

# PENERAPAN TEKNIK KLASIFIKASI PADA SISTEM REKOMENDASI MENGGUNAKAN ALGORITMA GENETIKA

Rita Rismala<sup>1</sup>, Mahmud Dwi Sulistiyo<sup>2</sup>

Program Studi S1 Teknik Informatika, Fakultas Informatika  
Universitas Telkom<sup>1,2</sup>

Jl. Telekomunikasi - Terusan Buah Batu - Bandung 40257, Jawa Barat, Indonesia<sup>1,2</sup>  
ritaris@telkomuniversity.ac.id<sup>1</sup>, mahmuddwis@telkomuniversity.ac.id<sup>2</sup>

## Abstrak

Sistem rekomendasi yang dibangun dalam penelitian ini adalah sistem rekomendasi yang dapat memberikan rekomendasi sebuah *item* terbaik kepada *user*. Dari sisi *data mining*, pembangunan sistem rekomendasi satu *item* ini dapat dipandang sebagai upaya untuk membangun sebuah model *classifier* yang dapat digunakan untuk mengelompokkan data ke dalam satu kelas tertentu. Model *classifier* yang digunakan bersifat linier. Untuk menghasilkan konfigurasi model *classifier* yang optimal digunakan Algoritma Genetika (AG). Performansi AG dalam melakukan optimasi pada model klasifikasi linier yang digunakan cukup dapat diterima. Untuk *dataset* yang digunakan dengan kombinasi nilai parameter terbaik yaitu ukuran populasi 50, probabilitas *crossover* 0.7, dan probabilitas mutasi 0.1, diperoleh rata-rata akurasi sebesar 72.80% dengan rata-rata waktu proses 6.04 detik, sehingga penerapan teknik klasifikasi menggunakan AG dapat menjadi solusi alternatif dalam membangun sebuah sistem rekomendasi, namun dengan tetap memperhatikan pengaturan nilai parameter yang sesuai dengan permasalahan yang dihadapi.

Kata kunci:

sistem rekomendasi, klasifikasi, Algoritma Genetika

## Abstract

*In this study was developed a recommendation system that can recommend top-one item to a user. In terms of data mining, it can be seen as a problem to develop a classifier model that can be used to classify data into one particular class. The model used was a linear classifier. To produce the optimal configuration of classifier model was used Genetic Algorithm (GA). GA performance in optimizing the linear classification model was acceptable. Using the case study dataset and combination of the best*

*parameter value, namely population size 50, crossover probability 0.7 and mutation probability 0.1, obtained average accuracy 72.80% and average processing time of 6.04 seconds, so that the implementation of classification techniques using GA can be an alternative solution in developing a recommender system, due regard to setting the parameter value depend on the encountered problem.*

*Keywords:*

*Recommendation system, classification, Genetic Algorithm*

## I. PENDAHULUAN

Klasifikasi merupakan salah satu teknik dasar dalam bidang *data mining*. Teknik klasifikasi diperlukan untuk dapat melakukan pengenalan pola, prediksi nilai, hingga pengambilan keputusan. Dalam hal pengambilan keputusan, diperlukan sistem rekomendasi yang berperan dalam memberikan rekomendasi sejumlah *item* kepada *user* sehingga dapat membantu *user* dalam mengambil keputusan. Adapun sistem rekomendasi yang dibangun dalam penelitian ini adalah sistem rekomendasi yang dapat memberikan rekomendasi sebuah *item* kepada *user*. Pendekatan yang umumnya digunakan untuk membangun sebuah sistem rekomendasi diantaranya *content-based*, *collaborative*, dan *hybrid* (Adomavicius & Tuzhilin, 2005). Namun dari sisi *data mining*, pembangunan sistem rekomendasi satu *item* ini dapat dipandang sebagai upaya untuk membangun sebuah model *classifier* yang dapat digunakan untuk mengelompokkan data ke dalam satu kelas tertentu. Untuk menghasilkan model *classifier* dengan performansi tinggi, umumnya harus melalui mekanisme yang kompleks. Hal ini tentu berpengaruh terhadap semakin “mahal”-nya biaya komputasi yang dibutuhkan untuk membangun model

tersebut. Terkait dengan hal tersebut, maka pada penelitian ini dibangun model *classifier* yang sederhana, namun dapat menghasilkan performansi yang cukup baik dan tetap menjaga biaya komputasi yang “rendah”.

Agar model *classifier* memiliki performansi yang baik, maka dibutuhkan tahap pembelajaran untuk menghasilkan konfigurasi yang optimal. Jika dikaitkan dengan permasalahan di dunia kecerdasan buatan, proses ini dapat dipandang sebagai permasalahan optimasi. Oleh sebab itu, algoritma-algoritma optimasi dapat digunakan sebagai alternatif cara untuk membangun model *classifier*. Sejauh ini, Algoritma Genetika merupakan algoritma optimasi yang telah diaplikasikan secara luas dalam bidang *engineering* karena kemampuannya yang cukup *powerful* untuk menemukan solusi optimal dalam waktu yang relatif cepat (Hegerty & Kasprak, 2009). Selain itu, dalam penelitian (Widodo & Mahmudy, 2010) juga didapatkan bahwa waktu proses Algoritma Genetika untuk menyelesaikan permasalahan kombinatorial pada sistem rekomendasi wisata kuliner terbukti cukup cepat. Oleh karena itu, pada penelitian ini digunakan Algoritma Genetika untuk membangun sistem rekomendasi dengan menerapkan teknik klasifikasi.

## II. KAJIAN LITERATUR

### II.1. Sistem Rekomendasi

Sistem rekomendasi memberikan rekomendasi *item* kepada *user* dengan cara menganalisis data *user* serta informasi lain yang berhubungan dengan *user* tersebut, seperti informasi mengenai *user* lain yang memiliki preferensi yang sama atau informasi mengenai *environment* dari *user* tersebut (Sebastia, Garcia, Onaindia, & Guzman, 2009). Terdapat dua fungsi utama sistem rekomendasi: (1) Rekomendasi *top-N item*, yaitu sistem rekomendasi memberikan daftar N *item* yang paling direkomendasikan untuk *user*. (2) Prediksi *rating*, yaitu memprediksi *preference score* dari seorang *user* untuk *item* tertentu (Thai-Nghe, Drumond, Krohn-Grimberghe, & Schmidt-Thieme, 2010).

### II.2. Klasifikasi

Klasifikasi merupakan salah satu teknik dasar dalam *data mining* yang saat ini masih menjadi *open*

*topic* terutama untuk dimensi data yang besar (Yang & Wu, 2006). Proses pembelajaran dalam klasifikasi dilakukan untuk membangun model *classifier* berdasarkan data latih sehingga dapat digunakan untuk mengklasifikasikan data “masa depan” (Fung, Wang, & Yu, 2007). Jika dipandang dari dunia kecerdasan buatan, proses ini dapat dipandang sebagai permasalahan optimasi untuk mendapatkan model *classifier* yang dapat meminimumkan *error* klasifikasi.

### II.3. Algoritma Genetika

Algoritma Genetika (AG) telah diaplikasikan secara luas dalam bidang *engineering*. Pada AG, kandidat solusi untuk suatu permasalahan direpresentasikan dalam bentuk kromosom. Kemudian kromosom tersebut dievolusi dengan menerapkan tiga operator genetika yaitu rekombinasi, mutasi dan seleksi, untuk menghasilkan kandidat solusi yang lebih baik. Beberapa nilai parameter perlu diatur agar AG berjalan secara optimal, diantaranya: ukuran populasi, maksimum iterasi, jenis seleksi, jenis dan probabilitas rekombinasi, serta jenis dan probabilitas mutasi (Kachitvichyanukul, 2012). AG dalam bentuk *pseudocode* dapat dilihat pada Kode 1 (Hegerty & Kasprak, 2009).

#### Kode 1. Pseudocode Algoritma Genetika

```

Inisialisasi populasi secara acak.
Hitung nilai fitness dari populasi awal.
REPEAT
    Pilih pasangan orang tua berdasarkan nilai fitness.
    Rekombinasikan pasangan orang tua tersebut sehingga dihasilkan dua anak.
    Lakukan mutasi pada setiap anak hasil rekombinasi.
    Evaluasi anak hasil mutasi.
    Seleksi individu (orang tua/ anak) yang akan dipilih menjadi populasi baru.
UNTIL kondisi berhenti
    
```

## III. ANALISIS DAN PERANCANGAN

### III.1 Data

Sistem rekomendasi yang dibangun sebagai studi kasus dalam penelitian ini adalah sistem rekomendasi

pemilihan bidang keahlian tugas akhir. Data yang digunakan adalah data akademik mahasiswa Program Studi S1 Teknik Informatika Universitas Telkom dari tahun 2008 – 2013. Data ini berisi nilai-nilai mata kuliah yang telah diambil oleh mahasiswa beserta pilihan bidang keahlian dari setiap mahasiswa tersebut. Data bersifat *non-linearly separable*.

Sebelum diproses ke dalam sistem, data mentah di-*preprocess* terlebih dahulu dengan cara: (1) Mengubah data nilai indeks menjadi numerik, dengan ketentuan: A = 4; B = 3; C = 2; D = 1; E = 0. (2) Mengelompokkan mata kuliah (MK) menjadi MK dasar, MK bidang keahlian 1, MK bidang keahlian 2, dan MK bidang keahlian 3. (3) Menghitung rata-rata nilai MK untuk setiap kelompok. Atribut data hasil *preprocessing* dapat dilihat pada Tabel 1. Kemudian data dibagi menjadi data latih dan data uji dengan proporsi 308 data latih dan 131 data uji.

**Tabel 1. Atribut Data**

Nama	Kode	Nilai	Keterangan
MK dasar	$x_1$	Real [0 .. 4]	
MK bidang keahlian 1	$x_2$	Real [0 .. 4]	
MK bidang keahlian 2	$x_3$	Real [0 .. 4]	Masukan
MK bidang keahlian 3	$x_4$	Real [0 .. 4]	
Kelas	$T$	bidang keahlian 1 = 1, bidang keahlian 2 = 2, bidang keahlian 3 = 3	Keluaran

### III.2 Model Klasifikasi

Penelitian ini menggunakan model klasifikasi linier seperti pada persamaan (1), (2), (3), dan (4).

$$y_1 = k_{11}x_1 + k_{12}x_2 + \dots + k_{1n}x_n \quad \dots [1]$$

$$y_2 = k_{21}x_1 + k_{22}x_2 + \dots + k_{2n}x_n \quad \dots [2]$$

$$y_3 = k_{31}x_1 + k_{32}x_2 + \dots + k_{3n}x_n \quad \dots [3]$$

$$T \begin{cases} 1 & ; \max (y_1, y_2, y_3) = y_1 \\ 2 & ; \max (y_1, y_2, y_3) = y_2 \\ 3 & ; \max (y_1, y_2, y_3) = y_3 \end{cases} \quad \dots [4]$$

yang mana:

$y_1$ : nilai kelayakan untuk kelas 1

$y_2$ : nilai kelayakan untuk kelas 2

$y_3$ : nilai kelayakan untuk kelas 3

$x_1..x_n$ : atribut masukan

$k_{11}..k_{1n}$ : bobot atribut masukan 1 s.d. n untuk nilai kelayakan kelas 1

$k_{21}..k_{2n}$ : bobot atribut masukan 1 s.d. n untuk nilai kelayakan kelas 2

$k_{31}..k_{3n}$ : bobot atribut masukan 1 s.d. n untuk nilai kelayakan kelas 3

### III.3 Pemetaan Data Ke Dalam Sistem Rekomendasi

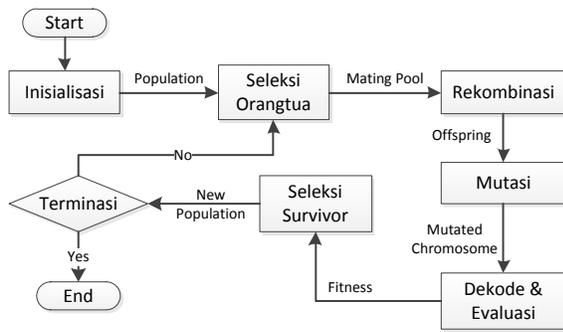
Pada sistem rekomendasi, data yang tersedia dipetakan menjadi *user*, *item*, dan *rating* (Thai-Nghe, Drumond, Krohn-Grimberghe, & Schmidt-Thieme, 2010). Dalam penelitian ini, mahasiswa dipetakan sebagai *user*, atribut data sebagai *item*, dan nilai atribut sebagai *rating*, seperti diilustrasikan pada Tabel 2. Proses untuk menentukan nilai kelayakan seperti dijelaskan pada sub-bab 2.2, dapat dipandang sebagai proses untuk menentukan rating untuk item  $y_1$ ,  $y_2$ , dan  $y_3$ . Kemudian berdasarkan *rating* tersebut dilakukan proses klasifikasi untuk menentukan kelas yang sesuai bagi *user*.

**Tabel 2. Matriks Pemetaan User – Item – Rating**

	$x_1$	$x_2$	$x_3$	$x_4$	$y_1$	$y_2$	$y_3$
<b>Mhs 1</b>	2,41	3,21	1,24	1,87	?	?	?
<b>Mhs 2</b>	3,12	1,98	1,76	4,00	?	?	?
<b>Mhs 3</b>	2,78	3,21	3,56	2,89	?	?	?
...							
<b>Mhs n</b>	3,56	3,21	3,11	2,98	?	?	?

### III.4 Perancangan Algoritma Genetika

Algoritma Genetika (AG) berperan dalam menemukan nilai-nilai yang tepat untuk semua bobot pada model klasifikasi, sehingga mampu untuk mengklasifikasikan setiap pola data masukan ke dalam kelas-kelas yang sesuai dengan akurasi sebaik mungkin. Skema AG yang digunakan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Skema Algoritma Genetika

### III.4.1 REPRESENTASI KROMOSOM

Representasi kromosom yang digunakan adalah representasi bilangan *real*, di mana setiap gen menyatakan nilai bobot untuk setiap atribut masukan pada model. Ilustrasi representasinya dapat dilihat pada Gambar 2.

$k_{11}$	$k_{12}$	..	$k_{1n}$	$k_{21}$	$k_{22}$	..	$k_{2n}$	$k_{31}$	$k_{32}$	..	$k_{3n}$
bobot atribut masukan 1 s.d. n untuk nilai kelayakan kelas 1				bobot atribut masukan 1 s.d. n untuk nilai kelayakan kelas 2				bobot atribut masukan 1 s.d. n untuk nilai kelayakan kelas 3			

Gambar 2. Representasi Kromosom pada GA

### III.4.2 SELEKSI ORANG TUA

Seleksi orangtua pada GA merupakan tahap memilih sejumlah pasangan kromosom untuk ditempatkan pada *mating pool*. Metode yang digunakan dalam penelitian ini adalah metode *roulette wheel*. Dengan metode ini, semakin tinggi nilai *fitness* suatu kromosom, maka peluang terpilihnya kromosom tersebut akan semakin besar.

### III.4.3 REKOMBINASI

Pada tahap rekombinasi, digunakan cara pindah silang dengan menggunakan 1 titik potong yang ditentukan secara acak dengan probabilitas tertentu, sebagaimana diilustrasikan pada Gambar 3.

Kromosom Orang Tua 1  
Kromosom Orang Tua 2

0,41	0,10	0,50	0,27	0,39	0,35
0,6	0,32	0,21	0,98	0,13	0,43

Kromosom Anak 1  
Kromosom Anak 2

0,41	0,10	0,21	0,98	0,13	0,43
0,6	0,32	0,50	0,27	0,39	0,35

Gambar 3. Rekombinasi Satu Titik

### III.4.4 MUTASI

Proses mutasi menggunakan metode mutasi *non-uniform* dengan probabilitas tertentu. Pada mutasi *non-uniform* dengan distribusi tetap, nilai gen hasil mutasi diperoleh dari nilai gen lama ditambah dengan bilangan *real* yang dibangkitkan secara acak dengan distribusi tertentu.

### III.4.5 SELEKSI SURVIVOR

Pada tahap seleksi *survivor*, ada dua metode yang digunakan, yaitu *generational replacement* dan *steady state*. Dengan *generational replacement*, populasi lama digantikan oleh kromosom hasil elitisme dan kromosom-kromosom lain hasil rekombinasi dan mutasi. Adapun dengan *steady state*, populasi lama digabung dengan populasi baru dan dipilih berdasarkan *fitness* yang tertinggi.

### III.4.6 TERMINASI

Untuk kriteria pemberhentian evolusi, terdapat dua kondisi, yaitu jika telah memenuhi maksimum generasi yang sudah ditentukan atau jika *fitness* terbaiknya tidak berubah-ubah selama sejumlah generasi berturut-turut.

### III.4.7 FUNGSI FITNESS

Kromosom dievaluasi dengan menggunakan rumus *fitness* yang didefinisikan pada persamaan (5).

$$fitness = \frac{nTrueData}{nAllData} \quad \dots [5]$$

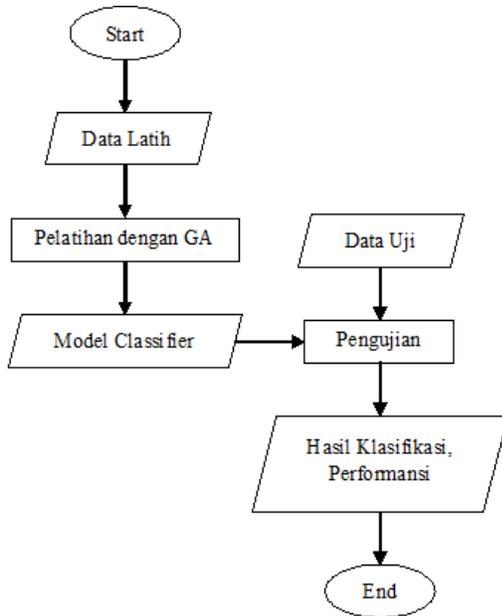
yang mana:

$nTrueData$  = jumlah data yang terklasifikasikan dengan benar

$nAllData$  = jumlah semua data

#### IV. HASIL DAN PEMBAHASAN

Selama percobaan, dilakukan pelatihan melalui proses evolusi GA dengan menggunakan data latih, dan pengujian untuk melihat performansi model *classifier* yang sudah dilatih dengan menggunakan data uji, seperti dapat dilihat pada Gambar 4. Tujuan dari percobaan ini adalah untuk menganalisis pengaruh nilai-nilai parameter terhadap jalannya proses evolusi serta untuk mendapatkan kombinasi nilai parameter yang menghasilkan akurasi yang baik.



**Gambar 4. Skema Percobaan**

Adapun parameter yang diobservasi meliputi ukuran populasi, probabilitas *crossover*, dan probabilitas mutasi. Secara umum, hasil observasi setiap nilai parameter tertentu akan berpengaruh pada setting nilai untuk observasi parameter selanjutnya. Untuk setiap kombinasi nilai parameter dilakukan observasi sebanyak 10 kali.

Keterangan pada Tabel 3 s.d. Tabel 8: Ukpop = ukuran populasi; Sel = seleksi *survivor* (1: *generational replacement*, 2: *steady state*); Pc = probabilitas *crossover*; Pm = probabilitas mutasi; accTr = rata-rata tingkat akurasi data latih (%); accTs = rata-rata tingkat akurasi data uji (%); accRata = rata-rata tingkat akurasi accTr dan accTs, acc = rata-rata tingkat akurasi accRata untuk setiap nilai parameter, dan Wkt = rata-rata waktu proses (detik).

#### IV.1 Observasi terhadap Ukuran Populasi

Pada observasi GA tahap pertama, belum ada informasi terkait nilai-nilai parameter yang sebaiknya digunakan. Oleh karenanya, ditentukan nilai-nilai yang sudah umum digunakan, yaitu probabilitas *crossover* yang nilainya mendekati 1, yaitu sebesar 0.9 dan probabilitas mutasi yang nilainya mendekati 0, yaitu sebesar 0.1.

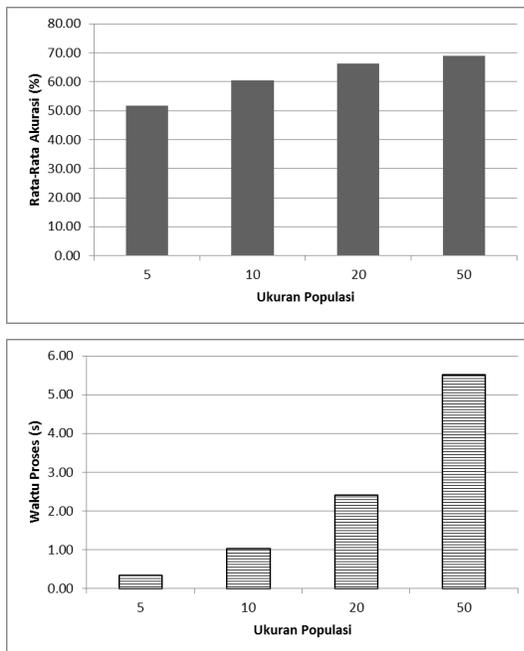
**Tabel 3. Hasil Observasi Pengaruh Ukuran Populasi terhadap Performansi Sistem**

Parameter	Performansi				
	Ukpop	Sel	accTr	accTs	accRata
5	1	51.69	50.76	51.41	0.35
10	1	60.84	59.92	60.57	0.94
20	1	69.9	67.56	69.20	2.35
50	1	70.39	66.72	69.29	4.91
5	2	55.36	52.67	54.56	0.31
10	2	63.44	60.69	62.62	1.12
20	2	66.92	64.89	66.31	2.44
50	2	75.26	71.3	74.08	6.11

Dari data pada Tabel 3, dilakukan pengolahan hingga diperoleh rata-rata untuk tingkat akurasi dan waktu proses yang dihasilkan dari beberapa nilai ukuran populasi. Tingkat akurasi di sini menunjukkan kemampuan model dalam mengklasifikasikan data, baik pada proses latih maupun uji. Sedangkan waktu proses di sini menunjukkan kecepatan sistem menuju kondisi konvergen, sehingga solusi yang dihasilkan sudah tidak berubah lagi.

**Tabel 4. Ringkasan Hasil Observasi untuk Setiap Ukuran Populasi**

Ukpop	5	10	20	50
Acc (%)	51.72	60.31	66.23	<b>69.01</b>
Wkt (s)	<b>0.33</b>	1.03	2.40	5.51



**Gambar 5. Pengaruh Ukuran Populasi terhadap Performansi AG**

Berdasarkan Tabel 4 dan Gambar 5, dalam hal tingkat akurasi terlihat bahwa semakin besar ukuran populasi, maka semakin bagus pula akurasi model. Dalam percobaan ini, nilai terbesar ukuran populasi yang digunakan adalah 50 dan pada nilai tersebutlah akurasi terbaik diperoleh. Hal ini dikarenakan ketika ukuran populasi yang digunakan semakin besar, maka jumlah *sample* yang diambil sebagai calon solusi pun semakin banyak. Hal tersebut tentu akan menambah kemungkinan diambilnya solusi yang terbaik di antara ruang solusi yang sangat besar. Namun, tidak ada jaminan bahwa jika ukuran populasi terus ditambah, maka akurasinya juga akan bertambah. Bisa jadi tingkat akurasi yang dihasilkan tidak akan beda jauh. Hal tersebut dapat diperkirakan dengan melihat akurasi yang tidak terlalu berbeda jauh antara ukuran populasi 20 dan 50.

Adapun jika melihat dari sisi waktu proses, terlihat bahwa semakin kecil ukuran populasi, semakin cepat pula proses pencarian yang dilakukan. Jika diasumsikan proses pencarian selalu berhenti karena telah mencapai maksimum generasi, hal ini sangat jelas karena dengan sedikitnya jumlah calon solusi yang dievaluasi, semakin cepat pula AG menyelesaikannya. Jika diasumsikan proses pencarian berhenti karena telah mencapai titik konvergen, maka dapat dipahami bahwa dengan

jumlah sample yang diamati lebih sedikit, variasi calon solusi akan lebih sedikit, sehingga mengakibatkan konvergensi prematur akan lebih mudah terjadi. Jadi, cepatnya proses pencarian tersebut tidak menjamin kualitas solusi yang bagus. Oleh karenanya, nilai 50 sebagai ukuran populasi ini diambil untuk proses observasi selanjutnya.

#### IV.2 Observasi terhadap Probabilitas Crossover

Pada observasi tahap kedua ini, ukuran populasi yang digunakan adalah 50, sebagaimana hasil observasi terbaik pada tahap sebelumnya. Adapun untuk probabilitas mutasi, digunakan nilai *default* seperti tahap sebelumnya, yakni 0.1.

Dari data pada Tabel 5, dilakukan pengolahan hingga diperoleh rata-rata untuk tingkat akurasi dan waktu proses yang dihasilkan dari beberapa nilai probabilitas *crossover*.

Berdasarkan Tabel 6 dan Gambar 6, dalam hal tingkat akurasi terlihat bahwa nilai probabilitas *crossover* ( $P_c$ ) terbaik adalah 0.7.  $P_c$  berpengaruh terhadap munculnya individu-individu baru atau dalam hal ini berpengaruh terhadap kemampuan eksploratif GA. Semakin besar  $P_c$ , GA akan lebih eksploratif atau menyebar dalam proses pencariannya. Artinya, lebih besar pula kemungkinan untuk mendapatkan solusi yang lebih baik dari generasi ke generasi. Namun, sejauh ini tidak ada nilai  $P_c$  terbaik yang pasti untuk semua kasus optimasi karena sangat sangat dipengaruhi oleh kondisi data yang dihadapi. Dalam percobaan ini, terlihat bahwa akurasi GA naik ketika nilai  $P_c$  ditambah dari 0.5 menjadi 0.7. Namun, kondisinya malah berbalik ketika  $P_c$  terus ditambah hingga nilai maksimalnya, yaitu 1. Penurunan tersebut dimungkinkan terjadi karena saat GA menjadi sangat eksploratif, kemampuan GA untuk mencapai kondisi konvergen di titik dengan solusi yang optimal menjadi terpengaruh. Bisa jadi dengan kondisi data klasifikasi yang digunakan di sini, GA justru menjadi lebih sulit menentukan titik optimum global ketika persebaran calon solusinya sangat luas.

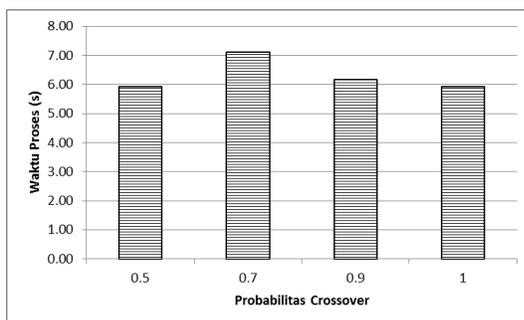
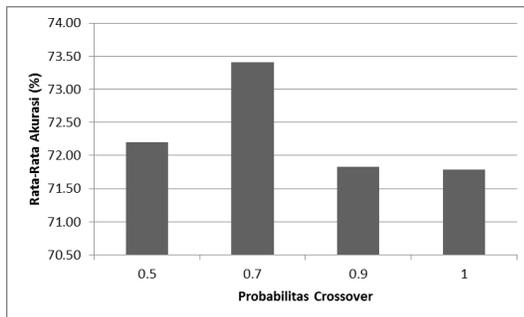
Adapun dari segi waktu proses, pada grafik terlihat hasil yang fluktuatif antara 0.5 sampai 0.9. Hal ini dapat dipandang bahwa  $P_c$  tidak berpengaruh terhadap waktu komputasi GA. Oleh sebab itu, nilai 0.7 diambil sebagai  $P_c$  untuk proses observasi selanjutnya.

**Tabel 5. Hasil Observasi Pengaruh Probabilitas Crossover terhadap Performansi Sistem**

Parameter	Performansi				
	Pc	Sel	accTr	accTs	accRata
0.5	1	70.81	70.53	70.73	5.60
0.7	1	72.56	71.76	72.32	6.10
0.9	1	70.68	70.61	70.66	5.34
1	1	70.75	69.39	70.34	5.46
0.5	2	74.29	72.21	73.67	6.23
0.7	2	75.1	73.05	74.49	8.10
0.9	2	73.8	71.15	73.01	6.98
1	2	73.44	72.75	73.23	6.39

**Tabel 6. Ringkasan Hasil Observasi untuk Setiap Nilai Pc**

Pc	0.5	0.7	0.9	1
Acc (%)	72.20	73.40	71.83	71.79
Wkt (s)	5.91	7.10	6.16	5.92



**Gambar 6. Pengaruh Probabilitas Crossover terhadap Performansi GA**

### IV.3 Observasi terhadap Probabilitas Mutasi

Pada observasi ini, digunakan nilai tetap untuk beberapa parameter, yaitu ukuran populasi = 50 dan probabilitas *crossover* = 0.7. Kedua nilai tersebut dipilih berdasarkan observasi di tahap sebelumnya, yang menghasilkan performansi terbaik di antara pilihan nilai lainnya.

Dari data pada Tabel 7, dilakukan pengolahan hingga diperoleh rata-rata untuk tingkat akurasi dan waktu proses yang dihasilkan dari beberapa nilai probabilitas mutasi.

Berdasarkan Tabel 8 dan Gambar 7, dari sisi keakuratan model terlihat jelas bahwa probabilitas mutasi (Pm) 0.1 memberikan hasil yang paling baik dibandingkan nilai-nilai yang lain. Proses mutasi pada GA memungkinkan sebuah kromosom menjadi lebih baik dengan mengubah beberapa nilai gen. Pm merupakan parameter untuk mengendalikan seberapa banyak gen yang akan dimutasi. Semakin besar nilai Pm, semakin besar kemungkinan dilakukannya mutasi untuk setiap gen, begitu pula sebaliknya. Mutasi pada banyak gen akan berdampak tidak baik karena dapat menyebabkan perubahan yang terlalu ‘radikal’. Bisa jadi kromosom yang akan dimutasi sudah merepresentasikan solusi yang baik. Namun, karena terlalu banyak gen yang berubah, maka solusi yang sudah baik tersebut menjadi ‘rusak’.

Adapun dari segi waktu proses, tidak adanya pola yang konsisten menunjukkan bahwa nilai Pm tidak berpengaruh terhadap kompleksitas algoritma. Artinya, besar kecilnya Pm hanya akan berpengaruh terhadap persebaran atau kualitas solusi yang dihasilkan, tidak pada kebutuhan algoritma akan waktu pemrosesan data. Dengan demikian, pada observasi ini diambil nilai 0.1 sebagai Pm yang memberikan performansi paling baik di antara nilai-nilai yang lainnya.

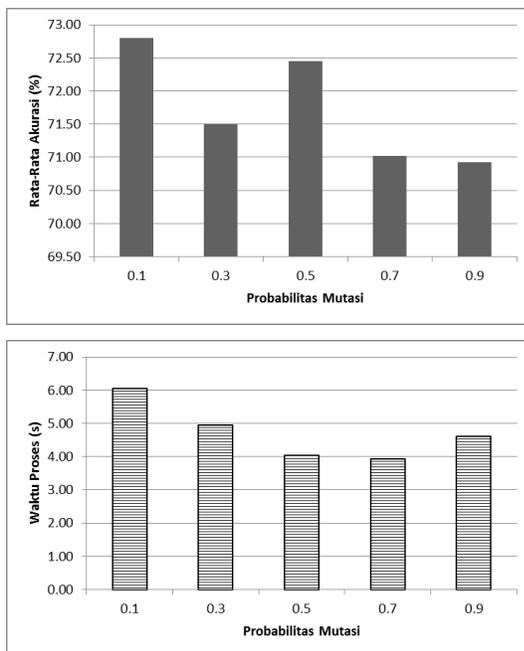
**Tabel 7. Hasil Observasi Pengaruh Probabilitas Mutasi terhadap Performansi Sistem**

Parameter	Performansi				
	Pm	Sel	accTr	accTs	accRata
0.1	1	70.88	68.4	70.14	5.20
0.3	1	72.11	71.68	71.98	5.11
0.5	1	69.12	68.78	69.02	3.50
0.7	1	69.64	71.07	70.07	4.17
0.9	1	70.52	68.47	69.91	4.31

0.1	2	76.72	72.52	75.47	6.88
0.3	2	71.53	69.77	71.00	4.79
0.5	2	76.49	74.43	75.88	4.56
0.7	2	72.08	71.76	71.98	3.67
0.9	2	71.72	72.44	71.93	4.89

**Tabel 8. Ringkasan Hasil Observasi untuk Setiap Nilai Pm**

Pm	0.1	0.3	0.5	0.7	0.9
Acc (%)	72.80	71.49	72.45	71.03	70.92
Wkt (s)	6.04	4.95	4.03	3.92	4.60



**Gambar 7. Pengaruh Probabilitas Mutasi terhadap Performansi GA**

Berdasarkan hasil observasi, dapat dilihat bahwa penerapan teknik klasifikasi menggunakan AG dapat menjadi solusi alternatif dalam membangun sebuah sistem rekomendasi, namun dengan tetap memperhatikan pengaturan nilai parameter yang sesuai dengan permasalahan yang dihadapi. Dalam penelitian ini, dengan menggunakan kombinasi nilai parameter terbaik yaitu ukuran populasi 50, probabilitas *crossover* 0.7, dan probabilitas mutasi

0.1, didapatkan rata-rata akurasi sebesar 72.80% dengan rata-rata waktu proses 6.04 detik. Hal ini menunjukkan bahwa performansi AG dalam membangun model *classifier* untuk data yang *non-linearly separable* cukup dapat diterima.

## V. KESIMPULAN DAN SARAN

Pada AG, parameter ukuran populasi, probabilitas *crossover*, dan probabilitas mutasi berpengaruh terhadap hasil yang diperoleh dari proses evolusi dipandang dari segi tingkat akurasi. Berdasarkan percobaan yang dilakukan, nilai terbaik untuk ketiga parameter tersebut berturut-turut adalah 50, 0.7, dan 0.1. Adapun jika dipandang dari segi waktu proses, hanya ukuran populasi yang berpengaruh terhadap kompleksitas proses komputasi, sedangkan probabilitas *crossover* dan mutasi relatif tidak berpengaruh terhadap waktu proses yang dibutuhkan.

Dari percobaan dan analisis yang telah dilakukan, performansi AG dalam melakukan optimasi pada model klasifikasi linear yang digunakan cukup dapat diterima. Untuk *dataset* yang digunakan dengan kombinasi nilai parameter terbaik, diperoleh rata-rata akurasi sebesar 72.80% dengan rata-rata waktu proses 6.04 detik, sehingga penerapan teknik klasifikasi menggunakan AG dapat menjadi solusi alternatif dalam membangun sebuah sistem rekomendasi, namun dengan tetap memperhatikan pengaturan nilai parameter yang sesuai dengan permasalahan yang dihadapi.

Untuk keperluan pengembangan studi ke depannya, dapat digunakan model klasifikasi yang lebih kompleks dan dapat menangani berbagai kondisi persebaran data, baik yang *linearly separable* maupun *non-linearly separable*. Selain itu, dapat pula menggunakan metode optimasi lain, baik yang berbasis *Evolutionary Algorithms* maupun *Swarm Intelligence*.

## REFERENSI

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6), 734-749.

- Hegerty, B., Hung, C. C., & Kasprak, K. (2009, November). A comparative study on differential evolution and genetic algorithms for some combinatorial problems. In Proceedings of 8th Mexican International Conference on Artificial Intelligence (pp. 9-13).
- Widodo, A. W., & Mahmudy, W. F. (2010). Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner. *Jurnal Ilmiah Cursor*, 3(4), 205-2011.
- Sebastia, L., Garcia, I., Onaindia, E., & Guzman, C. (2009). e-Tourism: a tourist recommendation and planning application. *International Journal on Artificial Intelligence Tools*, 18(05), 717-738.
- Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., & Schmidt-Thieme, L. (2010). Recommender system for predicting student performance. *Procedia Computer Science*, 1(2), 2811-2819.
- Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04), 597-604.
- Fung, B., Wang, K., & Yu, P. S. (2007). Anonymizing classification data for privacy preservation. *Knowledge and Data Engineering, IEEE Transactions on*, 19(5), 711-725.
- Kachitvichyanukul, V. (2012). Comparison of three evolutionary algorithms: GA, PSO, and DE. *Industrial Engineering and Management Systems*, 11(3), 215-223.