

# Implementasi *Particle Swarm Optimization* pada Analisis Sentimen Ulasan Aplikasi Jaminan Kesehatan Nasional (*JKN Mobile*) Menggunakan Algoritma *Support Vector Machine*

Ramadhana Candra Kirana dan R. Mohamad Atok

Departemen Aktuaria, Institut Teknologi Sepuluh Nopember (ITS)

*e-mail:* moh\_atok@statistika.its.ac.id

**Abstrak**—Pengguna internet di Indonesia setiap tahun terus meningkat. Karena kepopuleran internet dan telepon seluler terus meningkat, maka muncul teknologi bernama *m-health*. Layanan *m-health* merupakan layanan medis dan kesehatan masyarakat yang dapat diakses melalui ponsel. BPJS Kesehatan sebagai penyelenggara Jaminan Kesehatan Nasional berupaya untuk meningkatkan kualitas pelayanan dan kemudahan aksesibilitas kesehatan melalui *m-health*. Maka dari itu, BPJS Kesehatan meluncurkan aplikasi Jaminan Kesehatan Nasional (*JKN Mobile*). Untuk melihat kualitas dan kepuasan pengguna terhadap aplikasi ini dapat menggunakan analisis sentimen melalui ulasan yang telah diberikan. Salah satu algoritma dapat digunakan untuk menganalisis sentimen pengguna adalah dengan menggunakan *Support Vector Machine* (SVM). Namun karena SVM mempunyai banyak atribut yang digunakan, diperlukan suatu algoritma lain yang berfungsi sebagai seleksi fitur, maka dari itu dipilihlah seleksi fitur menggunakan *Particle Swarm Optimization* (PSO). Data yang digunakan berupa data ulasan pengguna *JKN Mobile* di *Google Play Store*. Dari data tersebut akan dibagi menjadi dua kelas sentimen, yaitu positif dan negatif. Selanjutnya data akan diklasifikasi menggunakan SVM dan SVM menggunakan PSO. Hasil penelitian menunjukkan bahwa dengan adanya seleksi fitur *Particle Swarm Optimization*, nilai akurasi SVM meningkat. Untuk model paling baik adalah SVM Kernel RBF menggunakan PSO dengan akurasi sebesar 92,39%, F1-Score sebesar 83,74%, dan AUC sebesar 89,75%.

**Kata Kunci**—Analisis Sentimen, *JKN Mobile*, *Particle Swarm Optimization*, *Support Vector Machine*, Ulasan Aplikasi.

## I. PENDAHULUAN

PENGGUNA internet di Indonesia setiap tahun terus meningkat. Dengan meningkatnya jumlah pengguna internet, teknologi yang diciptakan semakin berkembang dan menyebar ke berbagai aspek kehidupan masyarakat, seperti aspek kesehatan. Oleh karena itu muncul layanan *m-health* merupakan layanan medis dan kesehatan masyarakat yang dapat diakses melalui ponsel dan perangkat nirkabel lainnya [1].

Untuk memaksimalkan pelayanan dan berkembang sesuai teknologi, BPJS Kesehatan meluncurkan aplikasi jaminan kesehatan nasional bernama *JKN Mobile*. *JKN Mobile* memberikan layanan seperti layanan administrasi secara daring, melihat status kepesertaan seluruh anggota keluarga, dan dapat mengunduh kartu KIS digital jika diperlukan.

Untuk melihat opini pengguna terhadap kepuasan dan kualitas aplikasi *JKN Mobile*, dapat dilakukan analisis sentimen untuk menganalisis suatu opini tertulis dengan tujuan menentukan apakah opini tersebut termasuk dalam opini netral, positif, atau negatif. Sehingga, dari hasil analisis sentimen akan menggambarkan keadaan emosional seperti kegembiraan,

kearahan, atau kesedihan [2].

Penelitian mengenai analisis sentimen ulasan aplikasi menggunakan *Support Vector Machine* telah banyak dilakukan sebelumnya karena memiliki keunggulan seperti mempunyai tingkat akurasi yang lebih baik daripada dengan menggunakan klasifikasi lain. Namun *Support Vector Machine* juga mempunyai kelemahan umum yang sering terjadi pada analisis sentimen, yaitu banyaknya atribut yang digunakan sehingga data bervolume besar. Hal ini akan mempengaruhi kinerja klasifikasi sehingga dapat menyebabkan akurasi tidak optimal [3]. Oleh karena itu, dibutuhkan optimasi dengan seleksi fitur *Particle Swarm Optimization* untuk mencari nilai range parameter masukan yang dibutuhkan *Support Vector Machine*. *Particle Swarm Optimization* banyak digunakan dan dikembangkan dengan cepat karena penerapannya yang mudah dan hanya sedikit partikel yang perlu disesuaikan.

Pada penelitian analisis sentimen pengguna aplikasi Jaminan Kesehatan Nasional (*JKN Mobile*) menggunakan algoritma *Support Vector Machine* berbasis *Particle Swarm Optimization*. Proses pengujian data pada *Python* dimulai dengan melakukan pengambilan data dari *Google Play Store*. Lalu *raw* data tersebut dibersihkan (*preprocessing*) agar dapat diklasifikasikan dengan menggunakan *Support Vector Machine*. Selanjutnya akan dilakukan optimasi dengan *Particle Swarm Optimization*. Setelah dilakukan analisis dan optimasi maka untuk mengetahui mana model yang terbaik akan dilakukan pengujian nilai akurasi dengan confusion matrix antara algoritma *Support Vector Machine* dan *Support Vector Machine* dengan *Particle Swarm Optimization*.

## II. TINJAUAN PUSTAKA

### A. Aplikasi Jaminan Kesehatan Nasional (*JKN-Mobile*)

Aplikasi Jaminan Kesehatan Nasional (*JKN Mobile*) merupakan aplikasi *mobile* dari BPJS Kesehatan untuk memberikan kemudahan akses dan pelayanan yang optimal kepada peserta Jaminan Kesehatan Nasional. Peserta dapat dengan mudah mengakses berbagai informasi terkait program Jaminan Kesehatan Nasional yang diselenggarakan oleh BPJS Kesehatan melalui *JKN Mobile*.

### B. Analisis Sentimen

Analisis sentimen merupakan metode menganalisis suatu opini tertulis untuk menentukan apakah opini tersebut termasuk dalam opini netral, positif, atau negatif. Analisis sentimen bekerja dengan mengelompokkan polaritas dalam teks yang terdapat dalam data, kalimat, atau level fitur untuk menentu-

Tabel 1.  
Proporsi Kelas *Imbalanced* Data

Tingkat Ketidakseimbangan	Proporsi Kelas Minoritas
<i>Mild</i>	20% - 40% dari dataset
<i>Moderate</i>	1% - 20% dari dataset
<i>Extreme</i>	<1% dari dataset

kan kelas label sentimen yang cocok dengan data tersebut. Sehingga akan menggambarkan keadaan emosional seperti kegembiraan, kemarahan, atau kesedihan [2].

### C. Preprocessing Data

*Preprocessing* adalah langkah yang dilakukan sebelum klasifikasi untuk membersihkan, menghapus, dan mengubah data sumber dalam hal karakter non-alfabet dan kata-kata yang tidak perlu agar data yang digunakan dalam proses klasifikasi lebih optimal [4]. Terdapat enam tahapan *preprocessing* yang digunakan dalam penelitian ini, yaitu: (1) *Case Folding* merupakan proses mengubah semua huruf dalam suatu kalimat menjadi huruf kecil (*lowercase*). Selain itu, akan dilakukan pembersihan data untuk karakter yang tidak digunakan, (2) *Normalization* merupakan proses mengubah kata yang tidak baku menjadi kata baku dan mengubah singkatan kembali ke kata aslinya. (3) *Stemming* merupakan proses menghilangkan infleksi suatu kata untuk mengungkapkan bentuk dasarnya, (4) *Tokenization* merupakan proses memisahkan setiap kata dalam kalimat menjadi kata-kata individual dengan menggunakan spasi sebagai pembatas, dan *Stop-word Removal* merupakan proses menghilangkan kata-kata yang sering muncul dan tidak memiliki arti atau tidak relevan.

### D. Term Frequency-Inverse Document Frequency

Salah satu proses dari teknik ekstraksi fitur adalah pembobotan kata dengan *term frequency-inverse document frequency* (TF-IDF) yang didasarkan pada seberapa sering sebuah kata muncul dalam sebuah dokumen (TF) dan seberapa jarang kata tersebut muncul dalam dokumen yang ada (IDF). Frekuensi kemunculan kata dalam dokumen menentukan penilaian TF-IDF [2].

Rumus yang digunakan untuk menghitung TF ditunjukkan pada Persamaan (1).

$$TF_{ij} = \frac{f_{ij}}{\sum f_{ij}} \quad (1)$$

dengan  $TF_{ij}$  adalah jumlah kemunculan kata  $i$  pada dokumen ke- $j$ ,  $f_{ij}$  adalah jumlah kata  $i$  untuk setiap dokumen, dan  $\sum f_{ij}$  adalah banyaknya dokumen yang mengandung kata- $i$ . Sementara itu, untuk mendapatkan hasil IDF dapat menggunakan rumus pada Persamaan 2.

$$IDF = \log \frac{N}{DF_{ij}} \quad (2)$$

dengan  $N$  adalah jumlah seluruh dokumen, dan  $DF_{ij}$  adalah banyaknya dokumen ke- $j$  yang mengandung kata  $i$ . Sehingga dari Persamaan 1 dan Persamaan 2 akan didapatkan bobot TF-IDF seperti Persamaan (3),

$$W_{ij} = TF_{ij} \times IDF \quad (3)$$

dengan  $W_{ij}$  adalah bobot kata  $i$  pada dokumen- $j$  (bobot *term frequency-inverse document frequency*).

### E. Synthetic Minority Oversampling Technique

*Synthetic Minority Oversampling Technique* merupakan suatu teknik dalam *oversampling* untuk menyeimbangkan da-

ta yang tidak seimbang (*imbalanced data*). Suatu data dikatakan *imbalanced* jika proporsi kelas seperti yang ditunjukkan pada Tabel 1.

*Synthetic Minority Oversampling Technique* (SMOTE) yang bekerja dengan menambah jumlah sampel kelas minor agar dapat setara dengan kelas mayor dengan cara membangkitkan data *synthetic* berdasarkan tetangga terdekat *k-nearest neighbour* di mana tetangga terdekat dipilih berdasarkan jarak *euclidean* antara kedua data.

### F. Support Vector Machine

*Support Vector Machine* merupakan algoritma yang menggunakan pemetaan non linier untuk mengubah data pelatihan asli menjadi dimensi yang lebih tinggi. Dalam dimensi baru ini, ia mencari *hyperplane* pemisah linier optimal atau batas keputusan yang memisahkan kumpulan nilai dari satu kelas dari yang lain. Dengan pemetaan non linier yang tepat untuk dimensi yang cukup tinggi, data dari dua kelas selalu dapat dipisahkan oleh sebuah *hyperplane* [5]. *Hyperplane* terbaik ditentukan dengan mencari *hyperplane* marginal maksimum, atau *hyperplane* dengan margin terbesar.

Untuk menyelesaikan permasalahan non linier, maka dapat dilakukan modifikasi dengan memasukkan fungsi kernel. Beberapa fungsi kernel yang biasa digunakan ditunjukkan pada Persamaan (4), Persamaan (5), dan Persamaan (6).

Fungsi Linier (tanpa kernel):

$$K(x_i, x_j) = x_i \cdot x_j \quad (4)$$

Fungsi Polinomial:

$$K(x_i, x_j) = (x_i \cdot x_j + r)^d \quad (5)$$

Fungsi *Radial Basis Function* (RBF):

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \text{ dengan } \gamma = 1/2\sigma^2 \quad (6)$$

Keterangan dari ketiga fungsi kernel di atas adalah sebagai berikut:  $x_i$  merupakan vektor data *training* ke- $i$  dengan  $i = 1, \dots, n$ ;  $x_j$  merupakan vektor data *training* ke- $j$  dengan  $j = 1, \dots, n$ ;  $d$  merupakan jumlah derajat dari fungsi polinomial; dan  $\gamma$  merupakan ukuran rentangan.

### G. Particle Swarm Optimization

*Particle Swarm Optimization* digunakan sebagai proses pencarian dan optimasi stokastik berbasis populasi. Dalam *Particle Swarm Optimization*, populasi disebut *swarm* sementara individu disebut partikel. Setiap partikel bergerak dengan kecepatan yang sesuai untuk area pencarian dan dipertahankan sebagai posisi terbaik yang pernah dicapai [6].

*Particle Swarm Optimization* berjalan dengan tiga proses tahapan, yaitu inialisasi kecepatan awal, memperbarui kecepatan, dan memperbarui *pBest* (*local best*) dan *gBest* (*global best*) [2]. Pada tahap inialisasi kecepatan awal, saat iterasi ke-0 semua partikel memiliki nilai kecepatan awal 0, sementara itu posisi awal partikel dapat dihasilkan oleh Persamaan (7).

$$x = x_{min} + \text{rand}[0,1]_x(x_{max} - x_{min}) \quad (7)$$

Pada iterasi ke-0, *pBest* akan sama dengan nilai posisi awal partikel. Sedangkan *gBest* akan dipilih berdasarkan *pBest* yang memiliki *fitness* tertinggi. Pada tahap kedua adalah memperbarui kecepatan *particle*. Untuk memperbarui performa kecepatan, dapat digunakan Persamaan (8).

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1 r_1 (pBest_{i,j}^t - x_{i,j}^t) + c_2 r_2 (gBest_{g,j}^t - x_{i,j}^t) \quad (8)$$

Tabel 2.  
Confusion Matrix

		Aktual	
		Positif	Negatif
Prediksi	Positif	<i>True Positive</i>	<i>False Positive</i>
	Negatif	<i>False Negative</i>	<i>True Negative</i>

Tabel 3.  
Nilai AUC dan Jenis Klasifikasi

Nilai	Jenis Klasifikasi
0,90 -1,00	<i>Excellent Classification</i>
0,80 -0,90	<i>Good Classification</i>
0,70 -0,80	<i>Fair Classification</i>
0,60 -0,70	<i>Poor Classification</i>
0,50 -0,60	<i>Failure</i>

dengan:  $v_{i,j}$  merupakan komponen kecepatan;  $w$  merupakan bobot inersia (*weight*);  $c$  merupakan konstanta percepatan;  $r$  merupakan parameter random dengan nilai di antara 0 dan 1;  $pbest_{i,j}^t$  merupakan *pBest* partikel ke- $i$  pada dimensi ke- $j$ ; dan  $gbest_{g,j}^t$  merupakan *gBest* partikel ke- $g$  dalam dimensi ke- $j$ .

Selanjutnya untuk memperbarui posisi dan menghitung *fitness*. Untuk menghitung hal tersebut digunakan Persamaan (9) berikut,

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{9}$$

dengan  $x_{i,j}$  adalah posisi partikel ke- $i$  dalam dimensi ke- $j$ . Tahap terakhir yaitu memperbarui *pBest* dan *gBest*. *pBest* pada iterasi sebelumnya dibandingkan dengan hasil memperbarui posisi. Sehingga yang akan menjadi *pBest* baru adalah *fitness* yang lebih tinggi. *pBest* terbaru yang memiliki *fitness* tertinggi tersebut yang akan akan menjadi *gBest* baru.

H. *k-Fold Cross Validation*

Metode *k-fold cross validation* digunakan untuk memvalidasi metode klasifikasi dalam *data mining*. Metode ini memisahkan data awal secara acak menjadi  $k$  bagian yang terbagi rata (*fold*). Fungsi *k-fold cross validation* untuk memastikan bahwa tidak ada tumpang tindih data yang terjadi pada data pengujian. Salah satu segmen  $k$  digunakan sebagai data pengujian sedangkan segmen lainnya digunakan sebagai data pelatihan untuk membangun model baru [7]. Pada penelitian ini  $k$  yang digunakan adalah 10 untuk menduga akurasi karena dinilai memiliki bias dan ragam yang relatif rendah.

I. *Wordcloud n-Gram*

*Word cloud* adalah representasi grafis dua dimensi dari dokumen yang dibuat dengan *plotting* kata-kata yang sering muncul. Ukuran huruf pada *word cloud* menunjukkan seberapa sering kata tersebut muncul [8]. Untuk menghasilkan multikata pada *word cloud*, dapat digunakan teknik *n-gram*. Teknik *n-gram* dalam *word cloud* adalah teknik yang dapat memperhitungkan multikata dengan  $n$  elemen. Ini dimaksudkan untuk mengaitkan pasangan *n-gram* untuk menunjukkan seberapa dekat setiap pasangan terkait.

J. *Confusion Matrix*

*Confusion Matrix* adalah tabel yang menjelaskan seberapa baik performa model atau algoritme tertentu. Setiap baris matriks mewakili kelas data aktual, dan setiap kolom mewakili kelas data yang diprediksi (atau sebaliknya).

Tabel 4.  
Variabel Penelitian

Variabel	Keterangan	Skala Data
X	Bobot kata kunci dari setiap ulasan yang diperoleh dari hasil TF-IDF	Rasio
Y	Sentimen 0 = Sentimen negatif 1 = Sentimen positif	Nominal

Terdapat 4 label pada Tabel 2, penjelasan mengenai label tersebut adalah sebagai berikut: (1) *True Positive (TP)* adalah data aktual positif yang diprediksi positif, (2) *True Negative (TN)* adalah data aktual negatif yang diprediksi negatif, (3) *False Positive (FP)* adalah data aktual negatif tetapi yang diprediksi data positif. (4) *False Negative (FN)* adalah data aktual positif tetapi yang diprediksi data negatif.

Dari *confusion matrix* akan diperoleh data lain yang berfungsi untuk mengukur performa model, hal tersebut antara lain: *Accuracy* adalah proporsi jumlah prediksi yang benar.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

*Precision* adalah proporsi seberapa sering model memprediksi positif yang benar.

$$Precision = \frac{TP}{FP+TP} \tag{11}$$

*Recall (sensitivity)* adalah proporsi ketika kelas aktual positif, seberapa sering model memprediksi positif.

$$Recall = \frac{TP}{FN+TP} \tag{12}$$

*F1 Score* adalah rata-rata harmonik dari *precision* dan *recall*.

$$F1\ Score = 2 \times \frac{precision \times recall}{precision+recall} \tag{13}$$

K. *Area Under the Curve (AUC)*

AUC berfungsi mengukur sejauh mana model mampu membedakan antara sampel positif dan negatif dengan membandingkan peringkat relatif dari skor probabilitas yang dihasilkan oleh model. Dalam konteks analisis sentimen, AUC digunakan untuk mengevaluasi sejauh mana model dapat membedakan antara ulasan positif dan ulasan negatif. AUC dapat dihitung melalui *confusion matrix* seperti pada Persamaan (14), sementara untuk jenis klasifikasi berdasarkan nilai AUC dapat dilihat pada Tabel 3.

$$AUC = \frac{1}{2} \left( \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right) \tag{14}$$

III. METODOLOGI PENELITIAN

A. *Sumber Data dan Variabel Penelitian*

Data berasal dari ulasan pada halaman situs aplikasi JKN *Mobile* di *Google Play Store*. Periode pengamatan dalam penelitian ini dimulai dari 2 Januari 2023 hingga 2 Februari 2023. Sehingga terdapat 5.259 ulasan pengguna dari aplikasi JKN *Mobile*. Tabel 4 merupakan variabel penelitian yang digunakan, sementara struktur data variabel penelitian ditunjukkan pada Tabel 5.

B. *Tahapan Penelitian*

Langkah-langkah dalam pelaksanaan penelitian ini adalah sebagai berikut: (1) Mengumpulkan data ulasan pengguna JKN *Mobile* dari tanggal 2 Januari 2023–2 Februari 2023 tanpa ulasan dengan *rating* 3 serta memberikan label sentimen positif/negatif kepada data ulasan berdasarkan *rating*. (2) Me-

Tabel 5.  
Struktur Data Variabel Penelitian

Ulasan Ke-	Sentimen (Y)	Kata Kunci ( $X_1$ )	Kata Kunci ( $X_2$ )	...	Kata Kunci ( $X_i$ )
1	$Y_1$	$X_{1,1}$	$X_{2,1}$	...	$X_{i,1}$
2	$Y_2$	$X_{1,2}$	$X_{2,2}$	...	$X_{i,2}$
3	$Y_3$	$X_{1,3}$	$X_{2,3}$	...	$X_{i,3}$
.	.	.	.	.	.
.	.	.	.	.	.
5.259	$Y_{5,259}$	$X_{1,5,259}$	$X_{2,5,259}$	...	$X_{i,5,259}$

Tabel 6.  
Contoh Data Sebelum dan Sesudah Dilakukan *Case Folding*

Sebelum	Sesudah
Sudah ganti faskes online tp pas ke klinik yg baru katanya sebulan setelah pindah onlen baru bisa pakai faskes baru .jadi suruh bayar δÿ±	sudah ganti faskes online tp pas ke klinik yg baru katanya sebulan setelah pindah onlen baru bisa pakai faskes baru jadi suruh bayar

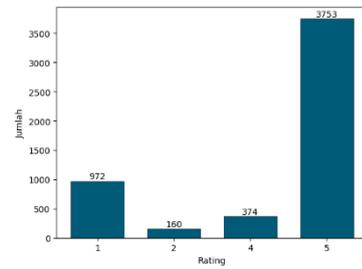
lakukan pengecekan label sentimen, apabila terdapat *rating* yang tidak sesuai dengan ulasan, label akan diganti. (3) Melakukan *preprocessing* data. (4) Melakukan pembobotan kata dengan *term frequency-inverse document frequency* (TF-IDF). (5) Membagi data menjadi data *training* dan data *testing* dengan *k-fold cross validation*. (5) Melakukan *oversampling* dengan SMOTE pada data *training*. (6) Klasifikasi menggunakan SVM kernel linier dan kernel RBF. Pemilihan parameter C dan  $\gamma$  optimal dilakukan dengan *grid search*. Parameter C baik SVM kernel linier dan RBF antara 0,01; 0,1; 1; 10; atau 100. Sedangkan nilai  $\gamma$  untuk SVM kernel RBF antara 0,01; 0,1; 1; 10; atau 100. (7) Seleksi fitur menggunakan *Particle Swarm Optimization*. Pertama, dilakukan pencarian parameter PSO optimal dengan *scikit-optimize*. Kedua, mendefinisikan fungsi *fitness* yang akan digunakan. Ketiga, menentukan partikel dan nilai *threshold* yang akan digunakan. Terakhir, menerapkan seleksi fitur PSO pada SVM dengan parameter yang telah didapatkan pada langkah 10. (8) Melakukan evaluasi model SVM dan SVM-PSO menggunakan *confusion matrix* dan nilai AUC. (9) Memilih model terbaik berdasarkan kriteria nilai parameter evaluasi. (10) Penarikan kesimpulan.

IV. HASIL DAN PEMBAHASAN

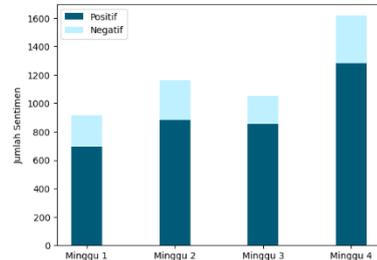
A. Karakteristik Data

Dari data ulasan yang diberikan oleh pengguna JKN *Mobile* dari tanggal 2 Januari 2023 hingga 2 Februari 2023 akan digunakan ulasan dan *rating* dari pengguna. Dalam grafik pada Gambar 1, secara keseluruhan terlihat bahwa *rating* yang paling banyak diberikan oleh pengguna adalah *rating* 5 dengan jumlah *rating* sebanyak 3.753. Selanjutnya, diikuti oleh *rating* 1 sebanyak 972 buah, *rating* 4 sebanyak 374 buah, dan terakhir *rating* 2 sebanyak 160 buah. Selain itu, dari Gambar 1 pengguna yang cenderung memberikan penilaian baik (*rating* tinggi) lebih banyak daripada pengguna yang memberikan penilaian buruk (*rating* rendah).

Untuk melihat bagaimana penilaian pengguna dari waktu ke waktu selama satu bulan tersebut dapat dilihat pada Gambar 2. Dengan mengasumsikan *rating* 1 dan 2 sebagai penilaian negatif dan *rating* 4 dan 5 sebagai penilaian positif, maka didapatkan proporsi perbandingan sentimen awal pengguna berdasarkan *rating* seperti pada Gambar 2. Pengguna



Gambar 1. Jumlah *rating* yang diberikan pengguna.



Gambar 2. Proporsi penilaian yang diberikan pengguna tiap minggu.

dari waktu ke waktu cenderung memberikan penilaian positif kepada aplikasi JKN *Mobile*. Hal ini menunjukkan bahwa kepuasan pengguna terhadap aplikasi JKN *Mobile* cukup tinggi.

B. Pelabelan

Kategori label yang akan diberikan ada 2 macam, yaitu positif dan negatif. Pelabelan ulasan akan didasarkan pada *rating* yang diberikan oleh pengguna di mana *rating* 4 dan 5 akan diberikan label positif, sedangkan *rating* 1 dan 2 akan diberikan label negatif. Selain melihat *rating*, dilakukan pengecekan label secara manual untuk memastikan bahwa label yang diberikan sesuai dengan isi ulasan. Dari 5.259 data ulasan, diperoleh data dengan kategori sentimen positif sebanyak 4.061 ulasan dan data dengan kategori sentimen negatif sebanyak 1.198 ulasan.

C. Preprocessing Data

Data yang diperoleh kemudian di proses menjadi data yang sesuai format agar dapat digunakan untuk analisis berikutnya. Beberapa tahapan *preprocessing* data sebagai berikut: (1) *Case Folding* seperti pada Tabel 6 menunjukkan contoh data sebelum dan sesudah dilakukan *case folding*. (2) *Normalization* seperti pada Tabel 7 menunjukkan contoh data sebelum dan sesudah dilakukan *normalization*. (3) *Stemming* seperti pada Tabel 8 menunjukkan contoh data sebelum dan sesudah dilakukan *stemming*. (4) *Tokenization* seperti pada Tabel 9 menunjukkan contoh data sebelum dan sesudah dilakukan *tokenization*. (5) *Stopword Removal* seperti pada Tabel 10 menunjukkan contoh data sebelum dan sesudah dilakukan *stopword removal*.

D. Term Frequency-Inverse Document Frequency

Untuk mendapatkan bobot TF-IDF, maka dapat dilakukan perhitungan dengan menggunakan Persamaan (1), Persamaan (2), dan Persamaan (3). Berikut merupakan contoh perhitungan untuk kata ‘kasih’:

$$TF_{ij} = \frac{1}{2} = 0,5$$

$$IDF = \log \frac{1350}{53} = 1,4$$

$$W_{ij} = 0,5 \times 1,4 = 0,7$$

Tabel 7.

Contoh Data Sebelum dan Sesudah Dilakukan *Normalization*

Sebelum	Sesudah
sudah ganti faskes online tp pas ke klinik yg baru katanya sebulan setelah pindah onlen baru bisa pakai faskes baru jadi suruh bayar	sudah ganti faskes online tapi saat ke klinik yang baru katanya sebulan setelah pindah online baru bisa pakai faskes baru jadi suruh bayar

Tabel 8.

Contoh Data Sebelum dan Sesudah Dilakukan *Stemming*

Sebelum	Sesudah
sudah ganti faskes online tapi saat ke klinik yang baru katanya sebulan setelah pindah online baru bisa pakai faskes baru jadi suruh bayar	sudah ganti faskes online tapi saat ke klinik yang baru kata bulan telah pindah online baru bisa pakai faskes baru jadi suruh bayar

Tabel 9.

Contoh Data Sebelum dan Sesudah Dilakukan *Tokenization*

Sebelum	Sesudah
sudah ganti faskes online tapi saat ke klinik yang baru kata bulan telah pindah online baru bisa pakai faskes baru jadi suruh bayar	['sudah', 'ganti', 'faskes', 'online', 'tapi', 'saat', 'ke', 'klinik', 'yang', 'baru', 'kata', 'bulan', 'telah', 'pindah', 'online', 'baru', 'bisa', 'pakai', 'faskes', 'baru', 'jadi', 'suruh', 'bayar']

Tabel 10.

Contoh Data Sebelum dan Sesudah Dilakukan *Stopword Removal*

Sebelum	Sesudah
['sudah', 'ganti', 'faskes', 'online', 'tapi', 'saat', 'ke', 'klinik', 'yang', 'baru', 'kata', 'bulan', 'telah', 'pindah', 'online', 'baru', 'bisa', 'pakai', 'faskes', 'baru', 'jadi', 'suruh', 'bayar']	['ganti', 'faskes', 'online', 'klinik', 'pindah', 'online', 'pakai', 'faskes', 'suruh', 'bayar']

Pembobotan TF-IDF dilakukan untuk semua kata dalam ulasan yang telah dilakukan *preprocessing* sebelumnya dengan jumlah 1.350 kata di mana hasil dari pembobotan tersebut akan digunakan untuk analisis klasifikasi.

**E. Oversampling**

Pada data *training* didapatkan bahwa data yang bersentimen positif berjumlah 3.655 dan yang bersentimen negatif berjumlah 1.079. Dengan proporsi sebesar 77,2:22,8, merujuk pada Tabel 11 dapat disimpulkan bahwa *dataset* mengalami ketidakseimbangan tingkat *mild*. Karena data tidak seimbang, maka dibutuhkan teknik *oversampling* agar kelas label seimbang. Teknik *oversampling* SMOTE bekerja dengan menambahkan jumlah data pada kelas minoritas di mana dalam penelitian ini adalah kelas dengan label negatif sehingga diharapkan rasio antar kelas dapat lebih seimbang.

**F. Visualisasi Sentimen Pengguna dengan Wordcloud**

Dalam penelitian ini, *wordcloud* digunakan untuk memvisualisasikan ulasan pengguna yang mempunyai sentimen positif dan sentimen negatif. *Wordcloud* yang digunakan adalah *wordcloud 2-gram* untuk sentimen positif dan *3-gram* untuk sentimen negatif, yang nantinya akan menghasilkan gabungan 2 dan 3 kata dalam visualisasi.

Gambar 3 merupakan gambar visualisasi *wordcloud* dengan sentimen positif. Dapat dilihat bahwa kata ‘bagus’ dan ‘mudah’ mendominasi dalam visualisasi tersebut. Selain ungkapan aplikasi sudah bagus dan mudah digunakan dari Gambar 3 dapat disimpulkan bahwa aplikasi *JKN Mobile* sudah membantu melayani masyarakat, ubah fasilitas kesehatan



Gambar 3. Visualisasi *wordcloud* positif.



Gambar 4. Visualisasi *wordcloud* negatif.

(FASKES) lebih mudah, layanan bagus, respons cepat, dan mendapat harapan dari pengguna agar aplikasi *JKN Mobile* terus meningkatkan layanannya.

Sementara itu, untuk sentimen negatif dari pengguna dapat dilihat pada Gambar 4 yang merupakan gambar visualisasi *wordcloud* dengan sentimen negatif. Dapat dilihat bahwa kata ‘akses’ dan ‘tidak praktis’ mendominasi dalam visualisasi tersebut. Dari Gambar 4 dapat disimpulkan bahwa aplikasi *JKN Mobile* mendapat sentimen negatif dari pengguna di antaranya seperti jam akses yang tidak sesuai, pendaftaran yang tidak praktis, aplikasi setelah diperbarui tidak dapat dibuka, dan kode OTP harus memakai pulsa.

**G. Klasifikasi Menggunakan Support Vector Machine**

Dalam penelitian ini, digunakan dua jenis kernel: kernel linier dan kernel Radial Basis Function (RBF). Setelah membagi data *training* dan *testing* serta melakukan *oversampling* pada data *training*, langkah selanjutnya adalah menentukan parameter *cost* (C) dan  $\gamma$ . Parameter C adalah parameter optimalisasi SVM yang mengurangi kesalahan klasifikasi pada sampel. Parameter  $\gamma$  mengontrol seberapa jauh pengaruh suatu sampel meluas. Pada SVM kernel linier, hanya menggunakan parameter C, sedangkan SVM kernel RBF menggunakan dua parameter, yaitu C dan  $\gamma$ .

**1) Support Vector Machine Kernel Linier**

Untuk memilih nilai C yang paling optimum, digunakan teknik *grid search*. Nilai C ini kemudian dapat digunakan untuk melatih model SVM kernel linier pada seluruh data *training* sebelumnya, sehingga dapat memiliki model yang optimal untuk digunakan pada data pengujian. Dari teknik *grid search*, didapatkan ketepatan klasifikasi optimum menggunakan SVM kernel linier adalah menggunakan parameter C sebesar 10. Setelah parameter C dilatih pada model dengan *10-fold*, didapatkan hasil *accuracy data testing* seperti pada Tabel 12.

Tabel 12 menunjukkan untuk rata-rata keseluruhan subset didapatkan akurasi sebesar 0,9669 untuk data *training* dan 0,9074 untuk data *testing*. Sehingga dapat disimpulkan tidak ada *overfitting/underfitting*. Setelah melihat nilai *accuracy*

Tabel 11.  
Hasil *Accuracy* SVM Linear

Subset	Data Training	Data Testing
1	0,9689	0,9202
2	0,9658	0,9392
3	0,9688	0,9221
4	0,9669	0,9259
5	0,9668	0,9240
6	0,9666	0,9221
7	0,9681	0,9087
8	0,9653	0,9316
9	0,9663	0,9259
10	0,9650	0,9181
Rata-rata	0,9669	0,9237

Tabel 11.  
Hasil *Accuracy* SVM Linier

Subset	Data Training	Data Testing
1	0,9689	0,9202
2	0,9658	0,9392
3	0,9688	0,9221
4	0,9669	0,9259
5	0,9668	0,9240
6	0,9666	0,9221
7	0,9681	0,9087
8	0,9653	0,9316
9	0,9663	0,9259
10	0,9650	0,9181
Rata-rata	0,9669	0,9237

Tabel 12.  
*Confusion Matrix* SVM Linier

		Aktual			
		Data Training		Data Testing	
		Positif	Negatif	Positif	Negatif
Prediksi	Positif	1.048	30	107	12
	Negatif	144	3.510	36	370

Tabel 13.  
Ketepatan Klasifikasi SVM Linier

Parameter Evaluasi	Data Training	Data Testing
<i>Accuracy</i>	0,9632	0,9085
<i>Precision</i>	0,9722	0,8992
<i>Recall</i>	0,8792	0,7482
<i>F1-Score</i>	0,9233	0,8168
AUC	0,9354	0,8584

akan dilakukan pengukuran ketepatan klasifikasi dengan membentuk *confusion matrix* pada Tabel 13.

Dari Tabel 13 dapat diinterpretasikan untuk data *testing* terdapat 107 sentimen positif yang terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 12 sentimen yang masuk ke dalam sentimen negatif. Sementara itu, sebanyak 370 sentimen negatif telah terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 36 sentimen yang masuk ke dalam sentimen positif. Untuk ketepatan klasifikasi SVM kernel linier dapat dilihat pada Tabel 14.

## 2) Support Vector Machine Kernel RBF

Sama seperti pada SVM kernel linier, untuk memilih nilai C yang paling optimum digunakan teknik *grid search*. Sehingga untuk SVM kernel RBF akan menggunakan parameter C sebesar 10 dan  $\gamma$  sebesar 1. Setelah didapatkan parameter yang sesuai untuk algoritma SVM kernel RBF, akan dilakukan pemecahan data menjadi data *training* dan data *testing* dengan menggunakan metode *k-fold cross validation*. Pada proses ini akan dilakukan dengan menggunakan *10-fold*. Tabel 14 menunjukkan hasil *accuracy* pada data *testing* dengan menggunakan *10-fold cross validation*.

Tabel 15 menunjukkan untuk rata-rata keseluruhan *subset*

Tabel 14.  
Hasil *Accuracy* SVM RBF

Subset	Data Training	Data Testing
1	0,9772	0,9335
2	0,9780	0,9278
3	0,9776	0,9144
4	0,9784	0,9144
5	0,9774	0,9259
6	0,9787	0,9278
7	0,9784	0,9297
8	0,9782	0,9030
9	0,9778	0,9297
10	0,9780	0,9219
Rata-rata	0,9779	0,9228

Tabel 15.  
*Confusion Matrix* SVM RBF

		Aktual			
		Data Training		Data Testing	
		Positif	Negatif	Positif	Negatif
Prediksi	Positif	1.049	28	103	16
	Negatif	179	3.475	24	382

Tabel 16.  
Ketepatan Klasifikasi SVM RBF

Parameter Evaluasi	Data Training	Data Testing
<i>Accuracy</i>	0,9562	0,9238
<i>Precision</i>	0,9740	0,8655
<i>Recall</i>	0,8542	0,8110
<i>F1-Score</i>	0,9102	0,8374
AUC	0,9231	0,8854

Tabel 17.  
Hasil *Accuracy* SVM Kernel Linier - PSO

Subset	Data Training	Data Testing
1	0,8710	0,9183
2	0,8595	0,9411
3	0,8781	0,9087
4	0,8728	0,9202
5	0,8758	0,9087
6	0,8743	0,9240
7	0,8806	0,9087
8	0,8650	0,9297
9	0,8677	0,9240
10	0,8669	0,9238
Rata-rata	0,8712	0,9207

didapatkan *accuracy* untuk data *training* sebesar 0,9779 dan untuk data *testing* sebesar 0,9228. Setelah melihat nilai *accuracy* akan dilakukan pengukuran ketepatan klasifikasi dengan membentuk *confusion matrix*.

Dari Tabel 16 dapat diinterpretasikan untuk data *testing* terdapat 103 sentimen positif yang terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 16 sentimen yang masuk ke dalam sentimen negatif. Sementara itu, sebanyak 382 sentimen negatif telah terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 24 sentimen yang masuk ke dalam sentimen positif. Untuk ketepatan klasifikasi SVM Kernel RBF dapat dilihat pada Tabel 17.

## H. Implementasi Particle Swarm Optimizatio

Untuk melakukan optimasi menggunakan *Particle Swarm Optimization* (PSO) digunakan parameter SVM sama dengan yang dipakai sebelumnya. Setiap kombinasi fitur akan dievaluasi menggunakan fungsi *fitness* pada untuk menentukan banyaknya fitur terbaik. Nilai parameter yang digunakan pada penelitian ini, yaitu  $c_1 = 1,6$ ;  $c_2 = 0,4$ ;  $w = 0,8$ ;  $k = 6$  dengan 50 iterasi. Parameter tersebut dipilih setelah dilakukan optimasi parameter terbaik PSO dengan *library scikit-optimize*.

Tabel 18.  
Confusion Matrix SVM Kernel Linier - PSO

		Aktual			
		Data Training		Data Testing	
		Positif	Negatif	Positif	Negatif
Prediksi	Positif	2.850	804	101	18
	Negatif	137	3.517	23	382

Tabel 20.  
Ketepatan Klasifikasi SVM Kernel RBF - PSO

Parameter Evaluasi	Data Training	Data Testing
Accuracy	0,8712	0,9217
Precision	0,7799	0,8487
Recall	0,9541	0,8145
F1-Score	0,8583	0,8313
AUC	0,8840	0,8848

Tabel 19.  
Hasil Accuracy SVM Kernel RBF - PSO

Subset	Data Training	Data Testing
1	0,9689	0,9202
2	0,9658	0,9392
3	0,9688	0,9221
4	0,9669	0,9259
5	0,9668	0,9240
6	0,9666	0,9221
7	0,9681	0,9087
8	0,9653	0,9316
9	0,9663	0,9259
10	0,9650	0,9181
Rata-rata	0,9669	0,9237

1) Implementasi PSO pada SVM Kernel Linier

Metode *grid search* untuk klasifikasi SVM Kernel Linier sebelumnya menghasilkan ketepatan klasifikasi tertinggi saat C bernilai 10. Parameter ini akan digunakan dengan sebagai parameter dalam seleksi fitur SVM Kernel Linier. Pada proses ini akan dilakukan dengan menggunakan *10-fold*.

Setelah dilakukan percobaan pada partikel dan *threshold*, diketahui nilai *accuracy* tertinggi yang dapat dicapai adalah dengan menggunakan partikel 21 *threshold* 0,4. Sehingga, untuk nilai *accuracy* per *subset* ditunjukkan pada Tabel 18.

Tabel 18 menunjukkan rata-rata keseluruhan *subset* didapatkan nilai *accuracy* untuk data *training* sebesar 0,8712 dan untuk data *testing* sebesar 0,9207. Pada seleksi fitur menggunakan PSO, karena menggunakan *10-fold* maka didapatkan juga fitur sebanyak 10 kombinasi.

Banyaknya fitur terpilih pada setiap *fold* tidak terlalu berbeda secara signifikan yaitu berada pada kisaran 817–823 fitur. Jumlah ini berada pada kisaran 60,15% dari banyaknya fitur awal, yaitu 1.350 fitur. Selanjutnya untuk mengukur ketepatan klasifikasi dari SVM Kernel Linier-PSO, maka diperlukan sebuah *confusion matrix*. Berikut adalah hasil *confusion matrix* yang ditunjukkan pada Tabel 19.

Berdasarkan Tabel 19 dapat diinterpretasikan untuk data *testing* terdapat 101 sentimen positif yang terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 18 sentimen yang masuk ke dalam sentimen negatif. Sementara itu, sebanyak 382 sentimen negatif telah terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 23 sentimen yang masuk ke dalam sentimen positif. Untuk ketepatan klasifikasi SVM Kernel RBF dapat dilihat pada Tabel 20.

2) Implementasi PSO pada SVM Kernel RBF

Sama seperti optimasi dalam SVM Kernel Linier-PSO, dalam SVM Kernel RBF-PSO juga menggunakan parameter C

Tabel 20.  
Confusion Matrix SVM Kernel RBF - PSO

		Aktual			
		Data Training		Data Testing	
		Positif	Negatif	Positif	Negatif
Prediksi	Positif	3.429	225	97	23
	Negatif	16	3.638	17	389

Tabel 21.  
Ketepatan Klasifikasi SVM Kernel RBF-PSO

Parameter Evaluasi	Data Training	Data Testing
Accuracy	0,9670	0,9239
Precision	0,9384	0,8083
Recall	0,9954	0,8509
F1-Score	0,9660	0,8291
AUC	0,9685	0,8975

yang sama dengan parameter yang di dapatkan pada *grid search*. Metode *grid search* untuk klasifikasi SVM Kernel RBF sebelumnya menghasilkan akurasi tertinggi saat C bernilai 10 dan  $\gamma$  bernilai 1. Parameter ini akan digunakan dengan sebagai parameter dalam seleksi fitur SVM Kernel RBF.

Setelah dilakukan percobaan pada partikel dan *threshold*, diketahui nilai *accuracy* tertinggi yang dapat dicapai adalah dengan menggunakan partikel 5 *threshold* 0,1. Sehingga, untuk nilai *accuracy* per sub set ditunjukkan pada Tabel 21.

Tabel 21 menunjukkan rata-rata keseluruhan subset didapatkan nilai *accuracy* untuk data *training* sebesar 0,9669 dan untuk data *testing* sebesar 0,9237. Pada seleksi fitur menggunakan PSO, karena menggunakan *10-fold* maka didapatkan juga fitur sebanyak 10 kombinasi.

Banyaknya fitur terpilih pada setiap *fold* tidak terlalu berbeda secara signifikan yaitu berada pada kisaran 1.197–1.204 fitur. Jumlah ini berada pada kisaran 89% dari sedikit jumlah fitur pada SVM (1.350 fitur). Selanjutnya untuk mengukur ketepatan klasifikasi dari SVM Kernel Linier-PSO, maka diperlukan sebuah *confusion matrix*. Berikut adalah hasil *confusion matrix* yang disajikan di Tabel 22.

Dari Tabel 22 dapat diinterpretasikan untuk data *testing* terdapat 96 sentimen positif yang terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 23 sentimen yang masuk ke dalam sentimen negatif. Sementara itu, sebanyak 389 sentimen negatif telah terklasifikasi dengan benar dan terdapat kesalahan prediksi sebanyak 17 sentimen yang masuk ke dalam sentimen positif. Untuk ketepatan klasifikasi SVM Kernel RBF dapat dilihat pada Tabel 23.

I. Perbandingan Klasifikasi

Setelah memperoleh hasil ketepatan klasifikasi dari SVM dan SVM-PSO, langkah selanjutnya adalah membandingkan hasil ketepatan klasifikasi dari kedua algoritma tersebut. Perbandingan algoritma SVM dan SVM-PSO tiap kernelnya pada penelitian ini adalah membandingkan hasil terbaik dari rata-rata pengukuran ketepatan klasifikasi berupa nilai *accuracy*, nilai *F1-score*, dan AUC dari tiap algoritma karena nilai rata-rata tersebut dapat mewakili keseluruhan nilai dari 10 sub set.

Pada Tabel 24 menunjukkan bahwa dengan menggunakan PSO *accuracy* SVM baik kernel linier dan kernel RBF meningkat. Di mana untuk nilai *accuracy*, SVM kernel RBF dengan PSO memiliki nilai tertinggi dan SVM kernel linier tanpa PSO memiliki nilai terendah. Untuk *precision* didapatkan dengan adanya PSO menyebabkan menurunnya nilai *pre-*

Tabel 22.  
Perbandingan Ketepatan Klasifikasi

Metode	Accuracy	Precision	Recall	F1-Score	AUC
SVM Linier	0,9085	0,8992	0,7482	0,8168	0,8584
SVM RBF	0,9238	0,8655	0,8110	0,8374	0,8854
SVM Linier - PSO	0,9217	0,8487	0,8145	0,8313	0,8848
SVM RBF - PSO	0,9239	0,8083	0,8509	0,8291	0,8975

*cision*. Berbanding terbalik dengan *precision*, adanya penggunaan PSO menyebabkan nilai *recall* semakin meningkat. Jika dilihat dari nilai *F1-Score* terlihat bahwa SVM kernel linier terdapat peningkatan nilai *F1-Score* sementara untuk SVM kernel RBF memiliki penurunan nilai. Sedangkan jika melihat dari nilai AUC, penggunaan seleksi fitur menyebabkan nilai AUC semakin bertambah. Oleh karena itu, dapat disimpulkan bahwa untuk mengklasifikasikan sentimen pengguna aplikasi JKN Mobile penggunaan algoritma SVM Kernel RBF-PSO lebih baik daripada ketiga metode klasifikasi lainnya. Hal tersebut dilihat dari parameter evaluasi, di mana sebagian besar nilai parameter evaluasi pada SVM Kernel RBF dengan PSO lebih tinggi daripada parameter evaluasi di klasifikasi yang lain.

## V. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut: (1) Sentimen positif dari pengguna adalah JKN *Mobile* antara lain aplikasi JKN *Mobile* sudah membantu melayani masyarakat, ubah fasilitas kesehatan (FASKES) lebih mudah, layanan bagus, respons cepat, dan mendapat harapan dari pengguna agar aplikasi JKN *Mobile* terus meningkatkan layanannya. Sementara itu, untuk sentimen negatif yang didapatkan adalah jam akses yang tidak sesuai, pendaftaran yang tidak praktis, aplikasi setelah diperbarui tidak dapat dibuka, dan kode OTP harus memakai pulsa. (2) Dari algoritma *Support Vector Machine* dan *Support Vector Machine* dengan *Particle Swarm Optimization*, didapat nilai akurasi untuk *Support Vector Machine* kernel linier sebesar 0,9074 dan untuk kernel RBF sebesar 0,9228. Sementara itu, dengan menggunakan seleksi fitur *Particle Swarm Optimization*, didapat nilai akurasi untuk *Support Vector Machine* kernel linier sebesar 0,9101 dan kernel RBF sebesar 0,9238. (3) Dengan menggunakan seleksi fitur *Particle Swarm Optimization* dapat meningkatkan nilai *accuracy* pada *Support Vector Machine* kernel linier maupun kernel RBF. Sehingga untuk pemilihan model terbaik dengan melihat parameter evaluasi yang digunakan, ditetapkan model yang paling baik adalah *Support Vector Machine* kernel RBF dengan *Particle Swarm Optimization*.

Adapun saran yang dapat Penulis berikan adalah sebagai berikut: (1) Menambah variasi bahasa gaul pada Kamus Alay (*colloquial Indonesian lexicon*) agar proses normalisasi kata dapat dilakukan dengan lebih efektif dan efisien. (2) Melakukan penentuan iterasi optimal untuk algoritma *Support Vector Machine* dengan menggunakan metode seleksi fitur *Particle Swarm Optimization* agar dapat meningkatkan efisiensi dan akurasi model SVM dalam memilih fitur-fitur yang paling relevan dan signifikan dalam proses klasifikasi data. (3) Bagi pengembang aplikasi JKN *Mobile* dapat memperbaiki jam akses untuk *log in* ke aplikasi dan pengiriman kode OTP untuk peserta BPJS Kesehatan dapat melalui *WhatsApp* karena tidak menggunakan pulsa.

## DAFTAR PUSTAKA

- [1] A. D. Permatasari, I. Trihandini, R. BaharuddinNur, dan R. Kurniawan, "Manfaat penggunaan mobile health (m-health) dalam pencatatan dan pelaporan kesehatan ibu," *J. Biostat. Kependudukan, dan Inform. Kesehat.*, vol. 1, no. 2, hal. 100, 2021, doi: 10.51181/bikfokes.v1i2.4810.
- [2] M. Syamsul Bahri, A. Hermawan, E. Kondy, R. Semida, dan S. Siswanto, "Performance comparison of supporting vector machine method without or with particle swarm optimization based on sentiment analysis whatsapp review," *Int. J. Acad. Appl. Res.*, vol. 6, no. 6, hal. 94–101, Jun 2022.
- [3] P. Arsi, R. Wahyudi, dan R. Waluyo, "Optimasi SVM berbasis PSO pada analisis sentimen wacana pindah ibu kota Indonesia," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2 SE-Information Systems Engineering Articles, Apr 2021, doi: 10.29207/resti.v5i2.2698.
- [4] F. A. Muttaqin, "Implementasi Teksmining pada Aplikasi Pengawasan Penggunaan Internet Anak 'Dodo Kids Browser,'" Departemen Teknik Informatika, Universitas Komputer Indonesia, Bandung, 2015.
- [5] J. Han, M. Kamber, dan J. Pei, *Data Mining Concepts and Techniques*, 3rd ed. Massachusetts: Elsevier, 2012. ISBN: 978-0-12-381479-1.
- [6] H. Mubarak, I. Ernawati, dan N. Chamidah, "Optimasi algoritma support vector machine menggunakan seleksi fitur particle swarm optimization pada analisis sentimen terhadap kebijakan pppm," in *Pros. Sem. Nas. Mahasiswa Bid. Ilmu Komput. dan Aplikasinya*, Jakarta, 2022.
- [7] T. Astuti dan Y. Astuti, "Analisis sentimen review produk skincare dengan naïve bayes classifier berbasis particle swarm optimization (PSO)," *J. Media Inform. Budidarma*, vol. 6, no. 4, hal. 1806, 2022, doi: 10.30865/mib.v6i4.4119.
- [8] Q. Castellà dan C. Sutton, "Word storms: multiples of word clouds for visual comparison of documents," in *Proceedings of the 23rd International Conference on World Wide Web*, New York, 2014, hal. 665–676. doi: 10.1145/2566486.2567977.