

---

# IMPLEMENTASI ALGORITMA GENETIKA PADA PERANCANGAN SISTEM PENJADWALAN PERKULIAHAN DI STIKOM BALI

Ni Made Wangi Suryati, Ni Ketut Dewi Ari Jayanti, I Ketut Dedy Suryawan  
Sekolah Tinggi Manajemen Informatika dan Teknik Komputer (STMIK) STIKOM Bali  
Jln. Raya Puputan No. 86 Renon-Denpasar, 0361-244445  
email : [wangi.suryati@yahoo.com](mailto:wangi.suryati@yahoo.com), [daj@stikom-bali.ac.id](mailto:daj@stikom-bali.ac.id), [dedymeng@stikom-bali.ac.id](mailto:dedymeng@stikom-bali.ac.id)

## Abstrak

*Penjadwalan perkuliahan dalam sebuah institusi pendidikan seringkali menjadi kendala, sebab dalam penyusunan sebuah jadwal perkuliahan dengan skala besar, membutuhkan waktu yang cukup lama dengan kompleksitas yang tinggi. Untuk menyusun jadwal perkuliahan, harus mempertimbangkan beberapa komponen seperti matakuliah, ruang perkuliahan, waktu berlangsungnya proses perkuliahan, dan karakteristik mata kuliah. Selain itu penjadwalan perkuliahan juga harus mempertimbangkan kebijakan dari institusi pendidikan tersebut. Apabila dalam penjadwalan perkuliahan tidak dilakukan dengan tepat, akan menjadi kendala di kemudian hari, misalnya penumpukan jadwal mata kuliah di ruangan yang sama, hal tersebut dapat berdampak pada proses perkuliahan, sehingga proses perkuliahan menjadi terganggu. Pembuatan sistem penjadwalan perkuliahan dengan menggunakan algoritma genetika ini mempermudah penyusunan jadwal perkuliahan dengan cara menggunakan metode genetika dalam proses penjadwalannya, karena pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik, sehingga proses penjadwalan di STIKOM Bali dapat mengurangi kesalahan dalam proses penjadwalan.*

**Kata Kunci :** Penjadwalan, Algoritma Genetika, Aplikasi

## Abstract

*Scheduling of lectures in an educational institution is often becoming an obstacle, for the preparation of a large-scale lecture schedules, as it takes a long time with high complexity. For scheduling lecture, should consider some of the components like the course, lecture hall, the duration of the lecture, and the characteristics of the subjects. Besides scheduling lectures should also consider the policies of the educational institution. If in the lecture scheduling is not done properly, it will become an obstacle in the future, such as the buildup of lecture schedule in the same room, it can have an impact on the lecture, so that the lecture could be disturbed. Making lecture scheduling system using a genetic algorithm to facilitate the preparation of the lecture schedule by using genetic methods in the process of scheduling, due to the approach taken by this algorithm is to combine randomly in any kind of the best solution choice in a collection in order to get a best solution generation, so the process of scheduling in STIKOM Bali can reduce errors or mistake in the scheduling process.*

**Keywords:** Scheduling, Genetics Algorithm, Applications

## 1. Pendahuluan

Penyusunan sebuah jadwal perkuliahan dengan skala besar, membutuhkan waktu yang cukup lama dengan kompleksitas yang tinggi. Selain itu penjadwalan perkuliahan juga harus mempertimbangkan kebijakan dari institusi pendidikan tersebut. Apabila dalam penjadwalan perkuliahan tidak dilakukan dengan tepat, akan menjadi kendala di kemudian hari, misalnya bentrok jadwal mata kuliah di ruangan yang sama, hal tersebut dapat berdampak pada proses perkuliahan, sehingga proses perkuliahan menjadi terganggu.

STIKOM Bali merupakan salah satu sekolah tinggi bidang komputer di Bali. Hingga kini STIKOM Bali mempunyai tiga program studi yaitu Sistem Komputer (S1), Sistem Informasi (S1) dan Manajemen Informatika (D3). Dari data yang diperoleh berdasarkan kurikulum terbaru tahun 2012/2013 pada STIKOM Bali, program studi Sistem Komputer terdapat 52 Mata Kuliah, pada program studi Sistem

---

Informasi terdapat 70 matakuliah, dan pada program studi Manajemen Informatika terdapat 43 matakuliah. Total kelas matakuliah pada semester ganjil tahun ajaran 2012/2013 mencapai 659 kelas, sehingga proses penjadwalan akan lebih kompleks dan memakan waktu yang lama.

Salah satu cara yang dapat digunakan dalam optimasi penjadwalan adalah dengan menggunakan Algoritma Genetika (*Genetic Algorithms*). Algoritma genetika diadaptasi dari gagasan yang menirukan proses evolusi dalam pemecahan suatu masalah. Dalam proses evolusi, individu secara terus menerus mengalami perubahan gen melalui proses perkembangbiakan. Algoritma ini dapat dipakai untuk mendapatkan solusi yang tepat untuk masalah yang kompleks. Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya.

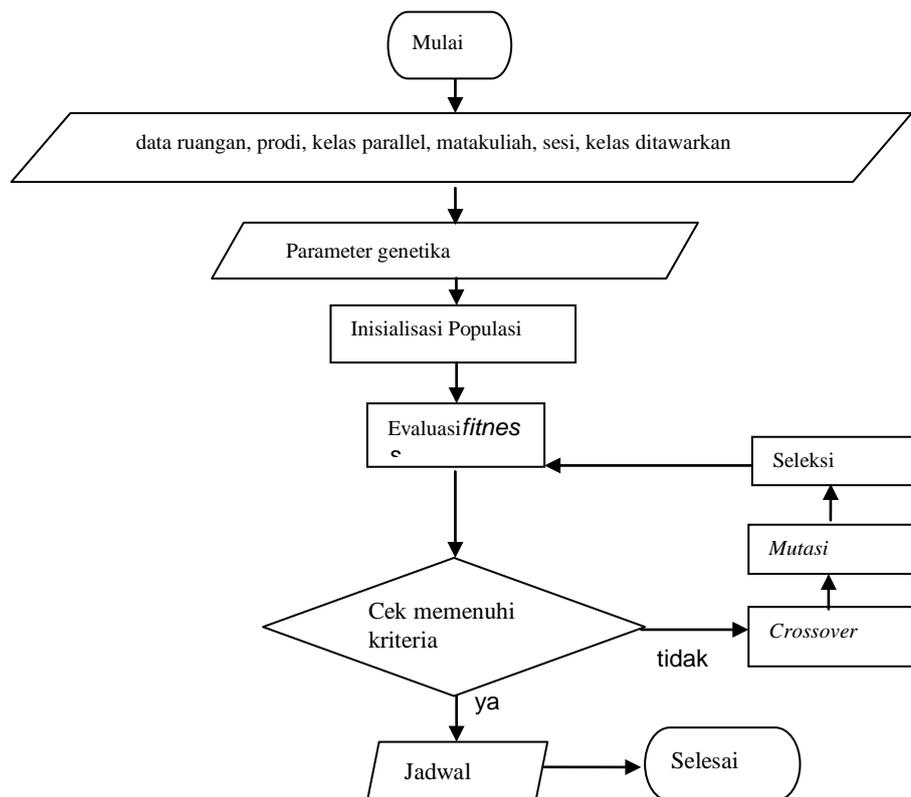
Maka dirancang sebuah sistem panjadwalan perkuliahan di STIKOM Bali berbasis desktop dengan mengimplementasikan algoritma genetika sebagai metode perancangannya. Pembuatan sistem penjadwalan perkuliahan ini diharapkan dapat mempermudah petugas akademik dalam proses penjadwalan. Sistem penjadwalan ini juga dapat digunakan untuk mempersingkat waktu penjadwalan, sehingga proses penjadwalan dapat berjalan dengan efektif. Selain itu dengan dibuatkannya sebuah sistem penjadwalan, maka dapat mengurangi kesalahan dalam proses penjadwalan.

## 2. Metode Penelitian

Penelitian (research) merupakan rangkaian kegiatan ilmiah dalam rangka pemecahan suatu masalah. Penelitian itu sendiri bekerja atas dasar asumsi, teknik dan metode. Tujuan dari penelitian adalah memecahkan suatu masalah dengan menggunakan metode yang sesuai.

Sistem penjadwalan perkuliahan di STIKOM Bali dengan menggunakan algoritma genetika ini dilatarbelakangi oleh permasalahan yang muncul pada proses penjadwalan, seperti lamanya proses penjadwalan, penumpukan jadwal mata kuliah di ruangan yang sama, serta banyaknya aturan perkuliahan yang ada di STIKOM Bali. Sehingga diperlukannya sebuah sistem penjadwalan yang efektif dan efisien untuk dapat mengurangi kesalahan dalam proses penjadwalan. Dalam penelitian ini, referensi diperoleh dari jurnal, buku dan artikel laporan yang berhubungan mengenai penelitian ini, sedangkan data yang digunakan, diperoleh langsung dari STIKOM Bali dengan cara observasi dan wawancara. Dalam sistem penjadwalan BAAK akan melakukan pengolahan data master berupa data prodi, ruangan, data paralel kelas, data matakuliah, data matakuliah ditawarkan, inputan kelas seni dan agama dan data sesi.

Sistem penjadwalan perkuliahan di STIKOM Bali dimulai dengan Prodi menentukan matakuliah yang akan ditawarkan pada semester yang akan dilakukan penjadwalan, serta jumlah kelas yang dibuka. Dalam sistem penjadwalan BAAK akan melakukan pengolahan data master berupa data prodi, ruangan, data paralel kelas, data matakuliah, data matakuliah ditawarkan, inputan kelas seni dan agama dan data sesi. BAAK akan menentukan parameter genetika yang akan digunakan dalam proses genetika, seperti jumlah populasi yang akan digunakan. Sistem akan mengecek jadwal perkuliahan satu per satu sesuai dengan aturan-aturan perkuliahan di STIKOM Bali, apabila jadwal perkuliahan belum memenuhi kriteria, maka jadwal perkuliahan akan diproses lagi dengan operator genetika yaitu diproses dengan seleksi, *crossover*, dan mutasi secara berulang hingga jadwal perkuliahan memenuhi aturan aturan yang telah ditetapkan. Dalam proses penjadwalan perkuliahan, yang menjadi tujuan utama adalah mendapatkan jadwal perkuliahan sesuai dengan aturan sehingga tidak terjadi kelas bentrok. Untuk lebih jelasnya, proses penjadwalan perkuliahan di STIKOM Bali digambarkan ke dalam Flowchart dibawah :



Gambar 1. Flowchart Sistem Penjadwalan di STIKOM Bali

**3. HASIL DAN PEMBAHASAN**

Sistem yang akan dibangun merupakan sistem penjadwalan perkuliahan di STIKOM Bali dengan menggunakan algoritma genetika sebagai metodenya, Berikut ini merupakan analisa dan hasil dari sistem Penjadwalan Perkuliahan di STIKOM Bali :

**3.1. Analisa Dan Perancangan**

Analisa dan perancangan sistem yang digunakan pada Sistem Penjadwalan Perkuliahan di Stikom Bali yaitu pemodelan sistem dengan menggunakan DFD (*Data Flow Diagram*), Berikut merupakan analisa dan perancangan dari Sistem Penjadwalan Perkuliahan di Stikom Bali :

**a. DFD Level Konteks (Data Flow Diagram) Perkuliahan di STIKOM Bali**

Dalam DFD (*Data Flow Diagram*) sistem penjadwalan perkuliahan di STIKOM Bali terbagi kedalam tiga proses utama. Proses pertama adalah login, dalam proses login, yang bisa melakukan proses ini adalah user BAAK dan user Prodi. Dimana di dalam sistem, user BAAK akan melakukan maintenance data, sedangkan user Prodi dapat melihat data penjadwalan dan hasil dari penjadwalan.

Proses kedua adalah maintenance data, pada proses ini akan dilakukan proses maintenance data terhadap data prodi, ruangan, data matakuliah, data paralel kelas, data sesi, dan data matakuliah yang akan dibuka.

Proses ketiga adalah proses penjadwalan, dimana dalam proses ini akan ditentukan parameter genetika yang akan digunakan dalam proses penjadwalan, serta melakukan proses penjadwalan perkuliahan. Berikut ini merupakan DFD Level konteks dari Sistem Penjadwalan Perkuliahan Di STIKOM Bali.

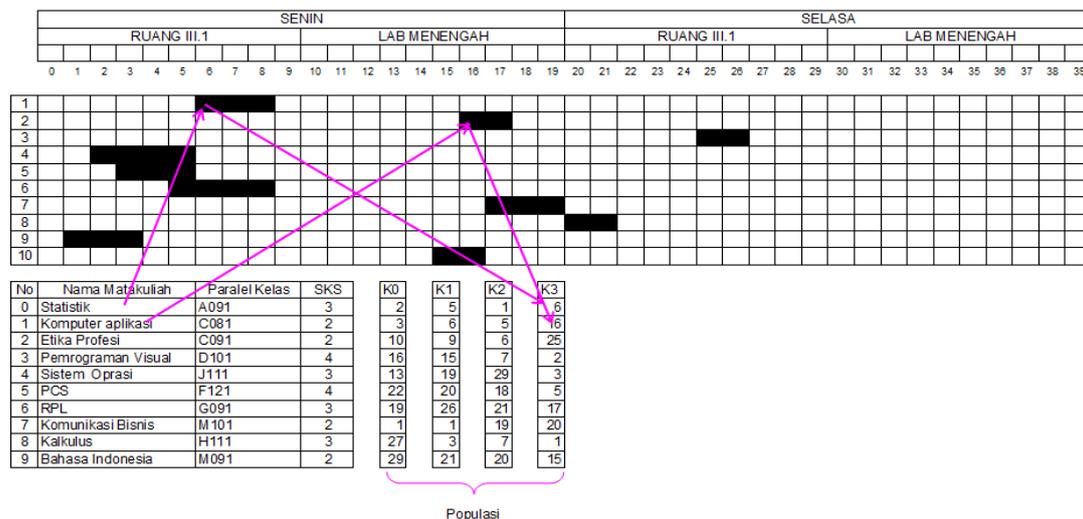


dalam tabel tidak boleh berubah, karena tabel tersebut akan dirubah ke dalam kromosom. Kelas perkuliahan nanti akan dimasukkan secara acak ke dalam slot ruang waktu. Nilai indeks awal dari indeks slot ruang waktu tersebutlah yang akan digunakan sebagai nilai kromosom.

Tabel 2. Kelas Matakuliah

No	Nama matakuliah	Paralel kelas	SKS
1	Statistik	A091	3
2	Komputer Aplikasi	C081	2
3	Etika Profesi	C091	2
4	Pemrograman Visual	D101	4
5	Sistem operasi	J111	3
6	Pemrograman Client Server	F121	4
7	Rekayasa Perangkat Lunak	G091	3
8	Komunikasi Bisnis	M101	2
9	Kalkulus	H111	3
10	Bahasa Indonesia	M091	2

d. Pembentukan Kromosom



Gambar 3. Contoh pembentukan kromosom

Pembentukan kromosom pada contoh *sample* diatas menggunakan 10 *sample* matakuliah sebagai ilustrasi pembentukan kromosom dan besar populasinya adalah 4 yaitu K0, K1, K2, K3. Masing-masing matakuliah memiliki keterangan paralel kelas dan sks. Indeks kromosom diletakkan secara acak oleh sistem ke dalam slot ruang waktu, nilai indeks awal tersebutlah yang akan digunakan ke dalam kromosom.

e. Evaluasi *Fitness*

Masing-masing kelas perkuliahan mempunyai nilai *fitness* lokal, dimana nilai *fitness* lokal maksimum adalah 3. Selanjutnya semua nilai *fitness* lokal dari dalam satu buah kromosom dijumlahkan dan dibagi dengan jumlah nilai *fitness* lokal maksimum, sehingga menghasilkan nilai *fitness* kromosom.

Tabel 3. Contoh Kromosom pertama (K0)

K0	MK	Cek Bentrok Ruang	Cek Batas	Cek Ruang
1	Statistik <b>A081</b>		1	1
2	Komputer aplikasi <b>A091</b>		1	
3	Etika profesi <b>M101</b>		1	
4	Pemrograman visual <b>Lab6</b>	1	1	1
5	Sistem Operasi <b>D081</b>	1	1	
6	PCS <b>Lab6</b>	1	1	
7	RPL <b>L111</b>			
8	Kombis <b>M091</b>		1	1
9	Kalkulus <b>H101</b>		1	1
10	Bahasa Indonesia <b>H102</b>			1
<b>Total fitness kromosom 0 :</b>				16

$$f = \frac{f_{\text{lokal}}}{f_{\text{lokal maksimum}}}$$

Keterangan:

$f$  : Nilai *fitness* kromosom

$f_{\text{lokal}}$  : Total nilai *fitness* lokal untuk semua kelas perkuliahan

$f_{\text{lokal maksimum}}$  : Total nilai *fitness* lokal maksimum

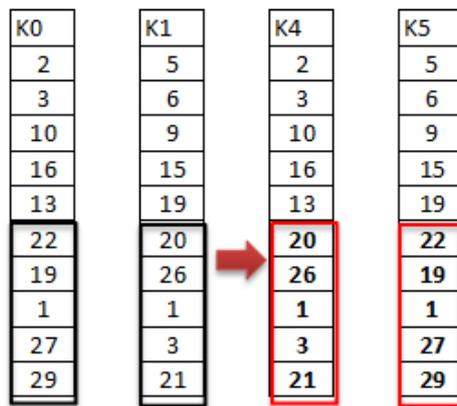
Misalkan pada contoh kromosom diatas dihasilkan jumlah total nilai *fitness* lokal untuk semua kelas perkuliahan  $K0 = 16$ , total nilai *fitness* lokal maksimum =  $3 \times 10 = 30$ , maka nilai *fitness* kromosom pertama (K0) adalah :

$$f_0 = \frac{16}{30} = 0,53$$

#### f. Crossover

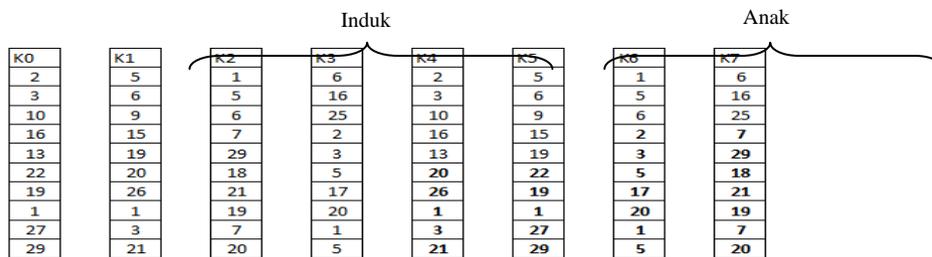
Langkah-langkah dari operasi *crossover* adalah :

- Tentukan pasangan kromosom yang akan dilakukan operasi *crossover*, pada *sample* yang akan digunakan, pasangan *sample* dari populasi diatas adalah K0-K1, K2-K3.
- Bangkitkan angka acak dari 0-1 untuk pasangan K0-K1 dan tentukan *crossover rate* dari 0-1. *Crossover rate* (rasio *crossover*) digunakan untuk menentukan probabilitas atau peluang suatu pasangan kromosom melakukan *crossover*. Misalkan *crossover rate* = 0,8 dan angka acak untuk pasangan kromosom (K0-K1) = 0,7.
- Jika angka acak lebih kecil dari *crossover rate*, maka pasangan kromosom (K0-K1) akan melakukan *crossover*.
- Tentukan titik *crossover* secara acak dengan rentang 0 s/d panjang kromosom, misalkan = 5. Panjang gen yang akan di *crossover* adalah panjang kromosom dikurangi dengan titik *crossover* Panjang gen =  $10 - 5 = 5$ .
- Lakukan *crossover*.



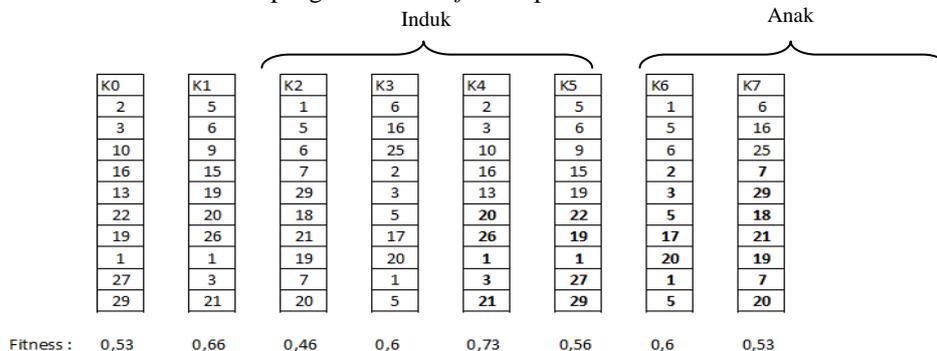
Gambar 4. Proses *crossover*

jumlah populasi baru = populasi awal (induk) + populasi anak seperti pada gambar dibawah :



Gambar 5. populasi setelah di *crossover*

f. Lakukan kembali pengecekan nilai *fitness* pada kromosom anak.



Gambar 6. *Fitness* induk dan anak

g. **Mutasi**

Proses mutasi akan dilakukan pada setiap kromosom induk. Adapun langkah-langkah dari proses mutasi adalah :

- a. Tentukan mutation rate dari 0-1 dan tentukan angka acak dari 0-1 pada masing-masing kromosom. Misal angka acak 0,09, mutation rate = 0,2. Nilai mutasi harus kecil, apabila nilai mutasi besar maka kromosom yang bagus akan berpeluang melakukan mutasi juga dan menjadi buruk.
- b. Cek angka acak  $\leq$  mutation rate, apabila kondisinya true maka tentukan *i*, dimana *i* merupakan angka acak indeks yang akan dilakukan mutasi dan *i* bernilai 0 sampai dengan panjang kromosom. Misal *i* = 3, maka nilai indeks (gen) ke[*i*] pada kromosom akan diubah dengan angka acak.

K0		K0
2		2
3		3
10		10
16	→	25
13		13
22		22
19		19
1		1
27		27
29		29

Gambar 7. Mutasi

c. Lakukan Pengecekan Nilai *Fitness*

#### h. Seleksi

Setelah dilakukan *crossover* dan mutasi ukuran populasi menjadi bertambah besar karena terdapat kromosom baru hasil dari proses *crossover* dan mutasi. Supaya ukuran populasi menjadi ukuran populasi awal, maka diperlukan proses seleksi untuk memilih kromosom terbaik dari populasi yang ada. Berikut adalah langkah-langkah proses seleksi :

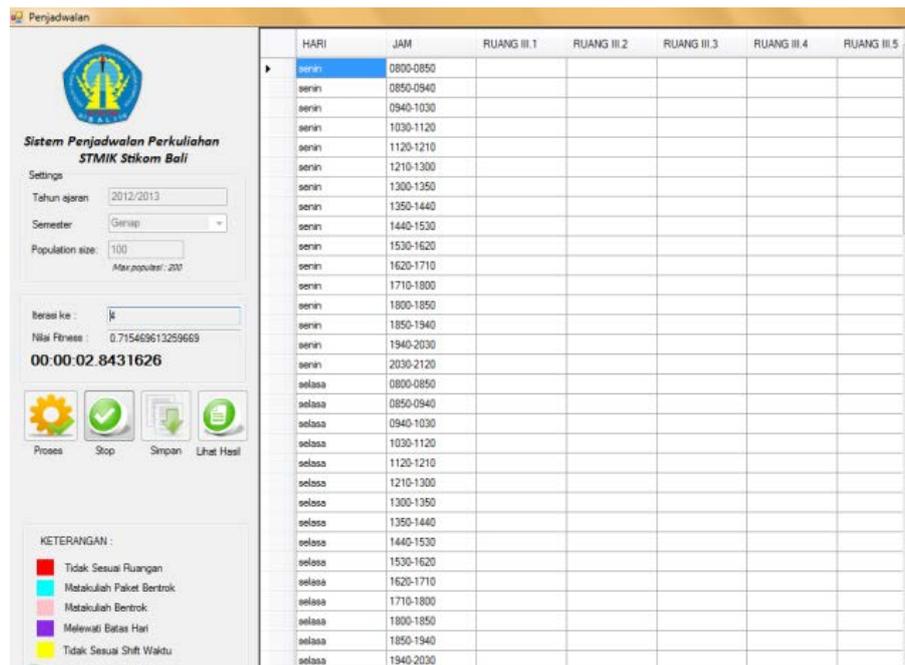
- Lakukan pengurutan kromosom secara ascending berdasarkan nilai *fitness*nya.
- Delete semua kromosom di luar jumlah populasi sebenarnya, pada *sample* yang digunakan, populasi awal adalah 4 sedangkan populasi setelah di *crossover* dan mutasi adalah 9, maka pilih 4 kromosom dengan nilai *fitness* terbesar dan sisanya dihapus.
- Setelah mendapatkan 4 kromosom terbesar, acak posisi kromosom dalam populasi untuk menghindari terjadinya *crossover* antar kromosom-kromosom yang bagus.
- Tentukan kromosom dengan nilai *fitness* terbaik dan lakukan evaluasi nilai *fitness*.
- Jika nilai *fitness* kromosom terbaik sama dengan 1, maka proses genetika berhenti. Jika tidak, maka akan dilakukan operasi genetika lagi (*crossover*, mutasi, dan seleksi) untuk iterasi selanjutnya.

### 3.2. Hasil

Sistem Penjadwalan Perkuliahan di STIKOM Bali yang dikembangkan menggunakan bahasa pemrograman C# 2008 dengan menggunakan IDE Microsoft Visual Studio 2008 dan database yang digunakan adalah Microsoft SQL Management Studio 2008 R2. Hasil dari sistem yang dikembangkan berfungsi untuk mempermudah petugas akademik untuk melakukan proses penjadwalan, mempersingkat waktu penjadwalan, sehingga proses penjadwalan berjalan dengan efektif dan dapat mengurangi kesalahan pada saat proses penjadwalan.

#### a. Form Penjadwalan

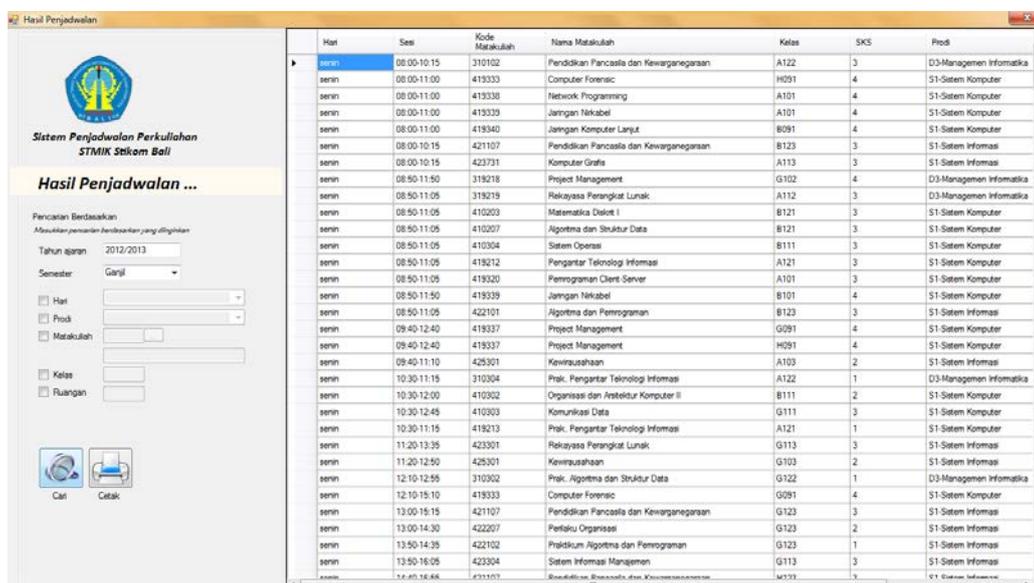
Berikut Merupakan tampilan dari form Sistem Penjadwalan Perkuliahan di STIKOM Bali, dalam form ini akan digunakan untuk melakukan proses penjadwalan dengan menggunakan metode genetika, disini user sebagai BAAK akan melakukan proses penjadwalan sesuai matakuliah yang ditawarkan.



Gambar 8. Tampilan Form Penjadwalan

**b. Tampilan Form Hasil Penjadwalan**

Setelah proses penjadwalan selesai, data penjadwalan dapat disimpan ke dalam database sistem penjadwalan. Berikut merupakan hasil penjadwalan perkuliahan setelah disimpan :



Gambar 9. Tampilan Form Hasil Penjadwalan

**c. Hasil Pengujian Parameter Genetika**

Hasil pengujian parameter genetika ini bertujuan untuk menganalisa kinerja algoritma genetika pada proses penyusunan penjadwalan perkuliahan di STIKOM Bali. Pengujian dilakukan dengan cara mengubah-ubah parameter pada proses penjadwalan. Pada Tabel dibawah ini menunjukkan hasil dari beberapa *sample* uji yang diujikan. *Sample* uji dilakukan untuk penjadwalan perkuliahan tahun 2012/2013 pada semester genap dengan rentang waktu 15 menit. Pada pengujian pertama, nilai dari Probabilitas *Crossover* (*Crossover Rate*) yang digunakan adalah 0,8 dan nilai dari Probabilitas Mutasi (*Mutation Rate*)

adalah 0,05 dengan menggunakan 3 nilai populasi yang masing-masing menggunakan jumlah kelas yang berbeda.

**Tabel 4.** Hasil Pengujian Pertama

Populasi	Probabilitas <i>Crossover</i>	Probabilitas Mutasi	<i>Fitness</i>	Jumlah Kelas Paralel			Iterasi
				Praktikum	Kelas	Total	
30	0.8	0.05	0.97046	33	204	237	6467
30	0.8	0.05	0.95143	47	266	313	5596
30	0.8	0.05	0.92162	60	333	393	4356
70	0.8	0.05	0.97299	33	204	237	3326
70	0.8	0.05	0.95718	47	266	313	2527
70	0.8	0.05	0.93025	60	333	393	2056
150	0.8	0.05	0.98227	33	204	237	1837
150	0.8	0.05	0.95399	47	266	313	1184
150	0.8	0.05	0.90127	60	333	393	957

Hasil pengujian diatas dapat dilihat bahwa pengujian untuk total kelas pertama yang sama pada 3 populasi yang berbeda yaitu populasi 30, 70 dan 150, nilai *fitness* terbesar adalah 0.98227 dalam iterasi 1837. Pada pengujian dengan data kedua nilai *fitness* terbesar ditunjukkan pada populasi 70 dengan nilai *fitness* 0.95718, dan pada pengujian data ketiga nilai *fitness* terbesar ditunjukkan pada populasi 70 dengan nilai *fitness* 0.95718.

Hasil pengujian kedua menggunakan Probabilitas *Crossover* (*Crossover Rate*) yang digunakan adalah 0,75 dan nilai dari Probabilitas Mutasi (*Mutation Rate*) adalah 0,25 dengan menggunakan 3 nilai populasi yang masing-masing menggunakan jumlah kelas yang berbeda.

**Tabel 5.** Hasil Pengujian Kedua

Populasi	Probabilitas <i>Crossover</i>	Probabilitas Mutasi	<i>Fitness</i>	Jumlah Kelas Paralel			Iterasi
				Praktikum	Kelas	Total	
30	0.75	0.25	0.99493	33	204	237	7576
30	0.75	0.25	0.98977	47	266	313	5181
30	0.75	0.25	0.9715	60	333	393	3865
70	0.75	0.25	0.99578	33	204	237	3339
70	0.75	0.25	0.98849	47	266	313	2246
70	0.75	0.25	0.96641	60	333	393	1735
150	0.75	0.25	0.99409	33	204	237	1605
150	0.75	0.25	0.98594	47	266	313	1064
150	0.75	0.25	0.93530	60	333	393	546

Hasil pengujian kedua dapat dilihat bahwa nilai *fitness* terbaik pada jumlah data pertama yaitu pada 70 populasi yaitu 0.99578, pada data kedua ditunjukkan pada 30 populasi. Sedangkan pada data ketiga ditunjukkan pada 30 populasi.

Hasil pengujian ketiga menggunakan Probabilitas *Crossover* (*Crossover Rate*) yang digunakan adalah 0,85 dan nilai dari Probabilitas Mutasi (*Mutation Rate*) adalah 0,3 dengan menggunakan 3 nilai populasi yang masing-masing menggunakan jumlah kelas yang berbeda.

Tabel 6. Pengujian Ketiga

Populasi	Probabilitas <i>Crossover</i>	Probabilitas Mutasi	<i>Fitness</i>	Jumlah Kelas Paralel			Iterasi
				Praktikum	Kelas	Total	
30	0.85	0.3	0.99578	33	204	237	6618
30	0.85	0.3	0.98466	47	266	313	4536
30	0.85	0.3	0.96844	60	333	393	3413
70	0.85	0.3	0.99662	33	204	237	2847
70	0.85	0.3	0.98594	47	266	313	1978
70	0.85	0.3	0.97099	60	333	393	1584
150	0.85	0.3	0.99409	33	204	237	1316
150	0.85	0.3	0.98146	47	266	313	931
150	0.85	0.3	0.95776	60	333	393	720

Hasil pengujian ketiga menunjukkan nilai *fitness* pada pengujian data pertama menunjukkan nilai *fitness* pada 70 populasi merupakan populasi terbaik yaitu 0.99662. pengujian data kedua nilai *fitness* terbaik ditunjukkan pada 70 populasi dimana nilai *fitness*nya adalah 0.98594. dan pada data ketiga nilai *fitness* terbaik ditunjukkan kembali pada 70 populasi.

Dari ketiga pengujian diatas dapat dilihat bahwa dengan mengubah parameter masukan seperti jumlah kelas dan parameter algoritma genetika seperti jumlah populasi, probabilitas *crossover* (*crossover* rate), probabilitas mutasi (mutation rate) menghasilkan data yang bervariasi dalam rentang waktu 15 menit, baik itu nilai *fitness* maupun iterasinya. Jadi ukuran parameter yang lebih besar, belum tentu menghasilkan jadwal terbaik.

Ukuran populasi mempengaruhi kinerja dan keefektifan algoritma genetika. Populasi yang lebih besar dibutuhkan untuk mempresentasikan keseluruhan ruang persoalan, namun selakin besar ukuran populasi maka kecepatan iterasi program penjadwalan akan semakin lambat, terlihat pada tabel diatas, semakin besar ukuran populasi, semakin kecil iterasi yang ditunjukkan.

#### 4. Kesimpulan

Dari penulisan skripsi yang berjudul “Implementasi Algoritma Genetika pada Penjadwalan Perkuliahan di STIKOM Bali” dapat ditarik kesimpulan sebagai berikut :

1. Sistem Penjadwalan Perkuliahan di STIKOM Bali dirancang dengan menggunakan DFD (Data Flow Diagram) dan ERD (Entity Relationship Diagram) untuk perancangan databasenya.
2. Sistem Penjadwalan Perkuliahan di STIKOM Bali dirancang untuk mempermudah petugas akademik dalam proses penjadwalan perkuliahan, sehingga proses penjadwalan dapat berjalan dengan efektif dan efisien.
3. Algoritma genetika dalam Sistem Penjadwalan Perkuliahan di STIKOM Bali dirancang dengan pembentukan kromosom berupa urutan kelas perkuliahan yang ditempatkan pada slot ruang waktu.
4. Dengan mengubah parameter masukan seperti jumlah kelas dan parameter algoritma genetika seperti jumlah populasi, probabilitas *crossover* (*crossover* rate), probabilitas mutasi (mutation rate) menghasilkan data yang bervariasi.
5. Ukuran parameter yang lebih besar, belum tentu menghasilkan jadwal terbaik.
6. Ukuran populasi mempengaruhi kinerja dan keefektifan algoritma genetika. Populasi yang lebih besar dibutuhkan untuk mempresentasikan keseluruhan ruang persoalan.
7. Semakin besar ukuran populasi maka kecepatan iterasi program penjadwalan akan semakin lambat.

#### DAFTAR PUSTAKA

- [1] Achmad Basuki, 2003. Algoritma Genetika. Politeknik Elektronika Negeri Surabaya.
- [2] Agus Widyadana, Andree Pamungkas, 2002. Perbandingan Kinerja Algoritma Genetika Dan Simulated Annealing Untuk Masalah Multiple Objective Pada Penjadwalan Flowshop. Universitas Kristen Petra.
- [3] Ahmad Riyad. Algoritma Genetika. Politeknik Batam, Batam
- [4] Anita Desiani & Muhammad Arhami, 2006. Konsep Kecerdasan Buatan, Andi Yogyakarta.

- [5] Annilia Septema, Perancangan Sistem Informasi Akutansi Atas Siklus Pengeluaran Pada Hotel Mutiara Di Boyolali. Universitas Gunadarma
- [6] Fadlisyah, Arnawan, Faizal, 2009. Algoritma Genetika, Graha Ilmu Yogyakarta.
- [7] Nico Saputro dan Erdo Dirgagutama, 2004. Penerapan Algoritma Genetic Pada Permainan Catur, Universitas Katolik Parahyangan, Bandung
- [8] Ragnu Ramakrishnan Johannes Gehrke, 2004. Sistem Managemen Database, Andi Yogyakarta
- [9] Samani, 2012. Rancang Bangun Sistem Penjadwalan Perkuliahan Dan Ujian Akhir Semester Dengan Pendekatan Algoritma Genetika. Universitas Diponegoro, Semarang
- [10] Sarwadi dan Anjar KSW, 2004. Algoritma Genetika Untuk Penyelesaian Masalah Vehicle Routing, Universitas Diponegoro Semarang.
- [11] Suyanto, 2005. Algoritma Genetika Dalam Matlab, Andi Yogyakarta.