



Seleksi Fitur menggunakan Algoritma Particle Swarm Optimization pada Klasifikasi Kelulusan Mahasiswa dengan Metode Naive Bayes

Evi Purnamasari¹, Dian Palupi Rini^{2*}, Sukemi³

^{1,2,3}Program Studi Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya Palembang

¹evipurnama8@gmail.com, ^{2*}dprini@unsri.ac.id, ³sukemi@ilkom.unsri.ac.id

Abstract

The study of the classification of student graduation at a university aims to help the university understand the academic development of students and to be able to find solutions in improving the development of student graduation in a timely manner. The Naive Bayes method is a statistical classification method used to predict a student's graduation in this study. The classification accuracy can be improved by selecting the appropriate features. Particle Swarm Optimization is an evolutionary optimization method that can be used in feature selection to produce a better level of accuracy. The testing results of the alumni data using the Naive Bayes method that optimized with the Particle Swarm Optimization algorithm in selecting appropriate features, producing an accuracy value of 86%, 6% higher than the classification without feature selection using the Naive Bayes method.

Keywords: *classification, student graduation, fitur, selection, naive bayes, PSO*

Abstrak

Studi tentang klasifikasi kelulusan mahasiswa di sebuah Perguruan Tinggi bertujuan untuk membantu pihak universitas memahami perkembangan akademik mahasiswa serta agar dapat mencari solusi dalam peningkatan perkembangan kelulusan mahasiswa secara tepat waktu. Metode Naive Bayes merupakan metode pengklasifikasian statistik yang digunakan untuk memprediksi suatu kelulusan mahasiswa dalam penelitian ini. Peningkatan akurasi klasifikasi dapat dilakukan dengan memilih fitur yang sesuai. Particle Swarm Optimization merupakan metode optimasi yang bersifat evolusioner yang dapat digunakan dalam seleksi fitur untuk menghasilkan tingkat akurasi yang lebih baik. Hasil pengujian terhadap data alumni menggunakan metode Naive Bayes yang dioptimasi dengan algoritma Particle Swarm Optimization dalam memilih fitur yang sesuai, menghasilkan nilai akurasi 86%, lebih tinggi 6% dibandingkan klasifikasi yang tanpa seleksi fitur menggunakan metode Naive Bayes.

Kata kunci: klasifikasi, kelulusan Mahasiswa, fitur, seleksi, naive bayes, PSO

1. Pendahuluan

Kualitas Perguruan Tinggi banyak dipengaruhi oleh sebab dan akibat salah satu yaitu keberhasilan mahasiswa dengan lulus tepat waktu atau tidak tepat waktu. Tingginya tingkat suatu keberhasilan mahasiswa akan sangat berpengaruh pada akreditasi di Perguruan Tinggi [1][2].

Faktor yang menyebabkan tingkat kelulusan mahasiswa di Perguruan Tinggi sangat penting diketahui, agar dapat mencari solusi untuk semua mahasiswa dapat lulus dengan tepat waktu. Ada banyak faktor yang

berpengaruh pada kelulusan mahasiswa diantaranya rendahnya kemampuan pada bidang akademik, program kuliah, indeks prestasi kumulatif bahkan banyak factor lainnya[3].

Peneliti tentang memprediksi kelulusan menggunakan metode Naive bayes tentang kelulusan mahasiswa yang menghasilkan nilai akurasi sebesar 80,72% dengan menggunakan 6 atribut [4].

Penelitian tentang prediksi mahasiswa kelulusan tepat waktu telah diuji coba. Penelitian yang menggunakan

metode C4.5 yang menghasilkan nilai akurasi sebesar 82% [5].

Metode naive bayes mendapatkan nilai baik dalam prediksi menurut [6] dengan nilai akurasi terbaik yaitu 83%.

Peningkatan akurasi dapat dilakukan dengan seleksi fitur menurut [7] subset fitur yang dikembangkan dengan cara memilih kurang dari setengah atau mendekati 10% dari dataset dengan fitur yang tersedia, tetapi mencapai kinerja klasifikasi yang menghasilkan jauh lebih baik dari pada menggunakan semua fitur pada dataset. PSO merupakan salah satu algoritma optimasi yang dapat digunakan untuk seleksi fitur.

PSO dapat secara otomatis mengembangkan subset fitur dengan jumlah fitur yang lebih kecil dan kinerja klasifikasi yang lebih tinggi dari pada menggunakan semua fitur, PSO merupakan metode optimasi yang akan digunakan untuk seleksi fitur. PSO diawali dengan menginisiasi posisi dari partikel-partikel secara acak, selanjutnya partikel-partikel tersebut bergerak pada ruang pencarian untuk mencari kandidat solusi yang paling optimal [8].

2. Metode Penelitian

Pada penelitian ini peneliti menggunakan metode Naive Bayes improved menggunakan algoritma Particle Swarm Optimization dengan seleksi fitur untuk menghasilkan nilai akurasi yang lebih baik.

2.1. Metode Naive Bayes

Metode Naive Bayes merupakan suatu klasifikasi statistik yang mampu digunakan untuk memprediksi kemungkinan sesuatu yang akan datang. Metode Naive Bayes berdasarkan teorema Bayes memiliki suatu kegunaan untuk klasifikasi kelas yang sama dengan menghasilkan pohon keputusan. Metode Naive Bayes terbukti menghasilkan suatu nilai akurasi dan kecepatan yang lebih tinggi pada saat pengujian dengan jumlah dataset yang lebih banyak [9][10].

Rumus Naive Bayes :

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (1)$$

Dimana

x = Data class yang belum diketahui

c = Hipotesis data x yang merupakan suatu class yang lebih spesifik

P(c|x) = posteriori probability

P(c) = prior probability

P(c|x) = Probabilitas berdasarkan kondisi hipotesis

P(x) = Probabilitas c

2.2. Algoritma Particle Swarm Optimization (PSO)

PSO merupakan proses algoritma yang terinspirasi dari suatu perilaku sosial pada sekumpulan hewan [11]. PSO adalah pendekatan yang berbasis populasi untuk dapat

menyelesaikan masalah optimasi yang terus menerus dilakukan. Pada saat iterasi, kecepatan saat partikel dibuat secara stochastic sesuai pengaruh untuk solusi yang terbaik, lalu menghitung titik baru untuk dievaluasi [12].

Rumus untuk perhitungan perpindahan posisi dan kecepatan partikel :

$$v_i(t) = v_i(t-1) + c_1 r_1 [x_{pbest_i} - x_i(t)] + c_2 r_2 [x_{gbest_i} - x_i(t)] \quad (2)$$

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

Dimana

$v_i(t)$: kecepatan partikel ke-i pada iterasi t

$x_i(t)$: posisi partikel ke-i pada iterasi t

c_1 dan c_2 : konstanta learning rate untuk kemampuan individu (*cognitive*) dan pengaruh sosial (*group*)

r_1 dan r_2 : bilangan acak berdistribusi uniform dalam jarak antara 0 dan 1

x_{pbest_i} : posisi terbaik partikel ke-i

x_{gbest_i} : posisi terbaik global

PSO dengan menggunakan nilai *fitness* untuk mengevaluasi setiap fitur. Pada setiap iterasi, nilai *fitness* pada atribut dihitung dengan menggunakan *fitness function*. Iterasi dari PSO akan berhenti setelah mencapai nilai *fitness target* atau mencapai iterasi maksimum.

2.3. Metode Naive Bayes Improved dengan Algoritma PSO

PSO digunakan untuk memilih atribut yang optimum untuk digunakan di dalam proses klasifikasi. PSO memerlukan nilai *fitness* saat seleksi fitur, untuk mencari kandidat solusi terbaik. Pada proses tersebut nilai *fitness* merupakan akurasi dari Naive Bayes yang mana pada proses klasifikasi menggunakan atribut yang direpresentasikan oleh setiap partikel. Representasi dari atribut pada setiap bagian dimensi di setiap partikel dapat dijadikan posisi dari partikel, dimana representasinya berupa *binary string* 0 dan 1, nilai 0 berarti nilai dari partikel tersebut tidak akan digunakan yang disebut nilai yang tidak aktif dan sebaliknya untuk nilai 1 akan digunakan disebut nilai yang aktif [13].

Dikarenakan representasi partikel yang merupakan *binary*, maka perlu sedikit modifikasi pada PSO. Pada tahun 1997, Kennedy dan Eberhart memperkenalkan modifikasi algoritma PSO yaitu, *Binary Particle Swarm Optimization* (BPSO) dimana posisi setiap partikel pada ruang pencarian direpresentasikan dengan *binary string*.

$$v_i = (v_i^1, v_i^2, \dots, v_i^n) \quad (4)$$

$$x_i = (x_i^1, x_i^2, \dots, x_i^n) \quad (5)$$

dimana :

x_i : posisi partikel ke- i

d : 1, 2, ..., n (n merupakan dimensi dari data)

Rumus menghitung perubahan untuk kecepatan setiap partikel BPSO :

$$v_i^d(t) = v_i^d(t-1) + c_1 r_1 [x_{pbest_i}^d - x_i^d(t)] + c_2 r_2 [x_{gbest}^d - x_i^d(t)] \quad (6)$$

Rumus menghitung perubahan untuk posisi setiap partikel pada BPSO dapat menggunakan fungsi *sigmoid* dengan menjadikan nilai kecepatan partikel sebagai parameternya :

$$Sigmoid(v_i^d) = \frac{1}{1 + e^{-v_i^d}} \quad (7)$$

$$if \text{sigmoid}(v_i^d) > r_3 \text{ then } x_i^d = 1 \text{ else } x_i^d = 0 \quad (8)$$

Dimana :

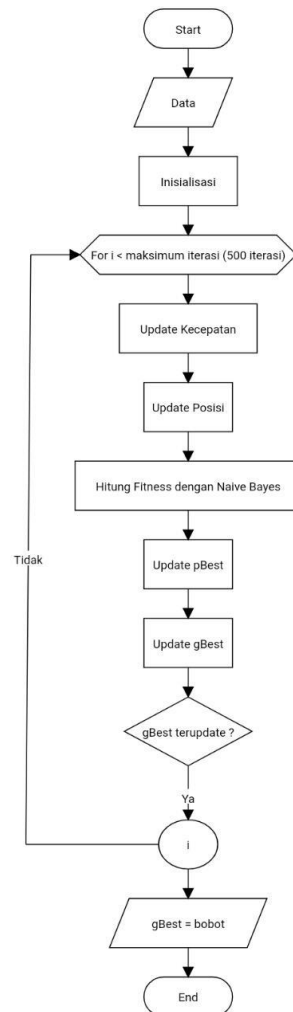
r_3 : nilai acak yang digunakan selama proses pencarian solusi

Langkah dalam menerapkan Naive Bayes-PSO dapat mengikuti langkah-langkah sebagai berikut :

1. Inisialisasi populasi secara acak dengan angka *biner* 0 dan 1. Posisi partikel tersusun dari *binary string* yang merepresentasikan dimensi pada data.
2. Pada dimensi yang bernilai 0, tidak akan digunakan dalam proses klasifikasi pada setiap partikel sehingga dimensi pada data yang bernilai 0 akan diabaikan.
3. Selanjutnya di lakukan proses klasifikasi dengan menggunakan Naive Bayes yang berdasarkan pada atribut dan rules yang ada pada setiap partikel.
4. Kemudian dari proses klasifikasi tersebut, di hitung nilai akurasi. Jadikan nilai akurasi tersebut sebagai nilai *pbest* pada partikel ke- i pada dimensi d .
5. Setelah sekian banyak partikel, pilih partikel dengan nilai akurasi yang tertinggi dan jadikan nilai akurasinya sebagai nilai *gbest* pada dimensi d .
6. Pada setiap partikel, bandingkan nilai akurasi dari posisi terbaik pada iterasi ke- t dengan akurasi dari populasi global terbaik pada iterasi ke- t . Jika posisi pada iterasi ke- $t-1$ lebih baik dari iterasi ke- t , maka posisi global sama dengan posisi pada iterasi ke- t .
7. Ubah posisi dan kecepatan pada setiap partikel. Sebelum iterasi mencapai iterasi maksimum, maka ulangi dari langkah ke empat untuk iterasi selanjutnya.
8. Partikel dengan nilai akurasi yang terbaik pada akhir iterasi, maka akan dijadikan kandidat solusi yang terbaik. Sehingga atribut-atribut yang dimiliki oleh partikel tersebut akan digunakan pada klasifikasi.

Penelitian ini menerapkan metode klasifikasi dengan metode Naive Bayes yang digunakan untuk memprediksi suatu kelulusan mahasiswa menggunakan seleksi fitur kemudian di tingkatkan menggunakan

algoritma PSO untuk menghasilkan tingkat akurasi yang lebih baik. Optimasi metode Naive Bayes dengan Improved PSO di tunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Naive Bayes Improve PSO

3. Hasil dan Pembahasan

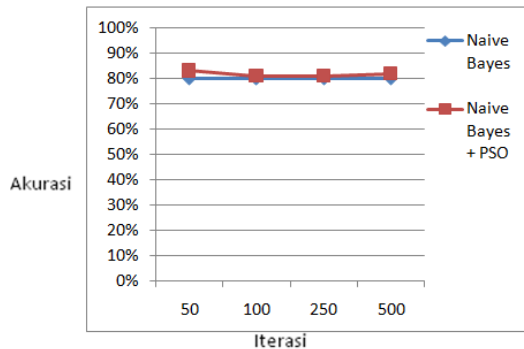
Dataset didapat melalui Sistem Informasi Akademik uigm. Peneliti melakukan uji coba menggunakan data alumni dengan jumlah dataset 894 data alumni dan terdapat 10 atribut yang digunakan. Peneliti melakukan uji coba untuk mendapatkan hasil akurasi dengan pembagian tahap pengujian menggunakan dua data pengujian yaitu data training dan data testing berbeda.

3.1 Uji coba dengan Menggunakan Data Training 70% dan Data Testing 30%

Uji coba dilakukan dengan menggunakan data training 70% dan testing 30% dan dengan perbandingan menggunakan jumlah iterasi yang berbeda akan tetapi dengan jumlah partikel dan c_1 c_2 yang sama. Berikut hasil uji coba ditunjukkan pada tabel 1 dan 2 serta pada gambar 2 dan 3.

Tabel 1. Tabel Iterasi dari Data Training 70%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	15	1	1	80%	86%
100	15	1	1	80%	81%
250	15	1	1	80%	81%
500	15	1	1	80%	81%

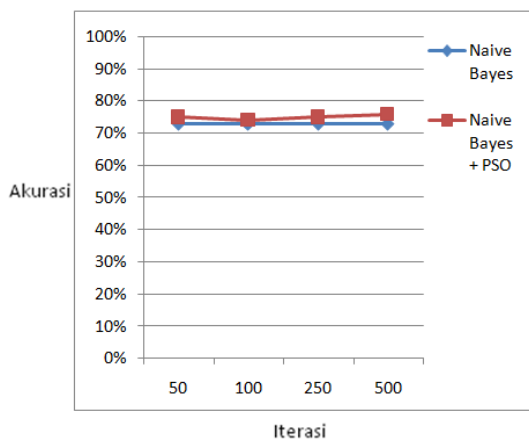


Gambar 2. Grafik Iterasi dari Data Training 70%

Hasil dari pengujian data training 70% terdapat di Tabel 1 dan grafik 2 menunjukkan bahwa hasil pada saat uji coba menggunakan jumlah partikel yang sama yaitu 15 partikel adalah iterasi terbaik terdapat pada jumlah ke 50 iterasi dengan hasil nilai akurasi Naive Bayes 80% dan Naive Bayes Improved PSO 86%.

Tabel 2. Tabel Iterasi dari Data Testing 30%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	15	1	1	71%	74%
100	15	1	1	71%	74%
250	15	1	1	71%	72%
500	15	1	1	71%	75%



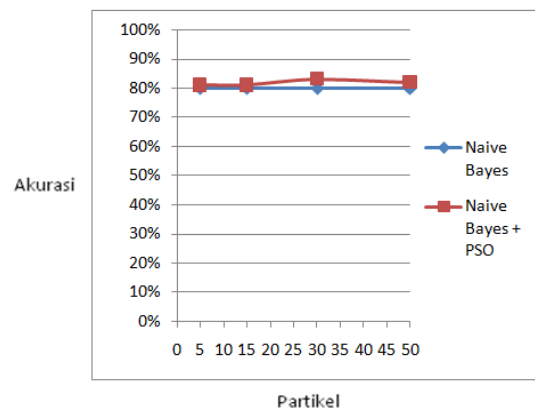
Gambar 3. Grafik Iterasi dari Data Testing 30%

Pada uji coba data testing 30% yang ada di Tabel 2 dan grafik 3 hasil yang didapat dari menggunakan jumlah 15 partikel dan $c_1 c_2$ yang sama yaitu iterasi terbaik didapat pada jumlah ke 500 iterasi menghasilkan nilai akurasi yang terbaik Naive Bayes 71% dan Naive Bayes PSO yang terbaik 75%.

Hasil dari uji coba menggunakan iterasi terbaik dari data training 70% dan data testing 30% dilakukan uji coba dengan jumlah partikel yang berbeda – beda, dan menghasilkan akurasi pada tabel 3 dan 4 serta grafik 4 dan 5.

Tabel 3. Tabel Partikel Data Training 70%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	5	1	1	80%	81%
50	15	1	1	80%	81%
50	30	1	1	80%	83%
50	50	1	1	80%	82%

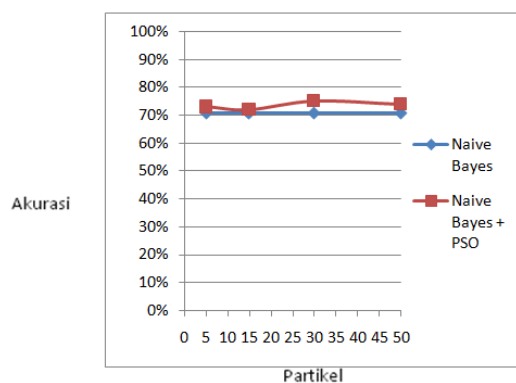


Gambar 4. Grafik Partikel Data Training 70%

Hasil percobaan yang dilakukan dari menggunakan iterasi terbaik ialah ke 50 iterasi dari data training 70% dan menggunakan beberapa jumlah partikel yang berbeda teruji pada saat jumlah ke 30 partikel adalah yang menghasilkan nilai akurasi terbaik dengan nilai akurasi Naive Bayes 80% dan Naive Bayes PSO 83%.

Tabel 4. Tabel Partikel Data Testing 30%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	15	1	1	71%	74%
100	15	1	1	71%	74%
250	15	1	1	71%	72%
500	15	1	1	71%	75%



Gambar 5. Grafik Partikel Data Testing 30%

Pengujian menggunakan data testing 30% dengan menggunakan iterasi terbaik ke 500 iterasi setelah itu diuji menggunakan partikel – partikel yang berbeda dan didapat hasil yang terbaik yaitu pada saat partikel ke 30 partikel dengan nilai akurasi terbaik yaitu Naive Bayes sebesar 71% dan Naive Bayes PSO 75%.

Kemudian hasil yang di dapat dari iterasi dan partikel yang terbaik dengan menggunakan data training dan data testing dilakukan uji coba menggunakan c_1 dan c_2 yang berbeda sehingga menghasilkan nilai akurasi pada tabel 5 dan 6.

Tabel 5. Tabel $c_1 c_2$ dengan Iterasi dan Partikel Terbaik Data Training 70%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	30	1	1	80%	81%
50	30	1	2	80%	81%
50	30	2	1	80%	81%
50	30	2	2	80%	82%

Data yang dihasilkan dari iterasi dan partikel yang terbaik pada saat proses uji coba menggunakan data training 70% kemudian dilakukan uji coba menggunakan data $c_1 c_2$ yang berbeda menghasilkan nilai akurasi terbaik yaitu Naive Bayes 80% dan Naive Bayes PSO 82% dengan menggunakan c_1 2 dan c_2 2.

Tabel 6. Tabel $c_1 c_2$ dengan Iterasi dan Partikel Terbaik Data Testing 30%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
500	30	1	1	71%	73%
500	30	1	2	71%	72%
500	30	2	1	71%	73%
500	30	2	2	71%	75%

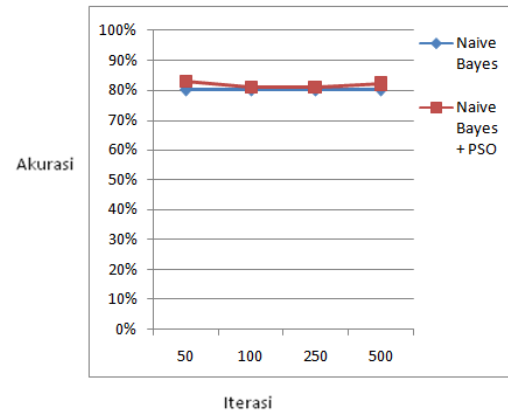
Hasil uji coba dari iterasi dan partikel terbaik dengan $c_1 c_2$ yang berbeda menggunakan data testing 30% nilai akurasi yang terbaik yaitu Naive Bayes 71% dan Naive Bayes PSO 75% dengan menggunakan c_1 2 dan c_2 2. Analisa hasil yang diperoleh saat uji coba menggunakan data training 70% menghasilkan jumlah iterasi terbaik yaitu ke 50 iterasi dan ke 30 partikel terbaik dengan uji coba menggunakan c_1 2 c_2 2 dan menghasilkan akurasi terbaik pada Naive Bayes 80% sedangkan Naive Bayes PSO 82%. Kemudian dari uji coba data testing 30% hasil yang didapat iterasi terbaik adalah pada saat ke 500 iterasi dan 30 partikel menggunakan c_1 2 dan c_2 2 menghasilkan nilai akurasi terbaik Naive Bayes 71% dan Naive Bayes 75%.

3.2 Uji coba dengan Menggunakan Data Training 60% dan Data Testing 40%

Uji coba dilakukan dengan menggunakan data training 60% dan data testing 40% dan dengan perbandingan menggunakan jumlah iterasi yang berbeda namun dengan jumlah partikel dan $c_1 c_2$ yang sama. Berikut hasil uji coba yang dilakukan ditunjukkan pada tabel 7 dan 8 serta pada gambar 6 dan 7.

Tabel 7. Tabel Iterasi dari Data Training 60%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	15	1	1	80%	83%
100	15	1	1	80%	81%
250	15	1	1	80%	81%
500	15	1	1	80%	82%

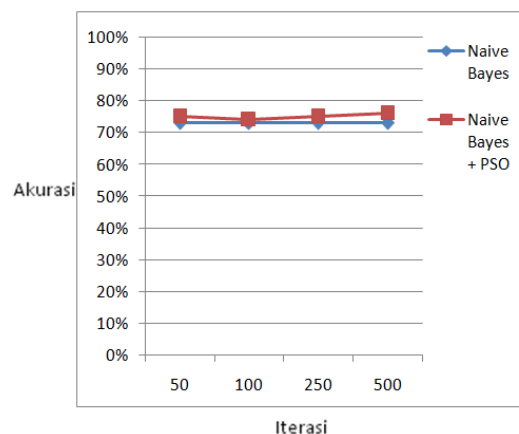


Gambar 6. Grafik Iterasi dari Data Training 60%

Hasil uji coba menggunakan data training 60% yang ditunjukkan pada Tabel 7 dan grafik 6 yang merupakan hasil akurasi dengan menggunakan jumlah partikel yang sama 15 partikel, dari iterasi terbaik ada pada jumlah 50 iterasi dengan menghasilkan nilai akurasi Naive Bayes 80% dan Naive Bayes PSO 83%.

Tabel 8. Tabel Iterasi dari Data Testing 40%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	15	1	1	74%	75%
100	15	1	1	74%	75%
250	15	1	1	74%	77%
500	15	1	1	74%	75%



Gambar 7. Grafik Iterasi dari Data Testing 40%

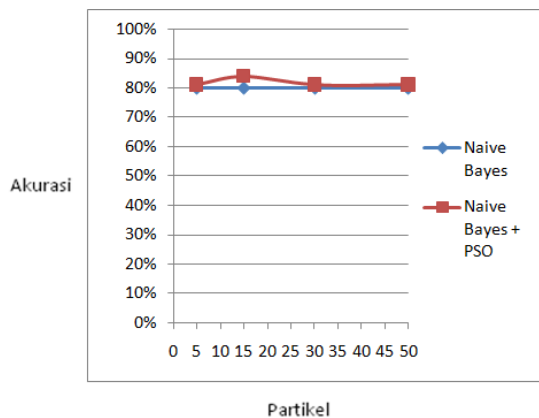
Hasil uji coba dari data testing 40% yang pada Tabel 8 dan grafik 7 bahwa dengan menggunakan jumlah partikel 15 partikel dengan c_1 dan c_2 yang sama, dari iterasi terbaik pada jumlah 250 iterasi dengan

menghasilkan nilai akurasi terbaik dari Naive Bayes 74% dan dari Naive Bayes PSO 77%.

Uji coba dari iterasi terbaik dengan menggunakan data training 60% dan data testing 40% dilakukan uji coba dengan jumlah partikel yang berbeda, dan menghasilkan nilai akurasi pada tabel 9 dan 10 serta grafik 8 dan 9.

Tabel 9. Tabel Partikel Data Training 60%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	5	1	1	80%	81%
50	15	1	1	80%	84%
50	30	1	1	80%	81%
50	50	1	1	80%	81%

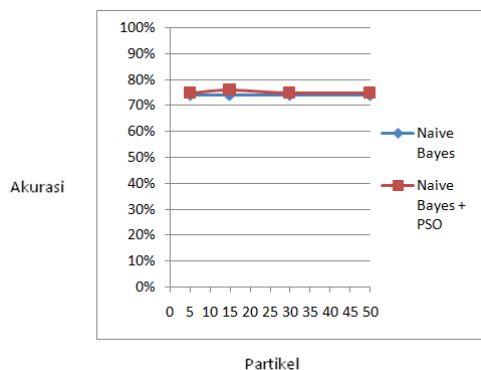


Gambar 8. Grafik Partikel Data Training 60%

Uji coba dilakukan dengan iterasi terbaik adalah 50 iterasi menggunakan data training 60% dengan menggunakan jumlah partikel berbeda terdapat di jumlah 15 partikel yang menghasilkan nilai akurasi terbaik dengan akurasi Naive Bayes 81% dan Naive Bayes PSO 84%.

Tabel 10. Tabel Partikel Data Testing 40%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
250	5	1	1	74%	75%
250	15	1	1	74%	76%
250	30	1	1	74%	75%
250	50	1	1	74%	75%



Gambar 9. Grafik Partikel Data Testing 40%

Hasil uji coba menggunakan data testing 40% dengan menggunakan data iterasi yang terbaik yaitu 250 iterasi dan selanjutnya diuji menggunakan beberapa partikel yang berbeda-beda dan didapat hasil yang terbaik yaitu pada 15 jumlah partikel dengan hasil nilai akurasi terbaik yaitu Naive Bayes 76% dan Naive Bayes PSO 74%.

Hasil dari iterasi dan partikel terbaik uji coba menggunakan data training dan data testing selanjutnya diuji coba dengan menggunakan $c_1 c_2$ yang berbeda sehingga dapat menghasilkan nilai akurasi pada tabel 11 dan 12.

Tabel 11. Tabel $c_1 c_2$ dengan Iterasi dan Partikel Terbaik Data Training 60%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
50	15	1	1	80%	81%
50	15	1	2	80%	81%
50	15	2	1	80%	84%
50	15	2	2	80%	82%

Hasil uji coba dengan menggunakan iterasi dan partikel yang terbaik dan dengan menggunakan $c_1 c_2$ yang berbeda dari data training 60% sehingga menghasilkan nilai akurasi terbaik yaitu pada Naive Bayes 80% dan Naive Bayes PSO 84% dengan menggunakan c_1 2 dan c_2 1.

Tabel 12. Tabel $c_1 c_2$ dengan Iterasi dan Partikel Terbaik Data Testing 40%

PSO Iterasi	Jumlah Partikel	c_1	c_2	Naive Bayes	Naive Bayes + PSO
250	15	1	1	74%	75%
250	15	1	2	74%	76%
250	15	2	1	74%	75%
250	15	2	2	74%	75%

Hasil uji coba dari iterasi dan partikel terbaik dengan $c_1 c_2$ yang berbeda menggunakan data testing 40% dihasilkan nilai akurasi terbaik Naive Bayes 74% dan Naive Bayes PSO 76% dengan menggunakan c_1 1 dan c_2 2.

Analisa hasil dari uji coba menggunakan data training 60% diperoleh jumlah iterasi terbaik yaitu 50 iterasi dan 15 jumlah partikel terbaik dengan uji coba menggunakan c_1 2 c_2 1 sehingga menghasilkan akurasi terbaik pada Naive Bayes 80% dan Naive Bayes PSO 84%. Sedangkan hasil uji coba dengan data testing 40% hasil yang diperoleh yaitu jumlah iterasi terbaik adalah pada saat 250 iterasi dan 15 jumlah partikel menggunakan c_1 1 dan c_2 2 yang menghasilkan nilai akurasi terbaik Naive Bayes 74% dan Naive Bayes 76%.

4. Kesimpulan

Berdasarkan hasil dari uji coba kesimpulan yang didapat adalah :

1. Pada hybrid metode Naive Bayes dan algoritma PSO, Seleksi fitur pada Naive Bayes menggunakan PSO mampu meningkatkan meningkatkan nilai akurasi dari klasifikasi
 2. Setelah dilakukan uji coba menggunakan jumlah dataset yang berbeda, bahwa pada saat iterasi ke 50 dengan jumlah partikel 15 memberikan nilai akurasi yang lebih baik, sedangkan dengan menguji coba menggunakan c_1 dan c_2 yang berbeda – beda tidak begitu mempengaruhi hasil nilai akurasi.
 3. Pada saat testing data beberapa hasil uji coba yang dilakukan bahwa untuk menggunakan metode Naive Bayes menghasilkan nilai akurasi terbaik 80% setelah di optimasi menggunakan algoritma PSO menghasilkan nilai akurasi 86%.
- Saran peneliti untuk peneliti selanjutnya, adalah sebagai berikut :
1. Saran yang diajukan agar peneliti selanjutnya menggunakan metode yang telah dilakukan uji coba pada penelitian ini dengan menggunakan studi kasus yang berbeda.
 2. Saran untuk peneliti selanjutnya agar melakukan uji coba dengan melakukan antara metode hybrid Naive Bayes dengan metode lain menggunakan dataset yang sama agar dapat menghasilkan pengujian metode hybrid yang lebih baik dan mendapatkan nilai akurasi yang lebih tinggi.
 3. Saran selanjutnya agar peneliti mencoba menggunakan dataset dengan jumlah yang lebih banyak agar menjadi bahan perbandingan lebih baik menggunakan dataset yang lebih banyak apakah menghasilkan nilai akurasi yang lebih baik atau tidak.
- Daftar Rujukan**
- [1] Aryasanti, A. (2018). *Sistem Komparasi Naive Bayes dan Decision Trees untuk Menentukan Klasifikasi Kegagalan Studi Mahasiswa*. Jurnal TICOM, Vol. 6 No.3 Mei.
 - [2] Romadhona, A. (2017). *Prediksi Kelulusan Mahasiswa Tepat Waktu*. Jurnal Teknologi Informasi Cyberku, Vol. 13 No. 1, pp 69–83.
 - [3] Amelia, M. W., Lumenta, A. S. M., dan Jacobus, A. (2017). *Prediksi Masa Studi Mahasiswa dengan Menggunakan Algoritma Naive Bayes*. E-Journal Teknik Informatika, Vol. 11, No. 1 ISSN: 2301-8364.
 - [4] Salmu, S., dan Solichin, A. (2017). *Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu Menggunakan Naive Bayes : Studi Kasus UIN Syarif Hidayatullah Jakarta*. Prosiding Seminar Nasional Multidisiplin Ilmu Universitas Budi Luhur, (April), pp 701–709.
 - [5] Astuti, I. P. (2017). *Prediksi Ketepatan Waktu Kelulusan Dengan Algoritma Data Mining C4.5*. Fountain of Informatics Journal, Vol. 2 No. 2, pp 41-45.
 - [6] Ridwan, M., Suyono, H., dan Sarosa, M. (2013). *Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier*. Jurnal EECIS Vol.7, No. 1, Juni.
 - [7] B. Xue, M. Zhang, W. Browne. (2014). *Novel initialisation and updating mechanisms in PSO for feature selection in classification*, in: Applications of Evolutionary Computation, vol. 7835 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2013, pp. 428–438.
 - [8] Xue, B., Zhang, M., Browne, W. N. (2014). *Particle Swarm Optimisation For Feature Selection in Classification : Novel Initialisation and Updating Mechanisms*. Applied Soft Computing, Vol. 18, 261–276.
 - [9] Pujianto, U., Azizah, E. N., dan Damayanti, A. S. (2017). *Naive Bayes Using to Predict Students ' Academic Performance at Faculty of Literature*. International Conference Electrical, Electronics and Information Engineering (ICEEIE), Malang, pp 163–169.
 - [10] Kesumawati, A., dan Utari, D. T. (2018). *Predicting patterns of student graduation rates using Naive bayes classifier and support vector machine*. AIP Conference Proceedings, 2021 (October). <https://doi.org/10.1063/1.5062769>
 - [11] Abadlia, H., Smairi, N., dan Ghedira, K. (2018). *A hybrid Immigrants schema for particle swarm optimization algorithm*. Procedia Computer Science, 126, pp 105–115. <https://doi.org/10.1016/j.procs.2018.07.214>
 - [12] Kaur, S., dan Goyal, M. (2014). *Fast and robust Hybrid Particle Swarm Optimization Tabu Search Association Rule Mining (HPSO-ARM) algorithm for Web Data Association Rule Mining (WDARM)*. International Journal of Advance Research in Computer Science and Management Studies, 2(9), pp 448–451.
 - [13] Primartha, R., Tama, B. A., Arliansyah, A., Miraswan, K. J. (2018). *Decision tree combined with PSO-based feature selection for sentiment analysis*. Journal of Physics: Conference Series, Volume 1196, International Conference on Information System, Computer Science and Engineering 26–27 November 2018, Palembang, Indonesia.