

## IMPLEMENTASI ALGORITMA GENETIKA DALAM PENGEMBANGAN SISTEM APLIKASI PENJADWALAN KULIAH

Oleh :

**Komang Setemen**

Jurusan Manajemen Informatika Fakultas Teknik dan Kejuruan  
Universitas Pendidikan Ganesha

### ABSTRAK

Pada penelitian ini dilakukan pengembangan sistem aplikasi penjadwalan kuliah dengan mengimplementasikan algoritma genetika. Sistem aplikasi penjadwalan kuliah ini dikembangkan karena masalah penjadwalan kuliah dalam pendidikan merupakan persoalan khusus dari masalah optimasi, yang membutuhkan waktu komputasi yang cukup tinggi untuk pencarian solusinya, terlebih lagi jika ukuran permasalahan semakin besar dengan bertambahnya jumlah komponen dan tetapan atau syarat yang ditentukan oleh institusi tempat jadwal tersebut digunakan. Sebagai ujicoba kemampuan sistem yang dikembangkan untuk pembuatan jadwal mata kuliah, digunakan data mata kuliah semester genap tahun akademik 2007/2008 pada Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha. Implementasi perangkat lunak yang dikembangkan, diujicobakan pada tiga kelompok data dengan jumlah mata kuliah setiap kelompok data berbeda-beda. Dari ujicoba ketiga kelompok data didapatkan hasil yakni, algoritma genetika cukup baik digunakan sebagai algoritma membuat jadwal mata kuliah, hal ini dapat dilihat dari hasil Ujicoba I sampai dengan Ujicoba III yang telah dilakukan. Keberhasilan ini dapat dilihat dari nilai *fitness* sama dengan nol yang dihasilkan setiap kali ujicoba, yang menandakan bahwa jadwal yang dihasilkan sudah tidak ada bentrok.

**Kata-kata kunci:** sistem aplikasi, jadwal kuliah, algoritma genetika.

### 1. PENDAHULUAN

Masalah penjadwalan dalam institusi pendidikan, masih menjadi isu yang menarik dan secara luas masih diteliti oleh banyak peneliti di dunia. Masalah penjadwalan dapat diklasifikasikan ke dalam dua kategori utama, yaitu: masalah penjadwalan kuliah dan masalah penjadwalan ujian (Aldy Gunawan, dkk.,2004). Schaerf, dalam S'andor (2005) mengusulkan klasifikasi dari masalah penjadwalan ini dibagi ke dalam tiga klasifikasi utama, yaitu: penjadwalan sekolah, penjadwalan kuliah dan penjadwalan ujian (S'andor, dkk.,2005). Masalah penjadwalan dalam pendidikan merupakan persoalan khusus dari masalah

optimasi yang ditemukan pada situasi nyata. Masalah ini membutuhkan waktu komputasi yang cukup tinggi untuk pencarian solusinya, terlebih lagi jika ukuran permasalahan semakin besar dengan bertambahnya jumlah komponen dan tetapan atau syarat yang ditentukan oleh institusi tempat jadwal tersebut digunakan. Selama proses, banyak aspek yang harus dipertimbangkan untuk memperoleh jadwal yang optimal, dan seringkali tidak dapat memuaskan karena tidak semua kebutuhan terpenuhi. Oleh karena itu perlu ditetapkan suatu batasan dalam penyusunan jadwal yang bersifat harus dipenuhi (*hard constraints*) dan tidak harus dipenuhi (*soft constraints*), tetapi tetap menjadi acuan dalam proses pembuatan jadwal.

Pada penelitian ini pembahasan difokuskan pada penjadwalan mata kuliah di perguruan tinggi. Penjadwalan mata kuliah merupakan persoalan umum dan sangat kompleks, karena harus mempertimbangkan tetapan-tetapan sesuai dengan kebijakan perguruan tinggi. Tujuan dari penjadwalan adalah bagaimana mengatur sejumlah komponen dengan tetapan dan syarat tertentu dalam satu waktu. Dalam penjadwalan mata kuliah ini komponen yang dimaksud adalah mahasiswa, dosen, mata kuliah, ruang dan waktu perkuliahan. Selain komponen-komponen tersebut, hal lain yang perlu diperhatikan adalah karakteristik dari mata kuliah. Mata kuliah yang berhubungan dengan praktikum harus ditempatkan di laboratorium sedangkan mata kuliah teori di tempatkan di ruang kuliah biasa. Jika pengalokasian dari komponen-komponen ini tidak di lakukan dengan baik, maka akan mengakibatkan penumpukan jadwal pada tempat dan waktu tertentu. Sehingga perlu dilakukan pengaturan mahasiswa, dosen, mata kuliah, ruang dan waktu secara optimal untuk menghindari penumpukan jadwal pada waktu tertentu.

Dengan kemajuan ilmu pengetahuan dalam bidang komputasi cerdas, maka masalah penjadwalan di atas dapat diotomatisasi sehingga dapat memberikan solusi optimal sesuai dengan batasan dan syarat yang sudah ditentukan. Sejumlah metode dari berbagai disiplin ilmu telah diusulkan dalam literatur, seperti: riset operasi, kecerdasan buatan, dan kecerdasan komputasional. Metode-metode tersebut dapat di bagi ke dalam 4 (empat) kategori, yaitu: *Sequential Methods*,

*Cluster Methods, Constraint Based Methods, dan Meta-heuristic Methods*, seperti *genetic algorithms, simulated annealing, dan tabu search* (Spyros, dkk., 2005).

Pada penelitian ini di uji cobakan otomatisasi masalah penjadwalan kuliah di perguruan tinggi dengan metode *meta-heuristic*, yaitu: Algoritma Genetika. Permasalahan yang ditangani pada penelitian ini adalah 1) pengelolaan komponen-komponen jadwal, seperti: data dosen, mahasiswa, ruangan, mata kuliah, dan waktu perkuliahan 2) pengelolaan pemetaan mata kuliah 3) pembuatan jadwal mata kuliah, dan 4) penyimpanan data hasil jadwal. Di harapkan dengan menggunakan algoritma ini akan dapat menyelesaikan masalah penjadwalan kuliah di perguruan tinggi, karena algoritma genetika dapat menyelesaikan permasalahan yang kompleksitasnya tinggi (Anita Desiani dan Muhammad Arhami, 2006).

*Genetic Algorithms* atau algoritma genetika adalah algoritma pencarian yang didasarkan 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 yang mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: “Bagaimana mendapatkan keturunan yang lebih baik” (Ahmad Basuki, 2003). Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (David E. Golberg, 1989).

## 2. PEMBAHASAN

Penelitian ini merupakan penelitian rekayasa sistem perangkat lunak, sehingga dalam pelaksanaan penelitian ini, digunakan metodologi standar dalam

pengembangan perangkat lunak. Metodologi yang digunakan adalah prosedur *Life Cycle System* yang dikenal dengan *waterfall model* yang merupakan metodologi tertua dan terbanyak digunakan dalam pengembangan perangkat lunak. *Life Cycle System* merupakan suatu proses penyusunan sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang sudah ada (Edward Yourdon, 1989).

Tahapan aktivitas yang dilakukan dalam prosedur implementasi perangkat lunak ini merupakan modifikasi dari tahapan-tahapan prosedur *Life Cycle System* di atas, meliputi :

- 1) Tahap Pengumpulan Data (*System Requirement*)
- 2) Tahap Analisis (*Analysis*)
- 3) Tahap Desain/Perancangan (*Design*)
- 4) Tahap Pengembangan (*Development / Coding*)
- 5) Tahap Uji Coba (*Testing*)
- 6) Tahap Penerapan (*Implementasi*)

Pengumpulan data dapat dilakukan dengan cara wawancara terkait dengan kebutuhan sistem kepada pengguna, karena para mahasiswa, pegawai, dosen serta staf yang lain nanti akan menggunakan sistem ini, sehingga dapat memberikan masukan-masukan yang bermanfaat, dan sistem yang dikembangkan ini dapat memenuhi secara maksimal kebutuhan pengguna. Setelah mendapatkan data dari hasil survei, maka dilakukan analisis yang meliputi kegiatan: pemilahan informasi yang diperoleh dari survei, membuat alur kerja (*Work Flow*), menyusun dokumen *User Specification*. Tugas yang paling penting dalam tahap ini adalah proses menemukan permasalahan dan menghasilkan alternatif pemecahan masalah. Dalam tahap ini akan dianalisis kebutuhan dari sistem yang akan dibuat. Analisis yang dimaksud adalah menentukan informasi-informasi apa saja yang akan dimasukkan ke dalam sistem ini, sehingga benar-benar dapat bermanfaat bagi pengguna (mahasiswa, pegawai, dosen, staf yang lain serta masyarakat luas).

Tahap desain dilakukan apabila tahapan analisis sistem sudah terpenuhi. Kegiatan-kegiatan yang dilakukan pada tahap Desain/Perencanaan ini adalah

mengkonversi *User Specification* menjadi dokumen teknis, menyusun pemodelan untuk menentukan modul-modul apa saja yang perlu dibuat, menyusun DFD (*Data Flow Diagram*), menyusun ERD (*Entity Relationship Diagram*), dan menyusun Dokumen Standar. Tahap berikutnya adalah melakukan pengembangan aplikasi yang merupakan hasil dari desain. Tahap pengembangan ini meliputi kegiatan: (1) membuat database di RDMS termasuk tabel, trigger, relationship, dll., (2) menyusun program sesuai dengan dokumen teknis dan dokumen standar. Program ini akan dikembangkan menggunakan Borland Delphi 7.0, salah satu bahasa pemrograman yang *reliable* untuk digunakan dalam pengembangan sistem ini, (3) melakukan optimasi terhadap program yang dibuat, sehingga dapat memiliki *respons time* yang kecil (akses informasi dapat berjalan dengan cepat).

Setelah selesai membuat kode program maka dilakukan testing terhadap Sistem Penjadwalan Kuliah ini. Testing ini meliputi: validasi terhadap data masukan, pemetaan pengasuh mata kuliah, ruangan, serta *interface* aplikasi. Jika terjadi ketidaksesuaian terhadap validasi di atas, maka sistem ini harus segera diperbaiki. Baik itu dari sisi databasenya maupun programnya. Proses ini akan dilakukan bersama-sama dengan user/pengguna. Sehingga *user* dapat memberikan pertimbangan-pertimbangan ataupun masukan-masukan sehingga dapat tercipta sebuah sistem yang sesuai dengan kebutuhan user. Tahap implementasi merupakan tahap pengimplementasian terhadap Sistem Penjadwalan Matakuliah yang telah dikembangkan. Sistem ini siap digunakan oleh semua Jurusan di lingkungan Fakultas Teknik dan Kejuruan UNDIKSHA Singaraja, tetapi tidak menutup kemungkinan jika sistem ini juga akan digunakan oleh Fakultas-fakultas lain yang ada di UNDIKSHA, dengan beberapa perubahan data.

Setelah melalui tahapan-tahapan dalam prosedur *Life Cycle System*, berikut ini adalah hasil dari antar muka perangkat lunak yang telah dikembangkan. Perancangan antar muka ini dibuat dengan menggunakan bahasa pemrograman *Borland Delphi 7.0* dan database *MySQL Server 5.0* dengan sistem operasi *Windows XP Professional*. Hasil perancangan antar muka seperti pada Gambar 1.

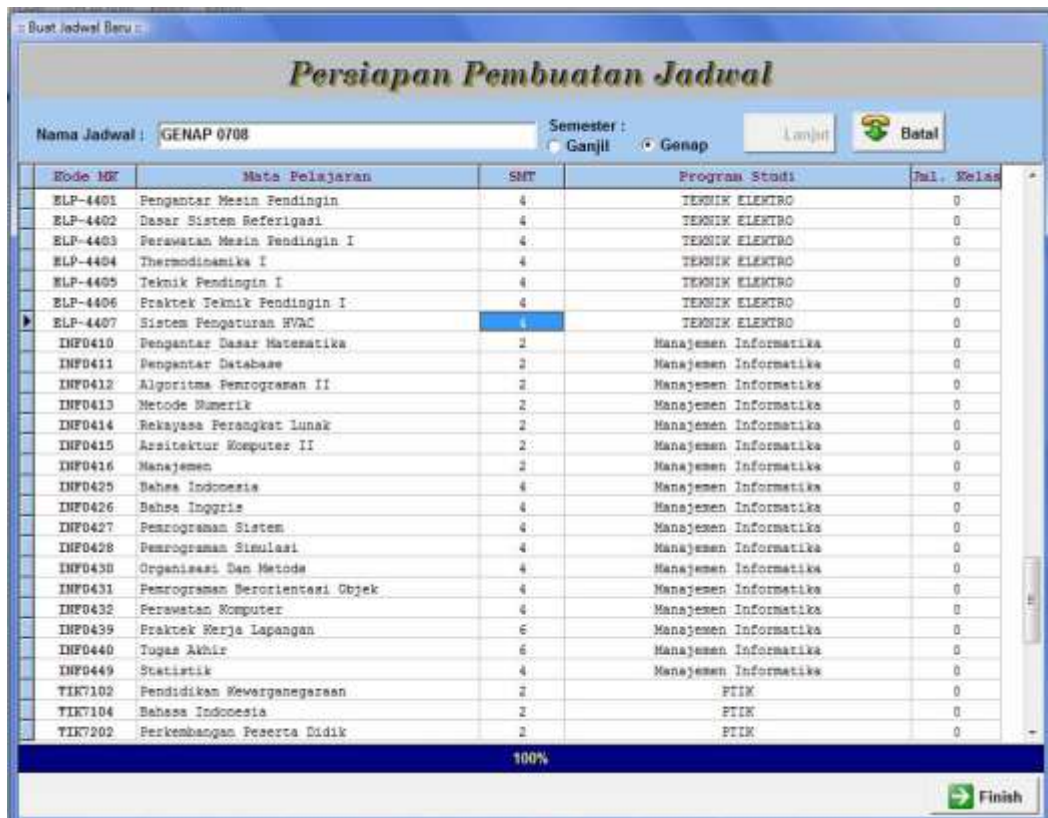


Gambar 1. Tampilan Utama Perangkat Lunak

Secara umum ada 5 (lima) langkah utama yang dilakukan untuk memproses penjadwalan mata kuliah, yaitu: input semua data pendukung penjadwalan yang terdapat pada menu *Master*, membuat atau membuka jadwal baru pada menu *File*, pemetaan pengajar mata kuliah beserta ruangan yang digunakan pada menu *Pemetaan*, proses *generate* data penjadwalan pada menu *Generate*, dan menampilkan hasil penjadwalan pada menu *Report*. Berikut akan ditampilkan beberapa *interface* dari langkah-langkah tersebut.

### ***Interface* Pembuatan Jadwal**

*Interface* pembuatan jadwal adalah untuk membuat jadwal baru yang digunakan pada proses pemetaan dan *generate* jadwal. Adapun *interface* dari pembuatan jadwal adalah tampak seperti pada Gambar 2.





Gambar 2. Interface Pembuatan Jadwal

Pada *edit box* dari *interface* tersebut dimasukkan nama jadwal yang akan dibuat. Setelah nama jadwal baru diisikan dan mengklik tombol lanjut, kemudian akan ditampilkan semua mata kuliah yang ada pada Jurusan Teknik Elektro. Pada kolom jumlah kelas, diisikan jumlah kelas yang akan dibuka untuk mata kuliah bersangkutan. Jika mata kuliah tersebut tidak ditawarkan, maka jumlah kelas diisi dengan 0, dan jika jumlah kelas sudah diisikan, maka dilanjutkan dengan menekan tombol *Finish*, agar semua mata kuliah dengan jumlah kelas yang dibuka dimasukkan ke dalam tabel pemetaan.

**Interface Pemetaan Matakuliah**

*Interface* pemetaan matakuliah adalah *interface* untuk memasukkan nama-nama pengampu matakuliah serta ruangan yang digunakan, sesuai dengan nama jadwal yang sudah dibuat sebelumnya. *Interface* pemetaan matakuliah seperti ditunjukkan pada Gambar 3.

Pemetaan Mata Kuliah Program Studi :: Manajemen Informatika

Cari Nama Mata kuliah   Cari  Tutup

Kode Mata Kuliah	Mata Kuliah	Semester	Kelas	Dosen	Kategori Ruangan
INF0410	Pengantar Dasar Matematika	2	A	Ni Wayan Marti, S.kom. M.kom	RUANG KULIAH UM
INF0414	Rekayasa Perangkat Lunak	2	A	Ni Ketut Kertiasih, S.Si.	LAB INFORMATIKA
INF0414	Rekayasa Perangkat Lunak	2	B	Ni Ketut Kertiasih, S.Si.	LAB INFORMATIKA
INF0414	Rekayasa Perangkat Lunak	2	C	Ni Ketut Kertiasih, S.Si.	LAB INFORMATIKA
INF0415	Arsitektur Komputer II	2	A	I Ketut Purnamawan, S.Kom	LAB INFORMATIKA
INF0415	Arsitektur Komputer II	2	B	I Ketut Purnamawan, S.Kom	LAB INFORMATIKA
INF0415	Arsitektur Komputer II	2	C	I Ketut Purnamawan, S.Kom	LAB INFORMATIKA
INF0416	Manajemen	2	A	Prof. Dr. I Made Candiasa, M.I.Komp	RUANG KULIAH UM
INF0416	Manajemen	2	B	Prof. Dr. I Made Candiasa, M.I.Komp	RUANG KULIAH UM
INF0416	Manajemen	2	C	Prof. Dr. I Made Candiasa, M.I.Komp	RUANG KULIAH UM
INF0413	Metode Numerik	2	C	Komang Setemen, S.Si.	LAB INFORMATIKA
INF0413	Metode Numerik	2	B	Komang Setemen, S.Si.	LAB INFORMATIKA
INF0410	Pengantar Dasar Matematika	2	B	Ni Wayan Marti, S.kom. M.kom	RUANG KULIAH UM
INF0410	Pengantar Dasar Matematika	2	C	Ni Wayan Marti, S.kom. M.kom	RUANG KULIAH UM
INF0411	Pengantar Database	2	A	I Gede Rasben Dantes, ST.,M.T.I.	LAB INFORMATIKA
INF0411	Pengantar Database	2	B	I Gede Rasben Dantes, ST.,M.T.I.	LAB INFORMATIKA
INF0411	Pengantar Database	2	C	I Gede Rasben Dantes, ST.,M.T.I.	LAB INFORMATIKA
INF0412	Algoritma Pemrograman II	2	A	Drs. I Nyoman Sukajaya, M.T	RUANG KULIAH UM
INF0412	Algoritma Pemrograman II	2	B	Drs. I Nyoman Sukajaya, M.T	RUANG KULIAH UM
INF0412	Algoritma Pemrograman II	2	C	Drs. I Nyoman Sukajaya, M.T	RUANG KULIAH UM
INF0413	Metode Numerik	2	A	Komang Setemen, S.Si.	LAB INFORMATIKA
INF0430	Organisasi Dan Metode	4	B	I Gede Rasben Dantes, ST.,M.T.I.	RUANG KULIAH UM
INF0430	Organisasi Dan Metode	4	C	I Gede Rasben Dantes, ST.,M.T.I.	RUANG KULIAH UM
INF0431	Pemrograman Berorientasi Objek	4	A	Luh Joni Erawati Dewi, ST	LAB INFORMATIKA
INF0431	Pemrograman Berorientasi Objek	4	B	Luh Joni Erawati Dewi, ST	LAB INFORMATIKA
INF0431	Pemrograman Berorientasi Objek	4	C	Luh Joni Erawati Dewi, ST	LAB INFORMATIKA
INF0432	Perawatan Komputer	4	A	Putu Hendra Suputra, S.Kom	LAB INFORMATIKA
INF0432	Perawatan Komputer	4	B	Putu Hendra Suputra, S.Kom	LAB INFORMATIKA
INF0432	Perawatan Komputer	4	C	Putu Hendra Suputra, S.Kom	LAB INFORMATIKA

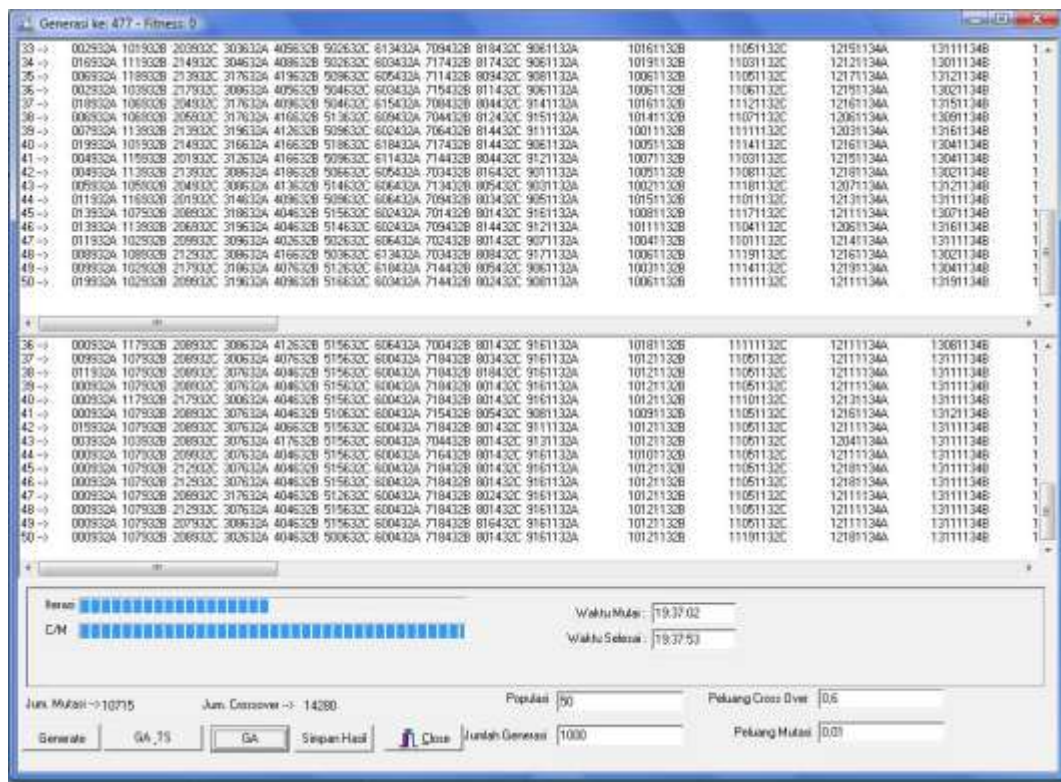
Gambar 3. *Interface* Pemetaan Matakuliah

Pada Gambar 3. terlihat bahwa matakuliah-matakuliah yang ditawarkan pada semester aktif sesuai dengan jadwal yang sudah dibuat sebelumnya. Dari *interface* pemetaan ini, dapat dilakukan pengisian nama dosen yang mengampu mata kuliah dan jenis ruangan yang digunakan pada matakuliah tersebut.

### ***Interface Generate Jadwal***

*Interface generate* jadwal adalah *interface* yang digunakan untuk membuat populasi awal dan proses regenerasi populasi dari nama jadwal yang sudah dipetakan. *Interface generate* jadwal ini tampak seperti pada Gambar 4.





Gambar 4. Interface Generate Jadwal

Pada gambar 4. dapat dilihat bahwa, populasi awal yang dibuat adalah sebanyak 50 (lima puluh), jumlah generasi yang dibentuk adalah 1000 (seribu), dan peluang *crossover* dan mutasi masing-masing adalah 0,6 dan 0,01. Parameter kontrol ini sesuai dengan aturan De Jong, yang mengisyaratkan untuk permasalahan yang mempunyai ruang solusi yang cukup besar digunakan parameter kontrol sesuai dengan nilai-nilai tersebut. Pada gambar tampak dengan jelas bahwa dari 1000 generasi yang direncanakan, ternyata setelah generasi ke-78 solusi sudah optimal yang ditunjukkan dengan nilai *fitness* sama dengan 0 (nol), yang berarti bahwa dari hasil regenerasi sampai dengan generasi ke-78 sudah tidak terjadi bentrokan jadwal.

Untuk menguji kemampuan perangkat lunak yang sudah dikembangkan, dilakukan dengan pengujian beberapa kelompok data. Setiap hasil ujicoba dari setiap kelompok data, akan diperlihatkan perbedaan hasil dari algoritma genetika berdasarkan kriteria berikut.

1. jumlah data yang diolah

2. nilai *fitness* yang dihasilkan, dan
3. jumlah iterasi untuk mencapai nilai *fitness* optimal.

Dari ketiga kriteria tersebut di atas, akan dapat dilihat kinerja hasil yang diperoleh dari penggunaan algoritma genetika. Pengujian perangkat lunak dengan parameter jumlah data masukan, dimaksudkan untuk mengetahui unjuk kerja perangkat lunak dilihat dari jumlah data yang diolah. Jumlah data yang diolah pada pengujian ini didasarkan atas jumlah program studi yang ada, dan jumlah kelas yang dibuka pada masing-masing program studi. Ujicoba yang menggunakan data Fakultas Teknik dan Kejuruan Semester Genap Tahun Akademik 2007/2008, dilakukan 3 (tiga) tahap pengujian yang didasarkan atas jumlah data yang diolah.

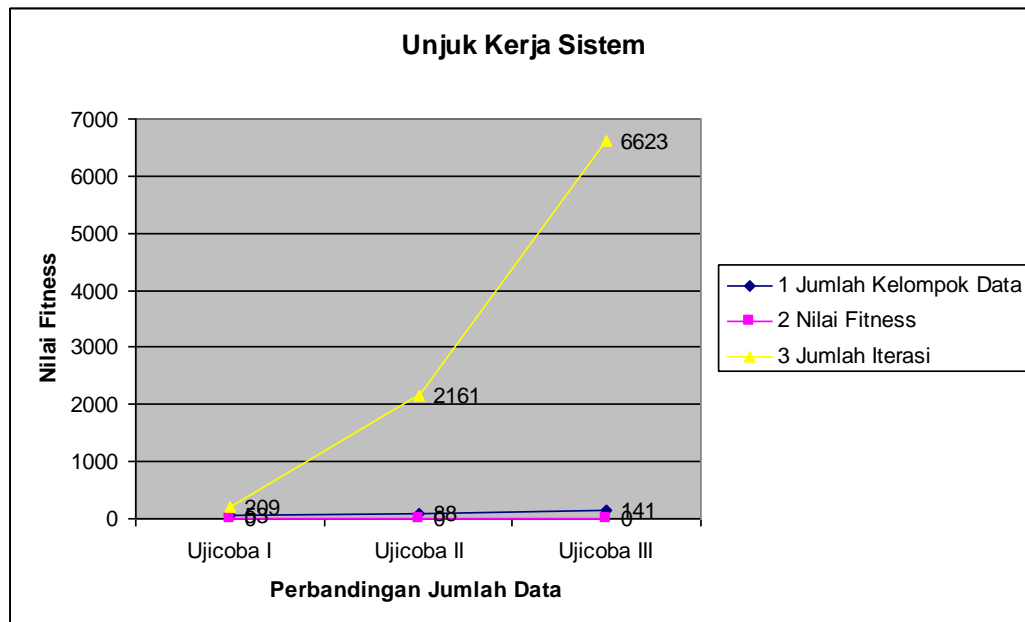
Ujicoba I dilakukan pada 53 jenis pemetaan mata kuliah, dengan rincian jumlah ruangan maksimal yang bisa digunakan adalah sebanyak 14 buah termasuk ruang kuliah biasa dan laboratorium, jumlah dosen pengampu mata kuliah 26 orang. Dari spesifikasi tersebut diperoleh hasil bahwa algoritma genetika mampu menyelesaikan pembuatan jadwal dengan tidak ada bentrokan sama sekali, yang ditunjukkan dengan nilai *fitness* sama dengan 0. Jumlah iterasi yang dibutuhkan dari algoritma ini adalah sebanyak 209 iterasi dari 1000 iterasi yang direncanakan.

Ujicoba II dilakukan pada 88 jenis pemetaan mata kuliah, dengan rincian jumlah mata kuliah ada 58 (ada beberapa mata kuliah kelas paralel), jumlah ruangan maksimal yang bisa digunakan adalah sebanyak 14 buah termasuk ruang kuliah biasa dan laboratorium, jumlah dosen pengampu mata kuliah 28 orang. Dari spesifikasi tersebut diperoleh hasil bahwa algoritma genetika mampu menyelesaikan pembuatan jadwal dengan tidak ada bentrokan sama sekali, yang ditunjukkan dengan nilai *fitness* sama dengan 0. Jumlah iterasi yang dibutuhkan dari algoritma ini adalah sebanyak 2161 iterasi dari 5000 iterasi yang direncanakan.

Ujicoba III dilakukan pada 141 jenis pemetaan mata kuliah (semua jurusan pada FTK), dengan rincian jumlah mata kuliah ada 101 (ada beberapa mata kuliah kelas paralel), jumlah ruangan maksimal yang bisa digunakan adalah sebanyak 14 buah termasuk ruang kuliah biasa dan laboratorium, jumlah dosen pengampu mata

kuliah 49 orang. Dari spesifikasi tersebut diperoleh hasil bahwa algoritma genetika mampu menyelesaikan pembuatan jadwal dengan tidak ada bentrokan sama sekali, yang ditunjukkan dengan nilai *fitness* sama dengan 0. Jumlah iterasi yang dibutuhkan dari algoritma ini adalah sebanyak 6623 iterasi dari 10.000 iterasi yang direncanakan.

Dari ketiga ujicoba tersebut di atas, dapat dilihat bahwa penerapan algoritma genetika sebagai algoritma generator jadwal mata kuliah, cukup baik digunakan karena telah dapat menyelesaikan pembuatan jadwal mata kuliah. Berikut ini adalah grafik perkembangan jumlah iterasi yang harus dilakukan untuk mencapai *fitness* 0 (nol), adalah seperti pada Gambar 5.



Gambar 5. Grafik Perkembangan Jumlah Iterasi

### 3. PENUTUP

Beberapa simpulan yang dapat diambil dari analisa dan ujicoba pengembangan sistem aplikasi generator jadwal mata kuliah menggunakan algoritma genetika adalah sebagai berikut.

Sistem aplikasi generator jadwal mata kuliah mampu menangani pengelolaan komponen-komponen jadwal, pengelolaan pemetaan mata kuliah, pembuatan jadwal mata kuliah, dan penyimpanan hasil jadwal mata kuliah.

Algoritma genetika cukup baik diimplementasikan sebagai algoritma untuk membuat sistem aplikasi penjadwalan mata kuliah, hal ini dapat dilihat dari hasil Ujicoba I sampai dengan Ujicoba III yang telah dilakukan. Keberhasilan ini dapat dilihat dari nilai *fitness* sama dengan nol yang dihasilkan setiap kali ujicoba, yang menandakan bahwa jadwal yang dihasilkan sudah tidak ada bentrok.

Untuk jumlah kelompok data yang kecil (53 sampai 88), algoritma genetika dapat menyelesaikan masalah penjadwalan dengan baik, dengan nilai *fitness* sama dengan 0 dan iterasi atau jumlah generasi maksimum 5000 generasi. Sedangkan untuk data yang cukup besar yaitu dengan jumlah pemetaan 141 mata kuliah, algoritma genetika masih mampu memperoleh *fitness* sama dengan 0, tetapi dengan jumlah iterasi atau generasi maksimum 10.000 generasi.

Adapun saran yang dapat diberikan guna penelitian lebih lanjut tentang kombinasi kedua algoritma ini adalah dengan mengujicobakan pada data yang berbeda dengan jumlah data yang lebih besar, dan membandingkannya dengan metode atau algoritma yang lain.

#### DAFTAR PUSTAKA

- Ahmad Basuki. 2003. Algoritma Genetika, Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning. PENS-ITS Surabaya.
- Aldy Gunawan, Hoon Liong Ong and Kien Ming Ng. 2004. APPLYING METAHEURISTICS FOR THE COURSE SCHEDULING PROBLEM. *Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference 2004*.
- Anita Desiani & Muhammad Arhami. 2006. Konsep Kecerdasan Buatan. Andi Offset. Yogyakarta.
- David E. Goldberg. 1989. Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley Publishing Company, Inc.
- Edward Yourdon, 1989. *Modern Structured Analysis*. New Jersey : Prentice - Hall International, Inc.
- Spyros Kazarlis Vassilios Petridis and Pavlina Fragkou. 2005. Solving University Timetabling Problems Using Advanced Genetic Algorithms. *Technological Educational Institute of Serres, Serres 621 24 Greece dan Aristotle University of Thessaloniki,, Thessaloniki 540 06, Greece*

Sándor Györi Zoltán Petres Annamária R. Varkonyi-Kóczy. 2005. Genetic Algorithms in Timetabling. A New Approach. Budapest University of Technology and Economics, Department of Measurement and Information Systems, Műegyetem rkp. 9., Budapest, Hungary, H-1521.