

## PEMANFAATAN COMPACT GENETIC ALGORITHM (cGA) UNTUK OPTIMASI PENJADWALAN PENGGUNAAN RUANG KULIAH DI U.K. PETRA

Gregorius Satia Budhi<sup>1</sup>, Andreas Handoyo<sup>2</sup>, Billy Soloment<sup>3</sup>

<sup>1, 2, 3</sup> Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Telp. (031)8494830, (031)8439040

E-mail: greg@petra.ac.id, handoyo@petra.ac.id

### ABSTRAK

Penjadwalan penggunaan ruang kuliah di Universitas Kristen Petra selama ini dilakukan secara manual dan membutuhkan waktu yang lama. Hal ini karena banyak sekali mata kuliah yang ditawarkan oleh semua jurusan yang ada, yaitu lebih dari 1000 mata kuliah per semester. Untuk mempercepat dan mengoptimalkan hasil, peneliti mencoba salah satu varian dari Algoritma Genetika yaitu compact Genetic Algorithm (cGA) untuk penyusunan jadwal penggunaan ruang kuliah tiap semester secara otomatis. Alasan dari pemilihan metode ini karena pada penelitian sebelumnya peneliti menemukan bukti bahwa proses cGA cukup cepat dan hasil optimasinya cukup baik. Perhitungan nilai fitness tiap kromosom disesuaikan dengan kriteria dan prioritas yang digunakan tanpa penyederhanaan dan asumsi - asumsi khusus. Hasil dari penelitian ini cukup menjanjikan karena rata - rata hasil pengujian lebih baik atau mendekati hasil proses manual. Selain itu hasil survei juga menyatakan bahwa hasil penelitian cukup sesuai dengan harapan dari calon pemakai.

**Kata Kunci:** Optimasi, compact Genetic Algorithm (cGA), Penjadwalan Ruang Kuliah.

### 1. PENDAHULUAN

Pada Universitas Kristen Petra dan mungkin juga pada universitas/sekolah lain proses penjadwalan penggunaan ruang kuliah adalah hal yang penting. Hal ini berhubungan langsung dengan lancar tidaknya kegiatan belajar mengajar.

Selama ini proses penjadwalan penggunaan ruang kuliah dilakukan secara manual oleh salah seorang karyawan pada Biro Administrasi Akademik (BAA). Penjadwalan ini berdasarkan atas kriteria dan prioritas yang telah dibuat, misalnya: Ruang kuliah sebuah jurusan sebaiknya pada gedung yang sama atau berdekatan ruang dosen dan ruang - ruang administratif jurusan tersebut, mata kuliah yang diajarkan oleh dosen luar biasa lebih diprioritaskan dibanding yang diajarkan oleh dosen tetap, dan lain sebagainya.

Setiap awal semester semua jurusan mengirimkan jadwal kuliah lengkap dengan kapasitas kelas dan dosen pengajarnya pada BAA. Dengan menganut kriteria serta aturan yang ada BAA melakukan penjadwalan penggunaan ruang kuliah untuk semua mata kuliah yang ditawarkan oleh semua jurusan. Proses penjadwalan ini tidak mudah, dan karena dilakukan secara manual proses ini rentan terhadap *human error* serta membutuhkan waktu cukup lama. Dengan menggunakan salah satu varian dari GA yaitu *compact Genetic Algorithm* (cGA) peneliti mencoba untuk membuat sebuah sistem penjadwalan penggunaan ruang kuliah yang otomatis (sangat mengurangi *human error*), optimal dan memiliki waktu proses yang relatif lebih cepat dibanding manual. Pemilihan varian GA ini didasarkan pada penelitian yang telah dilakukan oleh peneliti sebelumnya (Budhi, 2008). Pada penelitian itu didapat bukti bahwa *compact Genetic*

*Algorithm* memiliki waktu proses yang cepat dan hasil proses yang optimal (secara rata - rata lebih baik dibandingkan proses manual).

### 2. TINJAUAN PUSTAKA

#### 2.1 Algoritma Genetika

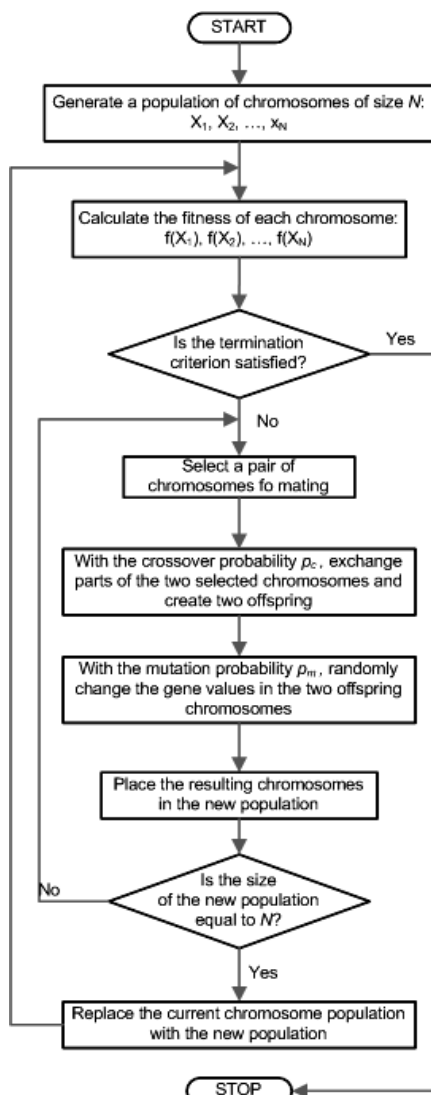
Awal dekade 1970an John Holland memperkenalkan konsep dari Algoritma Genetika. Tujuan dari konsep ini adalah menerapkan apa yang telah ada dilakukan oleh alam ke dalam komputer. Sebagai seorang ilmuwan, Holland memandang algoritma tersebut sebagai bentuk abstrak dari evolusi alam. Algoritma ini berisi step - step prosedur sekuensial yang memproses sebuah populasi kromosom buatan (*artificial*) menjadi populasi baru lainnya. Algoritma ini menggunakan proses seleksi 'natural' dan teknik - teknik yang terinspirasi dari teori genetika, yaitu: *crossover* dan mutasi (Negnevitsky, 2005). Secara umum, proses dari Algoritma Genetika ini dapat dilihat pada Gambar 1.

Kekuatan Algoritma Genetika dalam menemukan solusi optimal telah didemonstrasikan dalam berbagai bidang aplikasi seperti finansial, pengolahan citra, pengontrol pipa gas dan penjadwalan produksi (Langdon, 2000). Belakangan ini pemanfaatan Algoritma Genetika semakin meluas di banyak program aplikasi lain seperti game programming dan text mining.

#### 2.2 Compact Genetic Algorithm (cGA)

*Compact Genetic Algorithm* (cGA) dikembangkan oleh George R. Harik dan kawan - kawan pada tahun 1997. cGA merupakan sebuah *special class* dari Algoritma Genetika. cGA menggambarkan populasi seperti sebuah

probabilitas distribusi dari kumpulan solusi, oleh karena itu, tidak semua populasi perlu untuk disimpan. Dalam setiap generasi, cGA menghasilkan individu-individu berdasarkan pada probabilitas yang dispesifikasikan dalam *probability vector*. Individu-individu tersebut akan dievaluasi dan *probability vector* akan berubah berdasarkan individu yang terbaik. cGA mempunyai keuntungan dalam menggunakan memori yang sedikit dan mencapai kualitas yang sebanding dan mempunyai hasil dengan jumlah yang hampir sama dengan yang dihasilkan oleh *Basic Genetic Algorithm* (Rimcharoen, 2006).



Gambar 1. Basic Genetic Algorithm (Negnevitsky, 2005).

Pada Gambar 2 dan Gambar 3 *pseudocode* dari *compact Genetic Algorithm* dan metode *update*-nya.

1. Initialize probability vector ( $l$  is chromosome length)
 

```
for i :=1 to l do p[i]:=0.5;
```
2. Generate two individuals from the vector
 

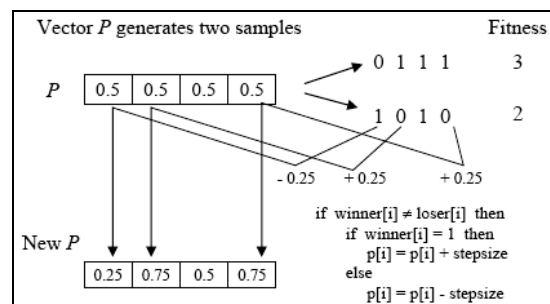
```
a := generate(p);
b := generate(p);
```
3. Let them compete (selection)
 

```
winner,loser := evaluate(a,b);
```
4. Update the probability vector towards the better one ( $n$  is population size)
 

```
for i :=1 to l do
if winner[i] ≠ loser[i] then
if winner[i] = 1 then
p[i] := p[i] + 1/n
else
p[i] := p[i] - 1/n;
```
5. Check if the vector has converged
 

```
for i :=1 to l do
if p[i]>0 and p[i]<1 then
return to step 2;
```
6.  $p$  represents the final solution

Gambar 2. *Pseudocode* dari *compact Genetic Algorithm* (Harik, 1998).



Gambar 3. Metode *update* untuk cGA (Rimcharoen, 2006)

### 3. ANALISA PERMASALAHAN

Setelah menerima seluruh jadwal kuliah dari masing-masing jurusan lengkap dengan kapasitas kelas dan dosen pengajarnya, BAA UK Petra akan melakukan penjadwalan penggunaan ruang kuliah dengan mempertimbangkan kriteria - kriteria sebagai berikut:

- a. Kapasitas Ruang: Pemakaian ruang kuliah disesuaikan antara kapasitas ruang dengan kapasitas kelas yang ditawarkan pada suatu mata kuliah. Kapasitas ruang tidak boleh kurang dari kapasitas kelas.
- b. Lokasi Gedung: Pemakaian ruang kuliah sesuai dengan gedung dimana ruang dosen dan ruang - ruang administratif jurusan berada (dapat dilihat pada Tabel 1). Bila terpaksa baru diperbolehkan menempati ruang - ruang di gedung lain.
- c. Mata kuliah Tepat Semester: Bila mata kuliah yang ditawarkan tepat semester, maka akan lebih diprioritaskan. Contoh mata kuliah tidak tepat semester misalnya: Mata kuliah A pada silabus berada pada semester II, namun dibuka pada semester gasal.
- d. Status Dosen: Bila status dosen pengajar adalah Dosen Tidak Tetap (Dosen Luar Biasa), maka kelas yang diajarnya akan mendapat prioritas untuk dijadwal sesuai hari dan waktu yang telah

ditetapkan jurusan sebelumnya. Bila terpaksa / tidak ada ruang tersisa pada hari dan jam yg ditetapkan, mata kuliah yang diajar oleh dosen tetap dapat diminta untuk di jadwal ulang hari atau waktunya.

Tabel 1. Lokasi Fakultas/Jurusan

Fakultas/Jurusan	Lokasi
Fakultas Teknik dan Fakultas Seni	Gedung P dan I
Jurusan Perhotelan	Gedung A
Jurusan Sastra Inggris	Gedung B
DMU dan Ilmu Komunikasi	Gedung C
Jurusan Sastra Tionghoa	Gedung T lantai 4
Fakultas Ekonomi	Gedung T lantai 2 dan 3

Pada penjadwalan ruang kuliah ini, keempat kriteria yang telah dibahas sebelumnya memiliki prioritas sebagai berikut:

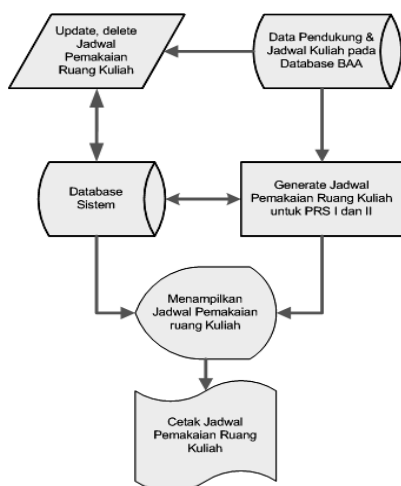
- Prioritas 1: Kapasitas Ruang.
- Prioritas 2: Lokasi Gedung.
- Prioritas 3: Mata Kuliah Tepat Semester
- Prioritas 4: Status Dosen

Proses penjadwalan penggunaan ruang kuliah ini memakan waktu sangat lama. Setelah semua data jadwal kelas kuliah masuk semua ke BAA, biasanya baru dua minggu kemudian jadwal penggunaan ruang kuliah diterbitkan oleh BAA.

#### 4. DESAIN APLIKASI

##### 4.1 Desain aplikasi penjadwalan penggunaan ruang kuliah

Desain aplikasi penjadwalan ruang kuliah UK Petra dapat dilihat pada Gambar 4.



Gambar 4. Desain aplikasi penjadwalan ruang kuliah UK Petra.

Desain dari aplikasi ini cukup sederhana, karena Jadwal kuliah dari masing - masing jurusan dan semua data pendukung seperti data dosen, mata kuliah, ruang dan gedung telah ada pada Database dari software adminstrasi yang dimiliki oleh BAA

UK Petra. Adapun penjelasan lebih detail dari aplikasi ini adalah sebagai berikut:

Sebelum Pendaftaran Rencana Studi I:

- a. Sebelum memulai penjadwalan ruang kuliah, user (BAA) harus memasukkan terlebih dahulu semua jadwal kuliah lengkap dengan kode dosen pengajarnya kedalam database BAA melalui software administrasi yang telah mereka miliki sebelumnya. Selain itu, perlu dipastikan pula bahwa data - data pendukung lainnya adalah data terkini (*up to date*).
- b. Data - data ruang kuliah khusus yang telah diset untuk mata kuliah tertentu diinputkan (contoh: Kelas Studio yang wajib di kelas studio tertentu, kelas programming yang dilaksanakan di lab - lab komputer yang ada). Data - data semacam ini tidak diikuti dalam proses cGA.
- c. *Generate* jadwal penggunaan ruang kuliah untuk PRS I menggunakan algoritma cGA. Selanjutnya jadwal penggunaan ruang kuliah yang terbentuk dapat disimpan pada database sistem, dilihat hasilnya dan dicetak.

Sebelum Pendaftaran Rencana Studi II:

- a. Setelah PRS I, kelas - kelas mata kuliah yang tidak diminati mahasiswa akan ditutup. Selain itu untuk kelas - kelas yang jumlah peminatnya jauh melebihi setting kapasitas kelas yang ada, akan dibuka kelas paralel untuk mata kuliah yang sama. Kebijakan kelas mana yang ditutup dan kelas paralel mana yang dibuka, kapan waktunya dan siapa dosen pengajarnya dilakukan oleh jurusan masing - masing, dan kemudian diinformasikan pada BAA.
- b. Selanjutnya BAA akan menghapus kelas - kelas yang ditutup dari tabel jadwal penggunaan ruang dan mengembalikan status ruang pada hari dan jam yang bersangkutan. Selain itu data - data kelas paralel yang baru perlu juga diupdate pada database BAA.
- c. *Generate* jadwal penggunaan ruang kuliah untuk PRS II menggunakan algoritma cGA. Disini penjadwalan hanya dilakukan pada kelas - kelas yang baru dibuka. Pada proses *generate* ini ruang - ruang yang dapat dipilih hanyalah ruang kuliah yang 'kosong' pada hari dan jam dari kelas yang baru dibuka itu.
- d. Selanjutnya semua jadwal penggunaan ruang kuliah yang terbentuk dapat disimpan pada database sistem, dilihat hasilnya dan dicetak. Jadwal ini adalah gabungan hasil proses sebelum PRS I dan sebelum PRS II.

Setelah Pendaftaran Rencana Studi III:

Pada PRS III, mahasiswa diijinkan untuk mengundurkan diri dari kelas - kelas yang telah diambil sebelumnya, baik pada PRS I maupun II. Akibat pengunduran diri ini ada kemungkinan

sebuah kelas yang pada PRS II belum ditutup menjadi kosong / jumlah mahasiswa yang terdaftar di kelas tersebut dibawah kapasitas minimal. Bila itu terjadi kelas bersangkutan akan ditutup oleh jurusan. Bila sebuah kelas ditutup, BAA harus menghapus kelas tersebut dari jadwal penggunaan ruang kuliah.

#### 4.2 Desain Kromosom

Seperti telah dibahas pada teori penunjang, untuk setiap iterasi, *compact GA* hanya menggunakan 2 kromosom / individu vektor, yaitu kromosom a dan kromosom b. Pada aplikasi ini kedua kromosom disimpan pada dua array dinamis yang panjangnya sesuai dengan jumlah mata kuliah yang ditawarkan, biasanya lebih dari 1000 mata kuliah per semester. Desain kromosom dapat dilihat pada Tabel 2.

Tabel 2. Contoh bentuk kromosom

<b>Novak</b>	DI4213A	DV4212 B	....	TI4215C
<b>Ruang</b>				
<b>Hari</b>	1	1	....	6
<b>Jam</b>	7:30	7:30	....	17:30
<b>Lama</b>	60	120	....	180
<b>FCNIP</b>	02-123	00-101	....	01-121

Keterangan Tabel 2:

Novak : Kode Mata Kuliah

Ruang : Ruang Kuliah

Hari : Hari mata kuliah berjalan

Waktu : Waktu mata kuliah mulai berjalan

Lama : Durasi mata kuliah

FC-NIP : Status Dosen

Posisi *gene (allele)* pada kedua kromosom ini didasarkan atas urutan hari dan jam kuliah dari mata kuliah yang ditawarkan secara ascending. Hal ini dibuat untuk memudahkan proses selanjutnya.

Dua kromosom awal diisi secara random. Untuk mempercepat proses *generate*, sebuah ruang yang telah diset pada sebuah mata kuliah tidak dapat diset lagi di mata kuliah lain pada hari dan jam yang sama. Pada setiap iterasi dua kromosom baru akan dibuat. Selanjutnya nilai dari *probability vector P* akan menentukan apakah isi sebuah *gene* pada kromosom pemenang pada iterasi sebelumnya akan di-copy-kan pada dua kromosom baru, atautan isi *gene* kromosom baru tersebut dirandom ulang. *Probability vector P* dibuat dengan array dinamis satu dimensi dengan panjang sama dengan panjang kromosom. Hasil akhir didapat dari kromosom pemenang terakhir sebelum proses berhenti. Iterasi akan berhenti bila populasi maksimum *n* dicapai atau semua nilai *probability vector P* telah diatas 1 atau dibawah 0.

#### 4.3 Perhitungan Fitness Cost

Rumus perhitungan Fitness Cost adalah sebagai berikut:

Pada setiap *gene* dalam sebuah kromosom dilakukan empat macam perhitungan berikut:

##### a. Fitness Cost Kapasitas Ruang (FCKR)

- Bila kondisi terpenuhi:

$$FCKR = \left[ \frac{\dots(1)}{\left[ \frac{\text{Kapasitas matakuliah}[i]}{\text{Kapasitas ruangan}} \times \text{Fitness kapasitas ruang} \right]} \right]$$

Bila kondisi tidak terpenuhi:

$$FCKR = 0 \dots\dots\dots(2)$$

Nilai default dari Fitness Kapasitas Ruang = 55

##### b. Fitness Cost Letak Ruang (FCLR)

- Bila kondisi terpenuhi:

$$FCLR = \text{Fitness letak ruang} = 25 \dots\dots\dots(3)$$

- Bila kondisi tidak terpenuhi:

$$FCLR = 0 \dots\dots\dots(4)$$

##### c. Fitness Cost Status Matakuliah (FCSM)

- Bila kondisi terpenuhi:

$$FCSM = \text{Fitness status matakuliah} = 15 \dots\dots(5)$$

- Bila kondisi tidak terpenuhi:

$$FCSM = 0 \dots\dots\dots(6)$$

##### d. Fitness Cost Status Dosen (FCSD)

- Bila kondisi terpenuhi:

$$FCSD = \text{Fitness Status Dosen} = 5 \dots\dots\dots(7)$$

- Bila kondisi tidak terpenuhi:

$$FCSD = 0 \dots\dots\dots(8)$$

Selanjutnya total nilai *fitness* dari setiap *gene* dihitung dengan rumus sebagai berikut:

$$\text{Fitness Gene} = FCKR + FCLR + FCSM + FCSD \dots\dots(9)$$

Kemudian nilai total sebuah kromosom dihitung dengan rumus berikut:

$$\text{Total Fitness Cost} = \sum_{i=0}^n \text{Nilai}[i] \dots\dots\dots(10)$$

### 5. OUTPUT APLIKASI

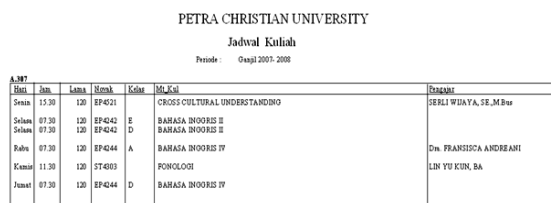
Selain dapat dilihat pada layar monitor, jadwal penggunaan ruang kuliah dapat dicetak untuk jadwal tiap ruang yang ada dan jadwal untuk tiap jurusan. Hal ini disesuaikan dengan dua macam form jadwal manual yang dikeluarkan oleh BAA, yaitu:

- Form jadwal untuk tiap ruang yang nantinya akan ditempel pada pintu tiap ruangan.
- Form jadwal untuk tiap jurusan. Form ini didistribusikan ke jurusan - jurusan yang bersangkutan untuk ditempel pada papan pengumuman.

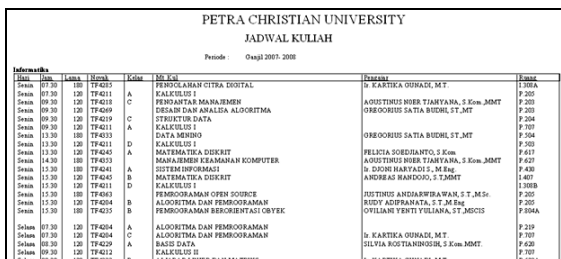
Tampilan output dari aplikasi penjadwalan penggunaan ruang kuliah ini dapat dilihat pada Gambar 5, Gambar 6 dan Gambar 7.



Gambar 5. Tampilan output untuk melihat hasil generate jadwal ruang kuliah.



Gambar 6. Hasil Cetak jadwal berdasarkan ruang



Gambar 7. Hasil cetak jadwal berdasarkan jurusan

## 6. PENGUJIAN

Pengujian dilakukan dengan komputer dengan spesifikasi sebagai berikut:

- Prosesor: Intel Core 2 Duo T5300 1.73 GHz
- Memory: 1GB DDR2
- Hard Disk: 160 GB
- O/S: Windows XP
- Database: SQL Server 2000
- Compiler: Ms. Visual Basic 6

Ada tiga macam pengujian yang dilakukan, yaitu: Pengujian kecepatan proses cGA dalam generate jadwal, Pengujian ketepatan/akurasi hasil generate jadwal dibandingkan dengan proses manual dan pengujian oleh calon pengguna. Hasil pengujian dapat dilihat pada Tabel 3 dan Tabel 4.

Prosentase ketepatan / akurasi dari pengujian ini dihitung dengan rumus sebagai berikut:

$$Akurasi = \frac{\text{fitness kromosom}}{\text{panjang kromosom} \times \text{nilai max tiap gen}} \times 100\% \dots (11)$$

Tabel 3. Hasil pengujian waktu proses dan akurasi

Jumlah Iterasi	Pengujian ke - n	Waktu	Akurasi (%)	Rata-rata Akurasi (%)
3	1	6m24s	66.35	66.238
	2	6m31s	66.33	
	3	5m58s	66.11	
	4	6m12s	66.09	
5	5	6m50s	66.31	66.21
	1	9m15s	66.39	
	2	9m50s	66.11	
	3	9m2s	66.63	
	4	10m11s	65.74	
7	5	9m8s	66.18	66.944
	1	13m21s	66.82	
	2	12m43s	67.42	
	3	13m40s	66.39	
	4	12m59s	66.51	
9	5	13m30s	67.58	66.24
	1	14m46s	65.65	
	2	15m12s	66.01	
	3	14m40s	66.59	
	4	16m41s	66.17	
11	5	15m24s	66.78	65.814
	1	20m10s	65.24	
	2	19m45s	66.25	
	3	19m12s	65.79	
	4	21m2s	65.47	
13	5	19m31s	66.32	65.934
	1	24m10s	66.37	
	2	22m59s	66.3	
	3	22m40s	65.38	
	4	23m10s	65.47	
15	5	22m19s	66.15	65.714
	1	26m43s	66.17	
	2	26m15s	65.12	
	3	27m9s	65.86	
	4	26m59s	65.19	
17	5	26m34s	66.23	64.682
	1	27m10s	65.14	
	2	25m41s	64.72	
	3	25m23s	64.67	
	4	26m38s	64.2	
19	5	25m	64.68	64.676
	1	27m28s	64.63	
	2	29m41s	65.76	
	3	28m9s	64.85	
	4	28m51s	64.12	
<b>Rata - rata akurasi semua pengujian:</b>			<b>65.828</b>	

Sementara itu akurasi proses manual dari set data yang sama adalah 61.27%. Capture modul untuk menghitung akurasi dari proses manual dapat dilihat pada Gambar 8.



Gambar 8. Hasil uji akurasi proses manual

Dari hasil pengujian ketepatan/akurasi dapat ditarik kesimpulan bahwa akurasi rata – rata jadwal yang di-generate oleh program lebih baik dari jadwal pemakaian ruang yang dibuat secara manual.

Tabel 4. Hasil Kuisisioner pengguna / nara sumber

Nomor	Kriteria	Nilai
1	Desain <i>interface</i>	4
2	Program sesuai kebutuhan user	4
3	Kemudahan dalam menggunakan program	3
4	Output mudah dipahami	4
5	Keakuratan output yang dihasilkan	4

Ket: Range nilai: 5 Sangat Baik s/d 1 Sangat Kurang

Kuisisioner ini diisi oleh karyawan BAA yang bertugas untuk meng-generate jadwal penggunaan ruang kuliah. Selain sebagai calon pengguna karyawan tersebut juga terlibat dalam penelitian sebagai nara sumber. Pada hasil kuisisioner ini dapat dilihat bahwa aplikasi dianggap cukup baik dan akurat oleh calon pengguna.

## 7. KESIMPULAN

Dari hasil pengujian secara keseluruhan didapatkan kesimpulan bahwa *compact Genetic Algorithm* (cGA) baik bila digunakan untuk melakukan optimasi penjadwalan yang memiliki sekuen jadwal panjang dan aturan / kriteria yang cukup banyak. Terbukti pada penelitian ini, dengan panjang kromosom lebih dari 1000 *gene*, waktu proses yang dibutuhkan, untuk mendapatkan hasil dengan akurasi lebih baik dari proses manual, tidak pernah lebih dari 30 menit. Dari hasil kuisisioner yang diisi oleh pengguna merangkap nara sumber terbukti pula bahwa hasil *generate* jadwal dengan cGA cukup akurat dan dapat digunakan.

## PUSTAKA

- Budhi, Gregorius S., Sundoro, David, dan Pongawa, Vince. (2008). Penggunaan Compact Genetic Algorithm (cGA) untuk Optimasi Penjadwalan Pengiriman Beton Ready-Mix. *Proceedings Konferensi Nasional Sistem & Informatika 2008*.
- Harik, George R., Lobo, Fernando G., dan Goldberg, David E. (1997). *The Compact Genetic Algorithm*. Urbana: University of Illinois, USA.
- Harik, George R., Lobo, Fernando G., dan Goldberg, David E. (1998). The Compact Genetic Algorithm. *Proceedings of the 1998 IEEE Conference on Evolutionary Computation*, pp. 523-528.
- Harik, George R., Lobo, Fernando G., dan Goldberg, David E. (1999). *The Compact Genetic Algorithm*. *IEEE Transaction on Evolutionary Computation*, 1999, 3(4): 287-297.

Holland J. H. (1975). *Adaption in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, USA.

Langdon, William B. (2000). *Genetic Programming And Data Structures: Genetic Algorithms + Data Structures = Automatic Programming!*. Kluwer Academic Publisher.

Langdon, William B. dan Poli, Ricardo. (1998). *Foundation of Genetic Programming*. Springer.

Negnevitsky, Michael. (2005). *Artificial Intelligence: A Guide to Intelligent Systems*. 2<sup>nd</sup> Edition. Addison Wesley.

Rimcharoen, S., Daricha, S., dan Prabhas, C. (2006). *Real options approach to finding optimal stopping time in compact genetic algorithm*. Bangkok: Chulalongkorn University. Diakses pada April 2007 dari <http://realoptions.org/papers2006/>