

## SISTEM PENJADWALAN KULIAH DENGAN MENGGUNAKAN ALGORITMA GENETIKA

Purwanto Simamora<sup>1</sup>, Tulus<sup>2</sup>, Fakhruddin R.Batubara<sup>3</sup>

<sup>1</sup>Mahasiswa Magister Teknik Elektro, Fakultas Teknik Universitas Sumatera Utara

<sup>2,3</sup>Dosen Magister Teknik Elektro, Fakultas Teknik, Universitas Sumatera Utara

Diterima 7 September 2013, disetujui untuk publikasi 22 September 2013

**Abstract** Course scheduling is process of placing courses into available time-and classroom slots. As number of course activities and requirement needed increase, the solution to scheduling problem become more complicated. In order to get a good course scheduling, there are some important components such as lecturer, subject, student, curriculum, classroom and time. The limitation of rooms and time is often become a problem in making the course scheduling. The addition number of the students and the subjects due to the application of the new curriculum, and the other activity in university give the effects the using of times and rooms. Course scheduling systems which is adaptable in every changing that happened. The course scheduling must be able to overcome the limitation of each component such as the number of the lecturer, times and rooms. This research uses two methods in making the course scheduling which are by using the scheduling algorithm and without using the scheduling algorithm. These two methods will be combined into the genetic algorithm in order to get a course scheduling which can reduce the problem in the availability of rooms and times. The research result shows without using the schedule algorithm these are 2 faults appear in the time schedule and the minimum number of room which used is 9 rooms and converged in the 127<sup>th</sup> generation. But, by using the schedule algorithm the number of the fault is no longer appear, the using of the room becomes 8 rooms and converget in the 85<sup>th</sup> generation. It is concluded that making the course scheduling by combining the schedule algorithm and genetic algorithm is a right method in reducing the time fault and minimizing the using of the room.

**Kata kunci:**  
Course  
Scheduling,  
Scheduling  
Algorithm, Genetic  
Algorithm,  
Convergen

### Pendahuluan

Masalah penjadwalan merupakan masalah penugasan sejumlah kegiatan dalam periode waktu tertentu. Penjadwalan kuliah merupakan kegiatan mengalokasikan sejumlah aktivitas kuliah dan batasan mata kuliah ke dalam *slot* ruang dan waktu yang telah tersedia. Untuk menghasilkan suatu penjadwalan kuliah, diperlukan beberapa komponen

penjadwalan yaitu dosen, mata kuliah, mahasiswa, kurikulum, ruang dan waktu (Burk dan Petrovic, 2002).

Algoritma genetika menggunakan teknik evolusi biologi seperti mutasi, kawin silang dan seleksi. Proses diawali dengan menentukan suatu objek populasi (inisialisasi). Selanjutnya dilakukan evaluasi *fitness* dari tiap objek (analisa). Ketika populasi yang diperoleh tidak cukup

*fit*, maka objek tersebut akan dibuang. Bila jumlah populasi sudah memenuhi harapan, maka mulai dilakukan proses seleksi untuk dua populasi terbaik. Proses pemilihan dilakukan dengan mutasi acak. Setelah seluruh proses terpenuhi, maka dilakukan evaluasi dan penempatan dalam populasi (Murugan, 2009).

Penggunaan jumlah ruang penting untuk diteliti, karena hasil optimasi yang digunakan akan memberikan gambaran jumlah ruang minimal yang harus ada apabila terjadi penambahan jumlah mahasiswa, jumlah mata kuliah akibat penerapan kurikulum baru, dan konsentrasi bidang keahlian atau sebagai masukan dalam rencana pengembangan kegiatan di universitas. Penggunaan ruang kuliah di Fakultas Teknik Universitas Sumatera Utara saat ini dinilai belum efektif. Hal ini terlihat dari seringnya menggunakan beberapa ruang tertentu secara terus menerus sementara ada ruang lain yang jarang digunakan sehingga distribusi pemakaian ruangan tidak merata.

Sistem penjadwalan kuliah yang diinginkan adalah sistem penjadwalan yang optimal dan cepat untuk mengatasi tiap perubahan yang terdapat pada data masukan, tetapi sistem yang dihasilkan harus tetap mengikuti aturan penjadwalan yang berlaku.

Berdasarkan latar belakang di atas, penelitian ini akan membandingkan proses penyusunan jadwal kuliah dengan menggunakan algoritma genetika dengan dua metode, yaitu dengan menggunakan algoritma penjadwalan dan tanpa menggunakan algoritma penjadwalan. Kedua algoritma nantinya harus dapat menemukan solusi yang lebih tepat

dalam mengoptimalkan penggunaan jumlah ruangan.

Tujuan dari penelitian ini adalah menghasilkan suatu sistem penjadwalan kuliah yang optimal pada penggunaan jumlah ruangan sehingga waktu yang dibutuhkan untuk pencarian solusi lebih cepat. Hasil penelitian ini dapat digunakan untuk membantu menyelesaikan masalah penjadwalan kuliah dengan seoptimal mungkin.

Masalah penjadwalan kuliah merupakan masalah keterbatasan dalam menggunakan sumber daya yang tersedia. Sumber daya tersebut merupakan komponen utama yang menjadi pertimbangan penting dalam penyusunan jadwal kuliah, dibagi dalam lima kelompok besar, yaitu: kelompok dosen, kelompok mata kuliah, kelompok kurikulum, kelompok waktu dan kelompok ruang.

Dosen, mata kuliah dan kurikulum disusun membentuk suatu aktivitas perkuliahan. Apabila ketiga kelompok ini ditempatkan ke dalam *slot* waktu dan ruang, maka akan timbul permasalahan. Permasalahan tersebut antara lain sulitnya mendapatkan alokasi waktu yang tepat terhadap ketersediaan waktu dosen dengan mata kuliah yang ditawarkan. Solusi dikatakan optimal jika seluruh aktivitas perkuliahan dapat disusun dalam waktu singkat.

Persyaratan penjadwalan antara lain: (1) Seluruh mata kuliah terjadwal dan dinyatakan dalam waktu yang jelas. (2) Kelas merupakan gabungan dari aktivitas perkuliahan yaitu dosen, mata kuliah dan mahasiswa. (3) Dosen hanya mengajar pada 1 kelas untuk waktu yang sama. (4) Mahasiswa hanya boleh mengambil mata kuliah pada satu kelas untuk waktu yang sama. (5) Kegiatan perkuliahan hanya

boleh dilaksanakan sebanyak satu kali dalam satu ruang dalam satu waktu tertentu. (6) Dosen tidak boleh melakukan perpindahan ruang untuk mata kuliah yang sama jika dosen yang bersangkutan berhalangan hadir.

Kesulitan yang sering dihadapi saat penyusunan jadwal kuliah adalah (1) Permohonan oleh beberapa dosen untuk mengajar menurut ketersediaan waktunya. (2) Penempatan dosen, mata kuliah, kurikulum dan waktu pada ruang yang terbatas menjadi masalah dalam penjadwalan kuliah. (3) Persyaratan tertentu yang ditambahkan mempengaruhi lama waktu penyelesaian jadwal. (4) Sering terkendala oleh persyaratan nyata ketika diaplikasikan dilapangan, sehingga tidak seluruhnya dapat direpresentasikan dengan tepat ke dalam sistem.

Beberapa tahapan penting dalam proses algoritma genetika yang harus dilaksanakan antara lain (Burke, dkk, 2009; Syadid, 2008) : Merepresentasikan materi genetik dari tiap solusi yang mungkin, inisialisasi populasi, evaluasi populasi, proses seleksi, membentuk kromosom baru dengan operator genetika, evaluasi kromosom. Jika memenuhi kriteria *termination*, maka proses akan berhenti dan menghasilkan kromosom terbaik, tetapi jika belum memenuhi maka kembali ke tahap evaluasi populasi.

Diperlukan suatu metode untuk mengkodekan solusi yang mungkin dari suatu permasalahan ke dalam bentuk tertentu sehingga dapat diproses (Adamanti, 2002). Salah satu pendekatan yang mungkin adalah dengan mengkodekan solusi ke dalam kode biner, yang ditulis dalam bit 1 dan 0 seperti ditunjukkan pada Tabel 1, dimana angka tiap posisi

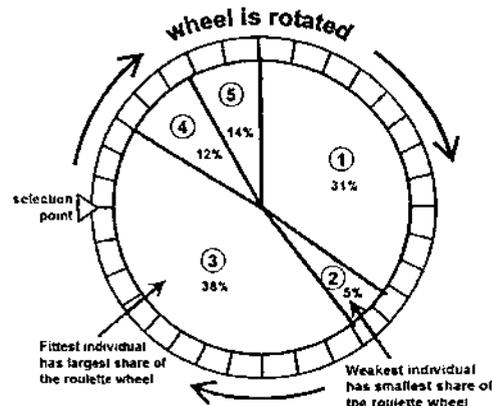
menunjukkan nilai dari beberapa masalah.

Tabel 1. Kromosom dengan Pengkodean Biner

Kromosom 1	101010101100110011100
Kromosom 2	111111000000011111100

Seleksi merupakan proses pemilihan calon induk. Proses seleksi merupakan teknik pemilihan kromosom yang akan dikawinsilangkan dari bakal induk yang baik pada proses evolusi selanjutnya. Tujuannya adalah menghasilkan keturunan yang terbaik dari suatu popuasi untuk diproses melalui rekombinasi dan mutasi. Dengan dasar inilah seleksi dilakukan dengan menggunakan tiga teknik, yaitu seleksi roda *roulette* (*fitness based selection*), seleksi rangking (*rank selectionn*) dan seleksi turnamen (*tournament selection*) (Harik, Lobo, dan Goldberg, 1999).

Pada proses seleksi *Roulette*, induk ditentukan menurut nilai *fitness*. Untuk *fitness* yang tinggi, harapan untuk terpilih semakin besar. Untuk mendapatkan solusi yang terbaik, maka program sebaiknya melakukan seleksi dengan nilai *fitness* yang terendah (Syadid, 2008). Prosesnya ditunjukkan pada Gambar 1.



Gambar 1. Seleksi roda *roulette*

Cara kerja roda tersebut adalah sebagai berikut: pertama, individu-

individu dipetakan pada segmen garis yang berdekatan, sehingga masing-masing individu berukuran sama sesuai dengan *fitness*. Kedua, nomor acak dibangkitkan dan seluruh individu berada pada daerah nomor acak yang telah dipilih. Ketiga, proses kemudian diulangi sampai ditemukan nomor individu yang sesuai keinginan. Kromosom dalam kelompok yang terpilih disusun dalam roda *roulette* berdasarkan pada nilai *fitness*. Kemudian roda diputar ke sejumlah kromosom yang ada dalam populasi, kromosom dengan *fitness* tertinggi memiliki kesempatan terbesar untuk terpilih selama kawin silang dan mutasi (Burke, dkk, 2009; Syadid, 2008).

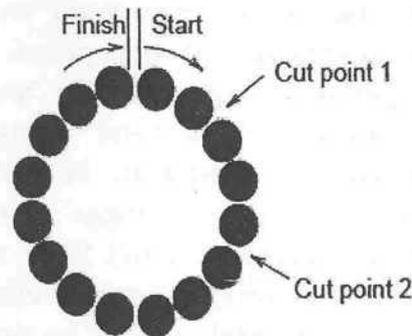
Seleksi turnamen memberikan tekanan seleksi dengan mengatur turnamen selama saingan *s* menurut ukuran turnamennya. Tekanan seleksi dengan meningkatkan individu yang lebih baik lagi disebut tekanan seleksi lanjutan. Tekanan seleksi ini mengatur algoritma genetika untuk memperbaiki hasil *fitness* dari suatu populasi. Pemenang dari turnamen adalah individu dengan nilai *fitness* tertinggi kemudian dimasukkan ke dalam kumpulan kawin silang. Kumpulan pasangan yang terdiri dari pemenang turnamen, memiliki nilai rata-rata *fitness* yang lebih tinggi dibandingkan dengan rata-rata *fitness* populasinya. Perbedaan dari nilai *fitness* ini memberikan tekanan seleksi, yang mengatur algoritma genetika untuk memperbaiki nilai *fitness* dari masing-masing generasi yang berhasil (Adamanti, 2002).

Kawin silang (*crossover*) sering digunakan untuk menghasilkan keturunan baru berdasarkan induk yang terpilih. Aturan kawin silang dalam algoritma genetika adalah

mengkombinasikan bit dari solusi yang *fit*. Proses pengulangan banyak digunakan oleh operator *crossover* untuk mendapatkan gen populasi (Harik, dkk, 1999).

Salah satu teknik yang diperkenalkan adalah *crossover* satu titik, dimana prinsip kerjanya adalah dengan mengambil dua induk, kemudian memotong kromosom *string*-nya beberapa posisi secara acak untuk menghasilkan dua bagian utama dan pendukung. Bagian pendukung kemudian ditukar untuk menghasilkan dua kromosom baru. Kedua induk yang dihasilkannya masing-masing mewarisi beberapa gen dari induknya. Teknik ini disebut kawin silang satu titik (Babu, dkk, 2010; Beasley, dkk, 1993).

Dalam kawin silang dua titik (*two point crossover*), kromosom ditampilkan sebagai *loop* yang dibentuk berdasarkan penggabungan bersama. Untuk menukar bagian satu *loop* dengan *loop* lain dibutuhkan seleksi dua titik potong, seperti yang ditunjukkan oleh Gambar 2. Secara umum, kawin silang dua titik menunjukkan fungsi yang sama seperti kawin silang satu titik. Penelitian yang menggunakan kawin silang dengan dua titik lebih disukai daripada kawin silang satu titik (Beasley, dkk, 1993).



Gambar 2. Kawin silang 2 titik

Elitisme merupakan satu teknik untuk mempertahankan kromosom yang memiliki nilai *fitness* terbaik

untuk tetap bertahan hidup pada generasi selanjutnya. Elitisme dilakukan dengan menyalin satu dan menyalin dua kromosom untuk jumlah populasi genap (Burke, dkk, 2009; Beasley, dkk, 1993)).

Pertama-tama kromosom anak yang akan diirialisasi bernilai kosong. Kemudian suatu vektor sepanjang  $n$  diisi secara acak pada sekumpulan elemen {1,2}. Vektor menggambarkan urutan yang dihasilkan oleh induk 1 dan 2. Setelah gen diwaliki oleh 1 induk dan yang lainnya dihilangkan, maka hasilnya diberikan kepada kromosom anak. Langkah tersebut diulangi sampai kedua kromosom induk hilang dan kromosom anak berisi seluruh gen anak yang dimasukkan (Bierwirth, dkk, 1998; Yamada dan Nakano, 1997):

induk 1	3 2 2 2 3 1 1 1 3
induk 2	1 1 2 2 2 1 2 3 3
gen dari induk	1 1 2 2 2 2 1 1 1
PPX anak	3 2 1 1 2 1 2 3 3

Gambar 3. Proses *Precedence Preservative Crossover (PPX)*

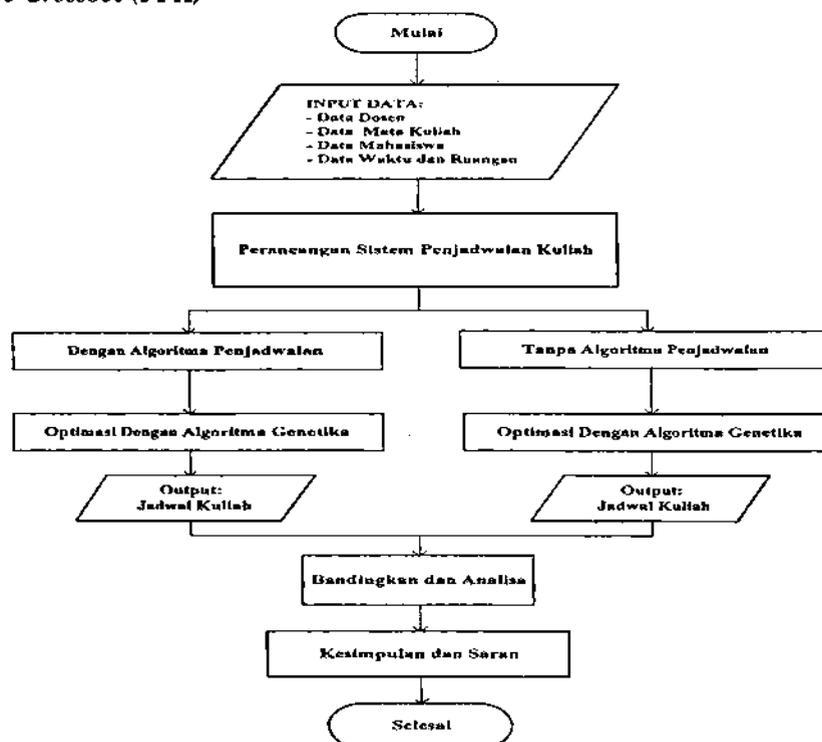
Konvergen merupakan pergerakan menuju keseragaman pertambahan secara berturut-turut. Gen pembawa sifat dikatakan konvergen jika kesamaan populasi mencapai 95%. Populasi disebut konvergen jika seluruh gen telah konvergen.

**Metode Penelitian**

Data berasal dari enam departemen yang ada di Fakultas Teknik Universitas Sumatera Utara untuk semester ganjil T.A. 2011/2012. Keenam departemen tersebut adalah Departemen Teknik Elektro, Departemen Teknik Mesin, Departemen Teknik Sipil, Departemen Teknik Industri, Departemen Teknik Kimia dan Departemen Arsitektur.

Data tersebut meliputi daftar nama dosen, daftar nama mata kuliah, daftar mahasiswa, dan daftar ruang kuliah yang ada. Data diambil sesuai dengan jumlah departemen yang ada.

Secara ringkas tahapan penelitian ini ditunjukkan dalam bagan alir pada Gambar 4.



Gambar 4. Diagram Alir Penelitian

Penjadwalan kuliah yang dihasilkan selanjutnya dioptimasi untuk mengurangi atau menghilangkan pelanggaran yang terjadi dan mengurangi penggunaan jumlah ruang. Empat tahapan penting dalam optimasi, yaitu: pemilihan representasi masalah, pemilihan operator genetika, penetapan fungsi *fitness*, dan penetapan parameter genetika. Penggunaan jenis atau metode yang berbeda tiap tahapan menghasilkan keluaran yang berbeda. Pengujian menggunakan parameter yang berbeda-beda untuk setiap tahapan untuk menemukan hasil yang paling optimal.

**Hasil Dan Pembahasan**

Algoritma Penjadwalan merupakan proses menerjemahkan suatu kromosom ke dalam bentuk jadwal. Dalam proses tersebut digunakan suatu algoritma yang telah mempertimbangkan hal-hal yang menjadi batasan dalam proses penjadwalan. Dengan demikian, diharapkan algoritma genetika akan memilih jadwal yang lebih baik untuk dijadikan solusi yang paling optimum.

Pada algoritma penjadwalan, yang menjadi masukan adalah data mata kuliah dan dosen, data mahasiswa dan data waktu kuliah. Sedangkan yang menjadi batasan adalah penggunaan jumlah ruangan dan pelanggaran waktu mengajar dosen.

Proses pencarian diawali dengan metode langsung (tanpa melalui algoritma penjadwalan). Pada metode ini kromosom akan langsung berisi kode waktu dan ruang yang digunakan. Fungsi objektif dibagi 2, yaitu meminimalkan pelanggaran waktu serta penggunaan jumlah ruangan. Untuk kondisi optimasi

dengan dua fungsi *fitness* seperti ini dapat didekati melalui metode *pareto optimization*. Metode ini tidak langsung menemukan suatu solusi yang menjadi solusi tunggal dari permasalahan namun merupakan kumpulan solusi terbaik yang mempertimbangkan seluruh fungsi *fitness* yang ada. Sekumpulan solusi dianggap telah mencapai *optimum* apabila perubahan pada satu variabel dalam solusi tidak menyebabkan variabel lainnya berubah dan ketika kondisi ini tercapai proses pencarian dihentikan.

Dalam proses optimasi diperlukan suatu batasan yang menjadi *soft* dan *hard constraint*. *Hard constraint* merupakan batasan mutlak yang harus dipenuhi atau dibuat nol sedangkan *soft constraint* merupakan batasan yang harus dipenuhi semaksimal mungkin namun apabila tidak dapat dipenuhi seluruhnya tidak menjadi masalah.

Tabel 2. Pengurutan Kromosom Berdasarkan *fitness f1*

Sebelum Sort			Setelah Sort		
Kromosom	<i>fitness f1</i>	<i>fitness f2</i>	Kromosom	<i>fitness f1</i>	<i>fitness f2</i>
1	26	12	1	23	13
2	24	13	2	24	13
3	24	13	3	25	13
4	26	12	4	26	12
5	34	13	5	26	12
6	25	13	6	26	13
7	35	10	7	27	13
8	26	13	8	27	9
9	29	13	9	27	12
10	33	11	10	27	13
11	27	13	11	29	13
12	29	13	12	29	13
13	37	12	13	29	13
14	36	8	14	29	13
15	29	13	15	30	13

16	30	13	16	33	11
17	27	9	17	34	13
18	29	13	18	34	11
19	35	13	19	35	10
20	27	12	20	35	13
21	36	13	21	36	8
22	38	11	22	36	13
23	27	13	23	36	13
24	34	11	24	37	12
25	36	13	25	38	11

Dari hasil pengujian, pencarian nilai *fitness* menghasilkan konvergen pada generasi ke-127 dengan beberapa populasi akhir seperti pada Tabel 3.

Tabel 3. Populasi Akhir dari Hasil Proses Optimasi

Kromosom	fitness $f_1$	fitness $f_2$	Rangkai ng	Fitness total
1	2	12	1	1,9928
2	2	9	1	1,9960
3	2	9	1	1,9960
4	2	9	1	1,9960
5	2	13	1	1,9916
6	2	13	1	1,9916
7	2	13	1	1,9916
8	2	9	1	1,9960
9	2	11	1	1,9940
10	2	10	1	1,9950
11	2	12	1	1,9928
12	2	13	1	1,9916
13	2	9	1	1,9960
14	2	9	1	1,9960
15	2	9	1	1,9960
16	2	13	1	1,9916
17	2	13	1	1,9916
18	2	13	1	1,9916
19	2	9	1	1,9960
20	2	13	1	1,9916

Berdasarkan hasil akhir dari proses pencarian,  $f_1$  merupakan jumlah pelanggaran waktu dan  $f_2$  merupakan penggunaan jumlah ruangan, diperoleh hasil dengan *Pareto*

*Optimum* jumlah populasi = 20, proses elitisme dengan menghilangkan 50% dari total kromosom pada populasi dan probabilitas mutasi sebesar  $P_m = 0,1$  pada pencarian hingga 127 generasi menghasilkan jadwal dengan jumlah pelanggaran waktu sebanyak 2 kali dan jumlah ruangan maksimal yang digunakan sebanyak 9 ruangan. Untuk *fitness* total terlihat bahwa nilainya berbeda-beda. Nilai *fitness* total akan semakin kecil jika nilai *fitness*  $f_2$  semakin kecil.

Dengan langkah yang sama, kromosom pada setiap generasi diperbaiki sampai diperoleh kromosom yang terbaik. Dari hasil pengujian, pencarian nilai *fitness* konvergen pada generasi ke-85 dengan beberapa populasi akhir seperti pada Tabel 4.

Tabel 4. Populasi akhir sebagai Solusi Optimal dari Proses Optimasi dengan Algoritma Penjadwalan

Kromosom	Fitness $f_1$	Fitness $f_2$	Ran k	Fitness Total
1	0	9	1	1,9960
2	0	8	1	1,9968
3	0	9	1	1,9960
4	0	9	1	1,9960
5	0	8	1	1,9968
6	0	9	1	1,9960
7	0	8	1	1,9968
8	0	9	1	1,9960
9	0	9	1	1,9960
10	0	9	1	1,9960
11	0	8	1	1,9968
12	0	8	1	1,9968
13	0	9	1	1,9960
14	0	9	1	1,9960
15	0	9	1	1,9960
16	0	9	1	1,9960
17	0	9	1	1,9960
18	0	8	1	1,9968
19	0	9	1	1,9960
20	0	9	1	1,9960

Pada populasi akhir di atas terlihat bahwa konvergensi dengan nilai *fitness*  $f_1$  menuju nilai nol tetapi nilai *fitness*  $f_2$  masih beragam dengan nilai paling kecil sebesar 9. Perbedaan nilai *fitness*  $f_2$  menyebabkan nilai *fitness* total tidak sama antara kromosom satu dengan kromosom yang lain. Hal ini menunjukkan bahwa nilai *fitness* total dalam percobaan ini tidak dapat konvergen karena nilai *fitness*  $f_2$  yang tidak konvergen menuju satu nilai tertentu.

Berdasarkan hasil akhir pencarian tersebut, pelanggaran waktu sebagai *hard constraint* telah berhasil dihilangkan dengan nilai  $f_1$  sebesar nol. Jumlah penggunaan ruangan paling sedikit adalah 8 ruangan. Hasil ini diperoleh berdasarkan optimasi dengan menggunakan algoritma penjadwalan dan diperbaiki dengan algoritma genetika melalui metode *pareto optimum* dengan parameter yang digunakan antara lain: jumlah populasi sebanyak 20 unit, proses elitisme dengan membuang 50% kromosom dari populasi dan probabilitas mutasi sebesar  $P_m = 0,1$ . Konvergensi tercapai pada generasi ke- 85.

Proses penjadwalan dengan metode penggabungan Algoritma Penjadwalan dan Algoritma Genetika dilakukan dalam dua tahap. Tahap pertama adalah kromosom yang merupakan hasil pengkodean dari urutan kegiatan diolah dalam algoritma penjadwalan sehingga menghasilkan suatu jadwal yang utuh yang didalamnya terdapat nilai-nilai untuk menghitung besarnya nilai *fitness*. Tahap kedua adalah tahapan penggunaan algoritma genetika dalam mengevaluasi setiap kromosom berdasarkan nilai *fitness* tiap kromosom. Algoritma genetika akan bekerja untuk memperbaiki

kromosom-kromosom melalui operasi genetika.

Tabel 5. Hasil Akhir Proses Optimasi dengan Algoritma Penjadwalan dan Tanpa Algoritma Penjadwalan

Kromosom	Metode Tanpa Algoritma Penjadwalan (Konvergen pada Generasi 127)		Metode dengan Algoritma Penjadwalan (Konvergen pada Generasi 85)	
	Jlh Pelanggaran Waktu ( $f_1$ )	Jlh Penggunaan Ruang ( $f_2$ )	Jlh Pelanggaran Waktu ( $f_1$ )	Jlh Penggunaan Ruang ( $f_2$ )
1	2	12	0	9
2	2	9	0	8
3	2	9	0	9
4	2	9	0	9
5	2	13	0	8
6	2	13	0	9
7	2	13	0	8
8	2	9	0	9
9	2	11	0	9
10	2	10	0	9
11	2	12	0	8
12	2	13	0	8
13	2	9	0	9
14	2	9	0	9
15	2	9	0	9
16	2	13	0	9
17	2	13	0	9
18	2	13	0	8
19	2	9	0	9
20	2	13	0	9

Setelah pengujian pada dua metode yang berbeda dilakukan, selanjutnya dilakukan analisa dari hasil pengujian

tersebut. Analisa dilakukan untuk melihat bagaimana kedua metode tersebut dapat menghasilkan jadwal yang terbaik. Oleh karena proses algoritma genetika yang digunakan melalui metode *pareto optimum*, maka hasil akhir yang dibandingkan ditunjukkan pada Tabel 5.

Dari hasil akhir terlihat bahwa metode tanpa penggunaan algoritma penjadwalan menghasilkan suatu kumpulan solusi akhir berupa jadwal kuliah yang memiliki pelanggaran waktu sedikitnya dua kali pelanggaran. Berbeda dengan metode dengan menggunakan algoritma penjadwalan terlebih dahulu, pada hasil akhir terlihat bahwa jadwal yang dihasilkan tidak memiliki pelanggaran waktu. Hal ini terlihat dari jumlah pelanggaran yang terjadi sebesar nol. Berdasarkan pertimbangan bahwa jumlah pelanggaran waktu adalah suatu *hard constraint* yang tidak boleh ada, maka jadwal yang dihasilkan pada metode tanpa algoritma penjadwalan merupakan suatu jadwal yang tidak mungkin untuk diterapkan.

Dengan algoritma penjadwalan lebih mampu mengurangi jumlah penggunaan ruangan. Pada metode dengan algoritma penjadwalan jumlah ruangan minimal yang berhasil diperoleh sebanyak 9 ruangan, sedangkan pada metode dengan algoritma genetika dapat dikurangi hingga 8 ruangan. Pada proses pencarian, lama pencarian untuk kedua metode tergantung dari lamanya pencapaian konvergen. Pencarian dengan metode tanpa algoritma penjadwalan konvergen pada generasi ke-127 sedangkan dengan algoritma penjadwalan konvergensi tercapai pada generasi ke-85. Hal ini disebabkan karena algoritma genetika mempercepat

terbentuknya konvergensi melalui pencarian-pencarian lokal pada jadwal sehingga mencegah terjadinya pelanggaran waktu.

### Kesimpulan Dan Saran

Dari perbandingan kedua variabel yaitu *hard constraint* dan *soft constraint* terlihat bahwa untuk masalah penjadwalan kuliah dengan parameter yang terpilih, algoritma penjadwalan atau *timetable builder* yang dirancang dalam penelitian dengan mempertimbangkan kondisi nyata dan aturan penjadwalan menghasilkan penjadwalan kuliah yang lebih baik dan lebih cepat daripada penggunaan algoritma genetika secara langsung. Hal ini menunjukkan bahwa algoritma penjadwalan melalui pencarian lokal

(*local searching*) dan pencarian *heuristik* mempercepat proses pencarian dengan mengurangi kemungkinan terjadinya pelanggaran waktu dan dalam mengalokasikan kegiatan perkuliahan.

### Daftar Pustaka

- Adamanti, J. (2002): *Penyelesaian Masalah Penjadwalan Mata Kuliah di Fakultas MIPA Universitas Gadjah Mada dengan Menggunakan Algoritma Genetika*. Departemen Pendidikan Nasional, Universitas Gadjah Mada: Skripsi.
- Babu, A.S., Chockalingam, R., dan Kavitha, S. (2010): *A Hybrid Genetic Algorithm Approach to a Departmental Class Timetabling Problem Using Efficient Data Structures*. *International Journal of Computer Applications* (0975-8887) Vol. 1 – No. 17.

- Beasley, D., Bull, D.R., dan Martin R.R. (1993): *An Overview of Genetic Algorithms: Part 1, Fundamental*. University Computing, Inter-University Committee on Computing.
- Beasley, D., Bull, D.R., dan Martin R.R. (1993): *An Overview of Genetic Algorithms: Part 2, Research Topics*. University Computing 15(4) 170 – 181. UCISA.
- Bierwirth, C., Mattfeld, D.C., dan Kopler, H. *On Permutation Representation for Scheduling Problems*. University of Bremen, Dept. of Economics, D-28334 Bremen, Germany.
- Burke, E.K., Petrovic, S. (2002): *Recent Research Direction in Automated Timetabling*. European Journal of Operational Research-EJOR, 140/2, 266-280
- Burke, E., Elliman, D., dan Weare, R. *A Genetic Algorithm Based University Timetabling System*. Department of Computer Science, University of Nottingham, University Park, NG7 2RD.
- Harik, G.R., Lobo, F.G., dan Goldberg, D.E. (1999): *The Compact Genetic Algorithm*. IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4.
- Murugan, A.S.R., (2009): *University Timetabling using Genetic Algorithm*. Master Thesis, Computing Engineering, Dalarna University Roda Vagen 3 S-781 88 Borlange Sweden.
- Syadid, M. (2008): *Penjadwalan Perkuliahan Menggunakan Algoritma Genetika*. Departemen Ilmu Komputer, Fakultas MIPA. Institut Pertanian Bogor: Skripsi.
- Yamada, T., Nakano, R. (1997): *Genetic Algorithms for Job-Shop Scheduling Problems*. Proceedings of Modern Heuristic for Decision Support, pp. 67-81, UNICOM Seminar, London