi PeduliLindungi memiliki 3 kata teratas setelah kata Peduli Lindungi itu sendiri merupakan kata Vaksin dengan frekuensi kemunculan diangka 212 kali, kemudian sertifikat dengan angka 119, dan aplikasi dengan kemunculan sebanyak 89 kali, disusul oleh 17 Kata lainnya. Sehingga berikut adalah hasil visualisasi 20 Kata yang memiliki frekuensi kemunculan terbanyak pada sentimen negatif berdasarkan data *tweet* mengenai aplikasi PeduliLindungi.

4 Kesimpulan

Berdasarkan pada penelitian berjudul Analisis Sentimen Terhadap Aplikasi Peduli Lindungi pada Jejaring Sosial Twitter menggunakan algoritma Naïve Bayes dan Seleksi Fitur Particle Swarm Optimization dapat dilakukan pengambilan keputusan bahwa:

- 1 Pada data tweet yang didapatkan dari media sosial twitter pada jangka waktu 13 Maret 2022 hingga 11 April 2022 mendapatkan sebanyak 501 data tweet yang mengandung sentimen opini publik terhadap aplikasi PeduliLindungi, yang kemudian dilakukan pelabelan secara manual oleh 3 orang annotator dengan perbandingan sebanyak 276 data tweet memiliki label negatif dan 225 tweet memiliki label positif. Selanjutnya data tersebut akan melalui tahapan praproses data. Dan data yang sudah melalui proses

tersebut akan diberikan nilai bobot melalui metode TF-IDF dan kemudian data tersebut akan dimasukan ke dalam proses klasifikasi menggunakan algoritma Naïve Bayes dengan perbandingan 80% data latih dan 20% data uji. Dari hasil modelling tersebut didapatkan performa klasifikasi menggunakan algoritma Naïve Bayes dengan nilai akurasi sebesar 76.23%, Recall sebesar 76.78%, serta Precission sebesar 79.62%.

2. Performa hasil pengujian Particle Swarm Optimization sebagai seleksi fitur untuk proses klasifikasi algoritma Naïve Bayes terhadap data tweet mengenai opini masyarakat terhadap aplikasi PeduliLindungi dan melalui proses persiapan data yang sama dengan proses klasifikasi melalui algoritma Naïve Bayes mendapatkan hasil terbaik pada proses iterasi PSO sebanyak 250 kali dengan peningkatan nilai akurasi menjadi 80.19% kemudian nilai recall menjadi 85.71% serta terdapat peningkatan pada precision menjadi 80%. Particle Swarm Optimization memiliki pengaruh terhadap performa akurasi pada Naïve Bayes dalam proses klasifikasi diakibatkan pada seleksi fitur yang dilakukan oleh algoritma Particle Swarm Optimization dengan pengaruh posisi terbaik dengan term yang memiliki nilai bobot terbaik dari setiap class yang berada dalam ruang pencarian, sehingga menghasilkan pengurangan fitur dari 809 term menjadi 406 term namun memberikan peningkatan terhadap akurasi dan recall dibandingkan hanya menggunakan algoritma klasifikasi Naïve Bayes.

Beberapa saran yang dapat diberikan pada penelitian di masa selanjutnya, antara lain:

1. Peningkatan Jumlah tweet yang digunakan dalam penelitian selanjutnya sehingga dapat memberikan jumlah kata yang lebih banyak dan dapat menghasilkan proses modelling menggunakan algoritma Naïve Bayes yang lebih baik dalam memprediksi class dari sentimen opini publik.
2. Melakukan proses normalisasi kata yang lebih menambah kosa kata slang word sehingga menghasilkan lebih banyak kata yang dapat dijadikan Bahasa baku sesuai kaidah Kamus Besar Bahasa Indonesia (KBBI)
3. Dalam penelitian selanjutnya, dapat lebih memberi perhatian pada tahapan pra proses sehingga data yang diproses ke dalam tahapan modelling lebih baik dan mengurangi noise pada data yang bisa mengurangi hasil terhadap performa algoritma klasifikasi.
4. Dapat menggunakan algoritma klasifikasi lain seperti Support Vector Machine serta dapat memilih beberapa algoritma untuk seleksi fitur seperti Information Gain atau Chi Square.

Referensi

- [1] PeduliLindungi, 2021, Available: <https://www.pedulilindungi.id/#tentang>
- [2] A. Faisal, Y. Alkhalifi, A. Rifai, dan W. Gata, "Analisis Sentimen Dewan Perwakilan Rakyat Dengan Algoritma Klasifikasi Berbasis Particle Swarm Optimization," *JOINTECS (Journal of Information Technology and Computer Science)*, vol.5, no.2, pp.61, 2020 <https://doi.org/10.31328/jointecs.v5i2.1362>
- [3] M. Syarifuddin, "Analisis Sentimen Opini Publik Mengenai Covid-19 Pada Twitter Menggunakan Metode Naïve Bayes dan Knn," *Inti Nusa Mandiri*, vol.15, no.1, pp.23–28, 2020.
- [4] V. K. S. Que, A. Iriani dan H. D. Purnomo, "Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization," *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, vol.9, no.2, pp.162–170, 2020, <https://doi.org/10.22146/jnteti.v9i2.102>
- [5] F. Afshoh, "Analisa Sentimen Menggunakan Naïve Bayes," *Informatika*, Program Studi Komunikasi, Fakultas Informatika, D A N Surakarta, Universitas Muhammadiyah, 12, 17.
- [6] Y. Cahyono, "Analisis Sentiment pada Sosial Media Twitter Menggunakan Naïve Bayes Classifier dengan Feature Selection Particle Swarm Optimization dan Term Frequency," *Jurnal Informatika Universitas Pamulang*, vol.2, no.1, pp.14, 2017 <https://doi.org/10.32493/informatika.v2i1.1500>
- [7] S. N. J. Fitriyyah, N. Safriadi, dan E. E. Pratama, "Analisis Sentimen Calon Presiden Indonesia 2019 dari Media Sosial Twitter Menggunakan Metode Naive Bayes," *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, vol.5, no.3, pp.279, 2019 <https://doi.org/10.26418/jp.v5i3.34368>
- [8] Karsito dan A. Taufiq, "Analisis Sentimen Terhadap Pemandangan Ibu Kota Pada Media Sosial Twitter Menggunakan Algoritma Naive Bayes Berbasis Particle Swarm Optimization," *SIGMA - Jurnal Teknologi Pelita Bangsa*, vol.10, no.3, pp.173–182, 2020.