SCIENCE AND INFORMATION TECHNOLOGY
SINTECH
JOURNAL

SINTECH JOURNAL | ISSN 2598-7305 | E-ISSN 2598-9642
Vol. 6 No 1 – April 2023 | https://s.id/sintechjournal
DOI : https://doi.org/10.31598
Publishing : Prahasta Publisher

Optimasi Parameter *Support Vector Machine* Dengan Algoritma Genetika Untuk Analisis Sentimen Pada Media Sosial Instagram

I Putu Dedy Wira Darmawan[1], Gede Aditra Pradnyana[2], Ida Bagus Nyoman Pascima[3]

[123]Teknik Informatika, Fakultas Teknik dan Kejuruan, Universitas Pendidikan Ganesha
Jl. Udayana No. 11, Singaraja, Indonesia

iputudedywiradarmawan@undiksha.ac.id [1],gede.aditra@undiksha.ac.id [2],gus.pascima@undiksha.ac.id [3]

Received : November, 2022	Accepted : April, 2023	Published : April, 2023

Abstract

Social media is an online media that users use to interact with each other by expressing themselves by giving comments, and one example is Instagram. All the collected comments will form a public opinion. This opinion can be used with sentiment analysis to become information. The method commonly used to carry out sentiment analysis is classification using machine learning. One of the machine learning that is often used is the Support Vector Machine (SVM). However, on non-linear problems such as sentiment analysis, SVM requires the kernel to map vectors into high-dimensional spaces to solve non-linear problems. The problem faced in using the kernel is to choose the optimal parameters for the classification model to produce a good classification model. This study proposes a new approach to obtain optimal parameters for SVM using Genetic Algorithm (GA). This study designed an SVM-GA classification model from the data collection, processing, classification, and evaluation stages. The results showed that the best accuracy produced with parameters optimized with the genetic algorithm was 81.6%, or an increase of 2.4% from the SVM sentiment analysis method without GA optimization.

Keywords: *Optimization, Sentiment Analysis, Instagram, Support Vector Machine, Genetic Algorithm*

Abstrak

Media sosial adalah sebuah media online yang digunakan penggunanya untuk saling berinteraksi dengan mengekspresikan diri dengan cara memberikan komentar, salah satu contohnya adalah Instagram. Semua komentar yang terkumpul tersebut akan membentuk suatu opini masyarakat. Opini tersebut bisa dimanfaatkan dengan analisis sentimen agar menjadi sebuah informasi. Metode yang umum digunakan untuk melakukan analisis sentimen adalah klasifikasi menggunakan machine learning. Salah satu machine learning yang sering digunakan adalah Support Vector Machine (SVM). Namun, pada masalah yang bersifat non-linear seperti analisis sentimen, SVM memerlukan kernel untuk memetakan vektor ke dalam ruang berdimensi tinggi agar dapat menyelesaikan permasalahan non-linear. Permasalahan yang dihadapi dalam menggunakan kernel adalah memilih parameter yang optimal untuk model klasifikasi agar dapat menghasilkan model klasifikasi yang baik. Penelitian ini mengusulkan pendekatan baru untuk mendapatkan parameter yang optimal untuk SVM menggunakan Genetic Algorithm (GA). Penelitian ini merancang sebuah model klasifikasi SVM-GA mulai dari tahap pengumpulan data, pengolahan data, klasifikasi, hingga evaluasi. Hasil penelitian menunjukan bahwa akurasi terbaik yang dihasilkan dengan parameter yang dioptimasi dengan algoritma genetika adalah 81,6% atau meningkat sebesar 2,4% dari metode analisis sentiment SVM tanpa optimasi GA.

Kata Kunci: *Optimasi, Analisis Sentimen, Instagram, Support Vector Machine, Algoritma Genetika*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi khususnya internet di Indonesia sangatlah pesat, sehingga membuat Indonesia menjadi salah satu negara yang memiliki pengguna internet aktif terbanyak dan terus tumbuh tiap tahunnya. Hal ini menyebabkan terjadinya perubahan kebiasaan di masyarakat, saat ini masyarakat yang biasa mengkonsumsi informasi melalui media cetak mulai beralih ke media internet.

Salah satu contoh media internet yang sering digunakan masyarakat adalah media sosial. Media sosial adalah sebuah media *online* yang membuat para penggunanya bisa dengan mudah mencari informasi, berpartisipasi, berbagi, dan menciptakan isi (konten) pada sebuah *platform* di internet. Beberapa media sosial yang populer digunakan di Indonesia contohnya seperti *Facebook, Instagram, dan Twitter.*

Pada umumnya masyarakat Indonesia menggunakan media sosial untuk mengekspresikan diri dengan cara membagikan foto, video, atau ceritanya. Pengguna media sosial juga dapat saling berinteraksi satu sama lain dengan cara saling memberikan *like* atau komentar. Semua buah pikiran yang terkumpul tersebut akan membentuk suatu opini masyarakat di dalam media sosial. Opini ini bisa saja menjadi penilaian terhadap suatu produk, pelayanan, atau keadaan yang sedang terjadi sehingga bisa dimanfaatkan dengan *sentiment analysis* agar menjadi sebuah informasi.

Sentiment analysis (analisis sentimen) adalah bidang studi yang menganalisis opini, sentimen, emosi, penilaian, dan sikap seseorang terhadap suatu produk atau sebuah kebijakan [1]. Beberapa penelitian telah menggunakan informasi dari analisis sentimen untuk memprediksi opini masyarakat terkait suatu produk atau kebijakan. Adilah (2020), menggunakan analisis sentimen untuk melihat opini publik terhadap layanan transportasi online Gojek. Hasil penelitian ini dapat dijadikan sebagai rekomendasi untuk meningkatkan kinerja transportasi online [2]. Adisoka (2019), membuat penelitian yang bertujuan untuk membuat sistem yang dapat mengklasifikasikan komentar apakah mengandung unsur *cyberbullying* di Instagram dengan menggunakan analisis sentimen [3]. Sanjay (2021), Dalam penelitiannya, mengumpulkan data dari Twitter tentang protes petani di India untuk memahami sentimen yang dibagikan publik di tingkat internasional dengan menggunakan analisis sentimen [4]. Kaswidjanti (2020), menggunakan analisis sentimen untuk menganalisis sentimen terhadap suvenir di Yogyakarta melalui media sosial Twitter dan Instagram [5].

Salah satu perguruan tinggi negeri di Bali yaitu Universitas Pendidikan Ganesha, juga memanfaaatkan potensi teknologi media sosial ini sebagai media informasi dan *branding* untuk perguruan tinggi. Jumlah pengguna yang besar dan pertambahan yang selalu konsisten meningkat setiap tahunnya membuat media sosial dapat digunakan menjadi *platform* untuk melakukan *branding*, diplomasi, dan penyampaian informasi yang lebih efektif. Saat ini Universitas Pendidikan Ganesha memiliki beberapa akun media sosial seperti Facebook, Twitter, Instagram, dan Youtube. Pertumbuhan interaksi paling aktif terjadi pada akun Instagram (@undiksha.bali) yang saat ini (Mei 2022) memiliki jumlah pengikut sebesar 31,6 ribu orang. Media sosial ini membuat penguna lain dapat berinteraksi, beropini, dan memberikan penilaian secara terbuka kepada Universitas Pendidikan Ganesha dengan cara memberikan komentar langsung pada akun tersebut.

Dari komentar yang masuk tersebut bisa dimanfaatkan untuk melakukan analisis sentimen. Analisis sentimen penting untuk dilakukan karena mengetahui sentimen yang diberikan publik kepada suatu lembaga, dapat membantu memberikan gambaran penilaian masyarakat apakah memberikan respon yang positif, netral, atau negatif terhadap kinerja atau kebijakan dari lembaga tersebut. Hasil analisis tersebut dapat digunakan sebagai acuan dalam melakukan strategi promosi, pengambilan keputusan, atau peningkatan layanan universitas.

Salah satu cara untuk melakukan analisis sentimen adalah dengan menggunakan model klasifikasi. Klasifikasi adalah pemberian kategori atau pengelompokkan data berdasarkan karakteristik atau pola yang sama untuk mempermudah pencarian informasi.

Metode umum yang digunakan untuk melakukan klasifikasi adalah dengan menggunakan *machine learning*. Salah satu algoritma klasifikasi yang sering digunakan untuk analisis sentimen adalah *Support Vector Machine* (SVM). SVM melakukan klasifikasi dengan cara mencari *hyperplane* (garis pemisah) terbaik untuk memisahkan data ke dalam beberapa kelas sentimen. Keunggulan dari SVM adalah akurasinya yang baik terutama untuk klasifikasi untuk dua kelas [6][7][8].

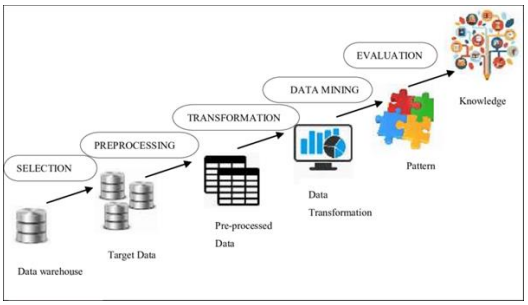
Tantangan dalam menggunakan SVM adalah ketika menerapkannya pada data yang tidak linear dan lebih dari dua kelas. Dalam melakukan klasifikasi data yang sifatnya tidak linear, SVM memiliki sebuah fungsi yang bernama fungsi kernel atau *kernel trick*. Konsep dari *kernel trick* ini adalah dengan menggunakan ruang dengan dimensi yang lebih tinggi untuk membuat *hyperplane*. Beberapa fungsi kernel yang digunakan di SVM contohnya seperti kernel *Radial Basis Function* (RBF). Pada kernel tersebut terdapat *hyperparameter* yang nilainya berpengaruh untuk menentukan kinerja klasifikasi yang dihasilkan oleh SVM. Kelemahan dari SVM adalah sulitnya menentukan nilai dari parameter kernel yang optimal [9][10]. Oleh sebab itu, direkomendasikan beberapa algoritma *metaheuristic* atau pencarian untuk mencari nilai parameter SVM yang optimal. Beberapa penelitian sebelumnya telah menggunakan metode ini untuk mendapatkan nilai parameter yang lebih optimal. Ramasamy (2021), pada penelitiannya menggunakan algoritma pencarian yaitu *Cuckoo Search Optimization* (CSO), *Ant Lion Optimizer* (ALO), dan *Polar Bear Optimization* (PBO) untuk mengoptimasi parameter SVM [11]. Harafani (2020), menggunakan *Genetic Algorithm* (GA) untuk mengoptimasi klasifikasi SVM untuk melakukan perkiraan terhadap masalah penyakit hati [12].

Salah satu algoritma pencarian yang sering digunakan adalah *Genetic Algorithm* (algoritma genetika). *Genetic Algorithm* (GA) adalah algoritma pencarian yang terinspirasi dari prinsip genetika dan seleksi alam dari teori evolusi [13]. Secara sederhana algoritma ini konsep utamanya adalah individu-individu yang paling unggul akan bertahan hidup (diteruskan), sedangkan individu-individu yang lemah akan dieliminasi. Dalam studi kasus ini, contoh dari individu yang dicari adalah nilai parameter kernel SVM-RBF dengan acuan hasil akurasi klasifikasi sebagai *fitness* (solusi) dari algoritma genetika.

Pada penelitian ini akan dilakukan optimalisasi parameter SVM dengan menggunakan metode GA untuk analisis sentiment dari data media sosia. Penelitian ini menguji seberapa besar pengaruh kombinasi algoritma *metaheuristic* (pencarian) terhadap kinerja dari algoritma *classification* (klasifikasi) dalam melakukan analisis sentimen.

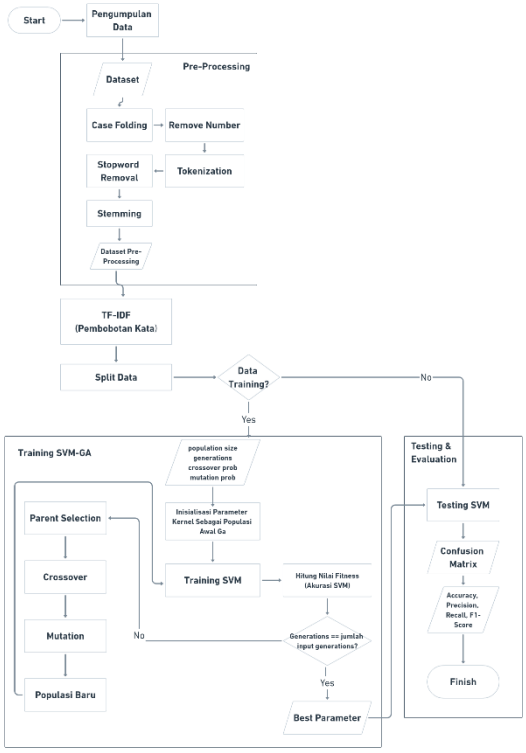
2. METODE PENELITIAN

Tahapan penelitian ini mengadopsi tahapan dari *Knowledge Discovery in Database* (KDD). KDD menggunakan metode *data mining* untuk mencari informasi-informasi yang berharga, pola yang ada di dalam data, yang melibatkan algoritma untuk mengidentifikasi pola pada data tersebut. KDD memilliki beberapa proses tahapan yaitu *Selection*, *Pre-processing*, *Transformation*, *Data Mining*, dan *Interpretation/Evaluation*. Alur dari setiap tahap terlihat pada Gambar 1.



Gambar 1. *Siklus Knowledge Discovery in Database* [Sumber: Ahmad Sabri[14]]

Dari siklus tersebut, dibuat menjadi metode yang digunakan dalam penelitian. Metode didesain menjadi beberapa langkah yang saling terkait untuk menyelesaikan permasalahan penelitian. *Flowchart* penelitian terlihat pada Gambar 2.



Gambar 2. *Flowchart* Metode Penelitian

1. Pengumpulan Data (*Selection*)

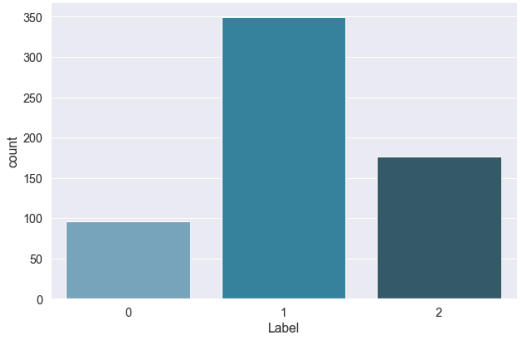
Pengambilan data dilakukan dengan menggunakan metode *scrapping* dengan

bantuan *library* Python yang bernama *Instaloader*. Modul ini dapat melakukan *scrape* untuk konten-konten Instagram seperti postingan pada akun publik dan komentar di dalam postingannya. Hasil yang didapatkan dari hasil scrapping adalah dokumen *spreadsheet* Microsoft Excel.

Data yang berhasil dikumpulkan adalah komentar Instagram yang didapatkan dari akun Instagram @undiksha.bali. Data yang terkumpul yang dijadikan sebagai dataset total berjumlah 621 baris komentar.

Dataset yang berhasil tersimpan dari hasil *scrapping* selanjutnya dilakukan pelabelan untuk mengelompokkan data ke dalam tiga kelompok sentimen yaitu sentimen positif (2), sentimen netral (1), dan sentimen negatif (0). Proses pelabelan dilakukan secara manual untuk mendapatkan data traning yang diperlukan.

Dari hasil pelabelan didapatkan data kelas positif berjumlah 176 baris data (28%), data kelas netral berjumlah 349 baris data (56%), dan data kelas negatif berjumlah 96 baris data (16%). Grafik dari perbandingan jumlah data terlihat pada Gambar 3.



Gambar 3. Perbandingan jumlah *dataset* kelas positif (2), kelas netral (1), dan kelas negatif (0)

2. *Pre-processing* Data

Pengolahan data awal (*pre-processing* data) dilakukan untuk menyederhanakan atau menghilangkan bagian-bagian yang tidak diperlukan untuk proses klasifikasi agar menjadi lebih efisien dan akurat. Langkah-langkah *pre-processing* berguna untuk membersihkan teks berdimensi tinggi, struktur yang buruk dan mengurangi *noise* pada data [2]. Tahapan *pre-processing* terdiri dari beberapa proses, mulai dari *case folding* hingga *stemming*.

a) *Case Folding*

Case folding bertujuan untuk mengubah kata yang berisikan huruf kapital didalamnya menjadi ke bentuk yang standar atau seragam yang dimana biasanya diubah ke huruf dengan format lower-case (huruf kecil). Sehingga data menjadi konsisten ketika menjadi sebuah input di dalam model klasifikasi. Contoh *case folding* terlihat pada Tabel 1.

Tabel 1 Contoh *Case Folding*

Teks	***Case Folding***
Semoga bisa mengajak pemuda indonesia untuk bangkit dan maju melawan covid-19	semoga bisa mengajak pemuda indonesia untuk bangkit dan maju melawan covid-19

b) *Remove Number*

Dalam text *classification* khususnya analisis sentimen, biasanya angka (0, 1, 2, 3, 4, ….., 9) dihilangkan dari kalimat karena angka pengaruhnya tidak terlalu signifikan kepada hasil klasifikasi karena kemungkinan tidak memiliki makna dalam kalimat. Proses menghilangkan angka dapat membuat proses selanjutnya menjadi lebih efisien karena angka telah dieliminasi pada tahap ini, sehingga tidak perlu masuk ke tahap pengelompokkan kata atau *tokenization*. Contoh *remove number* terlihat pada Tabel 2.

Tabel 2 Contoh *Remove Number*

Case Folding	***Remove Number***
semoga bisa mengajak pemuda indonesia untuk bangkit dan maju melawan covid-19	semoga bisa mengajak pemuda indonesia untuk bangkit dan maju melawan covid-

c) *Tokenization*

Tokenization dilakukan untuk memecah kalimat menjadi sekumpulan kata. Tujuan dari *tokenization* adalah untuk menyederhanakan kalimat menjadi kumpulan kata tunggal atau *token* sehingga mempermudah proses pembobotan kata. Contoh *tokenization* terlihat pada Tabel 3.

Tabel 3 Contoh *Tokenization*

Remove Number	***Tokenization***
semoga bisa mengajak pemuda indonesia untuk bangkit dan maju melawan covid-	['semoga', 'bisa', 'mengajak', 'pemuda', 'indonesia', 'untuk', 'bangkit', 'dan', 'maju', 'melawan', 'covid-']

d) *Stopwords Removal*

Stopword Removal dilakukan untuk menghilangkan kata hubung yang tidak memiliki arti yang terlalu penting agar mempermudah proses dan meningkatkan kinerja klasifikasi. Penghilangan stopword ini dapat mengurangi ukuran index data dan waktu pemrosesan. Contoh *stopwords* misalnya seperti 'yang', 'dengan', 'dalam', 'untuk', dan sejenisnya. Contoh *stopword removal* terlihat pada Tabel 4.

Tabel 4. Contoh *Stopwords Removal*

Tokenization	***Stopwords Removal***
['semoga', 'bisa', 'mengajak', 'pemuda', 'indonesia', 'untuk', 'bangkit', 'dan', 'maju', 'melawan', 'covid-']	['semoga', 'mengajak', 'pemuda', 'indonesia', 'bangkit', 'maju', 'melawan', 'covid-']

e) *Stemming*

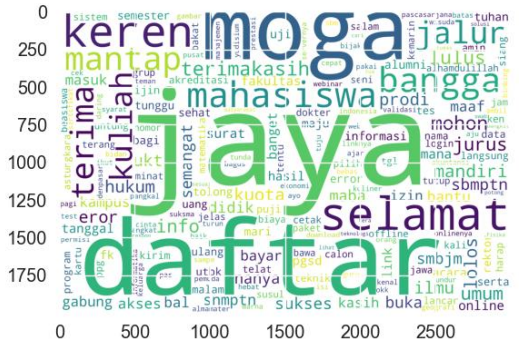
Tahap *cleaning* data terakhir adalah *Stemming*. *Stemming* digunakan untuk memperkecil index data dengan menghilangkan imbuhan awalan atau akhiran dari kata yang memiliki arti yang sama. Dalam membantu proses *stemming* ini dibutuhkan *library* yang bernama Sastrawi. Sastrawi merupakan *library* dalam bahasa pemrograman Python yang digunakan untuk mengubah kata dalam bahasa Indonesia ke akar katanya (*stem*). Contoh *stemming* terlihat pada Tabel 5.

Tabel 5. Contoh *Stemming*

Stopwords Removal	***Stemming***
['semoga', 'mengajak', 'pemuda', 'indonesia', 'bangkit', 'maju', 'melawan', 'covid-']	moga ajak pemuda indonesia bangkit maju lawan covid-

3. Pembobotan Kata (*Transformation*)

Berdasarkan hasil tokenisasi pada tahap *pre-processing* yang telah dilakukan sebelumnya, didapatkan frekuensi kata yang sering muncul pada *dataset*. Dari hasil visualisasi *wordcloud*, dapat dilihat bahwa kata yang sering muncul dicetak lebih besar dari kata lain. Ini berarti kata seperti 'jaya', 'selamat', 'daftar', 'keren', dan kata lainnya yang terlihat pada *Gambar 4* kemungkinan memiliki bobot nilai yang tinggi untuk menjadi kata kunci dalam pengelompokkan suatu kelas sentimen.



Gambar 4. Visualisasi Frekuensi Kata dengan *Wordcloud*

Metode yang digunakan untuk pembobotan kata adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). Metode TF-IDF ini menggabungkan dua konsep yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen dan *inverse* frekuensi dokumen yang mengandung kata tersebut [15].

Token (kata) yang didapatkan dari hasil *pre-processing* kemudian diubah menjadi representasi vektor dan dihitung menggunakan rumus TF-IDF. TF (*Term Frequency*) adalah jumlah kemunculan term dalam dokumen yang bersangkutan, dan IDF (*Inverse Document Frequency*) adalah perhitungan distribusi *term* yang muncul dalam dokumen.

Pada penelitian ini proses pembuatan *word vector* dan pembobotan kata (TF-IDF) dibantu dengan menggunakan *library* Python yang bernama *TfidfVectorizer*. *TfidfVectorizer* merupakan salah satu fungsi yang terdapat pada *library text feature extraction* yang dikembangkan oleh Scikit-learn, yang fungsinya adalah untuk melakukan proses TF-IDF.

4. Klasifikasi SVM-GA (*Data Mining*)

Tahap selanjutnya adalah klasifikasi model dengan SVM-GA, pada penelitian ini dataset dibagi menjadi data pelatihan dan data pengujian dengan proporsi 80% untuk latih dan 20% untuk uji. Selain itu, untuk mendapatkan kombinasi terbaik dari data yang telah dibagi sebelumnya, diterapkan juga *K-Fold Validation* dengan jumlah 10 *folds*. Tujuan dari proses pelatihan dan pengujian ini adalah membangun model klasifikasi dan menghitung tingkat kinerja dari metode SVM-GA pada aspek akurasi saat memprediksi data uji.

Tahap pengklasifikasian dilakukan dengan cara membangkitkan nilai parameter yang dibutuhkan kernel SVM-RBF secara acak dengan mengggunakan algoritma genetika. *library* Python yang digunakan untuk membantu

proses ini adalah **sklearn_genetic** dengan fungsi *GASearchCV*.

GASearchCV mempunyai beberapa parameter untuk melakukan konfigurasi algoritma genetika. Contohnya seperti *population size*, *generations*, *crossover probability*, dan *mutation probability* [16]. Parameter-parameter ini dibutuhkan agar algoritma genetika dapat menjalankan fungsinya. Berikut ini adalah rancangan *testing scenario* yang digunakan dalam proses pengujian dengan menggunakan beberapa kombinasi parameter GA.

5. Evaluasi (Evaluation)

Tahap evaluasi dilakukan untuk mengukur hasil dari model klasifikasi. Hasil dari klasifikasi ditampilkan ke dalam bentuk *confusion matrix* untuk membandingkan kelas *predicted* dan kelas *actual*. Model *confusion matrix* untuk tigas kelas sentimen terlihat seperti Tabel 6.

Tabel 6. *Confusion Matrix*

		Prediction		
		Positif	Netral	Negatif
Actual	Positif	Positif-Positif	Positif-Netral	Positif-Negatif
	Netral	Netral-Positif	Netral-Netral	Netral-Negatif
	Negatif	Negatif-Positif	Negatif-Netral	Negatif-Negatif

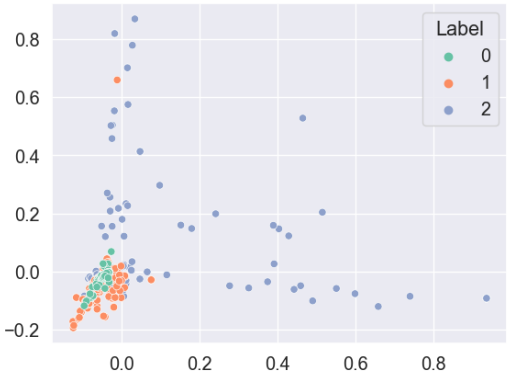
Dari *confusion matrix* tersebut, dapat diukur **performa** model klasifikasi dengan menggunakan *accuracy, precission, recall,* dan *f1-score*.

3. HASIL DAN PEMBAHASAN

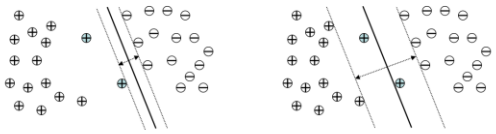
Klasifikasi sentimen analisis dengan menggunakan SVM berkerja dengan cara menggunakan pelatihan dan uji dari *dataset* untuk melakukan klasifikasi. Pada penelitian ini, SVM digunakan untuk melakukan klasifikasi untuk tiga kelas yaitu kelas positif, netral, dan negatif untuk analisis sentimen dan menggunakan data yang bentuknya tidak linear seperti yang terlihat pada Gambar 5. Sehingga untuk melakukan klasifikasi membutuhkan SVM dengan *kernel trick*.

Tantangan penggunaan kernel pada SVM adalah menentukan nilai parameter kernel yang dibutuhkan untuk melakukan klasifikasi. Nilai parameter ini sangat penting dalam pembuatan *hyperplane* untuk SVM-RBF, karena nilai parameter ini akan berpengaruh pada performa hasil klasifikasi nantinya. RBF kernel membutuhkan dua parameter kernel yaitu C dan γ. Parameter C dan γ nilainya ditentukan sebelum pelatihan model.

Parameter C menentukan seberapa **besar penalti** yang diberikan kepada data yang salah diklasifikasikan. Nilai parameter C yang besar membuat toleransi terhadap kesalahan dalam pengklasifikasian data menjadi lebih besar, sebaliknya dengan **nilai C yang rendah** maka *decision boundary* yang dihasilkan akan **lebih ketat** karena nilai toleransi terhadap kesalahan klasifikasi lebih rendah. Ilustrasi dari pengaruh parameter C dapat dilihat pada Gambar 6. Nilai C yang digunakan pada gambar sebelah kiri lebih besar dari gambar sebelah kanan.

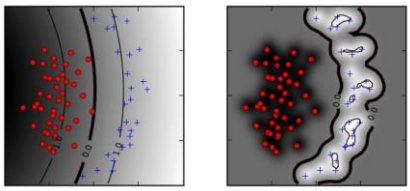


Gambar 5. Bentuk Kelas *Dataset* (0) Negatif, (1) Netral, (2) Positif



Gambar 6. Ilustrasi Pengaruh Nilai Parameter C [Sumber: Braga[17]]

Parameter γ (dibaca *Gamma*) merupakan parameter yang digunakan untuk SVM yang menggunakan *kernel trick* seperti SVM-RBF. Parameter *gamma* digunakan untuk mengontrol jarak pengaruh satu titik pelatihan. Nilai *gamma* yang rendah menunjukkan radius kesamaan yang besar yang menghasilkan lebih banyak titik yang dikelompokkan bersama. Sedangkan pada nilai gamma yang tinggi, titik-titik tersebut harus sangat dekat satu sama lain agar dapat dipertimbangkan dalam kelompok (atau kelas) yang sama. Ilustrasi pengaruh parameter γ terlihat pada Gambar 7. Nilai γ pada gambar sebelah kanan memiliki nilai lebih besar dari gambar sebelah kiri.



Gambar 7. Ilustasi Pengaruh Nilai Parameter γ [Sumber: Braga[17]]

Namun tantangan dalam menggunakan parameter dalam SVM adalah **sulitnya menemukan nilai yang tepat** untuk mendapatkan performa terbaik. Menggunakan nilai acak untuk mencari nilai parameter perlu melakukan banyak tebakan hingga mendapatkan performa yang baik, jika tanpa referensi penelitian atau percobaan yang dilakukan oleh orang lain sebelumnya, maka akan sangat sulit untuk menentukan nilai tebakan awal untuk parameter dan memerlukan banyak waktu sampai mendapatkan tebakan yang memberikan performa yang baik. Jadi, untuk mencari nilai parameter paling optimal, peneliti mencoba menggunakan opsi lainnya yaitu dengan menggunakan algoritma genetika.

Untuk mendapatkan konfigurasi terbaik untuk algoritma genetika yang akan digunakan dalam model klasifikasi. Penelitian ini mencoba beberapa kombinasi dari *crossover probability* (0.5, 0.7, dan 0.8), *mutation probability* (0.05, 0.1, dan 0.2)*, population size* (10, 50, dan 250), dan *generations* (10, 30, dan 20). Kombinasi dari parameter *crossover probability* (CP), *mutation probability* (MP)*, population size* (PS), dan *generations (Gen)* ini menghasilkan 81 kombinasi parameter GA. Sampel hasil terbaik dari pengujian terlihat pada Tabel 7.

Tabel 7. Percobaan SVM-GA dengan Beberapa Kombinasi Parameter GA

CP	MP	PS	Gen	Akurasi		Parameter Terbaik	
				Train	Test	C	γ
0,7	0,2	250	30	99%	81,6%	18.61	0.04
0,8	0,05	250	50	99%	80,8%	11.25	0.061
0,8	0,2	250	30	98,8%	80,8%	18.104	0.036
0,8	0,2	250	50	98,8%	80%	10.351	0.054
0,5	0,2	10	10	99,6%	78,4%	38.354	0.103
0,7	0,2	250	30	99%	81,6%	18.61	0.04

Dari total 81 kali percobaan menggunakan kombinasi parameter algoritma genetika, didapatkan kesimpulan. Nilai ***crossover*** pada algoritma genetika digunakan untuk menghindari duplikasi dari *parents* populasi lama ke keturunan selanjutnya. Ini dilakukan agar populasi baru mendapatkan atau memiliki kualitas dari orang tuanya [18]. Dari Tabel 7 terlihat jika nilai 0,7 merupakan nilai crossover probability terbaik, yang artinya ada 70% kemungkinan kromosom populasi baru dibuat dengan crossover dari generasi sebelumnya.

Mutasi umumnya terjadi setelah melakukan *crossover*, nilai mutasi menerapkan perubahan acak ke satu atau lebih gen untuk menghasilkan keturunan baru yang lebih variatif dan bisa menjadi solusi adaptif baru. Misalnya, satu atau lebih gen yang dipilih secara acak dapat dialihkan dari 0 ke 1 atau dari 1 ke 0. Pada percobaan ini 0,2 adalah nilai *mutation probability* terbaik yang artinya kemungkinan terjadi mutasi adalah 20%.

Generations pada penelitian ini digunakan sebagai *termination* (*stopping*) *condition* untuk algoritma genetika. Algoritma genetika akan terus melakukan iterasi sesuai dengan jumlah generations yang ditetapkan. Hasil pada percobaan menunjukkan jika penggunaan *generations* di atas 30 kali membuat hasil klasifikasi mendapatkan hasil yang lebih stabil.

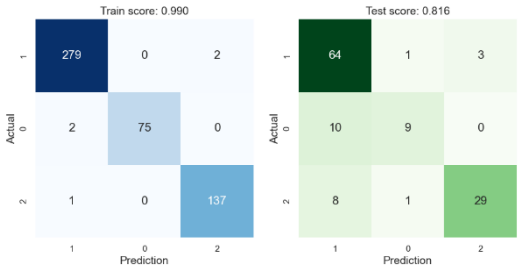
Parameter yang paling mempengaruhi performa dari SVM-GA adalah jumlah *population size*. Ukuran populasi (ruang pencarian) yang kecil, ini berarti sedikit ruang pencarian yang tersedia, sedangkan jika ukuran populasi besar, maka area pencarian meningkat dan beban komputasi menjadi tinggi [16]. Penelitian ini mencoba untuk menguji kembali model klasifikasi dengan *population size* yang lebih besar untuk melihat apakah penambahan jumlah population size memberikan peningkatan performa yang signifikan pada model SVM-GA. Pengujian dilakukan dengan menggunakan *population size* berjumlah 10, 50, 250, 500, dan 1000 dengan konfigurasi yang sama yaitu *crossover probability* dengan nilai 0,7, *mutation probability* dengan nilai 0,2, dan *generations* sejumlah 30. Hasilnya pengujian terlihat pada Tabel 8.

Tabel 8. Percobaan SVM-GA dengan Beberapa Nilai Population Size

Population Size	Akurasi		Waktu Eksekusi (Seconds)	
	Train	Test	Train	Test
10	99,6%	68%	3240	1,7
50	99,6%	77,6%	15720	2,2
250	99%	81,6%	72000	2,2
500	98,8%	80,8%	96000	2,3
1000	97,2%	78,4%	228000	2,5

Dari hasil pengujian pada Tabel 8 diketahui jika nilai ideal untuk jumlah *population size* pada penelitian ini adalah pada rentang 250 – 500, karena memberikan peningkatan performa yang signifikan dibanding penggunaan jumlah *population size* yang lebih rendah. Penggunaan dengan jumlah *population size* kurang 250 memberikan performa yang tidak terlalu baik pada saat *testing*. Sedangkan penggunaan *population size* pada jumlah besar yakni 1000, tidak terlalu memberikan peningkatan performa yang signifikan, performanya justru cenderung menurun dan menambah lama waktu eksekusi yang membuatnya membutuhkan waktu lebih lama dan sumber daya yang lebih besar untuk melakukan *training*.

Performa terbaik SVM-GA didapatkan dengan konfigurasi *population size* sebesar 250, *generations* sejumlah 30, serta *crossover probability* dan *mutation probability* masing-masing dengan nilai 0,7 dan 0,2 dengan pencarian nilai parameter C dan γ dibatasi pada *range* 0,01 sampai 100. Performa akurasi terbaik yang dihasilkan adalah 81,6%. Detail dari performa model terlihat pada Gambar 8 dan Gambar 9.

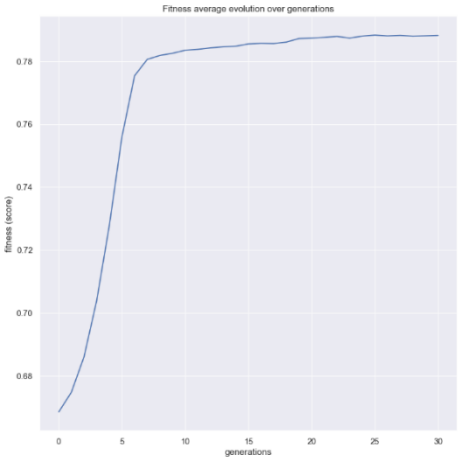


Gambar 8. Hasil *Confusion Matrix*

```
Train report
              precision    recall  f1-score   support

           0       1.00      0.97      0.99        77
           1       0.99      0.99      0.99       281
           2       0.99      0.99      0.99       138

    accuracy                           0.99       496
   macro avg       0.99      0.99      0.99       496
weighted avg       0.99      0.99      0.99       496


Test report
              precision    recall  f1-score   support

           0       0.82      0.47      0.60        19
           1       0.78      0.94      0.85        68
           2       0.91      0.76      0.83        38

    accuracy                           0.82       125
   macro avg       0.83      0.73      0.76       125
weighted avg       0.82      0.82      0.81       125
```

Gambar 9. Hasil Pengukuran Evaluasi Performa *Train* & *Test*

Dari hasil pencarian parameter dengan menggunakan SVM-GA, terjadi peningkatan nilai akurasi (*fitness*) seiring dengan bertambahnya jumlah *generations*. Seperti yang terlihat pada Gambar 10.



Gambar 10. Grafik Evolusi Algoritma Genetika

Hasil ini lebih baik dari penggunaan nilai parameter *default* **Scikit-learn** yang hanya menghasilkan 74,4% dan pemilihan parameter acak secara manual dengan parameter C = 10 dan γ = 0,1 (didapatkan dari beberapa kali percobaan dengan memilih kombinasi secara acak) yang menghasilkan akurasi 79,2%. Perbandingan performa SVM dengan optimasi dan tanpa optimasi dijabarkan pada Tabel 9.

Tabel 9. Perbandingan Performa SVM Tanpa Optimasi dan SVM-GA pada Data Testing

	Performance Metric			
	Accuracy	***Precision***	***Recall***	***F-1 Score***
SVM-RBF *Default*	74,4%	86	61	65
SVM-RBF Parameter Manual	79,2	80	66	70
SVM-RBF GA	81,6	83	73	76

3.2 Pembahasan

Dari proses pengujian tersebut didapatkan kesimpulan jika yang paling mempengaruhi performa algoritma genetika adalah besar *population size* dan *generations*. Namun, penggunaan *population size* yang terlalu besar tidak terlalu memberikan peningkatan performa yang signifikan tetapi menggunakan sumber daya dan waktu eksekusi yang jauh lebih lama sehingga tidak efisien jika

digunakan. Jadi pemilihannya juga tidak bisa terlalu tinggi karena akan mengurangi efisiensi algoritma genetika.

Sementara, nilai *crossover probability* dan *mutation probability* sifatnya tidak pasti karena kebutuhannya tergantung pada masalah dan skala data untuk klasifikasi. Oleh karena itu, pada penelitian ini digunakan *trial* & *error* dalam menemukan nilai terbaik dengan mencoba beberapa nilai. Tujuan dilakukan *crossover* dan *mutation* adalah untuk mencegah solusi yang dihasilkan algoritma genetika terjebak di *local optima*. Namun, khusus untuk *mutation probability* baiknya penggunaan dijaga cukup rendah agar tidak terlalu sering terjadi mutasi dan mencegah algoritma genetika memberikan solusi yang terlalu acak [16].

Perbandingan nilai akurasi yang dihasilkan, ditemukan bahwa menggunakan algoritma genetika untuk mencari nilai parameter SVM-RBF menghasilkan nilai yang sedikit lebih baik daripada menggunakan pemilihan parameter secara manual dengan nilai acak dan menggunakan parameter dengan nilai *default*.

Selain itu, solusi yang dihasilkan dari algoritma genetika sifatnya adalah *global optima*. Pada studi kasus ini, solusi global optima lebih cocok karena tidak terdapat informasi berapa nilai atau range parameter terbaik dari SVM-RBF. *Global optima* dapat menemukan nilai optimal dari keseluruhan input, meskipun terkadang solusinya bukan yang terbaik.

Namun, kelemahan dari model SVM-GA ini adalah solusi yang dihasilkan dapat berbeda-beda meskipun menggunakan konfigurasi parameter GA yang sama pada setiap *training*, sehingga performa yang dihasilkan juga berbeda setiap *training* dieksekusi. Hal ini disebabkan karena algoritma genetika adalah model stokastik yang mengambil sampel populasinya secara acak, sehingga parameter terbaik atau solusi yang dihasilkan pun dapat berbeda karena populasi yang dibangkitkan berbeda pada setiap eksekusi. Pada penelitian ini misalnya, solusi terbaik didapatkan dengan konfigurasi *population size* sebesar 250, *generations* sejumlah 30, serta *crossover probability* dan *mutation probability* masing-masing dengan nilai 0,7 dan 0,2. Dari tiga kali percobaan menghasilkan solusi yang berbeda-beda, pada percobaan pertama menghasilkan akurasi sebesar 81,6%, pada percobaan kedua menghasilkan akurasi 78%, dan pada percobaan ketiga menghasilkan akurasi 80%. Kelemahan lainnya adalah waktu eksekusi dari algoritma genetika terbilang cukup lama dan akan semakin meningkat dengan bertambahnya jumlah *population size* dan *generations* yang digunakan, yang menyebabkan semakin lamanya proses eksekusi pada saat *training*.

4. SIMPULAN

Tahap perancangan model klasifikasi dilakukan dengan mengadopsi tahapan dari *Knowledge Discovery in Database* yakni melalui tahap pengumpulan data, *pre-processing data*, pembobotan kata, klasifikasi (*training* dan *testing*), dan evaluasi model. Dari semua proses yang telah dilakukan sesuai dengan tahapan tersebut, model klasifikasi yang dihasilkan telah mampu melakukan klasifikasi dengan optimal dan membantu meningkatkan performa SVM-RBF untuk melakukan analisis sentimen.

Hasil penelitian menunjukkan bahwa skor akurasi dengan mencari nilai parameter optimal SVM-RBF menggunakan algoritma genetika menghasilkan akurasi yang sedikit lebih baik dibandingkan dengan hasil SVM-RBF tanpa optimasi. Akurasi dari performa parameter yang dioptimasi dengan algoritma genetika adalah 81,6%. Hasil ini lebih baik dari menggunakan parameter default SVM-RBF yang hanya mendapatkan akurasi 74,4% dan pemilihan parameter acak secara manual yang menghasilkan 79,2%. Solusi yang dihasilkan dari algoritma genetika sifatnya juga adalah global optima. Sehingga sangat cocok digunakan untuk studi kasus optimasi parameter kernel karena nilai optimal parameter yang tidak pasti dan tidak diketahui. Tantangan utama dari model SVM-GA solusi yang didapatkan tidak selalu yang paling optimal, karena sifat algoritma genetika adalah model stokastik yang mengambil sampel populasinya secara acak sehingga solusinya kemungkinan besar berbeda setiap kali dieksekusi. Selain itu, waktu eksekusi dari GA juga cukup lama ketika menggunakan jumlah *population size* yang besar sehingga memakan banyak waktu dalam melakukan eksekusi.

DAFTAR PUSTAKA

[1] Liu Bing, *Sentiment Analysis – Mining Opinions, Sentiments, and Emotions*. New York: Cambridge University Press, 2015.

[2] M. Tika Adilah, H. Supendar, R. Ningsih, S. Muryani, and K. Solecha, "Sentiment Analysis of Online Transportation Service using the Naïve Bayes Methods," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-

6596/1641/1/012093.

[3] M. Z. Naf'an, A. A. Bimantara, A. Larasati, E. M. Risondang, and N. A. S. Nugraha, "Sentiment Analysis of Cyberbullying on Instagram User Comments," *J. Data Sci. Its Appl.*, vol. 2, no. 1, pp. 88–98, 2019, doi: 10.21108/jdsa.2019.2.20.

[4] A. S. Neogi, K. A. Garg, R. K. Mishra, and Y. K. Dwivedi, "Sentiment analysis and classification of Indian farmers' protest using twitter data," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100019, 2021, doi: 10.1016/j.jjimei.2021.100019.

[5] W. Kaswidjanti, H. Himawan, and P. D. P. Silitonga, "The accuracy comparison of social media sentiment analysis using lexicon based and support vector machine on souvenir recommendations," *Test Eng. Manag.*, vol. 82, no. 3–4, pp. 3953–3961, 2020.

[6] P. M. Nirmala Dharmapatni and N. L. P. Merawati, "Penerapan Algoritma Support Vector Machine Dalam Sentimen Analisis Terkait Kenaikan Tarif BPJS Kesehatan," *J. Bumigora Inf. Technol.*, vol. 2, no. 2, pp. 105–112, 2020, doi: 10.30812/bite.v2i2.904.

[7] W. A. Luqyana, I. Cholissodin, and R. S. Perdana, "Analisis Sentimen Cyberbullying Pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 4704–4713, 2018.

[8] D. J. Haryanto, L. Muflikhah, and M. A. Fauzi, "Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode Support Vector Machine Dan Query Expansion," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2909–2916, 2018.

[9] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, "Expert Systems with Applications Feature selection and parameter optimization for support vector machines : A new approach based on genetic algorithm with feature chromosomes," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5197–5204, 2011, doi: 10.1016/j.eswa.2010.10.041.

[10] F. Samadzadegan, A. Soleymani, and R. Ali Abbaspour, "Evaluation of genetic algorithms for tuning SVM parameters in multi-class problems," *11th IEEE Int. Symp. Comput. Intell. Informatics, CINTI 2010 - Proc.*, pp. 323–327, 2010, doi: 10.1109/CINTI.2010.5672224.

[11] L. K. Ramasamy, S. Kadry, and S. Lim, "Selection of optimal hyper-parameter values of support vector machine for sentiment analysis tasks using nature-inspired optimization methods," *Bull. Electr. Eng. Informatics*, vol. 10, no. 1, pp. 290–298, 2021, doi: 10.11591/eei.v10i1.2098.

[12] H. Harafani, "Support Vector Machine Parameter Optimization to Improve Liver Disease Estimation with Genetic Algorithm," *SinkrOn*, vol. 4, no. 2, p. 106, 2020, doi: 10.33395/sinkron.v4i2.10524.

[13] D. Whitley, *Computer Science A Genetic Algorithm Tutorial*. 1993.

[14] I. A. Ahmad Sabri, M. Man, W. A. W. Abu Bakar, and A. N. Mohd Rose, "Web Data Extraction Approach for Deep Web using WEIDJ," *Procedia Comput. Sci.*, vol. 163, no. July, pp. 417–426, 2019, doi: 10.1016/j.procs.2019.12.124.

[15] M. P. Simatupang and D. P. Utomo, "Analisa Testimonial Dengan Menggunakan Algoritma Text Mining Dan Term Frequency- Inverse Document Frequence (Tf-Idf) Pada Toko Allmeeart," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 808–814, 2019, doi: 10.30865/komik.v3i1.1697.

[16] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach," *Inf.*, vol. 10, no. 12, 2019, doi: 10.3390/info10120390.

[17] I. Braga, L. P. Do Carmo, C. C. Benatti, and M. C. Monard, "A note on parameter selection for support vector machines," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8266 LNAI, no. PART 2, pp. 233–244, 2013, doi: 10.1007/978-3-642-45111-9_21.

[18] P. Bajpai and M. Kumar, "Genetic algorithm–an approach to solve global optimization problems," *Indian J. Comput. Sci. Eng.*, vol. 1, no. 3, pp. 199–206, 2010.