

KLASIFIKASI EMOSI PADA LIRIK LAGU MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE DAN OPTIMASI PARTICLE SWARM OPTIMIZATION

CLASSIFICATION OF EMOTIONS IN SONG LYRICS USING SUPPORT VECTOR MACHINE ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Wahyudi Hermanto¹, Budhi Irawan^{2,3}, Casi Setianingsih³

^{1,2,3} Universitas Telkom, Bandung

wahyudihermanto@student.telkomuniversity.ac.id¹, budhiirawan@telkomuniversity.ac.id²,
setiacasie@telkomuniversity.ac.id³

Abstrak

Lagu adalah suatu kesatuan musik yang terdiri dari susunan nada dan terdapat lirik di dalamnya. Lirik adalah syair kata-kata yang disuarakan mengiringi melodi. Pada lirik lagu berperan dalam membangun emosi. Emosi pada lagi menjelaskan makna emosional yang melekat pada sebuah lirik lagu. Penelitian ini akan dilakukan klasifikasi emosi pada lirik lagu diawali dengan pengumpulan dataset berupa lirik lagu di website <https://lirik.kapanlagi.com/>, <https://liriklaguindonesia.net/>, dan <http://liriklaguanak.com/> sebagai penyedia lirik lagu. Kemudian dilakukan *preprocessing* data yang terdiri dari *transform cases*, *tokenizing*, *stop removal*, dan *stemming*. Setelah itu dilakukan proses *part of speech (POS) tagging* untuk memberikan label pada kata di dalam teks sesuai dengan kelas kata secara otomatis. Proses memberikan label pada kata apakah itu kata kerja, kata sifat, atau keterangan. Untuk dapat menentukan emosional pada lirik lagu sesuai dengan apa yang kita dengarkan, maka dibutuhkan metode yang tepat dalam melakukan klasifikasi teks. Dalam penelitian Fatimah Wulandini yang berjudul “*Text Classification Using Support Vector Machine for Web Mining Based Spatio Temporal Analysis of the Spread of Tropical Diseases*” bahwa metode SVM menunjukkan hasil akurasi paling baik sebesar 90% pada kasus klasifikasi informasi teks Bahasa Indonesia dibandingkan dengan metode Naive Bayes classifier dan k-Nearest Neighbor classifier dengan akurasi hanya sebesar 80%.

Kata Kunci: *support vector machine, particle swarm optimization, emosi*

Abstract

Song is a musical unity consisting of an arrangement of tones and there are lyrics in it. Lyrics are words that are voiced alongside melodies. Song lyrics play a role in building emotions. Emotions on again explain the emotional meaning inherent in song lyrics. This research will be conducted emotional classification on song lyrics starting with dataset collection in the form of song lyrics on <https://lirik.kapanlagi.com/> website, <https://liriklaguindonesia.net/>, and <http://liriklaguanak.com/> as a provider of song lyrics. Then preprocessing data consists of transform cases, tokenizing, stop removal, and stemming. After that, the part of speech (POS) tagging process is done to label the word in the text according to the word class automatically. The process of labelling a word whether it's a verb, adjective, or attack. To be able to determine the emotional lyrics of the song according to what we listen to, it takes the right method of classifying the text. Fatimah Wulandini's study entitled "Text Classification Using Support Vector Machine for Web Mining Based Spatio Temporal Analysis of the Spread of Tropical Diseases" that SVM method shows the best accuracy of 90% in the case of classification of Indonesian text information compared to Naive Bayes classifier and k-Nearest Neighbor classifier method with only 80% accuracy.

Keywords: Object Detection, CNN, Faster R-CNN

1. Pendahuluan

Setiap orang pernah merasakan emosi. Emosi adalah sebuah perasaan seseorang terhadap lingkungan atau keadaan yang mempengaruhi tingkah laku. Emosi dapat ditunjukkan seperti merasa senang terhadap suatu keadaan, marah pada seseorang, ataupun takut terhadap suatu kejadian. Emosi bersifat subjektif dan temporer yang disebabkan oleh stimulus dari lingkungan atau orang sekitar. Salah satu lingkungan yang dapat merangsang emosi seseorang adalah melalui lagu yang didengarkan. Lagu sudah menjadi salah satu bagian dari kesenangan manusia yang mendengarkannya. Seni nada atau suara dalam urutan, kombinasi dan hubungan temporal yang biasanya diiringi dengan alat musik. Suatu nada itu nantinya akan membentuk suatu irama dan lirik berupa syair atau kata-kata yang mengiringi irama tersebut. Lirik lagu sendiri merupakan salah satu bagian dalam sebuah lagu yang berperan sebagai pembangun emosi. Pemilihan kata-kata yang tepat diperlukan untuk membangun emosi. Lirik pada lagu menunjukkan suasana hati orang yang menciptakannya. Dalam mengekspresikan emosinya, pencipta lagu Menyusun kata-kata untuk dijadikan lirik lagu. Mereka mencoba menggunakan emosi selain untuk mendapatkan *metadata* konvensional seperti (marah, senang, sedih, tenang, takut, dll).

Pada penelitian ini dibuat sebuah sistem yang mampu mengkategorikan lirik lagu berdasarkan topik lagu sebagai data penelitian. Secara garis besar, penelitian ini terdiri dari beberapa tahapan agar mencapai hasil output yang diinginkan yaitu *preprocessing* data, ekstraksi fitur dengan metode pembobotan WIDF dan TF-IDF, optimasi menggunakan *Particle Swarm Optimization*, klasifikasi menggunakan metode *Support Vector Machine*, dan pengujian sistem. *Support Vector Machine* (SVM) merupakan metode klasifikasi yang berakar dari teori pembelajaran statistik yang hasilnya sangat menjanjikan untuk memberikan hasil yang lebih baik dari metode yang lain. Selain itu, SVM juga bekerja dengan baik pada set data dengan dimensi yang tinggi, seperti teks.

Berdasarkan permasalahan diatas maka Tugas Akhir ini akan mengimplementasikan suatu sistem klasifikasi dengan metode *Support Vector Machine* dengan pemilihan fitur *Particle Swarm Optimization*. Dengan metode tersebut diharapkan dapat menentukan klasifikasi emosi apakah sesuai dengan lirik lagu dan mendapatkan akurasi yang tinggi.

2. Dasar Teori

2.1. Text Mining

Penambangan teks menurut Feldman, R. dan Sanger, J., adalah proses pengetahuan intensif yang bekerja dengan dokumen menggunakan beberapa alat analisis. *Text mining* juga dapat mengekstrak informasi yang berguna dari sumber data untuk identifikasi dan eksplorasi pola yang menarik.[1]

2.2. Preprocessing Text

Pada tahap pengolahan kata, karakter yang terhubung harus diurai. Ini dapat dilakukan pada beberapa level yang berbeda. Dokumen dapat dibagi menjadi bab, subbagian, paragraf, kalimat, kata, dan bahkan suku kata atau fonem. Langkah-langkah untuk melakukan preprocess teks adalah sebagai berikut:[2]

1. *Transform Cases*, Dengan menggunakan *Transform Cases* secara otomatis semua huruf pada teks akan diubah menjadi huruf kecil semua atau menjadi huruf kapital semua.[1]
2. *Tokenization*, Tokenisasi terkadang juga disebut sebagai ekstraksi yang merupakan proses mengubah aliran teks menjadi kata-kata unit tunggal. Proses pemotongan kalimat menjadi beberapa potongan kata atau karakter yang disebut dengan istilah token.
3. *Stopword Removal*, Tahapan ini adalah proses pengambilan kata-kata penting dan pembuangan kata-kata yang dianggap tidak penting. Contoh *stopword* pada kamus adalah “ke”, “di”, “dari”, “dan” dan sebagainya.

4. *Stemming, stemming* adalah proses pengubahan kata yang terdapat dalam dokumen menjadi kata dasar dalam aturan tertentu atau bisa juga dianggap sebagai penghapusan kata tambahan.

2.3. Part of Speech Tagger (POS) Tagging

Part -of -speech (POS) tagger adalah memberikan label pada kata di dalam teks sesuai dengan kelas kata secara otomatis. Proses memberikan label pada kata apakah itu kata kerja, kata sifat, atau keterangan. Berikut adalah tabel dari *Pos-Tagging*. [3]

Tabel 1 Label *POS-Tagging*

Label	Keterangan
CD (cardinal numerals)	Bilangan kardinal
CC (coordinate conjunction)	Konjungsi koordinasi
OD (ordinal number)	Bilangan urutan
IN (prepositions)	Preposisi
FW (foreign words)	Kata serapan/kata asing
JJ (adjectives)	Kata sifat
NEG (negations)	Negasi
MD (modal or auxiliaries verbs)	Kata kerja bantu/modal
NN (common nouns)	Kata benda umum, tidak spesifik
PR (common pronouns)	Pengolahan kata benda secara umum
PRP (personal pronouns)	Kata ganti orang
VB (verbs)	Kata kerja
SYM (symbols)	Simbol
SC (subordinate conjunction)	Kata sambung/penghubung
RB (adverbs)	Keterangan waktu

Berikut tabel contoh dari proses *Post Tagging*:

Tabel 2 Proses *Pos-Tagging*

Input	Output
cinta hati maaf salah cinta cinta	cintaNN hatiNN maafNN salahJJ cintaNN cintaNN

2.4. Feature Selection

2.4.1. Weighted Inverse Document Frequency (WIDF)

Proses pemilihan fitur dengan metode pembobotan kata ini dilakukan untuk mendapatkan nilai setiap kata (istilah). Pembobotan yang akan digunakan dalam Tugas Akhir ini adalah WIDF (*Weighted Inverse Document Frequency*), pengembangan pembobotan TF-IDF (*Term Frequency-inverse Document Frequency*). WIDF meringkas semua frekuensi istilah dari satu set teks, dengan kata lain WIDF adalah bentuk

normalisasi dari satu set teks. [4] Pembobotan WIDF juga dapat menghitung semua koleksi dokumen. [8] Di sini persamaan WIDF dirumuskan dalam persamaan berikut.

$$WIDF(d, j) = \frac{TF(d, j)}{\sum_i TF(i, j)} \quad (1)$$

Keterangan:

d = pengumpulan dokumen
 t = kata atau istilah
 i = dokumen terkait
 TF (d, t) = munculnya kata (t) dalam dokumen yang dibagi dengan TF (i, t), yang merupakan jumlah total kata (t) dalam dokumen terkait (i).

2.4.2. TF-IDF (term frequency-inverse document frequency)

TF-IDF, metode ini merupakan metode yang digunakan untuk menentukan seberapa jauh keterhubungan kata terhadap dokumen dengan memberikan bobot di setiap katanya. Metode TF-IDF ini adalah metode yang menggabungkan 2 konsep yaitu frekuensi dari kemunculan suatu kata yang terdapat pada dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut. [5] Berikut persamaan TF-IDF dirumuskan pada persamaan berikut.

$$IDF(word) = \log \left(\frac{N}{df} \right) \quad (2)$$

$$Wtd = tf \times \log \left(\frac{N}{df} \right) \quad (3)$$

Dimana:

d : dokumen ke-d
 t : kata ke-t dari kata kunci
 W : bobot dokumen ke-d terhadap kata ke-t
 tf : banyak kata yang dicari pada sebuah dokumen
 IDF : *Inversed Document Frequency*
 N : jumlah total dokumen
 df : banyak dokumen yang mengandung token/kata

Alur algoritma PSO dimulai dengan sekumpulan partikel dengan solusi potensial untuk masalah yang sedang diselidiki dan kecepatan yang diinisialisasi secara acak dalam ruang pencarian. Setiap iterasi melakukan pencarian posisi optimum dengan memperbarui kecepatan dan posisi partikel. Demikian pula, pada setiap iterasi, nilai fitness untuk setiap posisi partikel ditentukan dengan menggunakan fungsi fitness. Kecepatan setiap partikel diperbarui dengan dua posisi optimal, posisi individu yang optimal dan posisi keseluruhan yang optimal.

2.5. Particle Swarm Optimization (PSO)

Swarm Optimization (PSO) adalah teknik optimasi berbasis populasi yang terinspirasi oleh kawanan burung dan kawanan ikan, awalnya dirancang dan diperkenalkan oleh Kennedy dan Eberhart (1995), dan berbasis iteratif / generasi. *Swarm intelligence system* melakukan penyebaran kecerdasan yang inovatif dalam menyelesaikan masalah optimasi dengan mengambil inspirasi dari contoh fenomena makhluk hidup, seperti fenomena kelompok (*swarm*) pada hewan, dimana setiap kelompok memiliki perilaku individu dalam melakukan tindakan bersama untuk mencapai tujuan yang sama. [6]

2.6. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma klasifikasi dimana vektor yang diinputkan menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) berdimensi tinggi. Konsep kerja dari algoritma *Support Vector Machine* (SVM)

adalah dengan cara mendefinisikan batas yang disebut dengan *hyperplane* yang terletak antara dua kelas yang berbeda dengan ketentuan jarak maksimal dari data yang terdekat. *Hyperplane* dengan nilai yang terbaik berfungsi untuk mendapatkan suatu batasan maksimal antar kelas pada *input space* yang diperoleh dengan cara mengukur dari *margin hyperplane* serta mencari titik maksimalnya. *Margin* merupakan jarak yang terletak di antara *hyperplane* dengan titik terdekat dari masing-masing kelas. Titik terdekat ini yang disebut dengan *support vector*. [8] Untuk menentukan *Hyperplane* terbaik dapat ditentukan dengan ketentuan rumus sebagai berikut:

$$\text{Max } L_D = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n a_i a_j y_i y_j x_i x_j \quad j = 1 \quad (4)$$

Atau

$$\text{Min } L_D = \frac{1}{2} \sum_{j=1}^n a_i a_j y_i y_j x_i x_j - \sum_{i=1}^n a_i \quad (5)$$

Dengan syarat:

$$\sum_{i=0}^n a_i y_i = 0, a_i \geq 0 \quad (2.8)$$

Pada umumnya Support Vector Machine hanya dapat melakukan proses klasifikasi data ke dalam dua kelas (klasifikasi biner). [9] SVM secara umum memiliki 4 kernel, yaitu:

Linear Kernel

$$K(x_i, x_j) = x_i^T x_j \quad (6)$$

Polynomial Kernel dengan derajat d

$$K(x_i, x_j) = \exp(Y x_i^T x_j + 1)^d, Y > 0 \quad (7)$$

Radial Basis Function Kernel (RBF)

$$K(x_i, x_j) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2Y^2}\right), Y > 0 \quad (8)$$

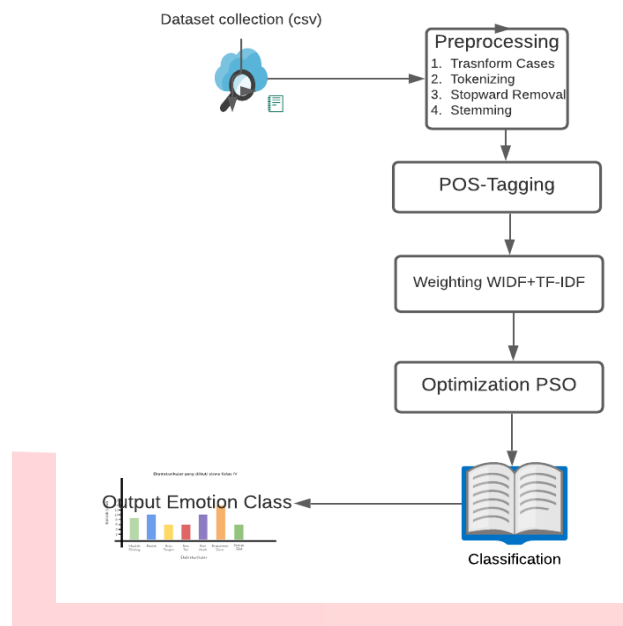
Sigmoid Kernel

$$K(x_i, x_j) = \tanh(Y x_i^T x_j + r) \quad (9)$$

3. Perancangan Sistem

3.1. Desain Sistem

Sistem yang akan dibangun pada tugas akhir adalah sistem yang mampu mendeteksi kelas emosi pada lirik lagu. Lirik lagu yang akan digunakan pada tugas akhir ini adalah bahasa Indonesia. Kelas yang akan dideteksi oleh sistem ini adalah sedih, bahagia, cinta, dan marah. Alur kerja sistem ini sebagai berikut: pengumpulan dataset berupa teks, *preprocessing*, *POS-Tagging*, pembobotan kata, seleksi fitur dan klasifikasi menggunakan algoritma *support vector machine*.



Gambar 1 Alur Sistem

Pengumpulan data sampai mendapatkan suatu kesimpulan berupa hasil klasifikasi. Berikut adalah beberapa proses yang terjadi pada gambar 3.1:

1. Pengumpulan Data: Proses ini merupakan pengumpulan data berupa lirik lagu dan nanti akan disatukan dalam file CSV.
2. *Preprocessing*: Pada tahapan ini akan dilakukan proses mengubah huruf besar menjadi huruf kecil, *tokenizing*, *stopword removal*, dan *stemming*.
3. Ekstraksi fitur: Pada tahapan ini dilakukan ekstraksi fitur menggunakan algoritma WIDF dan akan diberikan bobot nilai pada setiap kata.
4. Optimasi: Pada tahapan ini dilakukan seleksi fitur menggunakan algoritma *particle swarm optimization* berguna untuk memilih fitur-fitur yang relevan dan meningkatkan efektifitas dan efisiensi kinerja dari algoritma klasifikasi.
5. Klasifikasi: Data yang telah diberikan bobot nilai akan diklasifikasikan menggunakan algoritma *support vector machine*.
6. Kesimpulan klasifikasi: Data yang telah diklasifikasikan kemudian akan dikelompokkan sesuai dengan sub kelasnya.

3.2. Preprocessing

Preprocessing adalah tahapan pembersihan data berupa teks lirik lagu berbahasa Indonesia. Tujuan dari *preprocessing* data adalah membersihkan data lirik lagu untuk mengurangi teks yang tidak ada makna agar sistem bisa mengklasifikasi emosi pada lirik lagu. Tahapan dalam proses *preprocessing* sebagai berikut:

3.2.1. Case Folding

Dengan menggunakan *Case Folding* secara otomatis semua huruf pada teks akan diubah menjadi huruf kecil semua atau menjadi huruf kapital semua dapat dilihat ada table

Tabel 3 Proses *Case Folding*

Input	Output
Jika masih ada Cinta di hatimu Maka maafkanlah Segala kesalahan Cintailah cinta	jika masih ada cinta di hatimu maka maafkanlah segala kesalahan cintailah cinta

3.2.2. Stemming

Stemming merupakan suatu proses transformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya dengan aturan-aturan tertentu, atau dapat dianggap sebagai pembuangan imbuhan kata, atau dianggap sebagai pembuangan imbuhan kata dapat dilihat pada table dibawah

Tabel 4 Proses *Stemming*

Input	Output
jika masih ada cinta di hatimu maka maafkanlah segala kesalahan cintailah cinta	jika masih ada cinta di hati maka maaf segala salah cinta cinta

3.2.3. Stop Removal

Stopword Removal yaitu menghilangkan kata pada dokumen lirik lagu yang dianggap tidak memiliki arti atau tidak mengandung kata emosi atau juga tidak memperkuat kata emosi. Pembuangan kata dilakukan dengan proses mencocokkan kata pada kamus *stopword removal*. Berikut contoh proses *Stopword Removal* pada table dibawah

Tabel 5 Proses *Stop Removal*

Input	Output
jika masih ada cinta di hati maka maaf segala salah cinta cinta	Cinta hati maaf salah cinta cinta

3.2.4. Tokenizing

Proses *tokenizing* dilakukan untuk memecahkan dokumen lirik lagu menjadi term berdasarkan spasi. Masukan pada tahapan ini adalah hasil keluaran dari proses *Stop Removal*. Tabel dibawah menunjukan proses dari *tokenizing*.

Tabel 6 Proses *Tokenizing*

Input	Output
Cinta hati maaf salah cinta cinta	['cinta','hati','maaf','salah','cinta','cinta']

3.3. POS-Tagging

Tahapan selanjutnya adalah *POS-Tagging*. Pada tugas akhir ini akan dilakukan proses *POS-Tagging* dengan memanfaatkan alat yang sudah ada. *POS-Tagging* merupakan proses pemberian penanda POS atau kelas sintatik pada tiap kata. Tahapan *POS-Tagging* akan dilakukan pelabelan kelas sebuah kata pada suatu kalimat lirik lagu yang sudah dilakukan proses *preprocessing*. Berikut contoh proses *Pos-Tagging*

3.4. Seleksi Fitur

Berikut contoh dokumen yang akan diproses dalam seleksi fitur menggunakan metode WIDF dan TF-IDF.

Tabel 7 Contoh Dokumen

Dokumen	Lirik	Label
D1	Risau hati cinta 7aying pernah letih hati cinta 7aying segenap milik rasa hati kamu sayang	Cinta

D2	Engkau 8aying hati aku cinta segenap hati diri aku 8aying ka	Cinta
D3	Rasa marah diri aku pernah sengaja temu kamu kesal	Marah
D4	Engkau senang riang pagi cerah matahari sinar	Senang
D5	Aku sedih hati niat ubah aku 8aying rasa hati aku kesal kamu	Sedih

Tahapan pertama pada metode WIDF adalah melihat banyak tingkat kemunculan sebuah kata atau *term frequency* dari sebuah dokumen. Misal nya kata “cinta” muncul 3 kali, berarti $tf=3$, diketahui kata “cinta” muncul pada dokumen 1 dan dokumen 2. Dibawah ini contoh perhitungan WIDF pada kata “cinta” pada dokumen 1.

$$Tf = 3; df = 3; N = 5$$

$$IDF = \log \left(\frac{5}{3} \right) = 0.221$$

$$\begin{aligned} WIDF(d, j) &= \frac{TF(d, j)}{\sum_{i \in D} TF(i, j)} \times \log \left(\frac{N}{df} \right) \\ &= \frac{2}{3} \times \log \left(\frac{5}{3} \right) \\ &= 0.666 \times 0.221 \\ &= 0.117 \end{aligned}$$

Bobot nilai kata “cinta” yang terdapat pada dokumen sebesar 0.117

Berikut table hasil perhitungan seluruh dokumen

Contoh data yang sudah melalui tahap *preprocessing* yang akan dilakukan pembobotan kata TF-IDF. Label emosi pada D1 dan D2 adalah cinta, D3 marah, D4 senang dan D5 sedih. Tahapan pertama pada metode pembobotan kata TF-IDF ini adalah melihat banyak tingkat kemunculan sebuah kata atau *term frequency* dari sebuah kalimat atau dokumen. Misalkan kata “cinta” hanya terdapat di dokumen D1 dan D2 saja. Di bawah ini adalah contoh perhitungan TF-IDF sederhana untuk kata “cinta”.

$$Tf = 2; df = 3; N = 5$$

$$IDF = \log \left(\frac{5}{2} \right) = 0.397$$

$$Wtd = 2 \times 0.397 = 0.794$$

Nilai kata “cinta” pada dokumen 1 dan 2 adalah 0.794

Tabel 8 WIDF

Term	W				
	D1	D2	D3	D4	D5
cinta	0.397	0.698	0.000	0.000	0.000
risau	0.698	0.000	0.000	0.000	0.000
hati	0.221	0.397	0.000	0.000	0.397
sayang	0.221	0.397	0.000	0.000	0.698
pernah	0.698	0.000	0.698	0.000	0.000
letih	0.698	0.000	0.000	0.000	0.000
segenap	0.698	0.000	0.000	0.000	0.000
milik	0.698	0.000	0.000	0.000	0.000

Term	W				
	D1	D2	D3	D4	D5
rasa	0.698	0.000	0.698	0.000	0.698
kamu	0.698	0.698	0.698	0.000	0.698
engkau	0.000	0.698	0.000	0.698	0.000
aku	0.000	0.397	0.698	0.000	0.221
diri	0.000	0.698	0.698	0.000	0.000
marah	0.000	0.000	0.698	0.000	0.000
sengaja	0.000	0.000	0.698	0.000	0.000
temu	0.000	0.000	0.698	0.000	0.000
kesal	0.000	0.000	0.698	0.000	0.698
senang	0.000	0.000	0.000	0.698	0.000
riang	0.000	0.000	0.000	0.698	0.000
pagi	0.000	0.000	0.000	0.698	0.000
cerah	0.000	0.000	0.000	0.698	0.000
matahari	0.000	0.000	0.000	0.698	0.000
sinar	0.000	0.000	0.000	0.698	0.000
sedih	0.000	0.000	0.000	0.000	0.698
niat	0.000	0.000	0.000	0.000	0.698
ubah	0.000	0.000	0.000	0.000	0.698

3.5. Particle Swarm Optimization (PSO)

Pada tahapan seleksi fitur menggunakan algoritma *particle swarm optimization* (PSO) akan menggunakan data dari perhitungan pembobotan kata *weighted inverse document frequency* (WIDF) dari tabel 3.8. Untuk menyelesaikan perhitungan menggunakan algoritma *particle swarm optimization* (PSO) adalah sebagai berikut:

1. Langkah pertama adalah mengambil kata yang sering muncul pada seluruh dokumen. Berikut tabel 3.11 menunjukkan kumpulan dari kata-kata yang telah diambil dari hasil WIDF.

Tabel 9 Kumpulan Kata Sering muncul

Dokumen	Sayang	Rasa	Cinta	Pernah	Engkau	Hati	Kesal	Aku	Kamu	Diri
D1	3	1	2	1	0	3	0	0	1	0
D2	2	0	1	0	1	2	0	2	0	1
D3	0	1	0	1	0	0	1	1	1	1
D4	0	0	0	0	1	0	0	0	0	0
D5	1	1	0	0	0	2	1	3	1	0

2. Tahap selanjutnya adalah menentukan bilangan random dengan matriks $M \times N$, dimana M =adalah jumlah fitur sebanyak 10 fitur, dan N =adalah jumlah partikel sebanyak 5 partikel. Isi dari matriks $M \times N$ adalah sejumlah bilangan random.

- Menghasilkan bilangan random (rand 1 dan rand 2) dengan menggunakan excel untuk setiap kata.
- Bilangan rand 2 dijadikan sebagai nilai *velocity*/tabel kecepatan partikel.

Berikut adalah tabel rand1 dan rand2:

Tabel 10 Tabel Rand 1

Rand1

D1	0.85181 7	0.41984 9	0.67459 1	0.45305 3	0.26121 2	0.54103 0	0.63626 2	0.77955 2	0.43566 8	0.73580 4
D2	0.30788 4	0.04667 1	0.54013 3	0.31857 0	0.29677 9	0.45530 1	0.17078 9	0.62526 6	0.07125 3	0.97847 4
D3	0.53064 8	0.31056 0	0.98216 4	0.03282 5	0.67061 9	0.49705 6	0.90068 0	0.85067 2	0.07228 7	0.88874 9
D4	0.92256 3	0.03846 3	0.12493 1	0.08609 6	0.81712 3	0.70902 0	0.93614 5	0.17901 4	0.96572 2	0.79599 9
D5	0.21098 3	0.60228 1	0.12841 5	0.21937 4	0.98604 1	0.53820 4	0.25252 3	0.21379 6	0.15056 8	0.33274 1

Tabel 11 Tabel Rand 2

Rand2										
D1	0,04412 8	0,68149 2	0,76275 6	0,42067 5	0,70296 9	0,95407 7	0,00421 0	0,88177 8	0,21990 3	0,23231 1
D2	0,04848 0	0,17560 5	0,54319 0	0,71834 0	0,53231 4	0,57612 3	0,07962 3	0,42055 8	0,13373 3	0,52738 2
D3	0,15769 1	0,19145 5	0,74224 1	0,53610 1	0,31485 7	0,17001 1	0,21766 8	0,69416 9	0,38399 3	0,62876 4
D4	0,48189 4	0,81113 3	0,07472 3	0,60756 8	0,48558 1	0,45039 8	0,43313 2	0,09029 0	0,52391 0	0,99781 9
D5	0,78775 9	0,41380 0	0,87333 1	0,36387 3	0,28627 6	0,98390 3	0,23096 2	0,23617 2	0,69890 8	0,70264 5

3. Tahapan selanjutnya yaitu mencari nilai sigmoid menggunakan tabel rand 2 dengan menggunakan persamaan (2.5) sebagai berikut:

$$S(v_{pd}) = \frac{1}{(1 + e^{-v_{pd}})} \quad (7)$$

Tabel 12 Tabel Sigmoid

Sigmoid										
D1	0.51103 0	0.66407 2	0.68195 2	0.60364 5	0.66884 9	0.72193 4	0.50105 2	0.70719 1	0.55455 1	0.55781 8
D2	0.51211 8	0.54378 9	0.63255 4	0.67224 1	0.63002 3	0.64017 5	0.51989 5	0.60361 7	0.53338 4	0.62887 2
D3	0.59341 8	0.54771 8	0.67748 6	0.63090 5	0.57807 0	0.54240 1	0.55420 3	0.72991 1	0.59483 6	0.65220 9
D4	0.61819 5	0.69235 1	0.51867 2	0.64738 6	0.61906 5	0.61073 4	0.60662 1	0.52255 7	0.62806 2	0.73063 0
D5	0.68735 0	0.60199 9	0.70543 8	0.58997 8	0.57108 4	0.72788 2	0.55748 5	0.55877 0	0.66794 6	0.66877 4

4. Selanjutnya membandingkan nilai sigmoid dengan nilai rand1. Apabila nilai rand1 lebih kecil dibandingkan nilai sigmoid maka nilai sigmoid akan dipilih atau diberi nilai 1, jika nilai rand1 besar dari sigmoid maka tidak dipilih atau diberi nilai 0.

Tabel 13 Hasil perbandingan

Dokumen	Sayang	Rasa	Cinta	Pernah	Engkau	Hati	Kesal	Aku	Kamu	Diri
D1	0	1	1	1	1	1	0	0	1	0

D2	1	1	1	1	1	1	1	0	1	0
D3	1	1	0	1	0	1	0	0	1	0
D4	0	1	1	1	0	0	0	1	0	0
D5	1	0	1	1	0	1	1	1	1	1

5. Tahapan selanjutnya mencari nilai pBest pada setiap dokumen menggunakan rumus

$$F(x) = 10x(x_1 - 1)^2 + 20x(x_2 - 2)^2 + 30x(x_3 - 3)^2.$$

$$F(D1) = 10x(0 - 1)^2 + 20x(1 - 2)^2 + 30x(1 - 3)^2 + 40x(1 - 4)^2 + 50x(1 - 5)^2 + 60x(1 - 6)^2 + 70x(1 - 7)^2 + 80x(1 - 8)^2 + 90x(1 - 9)^2 + 100x(1 - 10)^2 = 28.650$$

$$F(D2) = 10x(1 - 1)^2 + 20x(1 - 2)^2 + 30x(1 - 3)^2 + 40x(1 - 4)^2 + 50x(1 - 5)^2 + 60x(1 - 6)^2 + 70x(1 - 7)^2 + 80x(0 - 8)^2 + 90x(1 - 9)^2 + 100x(1 - 10)^2 = 24.300$$

$$F(D3) = 10x(1 - 1)^2 + 20x(1 - 2)^2 + 30x(0 - 3)^2 + 40x(1 - 4)^2 + 50x(0 - 5)^2 + 60x(1 - 6)^2 + 70x(0 - 7)^2 + 80x(0 - 8)^2 + 90x(1 - 9)^2 + 100x(0 - 10)^2 = 27.710$$

$$F(D4) = 10x(0 - 1)^2 + 20x(1 - 2)^2 + 30x(1 - 3)^2 + 40x(1 - 4)^2 + 50x(0 - 5)^2 + 60x(0 - 6)^2 + 70x(0 - 7)^2 + 80x(1 - 8)^2 + 90x(0 - 9)^2 + 100x(0 - 10)^2 = 28.560$$

$$F(D5) = 10x(1 - 1)^2 + 20x(0 - 2)^2 + 30x(1 - 3)^2 + 40x(1 - 4)^2 + 50x(0 - 5)^2 + 60x(1 - 6)^2 + 70x(1 - 7)^2 + 80x(1 - 8)^2 + 90x(1 - 9)^2 + 100x(1 - 10)^2 = 23.61$$

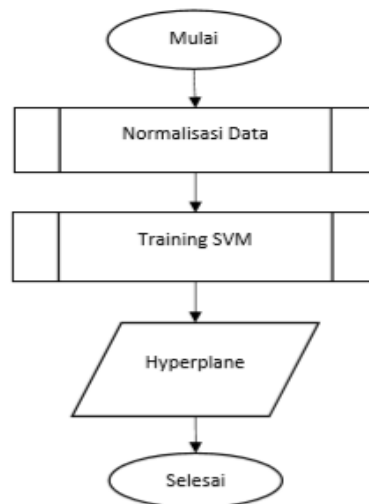
Pada perhitungan diatas untuk mencari nilai Pbest maka dapat disimpulkan bahwa nilai Pbest terbaik terletak pada dokumen 1(D1) sebesar 28.650. Maka fitur yang terpilih dari dokumen 1 adalah rasa, cinta, pernah, engkau, hati, kamu.

Berdasarkan dari hasil optimasi diatas maka bentuk tabel TF-IDF nya adalah:

Tabel 14 Hasil Optimasi WIDF

Term	W				
	D1	D2	D3	D4	D5
rasa	0.698	0.000	0.698	0.000	0.698
cinta	0.397	0.698	0.000	0.000	0.000
pernah	0.698	0.000	0.698	0.000	0.000
engkau	0.000	0.698	0.000	0.698	0.000
Hati	0.221	0.397	0.000	0.000	0.397
kamu	0.698	0.698	0.698	0.000	0.698

3.6. Algoritma Support Vector Machine (SVM)



Gambar 2 Alur SVM

Pada gambar 3.5 pada tahapan proses *training*, hasil dari proses TF-IDF sebagai masukan, kemudian data di normalisasi. Disini data yang akan dimasukan adalah D1, D2, D3 dari hasil optimasi menggunakan PSO di atas. Kemudian dilakukan proses *training* SVM dan hasil akhirnya adalah *hyperplane*.

Tabel 15 Hasil TF-IDF

Term	W		
	D1	D2	D3
rasa	0.698	0.000	0.698
cinta	0.397	0.698	0.000
pernah	0.698	0.000	0.698
engkau	0.000	0.698	0.000
Hati	0.221	0.397	0.000
kamu	0.698	0.698	0.698

a. Normalisasi

Normalisasi adalah bertujuan untuk menstandarisasi nilai bobot ke interval 0 sampai 1. Karena hasil dari proses TF-IDF dengan D1, D2 dan D3 memenuhi syarat interval maka pada tahapan ini tidak dilakukan normalisasi data.

b. Analisis *Training* SVM

Selanjutnya harus dilakukan pada data latih dari fitur dimensi lama. Sehingga mendapatkan set data dengan fitur baru berdimensi tinggi. Dengan kernel $K(x, y) = \exp(-\gamma \|x - x_i\|^2)$, $\gamma > 0$, dan dataset berdimensi $N \times 1$, maka akan didapatkan dimensi baru $N \times N$, dimana N adalah banyaknya data. Pada fungsi kernel akan digunakan $\gamma = 0,5$. Dimana k adalah matrix kernel. Pada tabel berikut adalah pengurangan $x - x_i$.

Tabel 16 Hasil $x - x_i$

$X_1 - x_1$	$x_1 - x_2$	$X_1 - x_3$
0	0.698	0
0	-0.301	0.397
0	0.698	0
0	-0.698	0
0	-0.176	0.221

0	0	0
X2-x1	X2-x2	X2-x3
-0.698	0	-0.698
0.301	0	0.698
-0.698	0	-0.698
0.698	0	0.698
0.176	0	0.397
0	0	0
X3-x1	X3-x2	X3-x3
0	0.698	0
-0.397	-0.698	0
0	0.698	0
0	-0.698	0
-0.221	-0.397	0
0	0	0

Setelah didapatkan hasil pengurangan $x-x_i$. Maka selanjutnya dilakukan perhitungan untuk mendapatkan panjang vector. Panjang vector didapatkan dengan persamaan berikut:

$$||x - x_i|| = \sqrt{x^2 + y^2 + z^2}$$

Tabel 17 Tabel Perhitungan Panjang Vektor

$ x_1 - x_i $	$\sqrt{x^2 + y^2 + z^2}$	Hasil
$ x_1 - x_1 $	$\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2}$	0
$ x_1 - x_2 $	$\sqrt{0.698^2 + -0.301^2 + 0.698^2 + -0.698^2 + -0.176^2 + 0^2}$	1.076
$ x_1 - x_3 $	$\sqrt{0^2 + 0.397^2 + 0^2 + 0^2 + 0.221^2 + 0^2}$	0.4543
$ x_2 - x_1 $	$\sqrt{-0.698^2 + 0.301^2 + -0.698^2 + 0.698^2 + 0.176^2 + 0^2}$	1.261
$ x_2 - x_2 $	$\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2}$	0
$ x_2 - x_3 $	$\sqrt{-0.698^2 + 0.698^2 + -0.698^2 + 0.698^2 + 0.397^2 + 0^2}$	1.450
$ x_3 - x_1 $	$\sqrt{0^2 + -0.397^2 + 0^2 + 0^2 + -0.221^2 + 0^2}$	0.452
$ x_3 - x_2 $	$\sqrt{0.698^2 + -0.698^2 + 0.698^2 + -0.698^2 + -0.397^2 + 0^2}$	1.450
$ x_3 - x_3 $	$\sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2}$	0

Selanjutnya untuk perhitungan nilai $\exp(-\gamma ||x - x_i||^2)$ sebagai berikut:

$$\exp(x, y) = \exp(-\gamma ||x - x_i||^2), \gamma = 0.5$$

$$\begin{aligned}
 K(1,1) &= \exp(-\gamma ||x - x_i||^2) & K(1,2) &= \exp(-\gamma ||x - x_i||^2) \\
 &= \exp(-0.5 (0)^2) & &= \exp(-0.5 (1.076)^2) \\
 &= \exp 0 & &= \exp(-0.5 (1.157)) \\
 &= 1 & &= \exp(-0.5785) \\
 & & &= 0.5607
 \end{aligned}$$

$$\begin{aligned}
 K(1,3) &= \exp(-\gamma ||x - x_i||^2) \\
 &= \exp(-0.5 (0.4543)^2) \\
 &= \exp(-0.1031) \\
 &= 0.9020
 \end{aligned}$$

$$K(2,1) = \exp(-\gamma ||x - x_i||^2) \quad K(2,2) = \exp(-\gamma ||x - x_i||^2)$$

$$\begin{aligned}
 &= \exp(-0.5 (1.261)^2) &= \exp(-0.5 (0)^2) \\
 &= \exp(-0.5 (1.590)^2) &= \exp 0 \\
 &= \exp(-0.7950) &= 1 \\
 &= 0.4515 \\
 K(2,3) &= \exp(-\gamma ||x - x_i||^2) \\
 &= \exp(-0.5 (1.450)^2) \\
 &= \exp(-1.051) \\
 &= 0.349 \\
 K(3,1) &= \exp(-\gamma ||x - x_i||^2) &K(3,2) &= \exp(-\gamma ||x - x_i||^2) \\
 &= \exp(-0.5 (0.452)^2) &&= \exp(-0.5 (1.450)^2) \\
 &= \exp(-0.1021) &&= \exp(-1.051) \\
 &= 0.9029 &&= 0.349 \\
 &&K(3,3) &= \exp(-\gamma ||x - x_i||^2) \\
 &= \exp(-0.5 (0)^2) \\
 &= \exp 0 \\
 &= 1
 \end{aligned}$$

Berikut hasil perhitungan kernel RBF secara keseluruhan.

Tabel 18 Hasil Perhitungan Kernel RBF Keseluruhan

K K	1	2	3
1	1	0.5607	0.9020
2	0.4515	1	0.349
3	0.9029	0.349	1

Setelah dilakukan perhitungan maka matriks kernel K sebagai berikut:

$$K = \begin{bmatrix} K(1,1) & K(2,1) & K(3,1) \\ K(1,2) & K(2,2) & K(3,2) \\ K(1,3) & K(2,3) & K(3,3) \end{bmatrix}$$

$$K = \begin{bmatrix} 1 & 0.5607 & 0.9020 \\ 0.4515 & 1 & 0.4515 \\ 0.9029 & 0.349 & 1 \end{bmatrix}$$

Kemudian tahap selanjutnya adalah melakukan terhadap y. Nilai y merupakan nilai dari label kelas yang telah diberikan. Berikut nilai y pada tabel berikut.

Tabel 19 Label kelas y_i

y_1	y_2	y_3
1	-1	-1

Selanjutnya melakukan perhitungan nilai $y_i y_j$ sebanyak N data. Berikut hasil dari perhitungan tersebut.

Tabel 20 Perhitungan Nilai $y_i y_j$

$y_1 y_1$	$y_1 y_2$	$y_1 y_3$
-----------	-----------	-----------

1	-1	-1
y_2y_1	y_2y_2	y_2y_3
-1	1	1
y_3y_1	y_3y_2	y_3y_3
-1	1	1

Sehingga matriks yang didapatkan sebagai berikut:

$$y_i y_j = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

Matriks K diatas setiap elemennya merupakan hasil dari $\exp(-\gamma \|x - x_i\|^2)$ yang akan berkorelasi dengan $a_i a_j$. Dengan menggunakan kernel K sebagai pengganti *dot product* $x_i x_j$ dengan persamaan dualitas Lagrange Multiplier. Berikut persamaan Large Multiplier yang akan digunakan.

$$\text{Min } Ld = \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i \cdot x_j - \sum_{i=1}^N a_i \quad (6)$$

Dengan syarat 1:

$$\begin{aligned} \text{Min } Ld &= \frac{1}{2} \sum_{i,j} a_i a_j \begin{bmatrix} 1 & 0.5607 & 0.9020 \\ 0.4515 & 1 & 0.4515 \\ 0.9029 & 0.349 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} - a_1 + a_2 + a_3 \\ &= \frac{1}{2} \sum_{i,j} a_i a_j \begin{bmatrix} -0.4627 & 0.4627 & 0.4627 \\ -1 & 1 & 1 \\ -0.4461 & 0.4461 & 0.4461 \end{bmatrix} - a_1 + a_2 + a_3 \\ &= \frac{1}{2} \left(-0.4627a_1^2 - a_1a_2 - a_1a_3 + 0.4627a_2a_1 + a_2^2 \right. \\ &\quad \left. + 0.4461a_2a_3 + 0.4627a_3a_1 + a_3a_2 + 0.4461a_3^2 \right) - a_1 + a_2 + a_3 \\ &= \frac{1}{2} \left(-0.4627a_1^2 - (a_1a_2 - 0.4627a_2a_1) - (a_1a_3 - 0.4627a_3a_1) \right. \\ &\quad \left. + a_2^2 + (0.4461a_2a_3 + a_3a_2) + 0.4461a_3^2 \right) - a_1 + a_2 + a_3 \\ &= \frac{1}{2} \left(-0.4627a_1^2 - 0.5373a_1a_2 - 0.5373a_1a_3 + a_2^2 \right. \\ &\quad \left. + 1.4461a_2a_3 + 0.4461a_3^2 \right) - a_1 + a_2 + a_3 \\ &= \left(0.23135a_1^2 - 0.2686a_1a_2 - 0.2686a_1a_3 + 0.5a_2^2 \right. \\ &\quad \left. + 0.72305a_2a_3 + 0.22305a_3^2 \right) - a_1 + a_2 + a_3 \end{aligned}$$

Misalkan C = 1 (default), maka $0 \leq a_1, a_2, a_3 \leq 1$.

Setelah hasil perhitungan *large multiplier* sementara untuk mencapai optimum dari fungsi *Lagrange* (Ld) sehingga didapatkan persamaan sebagai berikut:

1. $\frac{\partial L}{\partial a_1} = 0.23135a_1 - 0.537a_2 - 0.537a_3 = 0$
2. $\frac{\partial L}{\partial a_2} = 0.2686a_1 + 0.5a_2 + 0.72305a_3 = 0$
3. $\frac{\partial L}{\partial a_3} = 0.2686a_1 + 0.72305a_2 + 0.22305a_3 = 0$
4. $a_1 = a_2 + a_3$

Persamaan 4 disubstitusikan terhadap persamaan 1,2 dan 3.

1. $\frac{\partial L}{\partial a_1} = 0.23135a_2 + 0.23135a_3 - 0.537a_2 - 0.537a_3 = 0$
 $= 0.30565a_2 + 0.3057a_3 = 0$
2. $\frac{\partial L}{\partial a_2} = 0.2686a_2 + 0.2686a_3 + 0.5a_2 + 0.72305a_3 =$
 $= 0.7686a_2 + 0.99165a_3 = 0$

$$\begin{aligned}
 3. \quad \frac{\partial L}{\partial a_1} &= 0.2686a_2 + 0.2686a_3 + 0.72305a_2 + 0.22305a_3 = \\
 &= 0.99165a_2 + 0.49165a_3 = 0
 \end{aligned}$$

Selanjutnya mencari nilai a_2 dan a_3 memanfaatkan hasil substitusi diatas, sehingga didapatkan nilai $a_1 = 1$, $a_2 = 0$, $a_3 = 1$. Karena $a_2 = 0$ maka bukan termasuk *support vector*. Hanya $a_1 = 1$ dan $a_3 = 1$ maka itulah *support vector* yang didapatkan dari proses *training*. *Hyperplane* sudah dapat ditentukan dengan 2 titik yaitu titik (1,1) dan (3,3).

4. Hasil Pengujian dan Analisis

Tujuan pengujian sistem ini adalah untuk melihat apakah sistem sudah berjalan dengan baik dalam bentuk desktop dengan menguji fitur yang terkandung dalam sistem. Pengujian sistem menggunakan model yang telah dilatih menggunakan dataset yang telah dibuat.

Akurasi adalah salah satu parameter yang dapat digunakan dalam pengujian apakah suatu model dapat bekerja secara optimal atau tidak. Akurasi dapat menentukan apakah sistem dapat mengklasifikasikan lirik lagu apakah mereka diberi label atau tidak. Untuk mendapatkan akurasi yang optimal, dibutuhkan parameter optimal yang sangat mempengaruhi akurasi. Berikut adalah parameter yang mempengaruhi keakuratan model yang dibuat:

1. Jumlah data pengujian
2. Nilai kernel, gamma, dan C
3. Nilai iterasi, c1, c2, dan berat
4. Dataset disusun menjadi lima ukuran yang berfungsi sebagai data pengujian: 50%, 40%, 45%, 30%, dan 20%.

Untuk spesifikasi dataset yang digunakan dalam tes dapat dilihat pada tabel di bawah ini:

Tabel 21 Spesifikasi Data Latih

Label Emosi	Jumlah Lirik Lagu
Cinta	25
Marah	25
Senang	25
Sedih	25

4.1. Pengujian *Preprocessing*

Ada beberapa tahap *preprocessing*: case folding & stem, stop word removal, dan tokenizing.

4.2. Pengujian POS-Tagging

Pengujian POS-Tagging adalah tahap memeriksa label subjek, predikat, objek dalam kalimat yang telah melewati tahap *praproses* sebelumnya.

4.3. Pengujian WIDF dan TF-IDF

Setelah proses *preprocessing*, dataset kemudian masuk ke dalam proses pembobotan kata dengan metode pembobotan kata WIDF dan TF-IDF. Sedangkan proses pembobotan kata selesai dataset akan langsung digunakan untuk proses klasifikasi SVM. Tes ini dilakukan dengan membuat berbagai model klasifikasi dan deteksi berdasarkan parameter di atas. Tujuannya adalah untuk menemukan nilai akurasi tertinggi dari masing-masing tes parameter di atas sehingga sistem dapat bekerja secara optimal.

4.4. Implementasi *Support Vector Machine* (SVM)

Implementasi algoritma SVM dilakukan dengan menggunakan *library* SVC dan kernel rbf. Pada pengujian kinerja sistem dengan pertisi data, data akan dibagi menjadi data uji dan data latih.

Data uji adalah data yang diproses untuk menguji algoritma yang digunakan. Sedangkan data latih adalah data yang digunakan untuk mengetahui performa algoritma dalam melakukan klasifikasi. Berikut tabel pembagian untuk pengujian partisi data:

Tabel 22 Partisi data

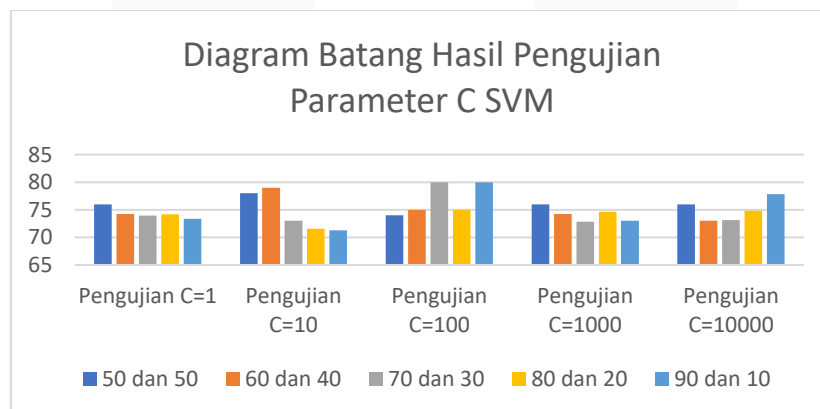
Pengujian	Data Latih (%)	Data Uji (%)
1	50	50%
2	60	40%
3	70	30%
4	80	20%
5	90	10%

Untuk pengujian SVM nilai banyak data latih, nilai C, gamma akan di *input* dengan nilai sebagai berikut: banyak data latih (50%, 40%, 30%, 20%, dan 10%), nilai C (1, 10, 100, 1000), dan nilai gamma (10^{-3} , 10^{-4}). Hasil yang ditampilkan dari pengujian partisi data tersebut adalah:

Tabel 23 Pengujian SVM

Nilai C	Data Latih	Data Uji	Akurasi (%)	Presisi (%)	Recall (%)
Pengujian 1					
1	50	50	76	62	75
	60	40	74.25	61.75	61.25
	70	30	73.94	61	66.5
	80	20	74.20	62	67
	90	10	73.36	61.25	70
Pengujian 2					
10	50	50	78	61	75
	60	40	79	62	75
	70	30	73	60	75
	80	20	71.58	61	75
	90	10	71.26	62	75
Pengujian 3					
100	50	50	74	62	66.25
	60	40	75	61.75	66
	70	30	80	61	66.5
	80	20	75	62	66.5

Nilai C	Data Latih	Data Uji	Akurasi (%)	Presisi (%)	Recall (%)
	90	10	80	61.25	62.5
Pengujian 4					
1000	50	50	76	59	75
	60	40	74.25	62	75
	70	30	72.83	62	75
	80	20	74.625	62	75
	90	10	73	58	75
Pengujian 5					
10000	50	50	76	60	75
	60	40	73	61.25	75
	70	30	73.11	61.75	75
	80	20	74.83	57.25	75
	90	10	77.86	57.25	75



Gambar 3 Grafik Pengujian SVM

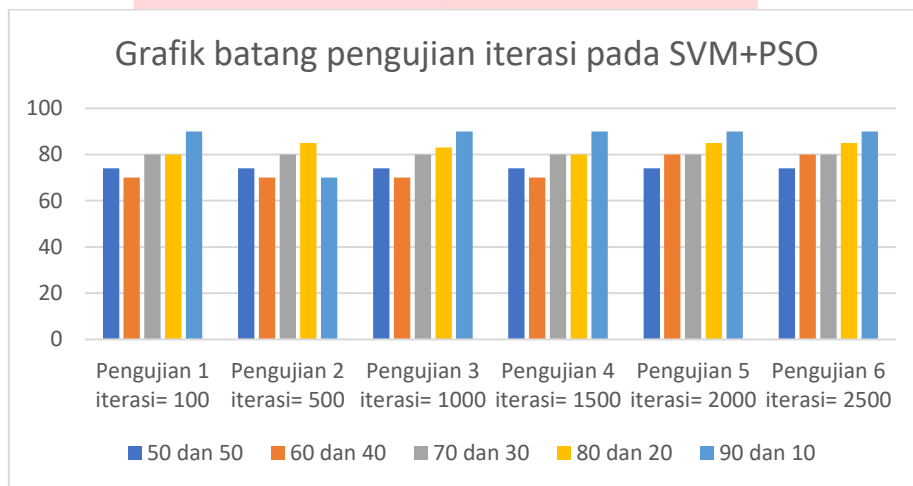
4.5. Implementasi Particle Swarm Optimization (PSO) dan Support Vector Machine (SVM)

Implementasi seleksi fitur PSO *library* pyswarm. Hasil dari implementasi PSO dapat dilihat pada lampiran. Untuk hasil pengujian yang dilakukan adalah mengatur nilai iterasi (100, 500, 1000, 1500, 2000, 500), $C1=0.6$, $C2=0.8$, $Weight=0.9$, dan menggunakan pengujian partisi data uji sebanyak 50%, 40%, 30%, 20%, dan 10%. Maka hasil pengujian berdasarkan parameter tersebut dapat dilihat pada tabel dibawah ini:

Tabel 24 Pengujian nilai iterasi dengan SVM+PSO

Pengujian	Data Train (%)	Data Testing (%)	Accuracy (%)	Precision (%)	Recall (%)
Parameter {iterasi=100, default (c1=06, c2=0.8, w=0.9)}					
Pengujian 1	50	50	74	62	75
	60	40	70	62	75
	70	30	80	61	75
	80	20	80	61	75
	90	10	90	58	75
Parameter {iterasi =500, default (c1=06, c2=0.8, w=0.9)}					
Pengujian 2	50	50	74	63	75
	60	40	70	62	75
	70	30	80	64	75
	8	20	85	58	75
	90%	10	70	64	75
Parameter {iterasi =1000, default (c1=06, c2=0.8, w=0.9)}					
Pengujian 3	50	50	74	62	75
	60	40	70	62	75
	70	30	80	61	75
	80	20	83	61	75
	90	10	90	67	75
Parameter {iterasi =1500, default (c1=06, c2=0.8, w=0.9)}					
Pengujian 4	50	50	74	62	75
	60	40	70	60	75
	70	30	80	61	75
	80	20	80	63	75
	90	10	90	67	75
Parameter {iterasi =2000, default (c1=06, c2=0.8, w=0.9)}					
Pengujian 5	50	50	74	62	75
	60	40	80	62	75
	70	30	80	68	75

Pengujian	Data Train (%)	Data Testing (%)	Accuracy (%)	Precision (%)	Recall (%)
	80	20	85	61	75
	90	10	90	67	75
Parameter { iterasi =2500, default (c1=0.6, c2=0.8, w=0.9) }					
Pengujian 6	50	50	74	62	75
	60	40	80	62	75
	70	30	80	61	75
	80	20	85	68	75
	90	10	90	67	75



Gambar 4 Grafik Pengujian SVM+PSO

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan implementasi sistem dan pengujian yang telah dilakukan, berikut adalah kesimpulan yang dapat ditarik penulis:

1. Sistem klasifikasi emosi berdasarkan lirik lagu berbahasa Indonesia sudah berjalan dengan baik dengan hasil uji *alpha* atau fungsionalitas sebesar 100%
2. Sistem klasifikasi *Support Vector Machine* (SVM) dengan atau tanpa optimasi PSO memiliki tingkat performa akurasi, presisi dan *recall* yang hampir sama.
3. Tingkat akurasi tertinggi yang didapatkan oleh algoritma *Support Vector Machine* (SVM) tanpa Optimasi PSO dan dengan Optimasi PSO sebesar 90%. Optimasi PSO dapat meningkatkan tingkat akurasi dari klasifikasi, pada dataset dengan ukuran distribusi data 90:10, tingkat akurasi meningkat sebesar 10%

5.2. Saran

Adapun saran dari penulis untuk penelitian tugas akhir dengan tema seperti ini di masa yang akan datang, serta diharapkan dapat meningkatkan kinerja dari penggunaannya sebagai berikut:

1. Menambahkan dataset
2. Menambahkan kamus POS – *Tagging* dan *Stopword removal*

6. Referensi

- [1] A. Deolika, K. Kusriani, and E. T. Luthfi, "Analisis Pembobotan Kata Pada Klasifikasi Text Mining," *J. Teknol. Inf.*, vol. 3, no. 2, p. 179, 2019, doi: 10.36294/jurti.v3i2.1077.
- [2] A. T. J. H, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining," *Inform. UPGRIS*, vol. 1, pp. 1–9, 2015.
- [3] R. Muzzammel, *Intelligent Technologies and Applications*, vol. 932. 2019.
- [4] S. E. Pratama, W. Darmalaksana, D. Sa'adillah Maylawati, H. Sugilar, T. Mantoro, and M. A. Ramdhani, "Weighted inverse document frequency and vector space model for hadith search engine," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 1004–1014, 2020, doi: 10.11591/ijeecs.v18.i2.pp1004-1014.
- [5] M. Jyoti and D. Ela, "Semantic Web Improved with the Weighted IDF Feature," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 2, pp. 164–173, 2015, doi: 10.14569/ijacsa.2015.060224.
- [6] F. Yigit and O. K. Baykan, "A new feature selection method for text categorization based on information gain and particle swarm optimization," *CCIS 2014 - Proc. 2014 IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst.*, pp. 523–529, 2014, doi: 10.1109/CCIS.2014.7175792.
- [7] Y. Lu, M. Liang, Z. Ye, and L. Cao, "Improved particle swarm optimization algorithm and its application in text feature selection," *Appl. Soft Comput. J.*, vol. 35, pp. 629–636, 2015, doi: 10.1016/j.asoc.2015.07.005.
- [8] S. Herrero-Lopez, *Multiclass support vector machine*, no. 2. NVIDIA Corporation and Wenmei W. Hwu, 2011.
- [9] H. Murfi, "6.2 Radial Basis Function Networks."