

Analisis Seleksi Mahasiswa Baru Jalur Non-tulis Menggunakan Algoritma *Ant-miner*

M. Abdul Jabbar¹ Indwiarti² Fitriyani³

^{1,2,3}Prodi Ilmu Komputasi Telkom University, Bandung

¹Abdul.Jabbar.3794@gmail.com ²Indwindwi@gmail.com ³Fitriyani.v@gmail.com

Abstrak

Seleksi mahasiswa baru jalur non-tulis adalah salah satu dari jalur masuk Universitas dengan jumlah pendaftar yang tinggi. Tugas akhir ini membahas bagaimana cara melakukan seleksi mahasiswa baru jalur non-tulis dengan menggunakan *rules* yang didapat dari metode klasifikasi. Diharapkan *rules* hasil klasifikasi dapat digunakan untuk membantu mengevaluasi penerimaan mahasiswa baru dari jalur non-tulis. Klasifikasi terhadap data seleksi mahasiswa baru jalur non-tulis dapat dilakukan dengan algoritma *ant-miner*. *Ant-miner* (ant-colony based data miner) adalah algoritma yang digunakan untuk mengekstrak *rules* klasifikasi dari data dan telah memberikan hasil yang cukup memuaskan dalam beberapa jenis data kompleks yang telah diujikan. Tugas akhir ini juga bertujuan untuk menghasilkan analisis dari penggunaan algoritma *ant-miner* terhadap data seleksi mahasiswa baru jalur non-tulis. Hasil penelitian tugas akhir ini menunjukkan bahwa algoritma *ant-miner* menghasilkan akurasi training dan testing yang cukup baik, tidak overfitting, dan menghasilkan *rules* dengan akurasi, *recall* dan presisi yang baik sehingga dapat digunakan untuk mengevaluasi penerimaan mahasiswa baru jalur non-tulis.

Kata kunci : Klasifikasi, Seleksi Mahasiswa Baru, *Ant-colony optimization*, *Ant-miner*.

Abstract

Non-written new student acceptance is one of the ways to get into University with high total of registrant. This bachelorial final project discusses how to select the new student from non-written path by using rules extracted from classification. It is hoped that the classification rules could be used to help to evaluate the acceptance of new student from the non-written path. Classification of non-written new student acceptance could be done with ant-miner algorithm. Ant-miner (ant-colony based data miner) is one of the algorithms that used to extract classification rules from data and have been able to give satisfying result in various complex datasets that have been tested with it. This final project also aims to produce analysis of the non-written new student acceptance using ant-miner algorithm. This final project result is that ant-miner produces good training and testing accuracy, no overfitting, and rules with great level of accuracy, recall and precision and could be used well to evaluate the non-written new student acceptance.

Keywords: Classification, New Student Admission, *Ant-colony optimization*, *Ant-miner*.

1. Pendahuluan [10 pts/Bold]

Penilaian tes non-tulis dilakukan manual oleh sebuah tim penilai yang berasal dari bagian admisi dan diperlukan banyak orang untuk menilai jumlah berkas yang jumlahnya ribuan. Masalah seleksi mahasiswa baru dapat diselesaikan dengan menggunakan klasifikasi dengan mencari data-data mana saja yang digolongkan sebagai calon mahasiswa mana saja yang lulus seleksi dan tidak lulus seleksi.

Ant-colony optimization adalah salah satu algoritma yang cukup berkembang pesat akhir-akhir ini. Algoritma yang menjadi bagian dari *swarm intelligence* ini terinspirasi dari bagaimana cara semut mencari jalan terpendek untuk pergi dari sarang koloni semut mereka ke sumber makanan dengan pendekatan jalan terpendek adalah jalan yang paling banyak dilewati semut lain setelah beberapa iterasi. Algoritma *ant-colony* dapat membuat sistem yang *robust* yang mampu membuat solusi kualitas tinggi dengan ruang pencarian yang besar. Algoritma *ant-colony optimization* telah dikembangkan sedemikian rupa sehingga algoritma

tersebut dapat dipakai untuk lebih banyak jenis masalah dan salah satunya adalah klasifikasi. Sebuah penelitian dari Rafael S. Parpinelli, Heitor S. Lopes, dan Alex A. Freitas berhasil membuat sebuah algoritma klasifikasi yang dapat mengklasifikasikan data-data berdasarkan model if-then menggunakan algoritma *ant-colony optimization*. Berdasarkan penelitian ini dapat dikatakan bahwa Algoritma *Ant-miner* hasil penelitian mereka dapat digunakan untuk mencari *rules-rules* klasifikasi dengan sangat akurat dan dapat digunakan untuk menyelesaikan masalah seleksi mahasiswa baru jalur non-tulis[7].

2. Tinjauan Pustaka

2.1 Data Mining

Data mining adalah proses untuk menemukan pola yang memberi pengetahuan, menarik dan belum diketahui, dan juga deskriptif, dapat dimengerti, dan model prediktif dari data berskala besar [1]. Data mining sudah menjadi salah satu bidang ilmu yang berkembang sangat pesat dalam era teknologi informasi yang mengalami perkembangan jumlah,

ukuran, dan dimensi data yang berkembang secara eksponensial. Pencirian pola dari data yang dilakukan dalam data mining sangatlah berguna untuk mendapat informasi lebih lanjut dari data sehingga dapat melakukan berbagai analisis terhadap pengambilan keputusan atau pengambilan kebijakan.

2.2 Ant-miner

Ant-miner adalah implementasi dari *ant-colony optimization* dalam kasus pengekstraksian *rules* klasifikasi. *Ant-miner* mencari sebuah daftar berurut dari *rule-rule* klasifikasi berdasarkan fungsi heuristik *information gain* dan feromon buatan[8]. *Ant-miner* dapat digunakan untuk mengekstraksi *rules* klasifikasi dengan tingkat akurasi yang cukup tinggi berdasarkan uji yang telah diujikan terhadap algoritma tersebut.

Dalam algoritma *ant-colony optimization* setiap semut secara bertahap membentuk atau memodifikasi sebuah solusi. Dalam kasus klasifikasi, solusi dari masalah adalah penemuan *rules* klasifikasi. *Rules* klasifikasi yang ingin ditemukan adalah

IF < Syarat1 AND Syarat2 AND Syaratn ...
> *THEN* < Class >

2.2.1 Fungsi Probabilitas Pengambilan Rule

Langkah pertama dari ant-miner adalah pembentukan rule dengan memilih syarat yang dimasukkan kedalam rule. Untuk perhitungan probabilitas pemilihan titik yang akan dilewati oleh semut dan dalam kasus ant-miner adalah bahwa syarat_i akan dimasukkan kedalam rule parsial sementara adalah sebagai berikut:

$$P_{ji} = \frac{\eta_{ji} * \tau_{ji}(t)}{\sum_{i=1}^a x_i * \sum_{j=1}^{b_i} (\eta_{ji} * \tau_{ji}(t))} \quad (2.1)$$

Keterangan:

P_{ji} = Peluang suatu syarat i,j diambil menjadi rule

η_{ji} = Fungsi Heuristik hasil normalisasi

τ_{ji} = Feromon syarat j,i

a = Jumlah atribut

b = jumlah nilai dalam domain atribut-i

x_i = Penanda atribut

2.2.2 Fungsi Entropy

Untuk setiap syarat_i yang akan ditambahkan kedalam rule, ant-miner harus menghitung nilai dari fungsi heuristik η_{ji} yang mana perkiraan dari kualitas syarat tersebut untuk meningkatkan akurasi prediksi dari rule. Nilai dari η_{ji} adalah pengukuran entropy (jumlah informasi) terhadap syarat tersebut. Untuk setiap syarat_i dari bentuk $A_i = V_{ji}$ dimana A_i adalah atribut ke-I dan V_{ji} adalah nilai ke-j pada A_i . Maka entropinya adalah:

$$\eta_{ji} = \frac{\log_2 k - H(W|A_i=V_{ji})}{\sum_{i=1}^a x_i * \sum_{j=1}^{b_i} (\log_2 k - H(W|A_i=V_{ji}))} \quad (2.2)$$

Keterangan:

η_{ji} = Fungsi Heuristik hasil normalisasi

k = Jumlah kelas dari atribut ke I

$H(W|A_i = V_{ji})$ = Fungsi heuristik sebelum normalisasi

2.2.3 Pemangkasan Rule

Pemangkasan *rule* adalah hal yang umum pada *data mining* [9]. Seperti disebutkan sebelumnya, tujuan utama dari pemangkasan *rule* adalah untuk membuang syarat yang tidak perlu yang bisa masuk ke dalam *rule*. Pemangkasan *rule* bisa meningkatkan kemampuan prediksi dari *rule*, membantu untuk menghindari *overfitting* terhadap data training. Pemangkasan *rule* juga dapat membuat *rule* menjadi lebih sederhana dan lebih mudah dimengerti.

Cara pemangkasan *rule* adalah dengan secara berurutan menghapus syarat demi syarat dari *rule* selama proses ini meningkatkan kualitas dari *rule*. Lebih tepatnya, dalam iterasi pertama dimulai dengan *rule* penuh dengan semua syarat. Kemudian untuk sementara dia mencoba untuk menghapus syarat dari setiap *rule* satu setiap giliran dan kualitas dari *rule* yang telah dihilangkan syaratnya dihitung menggunakan fungsi kualitas *rule*. Syarat yang dihapus yang paling banyak meningkatkan kualitas dari *rule* dengan efektif benar-benar dihapus dari *rule*, dan menyelesaikan iterasi pertama. Dalam iterasi berikutnya dicari lagi syarat yang jika dihapus bisa meningkatkan kualitas dari *rule* dan seterusnya. Proses ini diulangi terus hingga tinggal ada satu syarat atau tidak ada lagi syarat yang jika dihapus dapat meningkatkan kualitas *rule*.

Berikut fungsi kualitas rule:

$$Q = \frac{TP}{TP+FN} * \frac{TN}{FP+TN} \quad (2.3)$$

Keterangan:

TP = True Positives(Jumlah data yang diprediksi dengan benar oleh *rule*)

FP = False Positives(Jumlah data yang dicakupi *rule* dan diprediksi dengan salah oleh *rule*)

FN = False Negatives(Jumlah data yang tidak dicakupi *rule* tetapi diprediksi oleh *rule*)

TN = True Negatives(Jumlah data yang tidak dicakupi *rule* dan tidak diprediksi oleh *rule*)

Q = Kualitas *rule* (dalam rentang 0-1)

2.2.4 Pembaruan Feromon

Setiap syarat_i yang berhubungan dengan jalur yang bisa dilewati oleh semut ant, pada setiap iterasi WHILE dari algoritma *ant-miner* diinisialisasi dengan jumlah feromon yang sama pada awalnya. Jumlah feromon awal yang diberikan pada posisi

jalur berbanding terbalik dengan nilai semua atribut, dan didefinisikan oleh persamaan berikut:

$$\tau_{ji}(t = 0) = \frac{1}{\sum_{i=1}^a b_i} \quad (2.4)$$

Keterangan:

τ_{ji} = Feromon syarat j, i

a = jumlah atribut

b_i = Jumlah nilai yang mungkin dari atribut A_i

Untuk meningkatkan jumlah feromon dilakukan dengan rumus berikut:

$$\tau_{ji}(t + 1) = \tau_{ji}(t) + \tau_{ji}(t) * Q, \forall i, j \in R \quad (2.5)$$

Keterangan:

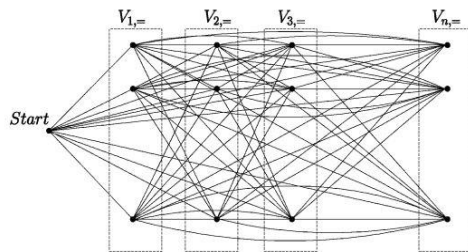
R = Jumlah syarat yang muncul dalam *rule* yang dibangun ant_t

Untuk penguapan feromon yang terjadi jika syarat_{ji} tidak ada pada *rule*, dilakukan dengan cara normalisasi dari nilai feromon τ_{ji} dengan cara membagi nilai feromon τ_{ji} dengan jumlah dari seluruh τ_{ji} untuk semua i, j [7].

$$\tau_{ji}(t + 1) = \frac{\tau_{ji}}{\sum_{i,j=1}^n \tau_{ji}} \quad (2.6)$$

Keterangan:

n = Jumlah seluruh data training



Gambar 2. 1 Skema Algoritma *Ant-miner*

3. Perancangan Sistem

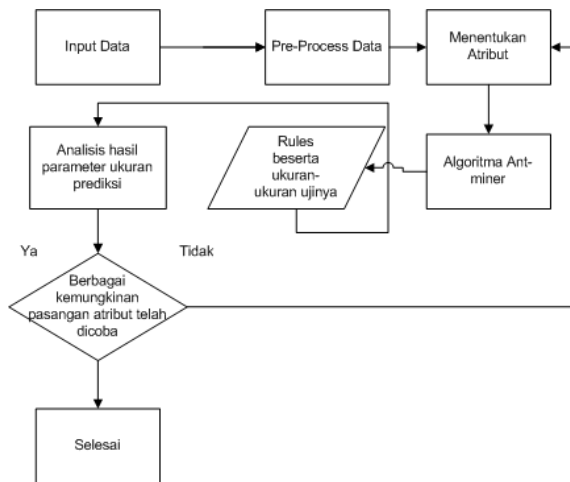
Perangkat lunak ini adalah perangkat lunak yang digunakan untuk memprediksi siswa-siswa SMA yang mendaftar kuliah ke Universitas Telkom melalui jalur nilai rapor non-tulis. Perangkat lunak dapat dioperasikan dengan mudah dengan menginput data nilai mahasiswa, kemudian perangkat lunak melakukan diskretisasi data sehingga data dapat diolah menggunakan algoritma *ant-miner*, kemudian data diolah sehingga didapat hasil *rules* berdasarkan data seleksi mahasiswa baru pada tahun sebelumnya yang menjadi data training dan dapat melakukan testing pada partisi data yang telah dipilih secara acak dari keseluruhan data dan mengeluarkan hasil untuk evaluasi berupa akurasi dan kualitas *rule*.

Hasil dari sistem perangkat lunak ini adalah *rule* berdasarkan algoritma *ant-miner*, kualitas dari masing-masing *rule*, akurasi rata-rata pengaplikasian algoritma pada data testing, dan akurasi rata-rata pengaplikasian algoritma pada data training. Pasangan atribut dengan akurasi paling tinggi digunakan sebagai pasangan atribut *default* program.

3.1 Rancangan Proses Perangkat Lunak

Perangkat lunak secara keseluruhan berjalan melalui proses-proses sebagai berikut:

- 1) Menginput data seleksi mahasiswa baru kedalam sistem untuk di pre-process.
- 2) *Pre-processing* data dilakukan dengan melakukan diskretisasi data, dan mengubah *noise*, *outlier*, dan *missing value* jika ada.
- 3) Menentukan pasangan atribut yang digunakan dalam program.
- 4) Olah data untuk mendapatkan *rules* beserta



Gambar 3. 1 Flowchart Sistem

3.1 Input Data

Input data dapat dilakukan dengan satu klik tombol pada GUI dan memilih file excel berisi data mana yang ingin diolah. Sistem perangkat lunak ini dapat membuka file excel.

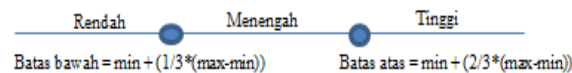
Pemilihan data training dan data testing dilakukan secara acak. Pertama-tama seluruh data dimasukkan sebagai data training, lalu dilakukan iterasi sebanyak jumlah data testing yang diinginkan dengan memindahkan barisan data training ke dalam list data testing.

3.2 Pre-processing Data

Pre-processing data yang dilakukan setelah data diinput adalah diskretisasi. Data numerik nilai yang ada harus diubah menjadi data nominal atau ordinal karena ant-miner hanya dapat mengolah data kategorikal.

$$\text{Batas bawah} = \min + \left(\frac{1}{3} * (\max - \min)\right) \quad (3.1)$$

$$\text{Batas atas} = \min + \left(\frac{2}{3} * (\max - \min)\right) \quad (3.2)$$



Gambar 3. 2 Gambar Selang Diskretisasi

Dibuat 3 tier atau golongan untuk seluruh program studi berdasarkan biaya masuk minimum dimana tier1 adalah golongan dengan tingkat persaingan paling tinggi dan biaya masuk yang tinggi, tier 2 golongan dengan persaingan tingkat menengah dan biaya masuk tingkat menengah, dan tier3 untuk golongan dan persaingan masuk tingkat rendah dengan biaya masuk minimum rendah. Berikut daftar program studi dan golongannya:

Tabel 3. 1 Tabel Diskretisasi Pilihan Program Studi

Tier1	S1 Teknik Telekomunikasi
	S1 Teknik Informatika
	S1 Teknik Industri
	S1 International ICT Business
	S1 MBTI
Tier2	S1 Teknik Elektro
	S1 Sistem Komputer
	S1 Teknik Fisika
	S1 Ilmu Komputasi
	S1 Ilmu Komunikasi
	S1 Sistem Informasi
	S1 Akuntansi
	S1 Administrasi Bisnis
	S1 Desain Komunikasi Visual
	S1 Desain Interior
	D3 Teknik Informatika
	D3 Teknik Telekomunikasi
Tier3	S1 Kriya Tekstil dan Mode
	S1 Desain Produk
	S1 Seni Rupa Murni
	D3 Teknik Komputer
	D3 Manajemen Informatika
	D3 Komputerisasi Akuntansi
	D3 Manajemen Pemasaran
D3 Perhotelan	

3.3 Pengukuran Kinerja

Kinerja dari klasifikasi diukur dalam akurasi, *recall* dan presisi. Dalam program *ant-miner* ini kinerja dari satu kali percobaan diukur berdasarkan akurasi testing rata-rata. Untuk mencari *rule* terbaik dari seluruh percobaan ukuran kinerja yang digunakan adalah akurasi *rule*, *recall rule* dan presisi *rule*.

Dalam kasus klasifikasi seleksi mahasiswa baru jalur non-tulis ini terdapat dua kelas sehingga kasus ini disebut sebagai *binary classification*. Perhitungan akurasi, *recall* dan presisi pada *binary classification* adalah sebagai berikut.

Tabel 3. 2 Tabel Confusion Matrix Untuk Pengukuran Kinerja

	Kinerja	
	Diprediksi Lulus	Diprediksi Tidak Lulus
Lulus	a	b
Tidak Lulus	c	d

Akurasi adalah ukuran ketepatan hasil prediksi klasifikasi dari *rule* yang telah dihasilkan.

$$\text{Akurasi} = \frac{a+d}{(\text{total sample})} \quad (3.3)$$

Recall adalah pengambilan data yang berhasil dilakukan terhadap bagian data yang relevan[11]. Rumus *Recall* adalah:

$$\text{Recall} = \frac{d}{c+d} \quad (3.4)$$

Dalam klasifikasi biner, *recall* disebut juga dengan *sensitivity*. Peluang munculnya data relevan dapat dilihat dengan *recall*.

Presisi adalah bagian data yang di ambil sesuai dengan informasi yang dibutuhkan[11]. Rumus presisi adalah:

$$\text{Presisi} = \frac{d}{b+d} \quad (3.5)$$

3.4 Deskripsi dan Analisis Data

Berikut deskripsi data yang digunakan pada penelitian tugas akhir ini:

1. Jumlah *record* data: 6686, penelitian tugas akhir ini menggunakan data dengan ukuran yang cukup besar untuk sebuah task data mining yaitu 6686 *record*.
2. Jumlah Atribut: 17 atribut. Atribut yang digunakan adalah hasil nilai inputan calon mahasiswa baru yang mendaftar ke Universitas Telkom melalui jalur non-tulis (JPA).
3. *Class Imbalance*: Data terdiri dari 4714 baris data dengan kelas 'lulus' dan 1972 baris data dengan kelas 'tidak lulus'. Kasus ini tidak dipengaruhi oleh *class imbalance* karena lebih fokus pada kelas 'lulus' dan tidak begitu memperhatikan kelas 'tidak lulus'.
4. Data tidak memiliki *missing value* atau *outlier*.

Kekurangan data adalah data yang diinput oleh siswa tidak semuanya lengkap, ada beberapa baris data dimana siswa tidak menginput nilai fisika.. Masalah nilai nol ini diatasi dengan diskretisasi pada pre-processing.

4. Implementasi Sistem

4.1 Analisis Hasil

Pengujian yang dilakukan pada percobaan tugas akhir ini adalah:

1. Menguji apakah terjadi *overfitting* dengan membandingkan hasil akurasi data training dengan data testing.

2. Melihat pengaruh dari setiap atribut yang diujikan terhadap parameter pengujian.

3. Mencari pasangan atribut yang menghasilkan kinerja terbaik yang diukur berdasarkan akurasi testing.

4. Mencari *rule-rule* terbaik yang dihasilkan oleh percobaan data mining berdasarkan pengukuran kinerja akurasi, *recall*, dan presisi *rule* terbaik.

Berikut tabel akurasi training dan testing hasil percobaan:

Ratio Training Testing	Jumlah Semut	Jumlah Rules	Akurasi Training Rata-Rata	Akurasi Testing Rata-Rata
50::50	40	10	67.59 %	68.73 %
50::50	40	15	67.43 %	68.44 %
50::50	50	10	68.05 %	69.28 %
50::50	50	15	66.89 %	67.72 %
50::50	60	10	67.18 %	68.74 %
50::50	60	15	67.63 %	69.21 %
60::40	40	10	67.51 %	68.95 %
60::40	40	15	66.41 %	68.44 %
60::40	50	10	68.29 %	69.16 %
60::40	50	15	67.17 %	67.84 %
60::40	60	10	67.42 %	69.48 %
60::40	60	15	68.38 %	68.80 %
70::30	40	10	68.49 %	69.37 %
70::30	40	15	69.71 %	67.93 %

70::30	50	10	68.72 %	68.72 %
70::30	50	15	69.00 %	68.82 %
70::30	60	10	70.92 %	67.78 %
70::30	60	15	67.69 %	68.78 %
80::20	40	10	68.62 %	68.82 %
80::20	40	15	68.35 %	67.93 %
80::20	50	10	68.85 %	68.72 %
80::20	50	15	69.71 %	68.81 %
80::20	60	10	67.88 %	67.78 %
80::20	60	15	69.27 %	68.78 %
90::10	40	10	66.00 %	66.05 %
90::10	40	15	70.62 %	69.88 %
90::10	50	10	69.35 %	68.85 %
90::10	50	15	69.57 %	68.74 %
90::10	60	10	68.39 %	67.94 %
90::10	60	15	67.79 %	67.69 %
Akurasi Training Terbaik :	70.92 %			
Akurasi Testing Terbaik :	69.88 %			
Akurasi Training Rata-rata keseluruhan:	68.30 %			
Akurasi Testing Rata-rata keseluruhan:	68.54 %			

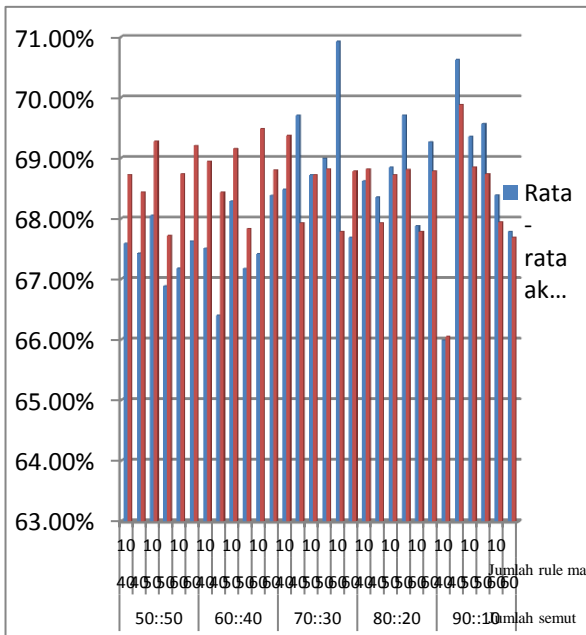
Tabel 4. 1Tabel Hasil Akurasi Training dan Testing

Dapat dilihat akurasi testing rata-rata terbaik ada pada pasangan atribut rasio data training: data testing 90:10, jumlah semut 40, dan jumlah *rule* maksimal 15 dengan akurasi testing rata-rata 69.88%. Akurasi training rata-rata terbaik ada pada pasangan atribut rasio data training: data testing 70:30, jumlah semut 60, dan jumlah *rule* maksimal 10.

Dari hasil akurasi testing tidak dapat dilihat jika ada pengaruh dari atribut yang diujikan yaitu jumlah semut, jumlah *rule* maksimal dan rasio data training dengan data testing terhadap ukuran akurasi testing.

Akurasi rata-rata keseluruhan juga menunjukkan bahwa percobaan ini berhasil untuk mendapatkan *rule-rule* dengan akurasi yang cukup baik dengan akurasi testing rata-rata seluruh percobaan 68.54%.

Berikut grafik hubungan akurasi training berbanding testing rata-rata dengan pasangan atribut yang telah diujikan:



Gambar 4. 1 Grafik Hasil Akurasi Training dan Testing

Berdasarkan grafik diatas selisih akurasi training dan testing rata-rata terbesar terjadi pada pasangan atribut jumlah semut 10, jumlah *rules* maksimal 15, dan rasio data training berbanding testing 70%:30% dengan selisih akurasi training dan testing rata-rata sebesar 3.14%. 3.14% adalah jumlah yang kecil untuk selisih akurasi training dan testing sehingga dapat disimpulkan bahwa dalam percobaan ini tidak terjadi *overfitting*.

Berikut tabel 10 *rule* dengan ukuran akurasi terbaik:

Tabel 4. 2 Tabel 10 Rule Terbaik Berdasarkan Akurasi

No	Akurasi Rule	Pasangan atribut	B.Ind 1	B.Ing 1	MTK 1	Fis 1	B.Ind 2	B.Ing 2	MTK 2	Fis 2	B.Ind 3	B.Ing 3	MTK 3	Fis 3	B.Ind 4	B.Ing 4	MTK 4	Fis 4	Pilihan	Kelas	
1	98.51%	40,10,50,50(5)	-	M	-	T	M	-	-	-	-	M	-	-	-	-	-	-	T	Tier1	L
2	98.41%	40,10,50,50(2)	-	M	-	-	M	-	-	-	-	-	-	-	-	-	-	-	M	-	L
3	98.41%	50,10,50,50(5)	-	M	-	-	M	-	-	-	-	-	-	-	-	-	-	-	M	-	L
4	98.41%	60,10,50,50(4)	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	L
5	97.99%	50,15,60,40(3)	-	-	-	M	-	M	-	-	-	-	-	-	-	-	-	M	-	-	L
6	97.97%	50,10,60,40(3)	-	-	-	-	M	-	M	-	-	M	-	-	-	-	-	-	-	-	L
7	97.97%	40,10,60,40(4)	-	-	-	-	M	-	M	-	-	M	-	-	-	-	-	-	-	-	L
8	97.92%	40,10,60,40(3)	-	M	-	-	M	-	M	-	-	-	-	-	-	-	-	-	M	-	L
9	97.76%	40,10,70,30(3)	-	-	-	-	-	-	M	-	-	M	-	-	-	-	-	-	-	-	L
10	97.36%	60,15,70,30(4)	-	M	-	-	-	-	M	-	-	-	-	-	-	-	-	-	M	-	L

Berikut tabel 10 *rule* dengan ukuran *recall* terbaik:

Tabel 4. 3 Tabel 10 Rule Terbaik Berdasarkan Recall

No	Recall Rule	Pasangan atribut	B.Ind 1	B.Ing 1	MTK 1	Fis 1	B.Ind 2	B.Ing 2	MTK 2	Fis 2	B.Ind 3	B.Ing 3	MTK 3	Fis 3	B.Ind 4	B.Ing 4	MTK 4	Fis 4	Pilihan	Kelas		
1	98.85%	40,10,50,50(3)	-	-	-	T	M	-	-	-	-	-	-	-	-	-	-	-	T	Tier2	L	
2	98.85%	50,10,50,50(2)	-	-	-	T	M	-	-	-	-	-	-	-	-	-	-	-	-	T	Tier3	L
3	98.70%	40,10,50,50(1)	-	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	-	-	L	
4	98.70%	40,10,50,50(4)	-	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	-	T	L	
5	98.70%	50,10,50,50(1)	-	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	-	T	L	
6	98.53%	50,10,60,40(5)	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	-	-	T	L	
7	98.53%	60,15,60,40(5)	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	-	-	T	L	
8	98.52%	40,15,60,40(1)	-	-	-	T	M	-	-	-	-	-	-	-	-	-	-	-	-	T	L	
9	98.51%	40,10,50,50(5)	-	M	-	T	-	-	-	-	M	-	-	-	-	-	-	-	-	T	L	
10	98.50%	50,15,70,30(1)	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	T	L	

Tabel 4. 4 Tabel 10 Rule Terbaik Berdasarkan Rataan Akurasi, Recall dan Presisi

No	Akurasi	Pasangan atribut	B.Ind 1	B.Ing 1	MTK 1	Fis 1	B.Ind 2	B.Ing 2	MTK 2	Fis 2	B.Ind 3	B.Ing 3	MTK 3	Fis 3	B.Ind 4	B.Ing 4	MTK 4	Fis 4	Pilihan	Kelas	
1	99.00%	40,10,50,50(5)	-	M	-	T	-	-	-	-	-	M	-	-	-	-	-	-	T	Tier1	L
2	98.94%	40,10,50,50(2)	-	M	-	-	M	-	-	-	-	-	-	-	-	-	-	-	M	-	L
3	98.94%	50,10,50,50(5)	-	M	-	-	M	-	-	-	-	-	-	-	-	-	-	-	M	-	L
4	98.70%	40,10,50,50(3)	-	-	-	-	M	-	M	-	-	-	-	-	-	-	-	-	M	-	L
5	98.66%	50,15,60,40(3)	-	-	-	-	M	-	M	-	M	-	-	-	-	-	-	-	-	-	L
6	98.65%	50,10,60,40(2)	-	-	-	-	M	-	M	-	-	-	-	-	-	-	-	-	M	-	L
7	98.65%	40,10,60,40(3)	-	-	-	-	M	-	M	-	-	-	-	-	-	-	-	-	M	-	L
8	98.62%	40,10,60,40(3)	-	M	-	-	M	-	M	-	-	-	-	-	-	-	-	-	M	-	L
9	98.58%	40,15,50,50(5)	-	-	-	-	-	-	M	-	-	-	-	-	-	-	-	-	-	T	L
10	98.57%	50,15,70,30(3)	-	-	-	-	-	M	-	M	-	-	-	-	-	-	-	-	M	T	L

5. Penutup
5.1 Kesimpulan

Berdasarkan hasil analisis pengujian yang dilakukan pada sistem perangkat lunak *ant-miner* ini maka dapat ditarik kesimpulan sebagai berikut:

1. Kombinasi atribut dengan tingkat akurasi testing rata-rata paling tinggi adalah pada kombinasi rasio data training banding testing 90:10, jumlah semut 40 dan jumlah *rule* maksimal 15 dengan akurasi testing rata-rata dari 5 kali percobaan 69.88%.
2. Tidak dapat dilihat hubungan langsung antara satu atribut dengan parameter hasil dikarenakan algoritma *ant-miner* memiliki ciri pembangkitan *rule* yang acak.
3. Akurasi rata-rata keseluruhan percobaan cukup bagus, 68.30% untuk training dan 67.54% untuk testing.

testing, yang berarti algoritma *ant-miner* cukup baik digunakan untuk klasifikasi data seleksi mahasiswa baru jalur non-tulis.

4. Berdasarkan hasil akurasi testing dan training dengan selisih rata-rata yang kecil (3.14%) maka dalam penelitian ini tidak terjadi *overfitting*.

5. Berdasarkan rule dengan ukuran rata-rata akurasi, *recall*, dan presisi, mata pelajaran dengan pengaruh paling besar terhadap kelulusan dilihat dari frekuensi kemunculannya pada rule-rule terbaik adalah Bahasa Indonesia semester 2 dan Matematika semester 2 dan 4. Sedangkan mata pelajaran dengan pengaruh paling minim adalah Bahasa Indonesia semester 1,3 dan 4, Matematika semester 1 dan 3, dan fisika semester 3.

5.2 Saran

1. *Ant-miner* telah berkembang dan memiliki berbagai ragam peningkatan, penelitian ke depan bisa menggunakan algoritma *ant-miner* yang lebih baru seperti *ant-miner+*, *e-ant-miner*, dan lain-lain.

2. Penggunaan library pembangkit nilai pseudo-random paralel seperti CuRand pada CUDA dengan algoritma *ant-miner* dapat dijadikan bahan penelitian ke depan.

Daftar Pustaka

- [1] Zaki, Meira. "DATA MINING AND ANALYSIS : Fundamental Concepts and Algorithms". Diambil dari <http://www.cs.rpi.edu/~zaki/PaperDir/DMABOOK.pdf>.
- [2] Wong, Summers, Ku, Xie. "Ant-colony optimization" PIC 10C Spring 2011 Diambil dari <http://www.math.ucla.edu/~wittman/10c.1.11s/Lectures/Raids/ACO.pdf>.
- [3] Tan, Steinbach, Kumar, "Data mining: Introduction" Diambil dari http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/cha_p1_intro.pdf
- [4] Tan, Steinbach, Kumar, "Data mining Classification: Basic Concepts, Decision Trees, and Model Evaluation" Diambil dari http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/cha_p4_basic_classification.pdf.
- [5] Sibaroni, (2002). "Analisis dan penerapan metode klasifikasi untuk pembangunan perangkat lunak sistem penerimaan mahasiswa baru jalur non-tulis." Institut Teknologi Bandung: Indonesia.
- [6] Martens, De Backer, Haesen, Baesens, Holvoet. "Ants Constructing Rule-Based Classifiers" Diambil dari <https://lirias.kuleuven.be/bitstream/123456789/123125/1/Antsconstructing.pdf>
- [7] Parpinelli, Rafael S., Heitor S. Lopes, dan Alex Alves Freitas. "Data mining with an ant-colony optimization algorithm." *Evolutionary Computation, IEEE Transactions on* 6.4 (2002): 321-332.
- [8] Smaldon, James, dan Alex A. Freitas. "A new version of the ant-miner algorithm discovering unordered rule sets." *Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM*, 2006.
- [9] Breslow, Leonard A., dan David W. Aha. "Simplifying decision trees: A survey." *The Knowledge Engineering Review* 12.01 (1997): 1-40.
- [10] Hussein A. Abbass, Ruhul A. Sarker dan Charles S. Newton. "Data mining A Heuristic Approach"
- [11] Powers, David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).
- [12] Metz, Charles E. "Basic principles of ROC analysis." *Seminars in nuclear medicine. Vol. 8. No. 4. WB Saunders*, 1978.