

SISTEM REKOMENDASI PENGAMBILAN MATA KULIAH DENGAN MENGGUNAKAN ALGORITMA GENETIKA DAN METODE *CONSTRAINT SATISFACTION*

I Made Arya Beta Widyatmika¹, Gloria Virginia², Rosa Delima³

^{1,2,3}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia
¹aryabwidyatmika@ti.ukdw.ac.id, ²virginia@ti.ukdw.ac.id, ³rosadelimaplg@ti.ukdw.ac.id

Abstract

Registration are routine activities performed by students before the start of the new school year. The number of classes offered at different times and professors require students to be able to determine the best combination of classes in the semester according to the curriculum that will be pursued. This research aims at establishing a system on courses that can help students in selecting subjects and making the schedule. This research applied a Genetic Algorithm combined with a Constraint Satisfaction Problem. The data used in this research are the data schedule of courses offered in the second semester of 2015/2016 in the Department of Information Engineering, UKDW. The analysis of testing results showed that by using genetic algorithms and Constraint Satisfaction Problem we are able to optimize the system recommendation with 100% accuracy, and that the system can produce schedules without violating the given constraints. In contrast, if only genetic algorithms are used, the accuracy obtained is 98%, and the system can produce schedules without violating the given constraints.

Keywords : *Genetic Algorithm, Constraint Satisfaction Problem, Recommendation System, Decision Support System,*

1. Pendahuluan

Penjadwalan merupakan hal penting yang sering dilakukan pada setiap kegiatan, baik dalam lingkup perusahaan, transportasi, sekolah, perguruan tinggi dan lain sebagainya. Pada perguruan tinggi, penjadwalan mata kuliah merupakan kegiatan wajib yang harus dilakukan setiap semester untuk dapat terlaksananya sebuah proses belajar mengajar. Penjadwalan mata kuliah di sebuah perguruan tinggi merupakan masalah yang sulit untuk dipecahkan (Gunawan & Poh, 2007). Permasalahan penjadwalan ini juga banyak ditemukan di universitas-universitas lain di seluruh dunia (Kumar, Husain, Upreti, & Gupta, 2010).

Dalam pengambilan mata kuliah, setiap mahasiswa wajib merencanakan mata kuliah yang akan diambil pada semester tersebut. Banyaknya kelas yang ditawarkan dengan waktu dan dosen yang berbeda mengharuskan mahasiswa untuk dapat menentukan kombinasi kelas yang paling baik menurut kurikulum pada semester yang akan ditempuh. Masalah penjadwalan perkuliahan

berbeda-beda dari satu universitas ke universitas yang lain, bahkan dari satu jurusan ke jurusan lain pada universitas yang sama (Buliali, Herumurti, & Wiriapradja, 2008). Dengan kemajuan ilmu pengetahuan di bidang komputasi cerdas, maka masalah penjadwalan oleh dosen dan pengambilan mata kuliah oleh mahasiswa dapat diotomatisasi sehingga dapat memberikan solusi dan rekomendasi yang optimal sesuai dengan batasan-batasan dan syarat yang sudah ditentukan.

Pada penelitian ini, studi kasus yang diambil adalah pada jurusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana. Oleh karena itu dibutuhkan sebuah sistem rekomendasi pengambilan mata kuliah untuk menyelesaikan masalah pengambilan mata kuliah berikut jadwal kuliah oleh mahasiswa Jurusan Teknik Informatika UKDW. Dalam penelitian ini sistem rekomendasi mata kuliah dan jadwal perkuliahan dibangun menggunakan algoritma Genetika dan *Constraint Satisfaction*.

2. Landasan Teori

2.1. Algoritma Genetika

Algoritma genetika adalah algoritma pencarian yang didasari atas mekanisme dari seleksi alam yang lebih dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. Hanya individu-individu yang kuat akan mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika, proses perkembangbiakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: “Bagaimana mendapat keturunan yang lebih baik” (Basuki, 2003).

Secara umum dalam algoritma genetika terdapat 5 (lima) proses, yaitu : pembentukan populasi awal, perhitungan nilai *fitness*, seleksi, regenerasi (*crossover* dan mutasi), dan penciptaan populasi baru hasil regenerasi (Setemen & Purnomo, 2008). Sebuah representasi algoritma genetika yang efektif dan nilai *fitness* yang bermakna adalah kunci keberhasilan dalam aplikasi genetika (Witary, Rachmat, & Inayatullah, 2005).

2.1.1 Populasi

Populasi adalah kumpulan kromosom. Kromosom ini dibentuk dari komponen-komponen penyusun yang disebut gen dan nilainya dapat berupa bilangan numerik, biner, simbol, ataupun karakter tergantung dari permasalahan yang ingin diselesaikan (Chairi, 2010). Berikut ini adalah contoh kromosom yang digunakan dalam menghasilkan individu awal pada sistem, kromosom berisi informasi :

1. Semester (s)
2. Maksimal pengambilan sks (k)
3. Mata kuliah wajib yang harus diambil (w)
4. Maksimal kelas pada satu mata kuliah (m)

Susunan *Course Object* pada kromosom yang baru adalah: **<s, k, w, m>**

2.1.2 Individu

Individu merupakan kumpulan gen dalam sistem algoritma genetika, bisa dikatakan sama dengan kromosom. Individu dalam algoritma genetika dapat juga menyatakan salah satu kemungkinan solusi yang dicari (Chairi, 2010).

Setelah kromosom terbentuk, maka langkah selanjutnya adalah memilih individu induk

yang akan digunakan untuk kawin silang kromosom. Individu induk yang digunakan merupakan individu yang sudah terpilih melalui berbagai macam proses.

2.1.3 Nilai *Fitness*

Nilai *fitness* merupakan suatu ukuran baik tidaknya suatu solusi yang dinyatakan sebagai satu individu, atau dengan kata lain nilai *fitness* menyatakan nilai dari fungsi tujuan. Algoritma genetika mempunyai tujuan untuk memaksimalkan nilai *fitness* atau mencari nilai *fitness* maksimal. Untuk permasalahan minimalisasi, nilai *fitness* adalah inversi dari nilai minimal yang diharapkan (Basuki, 2003). Proses inversi dapat dilakukan dengan rumus berikut :

$$fitness = \frac{1}{1+pinalti} \quad (1)$$

$$fitnes = \frac{1}{1+\sum Bobot\ pinalti.\sum Indikator\ pinalti} \quad (2)$$

Dari cara menghitung nilai *fitness* di atas maka nilai *fitness* yang terbaik adalah 1 karena tidak adanya *constraint* yang dilanggar.

2.1.4 Seleksi

Seleksi merupakan proses pemilihan kromosom dari generasi lama untuk dijadikan orang tua yang akan saling kawin silang untuk membentuk kromosom baru di generasi baru, dalam hal ini kita menggunakan seleksi roda *roulette* (*roulette wheel selection*). Pemilihan dua buah kromosom sebagai orang tua (*parent*) dilakukan secara proposional sesuai dengan nilai *fitness*nya. Kromosom yang memiliki nilai *fitness* tertinggi akan menempati potongan yang lebih besar pada lingkaran daripada kromosom dengan nilai *fitness* yang lebih rendah (Brigida, 2015). Menghitung probabilitas seleksi kromosom dan nilai *fitness* populasi dilakukan menggunakan persamaan :

$$Prob\ seleksi\ krom(i) = \frac{nilai\ fitness(i)}{nilai\ fitness\ populasi} \quad (3)$$

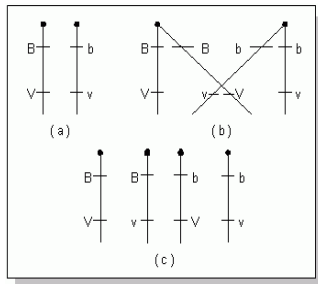
$$fitness\ populasi = \sum fitness\ kromosom \quad (4)$$

2.1.5 *Crossover* (Persilangan)

Perkawinan silang (*crossover*) adalah salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Perkawinan silang dilakukan dengan

melakukan pertukaran gen dari dua induk yang berbeda secara acak (Chairi, 2010).

Berikut adalah contoh *crossover* pada kromosom B – V, di mana Gambar (a) menunjukkan 2 buah kromosom induk, Gambar (b) merupakan proses *crossover*, dan Gambar (c) adalah 2 buah kromosom baru. Proses *Crossover* dapat dilihat pada gambar berikut:



Gambar 1. Contoh *Crossover* oleh Yatim (1980)

(<http://aimyaya.com/id/komputer/operator-algoritma->

2.1.6 Mutasi

Mutasi diperlukan untuk mengembalikan informasi bit yang hilang akibat pindah silang. Mutasi diterapkan dengan probabilitas yang sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi gen pada individu yang unggul akan dirusak.

2.2 Constraint Satisfaction Problem (CSP)

Constraint satisfaction problem (CSP) merupakan sebuah pendekatan dari problem yang bersifat matematis dengan tujuan menemukan keadaan atau obyek yang memenuhi sejumlah persyaratan (*constraint*) yang sudah ditetapkan (Ratnawati, Liliana, Regasari, & Muflikhah, 2012). Algoritma yang digunakan dalam *Constraint Satisfaction* adalah *Backtracking* yang dilakukan pada semua jadwal yang masih ada *constraint* yang terlanggar.

CSP terdiri dari komponen-komponen berikut :

- Variabel merupakan penampung yang dapat diisi berbagai nilai.
- *Constraint* merupakan suatu aturan yang ditentukan untuk mengatur nilai yang boleh dimasukkan ke variabel atau kombinasi variabel.

- Domain merupakan kumpulan nilai legal yang dapat diisi ke variabel.
- Solusi merupakan *assignment* nilai-nilai dari domain ke setiap variabel dengan tidak ada *constraint* yang dilanggar.

Constraint dari CSP terbagi atas 2 (dua) kategori yaitu :

1. *Hard Constraint* merupakan batasan yang harus dipenuhi dan tidak boleh dilanggar dalam pembuatan penyelesaian masalah. Dalam perhitungan nilai *fitness*, *hard constraint* akan memiliki nilai penalti 2.
2. *Soft Constraint* merupakan batasan tambahan yang biasanya merupakan sebuah permintaan. Dalam perhitungan nilai *fitness*, *soft constraint* akan memiliki nilai penalti 1.

2.2.1 Backtracking

Algoritma *backtracking* adalah algoritma yang paling umum dan banyak dipakai dalam aplikasi perhitungan untuk *Constraint Satisfaction Problem*. Pada dasarnya *backtracking* secara perlahan-lahan mencoba mengembangkan kumpulan *label* yang masih belum lengkap untuk beberapa variabel menuju ke kumpulan *label* yang lengkap, dengan secara berulang-ulang memilih nilai bagi variabel yang lain yang konsisten dengan solusi yang masih belum lengkap. Algoritma *backtracking* adalah pendekatan *brute-force* yang sudah diperbaiki, yang secara sistematis mencari sebuah solusi atas sebuah masalah di antara semua pilihan yang ada (Gurari, 1999).

2.3 Kombinasi Algoritma Genetika dengan CSP

Dari penelitian sebelumnya yang dilakukan oleh Buliali, Herumurti dan Wiriapradja (2008) tentang "Penjadwalan Mata Kuliah Dengan Menggunakan Algoritma Genetika dan Metode *Constraint Satisfaction*" bisa disimpulkan bahwa sistem pengolahan jadwal yang diteliti mampu menangani penjadwalan pada waktu tertentu dan telah mampu mengolah data mata kuliah yang ditawarkan, serta mampu menghasilkan jadwal tanpa ada *constraint* yang terlanggar (Buliali, Herumurti, & Wiriapradja, 2008)

Constraint Satisfaction Problem (CSP) diterapkan pada saat sebelum pemberian nilai *fitness* dengan menggunakan kromosom yang dihasilkan pada metode genetika. Kromosom yang dihasilkan pada metode Algoritma Genetika dapat diproses

dengan metode *Constraint Satisfaction Problem* sehingga dapat ditemukan batasan-batasan pada penjadwalan yang harus dipenuhi dengan cepat dan akurat.

3. Hasil dan Pembahasan

Dalam melakukan analisis apakah sistem dapat merekomendasikan mata kuliah yang baik sesuai kurikulum dan riwayat mata kuliah *user*, penulis akan melampirkan riwayat mata kuliah dan rekomendasi sistem dari beberapa *user*. Data *user* diambil secara acak dari beberapa mahasiswa program studi Teknik Informatika UKDW semester 4, 6 dan 8. Lampiran tersebut akan dinilai oleh beberapa dosen wali untuk mengetahui apakah sistem sudah dapat merekomendasikan mata kuliah yang benar. Pada setiap lampiran *user* akan disertakan lembar penilaian untuk memudahkan dosen wali dalam menilai rekomendasi sistem.

Skala penilaian sistem adalah dari skor 1 sampai 5. Kriteria penilaian tersebut adalah sebagai berikut:

- 5 = sangat baik, apabila rekomendasi mata kuliah sangat sesuai dengan kurikulum
- 4 = baik, apabila rekomendasi mata kuliah sesuai dengan kurikulum
- 3 = cukup, apabila sebagian besar rekomendasi sudah sesuai dengan kurikulum, namun ada sedikit mata kuliah yang belum sesuai
- 2 = kurang, apabila rekomendasi mata kuliah tidak sesuai dengan kurikulum
- 1 = sangat kurang, apabila semua rekomendasi mata kuliah tidak sesuai dengan kurikulum.

Hasil seluruh penilaian rekomendasi sistem oleh dosen wali terhadap 25 data uji akan digunakan untuk menentukan apakah sistem sudah dapat merekomendasikan mata kuliah yang baik sesuai kurikulum dan riwayat mata kuliah *user*. Tabel berikut akan memaparkan total hasil penilaian terhadap tiga aspek pengamatan dari beberapa dosen wali :

Tabel 1.
Hasil penilaian rekomendasi mata kuliah mahasiswa

No	Aspek Pengamatan	Total Skor Penilaian					Rata-Rata (kategori)
		1	2	3	4	5	
1.	Kecocokan rekomendasi mata kuliah wajib pada	0	0	0	17	8	4,32 (baik)

	semester yang ditempuh.						
2.	Kecocokan rekomendasi mata kuliah pilihan, menurut jumlah konsentrasi yang paling banyak ditempuh pada semester sebelumnya. (jika mahasiswa semester 4, maka semua konsentrasi ditampilkan pada rekomendasi karena belum pernah menempuh mata kuliah pilihan sebelumnya)	0	0	6	10	9	4,12 (baik)
3.	Kecocokan rekomendasi untuk mata kuliah wajib yang mendapatkan nilai < B- pada semester sebelumnya jika ada.	0	0	5	8	12	4,28 (baik)

Dalam pengujian selanjutnya dilakukan untuk mengetahui jumlah kromosom terbaik dalam satu populasi. Percobaan dilakukan dengan menjalankan sistem dengan beberapa jumlah kromosom yang telah ditetapkan. Dalam penetapan jumlah kromosom, penulis menggunakan dasar penelitian yang telah dilakukan sebelumnya dari beberapa jurnal ilmiah dengan algoritma Genetika. Pada tabel berikut akan dipaparkan jumlah kromosom yang telah digunakan dari beberapa sumber penelitian :

Tabel 2.
Tabel jumlah kromosom dari beberapa sumber

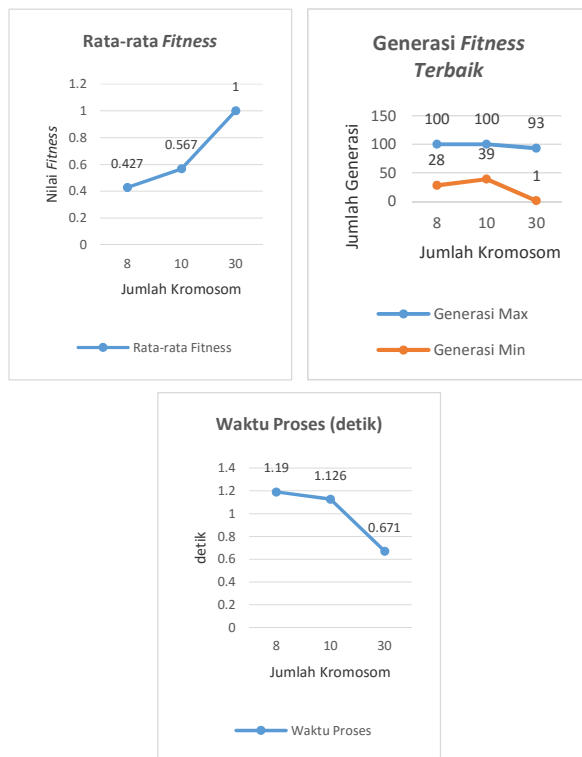
Jumlah Kromosom	Sumber
8	(Widyadana & Pamungkas, 2002)
10	(Mawaddah & Mahmudy, 2006), (Arifudin, 2011), dan (Buliali, Herumurti, & Wiriapradja, 2008)
30	(Saputro, 2004)

Pengujian dilakukan sebanyak 10 kali untuk setiap jumlah kromosom. Dalam setiap percobaan akan ditetapkan probabilitas *crossover* 0,5, probabilitas mutasi 0,15, dan maksimal generasi

100. Pada tabel berikut ditampilkan hasil pengujian yang dilakukan.

Tabel 3.
Hasil pengujian jumlah kromosom dari 10 kali percobaan

Jumlah Kromosom	Rata-rata Nilai Fitness	Generasi Fitness Terbaik		Rata-rata Waktu Proses
		Terkecil	Terbesar	
8	0,427	28	100	1,19 s
10	0,567	39	100	1,126 s
30	1	1	93	0,671 s



Gambar 2 Diagram hasil pengujian jumlah kromosom

Dari perbandingan data di atas, dapat diambil kesimpulan bahwa dengan jumlah kromosom 30 maka akan mendapatkan rata-rata nilai fitness lebih baik dibanding jumlah kromosom 8 dan 10.

Dalam analisis untuk mengetahui nilai probabilitas *crossover* (PC) dan probabilitas mutasi (PM), penulis akan melakukan beberapa percobaan. Penulis telah mengumpulkan beberapa sumber dari penelitian sebelumnya untuk mengetahui nilai PC dan PM yang akan diterapkan pada sistem. Kombinasi nilai PC dan PM dari beberapa sumber adalah sebagai berikut:

1. PC = 0,5 dan PM = 0,03 (Arifudin, 2011)
2. PC = 0,6 dan PM = 0,01 (Setemen, 2009)
3. PC = 0,8 dan PM = 0,2 (Saputro, 2004)

Setiap percobaan akan dilakukan pengujian sistem sebanyak 15 kali pengulangan dengan maksimal generasi 50.000. Dalam semua percobaan akan menggunakan 10 akun *user* dengan komposisi mata kuliah dan *soft constraint* yang berbeda. Hasil percobaan dari ketiga kombinasi di atas dapat dilihat pada tabel berikut :

Tabel 4.
Hasil analisis dengan PC 0,5 dan PM 0,03

User Nim	Jml SC	Generasi Fitness terbaik		Rata-rata fitness terbaik	Rata-rata waktu proses	Persentase fitness 1
		Min	Max			
71130048	2	16	799	1	0,975	100%
71130101	4	717	36.225	1	20,79	100%
71130105	2	36	50.000	0,938	27,285	93%
71130010	0	2	3.000	1	2,253	100%
71130037	0	1	8.063	1	6,39	100%
71140021	1	5.813	45.946	1	47,05	100%
71130084	1	1	1.367	1	1,18	100%
71140062	3	367	40.685	1	30,48	100%
71140023	4	50.000	50.000	0,0767	133,6	0%
71130096	4	535	50.000	0,937	29,69	93%
Rata-rata Total		5.748,8	28.608,5	0,89517	29,969	88,6%

Tabel 5.
Hasil analisis dengan PC 0,6 dan PM 0,01

User Nim	Jml SC	Generasi Fitness terbaik		Rata-rata fitness terbaik	Rata-rata waktu proses	Persentase fitness 1
		Min	Max			
71130048	2	1	617	1	1,083	100%
71130101	4	233	50.000	0,94	47,996	93%
71130105	2	1	21.382	1	17,914	100%
71130010	0	124	50.000	0,942	12,647	93%
71130037	0	5	11.782	1	17,555	100%
71140021	1	657	50.000	0,815	64,338	80%
71130084	1	1	1.178	1	1,451	100%
71140062	3	787	50.000	0,878	52,978	87%
71140023	4	50.000	50.000	0,081	131,8	0%
71130096	4	1.832	50.000	0,816	36,212	80%
Rata-rata Total		5.364,1	33.495,9	0,8472	38,397	83,3%

Tabel 6.
Hasil analisis dengan PC 0,8 dan PM 0,2

User Nim	Jml SC	Generasi Fitness terbaik		Rata-rata fitness terbaik	Rata-rata waktu proses	Persentase fitness 1
		Min	Max			
71130048	2	1	48	1	0,854	100%
71130101	4	59	1.477	1	2,912	100%
71130105	2	1	2.760	1	2,388	100%
71130010	0	1	439	1	1,164	100%
71130037	0	1	1.171	1	1,933	100%
71140021	1	164	5.574	1	3,515	100%
71130084	1	3	236	1	0,702	100%
71140062	3	129	2.353	1	2,416	100%
71140023	4	50.000	50.000	0,081	129,4	0%
71130096	4	20	3.881	1	1,967	100%
Rata-rata Total		5037,9	6.793,9	0,9081	14,725	90%

Tabel 7 menampilkan hasil keseluruhan dari ketiga percobaan di atas.

Tabel 7.
Hasil rata-rata tiga percobaan

Kombinasi PC dan PM	Rata-rata Nilai Fitness Terbaik	Rata-rata Maksimal Generasi Fitness Terbaik	Rata-rata Waktu Proses (second)	Persentase fitness 1 dari 15 percobaan
PC = 0,5; PM = 0,03	0,89517	28.608,5	29,969	88,6%
PC = 0,6; PM = 0,01	0,8472	33.495,9	38,397	83,3%
PC = 0,8; PM = 0,2	0,9081	6.793,9	14,7251	90%

Dari ketiga hasil percobaan seperti tampak pada Tabel 7 dapat diketahui dengan nilai kombinasi PC=0,8 dan PM=0,2 memungkinkan untuk mendapatkan nilai *fitness* terbaik akan semakin besar dengan jumlah generasi yang lebih sedikit dan dengan waktu proses yang lebih singkat. Dengan demikian maka dapat disimpulkan dengan nilai PC=0,8 dan PM=0,2 akan didapatkan hasil yang lebih baik dibandingkan dua kombinasi nilai yang lain.

Dengan data hasil percobaan pada Tabel 6 dapat dilihat pada semua kasus kecuali nomor 9 dengan nim 71140023, proses Genetika mampu memperoleh nilai *fitness* 1 dengan jumlah iterasi dari 1 sampai 5.574. Untuk kasus seperti nomor 9, di mana *soft constraint* yang ditetapkan sudah pasti terlanggar karena hari dan waktu pada *soft constraint* yang ditetapkan sama dengan hari dan waktu pada jadwal mata kuliah yang dipilih, proses genetika akan berhenti pada jumlah iterasi generasi maksimal. Dari data pada Tabel 6 maka dapat diketahui nilai maksimal generasi yang paling baik adalah pada *range* 5.000 sampai 6.000. Jumlah maksimal generasi dibatasi karena jumlah generasi akan mempengaruhi waktu proses jalannya program. Semakin besar iterasi yang dilakukan maka semakin lama proses Genetika berjalan.

Pengujian selanjutnya adalah untuk mengetahui berapa persen akurasi sistem dengan proses genetika dan CSP dalam memberikan rekomendasi jadwal dari mata kuliah yang telah dipilih. Pada proses pengujian akan dilakukan percobaan terhadap 10 *user* seperti pada percobaan sebelumnya. Selanjutnya akan ditampilkan hasil

pengujian dengan menampilkan berapa *constraint* yang terlanggar untuk setiap percobaan dan nilai *fitness*-nya. Pengujian dilakukan satu kali untuk setiap *user* dengan nilai PC 0,8, PM 0,2 dan maksimal generasi 5.000. Untuk menghitung akurasi sistem digunakan rumus sebagai berikut :

$$akurasi = \frac{\Sigma penalti - \Sigma \left(\frac{gen\ terlanggar}{maksimal\ gen\ terlanggar} \times penalti \right)}{\Sigma penalti} \times 100 \% \quad (5)$$

Tabel 8 menampilkan hasil pengujian akurasi kinerja sistem yang telah dilakukan.

Tabel 8.
Hasil analisis kinerja sistem

User Nim	Jumlah SC	Jumlah Constraint Terlanggar			Nilai Fitness	Nilai akurasi
		K1	K2	SC		
71130048	2	0	0	0	1	100%
71130101	4	0	0	0	1	100%
71130105	2	0	0	0	1	100%
71130010	0	0	0	0	1	100%
71130037	0	0	0	0	1	100%
71140021	1	0	0	0	1	100%
71130084	1	0	0	0	1	100%
71140062	3	0	0	0	1	100%
71140023	4	1	2	2	0,11	80%
71130096	4	0	0	0	1	100%

Keterangan :

- K1 : tidak mengambil lebih dari satu jadwal pada hari dan jam yang sama. Nilai penalti: 2
- K2 : Tidak ada jadwal mata kuliah yang diambil dua kali. Nilai penalti: 2
- SC : *Softconstraint* yang ditetapkan *user* di mana tidak bisa mengambil mata kuliah di jam dan hari yang telah ditentukan. Nilai penalti: 1

Pada hasil pengujian akurasi genetika di atas, 9 dari 10 percobaan mendapatkan akurasi 100% sedangkan satu percobaan mendapatkan akurasi 80%. Dengan demikian hasil rata-rata akurasi Algoritma Genetika dari 10 percobaan adalah 98%.

Pada percobaan kesembilan dengan nim 71140023 mendapatkan nilai *fitness* 0,14 dengan 3 mata kuliah melanggar *hard constraint* dan 2 mata kuliah melanggar *soft constraint*. Dengan demikian, maka selanjutnya akan dilakukan proses CSP untuk mendapatkan hasil jadwal yang tidak melanggar *hard constraint*. Untuk melihat perbandingan jadwal yang dihasilkan pada kasus di atas dapat dilihat pada Tabel 9 dan Tabel 10. Percobaan untuk *user* dengan nim 71140023, dengan *soft constraint* tidak dapat mengambil jadwal mata kuliah pada hari Senin sesi 1, Senin sesi 2, Senin sesi 3 dan Senin sesi 4.

Tabel 9.
Hasil jadwal proses Genetika nim 71140023

Kode	Nama Mata kuliah	Group	Hasi	Waktu
MH1053	ILMU SOSIAL & BUDAYA DASAR	A	Selasa	SESI 1
MH1053	ILMU SOSIAL & BUDAYA DASAR	D	Selasa	SESI 2
PIW213	PRAKTIKUM PEMROGRAMAN WEB	A	Rabu	SESI 1
TIW183	DESAIN DAN EVALUASI ANTAR MUKA	B	Selasa	SESI 3
PIP463	PRAKTIKUM BRIDGING DAN SWITCHING	B	Kamis	SESI 2
TIW183	DESAIN DAN EVALUASI ANTAR MUKA	A	Senin	SESI 3
TIW213	PEMROGRAMAN WEB	D	Selasa	SESI 2
TIW133	KEAMANAN KOMPUTER	B	Senin	SESI 2
TIW203	KECERDASAN BUATAN	C	Kamis	SESI 3
TIW113	SISTEM OPERASI	B	Kamis	SESI 4

Pada Tabel 9 di atas dapat dilihat terdapat beberapa mata kuliah yang melanggar *constraint* yaitu :

- Dua mata kuliah yang sama yaitu mata kuliah “Ilmu Sosial & Budaya Dasar” dan “Desain dan Evaluasi Antar Muka”.
- Jadwal dua mata kuliah yang bertabrakan yaitu pada hari Selasa sesi 2.
- Dua jadwal mata kuliah yang melanggar *soft constraint* yaitu hari Senin sesi 3 dan Senin sesi 2.

Tabel 10 adalah jadwal yang dihasilkan setelah proses CSP

Tabel 10.
Hasil jadwal setelah proses CSP pada nim 71140023

Kode	Nama Mata kuliah	Group	Hasi	Waktu
MH1053	ILMU SOSIAL & BUDAYA DASAR	A	Selasa	SESI 1
TIW213	PEMROGRAMAN WEB	D	Selasa	SESI 2
PIW213	PRAKTIKUM PEMROGRAMAN WEB	A	Rabu	SESI 1
TIW133	KEAMANAN KOMPUTER	B	Senin	SESI 2
PIP463	PRAKTIKUM BRIDGING DAN SWITCHING	B	Kamis	SESI 2
TIP213	PENGOLAHAN CITRA DIGITAL	A	Jumat	SESI 1
TIW183	DESAIN DAN EVALUASI ANTAR MUKA	D	Rabu	SESI 4
TJ3033	BRIDGING DAN SWITCHING	A	Senin	SESI 1
TIW203	KECERDASAN BUATAN	C	Kamis	SESI 3
TIW113	SISTEM OPERASI	B	Kamis	SESI 4

Pada hasil jadwal di atas tidak ada lagi *hard constraint* yang terlanggar. Masih terdapat *soft constraint* yang terlanggar karena dalam proses CSP tidak memperhitungkan *soft constraint*. Dengan percobaan di atas, maka dapat disimpulkan dengan proses CSP sistem mampu mengatasi jadwal yang masih memiliki *constraint* terlanggar setelah proses genetika.

4. Penutup

Kesimpulan didapat dari penelitian yang telah dilakukan tentang sistem rekomendasi mata kuliah antara lain sebagai berikut :

- a. Sistem dapat merekomendasikan mata kuliah dan jadwal bagi mahasiswa dalam membantu proses registrasi.
- b. Melalui analisis menggunakan skala likert, sistem dapat memberikan rekomendasi mata kuliah untuk membantu pengguna (mahasiswa) dalam memilih mata kuliah dengan nilai rata-rata 4,24 dari maksimal 5 (kategori baik). Rekomendasi mata kuliah dibagi ke dalam empat kelompok konsentrasi yaitu pemrograman, jaringan, *game*, dan sistem cerdas.
- c. Pada hasil pengujian akurasi Genetika, 9 dari 10 percobaan mendapatkan akurasi 100% sedangkan satu percobaan mendapatkan akurasi 80%. Dengan demikian maka hasil rata-rata akurasi Algoritma Genetika dari 10 percobaan adalah 98%.
- d. Jumlah kromosom dalam satu populasi dapat mempengaruhi hasil perhitungan pada proses genetika. Semakin banyak jumlah kromosom pada satu populasi maka semakin besar kemungkinan mendapatkan nilai *fitness* terbaik. Dari analisis yang dilakukan jumlah kromosom yang ditentukan pada populasi adalah 30.
- e. Setelah melakukan analisis dengan menggunakan kombinasi nilai probabilitas *crossover* 0,8 dan probabilitas mutasi 0,2 akan mendapatkan rata-rata nilai *fitness* lebih baik.
- f. Jumlah maksimal generasi yang paling baik dari analisis yang telah dilakukan adalah pada *range* 5.000 sampai 6.000 iterasi. Jumlah maksimal generasi mempengaruhi waktu proses jalannya program. Semakin banyak iterasi generasi yang dilakukan maka semakin lama proses genetika berjalan.
- g. Proses *Constraint Satisfaction Problem* (CSP) setelah proses Genetika mampu memberikan rekomendasi jadwal yang lebih baik tanpa ada *constraint* mata kuliah yang terlanggar.

Daftar Pustaka

- Arifudin, R. (2011). OPTIMASI PENJADWALAN PROYEK DENGAN PENYEIMBANGAN BIAYA MENGGUNAKAN KOMBINASI CPM DAN ALGORITMA GENETIKA .
Jurnal Masyarakat Informatika, Volume 2, Nomor 4, 1-14.
- Basuki, A. (2003). Algoritma Genetika, Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning. Surabaya: PENT-ITS.
- Brigida. (2015, September 16). *Algoritma Genetik*. Diakses pada 2 Maret 2016 dari informatika.web.id: <http://informatika.web.id/algoritma-genetik.htm>
- Buliali, J. L., Herumurti, D., & Wiriapradja, G. (2008). PENJADWALAN MATAKULIAH DENGAN MENGGUNAKAN. *Juti Volume 7, Nomor 1*, 29-38.
- Buliali, J. L., Herumurti, D., & Wiriapradja, G. (2008). Penjadwalan Matakuliah dengan menggunakan Algoritma Genetika dan Metode Constraint Satisfaction. *Juti Volume 7, Nomor 1*, 29-38.
- Chairi, L. S. (2010). Implementasi Algoritma Genetika untuk Penjadwalan Kegiatan Belajar Mengajar. In *Skripsi*. Yogyakarta: Fakultas Teknologi Informasi Universitas Kristen Duta Wacana.
- Gunawan, A., & Poh, K. (2007). Solving the Teacher Assignment-Course Scheduling Problem by a Hybrid Algorithm. *World Academy of Science Engineering and Technology 33*, 259-264.
- Kumar, M., Husain, M., Upreti, N., & Gupta, D. (2010). Genetic Algorithm : Review and Application. *International Journal of Information Technology and Knowledge Management 2*, 451-454.
- Mawaddah, N. K., & Mahmudy, W. F. (2006). Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika. *Kursor, vol. 2, no. 2*, 1-8.
- Ratnawati, D., Liliana, D., Regasari, R., & Muflikhah, L. (2012). Modul Bahan Ajar Kecerdasan Buatan. Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- Saputro, Y. N. (2004). PEMAKAIAN ALGORITMA GENETIK UNTUK PENJADWALAN JOB SHOP DINAMIS . *Jurnal Teknik Industri*, 61-70.
- Setemen, K., & Purnomo, M. H. (2008). Kombinasi Algoritma Genetika dan Tabu Search. *Seminar on Intelligent Technology and Its Applications 2008*, 375-378.
- Widyadana, I. G., & Pamungkas, A. (2002). Perbandingan Kinerja Algoritma Genetika dan Simulated Annealing untuk Masalah Multiple Objective pada Penjadwalan Flowshop. *Jurnal Teknik Industri Vol.4 No. 1*, 26-35.

Witary, V., Rachmat, N., & Inayatullah. (2005). Optimasi Penjadwalan Perkuliahan dengan Menggunakan Algoritma Genetika (Studi Kasus : AMIK MDP, STMIK GI MDP dan STIE MDP). *STMIK GI MDP JURNAL*, 1-7.

Biodata Penulis

I Made Arya Beta Widyatmika, memperoleh gelar S1 di Universitas Kristen Duta Wacana.

Gloria Virginia, memperoleh gelar S1 di Universitas Kristen Duta Wacana. Memperoleh gelar S2 di Katholieke Universiteit Leuven, Belgium. Memperoleh gelar S3 di University of Warsaw, Poland. Saat ini menjadi pengajar di Universitas Kristen Duta Wacana.

Rosa Delima, memperoleh gelar S1 di Universitas Bina Darma. Memperoleh gelar S2 di Universitas Gajah Mada. Saat ini menjadi pengajar di Universitas Kristen Duta Wacana.