

**ANALISIS TEKNIK CROSSOVER PADA PENYELESAIAN  
PERMASALAHAN PENJADWALAN PRAKTIKUM DENGAN  
ALGORITMA GENETIKA**

**SKRIPSI**

oleh:  
**NURAINI DWI UTAMI**  
**0410960043-96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA**

**2009**

**i**



**ANALISIS TEKNIK *CROSSOVER* PADA PENYELESAIAN  
PERMASALAHAN PENJADWALAN PRAKTIKUM DENGAN  
ALGORITMA GENETIKA**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh  
gelar Sarjana dalam bidang Ilmu Komputer

oleh:  
**NURAINI DWI UTAMI**  
**0410960043-96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA**

**2009**





**LEMBAR PENGESAHAN SKRIPSI**

**ANALISIS TEKNIK *CROSSOVER* PADA PENYELESAIAN  
PERMASALAHAN PENJADWALAN PRAKTIKUM DENGAN  
ALGORITMA GENETIKA**

Oleh:

**NURAINI DWI UTAMI**  
**0310963037-96**

Setelah dipertahankan di depan Majelis Penguji  
Pada tanggal 22 Januari 2009  
dan dinyatakan memenuhi syarat untuk memperoleh  
gelar Sarjana dalam bidang Ilmu Komputer

Pembimbing I

Wayan Firdaus M., SSi., MT  
NIP. 132 158 724

Pembimbing II

Candra Dewi, S.Kom, M.Kom  
NIP. 132 304 116

Mengetahui,  
Ketua Jurusan Matematika  
Fakultas MIPA Universitas  
Brawijaya

Dr. Agus Suryanto, M.Sc  
NIP. 132 126 049





**LEMBAR PERNYATAAN**

Saya yang bertanda tangan di bawah ini :

**Nama** : Nuraini Dwi Utami  
**NIM** : 0410960043-96  
**Jurusan** : Matematika  
**Penulis Tugas Akhir berjudul** : Analisis Teknik *Crossover* Pada Penyelesaian Permasalahan Penjadwalan Praktikum Dengan Algoritma Genetika

Dengan ini menyatakan bahwa :

1. Isi dari Tugas Akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
  2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.
- Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 22 Januari 2009  
Yang menyatakan,

(Nuraini Dwi Utami)  
NIM. 0410960043





## ANALISIS TEKNIK *CROSSOVER* PADA PENYELESAIAN PERMASALAHAN PENJADWALAN PRAKTIKUM DENGAN ALGORITMA GENETIKA

### ABSTRAK

*Crossover* mempunyai peran penting dalam proses algoritma genetika, yaitu menghasilkan individu baru dengan nilai *fitness* yang cenderung lebih baik. Pada penelitian ini, 3 teknik *crossover* yang berbeda yaitu *one point*, *two point* dan *uniform crossover* diimplementasikan pada permasalahan penjadwalan praktikum untuk mengetahui *performance* dari masing-masing teknik *crossover*. Parameter yang digunakan untuk mengukur *performance* teknik *crossover* yaitu nilai *cost* yang dihasilkan ketika proses genetika mencapai konvergensi.

Pengujian dilakukan dengan memberikan nilai parameter genetika yang berbeda yaitu probabilitas *crossover* dan ukuran populasi, untuk melihat pengaruh kedua parameter tersebut bagi kinerja operator *crossover*.

Hasil pengujian menunjukkan bahwa *performance one point crossover* paling baik diantara ketiga teknik *crossover*. Hal ini dikarenakan *one point crossover* menukar segmen individu pada level yang rendah serta kombinasi parameter genetika yang digunakan sesuai untuk teknik ini. *Uniform crossover* mempunyai *performance* terendah karena anak yang dihasilkan cenderung berbeda jauh dari induk sehingga sulit mencari solusi yang optimal.





## ANALYSIS CROSSOVER TECHNIQUE IN SOLVING LAB PRACTISE TIMETABLING PROBLEM USING GENETIC ALGORITHM

### ABSTRACT

Crossovers have an important role in the process of genetic algorithm, ie, to individuals with a new fitness values are likely to better. In the study, three different techniques crossover that is one point, two point and uniform crossover implemented in lab practise scheduling problems for the performance of each technique crossover. The parameter used to measure the performance crossover technique that is cost value generated when the process of genetic achieve convergence.

Tests conducted with the value parameter that is a different genetic crossover probability and population size, to see the influence of both these parameters for performance crossover operator.

Test results showed that one-point crossover have good performance, the most among the three techniques crossover. One point crossover segment shift on the individual level and a low parameter genetic combinations that are used according to this technique. Uniform crossover has the lowest performance because the results tend Anakan produced far different from the parent so difficult to find the optimal solution.

UNIVERSITAS BRAWIJAYA



x





## KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer.

Penelitian ini ditujukan untuk menganalisa teknik *crossover* pada algoritma genetika dengan studi kasus permasalahan penjadwalan praktikum.

Banyak pihak yang berperan atas terselesainya penelitian dan penulisan Skripsi ini. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Wayan Firdaus Mahmudy, SSi., MT selaku pembimbing I serta selaku Ketua Program Studi Ilmu Komputer. Terima kasih atas semua saran, bantuan, waktu dan bimbingannya.
2. Yusi Tyroni M., S.Kom dan Candra Dewi, S.Kom. M.Sc, selaku pembimbing pendamping. Terima kasih atas saran, bantuan, waktu dan bimbingannya.
3. Kasyful Amron, ST selaku Penasihat Akademik.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
5. Segenap staf dan karyawan di Jurusan Matematika FMIPA.
6. Kedua orang tua dan segenap keluarga yang tak pernah berhenti memberikan doa dan dukungannya.
7. Teman-teman ilkomers dan khususnya genk herry yang selalu memberi bantuan dan dukungannya.
8. M.Rifai, terima kasih untuk doa, senyuman, dukungan dan bantuan selama pengerjaan skripsi ini.
9. Dan semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak dapat Penulis sebutkan satu per satu.

Penulis sadari bahwa masih banyak kekurangan dalam laporan ini, oleh karena itu Penulis sangat menghargai saran dan kritik yang

sifatnya membangun demi perbaikan penulisan dan mutu isi tugas akhir ini untuk kelanjutan penelitian serupa di masa mendatang.

Sebuah harapan tulus semoga skripsi ini dapat bermanfaat sebesar-besarnya.

Malang, Januari 2009

Penulis



**DAFTAR ISI**

Halaman Judul .....	i
Halaman Pengesahan .....	iii
Halaman Pernyataan .....	v
Abstrak.....	vii
Abstract.....	ix
Kata Pengantar.....	xi
Daftar Isi .....	xiii
Daftar Gambar .....	xv
Daftar Tabel.....	xvii
Daftar Lampiran.....	xix
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 ..Latar Belakang.....	1
1.1 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat .....	3
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 Penjadwalan .....	5
2.1.1 Penjadwalan Praktikum di Laboratorium.....	6
2.2 Algoritma Genetika.....	7
2.2.1 Gambaran Umum.....	7
2.3 Representasi kromosom.....	8
2.4 Inisialisasi .....	9
2.5 Nilai Fitness .....	10
2.6 Seleksi .....	10
2.7 Operator Genetika.....	13
2.7.1 Operator Reproduksi.....	13
2.7.2 Operator Pindah Silang (Crossover) .....	13
2.7.3 Operator Mutasi .....	17
2.8 Analisis Crossover dan Penjadwalan dengan Algoritma Genetika .....	18





<b>BAB III METODOLOGI DAN PERANCANGAN .....</b>	<b>21</b>
3.1 Deskripsi Umum Sistem.....	21
3.3 Perancangan Proses Penjadwalan Dengan Algoritma Genetik.....	21
3.3.1 Model Genetika.....	21
3.3.2 Relasi Tabel Basis Data.....	23
3.3.3 Nilai <i>Fitness</i> .....	26
3.3.4 Seleksi.....	28
3.3.5 <i>Crossover</i> .....	28
3.3.6 Mutasi.....	29
3.3.7 Contoh Manual.....	30
3.4 Flowchart Algoritma Genetika.....	35
3.4.1 Pseudocode Proses <i>Crossover</i> .....	38
3.5 Perancangan Proses Perbandingan Teknik <i>Crossover</i> .....	39
3.5.1 Flowchart Proses Perbandingan Teknik <i>Crossover</i> .....	40
3.6 Perancangan Uji Coba.....	41
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN .....</b>	<b>43</b>
4.1 Implementasi Program.....	43
4.1.1 Inisialisasi Kromosom.....	43
4.1.2 Seleksi.....	43
4.1.3 <i>Crossover</i> (pindah silang).....	44
4.1.4 Mutasi.....	48
4.2 Implementasi Antarmuka.....	50
4.3 Implementasi Uji Coba.....	52
4.4 Hasil Uji Coba.....	53
4.4.1 Hasil Uji Coba Variasi Probabilitas <i>Crossover</i> .....	53
4.4.2 Hasil Uji Coba Variasi Ukuran Populasi.....	55
4.5 Analisis Hasil.....	58
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>63</b>
5.1 Kesimpulan.....	63
5.2 Saran.....	63
<b>DAFTAR PUSTAKA.....</b>	<b>65</b>
<b>LAMPIRAN.....</b>	<b>69</b>

DAFTAR GAMBAR

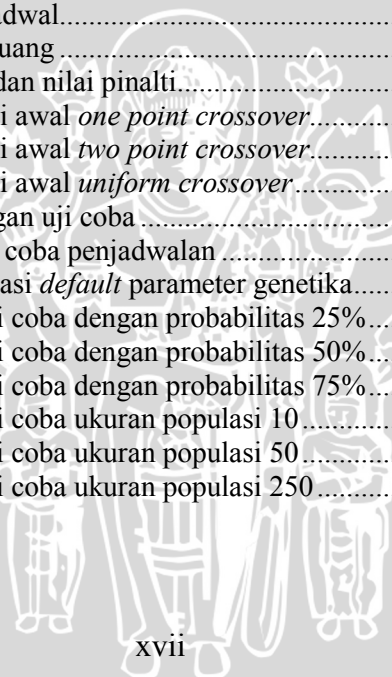
	Halaman
Gambar 2.1	Contoh Kromosom Biner..... 8
Gambar 2.2	Contoh Kromosom Float..... 8
Gambar 2.3	Contoh kromosom nilai..... 9
Gambar 2.4	Contoh kromosom permutasi..... 9
Gambar 2.5	Contoh probabilitas kromosom pada roda <i>roulette</i> 11
Gambar 2.6	Proses <i>one point crossover</i> ..... 14
Gambar 2.7	Proses <i>two point crossover</i> ..... 14
Gambar 2.8	Proses <i>n point crossover</i> ..... 15
Gambar 2.9	Proses <i>uniform crossover</i> ..... 15
Gambar 2.10	Contoh kromosom..... 19
Gambar 3.1	Model Genetika..... 22
Gambar 3.2	Relasi Antar Tabel..... 26
Gambar 3.4	Penyelesaian Dengan Algoritma Genetika..... 36
Gambar 3.5	<i>Pseudocode one point crossover</i> ..... 38
Gambar 3.6	<i>Pseudocode Two point crossover</i> ..... 38
Gambar 3.7	<i>Pseudocode Uniform crossover</i> ..... 39
Gambar 3.8	Flowchart Perbandingan Teknik <i>Crossover</i> ..... 40
Gambar 4.1	Antarmuka Proses Genetika..... 50
Gambar 4.2	Antarmuka nilai <i>cost</i> dan <i>fitness</i> ..... 51
Gambar 4.3	Antarmuka Jadwal Praktikum..... 51
Gambar 4.4	Grafik perbandingan rata-rata <i>cost</i> dengan probabilitas <i>crossover</i> berbeda..... 55
Gambar 4.5	Grafik perbandingan rata-rata <i>cost</i> dengan ukuran populasi berbeda..... 57
Gambar 4.6	Grafik hasil uji coba variasi probabilitas <i>crossover</i> 61
Gambar 4.7	Grafik terbaik uji variasi ukuran populasi..... 62





DAFTAR TABEL

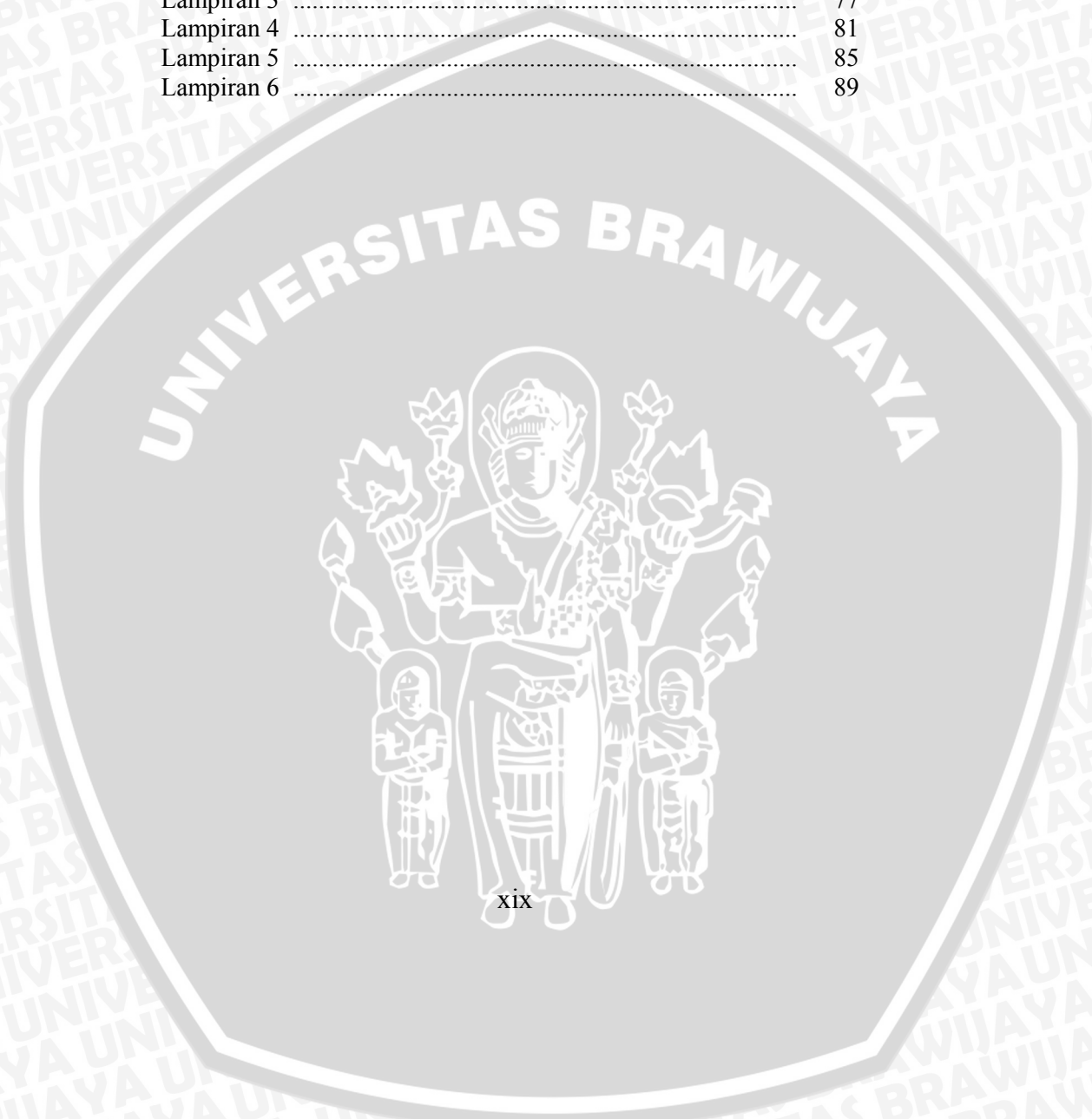
	Halaman
Tabel 2.1 Kromosom dan Nilai <i>fitnessnya</i> .....	11
Tabel 2.2 Keadaan kromosom sebelum dan sesudah Di ranking .....	12
Tabel 2.3 Hasil perbandingan teknik <i>crossover</i> .....	18
Tabel 3.1 Spesifikasi permasalahan .....	21
Tabel 3.2 Inisialisasi kromosom.....	22
Tabel 3.3 Tabel Mata Kuliah.....	23
Tabel 3.4 Tabel Program Studi .....	24
Tabel 3.5 Tabel Hari.....	24
Tabel 3.6 Tabel Waktu.....	24
Tabel 3.7 Tabel Asisten.....	24
Tabel 3.8 Kelas Mata Kuliah.....	25
Tabel 3.9 Tabel Asisten Kelas.....	25
Tabel 3.10 Tabel Jadwal.....	25
Tabel 3.11 Tabel Ruang .....	26
Tabel 3.12 Aturan dan nilai pinalti.....	27
Tabel 3.13 Generasi awal <i>one point crossover</i> .....	30
Tabel 3.14 Generasi awal <i>two point crossover</i> .....	32
Tabel 3.15 Generasi awal <i>uniform crossover</i> .....	34
Tabel 3.16 Rancangan uji coba .....	41
Tabel 4.1 Data uji coba penjadwalan .....	52
Tabel 4.2 Kombinasi <i>default</i> parameter genetika.....	52
Tabel 4.3 Hasil uji coba dengan probabilitas 25%.....	53
Tabel 4.4 Hasil uji coba dengan probabilitas 50%.....	54
Tabel 4.5 Hasil uji coba dengan probabilitas 75%.....	54
Tabel 4.6 Hasil uji coba ukuran populasi 10.....	56
Tabel 4.7 Hasil uji coba ukuran populasi 50.....	56
Tabel 4.8 Hasil uji coba ukuran populasi 250.....	57





## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 .....	69
Lampiran 2 .....	73
Lampiran 3 .....	77
Lampiran 4 .....	81
Lampiran 5 .....	85
Lampiran 6 .....	89







xx

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Algoritma genetika adalah suatu algoritma pencarian dan optimasi yang bersifat stokastik atau bekerja secara random, berbasis pada proses mekanika alamiah di mana sifat-sifat suatu individu sangat bergantung pada gen-gen dan susunannya. Algoritma genetika sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sulit diselesaikan dengan menggunakan metode konvensional. Contoh permasalahan yang dapat diselesaikan dengan algoritma genetika antara lain, TSP, desain jaringan, optimasi fungsi, penjadwalan serta permainan tertentu (Nurjaya, 2002).

Algoritma genetika membentuk suatu populasi acak dari individu atau kromosom yang merepresentasikan kemungkinan solusi yang mungkin. Setiap individu akan dievaluasi berdasarkan nilai kecocokan (*fitness*) yang menentukan kualitas dari individu. Proses evaluasi ini dijalankan secara iteratif sampai didapat solusi yang optimal atau individu yang terbaik.

Individu baru dibentuk menggunakan operator genetika yaitu reproduksi, persilangan (*crossover*) dan mutasi. Reproduksi merupakan proses menghasilkan individu baru dari *parent* yang dipilih secara acak dari populasi awal berdasarkan nilai *fitness*nya dimana individu dengan nilai *fitness* yang lebih baik mempunyai kemungkinan yang lebih tinggi dalam bereproduksi. Individu yang terpilih akan dipilih untuk operasi selanjutnya yaitu *crossover* dan mutasi. *Crossover* bertujuan menghasilkan individu baru yang berbeda dari induknya dan diharapkan kualitasnya lebih baik dari induknya. Sedangkan mutasi beroperasi dengan merubah secara acak satu atau lebih komponen dari individu yang dipilih untuk memperbaiki dan mencegah stagnasi yang mungkin terjadi selama proses pencarian solusi (Baluja dan Caruana, 1995).

*Crossover* mempunyai peran penting dalam proses genetika, yaitu menghasilkan individu baru dengan pertukaran segmen antara kromosom induk. Pertukaran segmen ini akan menghasilkan pertukaran informasi antara dua individu yang akan menghasilkan individu baru (anak) dengan nilai *fitness* yang cenderung lebih baik.

Analisis teknik *crossover* dilakukan dengan menerapkan teknik-teknik *crossover* pada satu permasalahan kemudian dibandingkan hasil dari masing-masing teknik *crossover*. Teknik dikatakan lebih baik jika individu yang dihasilkan mempunyai nilai *cost* atau *fitness* yang lebih baik serta kemampuan menghasilkan populasi yang cenderung divergen. Analisis *crossover* pada penelitian sebelumnya (De Jong dan Spears, 90 dan Baluja dan Caruana, 1995) dilakukan pada permasalahan *n-peaks problem*. Sedangkan untuk penelitian ini akan dilakukan pada permasalahan penjadwalan, yaitu penjadwalan praktikum.

Penjadwalan adalah pengalokasian sejumlah subjek kedalam periode tertentu, sedemikian hingga dapat memenuhi batasan-batasan yang diberikan dengan memenuhi sebanyak mungkin batasan tersebut.

Penjadwalan praktikum, seperti penjadwalan pada umumnya merupakan pengalokasian sumber daya pada periode tertentu dengan memperhatikan atau memenuhi batasan yang ditentukan.

Permasalahan atau batasan yang harus dipenuhi pada penyelesaian penjadwalan praktikum ini antara lain, keterbatasan ruang, mahasiswa dari beberapa program studi serta lintas jurusan, dan waktu yang harus menyesuaikan jadwal kuliah. Dengan berbagai batasan tersebut, penjadwalan praktikum ini akan diselesaikan dengan algoritma genetika.

## 1.2 Rumusan masalah

Rumusan masalah dari penelitian ini adalah bagaimana melakukan analisis kinerja teknik *crossover* dalam menyelesaikan masalah penjadwalan praktikum pada laboratorium Matematika, Fakultas MIPA, Universitas Brawijaya.

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Penjadwalan yang dijadikan studi kasus yaitu penjadwalan praktikum pada Laboratorium Matematika, Fakultas MIPA Universitas Brawijaya.
2. Teknik yang dibandingkan hanya *one point crossover*, *two point crossover* dan *uniform crossover*.



3. Fokus penelitian yaitu mempelajari pengaruh ketiga teknik *crossover* pada hasil proses genetika.

#### 1.4 Tujuan Penelitian

1. Menerapkan algoritma genetika dalam menyelesaikan masalah penjadwalan pada laboratorium Matematika, Fakultas MIPA Universitas Brawijaya.
2. Melakukan analisis teknik *crossover* untuk mengetahui teknik *crossover* dengan *performance* terbaik dibandingkan teknik lainnya.

#### 1.5 Manfaat

Manfaat dari penulisan tugas akhir ini adalah dihasilkannya gambaran mengenai karakteristik masing-masing teknik *crossover*. Dengan diketahuinya karakteristik dari teknik *crossover*, proses genetika selanjutnya dapat menggunakan kombinasi parameter genetika yang sesuai untuk mendapatkan hasil yang optimal.





## BAB II TINJAUAN PUSTAKA

### 2.1 Penjadwalan

Pengertian jadwal menurut Kamus Besar Bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal (Mawaddah, 2006).

Penjadwalan adalah permasalahan mengalokasikan kumpulan even ( kelas, praktikum atau dosen) pada time slot yang spesifik, dimana tidak ada orang atau sumber daya yang ditempatkan pada dua lokasi pada satu waktu secara bersamaan serta ketersediaan ruang yang sesuai untuk even yang dijadwalkan. Permasalahan penjadwalan sendiri mempunyai beberapa jenis yaitu, penjadwalan mata pelajaran, penjadwalan kuliah, penjadwalan praktikum, penjadwalan staff, dll.

Penjadwalan adalah pengalokasian sejumlah subjek kedalam periode tertentu, sedemikian hingga dapat memenuhi batasan-batasan yang diberikan dengan memenuhi sebanyak mungkin batasan tersebut. Permasalahan penjadwalan mempunyai beberapa jenis, seperti penjadwalan pertandingan olahraga, penjadwalan karyawan, penjadwalan perkuliahan, penjadwalan praktikum, dll. Seperti halnya permasalahan lain yang termasuk permasalahan *constraint satisfaction problems*, dimana tujuan utamanya memenuhi semua batasan permasalahan daripada optimasi jumlah dari objektifnya. Permasalahan penjadwalan juga dikenal sebagai permasalahan *NP-Complete* karena sulit dipecahkan, dengan banyaknya batasan yang harus diselesaikan serta luasnya ruang penyelesaian yang dapat dieksplorasi jika ukuran permasalahan meningkat (Kazarlis, 2005).

Banyak ide berbeda sudah diterapkan dalam menyelesaikan masalah penjadwalan seperti *coloring graph*, *tabu search*, algoritma semut, serta algoritma genetik (Irene, dkk., 2007).



### 2.1.1 Penjadwalan Praktikum di Laboratorium

Penjadwalan praktikum hampir sama dengan penjadwalan kuliah pada umumnya, yaitu mengalokasikan waktu dengan periode tertentu untuk beberapa mata kuliah praktikum serta sumber daya (ruang dan asisten) sehingga dapat memenuhi batasan atau aturan yang diberikan.

Batasan (*constraint*) pada penjadwalan ini dibagi menjadi dua kategori yaitu batasan keras (*hard constraint*) dan batasan lunak (*soft constraint*). Batasan keras adalah batasan yang harus dipenuhi, tidak boleh dilanggar, misalnya tidak ada sumber daya (ruang) yang dialokasikan untuk even berbeda pada waktu yang bersamaan. Sedangkan batasan lunak (*soft constraint*) adalah, batasan yang sebaiknya dipenuhi, tetapi tidak terlalu dibutuhkan untuk menghasilkan solusi yang valid. Oleh karena itu, batasan lunak ini dapat dipandang sebagai objek optimasi untuk algoritma pencarian. Contoh dari batasan lunak ini yaitu, meminimalkan jarak antara dua even berbeda (Kazarlis, 2005).

Formulasi penjadwalan praktikum ini hampir sama dengan formulasi penjadwalan kuliah. Permasalahan  $\{ P, S, R, C, W \}$  dimana  $P = \{p_1, p_2, p_3, \dots, p_n\}$  adalah periode untuk tiap subjek, tiap periode terdiri dari hari dalam satu minggu dimana dalam satu hari terdapat empat periode,  $S = \{s_1, s_2, s_3, \dots, s_n\}$  adalah subjek atau mata kuliah,  $R = \{r_1, r_2, r_3, \dots, r_n\}$  adalah ruangan laboratorium,  $C = \{c_1, c_2, c_3, \dots, c_n\}$  adalah *constraint* atau batasan yang harus dipenuhi atau bisa juga pinalti, dan  $W = \{w_1, w_2, w_3, \dots, w_n\}$  adalah bobot untuk tiap pinalti yang dilanggar (Kazarlis, 2006).

Dalam penyusunan jadwal praktikum di laboratorium terdapat beberapa aturan atau batasan yang harus dipenuhi. Pada penjadwalan praktikum ini aturan tersebut antara lain :

- a. Jadwal praktikum merupakan kombinasi dari komponen-komponen utama, yaitu ruang, mahasiswa, mata kuliah, asisten dan waktu. Apabila salah satu komponen tidak lengkap, maka akan dapat menyebabkan jadwal tidak dapat dipergunakan.
- b. Praktikum mata kuliah berbeda tidak dapat dilakukan dalam satu ruangan yang sama.
- c. Adanya keterbatasan ruang dan waktu
- d. Praktikum dilakukan di luar jam kuliah, dalam selang waktu jam perkuliahan efektif.

## 2.2 Algoritma Genetika

### 2.2.1 Gambaran Umum

Algoritma genetika adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*). 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 yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner. Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi.

Algoritma genetika sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode konvensional. Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operasi yaitu: operasi reproduksi, operasi persilangan (*crossover*), dan operasi mutasi. Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah – langkah sebagai berikut:

- a. Membangkitkan populasi awal secara random.
- b. Membentuk generasi baru dengan menggunakan tiga operasi diatas secara berulang-ulang sehingga diperoleh kromosom yang cukup untuk membentuk generasi baru sebagai representasi dari solusi baru.
- c. Evolusi solusi yang akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom hingga kriteria berhenti terpenuhi. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. beberapa kriteria berhenti yang sering digunakan antara lain:
  - berhenti pada generasi tertentu
  - berhenti jika sudah ditemukan solusi, yaitu solusi yang sesuai dengan kriteria



- berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi / terendah (tergantung persoalan) tidak berubah (konvergen).
- Kombinasi dari kriteria – kriteria diatas (Widhiyasa, 2007) .

### 2.3 Representasi kromosom

Representasi kromosom merupakan proses awal sebelum masuk proses regenerasi. Cara merepresentasikan permasalahan dalam kromosom merupakan suatu hal yang penting dalam algoritma genetik. Model representasi kromosom yang dapat dipergunakan untuk menyelesaikan suatu masalah, antara lain :

#### 1. Kromosom Biner

Kromosom biner merupakan kromosom yang disusun dari gen-gen yang bernilai 0 dan 1. Kromosom ini adalah model standar dalam algoritma genetika. Kromosom biner merupakan model yang sederhana dengan tingkat keberhasilan yang tinggi. Permasalahan yang direpresentasikan dengan kromosom biner antara lain, knapsack, TSP, Optimasi fungsi dll. Contoh representasi kromosom biner seperti pada gambar 2.1.

Kromosom A	1	0	1	1	0
Kromosom B	1	1	0	0	0

Gambar 2.1 Contoh Kromosom Biner

#### 2. Kromosom Float

Kromosom float merupakan model yang jumlah parameternya banyak. Tingkat keberhasilan dari bentuk kromosom ini rendah dalam kecepatan (jumlah generasi). Model *crossover* dan mutasi pada kromosom float ini sangat berbeda dengan model *crossover* dan mutasi pada kromosom biner. Sehingga perlu strategi khusus didalam menentukan model *crossover* dan mutasi. Contoh representasi kromosom float seperti pada gambar 2.2.

Kromosom A	1.2	3.4	1.6	2.2	4.1
Kromosom B	2.2	3.1	1.2	4.2	1.6

Gambar 2.2 Contoh Kromosom Float



### 3. Kromosom Nilai

Kromosom nilai tersusun dari gen – gen bernilai string dari suatu nilai (simbol). Nilai tersebut merupakan suatu hal yang berhubungan dengan masalah yang akan diselesaikan. Permasalahan yang menggunakan kromosom ini biasanya akan sulit dipecahkan jika menggunakan pengkodean biner (kromosom biner). Contoh representasi kromosom nilai seperti pada gambar 2.3.

Kromosom A	1,2 3,3 2,5 1,6 4,4 35
Kromosom B	AHDKHSTFNSFKSL
Kromosom C	(left) (right) (back)

Gambar 2.3 Contoh kromosom nilai

### 4. Kromosom Permutasi

Kromosom kombinatorial yaitu kromosom yang disusun dari gen – gen yang dinilai berdasarkan urutannya. Kromosom ini biasanya digunakan untuk menyelesaikan permasalahan pengurutan seperti *Ordering task* dan TSP. Contoh representasi kromosom permutasi seperti pada gambar 2.4.

Kromosom A	8 5 4 9 1 2 3 6 7
Kromosom B	9 1 2 4 3 8 5 7 6

Gambar 2.4 Contoh kromosom permutasi

Pada masalah TSP, representasi kromosom merupakan urutan dari kota-kota yang akan disinggahi secara berurutan dari kiri ke kanan. Sehingga, jika dimisalkan pada Gambar 2.2 terdapat 9 kota, representasi kromosom A berarti kota yang pertama disinggahi adalah berturut-turut kota 8, 5, 4, 9, 1, 2, 3, 6 dan kota 7 (Kusumadewi, 2003).

#### 2.4 Inisialisasi.

Pada awalnya solusi individual akan secara acak dibuat dalam bentuk sebuah inisial populasi. Besar populasinya sangat tergantung

pada keadaan masalah itu sendiri, tapi biasanya populasi mengandung sekitar beberapa ratus atau bahkan ribuan solusi yang mungkin. Secara sederhana, populasinya dibuat secara acak, dengan mengcover seluruh kemungkinan solusi (*search space*) (Widhiyasa, 2007).

## 2.5 Nilai *Fitness*

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Didalam evolusi alam, individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi  $h$  (dikenal sebagai masalah maksimasi), maka nilai *fitness* yang digunakan adalah nilai dari fungsi  $h$  tersebut, yakni *fitness*  $f = h$ . Tetapi jika masalahnya adalah meminimalkan fungsi  $h$  (masalah minimasi), maka fungsi  $h$  tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu nilai *fitness* yang bisa digunakan adalah  $f = 1/h$ , yang artinya semakin kecil nilai  $h$ , semakin besar nilai  $f$  (*fitness*). Tetapi hal ini akan menjadi masalah jika  $h$  bisa bernilai 0, yang mengakibatkan  $f$  bisa bernilai tak hingga. Untuk mengatasinya,  $h$  perlu ditambah sebuah bilangan yang dianggap sangat kecil sehingga nilai *fitness*nya menjadi :

$$f = \frac{1}{(h + a)} \dots \dots \dots (2.1)$$

dimana  $a$  adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan (Rein, 2006).

## 2.6 Seleksi

Seleksi digunakan untuk memilih individu untuk proses persilangan. Nilai *fitness* tiap individu akan menentukan terpilihnya individu dalam proses seleksi. Individu akan mempunyai probabilitas seleksi berdasarkan nilai *fitness*nya.

Metode seleksi yang umum digunakan antara lain :

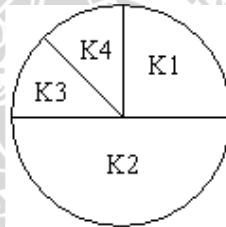
**a. Roulette Wheel**

Metode ini menirukan permainan *roulette-wheel* di mana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah. Tabel 2.1 dan gambar 2.5 berikut ini mengilustrasikan sebuah contoh penggunaan metode *roulette-wheel*.

Tabel 2.1 Kromosom dan Nilai *fitness*nya

Kromosom	Nilai <i>Fitness</i>
A	1
B	2
C	0,5
D	0,5
Jumlah	4

Tabel 2.1 menunjukkan suatu populasi dengan 5 kromosom dan telah dihitung masing-masing nilai *fitness*nya. Masing-masing kromosom memiliki *fitness* yang berbeda



Gambar 2.5 Contoh probabilitas kromosom pada roda *roulette*

Gambar 2.5 merupakan representasi nilai *fitness* masing-masing kromosom pada roda *roulette*.

**b. Seleksi Rangka (*Rank Selection*)**



Pada metode seleksi roda *roulette* akan bermasalah saat terdapat perbedaan *fitness* yang jauh. Sebagai contoh, jika *fitness* kromosom yang terbaik adalah 90% dari semua roda *roulette* dapat menyebabkan kromosom lain memiliki kesempatan yang kecil untuk dapat terpilih.

Proses dimulai dengan merangking atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*nya kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan nilai terburuk akan memiliki *fitness* baru nilai 1, terburuk kedua bernilai 2 dan begitu seterusnya, sehingga kromosom yang memiliki *fitness* terbaik akan memiliki nilai *fitness* N, dimana N adalah jumlah kromosom di dalam populasi. Seperti dapat dilihat pada tabel 2.2.

Tabel 2.2 Kondisi kromosom sebelum dan sesudah di ranking

Sebelum		Sesudah		
Kromosom	<i>Fitness</i>	Kromosom	<i>Fitness</i>	<i>Fitness</i> baru
A	15	B	5	1
B	5	D	5	2
C	10	E	5	3
D	5	C	10	4
E	5	A	15	5

Setelah adanya proses seleksi tersebut, maka saat ini seluruh kromosom mempunyai kesempatan untuk dipilih. Kelemahan metode ini terletak pada kecepatan pencapaiannya. Hal ini dikarenakan kromosom yang lebih baik tidak bisa menjadi petunjuk bagi populasi untuk mendapatkan solusi terbaik secara mudah, sehingga untuk mencapai solusi yang baik memerlukan waktu yang lebih lama.

### c. Seleksi Turnamen (Turnamen Selection)

Metode ini merupakan kombinasi dari seleksi roda *roulette* dan seleksi rangking. Sejumlah  $k$  kromosom tertentu dari populasi beranggota  $n$  kromosom ( $k \leq n$ ) dipilih secara acak dengan probabilitas yang sama. Dari  $k$  kromosom yang terpilih kemudian akan dipilih satu kromosom dengan *fitness* terbaik, yang diperoleh

dari hasil pengurutan rangking *fitness* semua kromosom terpilih. Perbedaan dengan seleksi roda *roulette* adalah pemilihan kromosom yang akan digunakan untuk berkembangbiak tidak berdasarkan skala *fitness* dari populasi.

## 2.7 Operator Genetika

Operator genetik dipergunakan untuk mengkombinasi (modifikasi) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya.

### 2.7.1 Operator Reproduksi

Langkah selanjutnya setelah proses seleksi dilakukan adalah dengan membuat generasi kedua dari populasi yang ada melalui persilangan (*crossover*) dan mutasi. Untuk setiap solusi baru yang dibentuk, sebuah pasangan "*parent*" atau orang tua solusi dipilih dari kumpulan populasi sebelumnya. Dengan membuat sebuah "*child*" atau anak solusi menggunakan metode diatas, yaitu persilangan dan mutasi, sebuah solusi baru telah dibuat, dimana pada umumnya anak akan mewarisi bagian-bagian dari orang tuanya. Orang tua baru dipilih lagi dan membuat suatu anak solusi lagi, dan berlanjut sampai suatu populasi solusi baru dengan ukuran yang cukup terbentuk.

### 2.7.2 Operator Pindah Silang (*Crossover*)

Operator pindah silang mempunyai peran yang paling penting dalam algoritma genetik karena didalamnya terdapat proses perkawinan (persilangan) gen antara dua individu (*parent*) yang menghasilkan dua individu baru (*offspring*) pada generasi berikutnya.

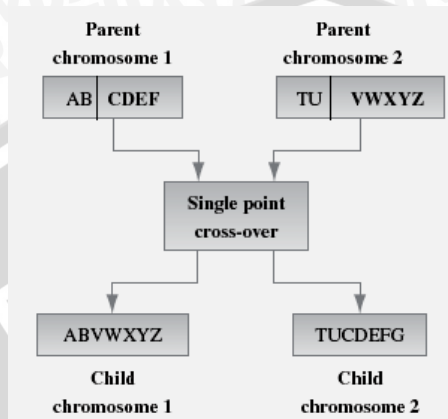
Penentuan individu yang berhak melakukan operasi *crossover* tergantung pada probabilitas *crossover* ( $P_c$ ) yang telah ditentukan. Langkah- langkah untuk melakukan operasi *crossover* sebagai berikut

- Tentukan probabilitas *crossover* ( $P_c$ ).
- Lakukan proses seleksi untuk memilih individu yang akan mengalami *crossover*
- Lakukan *crossover*

Beberapa teknik atau metode *crossover* yaitu (Cox, 2005) :

**a. Pindah Silang Satu Titik (*One Point Crossover*)**

Pada pindah silang satu titik (*one-point crossover*), satu titik sepanjang kromosom dipilih secara random. Segmen dari induk dari titik point ke kiri atau kekanan ditukar untuk menghasilkan individu baru. Mekanisme *one point crossover* diilustrasikan pada gambar 2.6.

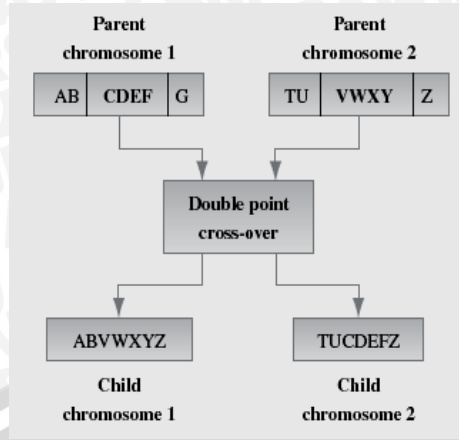


Gambar 2.6 Proses *one point crossover*

**b. Pindah Silang Dua Titik (*Two Point Crossover*)**

Pada pindah silang dua titik, dua titik sepanjang kromosom dipilih secara random. Segmen induk diantara kedua titik potong dipertukarkan untuk menghasilkan individu baru. Mekanisme *two point crossover* diilustrasikan pada gambar 2.7.

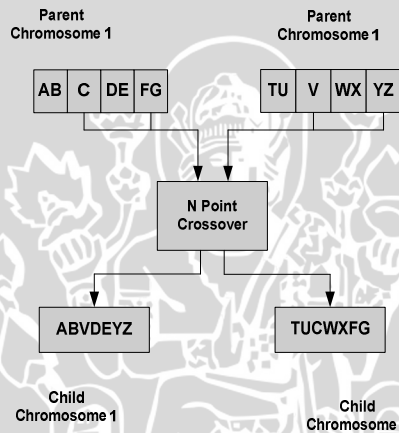




Gambar 2.7 Proses *two point crossover*

**c. Pindah Silang Banyak Titik (*N-point Crossover*)**

Pada pindah silang banyak titik, sejumlah N titik sepanjang kromosom dipilih secara random, dimana titik potong tidak boleh sama. Mekanisme *n point crossover* diilustrasikan pada gambar 2.7.

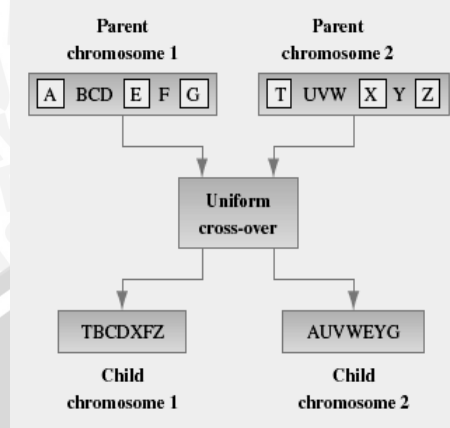


Gambar 2.8 Proses *n point crossover*

**d. Pindah silang Seragam (*Uniform Crossover*)**

Pada *uniform crossover*, sejumlah N titik ditentukan secara random baik itu jumlah maupun letaknya. Segmen yang dipertukarkan sesuai dengan jumlah titik potong yang didapatkan

secara random tersebut. *Uniform crossover* hampir sama dengan *N point crossover*. Untuk *uniform crossover*, jumlah titik ditentukan secara random sedangkan *N point* jumlah titik sudah ditentukan. *Uniform crossover* juga akan sama dengan *two point crossover* jika titik potong yang dihasilkan dua titik.



Gambar 2.9 Proses *uniform crossover*

**e. Pindah silang Aritmatik (*Arithmetic Crossover*)**

Pada pindah silang ini digunakan operator aritmatik untuk menghasilkan kromosom baru. Contoh :

Induk 1	:	1	1	0	0	1	0	1	1	1	0	0	0
Induk 2	:	1	0	0	1	1	1	0	0	1	1	0	1
Anak (OR)	:	1	1	0	1	1	1	1	1	1	1	0	1

**f. Pindah Silang untuk Pengkodean Permutasi**

Pindah silang dengan permutasi (*Permutation Crossover*) ini biasa menggunakan teknik pindah silang satu titik karena prosesnya yang sederhana dan agar tetap menjaga konsistensi urutan nilai pada kromosom.

Dalam proses penyilang ini, pertama kali ditentukan posisi titik pindah silang. Kemudian dari *gen* pertama sampai dengan titik perkawinan silang disalin dari induk pertama, sedangkan sisanya didapatkan dengan cara melihat satu persatu nilai pada orang tua kedua, jika belum ada pada kromosom keturunan, maka nilai tersebut ditambahkan. Berikut contoh :

Induk 1 : 1 2 3 4 5 6 7 8 9  
 Induk 2 : 4 5 3 6 8 9 7 2 1  
 Anak 1 : 1 2 3 4 5 6 8 7 9  
 Anak 2 : 4 5 3 6 8 1 2 7 9

### 2.7.3 Operator Mutasi

Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal. Sehingga mutasi akan meningkatkan variasi populasi.

Individu yang terpilih untuk proses mutasi dapat dilakukan dengan membandingkan nilai probabilitas mutasinya dengan probabilitas mutasi yang telah ditentukan atau dapat dipilih secara acak (random).

Representasi kromosom yang berbeda akan membedakan jenis mutasi yang digunakan. Metode mutasi yang biasanya digunakan yaitu :

- a. **Inversi**, mutasi ini dilakukan dengan mengubah nilai 1 menjadi 0 atau sebaliknya. Mutasi ini biasanya terapkan pada kromosom biner.

$$a(k) = a(k) - 1$$

$$C : 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1$$

$$C' : 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1$$

- b. **Random**, mutasi ini dilakukan dengan mengganti gen yang termutasi dengan nilai random.

$$a(k) = \text{random}$$

$$C : 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$

$$C' : 1 \ 2 \ 9 \ 4 \ 5 \ 6 \ 7$$

- c. **Shift**, mutasi ini dilakukan dengan menggeser nilai gen termutasi sebesar  $\epsilon$ , dimana  $\epsilon$  adalah nilai bilangan kecil yang ditentukan.

$$a(k) = a(k) + \epsilon$$

- d. **Swap**, mutasi ini dilakukan dengan menukar urutan *gen* yang terdapat pada satu kromosom, sehingga konsistensi urutan *gen* tetap terjaga. Contoh untuk pengkodean permutasi :

$$\text{Induk} : 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$



Anak : 1 4 3 2 5 6 7

Probabilitas mutasi yang baik berada pada kisaran 0 sampai dengan 0.3. Probabilitas mutasi yang terlalu kecil menyebabkan terjebak dalam optimum lokal, dan probabilitas mutasi yang terlalu besar menyebabkan konvergensi sulit didapatkan (Basuki, 2003).

**2.8 Analisis Crossover dan Penjadwalan dengan Algoritma Genetika.**

**a. Analisis Crossover**

Pada penelitian analisis crossover yang dilakukan K. De Jong dan W. Spears ini, perbandingan *crossover* pada algoritma genetika, diterapkan pada penyelesaian *n-peak problem* menggunakan algoritma genetika.

Penelitian dilakukan dengan membandingkan lima teknik *crossover* (*2-point*, *4-point*, *8-point*, *16-point* dan *uniform crossover*) dan empat ukuran populasi berbeda (20,50,100 dan 1000). Hasil dari penelitian seperti terlihat pada tabel 2.3 menunjukkan *uniform crossover* mendominasi pada permasalahan sederhana dan ukuran populasi yang lebih kecil, yaitu 20. Pada ukuran populasi yang lebih besar seperti 1000, *two point crossover* mempunyai *performance* yang lebih baik.

Tabel 2.3 Hasil Uji Coba Perbandingan Teknik *Crossover*

2-point vs. Uniform				
Problem	Population Size			
	20	50	100	1000
6-Peak	Uniform	Uniform	2-point	2-point
5-Peak	2-point	2-point	2-point	2-point
4-Peak	?	2-point	?	2-point
3-Peak	Uniform	?	2-point	2-point
2-Peak	Uniform	2-point	?	2-point
1-Peak	Uniform	Uniform	?	?

**b. Penjadwalan dengan Algoritma Genetika**

Penyelesaian penjadwalan praktikum dengan algoritma genetika sebelumnya sudah banyak dilakukan, salah satunya yang dilakukan S. Ghaemi dan M.T. Vakili. Penelitian dilakukan untuk menyelesaikan penjadwalan mata kuliah di fakultas elektro

universitas Tabriz, Iran. Permasalahan didefinisikan dengan lima entitas, yaitu dosen, mahasiswa, mata kuliah, ruang kuliah dan periode atau waktu kuliah.

Batasan (*constraints*) yang digunakan yaitu batasan keras, diberi pinalti 1000 sedangkan batasan lunak diberi pinalti antara 5-400. Untuk pengkodean kromosom, digunakan vektor dimana indeks pertama dari setiap gen menunjukkan mata kuliah dan indeks kedua menunjukkan periode (timeslot). Gambar 2.10 menunjukkan contoh representasi dr kromosom yang digunakan.

Subject	Electronic 1	Machin2	Machin1
Timeslot	(Mon,1,2) (Wed,2,2)	(Tue,2,1) (Sat,3,1)	(Tue,1,3) (Sun,2,2)

Gambar 2.10 Contoh kromosom

Pada penelitian ini, digunakan dua metode seleksi serta tiga teknik *crossover*. Metode seleksi yang digunakan yaitu metode *roulette wheel wighting* dan metode turnamen. Untuk teknik *crossover*, digunakan *single point crossover*, *uniform crossover* dan *modified uniform crossover*.

Hasil yang didapat dari penelitian ini yaitu, *single point crossover* pada tingkat mutasi lebih tinggi serta *uniform crossover* dan *modified uniform crossover* pada tingkat mutasi rendah mencapai hasil yang lebih baik. Hasil yang sama juga ditunjukkan pada perbedaan teknik seleksi yang digunakan, seleksi turnamen menghasilkan jumlah konflik yang lebih kecil.





## BAB III METODOLOGI DAN PERANCANGAN

### 3.1 Deskripsi Umum Sistem

Pada analisis *crossover* yang diterapkan pada penyelesaian penjadwalan praktikum ini, akan dibandingkan tiga teknik *crossover* yaitu *one point crossover*, *two point crossover* dan *uniform crossover*. Perbandingan akan dilakukan pada hasil akhir proses genetika atau setelah didapatkan hasil yang konvergen dimana nilai *cost* atau *fitness* tidak mengalami perubahan. Hal yang akan dibandingkan adalah rata – rata nilai *cost* dari individu yang dihasilkan oleh ketiga teknik *crossover* tersebut. Teknik *crossover* dikatakan lebih baik jika individu yang dihasilkan mempunyai nilai *cost* yang lebih baik dibandingkan teknik lainnya.

### 3.2 Analisa Data

Pada penelitian ini, data yang digunakan yaitu data komponen penjadwalan praktikum di laboratorium Jurusan Matematika Universitas Brawijaya. Komponen penjadwalan yang merupakan spesifikasi dari masalah penjadwalan ditunjukkan pada tabel 3.1.

Tabel 3.1 Spesifikasi Permasalahan

No	Deskripsi	Jumlah
1	Jumlah Ruang	3
2	Jumlah Mata Kuliah Praktikum	22
3	Jumlah Hari	5
4	Jumlah Periode Waktu Perhari	4

### 3.3 Perancangan Proses Penjadwalan Dengan Algoritma Genetika

#### 3.3.1 Model Genetika

Pengkodean yang akan digunakan pada penjadwalan praktikum ini adalah pengkodean nilai untuk mendapatkan nilai sebenarnya dari aturan yang direpresentasikan.

Pengkodean kromosom terdiri dari komponen- komponen penjadwalan, yaitu kelas mata kuliah, ruang dan waktu. Waktu disini merupakan gabungan dari hari dan jam, sehingga model genetika

dari penjadwalan ini terdiri dari kelas mata kuliah, ruang dan waktu. Panjang dari kromosom adalah sebanyak gen yang ada, dalam hal ini setiap gen mewakili mata kuliah yang ditawarkan.

Untuk tiap kromosom, kelas mata kuliah ditempatkan pada waktu serta ruang secara bersamaan, dimana kelas mata kuliah tidak diperbolehkan ditempatkan pada dua waktu dan ruang yang bersamaan.

Mata Kuliah 1	Mata Kuliah 2	Mata Kuliah 3	Mata Kuliah 5
H1,P3, R1	H2,P4,R2	H3,P1,R3	H4,H5,R1

Gambar 3.1 Model Genetika

Keterangan:

- Mata kuliah : Mata Kuliah yang diajarkan
- H : Hari Praktikum
- P : Periode praktikum
- R : Ruang laboratorium

Contoh penempatan mata kuliah seperti pada gambar 3.1 diatas, mata kuliah 1 dijadwalkan pada hari senin, periode ke-3 dan ditempatkan pada lab. A.

Tabel 3.2 Inisialisasi Kromosom

Mata Kuliah	1	2	3	4	5	6
Kromosom 1	1,3,A	2,1,B	3,3,C	4,2,B	2,2,A	5,1,C
Kromosom 2	2,2,B	1,4,,A	3,3,C	1,1,C	5,4,B	4,2,A
.....	...	...	...	...	...	...
Kromosom n	1,5,C	2,4,A	4,5,B	3,2,A	2,1,B	5,4,C

Pada Tabel 3.2 merupakan ilustrasi dari inisialisasi kromosom. Setiap kromosom adalah barisan gen yang terdiri dari tiga nilai yaitu nilai hari, jam dan ruang. Nilai gen didapatkan dengan membangkitkan bilangan secara random. Hari dinyatakan dengan

nilai 1 sampai 5 yang mewakili hari senin sampai dengan jumat. Sedangkan waktu dinyatakan dengan nilai yang sama yaitu 1 sampai 5 yang mewakili periode waktu perhari.

Setelah terjadi proses genetika dihasilkan kromosom-kromosom yang terdiri dari beberapa gen (sesuai dengan jumlah mata kuliah) yang berisi slot waktu (hari dan jam) untuk setiap mata kuliah.

Proses random dalam penyusunan komponen kromosom memungkinkan adanya gen bernilai sama baik hari, waktu dan ruang dalam satu kromosom. Kromosom dengan gen yang bernilai sama tidak dianggap ilegal dan tetap disertakan dalam proses genetika selanjutnya. Kromosom tersebut akan diberi nilai pinalti seperti akan dijelaskan pada subbab 3.3.3. Dengan pemberian nilai pinalti, kromosom akan mempunyai nilai *fitness* yang lebih rendah. Proses genetika yang akan berjalan akan menghasilkan individu yang terbaik, sehingga kromosom dengan nilai *fitness* yang rendah akibat gen yang bernilai sama tidak akan terpilih sebagai solusi terbaik.

### 3.3.2 Relasi Tabel Basis Data

Basis data yang akan dibangun merupakan komponen utama dan komponen tambahan dari penjadwalan. Komponen utama yaitu mata kuliah, ruang dan waktu. Sedangkan komponen tambahan yaitu kelas mata kuliah yang merupakan pengkodean kromosom yang merupakan hasil dari pembuatan jadwal.

Tabel Mata Kuliah ialah tabel yang digunakan untuk menyimpan data-data mata kuliah yaitu kode mata kuliah, nama mata kuliah, tahun, kode program studi, apakah ganjil dan apakah praktikum.

Tabel 3.3 Tabel Mata Kuliah

Field	Type	Allow Null
k_mk	Text (10)	No
tahun	Text (5)	No
nama	Text (50)	No
k_prog_studi	Text (3)	No
Is_ganjil	Yes/No	No
Is_praktikum	Yes/No	No



Tabel Prodi ialah tabel yang digunakan untuk menyimpan data-data program studi yaitu kode program studi, nama program studi.

Tabel 3.4 Tabel Program Studi

Field	Type	Allow Null
k_prog_studi	Text (3)	No
content	Text (50)	No

Tabel Hari ialah tabel yang digunakan untuk menyimpan data hari, yaitu kode hari dan nama hari.

Tabel 3.5 Tabel Hari

Field	Type	Allow Null
k_hari	Text (3)	No
content	Text (10)	No

Tabel Waktu ialah tabel yang digunakan untuk menyimpan data waktu, yaitu kode waktu dan nama waktu.

Tabel 3.6 Tabel Waktu

Field	Type	Allow Null
k_waktu	Text (3)	No
mulai	Date/time	No
selesai	Date/time	No

Tabel Asisten ialah tabel yang digunakan untuk menyimpan data asisten, yaitu kode asisten dan nama asisten.

Tabel 3.7 Tabel Asisten

Field	Type	Allow Null
k_asisten	Text (15)	No
nama	Text (50)	No

Tabel Kelas ialah tabel yang digunakan untuk menyimpan data kelas mata kuliah, yaitu kode mata kuliah, kode kelas, dan tahun.

Tabel 3.8 Kelas Mata Kuliah

Field	Type	Allow Null
k_mk	Text (10)	No
k_kelas	Text (5)	No
tahun	number	No

Tabel Asisten kelas ialah tabel yang digunakan untuk menyimpan data asisten yang mengajar kelas mata kuliah, yaitu kode mata kuliah, tahun, kode kelas dan kode asisten.

Tabel 3.9 Tabel Asisten Kelas

Field	Type	Allow Null
k_mk	Text (10)	No
tahun	number	No
k_kelas	Text (5)	No
k_asisten	Text (15)	No

Tabel Jadwal untuk menyimpan jadwal dari praktikum yang merupakan representasi dari kromosom. Tabel ini untuk menyimpan data kode mata kuliah, tahun, kode kelas, kode hari, kode waktu dan kode ruang.

Tabel 3.10 Tabel Jadwal

Field	Type	Allow Null
k_mk	Text (10)	No
tahun	number	No
k_kelas	Text (5)	No
k_hari	Text (3)	No
k_waktu	Text (3)	No
K_ruang	Text (3)	No





$$\text{Fungsi Cost } (f(x)) = \sum_{i=1}^m w_i \delta_i \quad \begin{cases} \delta_i = 1 \\ \delta_i = 0 \end{cases} \dots\dots(3.1)$$

Dimana  $\delta_i = 0$  jika pinalti tidak dilanggar dan  $\delta_i = 1$  jika pinalti dilanggar.  $m$  adalah jumlah pinalti dan  $W_i$  adalah bobot dari tiap pinalti.

Sedangkan fungsi *fitnessnya* :

$$F = \frac{1}{((f(x))+1)} \dots\dots\dots(3.2)$$

Pada penjadwalan praktikum yang akan dibuat diberikan sejumlah aturan beserta pinalti dari setiap aturan seperti pada tabel 3.12:

Tabel 3.12 Aturan dan Nilai Pinalti

No	Aturan	Nilai
1	Asisten mengajar dua kelas dalam satu waktu	3
2	Pengalokasian ruang pada waktu yang bersamaan	5
3	Tabrakan waktu praktikum dan kuliah	7

**Aturan Pinalti :**

- 1. Asisten mengajar dua kelas dalam satu waktu**  
Asisten yang menangani praktikum untuk beberapa mata kuliah tidak diperbolehkan mengajar dua kelas secara bersamaan dalam satu waktu. Jadwal yang menjadwalkan dua asisten secara bersamaan akan diberi nilai pinalti 3.
- 2. Pengalokasian ruang pada waktu yang bersamaan**  
Ruang yang terbatas harus dialokasikan pada mata kuliah dan waktu secara tepat. Apabila dalam proses penjadwalan terdapat ruang yang dialokasikan pada dua waktu yang bersamaan, maka jadwal akan diberi nilai pinalti 5.
- 3. Tabrakan waktu praktikum dan kuliah**  
Jadwal praktikum tidak boleh berbenturan dengan jadwal kuliah yang sudah disusun sebelumnya. Misalnya jadwal praktikum

pemrograman II dijadwalkan pada periode dua, ternyata bentrok dengan mata kuliah matematika lanjut, maka jadwal tersebut diberi pinalti 7.

**3.3.4 Seleksi**

Seleksi *roulette wheel*, semakin tinggi nilai *fitness* maka semakin besar kemungkinan untuk terpilih menjadi induk.

Pemilihan metode *roulette wheel* ini karena metode ini mudah untuk diimplementasikan serta kemungkinan terpilihnya individu dengan nilai *fitness* tinggi lebih besar dibandingkan individu dengan nilai *fitness* rendah.

**3.3.5 Crossover**

Pada penelitian ini digunakan tiga teknik *crossover* yaitu *one point crossover*, *two point crossover*, dan *uniform crossover*. Hasil individu baru atau keturunan dari masing – masing teknik *crossover* akan dibandingkan teknik manakah yang menghasilkan individu yang relatif lebih baik dibandingkan teknik lainnya.

**a. One Point Crossover**

Pada proses *crossover* ini, induk dibagi menjadi 2 segmen dimana hasil anakan merupakan kombinasi dari pertukaran segmen kedua induk.

Induk 1 :

3,1,B	2,5,A	1,4,C	5,2,A	1,1,B	4,3,C	2,5,C
-------	-------	-------	-------	-------	-------	-------

Induk 2:

2,4,A	3,5,C	1,2,B	4,4,B	2,2,C	1,3,A	3,4,A
-------	-------	-------	-------	-------	-------	-------

Anak :

3,1,B	2,5,A	1,4,C	5,2,A	<b>2,2,C</b>	<b>1,3,A</b>	<b>3,4,A</b>
-------	-------	-------	-------	--------------	--------------	--------------

**b. Two Point Crossover**

Pada proses *crossover* ini, induk dibagi menjadi 3 segmen dimana hasil anakan merupakan kombinasi dari pertukaran segmen kedua induk.

Induk 1 :

3,1,B	2,5,A	1,4,C	5,2,A	1,1,B	4,3,C	2,5,C
-------	-------	-------	-------	-------	-------	-------

Induk 2:

2,4,A	3,5,C	1,2,B	4,3,C	2,2,C	1,3,A	3,4,A
-------	-------	-------	-------	-------	-------	-------

Anak :

3,1,B	2,5,A	<b>1,2,B</b>	<b>4,3,C</b>	<b>2,2,C</b>	4,3,C	2,5,C
-------	-------	--------------	--------------	--------------	-------	-------

**c. Uniform Crossover**

Pada proses *crossover* ini, induk dibagi menjadi beberapa segmen dimana titik potong serta pembagian segmen induk ditentukan secara random. Hasil anakan merupakan kombinasi dari pertukaran segmen kedua induk.

Induk 1 :

3,1,B	2,5,A	1,4,C	5,2,A	1,1,B	4,3,C	2,5,C
-------	-------	-------	-------	-------	-------	-------

Induk 2:

2,4,A	3,5,C	1,2,B	4,4,B	2,2,C	1,3,A	3,4,A
-------	-------	-------	-------	-------	-------	-------

3,1,B	3,5,C	1,2,B	5,2,A	1,1,B	1,3,A	3,4,A
-------	-------	-------	-------	-------	-------	-------

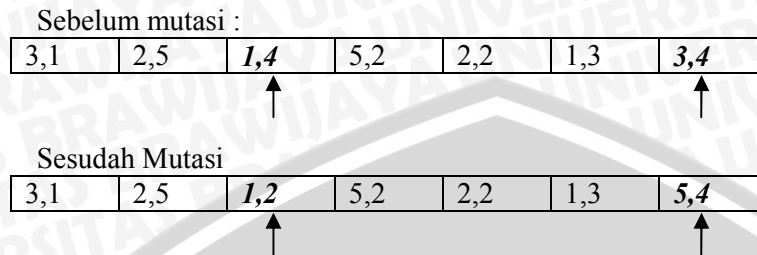
Apabila anak hasil *crossover* mempunyai gen yang identik seperti pada hasil *two point crossover* diatas, Individu tetap dianggap legal. Individu tersebut akan diberi nilai pinalti sesuai aturan pinalti pada subbab sebelumnya. Nilai pinalti yang diberikan akan berpengaruh pada nilai *fitness* individu, sehingga setelah iterasi proses genetika yang berulang individu dengan gen identik tidak akan terpilih sebagai solusi terbaik atau tereliminasi.

**3.3.6 Mutasi**

Setelah dilakukan *crossover* untuk menghasilkan individu baru, langkah selanjutnya adalah mutasi yang bertujuan untuk meningkatkan variasi dalam populasi atau mencegah konvergensi dini.



Mutasi yang diterapkan pada penelitian ini yaitu mutasi random, dimana gen yang dimutasi akan diganti nilainya dengan suatu nilai random. Untuk penentuan individu yang terpilih, juga dilakukan secara random. Ilustrasi mutasi secara random :



### 3.3.7 Contoh Manual

#### 1. One Point Crossover

##### ➤ Parameter Awal

- Jumlah individu dalam 1 generasi = 5
- Jumlah generasi = 2
- Probabilitas mutasi = 0,1

Dengan cara random didapatkan generasi awal seperti di tunjukkan pada tabel 3.13.

Tabel 3.13 Generasi Awal *One Point Crossover*

No	1	2	3	4	5	6	Cost	Fitness	PS	Range
1	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	8	0,125	0,24	1-24
2	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	11	0,091	0,17	25-41
3	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C	12	0,083	0,16	42-57
4	3,2,A	4,4,C	2,3,B	4,1,C	1,5,B	3,1,A	15	0,067	0,13	58-70
5	2,4,A	5,3,B	2,4,C	1,5,C	5,2,A	3,1,B	17	0,059	0,11	71-81
6	2,1,B	5,2,C	2,5,A	1,5,B	1,4,A	1,3,C	19	0,053	0,10	82-91
7	4,1,C	5,5,B	2,1,A	3,2,A	4,3,B	2,5,C	20	0,050	0,09	92-100

Untuk menentukan individu yang akan akan mengalami *crossover*, dilakukan proses seleksi. Proses seleksi dengan *roulette wheel* didapatkan individu 1 dan 2 yang akan disilangkan.

→ P1 Crossover P2

P1(1)	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	4,1,A
P2(2)	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	3,1,B

→ Hasil Crossover

C1	4,1,A	5,5,B	2,1,C	4,2,A	5,3,C	2,1,B	4,1,A
C2	3,1,B	1,2,A	3,5,C	3,2,B	4,3,C	2,5,A	3,1,B

Langkah Selanjutnya adalah mutasi, dimana prosesnya sebagai berikut :

Sebelum dimutasi :

2,4,A	5,3,B	2,4,C	1,5,C	5,2,A	3,1,B	2,4,A	16
-------	-------	-------	-------	-------	-------	-------	----

Setelah dimutasi

2,4,A	5,3,B	<b>2,2,C</b>	<b>1,3,C</b>	5,2,A	3,1,B	2,4,A	16
-------	-------	--------------	--------------	-------	-------	-------	----

Setelah proses mutasi dilakukan, nilai *fitness* masing- masing individu dihitung. Individu baru akan dimasukkan dalam populasi generasi berikutnya jika nilai *fitness*nya lebih baik dibandingkan individu pada populasi saat ini.

→ Generasi 1

	1	2	3	4	5	6	Cost	F(x)
1	4,1,A	5,5,B	2,1,C	<b>4,2,A</b>	<b>5,3,C</b>	<b>2,1,B</b>	7	0,143
2	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	8	0,125
3	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	11	0,091
4	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C	12	0,083
5	3,1,B	1,2,A	3,5,C	<b>3,2,B</b>	<b>4,3,C</b>	<b>2,5,A</b>	12	0,083

6	3,2,A	4,4,C	2,3,B	4,1,C	1,5,B	3,1,A	15	0,067
7	2,4,A	5,3,B	2,2,C	1,3,C	5,2,A	3,1,B	16	0,063

Pada perhitungan manual ini, iterasi berhenti pada generasi I. dari generasi I tersebut diperoleh jadwal terbaik yaitu 4,1,A; 5,5,B; 2,1,C; 4,2,A; 5,3,C; 2,1,B, dengan nilai *fitness* 0,143. Iterasi ini dapat diteruskan sampai didapat nilai *fitness* yang optimal atau sampai pada generasi tertentu yang sudah ditentukan.

2. **Two Point Crossover**

➤ Parameter Awal

- Jumlah individu dalam 1 generasi = 5
- Jumlah generasi = 2
- Probabilitas mutasi = 0,1

Dengan cara random didapatkan generasi awal seperti ditunjukkan pada tabel 3.14.

Tabel 3.14 Generasi Awal *Two Point Crossover*

No	1	2	3	4	5	6	Cost	Fitness	PS	Range
1	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	8	0,125	0,24	1-24
2	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	11	0,091	0,17	25-41
3	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C	12	0,083	0,16	42-57
4	3,2,A	4,4,C	2,3,B	4,1,C	1,5,B	3,1,A	15	0,067	0,13	58-70
5	2,4,A	5,3,B	2,4,C	1,5,C	5,2,A	3,1,B	17	0,059	0,11	71-81
6	2,1,B	5,2,C	2,5,A	1,5,B	1,4,A	1,3,C	19	0,053	0,10	82-91
7	4,1,C	5,5,B	2,1,A	3,2,A	4,3,B	2,5,C	20	0,050	0,09	92-100

Untuk menentukan individu yang akan mengalami *crossover*, dilakukan proses seleksi. Proses seleksi dengan *rhoutte wheel* didapatkan individu 4 dan 1 yang akan disilangkan.

➔ P4 Crossover P1

P(3)	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C
------	-------	-------	-------	-------	-------	-------

➔ Hasil Crossover



C1	1,3,C	3,5,A	3,5,C	4,2,A	3,1,A	1,2,C
----	-------	-------	-------	-------	-------	-------

Langkah Selanjutnya adalah mutasi, dimana prosesnya sebagai berikut :

Sebelum dimutasi :

3,2,A	4,4,C	2,3,B	4,1,C	1,5,B	3,1,A	15
-------	-------	-------	-------	-------	-------	----

Setelah dimutasi

3,2,A	1,3,C	2,3,B	4,1,C	5,2,B	3,1,A	15
-------	-------	-------	-------	-------	-------	----

Setelah proses mutasi dilakukan, nilai *fitness* masing- masing individu dihitung. Individu baru akan dimasukkan dalam populasi generasi berikutnya jika nilai *fitness*nya lebih baik dibandingkan individu pada populasi saat ini.

→ **Generasi I**

	1	2	3	4	5	6	Cost	F(x)
1	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	8	0,125
2	3,1,B	1,2,A	3,4,B	5,3,B	5,3,C	2,1,B	10	0,100
3	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	11	0,091
4	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C	12	0,083
5	1,3,C	3,5,A	3,5,C	4,2,A	3,1,A	1,2,C	12	0,083
6	3,2,A	1,3,C	2,3,B	4,1,C	5,2,B	3,1,A	13	0,077
7	2,4,A	5,3,B	2,4,C	1,5,C	5,2,A	3,1,B	17	0,059

Pada perhitungan manual ini, iterasi berhenti pada generasi I. dari generasi I tersebut diperoleh jadwal terbaik yaitu 4,1,A; 5,5,B; 2,1,C; 3,2,B; 4,3,C; 2,5,A dengan nilai *fitness* 0,125. Iterasi ini dapat diteruskan sampai didapat nilai *fitness* yang optimal atau sampai pada generasi tertentu yang sudah ditentukan.

### 3. Uniform Crossover

#### ➤ Parameter Awal

- Jumlah individu dalam 1 generasi = 5
- Jumlah *generasi* = 2
- Probabilitas mutasi = 0,1

Dengan cara random didapatkan generasi awal seperti ditunjukkan pada tabel 3.15

Tabel 3.15 Generasi Awal *Uniform Crossover*

	1	2	3	4	5	6	Cost	<i>Fitness</i>	PS	Range
1	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	8	0,125	0,24	1-24
2	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	11	0,091	0,17	25-41
3	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C	12	0,083	0,16	42-57
4	3,2,A	4,4,C	2,3,B	4,1,C	1,5,B	3,1,A	15	0,067	0,13	58-70
5	2,4,A	5,3,B	2,4,C	1,5,C	5,2,A	3,1,B	17	0,059	0,11	71-81
6	2,1,B	5,2,C	2,5,A	1,5,B	1,4,A	1,3,C	19	0,053	0,10	82-91
7	4,1,C	5,5,B	2,1,A	3,2,A	4,3,B	2,5,C	20	0,050	0,09	92-100

Untuk menentukan individu yang akan mengalami *crossover*, dilakukan proses seleksi. Proses seleksi dengan *rhoutte wheel* didapatkan individu 4 dan 1 yang akan disilangkan.

#### ➔ P2 Crossover P7

P(2)	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B
P(7)	4,1,C	5,5,B	2,1,A	3,2,A	4,3,B	2,5,C

#### ➔ Hasil Crossover

C(1)	3,1,B	5,5,B	3,5,C	3,2,A	5,3,C	2,5,C
C(2)	4,1,C	1,2,A	2,1,A	4,2,A	4,3,B	2,1,B

Langkah Selanjutnya adalah mutasi, dimana prosesnya sebagai berikut :

Sebelum dimutasi :

4,1,C	1,2,A	2,1,A	4,2,A	4,3,B	2,1,B	16
-------	-------	-------	-------	-------	-------	----

Setelah dimutasi

4,1,C	1,2,A	2,3,A	4,2,A	4,2,B	2,1,B	16
-------	-------	-------	-------	-------	-------	----

Setelah proses mutasi dilakukan, nilai *fitness* masing- masing individu dihitung. Individu baru dengan nilai *fitness* yang lebih baik dibandingkan individu pada generasi sebelumnya menggantikan posisi individu generasi sebelumnya pada populasi generasi selanjutnya.

→ Generasi 1

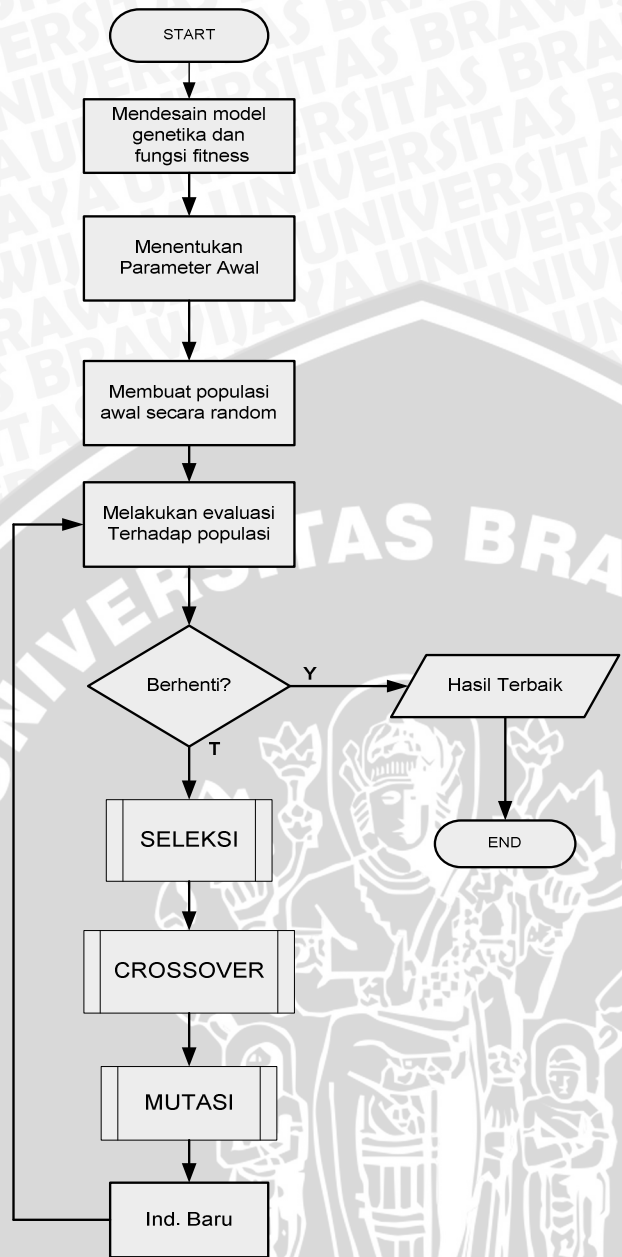
	1	2	3	4	5	6	Cost	F(x)
1	4,1,A	5,5,B	2,1,C	3,2,B	4,3,C	2,5,A	8	0,125
2	3,1,B	1,2,A	3,5,C	4,2,A	5,3,C	2,1,B	11	0,091
3	1,3,C	3,5,A	3,4,B	5,3,B	3,1,A	1,2,C	12	0,083
4	3,1,B	5,5,B	3,5,C	3,2,A	5,3,C	2,5,C	14	0,071
5	3,2,A	4,4,C	2,3,B	4,1,C	1,5,B	3,1,A	15	0,067
6	4,1,C	1,2,A	2,3,A	4,2,A	4,2,B	2,1,B	15	0,067
7	2,4,A	5,3,B	2,4,C	1,5,C	5,2,A	3,1,B	17	0,059

Pada perhitungan manual ini, iterasi berhenti pada generasi I. dari generasi I tersebut diperoleh jadwal terbaik yaitu 4,1,A; 5,5,B; 2,1,C; 3,2,B; 4,3,C; 2,5,A dengan nilai *fitness* 0,125. Iterasi ini dapat diteruskan sampai didapat nilai *fitness* yang optimal atau sampai pada generasi tertentu yang sudah ditentukan.

**3.4 Flowchart Algoritma Genetika**

Beberapa langkah perancangan proses algoritma genetika dengan dapat digambarkan dengan flow chart seperti pada gambar 3.4.





Gambar 3.4 Penyelesaian Dengan Algoritma Genetika

Langkah-langkah spesifik algoritma genetika yang digunakan antara lain :

1. Mendesain model genetika serta fungsi *fitness* algoritma genetika.
2. Menentukan semua parameter awal yang dibutuhkan  
Parameter-parameter yang digunakan dalam skema algoritma genetika ini adalah sebagai berikut :
  - Banyak individu dalam 1 generasi, ini dapat ditentukan sendiri tanpa ada aturan tertentu.
  - Probabilitas *Crossover*, perlu ditentukan besar kemungkinan terjadinya *crossover* pada satu individu. Nilai ini berkisar antara 0-1.
  - Probabilitas mutasi, perlu ditentukan besar kemungkinan terjadinya mutasi pada satu individu. Nilai ini berkisar antara 0-1.
  - Jumlah generasi, sebenarnya menunjukkan berapa kali iterasi yang dilakukan. Misalkan hendak dilakukan 10 kali iterasi, maka iterasi dihentikan apabila telah didapatkan generasi baru dari generasi ke-10.
3. Membuat populasi awal secara random.
4. Melakukan evaluasi terhadap populasi dengan mengitung nilai *fitness* masing-masing individu. Individu di ranking berdasarkan nilai *fitness*, dimana individu dengan nilai *fitness* yang baik yang bertahan dalam populasi.
5. Memeriksa kriteria proses genetika berhenti, pada penelitian ini proses genetika berhenti jika dicapai konvergensi.
6. Melakukan proses seleksi menggunakan metode *roulette wheel*, dimana individu dengan nilai *fitness* lebih baik mempunyai kemungkinan terpilih lebih besar.
7. Untuk *crossover* akan dilakukan dengan tiga teknik *crossover*.
8. Operator mutasi diberlakukan setelah didapatkan individu baru. Penentuan individu yang terkena mutasi dilakukan secara random. Bila terkena operator mutasi, gen yang termutasi diganti dengan nilai baru yang didapat secara random.
9. Menghasilkan individu baru setelah mengalami proses seleksi, *crossover* dan mutasi
10. Jadwal terbaik adalah individu terbaik pada generasi terakhir.

### 3.4.1 Pseudocode Proses Crossover

#### 1. One Point Crossover

Pseudocode :

```

Begin one-point crossover
find a random cutpoint
for i = 0 to cutpoint
  child A chromosom[x] = parent A gene[x]
  child B chromosom[x] = parent B gene[x]
end for
for i = cutpoint to gene length
  child A gene[x] = parent B gene[x]
  child B gene[x] = parent A gene[x]
end for
return children
End two-point crossover

```

Gambar 3.5 Pseudocode One Point Crossover

#### 2. Two point crossover

Pseudocode :

```

Begin two-point crossover
find two random cutpoint
for i = 0 to cutpoint 1
  child A gene[i] = parent A gene[i]
  child B gene[i] = parent B gene[i]
end for
for i = cutpoint1 to cutpoint2
  child A gene[i] = parent B gene[i]
  child B gene[i] = parent A gene[i]
end for
for i = cutpoint2 to gene length
  child A gene[i] = parent A gene[i]
  child B gene[i] = parent B gene[i]
end for
return children
End two-point crossover

```

Gambar 3.6 Pseudocode Two Point Crossover



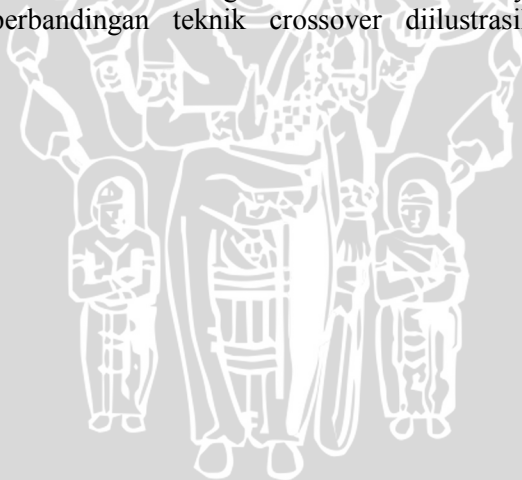
### 3. Uniform Crossover

```
Begin uniform crossover
for i = 0 to gene length
for x = 0 to arrJumcutpoint
if cekExist = false then
    child A gene[i] = parent A gene[i]
    child B gene[i] = parent B gene[i]
else
    child A gene[i] = parent B gene[i]
    child B gene[i] = parent A gene[i]
end for
return children
End uniform crossover
```

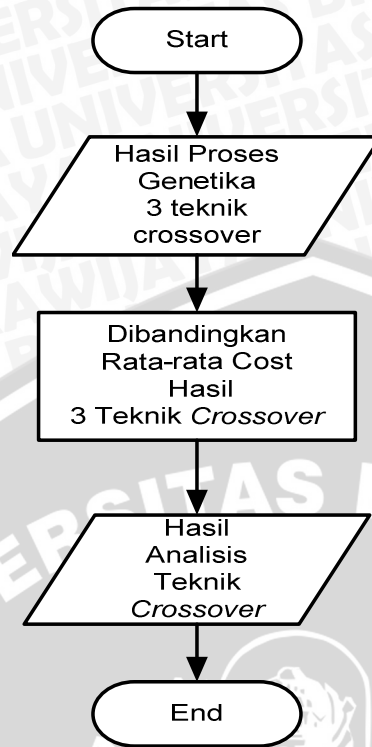
Gambar 3.7 Pseudocode Uniform Crossover

### 3.5 Perancangan Proses Perbandingan Teknik *Crossover*.

Untuk proses perbandingan teknik *crossover* dilakukan setelah proses genetika dijalankan. Perbandingan dilakukan pada ukuran populasi serta probabilitas *crossover* yang bervariasi. Perbandingan teknik *crossover* ini akan melihat nilai rata-rata *cost* yang dihasilkan dimana teknik *crossover* yang lebih baik adalah teknik *crossover* yang menghasilkan individu dengan nilai rata-rata *cost* yang lebih kecil. Proses perbandingan teknik *crossover* diilustrasikan pada gambar 3.8



### 3.5.1 Flowchart Proses Perbandingan Teknik *Crossover*



Gambar 3.8 Flowchart perbandingan teknik *crossover*

Langkah-langkah spesifik perbandingan dan analisa *crossover* pada permasalahan penjadwalan praktikum dengan algoritma genetika :

1. Menentukan parameter untuk mengukur performance ketiga teknik *crossover*. Parameter yang digunakan yaitu rata-rata nilai *cost* dari proses genetika setelah mencapai hasil konvergen.
2. Membandingkan nilai *cost* yang dihasilkan dari proses genetika dengan uji coba menggunakan parameter genetika yang berbeda.
3. Analisa hasil uji coba dengan memperhatikan pengaruh parameter genetika terhadap *performance* teknik *crossover*.
4. Teknik *crossover* dengan *performance* terbaik yaitu teknik dengan rata-rata *cost* yang paling baik (kecil).

### 3.6 Perancangan Uji Coba

Uji coba yang akan dilakukan untuk mengetahui *performance* dari masing-masing teknik *crossover* dalam menghasilkan solusi terbaik pada proses penjadwalan praktikum dengan algoritma genetika. Pengujian dilakukan dengan memberikan nilai parameter genetika yang berbeda yaitu probabilitas *crossover* dan ukuran populasi. Probabilitas *crossover* yang diberikan yaitu 25%, 50 % dan 75%, sedangkan untuk ukuran populasi yaitu 10,50, 100 dan 250 individu. Percobaan untuk masing-masing perlakuan akan dilakukan sebanyak 5 kali runtime dan dilakukan sampai didapat hasil yang konvergen.

Hasil yang akan dibandingkan yaitu populasi pada saat proses genetika mencapai konvergen, yaitu ketika nilai *fitness* tidak mengalami perubahan atau sudah didapat solusi yang terbaik. Tabel rancangan uji coba dapat dilihat pada tabel 3.16.

Tabel 3.16 Rancangan uji coba

Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform





## BAB IV IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Implementasi Program

Implementasi program penjadwalan praktikum dengan algoritma genetika ini, terdiri dari beberapa proses yaitu inisialisasi kromosom, seleksi, *crossover* (pindah silang) dan mutasi. Penjelasan dari proses-proses tersebut akan di jelaskan pada subbab berikut:

#### 4.1.1 Inisialisasi Kromosom

Komponen kromosom yaitu waktu dan ruang di set terlebih dahulu sebelum inisialisasi kromosom dilakukan. Inisialisasi kromosom dilakukan secara random yaitu dengan membangkitkan nilai hari, waktu dan ruang secara acak.

```
SetKompKrom;
rand:= RandomRange(0,length(hari));
rand:= RandomRange(0,length(waktu));
rand:= RandomRange(0,length(ruang));
```

Nilai random yang didapat disimpan pada variabel :

```
individu[i].Cromosom[j].k_hari:= hari[rand];
individu[i].Cromosom[j].waktu.k_waktu:=
waktu[rand].k_waktu;
individu[i].Cromosom[j].waktu.mulai:=
waktu[rand].mulai;
individu[i].Cromosom[j].waktu.selesai:=
waktu[rand].selesai;
individu[i].Cromosom[j].k_ruang:= ruang[rand];
```

Untuk *cost* serta *fitness* masing-masing individu disimpan pada variabel :

```
individu[i].Cost:= GetCost(individu[i].Cromosom);
individu[i].Fitness:= 1/(individu[i].Cost+1); .
```

#### 4.1.2 Seleksi

Proses seleksi dilakukan dengan terlebih dahulu membangkitkan nilai random antara 0 s/d 99 sebanyak jumlah individu pada populasi. Nilai yang didapat dibandingkan dengan nilai probabilitas kumulatif populasi kemudian dikalikan 100 untuk mencari range probabilitas seleksi tiap individu.

Untuk memilih individu, dibangkitkan nilai random antara 1s/d 99 kemudian dicek nilai tersebut ada populasi untuk mencari individu dengan range yang sesuai.

Listing program untuk proses seleksi :

```
for i:= 0 to jumParent-1 do
Randomize;
parental[i]:= range[RandomRange(0,length(range)-1)];
pos:= 0;
for j:= 0 to Length(range) - 1 do
begin
if range[j] <> parental[i] then
begin
range[pos]:= range[j];
pos:= pos+1
end;
end;
SetLength(range,pos+1);
```

#### 4.1.3 Crossover (pindah silang)

Setelah proses seleksi selesai dilakukan, proses selanjutnya adalah *crossover*. Sebelumnya, titik potong (*cutpoint*) ditentukan terlebih dahulu. Pada penelitian ini akan diuji tiga teknik *crossover* sehingga titik potong untuk masing-masing teknik *crossover* pun akan berbeda.

Titik potong ditentukan secara random dari titik 1 sampai panjang kromosom dikurangi 1. Individu hasil *crossover* merupakan kombinasi dari potongan kedua *parent* yang dipertukarkan. Listing program untuk masing-masing teknik *crossover* adalah sebagai berikut :

##### a. One Point Crossover

Untuk *one point crossover* ditentukan satu titik potong , dimana penentuannya dijelaskan pada listing berikut :



```

if RGMetCrossover.ItemIndex = 0 then
  begin
    Randomize;
    cutpoin:=
    RandomRange(1,length(Individu[0].Cromosom)-1);
  end

```

Untuk proses *crossover*-nya dijelaskan pada listing berikut ini.

```

if RGMetCrossOver.ItemIndex = 0 then
  begin
    for j:= 0 to cutpoin do
      begin
        child.Cromosom[j]:=individu[Parental[i]].Cromosom[j];
      end;

      if i < jumParent-1 then
        begin
          for j:=cutpoin+1 to Length(child.Cromosom)-1 do
            begin
              child.Cromosom[j]:= individu[Parental[i+1]].
              Cromosom[j]
            end;
          end
        else
          begin
            for j:=cutpoin+1 to Length(child.Cromosom)-1 do
              begin
                child.Cromosom[j]:=individu[Parental[0]].Cromosom[j];
              end;
            end
          end
        end

```

#### *b. Two Point Crossover*

Untuk *two point crossover* ditentukan dua titik potong sehingga setiap individu akan terbagi menjadi 3 bagian yang akan dipertukarkan. Penentuan titik potong dijelaskan pada listing berikut :

```

if RGMetCrossover.ItemIndex = 1 then
  begin
    Randomize;
    cutpoin:=
    RandomRange(1,length(Individu[0].Cromosom)-1);

    cutpoin1:= cutpoin;
    repeat

```

```

Randomize;
cutpoint:=
RandomRange(1,length(Individu[0].Cromosom)-1);
until (cutpoint <> cutpoint1) and (Abs(cutpoint-
cutpoint1)>2);

```

Untuk proses *crossover*nya dijelaskan pada listing berikut ini.

```

if RGMetCrossover.ItemIndex = 1 then
begin
//potongan pertama pada parent 1
for j:= 0 to cutpoint do
begin
child.Cromosom[j]:=individu[Parental[i]].Cromosom[j];
end;

//potongan kedua pada parent 2
if i < jumParent-1 then
begin
for j:=cutpoint+1 to cutpoint1 do
begin
child.Cromosom[j]:=
individu[Parental[i+1]].Cromosom[j];
end;
end
else
begin
for j:=cutpoint+1 to cutpoint1 do
begin
child.Cromosom[j]:=
individu[Parental[0]].Cromosom[j];
end;
end;

//potongan ketiga pada parent 1
for j:= cutpoint1+1 to Length(child.Cromosom)-1 do
begin
child.Cromosom[j]:=
individu[Parental[i]].Cromosom[j];
end;
end

```

### c. Uniform Crossover

Untuk uniform *crossover*, titik potong ditentukan secara random dengan titik potong lebih dari 3. Potongan bagian individu yang dipertikarkan sesuai dengan hasil pemotongan. Penentuan titik potong dijelaskan pada listing berikut :

```

if RGMetCrossover.ItemIndex = 2 then
  begin
  setlength(arrJumCutPoint, RandomRange(2, length(Individu
[0].Cromosom));
    SetLength(range, length(individu[0].Cromosom));
    for j:= 0 to Length(range)-1 do
      range[j]:= j;

    cnt:= length(range);
    for j:= 0 to Length(arrJumCutPoint)-1 do
    begin
      rand:= RandomRange(0, cnt);
      arrJumCutPoint[j]:= range[rand];
      range[rand]:= range[cnt-1];
      cnt:= cnt-1;
    end;
  end;

```

Untuk proses *crossover*nya dijelaskan pada listing berikut ini.

```

if RGMetCrossover.ItemIndex = 2 then
  begin
  for j:= 0 to Length(child.Cromosom)-1 do
  begin
    cekExist:= false;
    for x:= 0 to Length(arrJumCutPoint)-1 do
    begin
      if j <> arrJumCutPoint[x] then
        cekExist:= false
      else
        cekExist:= true;
      end;
      if cekExist = false then

```



```

        child.Cromosom[j]:=
        individu[Parental[i]].Cromosom[j];
    end;

    if i < jumParent-1 then
    begin
        for j:=0 to Length(child.Cromosom)-1 do
        begin
            cekExist:= false;
            for x:= 0 to Length(arrJumCutPoint)-1 do
            begin
                if j <> arrJumCutPoint[x] then
                cekExist:= false
                else
                cekExist:= true;
            end;

            if cekExist = true then
            child.Cromosom[j]:=
            individu[Parental[i+1]].Cromosom[j];
            end;

        end
        else
        begin
            for j:=0 to Length(child.Cromosom)-1 do
            begin
                cekExist:= false;
                for x:= 0 to Length(arrJumCutPoint)-1 do
                begin
                    if j <> arrJumCutPoint[x] then
                    cekExist:= false
                    else
                    cekExist:= true;
                end;

                if cekExist = true then
                child.Cromosom[j]:=
                individu[Parental[0]].Cromosom[j];
            end;
        end;
    end;

```

#### 4.1.4 Mutasi

Proses pertama sebelum mutasi dilakukan adalah memilih individu yang akan dimutasi. Pada sistem ini, pemilihan individu

yang akan dimutasi dilakukan secara random. Listing penentuan individu untuk dimutasi dijelaskan berikut ini.

```
lenIndividu:= length(Individu);
for i:= 0 to jumMutasi-1 do
begin
  Randomize;
  rand:= RandomRange(0,StrToInt(spJumIndividu.Text)-1);
```

Setelah didapatkan individu yang dimutasi, Posisi mutasi ditentukan secara random. Jika posisi mutasi 1 sama dengan posisi mutasi 2, maka akan dicari posisi mutasi yang baru secara random. Proses pencarian posisi mutasi secara lengkap dijelaskan pada listing program berikut :

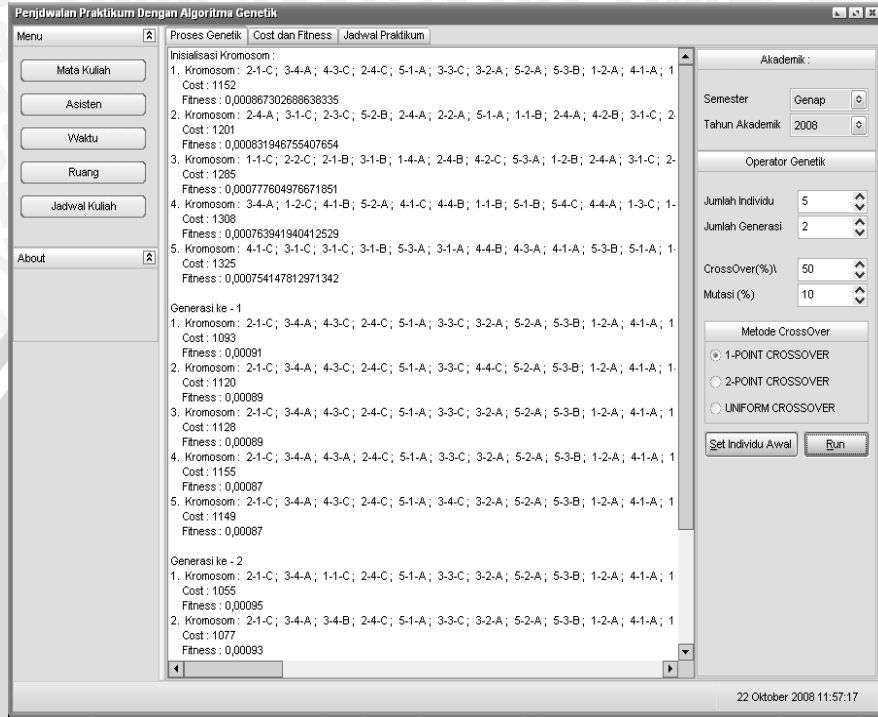
```
subCrom1:=
RandomRange(0,length(Individu[0].Cromosom)-1);
repeat
begin
  subCrom2:=
  RandomRange(0,length(Individu[0].Cromosom)-1);
end;
until(subCrom1<>subCrom2);
```

Proses mutasi dilakukan dengan mengganti dua gen yang dimutasi dengan nilai baru yang didapat secara random. Listing program untuk proses mutasi sebagai berikut.

```
rand:= RandomRange(0,length(hari));
mutant.Cromosom[subCrom1].k_hari:= hari[rand];
rand:= RandomRange(0,length(waktu));
mutant.Cromosom[subCrom1].waktu:= waktu[rand];
rand:= RandomRange(0,length(ruang));
mutant.Cromosom[subCrom1].k_ruang:= ruang[rand];

rand:= RandomRange(0,length(hari));
mutant.Cromosom[subCrom2].k_hari:= hari[rand];
rand:= RandomRange(0,length(waktu));
mutant.Cromosom[subCrom2].waktu:= waktu[rand];
rand:= RandomRange(0,length(ruang));
mutant.Cromosom[subCrom2].k_ruang:= ruang[rand];
```

## 4.2 Implementasi Antarmuka



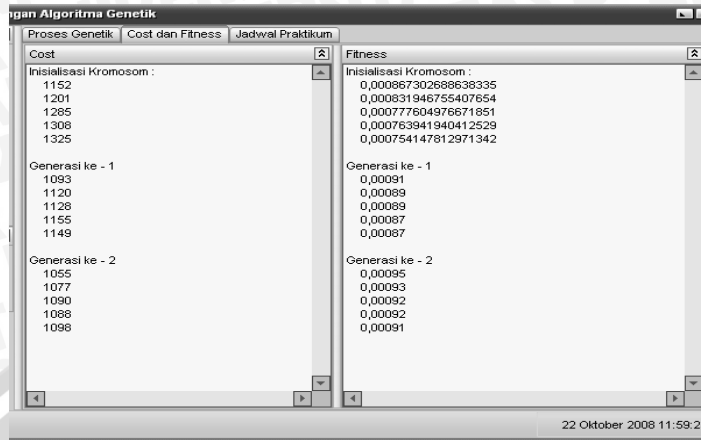
Gambar 4.1 Antarmuka Proses Genetika

Proses genetika seperti pada gambar 4.1 diatas, menghasilkan individu ( kromosom ) yang merupakan kombinasi dari hari, waktu dan ruang yang dihasilkan secara random. Jumlah individu sesuai dengan jumlah individu yang ditentukan, dan iterasi akan berhenti sebanyak jumlah generasi yang telah ditentukan pula.

Setelah iterasi berhenti, individu atau kromosom yang dihasilkan akan mengalami perubahan dari inisialisasi awal. Nilai *cost* dan *fitness* juga mengalami perubahan seiring dengan generasi yang

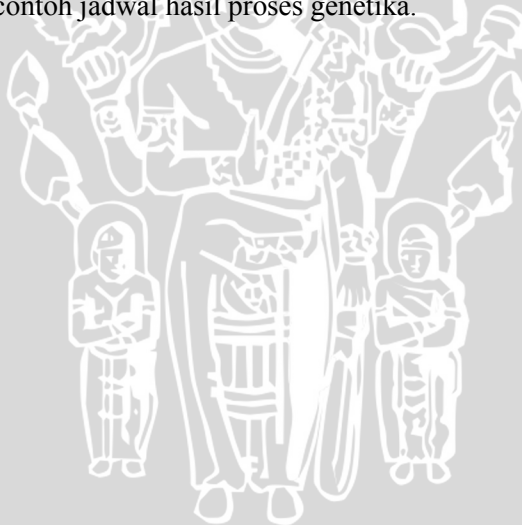


dihasilkan. Generasi terakhir akan mempunyai nilai *cost* yang semakin serta nilai *fitness* yang semakin besar. Gambar 4.2 menunjukkan individu serta nilai costnya hasil proses genetika.



Gambar 4.2 Antarmuka nilai *cost* dan *fitness*

Jadwal mata kuliah diperoleh dari kromosom yang memiliki nilai *fitness* terbaik pada iterasi terakhir. Secara umum, jumlah generasi berpengaruh dalam menentukan nilai *fitness*. Semakin banyak generasi maka nilai *fitness* yang diperoleh akan semakin baik. Hal tersebut dikarenakan pada setiap generasi dilakukan operasi genetika yaitu perkawinan silang dan mutasi. Gambar 4.3 menunjukkan contoh jadwal hasil proses genetika.



Kode MK	Nama MK	Program Studi	Hari	Mulai
MAS 4233	Analisis Multivariat	Statistik	Senin	07.00
MAI 4270	Struktur Data & Algoritma	Statistik	Senin	07.00
MAM 4433	Metode Numerik	Statistik	Senin	07.00
MAS 4221	Metode Statistika II	Statistik	Senin	07.00
MAI 4480	Algo. Dan Pemrograman	Matematika	Senin	08.00
MAS 4480	Statistika Dasar	Matematika	Senin	08.00
MAI 4270	Struktur Data & Algoritma	Statistik	Senin	08.00
MAI 4480	Algo. Dan Pemrograman	Matematika	Senin	08.00
MAS 4480	Statistika Dasar	Matematika	Senin	08.00
MAM 4433	Metode Numerik	Statistik	Senin	08.00
MAI 4180	Komputer Dasar	Kimia	Senin	08.00
MAS 4221	Metode Statistika II	Statistik	Senin	08.00
MAS 4480	Statistika Dasar	Matematika	Senin	08.00

Gambar 4.3 Antarmuka Jadwal Praktikum

### 4.3 Implementasi Uji Coba

Aplikasi yang sudah dibuat digunakan untuk melakukan perbandingan teknik *crossover* yang diterapkan pada penjadwalan praktikum dengan algoritma genetika.

Penerapan aplikasi dilaksanakan dengan memasukkan data penjadwalan semester genap jurusan matematika tahun ajaran 2007/2008. Adapun data yang dimasukkan adalah sebagai berikut:

Tabel 4.1 Data uji coba penjadwalan

Mata Kuliah	22
Ruang	3
Kelas Mata kuliah	62
Jumlah hari	5
Jumlah periode	4

Tabel 4.1 diatas menunjukkan bahwa terdapat 63 kelas praktikum yang harus dijadwalkan pada 3 ruang, selama 5 hari dimana dalam 1 hari terdapat 5 periode.

Tabel 4.2 Kombinasi *default* parameter genetika

Generasi	5
Populasi	10
Probabilitas <i>Crossover</i>	0.5
Probabilitas Mutasi	0.1

Pada Tabel 4.2 dijelaskan bahwa jumlah generasi atau iterasi yang dihasilkan adalah 5 generasi. Setiap generasi menghasilkan 10 populasi atau 10 kromosom.

Uji coba akan dilakukan dalam dua tahap. Tahap pertama dilakukan dengan mengganti nilai dari parameter genetika, yaitu probabilitas *crossover*. Tahap kedua dari uji coba dilakukan dengan mengganti parameter genetika yaitu ukuran populasi. Kedua uji coba dilakukan untuk mengetahui pengaruh kedua parameter genetika pada *crossover* dalam menghasilkan individu yang terbaik. Adapun variasi parameter genetika tersebut yaitu :

- Probabilitas *Crossover* : 25%, 50%, 75%
- Ukuran Populasi : 10, 100, 250
- Generasi : 10, 50, 100, 200, 500

#### 4.4 Hasil Uji Coba

##### 4.4.1 Hasil Uji Coba Variasi Probabilitas *Crossover*

Uji Coba dengan variasi probabilitas *crossover* dilakukan dengan mengganti nilai probabilitas *crossover* untuk setiap proses genetika yang dijalankan. Uji coba ini dilakukan dengan jumlah individu 100, serta nilai probabilitas mutasi 10%.

Berikut hasil dari uji coba variasi probabilitas *crossover* :

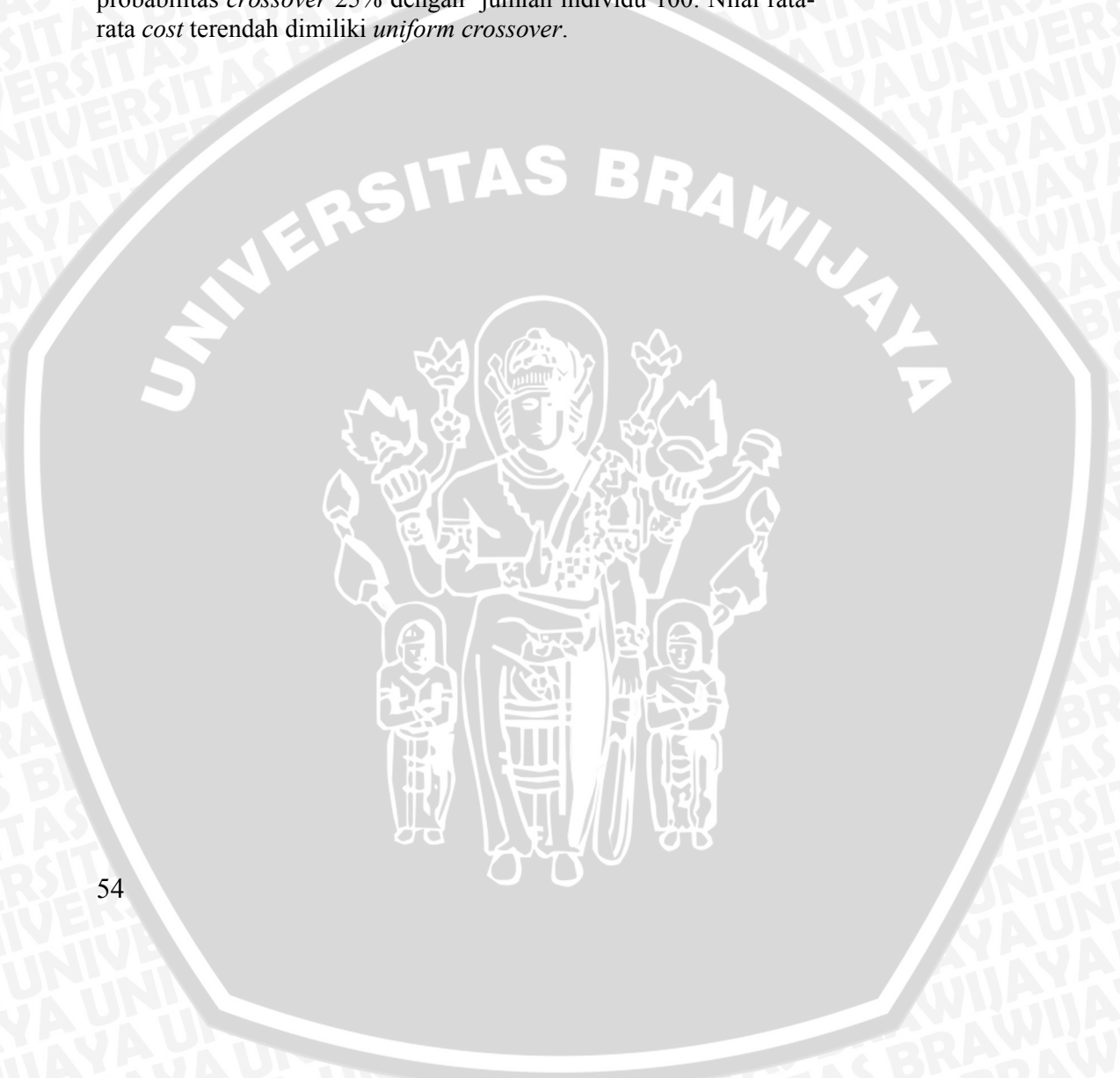
Tabel 4.3 Rata-rata *Cost* Hasil Uji Coba dengan Probabilitas 25%

Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform
1	1002,90	1013,16	1026,88



2	736,46	744,54	768,58
3	622,24	628,11	671,16
4	558,05	568,52	579,19
5	492,36	514,40	512,15
Rata-Rata	682,40	693,75	711,59

Tabel 4.3 menunjukkan bahwa nilai rata-rata *cost* dari *one point* lebih baik dibandingkan *two point* dan *uniform* pada nilai probabilitas *crossover* 25% dengan jumlah individu 100. Nilai rata-rata *cost* terendah dimiliki *uniform crossover*.



Tabel 4.4 Rata-rata *Cost* Hasil Uji Coba dengan Probabilitas 50%

Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform
1	494,93	497,78	504,77
2	491,02	493,62	517,88
3	504,29	493,92	508,46
4	492,32	493,93	532,96
5	490,76	490,49	506,29
Rata-Rata	494,66	493,95	514,07

Tabel 4.4 menunjukkan bahwa nilai rata-rata *cost* dari *one point* lebih baik dibandingkan *two point* dan *uniform* pada nilai probabilitas *crossover* 50% dengan jumlah individu 100. Nilai rata-rata *cost* terendah dimiliki *uniform crossover*.

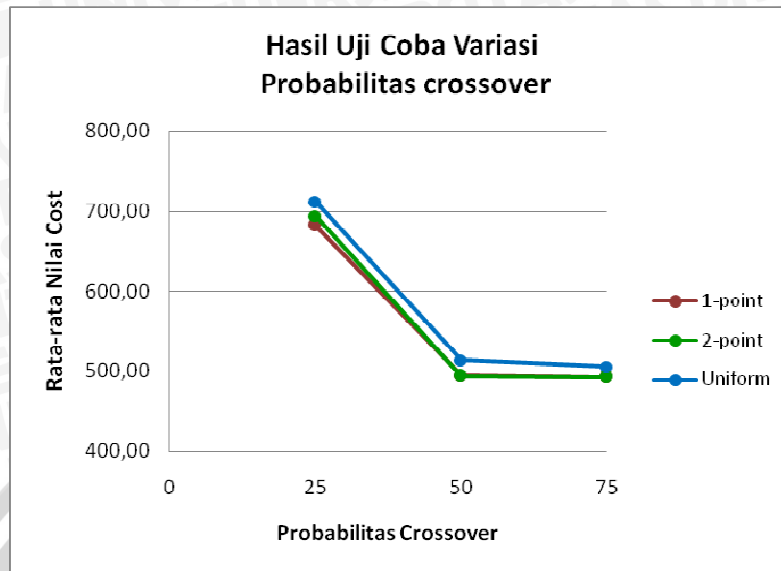
Tabel 4.5 Rata-rata *Cost* Hasil Uji Coba dengan Probabilitas 75%

Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform
1	492,29	486,83	505,17
2	488,49	490,22	504,95
3	493,41	490,66	515,5
4	500,65	521,84	503,7
5	486,65	495,99	497,91
Rata-Rata	492,3	497,11	505,45

Tabel 4.5 menunjukkan bahwa nilai rata-rata *cost* dari *one point* tetap lebih baik dibandingkan *two point* dan *uniform* pada nilai probabilitas *crossover* 75% dengan jumlah individu 100. Nilai rata-rata *cost* terendah dimiliki *uniform crossover*.

Grafik pada gambar 4.4 menunjukkan perbandingan nilai rata-rata *cost* yang dihasilkan ketiga teknik *crossover* pada uji coba variasi probabilitas *crossover*. Rata-rata nilai *cost* yang ditampilkan pada grafik yaitu nilai *cost* pada keadaan konvergen. Grafik

menunjukkan bahwa rata-rata nilai *cost* yang dihasilkan *one point crossover* paling rendah diantara ketiga teknik *crossover*.



Gambar 4.4 Grafik perbandingan rata-rata *cost* dengan probabilitas *crossover* berbeda.

#### 4.4.2 Hasil Uji Coba Variasi Ukuran Populasi

Uji Coba dengan variasi ukuran populasi dilakukan dengan mengganti nilai ukuran populasi untuk setiap proses genetika yang dijalankan. Uji coba ini dilakukan dengan nilai probabilitas *crossover* 75%, serta nilai probabilitas mutasi 10%.

Berikut hasil dari uji coba variasi probabilitas *crossover* :



Tabel 4.6 Rata-rata *Cost* Hasil Uji Coba Ukuran Populasi 10

Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform
1	529,73	526,02	532,13
2	532,05	522,21	539,47
3	515,16	535,49	535,3
4	524,12	517,94	534,54
5	518,84	516,61	541,86
Rata-rata	523,98	523,65	536,66

Tabel 4.6 diatas menunjukkan nilai rata-rata dari *two point* lebih baik dibandingkan *one point* dan *uniform*. *Uniform Crossover* mempunyai *performance* terendah diantara ketiga teknik *crossover*.

Tabel 4.7 Rata-rata *Cost* Hasil Uji Coba Ukuran Populasi 50

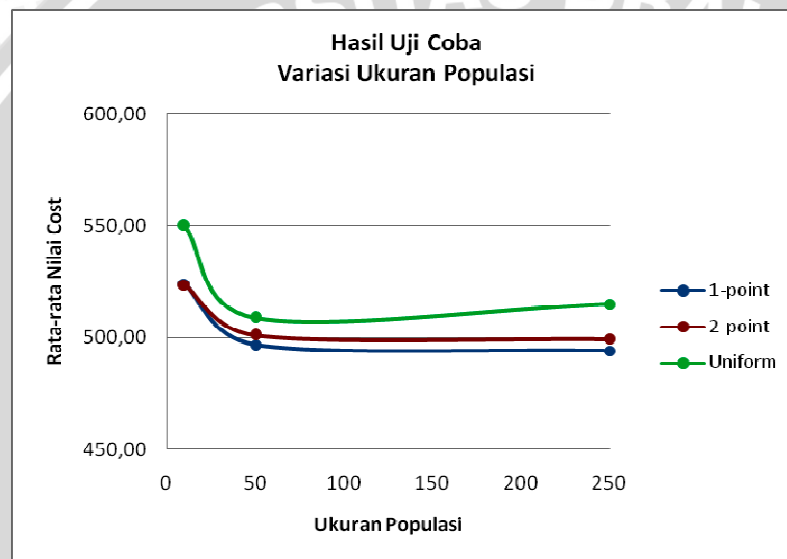
Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform
1	487,44	491,93	532,03
2	499,81	497,18	505,6
3	497,72	526,38	506,77
4	497,82	498,77	505,39
5	495,37	491,06	501,64
Rata-rata	495,63	501,06	510,29

Tabel 4.7 diatas menunjukkan nilai rata-rata *cost* dari *one point* lebih baik dibandingkan *two point* dan *uniform*. Nilai rata-rata *cost* terendah dimiliki *uniform crossover*.

Tabel 4.8 Rata-rata *Cost* Hasil Uji Coba Ukuran Populasi 250

Runtime	Teknik <i>Crossover</i>		
	1-point	2-point	Uniform
1	492,45	506,12	511,46
2	494,23	498,61	518,09
3	497,99	492,12	515,34
4	492,34	496,17	515,83
5	494,14	504,11	514,04
Rata-rata	494,23	499,43	514,95

Tabel 4.8 diatas menunjukkan nilai rata-rata *cost* dari *one point* lebih baik dibandingkan *two point* dan *uniform*. *Uniform crossover* tetap memiliki rata-rata *cost* terendah seperti uji coba lainnya.



Gambar 4.5 Grafik perbandingan rata-rata *cost* dengan ukuran populasi berbeda.

#### 4.5 Analisis Hasil

*Crossover* atau pindah silang mempunyai pengaruh besar pada proses algoritma genetika karena individu baru dihasilkan dari proses

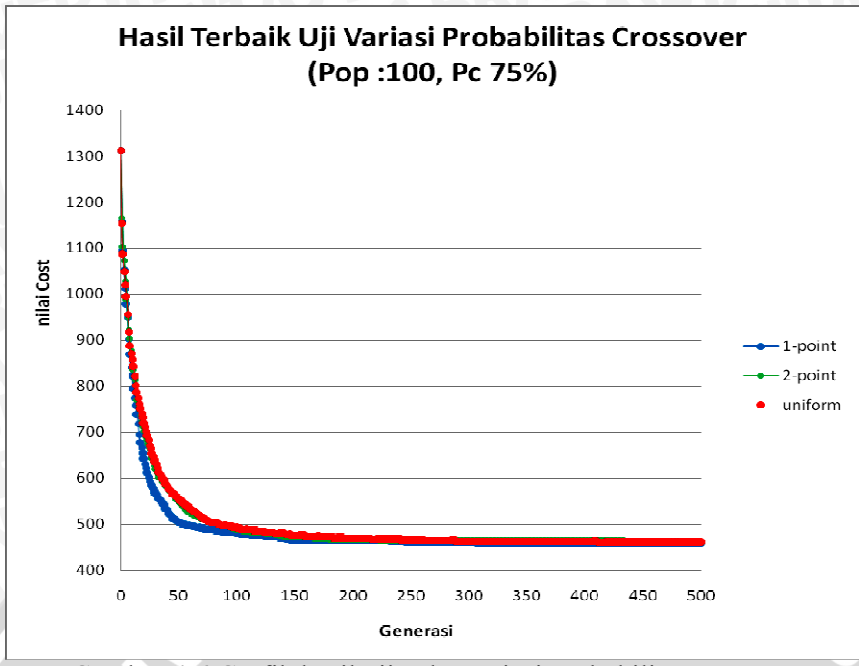
*crossover* (pindah silang). *Crossover* dipengaruhi oleh probabilitas *crossover* ( $P_c$ ). Jika  $n$  adalah banyaknya individu pada populasi, sebanyak  $P_c \times n$  individu akan mengalami pindah silang. Semakin besar nilai  $P_c$ , semakin cepat individu baru muncul dalam populasi. *Crossover* juga dipengaruhi oleh ukuran populasi. Ukuran populasi menentukan ruang pencarian dari algoritma genetika serta produktivitas *crossover*. Jika ukuran populasi terlalu kecil, ruang pencarian solusi semakin sempit serta memerlukan waktu yang relatif lama untuk menemukan solusi optimal. Disisi lain, ukuran populasi yang terlalu besar seringkali variasi genetiknya kurang karena luasnya ruang pencarian serta memerlukan iterasi yang panjang (jumlah generasi besar).

Tipe dari teknik *crossover* menentukan bagaimana informasi dari induk dikombinasikan ke anak. Perbedaan *one point*, *two point* dan *uniform crossover* pada jumlah titik potong yang akan berpengaruh terhadap jumlah segmen individu yang dipertukarkan. *Two point* dan *uniform crossover*, kombinasi *parent* untuk menghasilkan anak lebih beragam daripada *one point crossover*. Hal ini berpengaruh pada kemampuan *crossover* dalam meningkatkan keragaman populasi, yaitu menghasilkan anak yang berbeda dari induk. Pada *uniform crossover*, terdapat kerugian yaitu jika segmen yang dipertukarkan terlalu besar maka keragaman genetika terlalu besar sehingga proses pencarian akan sulit dalam menemukan solusi yang akurat. Sebaliknya, jika segmen yang dipertukarkan kecil maka keragaman genetika yang dihasilkan rendah sehingga pencarian tidak menyeluruh pada ruang pencarian solusi.

Hasil dari uji coba sebelumnya, seperti pada gambar 4.4, didapatkan bahwa pada variasi probabilitas *crossover*, *one point crossover* mempunyai rata-rata *cost* yang lebih baik dibandingkan *two point* dan *uniform crossover*. *Uniform crossover* mempunyai rata-rata *cost* terendah diantara ketika teknik *crossover*.

Grafik pada gambar 4.6 menunjukkan nilai rata-rata *cost* tiga teknik *crossover* pada generasi dimana hasil proses genetika konvergen. Uji coba dilakukan pada probabilitas 75% dengan jumlah individu 100. Pada probabilitas 75%, dapat dilihat bahwa rata-rata *cost* yang dihasilkan terbaik dibandingkan probabilitas 25% dan 50%.

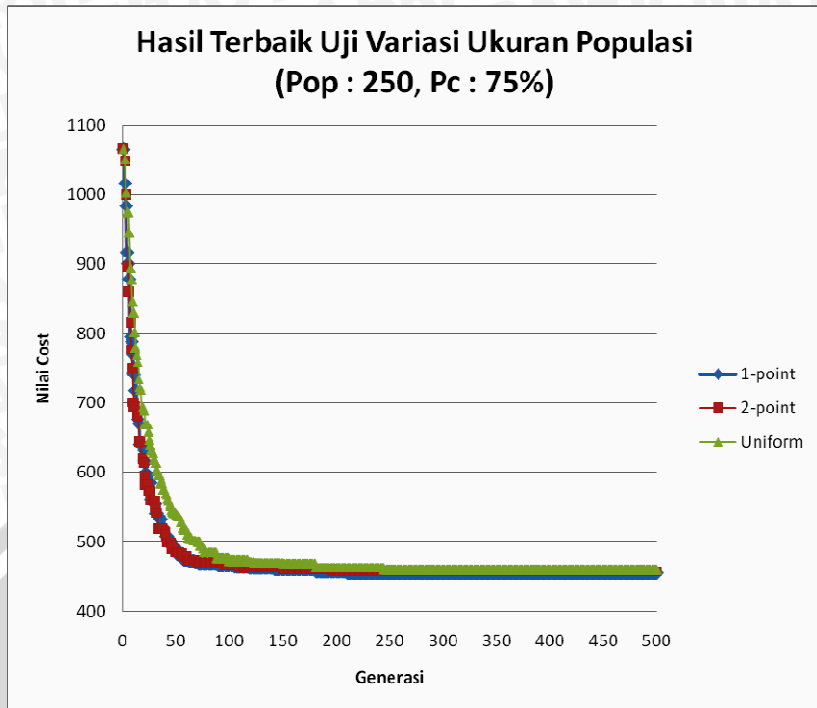




Gambar 4.6 Grafik hasil uji coba variasi probabilitas crossover

Uji coba tahap dua untuk mengetahui pengaruh dari ukuran populasi terhadap kinerja dari crossover. Hasil uji coba, seperti pada gambar 4.5, pada ukuran populasi 10, 50, dan 250 menghasilkan bahwa *one point* lebih baik dibandingkan *two point* dan *uniform crossover*. Keunggulan *one point* hampir dominan seperti pada uji coba dengan variasi probabilitas crossover. Hal ini terlihat pada *performance one point* yang melebihi *two point* dan *uniform crossover*. Dari grafik juga terlihat, garis *one point* pada generasi ketika proses genetika mencapai konvergen, nilai costnya terendah diantara ketiga teknik crossover.





Gambar 4.7 Grafik terbaik uji variasi ukuran populasi

Pada permasalahan penjadwalan praktikum dengan algoritma genetika, *one point crossover* lebih baik dibandingkan *two point crossover* maupun *uniform crossover*. Hasil tersebut terjadi pada kedua uji coba baik dengan variasi probabilitas *crossover* maupun ukuran populasi. Keunggulan *performance one point* pada dua uji coba yang telah dilakukan pada iterasi akhir tidak terlalu besar. Hasil ini dapat dilihat pada gambar 4.6 dan 4.7 dimana pada grafik garis ketiga teknik *crossover* yang berhimpit.

Pada nilai probabilitas *crossover* 75%, sebanyak  $0.75 \times$  jumlah individu akan mengalami *crossover* sehingga kemungkinan dihasilkan individu baru cukup besar. Nilai probabilitas juga berpengaruh dalam meningkatkan keragaman genetika, sehingga jika semakin besar nilai probabilitas *crossover* maka kemungkinan keragaman genetiknya bertambah semakin besar. Besarnya segmen yang dipertukarkan serta lingkungan dengan probabilitas *crossover*

75% berakibat pada rendahnya *performance uniform crossover*. *Uniform crossover* menghasilkan anakan yang berbeda jauh dari induk sehingga menyebabkan sulitnya mencari solusi yang optimal.

Ukuran populasi yang besar cenderung menghasilkan solusi yang lebih baik namun tidak selalu sesuai untuk semua teknik crossover. *One point* dan *two point crossover* lebih sesuai pada ukuran populasi besar, dibandingkan *uniform crossover* yang lebih sesuai pada ukuran populasi kecil atau terbatas.

Hasil dari proses genetika dipengaruhi oleh parameter serta operator genetika. Hasil proses genetika akan lebih baik jika kombinasi antara parameter genetika dan operator genetika sesuai. Pada penelitian yang telah dilakukan kombinasi probabilitas mutasi yang tinggi (10%) dan ukuran populasi yang besar lebih sesuai untuk teknik *crossover one point* dan *two point*. Hasil *performance* keduanya mampu mengungguli *uniform crossover*.

*One point crossover* mendominasi keseluruhan uji coba karena segmen individu yang ditukar pada level rendah, yaitu 1 dan 2. Level yang rendah menghasilkan anakan yang tidak jauh berbeda dari induk yang berpengaruh pada pencarian solusi terbaik. Level rendah juga lebih menguntungkan jika diterapkan pada permasalahan penjadwalan praktikum karena sulitnya mencari kombinasi yang pas antara hari, waktu dan ruang untuk menghasilkan jadwal yang optimal.







## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan, maka dapat diambil kesimpulan antara lain :

1. Implementasi algoritma genetika mampu menyelesaikan permasalahan penjadwalan praktikum, dimana jadwal praktikum diperoleh dari individu dengan nilai fitness terbaik.
2. *One point crossover* mendominasi uji coba dengan variasi probabilitas *crossover* serta uji coba dengan variasi ukuran populasi.
3. *One point* dan *two point crossover* mempunyai *performance* lebih baik dibandingkan *uniform crossover*, karena kombinasi parameter genetika yang sesuai untuk kedua teknik *crossover* pada uji proses genetika yang dilakukan.
4. *Uniform crossover* mempunyai *performance* terendah karena teknik ini segmen individu yang ditukar cenderung besar sehingga mengakibatkan sulitnya menemukan solusi yang optimal.

### 5.2 Saran

Saran yang dapat diberikan setelah pengerjaan penelitian ini adalah :

1. Uji coba dilakukan dalam ruang lingkup yang lebih luas lagi. Misalnya Permasalahan genetika dengan karakter berbeda (TSP, MST, Clustering, dan permasalahan optimasi lainnya), ukuran populasi yang lebih beragam, dan jumlah generasi yang lebih besar untuk memperoleh kesimpulan yang lebih akurat.
2. Pada penelitian ini hanya diuji satu teknik mutasi dan satu probabilitas mutasi, sehingga untuk kedepannya perlu dianalisa teknik mutasi lain dan nilai probabilitas yang lebih beragam sehingga didapat kombinasi lain agar hasil proses genetika semakin baik.





## DAFTAR PUSTAKA

- Baluja, S. dan Caruana, R. 1995. Removing the Genetics from the Standard Genetic Algorithm. *Proceedings of the Twelfth International Conference on Machine Learning*. Lake Tahoe, CA.
- Basuki, Achmad. 2003. Strategi Menggunakan Algoritma Genetika. ITS. Surabaya.
- De Jong, K. A. dan Spears, W. 1990. An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms, *Proceedings of the First Int'l Conf. on Parallel Problem Solving from Nature*, H. P. Schwefel and R. Manner (eds.), Springer-Verlag.
- Coax, Earl. 2005. Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration. E-book. Morgan Kaufman Publishing. USA
- Ghaemi, Sehraneh. 2003. Using Genetic Algorithm Tool to Solve University Timetable Scheduling Problem. *Journal faculty of electrician and computer engineering*. University of Tabriz.Iran
- Kazarlis, Spyros. 2005. Solving University Timetabling Problems Using Advanced Genetic Algorithms. Technological Educational Institute of Serres. Yunani
- Kusumadewi, Sri. 2003. Artificial Inteligence (Teknik dan Aplikasinya). Graha Ilmu. Yogyakarta
- Mawaddah, Nia K. 2006. Optimasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetik. Minor thesis. Universitas Brawijaya. Malang
- Noertjahyana, A. dan Yulia. 2003. Studi Analisa Pelatihan Jaringan Syaraf Tiruan Dengan Dan Tanpa Algoritma Genetika. Universitas Kristen Petra. Jakarta.
- Nurjaya, W. 2002. Analisis Proses Word Matching Problem Menggunakan Algoritma Genetika. Universitas Komputer Indonesia. Bandung.
- Potter, W.D , R.W. Robinson, J.A Miller, K.J Kochut dan D. Z Redys . 1993. Using The Genetic Algorithm to FindSnake-In-The-Box Codes. Univ. of Georgia. Georgia

Siregar, Ramdhan. H. 2004. Optimisasi Biaya Bahan Bakar Pembangkit Dalam Sistem Tenaga Listrik Dengan Menggunakan Kecerdasan Algoritma Genetik. Jurusan Teknik Elektro Fakultas Teknik. Universitas Syiah Kuala. Nangroe Aceh Darussalam. 2004

Solis, Juan F, J.M Vargas, M. Larre, J.L.G Ramos. 2002. Genetic Algorithm with Forced Diversity for the University TimeTabling Problem. Universitas Juárez Autonoma de Tabasco. Meksiko.

Spears, W. M. & Anand, V. 1991. A study of crossover operators in genetic programming. *Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems*, 409-418. Charlotte, NC: Springer-Verlag.

Triputra, Rein N. 2006. Pemecahan Masalah Rute Terpendek Pada Agen Travel Menggunakan Metode Hierarchical Clustering Dan Algoritma Genetika. Minor thesis. Universitas Brawijaya. Malang.

Widhiyasa, Arief. 2007. Kajian Genetic Algorithm Dalam Penyelesaian TSP. Institut Teknologi Bandung. Bandung







Lampiran 1 Jadwal praktikum dari individu terbaik hasil proses genetika pada uji coba variasi probabilitas *crossover*. Uji coba dilakukan dengan *one point crossover*, probabilitas *crossover* 75%, probabilitas mutasi 10% dan ukuran populasi 100 individu.

Hari	Jam	Mata Kuliah	Program Studi	Kelas	Ruang / Lab
Senin	07.30 - 09.30	Komputer Dasar	Fisika	C	C
Senin	07.30 - 09.30	Komputer Dasar	Fisika	A	B
Senin	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	B	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	B	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	A	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	D	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	E	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	B	A
Senin	11.30 - 13.00	Komputer Dasar	Fisika	B	C
Senin	13.00 - 15.00	Komputer Dasar	Kimia	D	A
Senin	13.00 - 15.00	Komputer Dasar	Fisika	D	C
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	C	A
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	D	C
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	A	B

Selasa	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	A	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	C	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	F	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	E	C
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	A	C
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	C	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	D	A
Selasa	11.30 - 13.00	Statistika Dasar	Biologi	C	B
Selasa	11.30 - 13.00	Statistika Dasar	Fisika	C	C
Selasa	13.00 - 15.00	Statistika Dasar	Biologi	B	B
Rabu	07.30 - 09.30	Komputer Dasar	Kimia	B	A
Rabu	07.30 - 09.30	Komputer Dasar	Fisika	A	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	B	A
Rabu	09.30 - 11.30	Metode Numerik	Matematika	B	B
Rabu	09.30 - 11.30	Metode Numerik	Matematika	C	A
Rabu	09.30 - 11.30	Metode Numerik	Matematika	A	B
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	B	C
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	A	C
Rabu	11.30 - 13.00	Statistika Dasar	Kimia	C	A

Rabu	13.00 - 15.00	Statistika Dasar	Kimia	B	A
Rabu	13.00 - 15.00	Statistika Dasar	Fisika	B	B
Kamis	07.30 - 09.30	Struktur Data & Algoritma	Statistika	C	C
Kamis	07.30 - 09.30	Analisis regresi lanjutan	Statistika	A	A
Kamis	07.30 - 09.30	Statistika Dasar	Kimia	A	B
Kamis	09.30 - 11.30	Struktur Data & Algoritma	Statistika	B	B
Kamis	09.30 - 11.30	Metode Numerik	Statistika	A	A
Kamis	09.30 - 11.30	Metode Numerik	Statistika	C	C
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	A	B
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	B	A
Kamis	09.30 - 11.30	Analisis Multivariat	Statistika	C	C
Kamis	11.30 - 13.00	Statistika Dasar	Fisika	A	B
Kamis	13.00 - 15.00	Statistika Dasar	Biologi	B	A
Kamis	13.00 - 15.00	Statistika Dasar	Kimia	D	C
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	A	B
Jumat	07.30 - 09.30	Metode Numerik	Statistika	B	A
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	D	B
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	C	A
Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	A	C



Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	B	C
Jumat	09.30 - 11.30	Basis Data	Matematika	A	C
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	E	B
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	F	B
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	E	A
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	C	C
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	D	A
Jumat	11.30 - 13.00	Riset Operasi II	Statistika	A	C
Jumat	11.30 - 13.00	Statistika Dasar	Kimia	E	B
Jumat	13.00 - 15.00	Komputer Dasar	Kimia	A	C
Jumat	13.00 - 15.00	Komputer Dasar	Kimia	C	A

Lampiran 2 Jadwal praktikum dari individu terbaik hasil proses genetika pada uji coba variasi probabilitas *crossover*. Uji coba dilakukan dengan *Two point crossover*, probabilitas *crossover* 75%, probabilitas mutasi 10% dan ukuran populasi 100 individu.

Hari	Jam	Mata Kuliah	Program Studi	Kelas	Ruang / Lab
Senin	07.30 - 09.30	Statistika Dasar	Biologi	C	A
Senin	07.30 - 09.30	Statistika Dasar	Kimia	C	B
Senin	07.30 - 09.30	Analisis Multivariat	Statistika	B	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	B	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	C	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	A	A
Senin	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	B	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	A	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	D	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	F	B
Senin	11.30 - 13.00	Komputer Dasar	Fisika	B	C
Senin	11.30 - 13.00	Statistika Dasar	Biologi	B	B
Senin	13.00 - 15.00	Statistika Dasar	Fisika	B	C
Senin	13.00 - 15.00	Statistika Dasar	Kimia	D	B

Selasa	07.30 - 09.30	Metode Numerik	Matematika	B	A
Selasa	07.30 - 09.30	Statistika Dasar	Matematika	A	C
Selasa	07.30 - 09.30	Metode Numerik	Matematika	C	B
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	E	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	D	C
Selasa	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	A	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	B	B
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	C	C
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	E	A
Selasa	13.00 - 15.00	Komputer Dasar	Fisika	D	B
Selasa	13.00 - 15.00	Komputer Dasar	Kimia	A	A
Rabu	07.30 - 09.30	Komputer Dasar	Fisika	C	C
Rabu	07.30 - 09.30	Statistika Dasar	Kimia	E	B
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	B	B
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	C	C
Rabu	09.30 - 11.30	Metode Numerik	Matematika	A	B
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	D	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	B	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	A	C



Rabu	11.30 - 13.00	Komputer Dasar	Kimia	D	A
Rabu	11.30 - 13.00	Komputer Dasar	Kimia	B	B
Rabu	13.00 - 15.00	Statistika Dasar	Kimia	B	A
Kamis	07.30 - 09.30	Statistika Dasar	Fisika	C	A
Kamis	07.30 - 09.30	Analisis Multivariat	Statistika	C	B
Kamis	09.30 - 11.30	Struktur Data & Algoritma	Statistika	B	B
Kamis	09.30 - 11.30	Metode Numerik	Statistika	B	C
Kamis	09.30 - 11.30	Metode Numerik	Statistika	A	B
Kamis	09.30 - 11.30	Struktur Data & Algoritma	Statistika	A	A
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	A	C
Kamis	09.30 - 11.30	Riset Operasi II	Statistika	A	A
Kamis	11.30 - 13.00	Komputer Dasar	Fisika	A	B
Kamis	11.30 - 13.00	Statistika Dasar	Fisika	A	C
Kamis	13.00 - 15.00	Komputer Dasar	Biologi	A	C
Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	A	C
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	D	B
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	C	B
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	B	A
Jumat	07.30 - 09.30	Analisis regresi lanjutan	Statistika	A	C

Jumat	07.30 - 09.30	Metode Statistika II	Statistika	C	A
Jumat	09.30 - 11.30	Basis Data	Matematika	A	C
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	E	A
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	F	A
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	E	B
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	C	B
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	D	C
Jumat	11.30 - 13.00	Metode Numerik	Statistika	C	C
Jumat	13.00 - 15.00	Statistika Dasar	Kimia	A	A
Jumat	13.00 - 15.00	Komputer Dasar	Kimia	C	C



Lampiran 3 Jadwal praktikum dari individu terbaik hasil proses genetika pada uji coba variasi probabilitas *crossover*. Uji coba dilakukan dengan *Uniform crossover*, probabilitas *crossover* 75%, probabilitas mutasi 10% dan ukuran populasi 100 individu.

Hari	Jam	Mata Kuliah	Program Studi	Kelas	Ruang / Lab
Senin	07.30 - 09.30	Statistika Dasar	Fisika	B	B
Senin	07.30 - 09.30	Metode Statistika II	Statistika	C	C
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	E	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	C	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	D	C
Senin	09.30 - 11.30	Statistika Dasar	Matematika	E	B
Senin	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	B	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	A	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	C	B
Senin	11.30 - 13.00	Statistika Dasar	Biologi	C	B
Senin	11.30 - 13.00	Komputer Dasar	Kimia	C	C
Senin	13.00 - 15.00	Komputer Dasar	Fisika	D	C
Senin	13.00 - 15.00	Statistika Dasar	Fisika	A	A
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	A	C



Selasa	07.30 - 09.30	Statistika Dasar	Matematika	C	A
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	C	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	B	C
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	A	A
Selasa	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	A	B
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	B	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	D	C
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	F	A
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	E	B
Selasa	11.30 - 13.00	Statistika Dasar	Kimia	B	D
Selasa	13.00 - 15.00	Statistika Dasar	Biologi	B	B
Rabu	07.30 - 09.30	Komputer Dasar	Fisika	A	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	F	A
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	A	C
Rabu	09.30 - 11.30	Metode Numerik	Matematika	C	A
Rabu	09.30 - 11.30	Metode Numerik	Matematika	A	B
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	B	C
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	A	B
Rabu	11.30 - 13.00	Komputer Dasar	Fisika	B	B

Rabu	11.30 - 13.00	Statistika Dasar	Kimia	A	C
Rabu	13.00 - 15.00	Komputer Dasar	Kimia	D	A
Rabu	13.00 - 15.00	Statistika Dasar	Fisika	C	C
Kamis	07.30 - 09.30	Statistika Dasar	Kimia	E	B
Kamis	07.30 - 09.30	Metode Statistika II	Statistika	A	C
Kamis	07.30 - 09.30	Statistika Dasar	Kimia	B	A
Kamis	09.30 - 11.30	Struktur Data & Algoritma	Statistika	B	C
Kamis	09.30 - 11.30	Analisis Multivariat	Statistika	A	B
Kamis	09.30 - 11.30	Metode Numerik	Statistika	B	A
Kamis	09.30 - 11.30	Metode Numerik	Statistika	A	B
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	B	C
Kamis	09.30 - 11.30	Analisis regresi lanjutan	Statistika	A	A
Kamis	11.30 - 13.00	Statistika Dasar	Biologi	A	A
Kamis	13.00 - 15.00	Komputer Dasar	Biologi	B	C
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	D	A
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	A	C
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	C	C
Jumat	07.30 - 09.30	Riset Operasi II	Statistika	A	B

Jumat	07.30 - 09.30	Metode Numerik	Statistika	C	A
Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	B	B
Jumat	09.30 - 11.30	Metode Numerik	Matematika	B	C
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	E	B
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	B	C
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	D	B
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	D	A
Jumat	09.30 - 11.30	Analisis Multivariat	Statistika	C	A
Jumat	11.30 - 13.00	Statistika Dasar	Kimia	C	A
Jumat	13.00 - 15.00	Komputer Dasar	Fisika	C	A
Jumat	13.00 - 15.00	Komputer Dasar	Kimia	A	C



Lampiran 4 Jadwal praktikum dari individu terbaik hasil proses genetika pada uji coba variasi ukuran populasi. Uji coba dilakukan dengan *one point crossover*, probabilitas *crossover* 75%, probabilitas mutasi 10% dan ukuran populasi 10 individu.

Hari	Jam	Mata Kuliah	Program Studi	Kelas	Ruang / Lab
Senin	07.30 - 09.30	Statistika Dasar	Biologi	B	B
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	E	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	C	A
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	A	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	D	C
Senin	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	B	C
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	A	A
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	F	B
Senin	11.30 - 13.00	Statistika Dasar	Kimia	C	B
Senin	11.30 - 13.00	Statistika Dasar	Kimia	B	A
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	F	B
Selasa	07.30 - 09.30	Statistika Dasar	Matematika	E	A
Selasa	07.30 - 09.30	Statistika Dasar	Matematika	A	C
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	B	A

Selasa	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	A	C
Selasa	09.30 - 11.30	Statistika Dasar	Matematika	D	C
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	B	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	C	B
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	D	A
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	E	B
Selasa	11.30 - 13.00	Statistika Dasar	Fisika	C	B
Selasa	11.30 - 13.00	Statistika Dasar	Biologi	C	C
Selasa	11.30 - 13.00	Komputer Dasar	Fisika	A	A
Selasa	13.00 - 15.00	Komputer Dasar	Fisika	D	B
Selasa	13.00 - 15.00	Statistika Dasar	Fisika	A	A
Rabu	07.30 - 09.30	Komputer Dasar	Fisika	C	A
Rabu	07.30 - 09.30	Statistika Dasar	Kimia	E	C
Rabu	09.30 - 11.30	Basis Data	Matematika	A	B
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	C	B
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	C	C
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	D	A
Rabu	09.30 - 11.30	Metode Numerik	Statistika	C	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	A	C

Rabu	11.30 - 13.00	Statistika Dasar	Kimia	A	B
Rabu	11.30 - 13.00	Komputer Dasar	Kimia	B	A
Rabu	11.30 - 13.00	Statistika Dasar	Biologi	A	C
Rabu	13.00 - 15.00	Komputer Dasar	Fisika	B	C
Kamis	07.30 - 09.30	Statistika Dasar	Fisika	B	A
Kamis	07.30 - 09.30	Analisis regresi lanjutan	Statistika	A	C
Kamis	09.30 - 11.30	Analisis Multivariat	Statistika	A	A
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	D	C
Kamis	09.30 - 11.30	Metode Numerik	Statistika	B	B
Kamis	09.30 - 11.30	Metode Numerik	Statistika	A	B
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	A	A
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	C	C
Kamis	11.30 - 13.00	Komputer Dasar	Kimia	D	C
Kamis	13.00 - 15.00	Komputer Dasar	Kimia	A	B
Kamis	13.00 - 15.00	Komputer Dasar	Kimia	C	C
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	B	A
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	A	B
Jumat	07.30 - 09.30	Riset Operasi II	Statistika	A	A
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	B	C



Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	B	B
Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	C	C
Jumat	09.30 - 11.30	Metode Numerik	Matematika	B	B
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	E	B
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	B	A
Jumat	09.30 - 11.30	Metode Numerik	Matematika	C	C
Jumat	09.30 - 11.30	Metode Numerik	Matematika	A	C
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	B	A
Jumat	11.30 - 13.00	Struktur Data & Algoritma	Statistika	C	B
Jumat	11.30 - 13.00	Statistika Dasar	Kimia	D	A

Lampiran 5 Jadwal praktikum dari individu terbaik hasil proses genetika pada uji coba variasi ukuran populasi. Uji coba dilakukan dengan *Two point crossover*, probabilitas *crossover* 75%, probabilitas mutasi 10% dan ukuran populasi 50 individu.

Hari	Jam	Mata Kuliah	Program Studi	Kelas	Ruang/ Lab
Senin	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	B	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	A	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	C	C
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	D	A
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	E	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	B	C
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	D	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	E	B
Senin	09.30 - 11.30	Metode Numerik	Matematika	A	B
Senin	11.30 - 13.00	Statistika Dasar	Biologi	C	A
Senin	13.00 - 15.00	Komputer Dasar	Fisika	B	C
Senin	13.00 - 15.00	Statistika Dasar	Biologi	B	B
Senin	13.00 - 15.00	Statistika Dasar	Fisika	B	A
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	A	A

Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	B	B
Selasa	07.30 - 09.30	Statistika Dasar	Matematika	B	C
Selasa	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	A	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	B	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	F	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	A	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	C	C
Selasa	09.30 - 11.30	Statistika Dasar	Matematika	C	C
Selasa	11.30 - 13.00	Komputer Dasar	Kimia	A	C
Selasa	11.30 - 13.00	Statistika Dasar	Fisika	A	B
Selasa	13.00 - 15.00	Komputer Dasar	Fisika	A	A
Selasa	13.00 - 15.00	Statistika Dasar	Kimia	D	C
Rabu	07.30 - 09.30	Komputer Dasar	Fisika	C	C
Rabu	07.30 - 09.30	Statistika Dasar	Biologi	A	A
Rabu	09.30 - 11.30	Basis Data	Matematika	A	B
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	C	C
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	E	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	F	B
Rabu	09.30 - 11.30	Metode Numerik	Matematika	C	A



Rabu	09.30 - 11.30	Statistika Dasar	Matematika	E	C
Rabu	11.30 - 13.00	Komputer Dasar	Fisika	D	C
Rabu	11.30 - 13.00	Statistika Dasar	Kimia	A	A
Rabu	11.30 - 13.00	Statistika Dasar	Kimia	B	B
Kamis	07.30 - 09.30	Riset Operasi II	Statistika	A	A
Kamis	09.30 - 11.30	Metode Numerik	Statistika	A	B
Kamis	09.30 - 11.30	Metode Numerik	Statistika	B	B
Kamis	09.30 - 11.30	Metode Numerik	Statistika	C	A
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	C	C
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	D	C
Kamis	09.30 - 11.30	Analisis Multivariat	Statistika	C	A
Kamis	11.30 - 13.00	Komputer Dasar	Kimia	B	C
Kamis	11.30 - 13.00	Statistika Dasar	Kimia	E	A
Kamis	13.00 - 15.00	Komputer Dasar	Kimia	C	B
Kamis	13.00 - 15.00	Komputer Dasar	Kimia	D	C
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	A	A
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	C	A
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	A	C
Jumat	07.30 - 09.30	Metode Statistika II	Statistika	B	C

Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	A	B
Jumat	07.30 - 09.30	Analisis regresi lanjutan	Statistika	A	B
Jumat	09.30 - 11.30	Struktur Data & Algoritma	Statistika	B	A
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	D	C
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	B	A
Jumat	09.30 - 11.30	Analisis Multivariat	Statistika	B	B
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	A	C
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	D	B
Jumat	11.30 - 13.00	Statistika Dasar	Fisika	C	B
Jumat	13.00 - 15.00	Statistika Dasar	Kimia	C	C



Lampiran 6 Jadwal praktikum dari individu terbaik hasil proses genetika pada uji coba variasi ukuran populasi. Uji coba dilakukan dengan *Uniform crossover*, probabilitas *crossover* 75%, probabilitas mutasi 10% dan ukuran populasi 250 individu.

Hari	Jam	Mata Kuliah	Program Studi	Kelas	Ruang / Lab
Senin	07.30 - 09.30	Statistika Dasar	Kimia	B	B
Senin	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	B	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	C	B
Senin	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	E	C
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	B	C
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	D	A
Senin	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	E	A
Senin	09.30 - 11.30	Metode Statistika II	Statistika	C	C
Senin	11.30 - 13.00	Statistika Dasar	Kimia	D	A
Senin	13.00 - 15.00	Statistika Dasar	Biologi	A	A
Senin	13.00 - 15.00	Statistika Dasar	Kimia	A	C
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	D	B
Selasa	07.30 - 09.30	Algo. Dan Pemrograman	Matematika	F	C
Selasa	07.30 - 09.30	Metode Numerik	Matematika	C	A



Selasa	09.30 - 11.30	Prak. Man. Jaringan	Ilmu Komputer	A	B
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	A	C
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	B	C
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	D	A
Selasa	09.30 - 11.30	Prak. Pemrograman II	Ilmu Komputer	F	A
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	A	B
Selasa	09.30 - 11.30	Prak. Basis Data I	Ilmu Komputer	C	A
Selasa	09.30 - 11.30	Statistika Dasar	Matematika	C	C
Selasa	11.30 - 13.00	Komputer Dasar	Kimia	C	B
Selasa	11.30 - 13.00	Statistika Dasar	Kimia	E	C
Selasa	13.00 - 15.00	Komputer Dasar	Fisika	A	C
Selasa	13.00 - 15.00	Komputer Dasar	Fisika	C	A
Selasa	13.00 - 15.00	Statistika Dasar	Biologi	B	B
Rabu	07.30 - 09.30	Statistika Dasar	Fisika	C	B
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	B	C
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	C	A
Rabu	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	E	C
Rabu	09.30 - 11.30	Metode Numerik	Matematika	B	A
Rabu	09.30 - 11.30	Statistika Dasar	Matematika	A	B

Rabu	09.30 - 11.30	Statistika Dasar	Matematika	E	B
Rabu	11.30 - 13.00	Komputer Dasar	Fisika	D	B
Rabu	13.00 - 15.00	Komputer Dasar	Fisika	B	C
Kamis	07.30 - 09.30	Statistika Dasar	Biologi	C	B
Kamis	07.30 - 09.30	Analisis regresi lanjutan	Statistika	A	C
Kamis	07.30 - 09.30	Statistika Dasar	Fisika	A	A
Kamis	09.30 - 11.30	Struktur Data & Algoritma	Statistika	A	C
Kamis	09.30 - 11.30	Struktur Data & Algoritma	Statistika	B	C
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	A	A
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	B	B
Kamis	09.30 - 11.30	Metode Statistika II	Statistika	D	A
Kamis	09.30 - 11.30	Analisis Multivariat	Statistika	A	B
Kamis	11.30 - 13.00	Komputer Dasar	Kimia	A	B
Kamis	11.30 - 13.00	Komputer Dasar	Kimia	D	C
Kamis	13.00 - 15.00	Komputer Dasar	Kimia	B	C
Kamis	13.00 - 15.00	Statistika Dasar	Kimia	C	A
Kamis	13.00 - 15.00	Statistika Dasar	Fisika	B	B
Jumat	07.30 - 09.30	Struktur Data & Algoritma	Statistika	C	B
Jumat	07.30 - 09.30	Metode Numerik	Statistika	A	A

Jumat	07.30 - 09.30	Metode Numerik	Statistika	B	C
Jumat	07.30 - 09.30	Metode Numerik	Statistika	C	C
Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	B	A
Jumat	07.30 - 09.30	Analisis Multivariat	Statistika	C	B
Jumat	09.30 - 11.30	Basis Data	Matematika	A	A
Jumat	09.30 - 11.30	Algo. Dan Pemrograman	Matematika	A	B
Jumat	09.30 - 11.30	Metode Numerik	Matematika	A	C
Jumat	09.30 - 11.30	Riset Operasi II	Statistika	A	B
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	B	A
Jumat	09.30 - 11.30	Statistika Dasar	Matematika	D	C







