KLASIFIKASI EMOSI PADA LIRIK LAGU MENNGUNAKAN ALGORITMA NAÏVE BAYES DAN PARTICLE SWARM OPTIMIZATION

CLASSIFICATION OF EMOTIONS ON SONG LYRICS USING NAÏVE BAYES ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Gerry Samhari Ramadhan[1], Budhi Irawan[2], Casi Setianingsih[3]

[1,2,3] Universitas Telkom, Bandung

gerrysamhari@student.telkomuniversity.ac.id, budhiirawan@telkomuniversity.ac.id, setiacasie@telkomuniversity.ac.id

Abstrak

Lagu adalah sebuah kesatuan suara yang berisikan nada dan terdapat lirik di dalamnya. Dalam sebuah lagu dapat mengandung berbagai macam emosi, emosi pada lagu dapat timbul karena perpaduan antara lirik dengan nada yang dapat membuat bunyi yang sangat indah dan harmoni. Dalam Tugas Akhir ini akan dilakukan proses klasifikasi emosi pada lirik lagu. Penelitian ini diawali dengan pengumpulan dataset berupa lirik lagu dari website https://lirik.kapanlagi.com/, https://liriklaguindonesia.net/, dan http://liriklaguanak.com/ sebagai penyedia lirik lagu. Kemudian dilakukan *preprocessing* data yang terdiri dari *case folding*, *tokenizing, stop removal*, dan *stemming*. Setelah itu dilakukan proses *part of speecs* (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 dan metode yang akan digunakan adalah metode *Naïve Bayes Classifier* dan *Partcile Swarm Optimization*, sebagai metode yang digunakan dalam melakukan klasifikasi teks. Pada beberapa penelitian menyebutkan bahwa metode *Naïve Bayes Classifier* dengan optimasi *Particle Swarm Optimization* menunjukkan hasil yang baik pada kasus klasifikasi informasi teks Bahasa Indonesia dengan performa akurasi sebesar 90% - 96% menggunakan nilai *Inertia Weight* 1.0

Kata kunci: *Naïve Bayes Classifier, Particle Swarm Optimization*, Emosi

Abstract

A song is a unity of sound that contains a tone and lyrics in it. A song can contain a variety of emotions, emotions in the song can arise because of the combination of lyrics with tones that can make a very beautiful sound and harmony. This Final Task will be carried out the process of classifying emotions in the lyrics of the song. This research began with the collection of datasets in the form of song lyric from https://lirik,kapanlagi.com, https://liriklaguindonesia.net and http://liriklaguanak.com/. Then preprocessing data consisting of case folding, tokenizing, stop removal, and stemming. After that, the part of speecs (POS) tagging process will label the word in the text according to the word class automatically. To be able to determine the emotional lyrics of the song according to what we listen to, it takes the right method and the method to be used is the Naïve Bayes Classifier and Partcile Swarm Optimization methods, as methods used in performing text classification. In some studies, mentioned that the Naïve Bayes Classifier method shows good results in the case of classification of Indonesian text information with an accuracy of 90% - 96% using inertia weight score 1.0

Keywords: Naïve Bayes Classifier, Particle Swarm Optimization, Emotion

1. Pendahuluan

Saat ini banyak sekali jenis-jenis musik / genre musik baru yang bermuculan seiring dengan perkembangan zaman. Banyak sekali musisi-musisi baru yang mengeluarkan inovasi dalam menciptakan sebuah lagu, inovasi tersebut dibuat dengan tujuan untuk menyegarkan jenis musik yang sudah ada sehingga para pendengar tidak bosan dengan lagu yang itu-itu saja. Karena semakin banyak lagu yang tercipta, semakin banyak pula tempat penyedia lagu sehingga pendengar dapat lebih mudah untuk mendengarkan lagu yang mereka suka sesuai dengan emosi yang sedang mereka alami sekarang. Emosi merupakan suatu perasaan yang dimiliki manusia yang dapat timbul akibat adanya reaksi terhadap suatu hal yang terjadi di sekitarnya. emosi yang timbul bisa bermacam-macam seperti marah, senang, sedih, cinta, kecewa, terharu dan malu. setiap orang memiliki caranya masing-masing dalam meluapkan emosinya, salah satunya adalah dengan lagu, lagu dapat menjadi suatu media untuk seseorang menyalurkan emosinya dengan mendngarkan lagu yang mereka suka.

Dalam melakukan pencarian terhadap suatu lagu, judul lagu merupakan acuan dalam menentukan apakah lagu tersebut sesuai dengan apa yang ingin dicari pendengar sesuai dengan emosi yang sedang mereka alami, namun terkadang judul lagu tidak sesuai dengan isi lagu dan penyampaian emosi yang ingin disampaikan oleh seorang musisi bisa berbeda dengan apa yang diterima oleh pendengar. Untuk mengetahui maksud dan emosi yang terkandung pada lagu tersebut pendengar harus mendengarkan seluruh isi lagu tersebut. Lirik yang terdapat pada sebuah lagu memiliki arti yang bermacam-macam karena lirik pada lagu memiliki nilai emosional, jadi setiap pendengar menerima makna dari lirik tersebut secara berbeda-beda. Dalam hal ini, sebuah lagu dapat mengisahkan tentang “kebahagiaan” namun disampaikan dengan nada yang “pelan” sehingga lagu tersebut terdengar seperti lagu “kesedihan”. Oleh sebab itu, untuk menghindari kesalahan dalam penafsiran suatu lagu diperlukan proses klasifikasi secara otomatis pada lirik lagu sehingga tidak perlu untuk mencermati lirik dari suatu lagu satu persatu

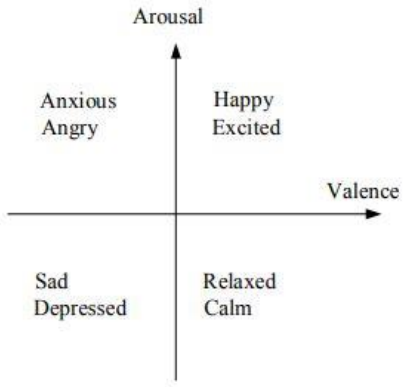
Tugas akhir ini bertujuan untuk menciptakan proses klasifikasi lirik lagu yang dapat menyelesaikan permasalahan tersebut, dengan menggunakan metode *Naïve Bayes* dengan optimasi *Particle Swarm Optimization* (PSO) yang dapat mengklasifikasikan lirik lagu berdasarkan emosi yang tekandung di dalam lirik-lirik lagu.

2. Dasar Teori

2.1. Emosi

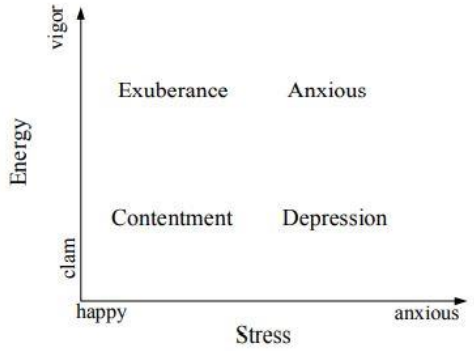
Emosi adalah fenomena dalam kehidupan. Dengan emosi, manusia mengkomunikasikan apa yang dia rasakan untuk ditanggapi secara serius agar mendekat pada sikap dan tindakan cinta kasih atau menjauh pada sikap dan tindakan benci agresif atau bahkan sekedar memberi sinyal-sinyal tertentu pada dirinya dan orang lain di sekitarnya[1].

Beberapa tokoh memodelkan emosi kedalam bentuk dimensi. Ada 2 model emosi yang ditentukan oleh dimensi yaitu model emosi Russel dan model emosi Thayer.



Gambar 2.1 Model Emosi Russel

Empat kuadran melambangkan empat jenis emosi. Kuadran pertama mewakili emosi bahagia dan emosi gembira, dan kuadran kedua mewakili emosi gelisah dan emosi marah, dan kuadran ketiga mewakili emosi sedih dan emosi tertekan, dan kuadran keempat mewakili emosi santai dan emosi tenang[2][3].



Gambar 2.2 Model Emosi Thayer

Model emosi Thayer Seperti yang ditunjukkan di atas, sumbu energi mencerminkan orang-orang vitalitas secara fisiologis, dari kerang hingga kekuatan. Sumbu stres mencerminkan proses orang dari bahagia menjadi cemas dalam psikologi. Ini membagi emosi musik menjadi empat kategori: kegembiraan, kecemasan, kepuasan dan depresi. Di satu sisi, model klasifikasi emosi ini memiliki beberapa kekurangan. Ini memampatkan emosi kompleks manusia ke model dimensi, dan dengan demikian akan menyebabkan hilangnya informasi musik sampai tingkat tertentu. Di sisi lain, masih ada beberapa keuntungan darinya. Pertama, model ini lebih cocok untuk pengenalan emosi karena karakteristik emosinya. Kedua, model menggunakan ide dimensi untuk mendeskripsikan emosi musik, dan mudah untuk menetapkannya[2].

2.2. Text Preprocessing

Dalam penelitian ini diterapkan *text processing* untuk data yang akan digunakan dalam penentuan emosi pada lirik lagu. Dimana data yang kita proses akan kita ambil informasi yang terkandung di dalamnya guna memudahkan kita dalam mengelola data. Berikut ini beberapa tahapan dalam *prepocessing* data[4].

1. *Case Folding*

Beberapa teks dalam sebuah dokumen memiliki huruf kapital, dengan *case folding* ini huruf kapital diubah menjadi huruf kecil, dan hanya huruf ‘a’ sampai ‘z’ yang akan diambil sedangkan karakter lain selain huruf dihilangkan.

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.

3. *Stopward Removal*

Tahapan ini adalah proses pengambilan kata-kata penting dan pembuangan kata-kata yang dianggap tidak penting. Contoh *stopward* pada kamus adalah *"ke", "di", "dari", "dan"* dan sebagainya.

4. *Tokenizing*

Tokenizing 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

2.3 Part of Speech (POS) Tagging

POS - *Tagging* adalah upaya untuk memberikan label pada setiap kata yang ada di dalam teks sesuai dengan katanya dan juga kemungkinan fitur morfologinya. *Pos tag* memberikan informasi mengenai definisi dan konteks kata[5]. Proses memberikan label pada kata apakah itu kata kerja, kata sifat, atau keterangan

Tabel 2.1 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 auxiliary's 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

2.4 Seleksi Fitur

Proses seleksi fitur dengan metode pembobotan kata ini dilakukan untuk mendapatkan nilai dari kata (*term*). Pembobotan yang akan digunakan pada Tugas Akhir ini adalah *Weighted Inverse Document Frequency* (WIDF) dan *Term Frequency Invers Document Frequency* (TF-IDF), WIDF merupakan pengembangan dari pembobotan TF-IDF. WIDF menjumlahkan semua *term frequency*

dari kumpulan teks, dengan kata lain WIDF merupakan bentuk normalisasi dari kumpulan teks. Pembobotan WIDF juga dapat menghitung semua koleksi dokumen. Berikut persamaan WIDF dirumuskan pada persamaan berikut ini [6] .

$$WIDF(d,j) = \frac{\text{TF(d,j)}}{\sum \text{i}\in\text{D TF(i,j)}} \qquad (2.1)$$

Keterangan:

∑TF = Jumlah *term* pada semua dokumen

Sedangkan pada Pembobotan 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[7]. Berikut persamaan TF-IDF dirumuskan pada persamaan berikut.

$$IDF(word) = \log\left(\frac{\text{N}}{\text{df}}\right) \qquad (2.2)$$

$$Wtd = \text{tf} \times \log\left(\frac{\text{N}}{\text{df}}\right) \qquad (2.3)$$

Dimana:

d : dokumen ke-d

t : kata ke-t dari kata kunci

W : bobot dukumen 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

IDF (*word*) adalah nilai IDF dari setiap kata yang akan dicari, td adalah jumlah dari keseluruhan dokumen yang ada, dan df adalah jumlah kemunculan kata pada semua dokumen

2.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) adalah salah satu teknik dasar dari *swarm intelligence system* untuk menyelesaikan masalah optimasi dalam pencarian ruang sebagai suatu solusi. (PSO) pertama kali diusulkan oleh James Kennedy dan Eberhart (1995), dirancang untuk mensimulasikan kebiasaan gerakan dari sekelompok burung. *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[8][9].

Algoritma PSO berhasil digunakan dalam berbagai domain seperti analisis daya, pengelompokan teks, bioinformatika, jaringan sensor nirkabel, pemilihan fitur dan pengelompokan data. PSO digunakan untuk menemukan subset fitur maksimum dengan menemukan campuran fitur terbaik saat mereka terbang dalam area masalah dari kumpulan data yang disiapkan. PSO sangat mudah diimplementasikan, memiliki parameter yang relatif lebih sedikit dan tidak ada asumsi matematis tentang ruang pencarian. Oleh karena itu, PSO banyak digunakan untuk pemilihan fitur

dalam proses klasifikasi. Dalam PSO, solusi diabstraksikan sebagai partikel (titik) tanpa masa dan volume dan meluas ke ruang dimensi N. Posisi partikel I dinyatakan sebagai vektor xi = (x1, x2, …, xN), dan kecepatannya adalah vektor vi = (v1, v1, …, vN). Setiap partikel mengetahui posisi terbaik yang ditemukan sejauh ini (*pbest*) dan posisi terbaik (*gbest*) yang ditemukan oleh gerombolan sejauh ini (*gbest*) juga merupakan nilai *pbest* (s) terbaik. Partikel mencari solusi optimal menggunakan informasi dari *pbest* dan *gbest*[10].

$$Vi + 1 = w \times vi + c1 \times r1 \times (pbesti - xi) + c2 \times r2 \times (gbesti - xi) \quad (2.4)$$

$$xi + 1 = xi + vi + 1 \quad (2.5)$$

Keterangan:

I =1, 2, ..., N

w = Berat inersia menentukan sejauh mana kecepatan arus partikel i

v = Nilai kecepatan partikel i

r1 dan r2= Bilangan random bernilai 0 sampai 1

c1 dan c2 = Koefisien percepatan

Langkah-langkah perhitungan menggunakan metode *Particle Swarm Optimization* sebagai berikut:

1. Melakukan inisialisai data yang akan dijadikan sebagai partikel
2. Menghitung nilai *fitness* masing-masing partikel dan menentukan *pbest* dan *gbest*. Nilai *fitness* awal dari masing-masing *pbest* sama dengan nilai *fitness* posisi awal partikel dan menghitung nilai *velocity*
3. Menghitung nilai *sigmoid* yang akan digunakan sebagai *activation function*. *Sigmoid* akan menerima angka tunggal dan mengubah nilai x menjadi sebuah nilai yang memiliki range mulai dari 0 sampai 1. Berikut fungsi dari sigmoid.

$$S\,(vpd) = \frac{1}{(1+e^{-v\,pd})} \quad (2.6)$$

2.6 Klasifikasi *Naïve Bayes*

Metode *Naïve Bayes* adalah metode yang tidak memiliki aturan, *Naïve Bayes* menggunakan cabang matematika yang dikenal sebagai teori probabilitas untuk menemukan kemungkinan klasifikasi yang paling mungkin, dengan cara melihat frekuensi tiap klasifikasi pada data training. Klasifikasi *Naïve Bayes* adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. Klasifikasi *bayesian* didasarkan pada teorema Bayes, diambil dari nama seorang ahli matematika yang juga menteri Prebysterian Inggris, Thomas Bayes (1702-1706)[11].

Bayes rule digunakan untuk menghitung probabilitas suatu class. Algoritma *Naive Bayes* memberikan suatu cara mengkombinasikan peluang terdahulu dengan syarat kemungkinan menjadi sebuah formula yang dapat digunakan untuk menghitung peluang dari tiap kemungkinan yang terjadi. Bentuk umum dari teorema bayes seperti dibawah ini:

$$P(H|X)\frac{P(X|H)P(H)}{P(X)} \quad (2.6)$$

Dimana:

X: Data dengan class yang belum diketahui

H: Hipotesis data X merupakan suatu class spesifik.

P(H|X) : Probabilitas hipotesis H berdasar kondisi X (*posteriori probability*)

P(H): Probabilitas hipotesis H (*prior probability*)

P(X|H): Probabilitas X berdasar kondisi pada hipotesis H

P(X): Probabilitas dari X

Naïve bayes adalah penyederhanaan metode bayes. Teorema bayes disederhanakan menjadi:

$P(\mathrm{H}|\mathrm{X}) = P(\mathrm{X}|\mathrm{H})\mathrm{P}$ (2.7)

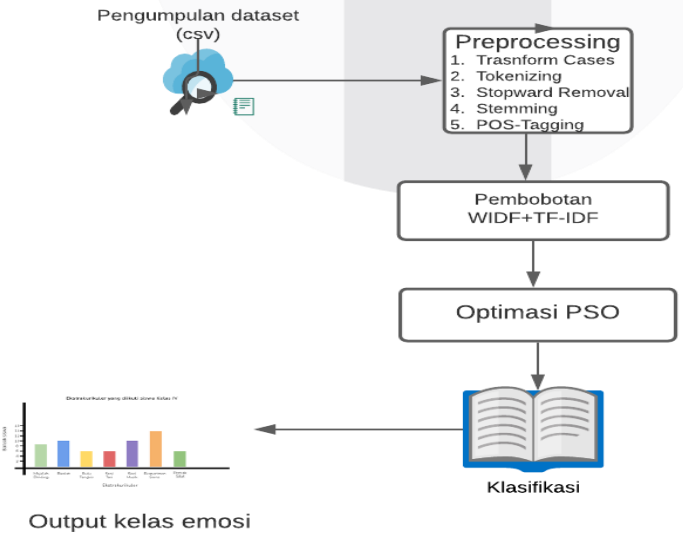
Bayes rule diterapkan untuk menghitung posterior dan probabilitas dari data sebelumnya. Dalam analisis bayesian, klasifikasi akhir dihasilkan dengan menggabungkan kedua sumber informasi (prior dan posterior) untuk menghasilkan probabilitas menggunakan aturan bayes[11].

Multinomial Naive Bayes (MNB) adalah jenis dari *Naïve Bayes* sering menggunakan metode pembelajaran parameter yang disebut *Frequency Estimate* (FE), yang memperkirakan probabilitas kata dengan menghitung frekuensi yang sesuai dari data. Keuntungan utama FE adalah mudah diimplementasikan, sering memberikan kinerja prediksi yang masuk akal, dan efisien. Karena biasanya biaya untuk memperoleh dokumen berlabel tinggi dan dokumen tak berlabel berlimpah

3 Perancangan Sistem

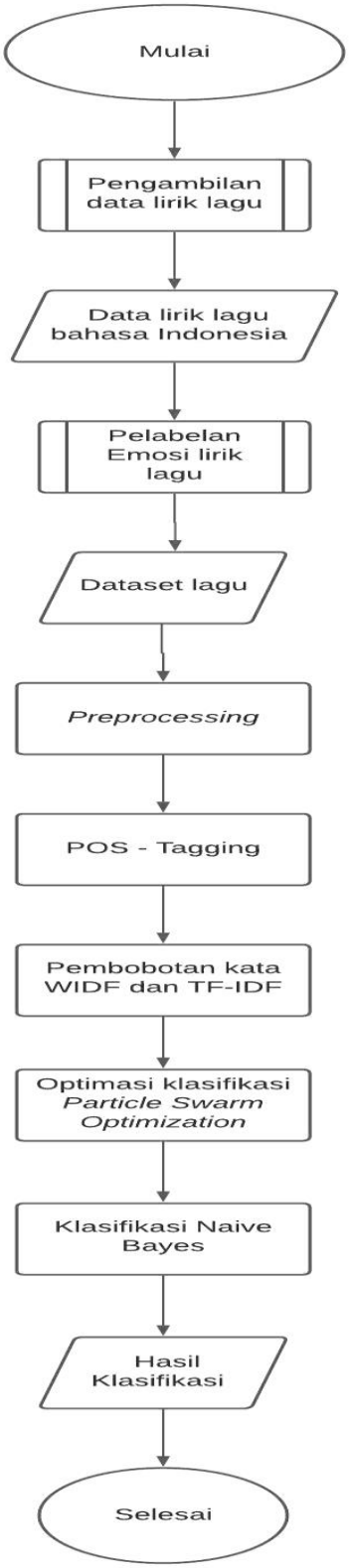
3.1 Desain Sistem

Sistem yang akan dibangun pada Tugas Akhir ini akan membuat sistem yang dapat mendeteksi emosi pada lirik lagu berbahasa Indonesia. Emosi yang akan di deteksi ada 4 kelas yaitu cinta, marah, senang dan sedih. Data lirik lagu yang sudah dikumpulkan dari berbagai situs penyedia lirik lagu nantinya bisa dijadikan dataset yang kemudian akan diproses dengan beberapa tahap, yaitu *preprocessing*, POS-*Tagging*, pembobotan kata, optimasi menggunakan algoritma *Particle Swarm Optimization* dan pengklasifikasian dengan menggunakan algoritma *Naïve Bayes*



Gambar 3.1 Gambaran Desain Sistem

3.2 Rancangan Sistem



Gambar 3.2 Diagram Alur Sistem

Pada gambar 3.2 merupakan diagram alir dari sistem di tugas akhir ini. Dimulai dari pengambilan data lirik lagu berupa lirik lagu berbahasa Indonesia dari *website* penyedia lirik lagu. Data lirik lagu nantinya dilakukan proses pelabelan emosi secara manual. Pelabelan kata dilakukan dengan mencocokan kata-kata pada lirik lagu dengan *rules* yang berisi kata-kata yang mengandung emosi. *Rules* yang dibuat akan divalidasi oleh Balai Bahasa nantinya. Tujuan validasi data adalah

untuk mengecek kinerja pada sistem. Tahap selanjutnya dilakukan *prepocessing* data untuk mendapatkan data bersih. Kemudian ke tahap POS-*Tagging*, pembobotan kata WIDF dan TF-IDF, klasifikasi menggunakan algoritma *Naïve Bayes*, dan terakhir adalah optimasi klasifikasi menggunakan algoritma *Particle Swarm Optimization*

4 Hasil Pengujian dan Analisis

4.1 Pengujian Distribusi Data

Pada pengujian distribusi 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 4.1 Skenario uji distribusi data

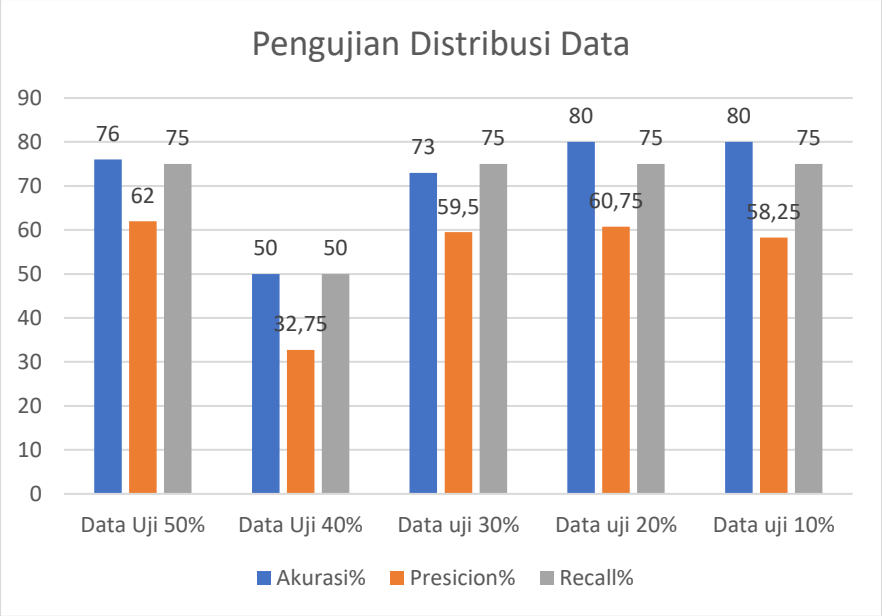
Pengujian	Data Latih (%)	Data Uji (%)
1	50	50
2	60	40
3	70	30
4	80	20
5	90	10

Untuk hasil pengujian distribusi data ditampilkan dengan hasil *confusion matrix*, sehingga dari pengujian ini didapatkan nilai *Precision, Recall*, dan *Accuracy*.

Tabel 4.2 Pengujian Naïve Bayes tanpa optimasi terhadap distribusi dataset

Pengujian	Data Uji (%)	*Accuracy* (%)	*Presicion (%)*	*Recall(%)*
1	50	76	62	75
2	40	50	32.75	50
3	30	73	59.5	75
4 (*)	20	80	60.75	75
5	10	80	58.25	75

Pada tabel 4.10 diatas menunjukkan hasil perubahan nilai akurasi terhadap jumlah data uji. Data uji dengan ukuran 20% dan 10% memiliki nilai *accuracy* lebih tinggi dibandingkan ukuran data uji lainnya, hal ini dapat terjadi karena semakin banyak data latih maka model performa model dalam memprediksi akan semakin meningkat



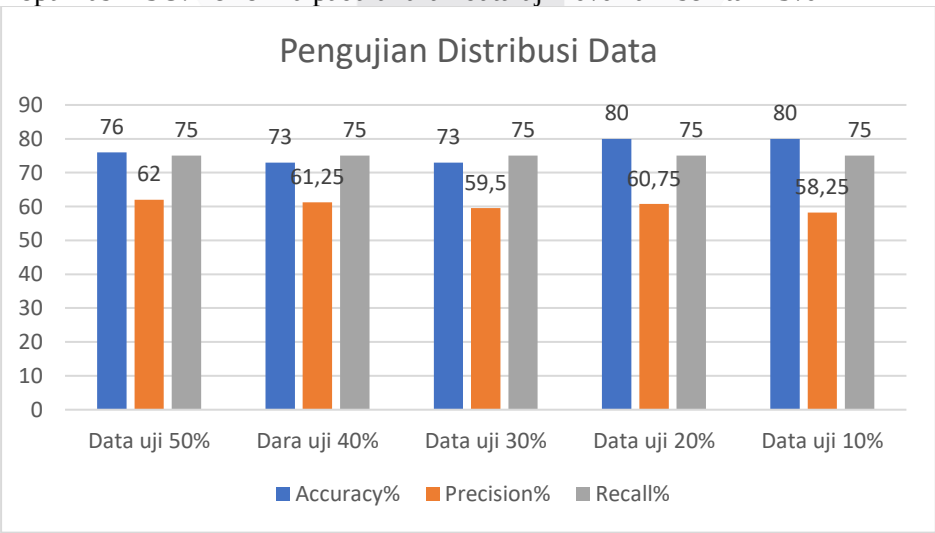
Gambar 4.1 Grafik batang hasil dari pengujian distribusi data

Pengujian distribusi data selanjutnya adalah pengujian dengan model klasifikasi Naïve Bayes yang telah dioptimasi menggunakan PSO

Tabel 4.3 Pengujian Naïve Bayes dengan optimasi PSO terhadap distribusi dataset

Pengujian	Data uji (%)	*Accuracy (%)*	*Presicion (%)*	*Recall (%)*
1	50	76	62	75
2	40	73	61.25	75
3	30	73	59.5	75
4	20	80	60.75	75
5	10	80	58.25	75

Pada tabel 4.7 diatas menunjukkan hasil perubahan nilai akurasi terhadap jumlah data uji. Tapi pada data uji dengan ukuran 40% memiliki nilai *accuracy* lebih tinggi dibandingkan dengan sebelum optimasi PSO. Performa pada ukuran data uji 40% naik sekitar 23%



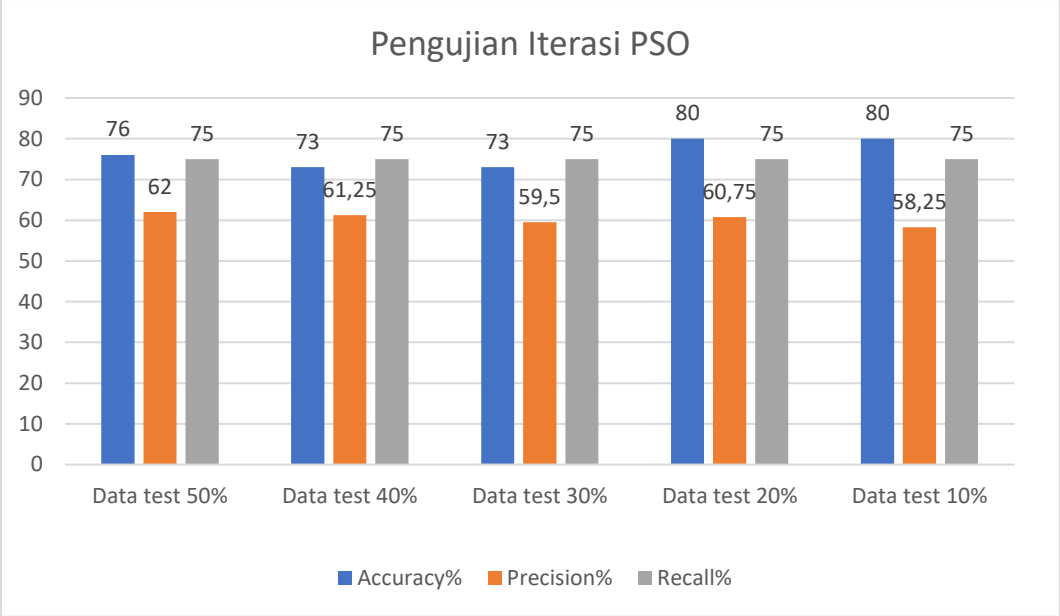
Gambar 4.2 hasil pengujian distribusi data dengan PSO

4.1 Pengujian Iterasi

Pengujian iterasi dilakukan saat optimasi PSO dengan parameter *default* (C1=0.8, C2=0.6, W=0.9), berikut adalah tabel pengujian Iterasi

Tabel 4.4 Pengujian Iterasi PSO terhadap performa *Naïve Bayes*

Iteration	*Data test* (%)	*Accuracy* (%)	*Presicion (%)*	*Recall(%)*
1	50	76	62	75
1	40	73	61.25	75
1	30	73	59.5	75
1	20	80	60.75	75
1	10	80	58.25	75
10	50	76	62	75
10	40	73	61.25	75
10	30	73	59.5	75
10	20	80	60.75	75
10	10	80	58.25	75
100	50	76	62	75
100	40	73	61.25	75
100	30	73	59.5	75
100	20	80	60.75	75
100	10	80	58.25	75
1000	50	76	62	75
1000	40	73	61.25	75
1000	30	73	59.5	75
1000	20	80	60.75	75
1000	10	80	58.25	75



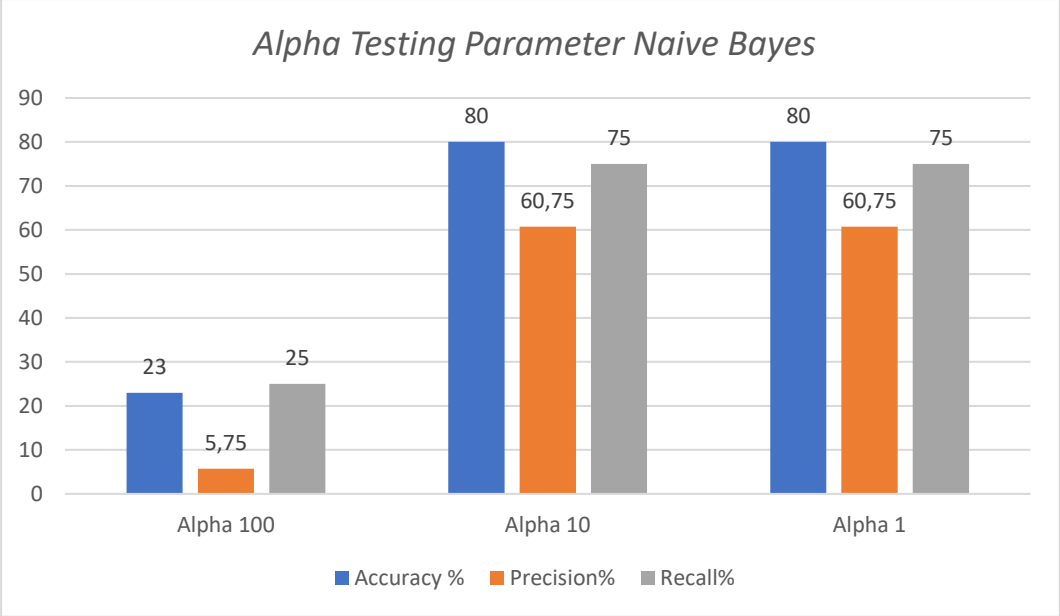
Gambar 4.3 Hasil Pengujian Iterasi PSO

4.2 Pengujian Nilai *Alpha*

Pengujian nilai *alpha* dilakukan untuk mencari nilai *smoothing* terbaik. Berikut adalah tabel pengujian nilai *alpha* terhadap performa klasifikasi *Naïve Bayes*

Tabel 4.5 Pengujian nilai *alpha* terhadap performa *Naïve Bayes*

Alpha	*Data test* (%)	*Accuracy* (%)	*Presicion (%)*	*Recall (%)*
100	50	22	5.5	25
100	40	23	5.75	25
100	30	17	4.25	25
100	20	15	3.75	25
100	10	10	2.5	25
10	50	48	32.5	50
10	40	23	5.75	25
10	30	43	30.75	50
10	20	80	60.75	75
10	10	80	58.25	75
1	50	76	62	75
1	40	50	32.75	75
1	30	73	59.5	75
1	20	80	60.75	75
1	10	80	58.25	75



Gambar 4.4 Hasil pengujian parameter *alpha*

Pada pengujian parameter *Alpha*, semakin tinggi nilai *Alpha* performa klasifikasi akan semakin menurun, begitupun sebaliknya jika nilai *Alpha* makin kecil maka performa klasifikasi akan semakin meningkat

5 Kesimpulan dan Saran

5.1 Kesimpulan

1. Sistem klasifikasi emosi berdasarkan lirik lagu berbahasa Indonesia sudah berjalan dengan baik dengan hasil uji alpha atau fungsionalitas sebesar 100%
2. Sistem klasifikasi Naïve Bayes dengan atau tanpa optimasi PSO memiliki tingkat performa akurasi, presisi dan recall yang hampir sama.
3. Tingkat akurasi tertinggi yang didapatkan oleh algoritma Naïve Bayes tanpa Optimasi PSO dan dengan Optimasi PSO sebesar 80%. Optimasi PSO dapat meningkatkan tingkat akurasi dari klasifikasi, pada dataset dengan ukuran distribusi data 60:40, tingkat akurasinya meningkat sebesar 23%.

5.2 Saran

Berdasarkan hasil perancangan dan pengujian sistem yang telah dilakukan pada tugas akhir ini, maka berikut adalah saran-saran yang diusulkan untuk penelitian lebih lanjut, yaitu:

1. Menambah jumlah dataset, agar model yang dilatih akan semakin baik.
2. Menambahkan kamus POS – Tagging dan Stopword removal

Referensi

[1] D. Hude, *Emosi: Penjelalajahan Religio Psikologis*. Erlangga, 2006.

[2] Y. An, S. Sun, and S. Wang, “Naive Bayes classifiers for music emotion classification based on lyrics,” *Proc. - 16th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2017*, no. 1, pp. 635–638, 2017, doi: 10.1109/ICIS.2017.7960070.

[3] S. Chauhan and P. Chauhan, “Music mood classification based on lyrical analysis of Hindi songs using Latent Dirichlet Allocation,” *2016 Int. Conf. Inf. Technol. InCITe 2016 - Next Gener. IT Summit Theme - Internet Things Connect your Worlds*, pp. 72–76, 2017, doi:

10.1109/INCITE.2016.7857593.

[4] A. T. J. H, “Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining,” *Inform. UPGRIS*, vol. 1, pp. 1–9, 2015.

[5] M. Kamayani, “Perkembangan Part-of-Speech Tagger Bahasa Indonesia,” *J. Linguist. Komputasional*, vol. 2, no. 2, p. 34, 2019, doi: 10.26418/jlk.v2i2.20.

[6] D. N. Armianti, Indriati, and S. Adinugroho, “Klasifikasi Emosi Lagu Berdasarkan Lirik pada Teks Berbahasa Indonesia Menggunakan K-Nearest Neighbor dengan Pembobotan WIDF,” *J. Nas. Teknol. dan Ilmu Komput.*, vol. 3, no. 10, pp. 10161–10167, 2019.

[7] A. Deolika, K. Kusrini, 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.

[8] M. Mansur, T. Prahasto, and F. Farikhin, “Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi,” *J. Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 11–19, 2014, doi: 10.21456/vol4iss1pp11-19.

[9] 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.

[10] X. Bai, X. Gao, and B. Xue, “Particle Swarm Optimization Based Two-Stage Feature Selection in Text Mining,” *2018 IEEE Congr. Evol. Comput. CEC 2018 - Proc.*, pp. 1–8, 2018, doi: 10.1109/CEC.2018.8477773.

[11] D. J. Hand, *Principles of data mining*, vol. 30, no. 7. 2007.