

**IMPLEMENTASI ALGORITMA GENETIKA DALAM  
PENJADWALAN PERKULIAHAN DAN PRAKTIKUM  
(STUDI KASUS : FASILKOM-TI DAN FMIPA USU)**

**SKRIPSI**

**YAKHDI PERARI PINEM  
131421088**



**PROGRAM STUDI EKSTENSI S1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2015**

**IMPLEMENTASI ALGORITMA GENETIKA DALAM  
PENJADWALAN PERKULIAHAN DAN PRAKTIKUM  
(STUDI KASUS : FASILKOM-TI DAN FMIPA USU)**

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat  
memperoleh ijazah Sarjana Ilmu Komputer**

**YAKHDI PERARI PINEM  
131421088**



**PROGRAM STUDI EKSTENSI S1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2015**

**PERSETUJUAN**

Judul : IMPLEMENTASI ALGORITMA GENETIKA DALAM  
PENJADWALAN PERKULIAHAN DAN  
PRAKTIKUM  
(STUDI KASUS : FASILKOM-TI DAN FMIPA USU)

Kategori : SKRIPSI

Nama : YAKHDI PERARI PINEM

Nomor Induk Mahasiswa : 131421088

Program Studi : EKSTENSI S1 ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA

Diluluskan di  
Medan, 29 Agustus 2015

Komisi Pembimbing :  
Pembimbing 2

Pembimbing 1

Dian Rachmawati, S.Si., M.Kom  
NIP. 19830723 200912 2 004

Dr. Poltak Sihombing, M.Kom  
NIP. 19620317 199103 1 001

Diketahui/Disetujui oleh  
Program Studi S1 Ilmu Komputer  
Ketua,

Dr. Poltak Sihombing, M.Kom.  
NIP. 19620317 199103 1 001

## **PERNYATAAN**

### **IMPLEMENTASI ALGORITMA GENETIKA DALAM PENJADWALAN PERKULIAHAN DAN PRAKTIKUM (STUDI KASUS : FASILKOM-TI DAN FMIPA USU)**

## **SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil kerja saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 29 Agustus 2015

Yakhdi Perari Pinem  
NIM. 131421088

## PENGHARGAAN

Puji Syukur ke hadirat Allah Subhanahu Wa Ta'ala atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian dan penyusunan skripsi yang berjudul ” Implementasi Algoritma Genetika dalam Penjadwalan Perkuliahan dan Praktikum (Studi Kasus : Fasilkom-TI dan FMIPA USU)”. Tulisan ini masih jauh dari sempurna, hal ini disebabkan oleh keterbatasan dan kemampuan penulis.

Dalam menyelesaikan skripsi ini penulis banyak mendapatkan masukan dan dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Muhammad Zarlis selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
2. Bapak Dr. Poltak Sihombing, M.Kom selaku Ketua Program Studi S1 Ilmu Komputer Universitas Sumatera Utara dan Dosen Pembimbing I yang telah banyak memberikan ilmu, bimbingan, saran, dan masukan kepada penulis dalam pengerjaan skripsi ini.
3. Ibu Maya Silvi Lydia, B.Sc., M.Sc selaku Sekretaris Program Studi S1 Ilmu Komputer Universitas Sumatera Utara.
4. Ibu Dian Rachmawati, S.Si., M.Kom selaku Dosen Pembimbing II yang telah memberikan ilmu, bimbingan, saran, dan masukan kepada penulis dalam pengerjaan skripsi ini.
5. Bapak Drs. Agus Salim Harahap, M.Si. selaku Dosen Pembimbing I yang telah memberikan kritik dan saran dalam penyempurnaan skripsi ini.
6. Bapak M. Andri Budiman, ST,M.Comp.Sc., MEM selaku Dosen Pembimbing II yang telah memberikan kritik dan saran dalam penyempurnaan skripsi ini.
7. Seluruh staf pengajar dan pegawai di Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
8. Teristimewa kepada kedua orang tua penulis yang selalu senantiasa mencurahkan kasih sayang serta doa kepada penulis.
9. Abang dan Adik penulis, Ikhsan Perdana Pinem dan Makhrai Pehulisa Pinem, yang selalu memberi semangat kepada penulis.

10. Kawan-kawan seperjuangan Leli Dahliana, Yohana Sitepu dan Sherly Melisa Sembiring yang selalu memberi semangat kepada penulis dalam mengerjakan skripsinya.
11. Sahabatku Yudho Murphy Harahap dan Dara Juwita yang selalu memberi semangat kepada penulis dalam mengerjakan skripsinya.
12. Teman-teman Ekstensi S1 Ilmu Komputer stambuk 2013 tanpa terkecuali, terima kasih buat kebersamaannya di kala suka maupun duka.
13. Serta semua pihak yang telah ikut membantu penulis namun tidak tercantum namanya.

Penulis menyadari bahwa dalam penulisan maupun penyajian dalam tulisan ini masih jauh dari kesempurnaan. Oleh karena itu dengan segala kerendahan hati penulis menerima kritik dan saran yang bersifat membangun. Kiranya skripsi ini dapat memberi manfaat bagi para pembaca.

Medan, 29 Agustus 2015

Penulis,

Yakhdi Perari Pinem

## ABSTRAK

Penjadwalan kuliah dan praktikum merupakan salah satu kegiatan dalam universitas yang membutuhkan banyak waktu untuk menyelesaikannya. Penjadwalan umumnya masih dilakukan dengan cara manual. Cara ini memiliki keakuratan yang kurang baik dan menyebabkan terjadinya jadwal bentrok. Jadwal yang baik adalah jadwal yang memperhatikan semua komponen dasar penyusunnya. Oleh karena itu, penelitian ini bertujuan untuk menciptakan sebuah penjadwalan perkuliahan dan praktikum yang baik. Algoritma yang digunakan dalam penelitian ini adalah algoritma genetika. Algoritma genetika adalah algoritma pencarian heuristik yang menggunakan prinsip dasar seleksi alam sesuai proses evolusi biologis. Proses yang terjadi dalam algoritma genetika adalah inisialisasi populasi, evaluasi *fitness*, seleksi, *crossover* dan mutasi. Hasil dari aplikasi penjadwalan perkuliahan dan praktikum akan didapat setelah salah satu syarat berhenti terpenuhi, berupa sebuah jadwal mengajar untuk setiap dosen yang terdaftar di Fakultas Ilmu Komputer dan Teknologi Informasi dengan mempertimbangkan jadwal dosen di Fakultas Matematika dan Ilmu Pengetahuan Alam.

Kata Kunci : **optimisasi, heuristik, genetika, penjadwalan, *wheel roulette*, *crossover* satu titik, mutasi *swap***

**IMPLEMENTATION OF GENETIC ALGORITHM  
IN SCHEDULING LECTURES AND PRACTICUM  
(CASE STUDY: FASILKOM-TI AND FMIPA USU)**

**ABSTRACT**

At universities, scheduling a lecture and practicum is one of the tasks that take a lot of time to be completed. Scheduling in general is still done manually. This method had a poor accuracy and could cause a clashing schedule. A good schedule notices all the basic components. Therefore, the purpose of this study is to create a good scheduling system. The algorithm that used on this study is the genetic algorithm. The genetic algorithm is a heuristic search algorithm that mimics the process of natural evolution, such as population initialization, fitness evaluation, selection, crossover and mutation. The results of scheduling lectures and practicum applications will be obtained after one of the stop conditions being met, in the form of a teaching schedule for each lecturer enrolled in the Faculty of Computer Science and Information Technology by considering the schedule of a lecturer at the Faculty of Mathematics and Natural Sciences.

**Keywords : optimization, heuristics, genetic, scheduling, roulette wheel, one-point crossover, swap mutation.**



## DAFTAR ISI

	<b>Halaman</b>
Persetujuan	ii
Pernyataan	iii
Penghargaan	iv
Abstrak	vi
<i>Abstract</i>	vii
Daftar Isi	viii
Daftar Tabel	xi
Daftar Gambar	xiii

### **BAB 1 PENDAHULUAN**

1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	6

### **BAB 2 LANDASAN TEORI**

2.1 Optimisasi	8
2.1.1 Klasifikasi optimisasi	8
2.2 Penjadwalan	10
2.3 Algoritma Genetika	10
2.3.1 Struktur dasar algoritma genetika	12
2.3.2 Syarat berhenti dan parameter algoritma genetika	13
2.3.3 Teknik pengkodean	15
2.3.4 Evaluasi <i>fitness</i>	18
2.3.5 Operator algoritma genetika	18
2.3.6 <i>Schema</i> algoritma genetika	25
2.3.7 Kelebihan dan kelemahan algoritma genetika	27

### **BAB 3 ANALISIS DAN PERANCANGAN SISTEM**

3.1	Batasan-Batasan pada Penjadwalan	28
3.1.1	Batasan pokok penjadwalan ( <i>hard constraints</i> )	28
3.1.2	Batasan tambahan penjadwalan ( <i>soft constraints</i> )	29
3.2	Analisis Sistem	29
3.3	Komponen Utama Penjadwalan	30
3.4	Proses Algoritma Genetika pada Penjadwalan	53
3.4.1	Membangkitkan populasi awal	53
3.4.2	Evaluasi <i>fitness</i>	58
3.4.3	Seleksi	61
3.4.4	<i>Crossover</i>	62
3.4.5	Mutasi	63
3.5	Perancangan Sistem	63
3.5.1	Rancangan data	64
3.5.2	Rancangan arsitektural	67
3.5.3	Rancangan prosedural	73
3.5.4	Rancangan interface	76

### **BAB 4 IMPLEMENTASI DAN PENGUJIAN**

4.1	Implementasi Sistem	86
4.1.1	Form splash screen	86
4.1.2	Form login	87
4.1.3	Form lupa password	88
4.1.4	Form menu utama	88
4.1.5	Form program studi	89
4.1.6	Form input dosen	90
4.1.7	Form input jadwal sibuk	90
4.1.8	Form input kelas	91
4.1.9	Form input ruangan	92
4.1.10	Form input matakuliah	92
4.1.11	Form input hari dan waktu	93
4.1.12	Form algoritma genetika	94
4.2	Pengujian Sistem	95

**BAB 5 KESIMPULAN DAN SARAN**

5.1 Kesimpulan 97

5.2 Saran 97

**DAFTAR PUSTAKA 98**

**LAMPIRAN-A A-1**

**LAMPIRAN-B**

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 2.1	Contoh Pengkodean Biner 15
Tabel 2.2	Contoh Pengkodean Nilai 16
Tabel 2.3	Contoh Pengkodean Permutasi 18
Tabel 3.1	Daftar Dosen dan Asisten pada Program Studi Ilmu Komputer 30
Tabel 3.2	Daftar Dosen dan Asisten pada Program Studi Teknologi Informasi 32
Tabel 3.3	Daftar Hari Masuk Perkuliahan dan Praktikum di Fasilkom-TI 33
Tabel 3.4	Daftar Waktu Masuk Perkuliahan dan Praktikum di Fasilkom-TI 34
Tabel 3.5	Daftar Ruangan pada Program Studi Ilmu Komputer 34
Tabel 3.6	Daftar Ruangan pada Program Studi Teknologi Informasi 35
Tabel 3.7	Daftar Kelas pada Program Studi Ilmu Komputer 35
Tabel 3.8	Daftar Kelas pada Program Studi Teknologi Informasi 36
Tabel 3.9	Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer 37
Tabel 3.10	Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi 45
Tabel 3.11	Representasi Sebuah Individu pada Prodi Ilmu Komputer 54
Tabel 3.12	Representasi Sebuah Individu pada Prodi Teknologi Informasi 55
Tabel 3.13	Inisialisasi Populasi pada Program Studi Ilmu Komputer 57
Tabel 3.14	Inisialisasi Populasi pada Program Studi Teknologi Informasi 58
Tabel 3.15	Evaluasi Fitness pada Individu-1 59
Tabel 3.16	Evaluasi Fitness pada Individu-2 60
Tabel 3.17	Evaluasi Fitness pada Individu-3 60
Tabel 3.18	Persentase Fitness pada Prodi Ilmu Komputer 61
Tabel 3.19	Struktur Tabel Dosen 64
Tabel 3.20	Struktur Tabel Hari 64
Tabel 3.21	Struktur Tabel Kelas 65
Tabel 3.22	Struktur Tabel Jadwal 65
Tabel 3.23	Struktur Tabel Matkul 65
Tabel 3.24	Struktur Tabel Ruang 66

Tabel 3.25	Struktur Tabel Sibuk	66
Tabel 3.26	Struktur Tabel Waktu	66
Tabel 3.27	Kamus Data Penjadwalan	72
Tabel 4.1	Pengujian Algoritma Genetika dalam Penjadwalan Perkuliahan Dan Praktikum	95

## DAFTAR GAMBAR

		Halaman
Gambar 2.1	Diagran Optimisasi Fungsi atau Optimisasi Proses	8
Gambar 2.2	Struktur Dasar Penerapan Algoritma Genetika untuk Menyelesaikan Suatu Masalah Optimisasi	12
Gambar 2.3	Pengkodean Pohon Tipe Edge Encoding	16
Gambar 2.4	Pengkodean Pohon Tipe Vertex Encoding	17
Gambar 2.5	Seleksi Metode Roulette Wheel	19
Gambar 2.6	Seleksi Metode Ranking	19
Gambar 2.7	Penyilangan Satu Titik	21
Gambar 2.8	Penyilangan Dua Titik	21
Gambar 2.9	Penyilangan PMX	22
Gambar 2.10	Penyilangan OX	22
Gambar 2.11	Mutasi Pembalikan	23
Gambar 2.12	Mutasi Penyisipan	24
Gambar 2.13	Mutasi Pemindahan	24
Gambar 2.14	Mutasi Penukaran	25
Gambar 2.15	Mutasi Penggantian	25
Gambar 2.16	Contoh Penggunaan Algoritma Genetika	26
Gambar 3.1	Representasi Kromosom	53
Gambar 3.2	Probabilitas Seleksi Roulette Wheel	62
Gambar 3.3	Ilustrasi Crossover Satu Titik	62
Gambar 3.4	Ilustrasi Swap Mutation	63
Gambar 3.5	Skema Relasi Database Penjadwalan	67
Gambar 3.6	Diagram Konteks	68
Gambar 3.7	DFD Level 1	69
Gambar 3.8	DFD Level 2 Proses 2	70
Gambar 3.9	DFD Level 2 Proses 3	71
Gambar 3.10	Flowchart algoritma genetika	74
Gambar 3.11	Flowchart Sistem Penjadwalan	75
Gambar 3.12	Rancangan Tampilan Splash Screen	77
Gambar 3.13	Rancangan Tampilan Form Login	77

Gambar 3.14	Rancangan Tampilan Form Lupa Password	78
Gambar 3.15	Rancangan Tampilan Menu Utama	79
Gambar 3.16	Rancangan Tampilan Pilih Program Studi	79
Gambar 3.17	Rancangan Tampilan Form Isian Dosen	80
Gambar 3.18	Rancangan Tampilan Form Isian Jadwal Sibuk/Tak Sedia	81
Gambar 3.19	Rancangan Tampilan Form Isian Kelas	82
Gambar 3.20	Rancangan Tampilan Form Isian Ruangan	82
Gambar 3.21	Rancangan Tampilan Form Isian Mata Kuliah	83
Gambar 3.22	Rancangan Tampilan Form Isian Hari dan Waktu	84
Gambar 3.23	Rancangan Tampilan Form Genetika	85
Gambar 4.1	Tampilan Splash Screen	87
Gambar 4.2	Tampilan Form Login	87
Gambar 4.3	Tampilan Form Lupa Password	88
Gambar 4.4	Tampilan Menu Utama	89
Gambar 4.5	Tampilan Pilih Program Studi	89
Gambar 4.6	Tampilan Form Isian Dosen	90
Gambar 4.7	Tampilan Form Isian Jadwal Sibuk/Tak Sedia	91
Gambar 4.8	Tampilan Form Isian Kelas	91
Gambar 4.9	Tampilan Form Isian Ruangan	92
Gambar 4.10	Tampilan Form Isian Mata Kuliah	93
Gambar 4.11	Tampilan Form Isian Hari dan Waktu	93
Gambar 4.12	Tampilan Form Genetika	94
Gambar 4.13	Tampilan Laporan Algoritma Genetika	95
Gambar 4.14	Grafik perubahan nilai fitness dalam pengujian Algoritma Genetika	96

## ABSTRAK

Penjadwalan kuliah dan praktikum merupakan salah satu kegiatan dalam universitas yang membutuhkan banyak waktu untuk menyelesaikannya. Penjadwalan umumnya masih dilakukan dengan cara manual. Cara ini memiliki keakuratan yang kurang baik dan menyebabkan terjadinya jadwal bentrok. Jadwal yang baik adalah jadwal yang memperhatikan semua komponen dasar penyusunnya. Oleh karena itu, penelitian ini bertujuan untuk menciptakan sebuah penjadwalan perkuliahan dan praktikum yang baik. Algoritma yang digunakan dalam penelitian ini adalah algoritma genetika. Algoritma genetika adalah algoritma pencarian heuristik yang menggunakan prinsip dasar seleksi alam sesuai proses evolusi biologis. Proses yang terjadi dalam algoritma genetika adalah inisialisasi populasi, evaluasi *fitness*, seleksi, *crossover* dan mutasi. Hasil dari aplikasi penjadwalan perkuliahan dan praktikum akan didapat setelah salah satu syarat berhenti terpenuhi, berupa sebuah jadwal mengajar untuk setiap dosen yang terdaftar di Fakultas Ilmu Komputer dan Teknologi Informasi dengan mempertimbangkan jadwal dosen di Fakultas Matematika dan Ilmu Pengetahuan Alam.

Kata Kunci : **optimisasi, heuristik, genetika, penjadwalan, *wheel roulette*, *crossover* satu titik, mutasi *swap***



**IMPLEMENTATION OF GENETIC ALGORITHM  
IN SCHEDULING LECTURES AND PRACTICUM  
(CASE STUDY: FASILKOM-TI AND FMIPA USU)**

**ABSTRACT**

At universities, scheduling a lecture and practicum is one of the tasks that take a lot of time to be completed. Scheduling in general is still done manually. This method had a poor accuracy and could cause a clashing schedule. A good schedule notices all the basic components. Therefore, the purpose of this study is to create a good scheduling system. The algorithm that used on this study is the genetic algorithm. The genetic algorithm is a heuristic search algorithm that mimics the process of natural evolution, such as population initialization, fitness evaluation, selection, crossover and mutation. The results of scheduling lectures and practicum applications will be obtained after one of the stop conditions being met, in the form of a teaching schedule for each lecturer enrolled in the Faculty of Computer Science and Information Technology by considering the schedule of a lecturer at the Faculty of Mathematics and Natural Sciences.

**Keywords : optimization, heuristics, genetic, scheduling, roulette wheel, one-point crossover, swap mutation.**

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Universitas merupakan lembaga ilmiah terdiri atas beberapa fakultas yang menyelenggarakan pendidikan dalam sejumlah disiplin ilmu tertentu. Seiring perkembangan zaman, universitas akan mengalami perubahan dan perkembangan secara fisik dan sistem. Pemisahan dan pendirian fakultas baru merupakan salah satu perubahan yang melanda hampir semua universitas.

Permasalahan pokok yang umumnya dialami fakultas baru adalah keterbatasan jumlah tenaga kerja yang disediakan universitas. Solusi permasalahan yang biasa dilakukan adalah melakukan perekrutan dosen dari fakultas lama. Oleh sebab itu, banyak sekali kegiatan yang harus dijadwalkan untuk mengurangi konflik yang tidak seharusnya terjadi dan efisiensi dari tenaga kerja yang jumlahnya terbatas. Contoh kegiatan tersebut adalah penjadwalan seminar-seminar, penjadwalan ujian pertengahan semester, penjadwalan ujian semester, dan penjadwalan mata kuliah.

Penjadwalan yang diteliti adalah penjadwalan mata kuliah, yang merupakan salah satu penjadwalan utama dalam universitas. Penjadwalan mata kuliah adalah masalah penempatan waktu dan ruangan pada sejumlah mata kuliah, dan kegiatan akademik lainnya seperti praktikum yang berhubungan dengan kapasitas ruang, waktu yang dibutuhkan, dan toleransi untuk ketersediaan dosen. Biasanya penjadwalan dibuat secara manual dengan menggunakan tabel. Hal tersebut dinilai tidak efektif, karena membutuhkan waktu dan pemikiran yang besar dalam merumuskan sebuah jadwal.

Fakultas Ilmu Komputer dan Teknologi Informasi (Fasilkom-TI) Universitas Sumatera Utara dibentuk pada tanggal 6 September 2011 dengan diterbitkannya surat keputusan Rektor USU Nomor 2360/UN5.1.R/SK/PRS/2011. Fasilkom-TI memiliki 3 program studi, yaitu Program Studi S-1 Ilmu Komputer, S-1 Teknologi Informasi dan S-2 Teknik Informatika. Keseluruhan program studi dulunya berada di bawah naungan FMIPA USU. Saat ini perumusan jadwal kegiatan perkuliahan dan praktikum yang terdapat di Fasilkom-TI, hanya terfokus pada dosen yang berasal dari Fasilkom-TI saja. Sehingga, sering ditemukan jadwal dosen berasal FMIPA yang mengalami bentrok dengan jadwal yang ada di Fasilkom-TI.

Inti dari permasalahan adalah bagaimana fakultas dapat menjadwalkan berbagai komponen yang terdiri dari mahasiswa, dosen, mata kuliah, ruang dan waktu dengan memperhatikan sejumlah batasan dan syarat tertentu. Untuk mengatasinya, dibutuhkan sebuah penjadwalan yang baik agar dapat mendistribusikan seluruh komponen secara efisien dan merata serta tidak ada terjadi jadwal bentrok antara satu dengan yang lain. Oleh karena itu, dibutuhkan sebuah metode yang dapat diterapkan untuk menyelesaikan masalah penjadwalan kegiatan perkuliahan secara otomatis dengan bantuan komputer. Dalam skripsi ini, metode yang digunakan adalah Algoritma Genetika, karena waktu yang dibutuhkan relatif lebih singkat bila dibandingkan dengan penjadwalan secara manual.

Algoritma genetika merupakan metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah teori evolusi Charles Darwin. Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland melalui sebuah penelitian. Penelitian tersebut didasarkan atas keinginan untuk menerapkan pemahaman proses evolusi alamiah dalam memecahkan masalah pada sebuah sistem buatan (*Artificial System*). Pendekatan yang dilakukan algoritma genetika adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya, yaitu dengan memaksimalkan kecocokannya atau disebut *fitness*. Generasi ini akan merepresentasikan perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner. Pada akhirnya, akan didapat solusi-solusi yang paling tepat bagi permasalahan yang dihadapi [KOZ92].

Berdasarkan hal tersebut, pada penelitian ini, penulis akan mengembangkan sebuah aplikasi yang bertujuan untuk merumuskan jadwal kegiatan perkuliahan dan praktikum di Fasilkom-TI USU.

## 1.2 Rumusan Masalah

Sesuai dengan latar belakang masalah yang telah diuraikan sebelumnya, maka yang menjadi rumusan masalah penelitian yaitu bagaimana mengoptimalkan perumusan jadwal kegiatan perkuliahan dan praktikum di Fasilkom-TI USU dengan menggunakan algoritma genetika, agar tidak membutuhkan waktu yang lama serta tidak ditemukannya jadwal bentrok terutama dosen yang berasal dari FMIPA USU, sehingga dapat menyelesaikan permasalahan penjadwalan dan memperoleh sistem penjadwalan yang baik.

## 1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah ditetapkan, maka dibuat batasan masalah adalah sebagai berikut:

1. Teknik pengkodean yang digunakan adalah pengkodean Integer dan operator yang digunakan adalah seleksi *roulette wheel*, *crossover* satu titik potong dan *swap mutation*.
2. Variabel yang digunakan yaitu mata kuliah, kelas, sks, hari, waktu, ruangan dan dosen.
3. Ukuran populasi bersifat dinamis, yaitu berdasarkan inputan user.
4. Maksimum generasi yang dilakukan bersifat statis, yaitu 500.
5. Jumlah dan panjang kromosom yang dilakukan bersifat statis.

6. Proses evaluasi fitness menggunakan rumus berikut.

$$Fitness = \frac{1}{1 + (\sum_{i=1}^6 p_i)} \quad (1.1)$$

dimana, p = penalti/pelanggaran batasan oleh kromosom

7. Kriteria penalti dalam penelitian ini adalah
- 1) Setiap pengajar tidak boleh mengajar pada hari dan waktu yang sama.
  - 2) Setiap kelas tidak boleh masuk kuliah/praktikum pada hari dan waktu yang sama.
  - 3) Setiap ruangan tidak boleh digunakan pada hari dan waktu yang sama.
  - 4) Dosen tidak boleh mengajar pada jam sibuk yang telah terdaftar.
  - 5) Perkuliahan harus dilakukan pada ruangan teori (non-lab).
  - 6) Seluruh perkuliahan dan praktikum akan dikosongkan setiap Jumat, mulai pukul 12.10 WIB hingga pukul 13.50 WIB. Hal ini sehubungan pelaksanaan ibadah Sholat Jumat bagi kaum muslimin.
8. Nilai Probabilitas Crossover ( $P_c$ ) dan Probabilitas Mutasi ( $P_m$ ) yang digunakan bersifat dinamis, yaitu berdasarkan inputan user.
9. Studi kasus yang diambil dalam penelitian ini adalah menyelesaikan masalah penjadwalan kuliah dan praktikum pada S1 Teknologi Informasi dan S1 Reguler Ilmu Komputer semester ganjil tahun ajaran 2015/2016 dengan memperhitungkan jadwal dosen yang berasal dari FMIPA USU.
10. Aturan waktu perkuliahan dan praktikum dijadwalkan sebanyak 6 hari, dari hari senin hingga sabtu dimana jadwal masuk perkuliahan dan praktikum dimulai dari jam 08.00 hingga 15.30 WIB dan pada hari jum'at pukul 12.10-13.50 WIB digunakan untuk sholat jum'at.
11. Aplikasi sistem penjadwalan perkuliahan dan praktikum menggunakan Visual Basic 2010 dan Microsoft Access 2013 sebagai *Database Management System* (DBMS).
12. Hasil output dari sistem aplikasi adalah berupa daftar penjadwalan kegiatan perkuliahan dan praktikum yang sudah optimal.

#### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah membangun aplikasi algoritma genetika yang berguna untuk menentukan penjadwalan perkuliahan yang efektif, yaitu terjadinya kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan.

#### **1.5 Manfaat Penelitian**

Manfaat yang didapat dari penelitian ini adalah meminimumkan konflik-konflik dalam penjadwalan kuliah yang dapat menghambat proses belajar-mengajar dan meningkatkan pemahaman tentang penggunaan algoritma genetika dalam memperoleh optimasi penjadwalan serta mengembangkan aplikasi algoritma genetika sebagai salah satu metode optimasi untuk melakukan penjadwalan kuliah.

#### **1.6 Metodologi Penelitian**

Penelitian ini dilakukan dengan menerapkan beberapa metode penelitian sebagai berikut:

a. Studi Literatur

Metode ini dilaksanakan dengan mengumpulkan bahan dan referensi dari buku, skripsi maupun jurnal sumber lain yang berhubungan dengan Optimisasi, Teknik Heuristik, Algoritma Genetika dan Kecerdasan Buatan.

b. Pengumpulan dan Analisis Data

Metode ini dilaksanakan dengan mengumpulkan data-data yang terkait dengan penjadwalan perkuliahan dan praktikum di Fasilkom-TI, seperti data pengajar, data kelas, data waktu perkuliahan, data mata kuliah, data ruangan, dan lain-lain.

c. Analisis dan Perancangan Sistem

Metode ini dimulai dengan tahap mengidentifikasi masalah, mengumpulkan data-data yang berkaitan dengan kriteria, alternatif, memahami kerja sistem yang akan dibuat dan merancang *flowchart* dan DFD (*Data Flow Diagram*) untuk aplikasi penjadwalan kegiatan perkuliahan dan praktikum dengan Algoritma Genetika.

d. Implementasi Sistem.

Metode ini adalah mengimplementasikan rancangan aplikasi penjadwalan kegiatan belajar mengajar dengan Algoritma Genetika ke dalam bahasa pemrograman Visual Basic .NET 2010.

e. Pengujian Sistem.

Metode ini adalah melakukan pengujian terhadap program sehingga diketahui apakah program sudah berjalan dengan benar dan sesuai dengan perancangan yang telah dilakukan.

f. Dokumentasi

Tahap akhir dari penelitian yang dilakukan, yaitu membuat laporan dan kesimpulan akhir dari hasil analisa dan pengujian.

## 1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi menjadi lima bab, yaitu sebagai berikut:

### **BAB 1                    PENDAHULUAN**

Bab ini berisikan penjelasan tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penulisan, Manfaat Penulisan, dan Metodologi Penelitian.

**BAB 2                    LANDASAN TEORI**

Bab ini akan membahas teori-teori yang berkaitan dengan optimisasi, penjadwalan, algoritma genetika, dan komponen utama algoritma genetika.

**BAB 3                    ANALISIS DAN PERANCANGAN**

Bab ini akan menganalisis penjadwalan yang biasa digunakan di Fasilkom-TI USU, aturan pokok yang digunakan, aturan tambahan yang digunakan, dan masalah yang timbul dari aturan-aturan yang digunakan. Bab ini juga akan menjelaskan rancangan sistem yang akan dibuat dengan menggunakan *flowchart*, DFD, serta bagaimana nantinya sistem akan berjalan dan garis besar penggunaan sistem.

**BAB 4                    IMPLEMENTASI DAN PENGUJIAN**

Bab ini mencakup bagaimana algoritma genetika akan diterapkan untuk masalah penjadwalan. Setelah penerapan dilaksanakan, maka aplikasi akan diuji apakah sesuai dengan tujuan awal dari pembangunan sistem.

**BAB 5                    KESIMPULAN DAN SARAN**

Bab ini berisi rangkuman keseluruhan dari penelitian yang telah dilakukan. Apakah algoritma dapat digunakan untuk penyelesaian masalah penjadwalan, atau sebaliknya, algoritma ini tidak dapat diterapkan untuk masalah penjadwalan.



## BAB 2

### LANDASAN TEORI

Pada bab ini akan membahas landasan atas teori-teori yang bersifat ilmiah untuk mendukung penulisan skripsi ini. Teori-teori yang dibahas mengenai optimisasi, pengertian penjadwalan, algoritma genetika, dan beberapa subpokok pembahasan lainnya yang menjadi landasan dalam penulisan skripsi ini.

#### 2.1 Optimisasi

Optimisasi adalah suatu proses yang berhubungan dengan penyesuaian masukan, pemilihan karakteristik peralatan, proses matematis dan pengujian yang dilakukan untuk menemukan output optimum. Proses optimisasi dapat dipandang sebagai proses minimasi atau proses maksimasi, tergantung sudut pandang yang digunakan [ZUK13]. Proses optimisasi diilustrasikan pada Gambar 2.1.



**Gambar 2.1** Diagram optimisasi fungsi atau optimisasi proses

##### 12.1.1 Klasifikasi optimisasi

Secara umum, persoalan optimisasi dikelompokkan menjadi 4 (empat) kelompok antara lain:

#### **12.1.1.1 Optimisasi tanpa pembatas (*Unconstraint Optimization*)**

Optimisasi tanpa pembatas atau dikenal dengan *unconstraint optimization* biasanya berkaitan dengan maksimasi atau minimasi dari suatu fungsi dengan beberapa variabel. Setiap variabel tidak memiliki pembatas dan mempunyai nilai ril.

#### **12.1.1.2 Optimisasi dengan pembatas (*Constraint Optimization*)**

Optimisasi tanpa pembatas atau *constraint optimization* sering dikenal dengan istilah *nonlinear programming*. Persoalan ini berkaitan dengan optimisasi suatu fungsi tujuan yang memiliki beberapa fungsi pembatas. Fungsi pembatas dapat berupa suatu persamaan ataupun pertidaksamaan. Persoalan ini banyak dijumpai dalam berbagai aplikasi teknik, operasi riset, matematika, ekonomi dan sebagainya.

#### **12.1.1.3 Optimisasi kombinatorik (*Combinatorial Optimization*)**

Persoalan optimisasi kombinatorik mempunyai ciri bahwa terdapat solusi yang layak dalam jumlah yang terhingga. Namun demikian, pada aplikasinya, persoalan yang dihadapi sangatlah besar sehingga jumlah solusi yang layak juga sangat banyak. Pada kondisi yang demikian sangat sulit memperoleh solusi optimal dari persoalan ini dengan metode tradisional.

#### **12.1.1.4 Optimisasi dengan beberapa fungsi tujuan (*Multi-Objective Optimization*)**

Persoalan yang memiliki beberapa fungsi tujuan dikenal dengan istilah multi-objective optimization. Pada persoalan ini, tidak selamanya terdapat solusi yang optimal untuk keseluruhan fungsi tujuan yang ada. Suatu solusi mungkin saja optimal untuk fungsi tujuan tertentu, namun sangat buruk untuk fungsi tujuan yang lain. Solusi untuk persoalan optimisasi dengan beberapa fungsi tujuan dikenal dengan istilah *non-dominated solution* atau *Pareto solution*.

## 2.2 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 untuk menjadwalkan atau memasukkan dalam jadwal. Berdasarkan pengertian tersebut, dapat disimpulkan bahwa penjadwalan merupakan permasalahan optimisasi dengan batasan/*constrain* yang harus dipenuhi untuk mencapai tujuan tertentu. Dalam lingkungan pendidikan, penjadwalan berhubungan dengan alokasi yang memuaskan antara sumber daya dan waktu untuk mencapai kelancaran kegiatan belajar mengajar di kelas [LAL03].

Perumusan jadwal perkuliahan biasanya melibatkan banyak aspek yang terlibat dalam proses penjadwalan. Aspek-aspek yang biasanya berkaitan penjadwalan dan sering dijadikan bahan pertimbangan utama adalah:

1. Terdapat jadwal-jadwal di mana dosen yang bersangkutan tidak bisa mengajar.
2. Distribusi jadwal perkuliahan diharapkan dapat merata tiap harinya untuk setiap kelas.
3. Pekerjaan penjadwalan mata kuliah ini akan semakin berat jika melibatkan banyak kelas per angkatan.

## 2.3 Algoritma Genetika

Algoritma genetika merupakan metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah teori evolusi Charles Darwin. Algoritma genetika menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*) [NUG08]. Proses pencarian atau terpilihnya suatu penyelesaian dalam algoritma ini berlangsung sama seperti terpilihnya suatu individu untuk bertahan hidup dalam proses evolusi. Perkembangan algoritma genetika berawal pada tahun 1975, ketika itu John Holland menerbitkan buku dengan judul

“*Adaption in Natural and Artificial System*”. Di buku tersebut ia berpendapat bahwa semua masalah yang berbentuk adaptasi (alami atau buatan) dapat diformulasikan dalam bentuk terminologi genetika [KUS05].

Algoritma genetika bekerja dengan menggunakan sekumpulan kandidat solusi  $n$  kromosom yang dikenal dengan istilah populasi (*population*). Masing-masing kromosom terdiri dari sejumlah bilangan atau simbol yang mempresentasikan suatu solusi yang layak (*feasible solution*) untuk persoalan yang akan diselesaikan. Kromosom-kromosom tersebut akan diperiksa nilai yang sebenarnya. Seperti halnya proses evolusi alamiah/seleksi alam, kromosom yang terpilih adalah kromosom yang mampu bertahan hingga akhir, yaitu kromosom yang memiliki nilai fitness tinggi. Selanjutnya, kromosom yang bertahan akan melakukan proses reproduksi melalui penyilangan (*crossover*) seperti proses perkawinan individu dalam proses evolusi. Sebagian kecil dari kromosom tersebut juga akan terkena mutasi. Hasil dari proses reproduksi ini akan melahirkan individu-individu baru. Gabungan dari individu baru dengan kromosom yang tidak melakukan proses reproduksi akan membentuk populasi baru pada generasi berikutnya. Serangkaian proses seperti ini akan berlangsung hingga sejumlah generasi tercapai. Penyelesaian yang akan ditemukan adalah kromosom dengan tingkat fitness tertinggi pada generasi terakhir [SYA14].

Algoritma genetika sangat tepat digunakan untuk menyelesaikan permasalahan optimasi yang kompleks. Di dalam algoritma genetika, solusi permasalahan direpresentasikan ke dalam bentuk kromosom. Tiga aspek yang penting untuk penggunaan algoritma genetika yaitu :

1. Fungsi *fitness*.

Fungsi fitness digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan. Fungsi ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan.

2. Implementasi representasi genetik berupa kromosom.

Representasi kromosom merupakan proses pengkodean dari penyelesaian asli dari suatu permasalahan. Pemilihan representasi kromosom yang digunakan harus dapat

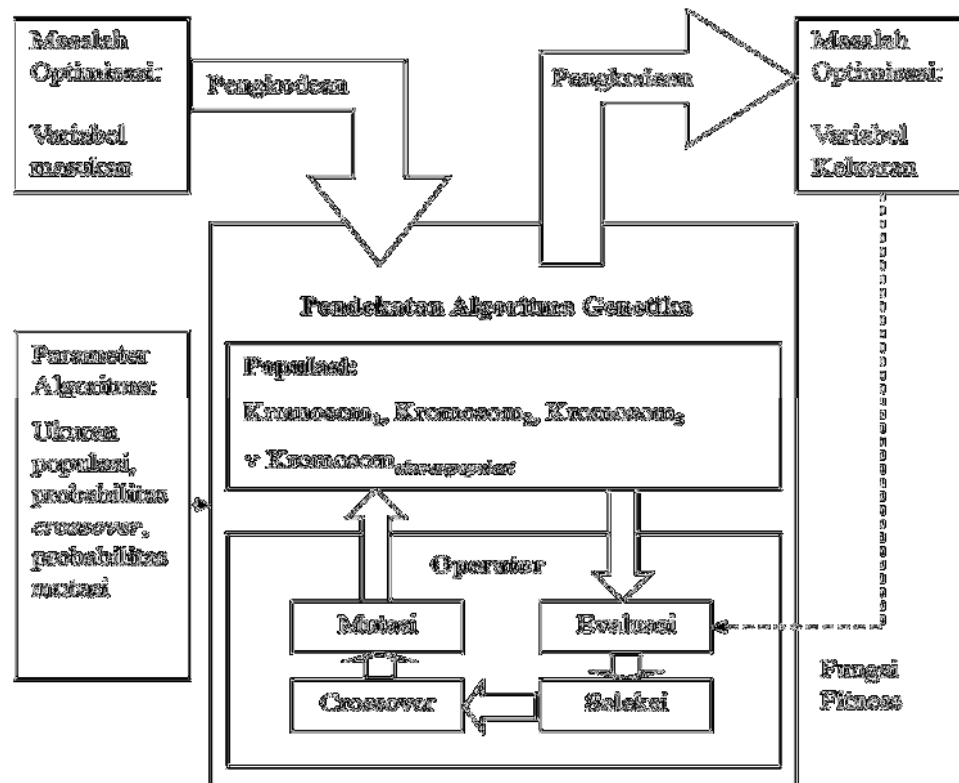
mempresentasikan semua parameter dari kandidat solusi yang mungkin untuk persoalan optimasi.

3. Implementasi operasi genetik berupa operator *crossover* dan mutasi.

Operasi genetik merupakan proses perubahan kromosom pada suatu populasi dengan menggunakan operator genetika tertentu. Operator yang memberikan perubahan terbesar dalam susunan kromosom suatu populasi adalah operator crossover dan mutasi. Pemilihan jenis operasi yang tepat dapat mempercepat laju evolusi dari suatu populasi.

### 2.3.1 Struktur dasar algoritma genetika

Struktur dasar yang biasa digunakan dalam penerapan Algoritma Genetika untuk menyelesaikan suatu masalah optimisasi ditunjukkan Gambar 2.2 [ZUK13].



**Gambar 2.2** Struktur dasar penerapan Algoritma Genetika untuk menyelesaikan suatu masalah optimisasi

Struktur dasar dari algoritma genetika sering disebut *Simple Genetic* oleh John Holland dinyatakan sebagai berikut:

1. [*Start*], generasi populasi pertama secara random sebanyak  $n$  individu.
2. [*Fitness*], evaluasi nilai fitness  $f(x)$  dari individu  $x$  didalam populasi.
3. [*New Population*], bentuk populasi baru dengan melakukan pengulangan langkah-langkah dibawah ini sehingga didapat populasi baru.
  - a. [*Selection*], pilih 2 individu sebagai orang tua dari sebuah populasi sesuai dengan fitness mereka (semakin baik fitness, maka semakin besar peluang mereka terpilih).
  - b. [*Crossover*], lakukan perkawilan silang antara kedua orang tua sesuai dengan probabilitas crossover untuk membentuk keturunan yang baru. Jika tidak terjadi persilangan maka keturunan yang dihasilkan akan sama persis dengan orang tuanya.
  - c. [*Mutation*], mutasai setiap keturunan yang baru sesuai dengan probabilitas mutasi di setiap gen.
  - d. [*accepting*], tempatkan keturunan yang baru sebagai populasi baru.
4. [*Replace*], gunakan populasi yang baru dibentuk untuk menjalankan algoritma.
5. [*Test*]. jika kondisi akhir dipenuhi maka berhenti dan tampilkan solusi dari populasi.

### 2.3.2 Syarat berhenti dan parameter algoritma genetika

Dalam paradigma algoritma genetika alamiah, proses yang terjadi adalah proses tiada berhenti (*never-ending process*). Namun, secara praktiknya, suatu algoritma genetika yang berjalan harus memiliki syarat untuk berhenti, proses akan berhenti bila syarat pemberhentian terpenuhi[KOZ92]. Dalam penelitian ini, yang menjadi syarat berhenti algoritma genetika adalah bila nilai fitness telah bernilai 1 dan tidak ditemukan kromosom bentrok (penalti=0).

Parameter genetika ditentukan berdasarkan permasalahan yang dipecahkan. Tidak ada aturan pasti tentang nilai setiap parameter ini. Beberapa parameter yang umum digunakan adalah:

### 1. Probabilitas Crossover

Adalah peluang untuk melakukan pertukaran gen-gen diantara kromosom. Pada biologi, crossover adalah hibridasi antara jenis yang berbeda. Untuk melakukan crossover, perlu ditentukan konstanta  $p_c$ , yaitu konstanta yang menyatakan besarnya peluang untuk melakukan crossover. Nilai  $p_c$  tersebut biasanya adalah sebesar 0.7. Iterasi dilakukan pada tiap kromosom dan dilakukan pengambilan nilai acak diantara 0 s/d 1 pada tiap kromosom. Jika nilai acak yang dihasilkan adalah  $\leq p_c$ , maka kromosom tersebut terpilih untuk dilakukan crossover, sedangkan jika  $> p_c$ , maka tidak dilakukan crossover pada kromosom tersebut [SIH10].

### 2. Probabilitas Mutasi

Parameter Probabilitas Mutasi atau dikenal juga  $P_m$  memiliki nilai 0 – 1. Nilai ini menggambarkan seberapa sering mutasi akan dilakukan. Artinya, jika nilai  $P_m$  adalah 0, maka dipastikan tidak dilakukan proses mutasi atau dengan kata lain seluruh kromosom pada generasi yang baru akan sama dengan kromosom hasil crossover. Sebaliknya, jika  $P_m$  bernilai 1, maka semua kromosom akan dipastikan melakukan mutasi [SYA14].

### 3. Ukuran Populasi (*Population size*)

Nilai parameter *pop\_size* menunjukkan jumlah kromosom pada populasi (setiap generasi). Apabila jumlah kromosom dalam suatu populasi terlalu kecil maka akan semakin sedikit kromosom yang melakukan crossover dan mutasi. Hal ini akan mempengaruhi kualitas solusi yang akan diperoleh. Sebaliknya, apabila *pop\_size* besar, proses algoritma genetika akan menjadi sangat lambat [SYA14].

### 4. Ukuran Generasi (*Generation size*)

Ukuran generasi yang sedikit mengakibatkan keterbatasan pilihan untuk crossover dan sebagian kecil dari domain solusi saja yang dieksplorasi setiap generasinya. Sedangkan bila terlalu besar, kinerja algoritma genetika menurun [SYA14].

### 2.3.3 Teknik pengkodean

Algoritma genetika dapat dijalankan berdasarkan teori evolusi, maka setiap solusi harus direpresentasikan dalam sebuah kode yang sesuai dalam persoalan. Kode yang digunakan harus dapat mewakili seluruh ruang penyelesaian [BER10].

Ada beberapa jenis pengkodean yang dapat digunakan dalam algoritma genetika yaitu pengkodean biner (*binary encoding*), pengkodean nilai (*value encoding*), pengkodean permutasi (*permutation encoding*), pengkodean pohon (*tree encoding*).

#### 2.3.3.1 Pengkodean biner

Pengkodean ini merupakan pengkodean yang sering digunakan dan paling sederhana. Pada pengkodean biner setiap kromosom direpresentasikan dalam barisan bit 0 atau 1, seperti pada tabel 2.1.

**Tabel 2.1** Contoh Pengkodean Biner

Kromosom A	1010101011
Kromosom B	1100001010

Pengkodean biner memberikan banyak kemungkinan untuk kromosom walaupun dengan jumlah nilai-nilai yang mungkin terjadi dalam suatu gen sedikit (0 atau 1). Pengkodean ini sering tidak sesuai untuk beberapa masalah terkadang harus dilakukan pengkoreksian setelah operasi *crossover* dan mutasi.

#### 2.3.3.2 Pengkodean nilai

Didalam pengkodean nilai, setiap kromosom adalah string dari suatu nilai yang merupakan representasi dari masalah seperti bilangan bulat, desimal ataupun karakter. Contoh pengkodean ini dapat dilihat pada tabel 2.2.



**Tabel 2.2** Contoh Pengkodean Nilai

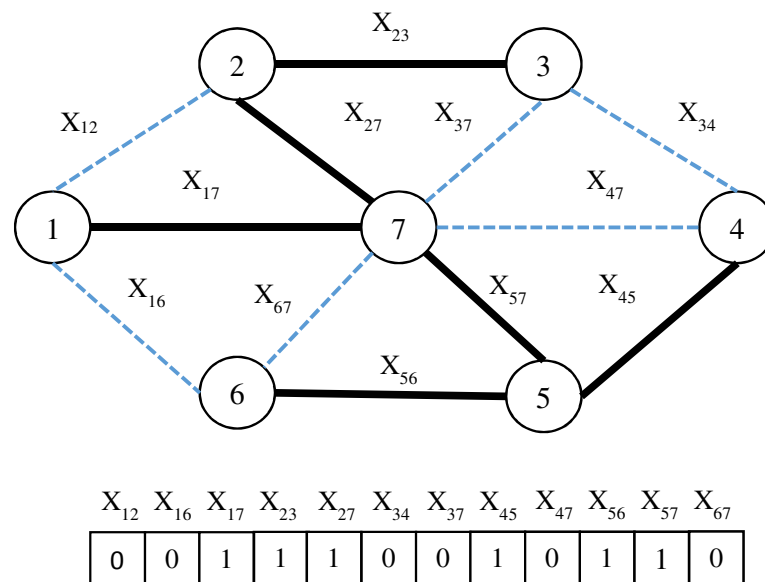
Kromosom A	1.345, 4.534, 7.654, 8.789
Kromosom B	ABC, ADC, CBC, BCA
Kromosom C	1,3,4,7,5
Kromosom D	Forward, backward, right, left

### 2.3.3.3 Pengkodean pohon

Metode pengkodean pohon dapat digunakan untuk persoalan yang solusinya memiliki struktur pohon. Implementasi algoritma genetika dengan pengkodean pohon termasuk persoalan logistik, persoalan transportasi dan persoalan *minimum spanning tree*. Ada dua kelompok metode pengkodean pohon. [SYA14]

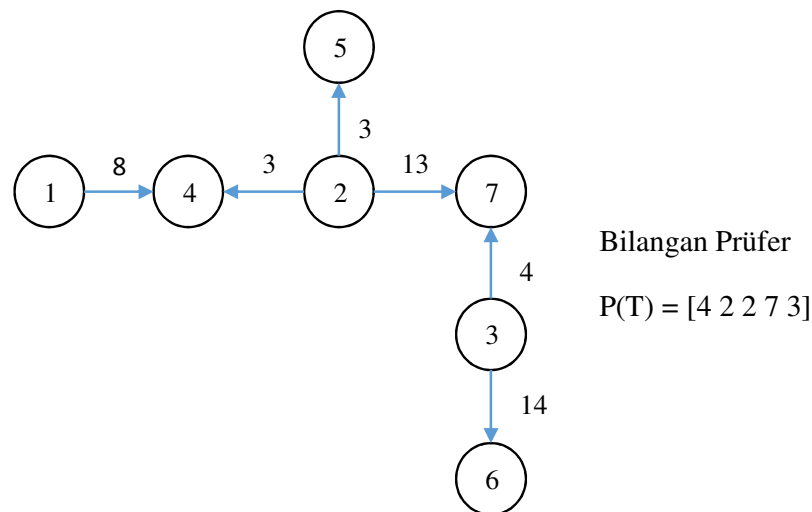
#### a. Edge Encoding

Metode ini menggunakan N digit bilangan biner yang merepresentasikan jaringan dengan N *arc*. Apabila digit ke-i bernilai 1 berarti *arc* ke-i tersebut merupakan bagian dari solusi. Contoh pengkodean pohon edge encoding dapat dilihat pada gambar 2.3. [SYA14]

**Gambar 2.3** Pengkodean Pohon Tipe Edge Encoding

### b. Vertex Encoding

Salah satu metode pengkodean yang cukup dikenal adalah pengkodean dengan representasi bilangan Prüfer. Bilangan Prüfer adalah suatu bilangan dengan  $N$  digit yang dapat merepresentasikan suatu pohon dengan  $N-2$  node. Metode ini dapat digunakan untuk menyelesaikan berbagai persoalan dengan solusi berbasis pohon. Contoh pengkodean pohon vertex encoding dapat dilihat pada gambar 2.4. [SYA14]



**Gambar 2.4** Pengkodean Pohon Tipe Vertex Encoding

#### 2.3.3.4 Pengkodean permutasi

Pengkodean permutasi adalah pengkodean yang digunakan dalam masalah pengurutan data (*ordering problem*), seperti masalah wiraniaga (*travelling salesman problem*), atau masalah pengurutan tugas (*task ordering problem*). Pada pengkodean ini setiap kromosom merupakan barisan angka yang merepresentasikan angka dalam urutan. Pengkodean ini berguna untuk masalah ordering, bahkan beberapa korelasi terhadap kromosom harus dilakukan untuk menjaga konsistensi representasi kromosom setelah proses *crossover* dan mutasi. Sebagai contoh, dapat dilihat pada tabel 2.3.

**Tabel 2.3** Contoh Pengkodean Permutasi

Kromosom A	2 6 7 5 1 3 4 9 8 10
Kromosom B	10 5 4 9 7 1 3 2 6 8

### 2.3.4 Evaluasi Fitness

Fungsi fitness digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan. Fungsi ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan. Fungsi fitness yang digunakan sebagai berikut.

$$Fitness = \frac{1}{1 + (\sum_{i=1}^6 p_i)} \quad (2.1)$$

Dari persamaan 2.1, nilai fitness ditentukan oleh nilai penalti/pelanggaran yang terjadi. Nilai yang dihasilkan tersebut menandakan seberapa optimal solusi yang diperoleh, dengan kata lain dalam penjadwalan perkuliahan semakin kecil jumlah pelanggaran yang dihasilkan maka solusi yang dihasilkan akan semakin baik [BAN12].

### 2.3.5 Operator algoritma genetika

Operator yang sering digunakan pada algoritma genetika adalah seleksi, *crossover* dan mutasi. Pemilihan jenis operator yang digunakan tergantung dari masalah yang akan diselesaikan.

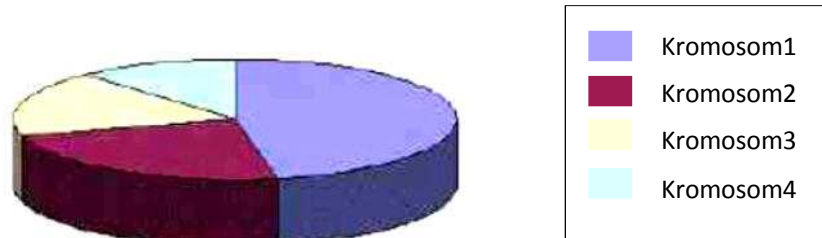
#### 2.3.5.1 Metode seleksi

Proses seleksi adalah proses menentukan individu-individu mana yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu terpilih tersebut [KUS05].

Pada proses algoritma genetika dikenal beberapa metode seleksi yang umum digunakan untuk memilih kromosom. Metode seleksi tersebut adalah sebagai berikut.

a. Seleksi *Roulette Wheel*

Metode *Roulette Wheel* merupakan metode seleksi pemilihan induk dengan menggunakan prosentasi fitness setiap individu, di mana setiap individu mendapatkan luas bagian sesuai dengan prosentase nilai fitnessnya [ADH03].

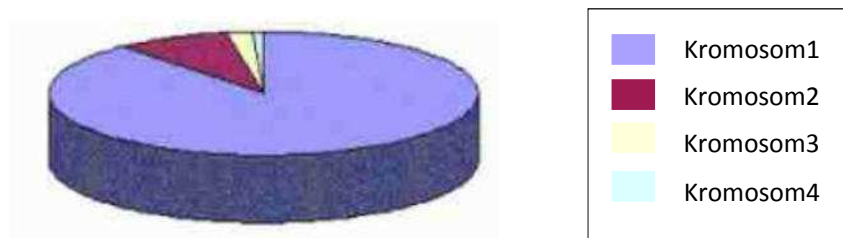


**Gambar 2.5** Seleksi Metode *Roulette Wheel*

b. Seleksi *Ranking*

Seleksi *ranking* merupakan metode seleksi dimana populasi diurutkan berdasarkan nilai *fitness*-nya sehingga nilai yang diharapkan dari tiap individu bergantung kepada urutannya bukan hanya kepada nilai *fitness*-nya. [APR12]

Dalam seleksi *ranking*, dilakukan perumpamaan sesuai dengan nilai *fitness*-nya, nilai *fitness* terkecil diberi nilai 1, yang terkecil kedua diberi nilai 2, dan seterusnya sampai yang terbagus diberi nilai N (jumlah kromosom dalam populasi). Nilai tersebut yang akan diambil sebagai presentasi tepat yang tersedia. Ilustrasinya dapat dilihat seperti pada Gambar 2.6.



**Gambar 2.6** Seleksi Metode *Ranking*

c. Elitisme

Dalam metode elitisme, sebuah kromosom yang paling tinggi nilai *fitness* pada generasi sebelumnya akan dipertahankan untuk populasi generasi berikutnya. Sisa kromosom pada generasi berikutnya akan diperoleh dengan cara reproduksi biasa. [SYA14]

d. Seleksi *Tournament*

Dalam metode ini, kromosom-kromosom dalam suatu populasi dibagi menjadi beberapa grup secara acak. Setiap grup akan beranggotakan minimal dua kromosom. Seleksi dilakukan dengan mempertahankan kromosom yang memiliki nilai *fitness* tertinggi pada setiap grup. [ZUK13]

### 2.3.5.2 Metode Crossover

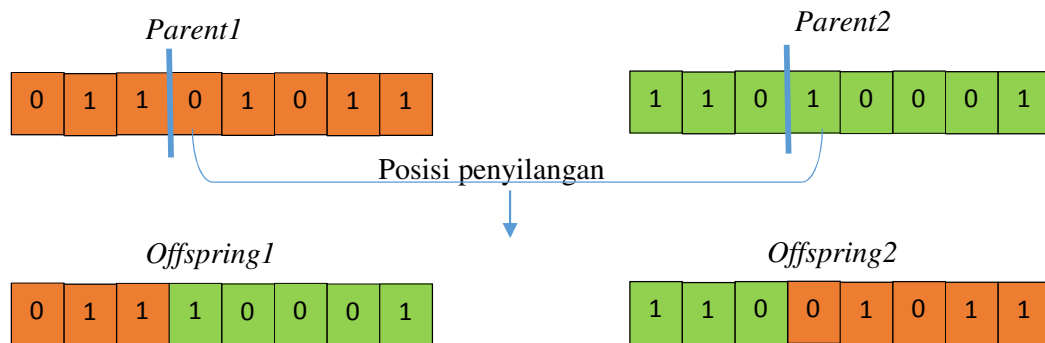
*Crossover* adalah proses pembentukan kromosom turunan (*offspring*) dengan menggabungkan elemen dari kromosom induk yang terpilih (*parent*). *Crossover* bertujuan menambah keanekaragaman string dalam populasi dan mendapatkan kromosom baru dengan solusi yang lebih baik.

Proses crossover dipengaruhi salah satu parameter algoritma genetika, yaitu probabilitas crossover ( $P_c$ ). Probabilitas crossover menunjukkan persentase kemungkinan persilangan terjadi antara 2 kromosom. Probabilitas crossover dari suatu individu biasanya dipilih cukup besar, persis seperti kejadian sebenarnya dalam kehidupan alamiah dengan tujuan mempercepat terbentuknya generasi baru. Interval nilai probabilitas crossover berkisar 0% hingga 100%. Artinya, jika  $P_c = 0\%$ , maka keturunan akan sama persis dengan kromosom induk. Jika  $P_c = 100\%$  maka semuanya keturunan dihasilkan adalah keturunan hasil crossover.

Pada proses algoritma genetika dikenal beberapa metode *crossover* yang umum digunakan. Metode *crossover* tersebut adalah sebagai berikut.

a. Penyilangan Satu Titik (*One-point Crossover*)

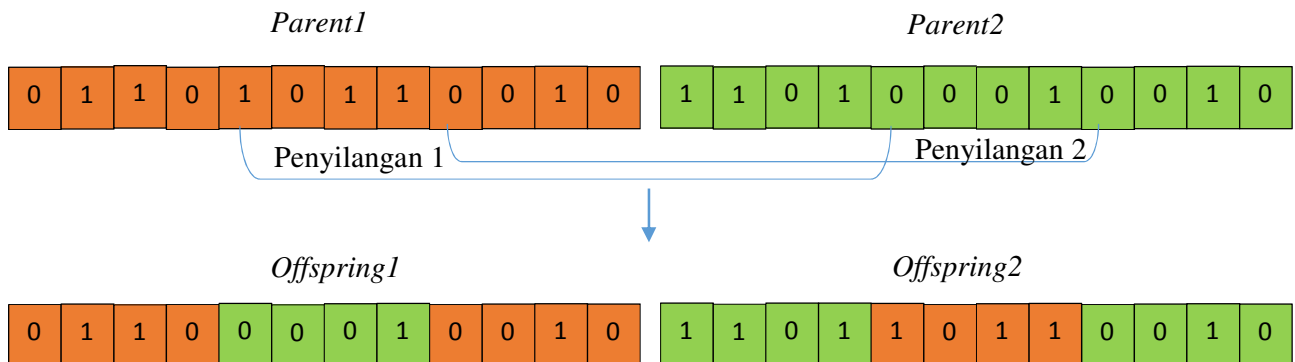
Penyilangan satu titik adalah metode crossover yang paling sering. Metode ini bekerja dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara sama untuk menghasilkan *offspring*. Dalam penelitian ini, penulis menggunakan metode penyilangan satu titik sebagai operator crossover. Sebagai contoh, dapat dilihat pada Gambar 2.7.



**Gambar 2.7** Penyilangan satu titik

b. Penyilangan Dua Titik (*Two-point Crossover*)

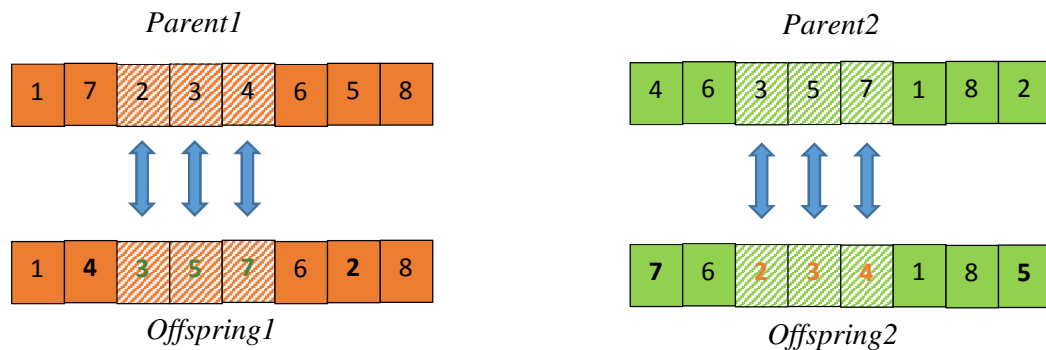
Metode ini dimulai dengan memilih dua titik *crossover*. Kromosom keturunan kemudian dibentuk dengan barisan bit dari awal kromosom sampai titik *crossover* pertama disalin dari orang tua pertama, bagian dari titik *crossover* pertama dan kedua disalin dari orang tua kedua, kemudian selebihnya disalin dari orang tua pertama lagi. Sebagai contoh, dapat dilihat pada gambar 2.8.



**Gambar 2.8** Penyilangan dua titik

c. *Partial Mapped Crossover (PMX)*

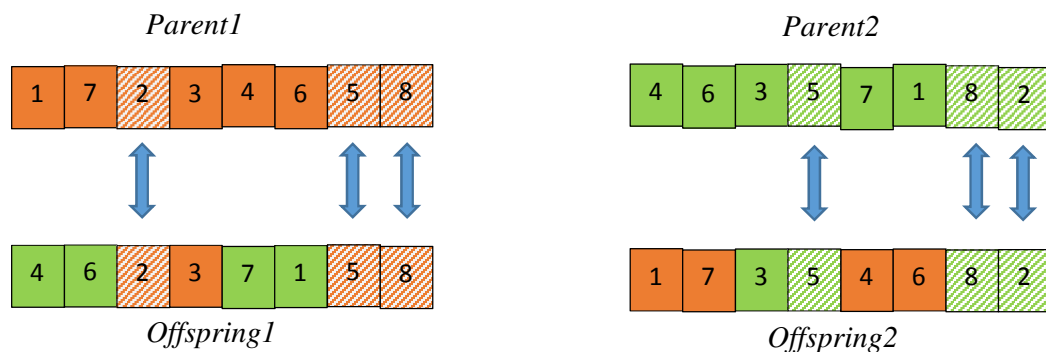
Metode ini biasanya digunakan untuk melakukan *crossover* pada persoalan yang menggunakan representasi permutasi. Metode ini merupakan perkembangan dari metode Penyilangan Dua Titik (*Two-point Crossover*). Sebagai contoh, dapat dilihat pada gambar 2.9.



**Gambar 2.9** Penyilangan PMX

d. *Order Crossover (OX)*

Metode ini diperkenalkan oleh Syswerda [SYA14]. Pada prinsipnya metode ini sangat mirip dengan metode PMX. Perbedaannya hanya terletak pada prosedur perbaikan, yaitu proses pengisian gen selain substring dilakukan secara silang dan dilakukan secara berurutan dari kiri ke kanan. Sebagai contoh, dapat dilihat pada gambar 2.10.



**Gambar 2.10** Penyilangan OX

### 2.3.5.3 Metode Mutasi

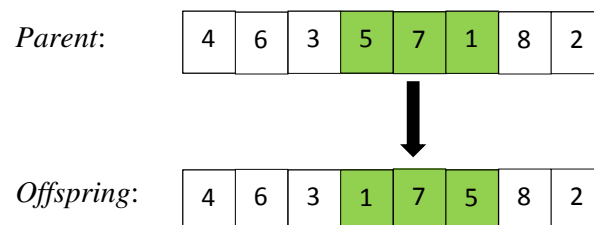
Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Individu yang telah melewati proses seleksi dan *crossover* akan menghasilkan individu baru (*offspring*) yang akan dimutasi untuk membantu mempercepat terjadinya perbedaan individu pada populasi. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi.

Proses mutasi dipengaruhi salah satu parameter algoritma genetika, yaitu probabilitas mutasi ( $P_m$ ). Probabilitas mutasi akan menentukan persentase kemungkinan terjadi proses mutasi. Probabilitas mutasi dari suatu gen biasanya dipilih sangat kecil, persis seperti kejadian sebenarnya dalam kehidupan alamiah yang memungkinkan terjadinya mutasi genetis tapi dalam prosentasi sangat kecil. Interval nilai probabilitas mutasi berkisar 0% hingga 100%. Artinya, jika  $P_m = 100\%$  maka semua kromosom akan dimutasi. Jika  $P_m = 0\%$  maka semua kromosom tidak ada yang mengalami mutasi, dan kromosom yang terbentuk sama persis seperti kromosom hasil crossover.

Pada proses algoritma genetika dikenal beberapa metode mutasi yang umum digunakan. Metode mutasi tersebut adalah sebagai berikut.

a. Metode pembalikan (*Inversion mutation*)

Metode mutasi pembalikan dilakukan dengan mengambil suatu *substring* yang terletak diantara dua titik pada kromosom. Pemilihan dua titik ini dilakukan secara acak. Selanjutnya dilakukan proses pembalikan (*invers*) gen pada substring tersebut. Ilustrasi metode ini dapat dilihat pada Gambar 2.11.

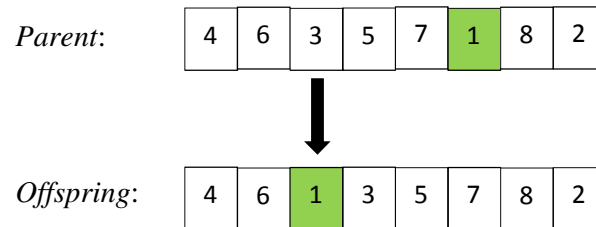


**Gambar 2.11** Mutasi pembalikan



b. Metode penyisipan (*Insertion mutation*)

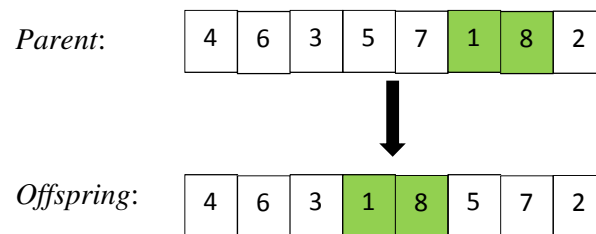
Metode penyisipan dilakukan dengan memilih salah satu gen yang ada pada kromosom. Selanjutnya gen tersebut disisipkan pada posisi yang dipilih secara acak juga. Ilustrasi metode dapat dilihat pada Gambar 2.12.



**Gambar 2.12** Mutasi penyisipan

c. Metode pemindahan (*Displacement mutation*)

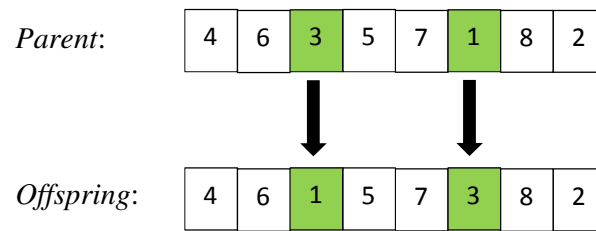
Berbeda dengan metode penyisipan yang hanya menyisipkan salah satu gen dari kromosom. Metode pemindahan dilakukan dengan memilih dua titik pada kromosom. Selanjutnya gen yang ada diantara kedua titik tersebut disisipkan pada suatu posisi yang juga dipilih secara acak. Ilustrasi metode ini dapat dilihat pada Gambar 2.13.



**Gambar 2.13** Mutasi pemindahan

d. Metode penukaran (*Swap mutation*)

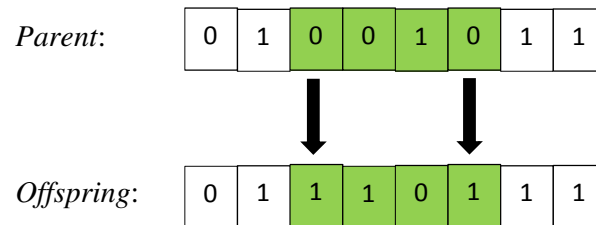
Metode penukaran dilakukan dengan memilih dua buah gen yang ada pada kromosom. Selanjutnya kedua gen tersebut dipertukarkan satu dengan yang lain. Penukaran terarah seperti yang dilakukan metode ini dinilai lebih efisien dan mempercepat proses perbedaan di antara individu. Ilustrasi metode ini dapat dilihat pada Gambar 2.14.



**Gambar 2.14** Mutasi penukaran

e. Metode penggantian (*Flip mutation*)

Metode penggantian ini digunakan untuk persoalan yang menggunakan teknik pengkodean biner. Metode ini dilakukan dengan mengganti nilai gen terpilih pada kromosom dengan nilai gen lainnya. Ilustrasi metode ini dapat dilihat pada Gambar 2.15. [APR12]

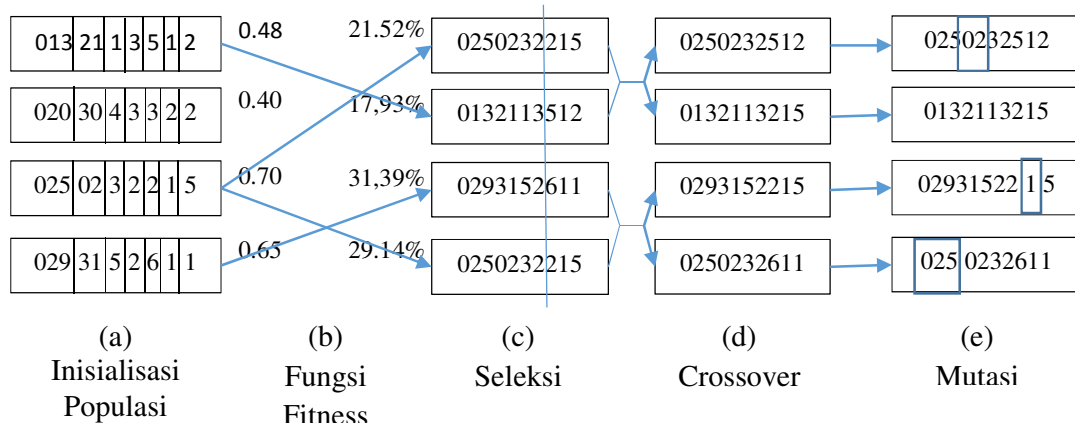


**Gambar 2.15** Mutasi penggantian

### 2.3.6 Schema algoritma genetika

Teori dari algoritma genetik menjelaskan cara kerjanya menggunakan ide dari suatu *schema*, suatu substring di mana beberapa posisi tidak disebutkan. Dapat ditunjukkan bahwa, bila *fitness* rata-rata dari *schema* berada di bawah mean maka jumlah instansi dari *schema* di dalam populasi akan bertambah seiring bertambahnya waktu. Jelas sekali bahwa efek ini tidak akan signifikan bila bit-bit yang bersebelahan sama sekali tidak berhubungan satu sama sekali, karena akan ada beberapa blok kontigu yang memberikan keuntungan yang konsisten. Algoritma genetik paling efektif dipakai bila *schema-schema* berkorespondensi menjadi komponen berarti dari sebuah solusi. Sebuah komponen yang baik cenderung akan berkerja baik pada rancangan yang

berbeda. Ini menunjukkan bahwa penggunaan algoritma genetika yang benar memerlukan rekayasa yang baik pada representasinya [NUG08].



**Gambar 2.16** Contoh penggunaan Algoritma Genetika

Berdasarkan Gambar 2.16, algoritma genetika berawal dari proses inisialisasi populasi. Pada contoh tersebut, dimisalkan populasi awal yang dibentuk sebanyak 4 individu. Satu individu tersusun atas banyak kromosom. Dalam penelitian ini, jumlah kromosom yang terbentuk pada suatu individu sesuai dengan jumlah mata kuliah dan praktikum. Teknik pengkodean yang digunakan adalah teknik integer, tepatnya *discrete decimal encoding*. Untuk susunan representasi kromosom secara berurutan adalah gen mata kuliah, dosen, kelas, SKS, ruang, hari dan waktu.

Setiap individu akan dihitung nilai fitnessnya menggunakan fungsi fitness. Fungsi fitness akan menghasilkan nilai dengan memanfaatkan pelanggaran batasan yang dilakukan oleh kromosom pada setiap individu, selanjutnya akan dihitung persentasi fitness dari keseluruhan populasi. Individu dengan persentasi yang besar, dalam contoh 31,39% akan berkemungkinan untuk dipilih sebagai *parent* dalam tahap crossover selanjutnya, dan individu dengan persentasi yang kecil akan punah karena tidak memiliki kemungkinan menjadi *parent* dalam tahap selanjutnya.

Pada tahap selanjutnya yaitu *crossover*, individu yang telah dipilih sebagai *parent* akan ditentukan terlebih dahulu titik potong kromosom untuk dikawinsilangkan dengan kromosom lainnya. Pada contoh ini, penulis memilih titik potong terletak antara bit ke-

7 dan ke-8. Hal tersebut karena gen ruang, hari dan waktu dinilai lebih bersifat bebas, dibandingkan dosen dan SKS yang terikat dengan matakuliah yang ada. Individu yang telah dipotong akan dikawinsilangkan sehingga menghasilkan individu anak (*offspring*) yang baru.

Pada tahap selanjutnya yaitu mutasi, gen dari kromosom individu offspring akan berkemungkinan dimutasi berdasarkan perbandingan bilangan rill random dan probabilitas mutasi yang ada. Gen yang dikenakan mutasi bersifat acak, namun tetap memperhatikan batasan yang ada.

### **2.3.7 Kelebihan dan kelemahan algoritma genetika**

Menurut Zukhri [ZUK13], beberapa kelebihan algoritma genetika jika dibandingkan dengan algoritma pencarian lainnya adalah:

1. Algoritma ini hanya melakukan sedikit perhitungan matematis yang berhubungan dengan masalah yang ingin diselesaikan. Karena sifat perubahan evolusi alamiah, maka algoritma ini akan mencari penyelesaian tanpa memperhatikan proses-proses yang berhubungan dengan masalah yang diselesaikan secara langsung. Algoritma ini juga dapat mengendalikan fungsi objektif dan batasan yang didefinisikan, baik pada ruang pencarian diskrit atau ruang pencarian analog.
2. Operator-operator evolusi membuat algoritma ini sangat efektif pada pencarian global.
3. Algoritma ini memiliki fleksibilitas yang tinggi untuk dihibridkan dengan metode pencarian lainnya agar lebih efektif.

Kelemahan algoritma genetika jika dibandingkan dengan algoritma pencarian lainnya adalah adanya ketidakpastian untuk menghasilkan solusi optimum global, karena sebagian besar dari algoritma ini berhubungan dengan bilangan random yang bersifat probabilistik. [WAH09].

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1. Batasan-Batasan pada Penjadwalan**

Penjadwalan di universitas memiliki batasan-batasan atas mata kuliah yang akan dijadwalkan. Maksud dari keterbatasan di sini adalah adanya mata kuliah yang harus dijadwalkan secara simultan, dan ada yang tidak perlu. Batasan-batasan dalam penjadwalan mata kuliah terdiri atas dua kategori, yaitu batasan pokok penjadwalan dan batasan tambahan penjadwalan.

##### **3.1.1 Batasan pokok penjadwalan (*hard constraints*)**

Batasan pokok penjadwalan adalah kondisi atau batasan yang harus terpenuhi untuk menghasilkan jadwal mata kuliah yang mudah dikerjakan dan diterima oleh semua variabel yang terlibat. Contoh aturan-aturan yang termasuk ke dalam batasan pokok penjadwalan adalah:

1. Setiap pengajar tidak boleh mengajar pada hari dan waktu yang sama.
2. Setiap kelas tidak boleh masuk kuliah/praktikum pada hari dan waktu yang sama.
3. Setiap ruangan tidak boleh digunakan pada hari dan waktu yang sama.
4. Dosen tidak boleh mengajar pada jam sibuk yang telah terdaftar.
5. Perkuliahan harus dilakukan pada ruangan teori (non-lab).
6. Seluruh perkuliahan dan praktikum akan dikosongkan setiap Jumat, mulai pukul 12.10 WIB hingga pukul 13.50 WIB. Hal ini sehubungan pelaksanaan Sholat Jumat.

### 3.1.2 Batasan tambahan penjadwalan (*soft constraints*)

Batasan tambahan penjadwalan adalah batasan ataupun sering disebut batasan khusus penjadwalan adalah aturan-aturan istimewa yang ditambahkan sesuai dengan kondisi tertentu dari variabel-variabel yang akan dijadwalkan. Batasan-batasan ini tidak terlalu dibutuhkan untuk menghasilkan penjadwalan yang layak pakai, tapi apabila aturan pokok tidak cukup untuk menghasilkan penjadwalan yang layak pakai, maka aturan tambahan dapat digunakan sehingga semua variabel yang terlibat tidak diacuhkan. Aturan tambahan penjadwalan memang kadang diperlukan, tapi tetap tidak memungkinkan untuk memasukkan semua aturan tambahan pada aturan pokok penjadwalan.

Aturan tambahan penjadwalan yang digunakan dalam penelitian ini adalah Praktikum dapat dilakukan pada ruangan Laboratorium dan ruangan Teori.

## 3.2 Analisis Sistem

Metode yang digunakan adalah algoritma genetika karena dapat menyelesaikan permasalahan penjadwalan perkuliahan yang sangat rumit dan kompleks. Algoritma genetika merupakan metode pencarian dimana dalam proses *encoding* (pengkodean) menghasilkan string yang susunan terkecilnya terdiri dari gen-gen. Gen dalam kasus ini adalah urutan tabel mata kuliah, ruangan dan waktu telah dikodekan terlebih dahulu sehingga membentuk suatu kromosom. Sebuah kromosom adalah representasi dari suatu mata kuliah, sehingga jumlah kromosom yang terbentuk harus sesuai dengan jumlah perkuliahan yang ada. Sedangkan individu merupakan satu kesatuan yang terdiri dari banyak kromosom. Dalam kasus algoritma genetika, karena individu yang dihasilkan tidak mungkin sama dengan individu lainnya, maka istilah populasi berarti sekumpulan individu berbeda jenis yang terbentuk dalam suatu generasi.

### 3.3 Komponen Utama Penjadwalan

Komponen utama dalam penjadwalan adalah mata kuliah, dosen, ruangan, kelas dan waktu. Setiap komponen memiliki aturan-aturan yang saling berhubungan agar dapat menghasilkan suatu penjadwalan yang baik. Daftar pengajar yang terdiri dari dosen dan asisten di prodi Ilmu Komputer dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Daftar Dosen dan Asisten pada Program Studi Ilmu Komputer

ID	INISIAL	JENIS	NAMA
1	AES	Praktikum	Angga Eriansyah Setiawan
2	APY	Praktikum	Agung Putu Yoga
3	DTR	Praktikum	Dias Tia Rahmadhani
4	AKH	Praktikum	Akhiruddin Nur
5	DMR	Praktikum	Dina Meiladya Rizki Sujiono
6	FQA	Praktikum	Furqan Alatas
7	MMH	Praktikum	Muhammad Miftahul Huda
8	AIS	Praktikum	Anhar Ismail
9	GEU	Praktikum	Geubrina Rizky
10	GUS	Praktikum	Gusra Algeri
11	ATR	Praktikum	Atika Rahayu
12	AMU	Praktikum	Andika Mulia Utama
13	YSN	Praktikum	Yogi Sulaiman
14	APR	Praktikum	Ahmad Pratama Ramadhan
15	NUR	Praktikum	Nurhasanah
16	RAM	Praktikum	Ramrudin, S.Kom
17	AMZ	Praktikum	Al Mizfar
18	DIN	Praktikum	Dini Islami
19	JIP	Praktikum	Jeklin Indriani Purba
20	NHY	Praktikum	Nurhayati
21	VAN	Praktikum	Muhammad Asri Zulfazri Siregar
22	ZUL	Praktikum	Zul Hidayat Daulay
23	WYN	Praktikum	Wynda Ariani Siregar
11	ATR	Praktikum	Atika Rahayu
12	AMU	Praktikum	Andika Mulia Utama
13	YSN	Praktikum	Yogi Sulaiman
14	APR	Praktikum	Ahmad Pratama Ramadhan
24	DPS	Praktikum	Dwi Puspita Sari Syahnan
25	FSY	Praktikum	Faaizah Asy Syuhada
26	FRH	Praktikum	Fitri Rahmadhani
27	IDH	Praktikum	Indri Hidayati
28	TRH	Praktikum	T Raisya Hannisa
29	FAG	Praktikum	Febri Aro Gea
30	MRP	Praktikum	M. Rizki Pratama
31	IRF	Praktikum	Rahmat Irfan Pohan

**Tabel 3.1** Daftar Dosen dan Asisten pada Program Studi Ilmu Komputer (Lanjutan)

ID	INISIAL	JENIS	NAMA
32	DIK	Praktikum	Andika Putra Hutahuruk
33	ATP	Praktikum	Aulia Tarindah Putri
34	NPY	Praktikum	Nurul Putri Yanti
35	RVP	Praktikum	Richard Valent P
36	LJN	Praktikum	Lestari Juwita Ningrum
37	MAR	Praktikum	Martina A. Sipayung
38	ABD	Praktikum	Abidah Novita
39	FAR	Praktikum	Farid Akbar Siregar
40	KHN	Praktikum	Kharunnisa Dewi
41	MUR	Praktikum	Murni Kurniati
42	ARP	Praktikum	Ahmad Royhan, S.Kom
43	DDM	Praktikum	Saddam Husein, S.Kom
44	FFN	Praktikum	Fanny Fairina, S.Kom
45	CLK	Praktikum	Cholik Indriyanto
46	VYN	Praktikum	Arifin
47	NBS	Praktikum	Nurhenida Br. Sitepu, S.Kom
48	LEN	Praktikum	Lennora Marbun
49	YUD	Praktikum	Yudha Prayogi S
50	AMS	Praktikum	Azizah Mei Sari Sebayang, S.Kom
51	YYG	Praktikum	Yayang Kurniati
52	GUN	Praktikum	Gunalan Anggirasa
53	HYN	Praktikum	Hayatun Nufus
54	SHL	Teori	Dra. Hj. Sahlia, M.Ag
55	BMS	Teori	Drs. BM Sembiring
56	LZS	Teori	Linda Zenita Simanjuntak, STP,M.DIV, M.Th.
57	MGL	Teori	Meng Lie, S.Ag.
58	OPD	Teori	Open Darnius,M.Sc
59	MAR	Teori	Drs. Marihat Situmorang,M.Kom
60	ACD	Teori	Drs. Achiruddin, M.Si.
61	SHD	Teori	Syahrul Humaidi, S.Si, M.Sc.
62	DWD	Teori	Dian Wirdasari, S.Si, M.Kom.
63	DRW	Teori	Dian Rachmawati, S.Si, M.Kom.
64	HDL	Teori	Handrizal, S.Si, M.Comp.Sc.
65	SDF	Teori	Siti Dara Fadilla, S.Si, MT
66	SEF	Teori	Dr. Syahril Efendi, S.Si, MIT
67	IRY	Teori	Prof. Dr. Iryanto, M.Si.
68	ANS	Teori	Ayu Nuriana Sebayang, S.Kom, M.Kom
69	HRY	Teori	Herriyance, ST, M.Kom
70	RHA	Teori	Rahmat Aulia, S.Kom, M.Sc.IT
71	SNU	Teori	Hj. Sinta Uli, SH, M.Hum
72	ADR	Teori	Abdul Rahman, SH, MH
73	ASH	Teori	Drs. Agus Salim Harahap, M.Si
74	PTS	Teori	Drs. Partano Siagian, M.Sc.
75	VWJ	Teori	Vera Wijaya, S.Kom, M.Kom
76	ADC	Teori	Ade Candra, ST, M.Kom



**Tabel 3.1** Daftar Dosen dan Asisten pada Program Studi Ilmu Komputer (Lanjutan)

ID	INISIAL	JENIS	NAMA
77	MSL	Teori	Maya Silvi Lydia, B.Sc., M.Sc.
78	AMS	Teori	Amer Sharif, S.Si. M.Kom
79	PLS	Teori	Dr. Poltak Sihombing, M.Kom
80	EMZ	Teori	Dr. Elviawaty Muisa Zamzami, ST,MT,MM
81	JTT	Teori	Jos Timanta Tarigan, S.Kom, M.Sc
82	MAB	Teori	M. Andri Budiman, ST, M.Comp.Sc, MEM
83	DLS	Teori	Drs. Dahlan Sitompul, M.Eng
84	BSM	Teori	Dr. Bisman P, M.Eng
85	MHP	Teori	Dr. Marhaposan, M.Sc.
86	SRS	Teori	Syahriol Sitorus, S.Si, MIT
87	SUY	Teori	Drs. Suyanto, M.Kom
88	ESN	Teori	Dr. Dra. Ester Sorta M Nababan, M.Sc
89	MZL	Teori	Prof. Dr. Muhammad Zarlis
90	ANH	Teori	Ainul Hizriadi, S.Kom, M.Comp.Sc.
91	EBN	Teori	Dr. Erna Budhiarti Nababan,MIT
92	OSS	Teori	Prof. Dr. Opim S. Sitompul, M.Sc.
93	DGN	Teori	Dani Gunawan, ST, MT
94	SJD	Teori	Sajadin Sembiring, S.Si, M.Comp.Sc.
95	ELW	Teori	Elwiwani, ST, S.Kom, M.Kom

Daftar pengajar yang terdiri dari dosen dan asisten di prodi Teknologi Informasi dapat dilihat pada Tabel 3.2.

**Tabel 3.2** Daftar Dosen dan Asisten pada Program Studi Teknologi Informasi

ID	INISIAL	JENIS	NAMA
1	PAR	Teori	Drs. Parlaungan Ritonga, M.Hum
2	MAR	Teori	Dra. Mardiningsih, M.Si.
3	OPM	Teori	Prof. Dr. Opim Salim Sitompul, M.Sc.
4	MFS	Teori	M. Fadly Syahputra, B.Sc., M.Sc.IT
5	SFR	Teori	M. Safri Lubis, ST, M.Com
6	RYT	Teori	Riyanto Sinaga, S.Si, M.Si
7	RDW	Teori	Drs. Ridwan Azhar, M.Hum
8	DDA	Teori	Dedy Arisandi, ST, M.Kom
9	MRH	Teori	Drs. Marihat Situmorang, M.Kom
10	DGW	Teori	Dani Gunawan, ST, MT
11	SWL	Teori	Drs. Sawaluddin, M.IT
12	BAI	Teori	Baihaqi Siregar, S.Si, M.T
13	SNM	Teori	Seniman, M.Kom
14	AML	Teori	Amelia, ST, MT
15	SYT	Teori	Drs. Suyanto, M.Kom
16	MAB	Teori	M. Andri Budiman, ST, M.Comp.Sc, M.E.M
17	ROM	Teori	Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
18	SRH	Teori	Sarah Purnamawati, ST. M.Sc.

**Tabel 3.2** Daftar Dosen dan Asisten pada Program Studi Teknologi Informasi  
(Lanjutan)

ID	INISIAL	JENIS	NAMA
19	MAM	Teori	M. Anggia Muchtar, ST, MM.IT
20	SJD	Teori	Sajadin Sembiring, S.Si, M.Comp.Sc.
21	ERN	Teori	Dr. Erna Budhiarti Nababan, MIT
22	MDN	Teori	Drs. Murdeni Muis, MM
23	MYD	Teori	Dr. Mahyudin, MIT
24	ATR	Praktikum	Atras Najwan
25	IDR	Praktikum	Indra Charisma
26	EKA	Praktikum	Eka Pratiwi Goenfi
27	YNI	Praktikum	Nurrahmadayeni
28	ANS	Praktikum	Annisa Faradina
29	FTM	Praktikum	Siti Fatimah
30	TAM	Praktikum	Tamrin Immanuel P
31	GRC	Praktikum	Grace U Sira Lumanauw
32	ERC	Praktikum	Eric Suwarno
33	ATM	Praktikum	Athmanathan
34	SUR	Praktikum	M. Suryansyah Manik
35	SIL	Praktikum	Silvi Evelyn
36	ICA	Praktikum	Tengku Chairunnisa
37	RZA	Praktikum	Reza Taqyuddin
38	STG	Praktikum	Sintong TM Siregar
39	ADR	Praktikum	Andreas TSM

Komponen kedua adalah hari dan waktu. Durasi waktu kuliah adalah 50 menit untuk setiap 1 SKS. Sedangkan durasi waktu praktikum adalah 100 menit untuk tiap 1 SKS. Daftar tabel hari masuk perkuliahan dan praktikum di prodi Ilmu Komputer dan Teknologi Informasi dapat dilihat pada Tabel 3.3.

**Tabel 3.3** Daftar hari masuk perkuliahan dan praktikum di Fasilkom-TI

ID_Hari	Hari
1	Senin
2	Selasa
3	Rabu
4	Kamis
5	Jumat
6	Sabtu

Daftar tabel waktu masuk perkuliahan dan praktikum di Fakultas Ilmu Komputer dan Teknologi Informasi dapat dilihat pada Tabel 3.4.

**Tabel 3.4** Daftar waktu masuk perkuliahan dan praktikum di Fasilkom-TI

<b>ID_Waktu</b>	<b>Waktu</b>
1	08:00 - 08:50
2	08:50 - 09:40
3	09:40 - 10:30
4	10:30 - 11:20
5	11:20 - 12:10
6	12:10 - 13:00
7	13:00 - 13:50
8	13:50 - 14:40
9	14:40 - 15:30
10	15:30 - 16:20

Komponen ketiga adalah ruangan, sebagai tempat berlangsungnya kuliah dan praktikum. Daftar tabel ruangan perkuliahan dan praktikum di prodi Ilmu Komputer dapat dilihat pada Tabel 3.5.

**Tabel 3.5** Daftar Ruangan pada Program Studi Ilmu Komputer

<b>ID</b>	<b>Nama Ruangan</b>	<b>Kapasitas</b>
1	Pascal	50
2	Fortran	50
3	Basic	50
4	Java	50
5	Visual Basic	50
6	Delphi	50
7	Lab I	20
8	Lab II	20
9	Lab III	20
10	Lab IV	20

Daftar tabel ruangan perkuliahan dan praktikum di prodi Teknologi Informasi dapat dilihat pada Tabel 3.6.

**Tabel 3.6** Daftar Ruangan pada Program Studi Teknologi Informasi

<b>ID</b>	<b>Nama Ruangan</b>	<b>Kapasitas</b>
1	TI 101	60
2	TI 102	60
3	TI 103	60
4	TI 104	60
5	TI 105	60
6	TI 106	60
7	Lab Multimedia dan Pemrograman	20
8	Lab Jaringan	20
9	Lab TA	20

Komponen keempat adalah kelas peserta didik yang terdaftar dalam program studi. Daftar tabel kelas perkuliahan dan praktikum di prodi Ilmu Komputer dapat dilihat pada Tabel 3.7.

**Tabel 3.7** Daftar Kelas pada Program Studi Ilmu Komputer

<b>ID Kelas</b>	<b>Nama Kelas</b>	<b>Jenis</b>	<b>Semester</b>
1	2015 A	Teori	1
2	2015 B	Teori	1
3	2015 C	Teori	1
4	2014 A	Teori	3
5	2014 B	Teori	3
6	2014 C	Teori	3
7	2013 A	Teori	5
8	2013 B	Teori	5
9	2013 C	Teori	5
10	2102 A	Teori	7
11	2012 B	Teori	7
12	2012 C	Teori	7
13	2015 A1	Praktikum	1
14	2015 A2	Praktikum	1
15	2015 B1	Praktikum	1
16	2015 B2	Praktikum	1
17	2015 C1	Praktikum	1
18	2015 C2	Praktikum	1
19	2014 A1	Praktikum	3
20	2014 A2	Praktikum	3
21	2014 B1	Praktikum	3
22	2014 B2	Praktikum	3
23	2014 C1	Praktikum	3
24	2014 C2	Praktikum	3

**Tabel 3.7** Daftar Kelas pada Program Studi Ilmu Komputer (Lanjutan)

ID Kelas	Nama Kelas	Jenis	Semester
25	2013 A1	Praktikum	5
26	2013 A2	Praktikum	5
27	2013 B1	Praktikum	5
28	2013 B2	Praktikum	5
29	2013 C1	Praktikum	5
30	2013 C2	Praktikum	5
31	2012 A1	Praktikum	7
32	2012 A2	Praktikum	7
33	2012 B1	Praktikum	7
34	2012 B2	Praktikum	7
35	2012 C1	Praktikum	7
36	2012 C2	Praktikum	7
37	2015 A,B,C	Teori	1
38	2013 A,B,C	Teori	5
39	2012 A,B,C	Teori	7

Daftar tabel kelas perkuliahan dan praktikum di prodi Teknologi Informasi dapat dilihat pada Tabel 3.8.

**Tabel 3.8** Daftar Kelas pada Program Studi Teknologi Informasi

ID Kelas	NAMA	JENIS	SEMESTER
1	2015 A	Teori	1
2	2015 B	Teori	1
3	2015 C	Teori	1
4	2014 A	Teori	3
5	2014 B	Teori	3
6	2014 C	Teori	3
7	2013 A	Teori	5
8	2013 B	Teori	5
9	2013 C	Teori	5
10	2012 A	Teori	7
11	2012 B	Teori	7
12	2015 A1	Praktikum	1
13	2015 A2	Praktikum	1
14	2015 B1	Praktikum	1
15	2015 B2	Praktikum	1
16	2015 C1	Praktikum	1
17	2015 C2	Praktikum	1
18	2014 A1	Praktikum	3
19	2014 A2	Praktikum	3
20	2014 B1	Praktikum	3
21	2014 B2	Praktikum	3

**Tabel 3.8** Daftar Kelas pada Program Studi Teknologi Informasi (Lanjutan)

ID Kelas	NAMA	JENIS	SEMESTER
22	2014 C1	Praktikum	3
23	2014 C2	Praktikum	3
24	2013 A1	Praktikum	5
25	2013 A2	Praktikum	5
26	2013 B1	Praktikum	5
27	2013 B2	Praktikum	5
28	2013 C1	Praktikum	5
29	2013 C2	Praktikum	5
30	2012 A1	Praktikum	7
31	2012 A2	Praktikum	7
32	2012 B1	Praktikum	7
33	2012 B2	Praktikum	7

Komponen terakhir adalah mata kuliah. Daftar tabel mata kuliah dan praktikum di prodi Ilmu Komputer dapat dilihat pada Tabel 3.9.

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
1	UNI1201	Pend. Agama Islam	2015 A,B,C	1	2	Teori	SHL
2	UNI1202	Pend. Agama Kristen Katolik	2015 A,B,C	1	2	Teori	BMS
3	UNI1203	Pend. Agama Kristen Protestan	2015 A,B,C	1	2	Teori	LZS
4	UNI1204	Pend. Agama Budha	2015 A,B,C	1	2	Teori	MGL
5	ILK1200	Matematika Dasar	2015 A	1	2	Teori	OPD
6	ILK1200	Matematika Dasar	2015 B	1	2	Teori	OPD
7	ILK1200	Matematika Dasar	2015 C	1	2	Teori	MAR
8	UNI1222	Fisika I	2015 A	1	2	Teori	ACD
9	UNI1222	Fisika I	2015 B	1	2	Teori	ACD
10	UNI1222	Fisika I	2015 C	1	2	Teori	SHD
11	ILK1201	Konsep Dasar Pemrograman	2015 A	1	3	Teori	DWD
12	ILK1204	Prak. Konsep Dasar Pemrograman	2015 A1	1	1	Praktikum	AKH
13	ILK1204	Prak. Konsep Dasar Pemrograman	2015 A2	1	1	Praktikum	AKH

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
14	ILK1201	Konsep Dasar Pemrograman	2015 B	1	3	Teori	DRW
15	ILK1204	Prak. Konsep Dasar Pemrograman	2015 B1	1	1	Praktikum	AES
16	ILK1204	Prak. Konsep Dasar Pemrograman	2015 B2	1	1	Praktikum	AES
17	ILK1201	Konsep Dasar Pemrograman	2015 C	1	3	Teori	HDL
18	ILK1204	Prak. Konsep Dasar Pemrograman	2015 C1	1	1	Praktikum	DTR
19	ILK1204	Prak. Konsep Dasar Pemrograman	2015 C2	1	1	Praktikum	DTR
20	ILK1202	Logika Komputer	2015 A	1	3	Teori	SDF
21	ILK1202	Logika Komputer	2015 B	1	3	Teori	SEF
22	ILK1202	Logika Komputer	2015 C	1	3	Teori	IRY
23	ILK1203	Pengantar Ilmu Komputer	2015 A	1	2	Teori	ANS
24	ILK1205	Prak. Pengantar Ilmu Komputer	2015 A1	1	1	Praktikum	GEU
25	ILK1205	Prak. Pengantar Ilmu Komputer	2015 A2	1	1	Praktikum	GEU
26	ILK1203	Pengantar Ilmu Komputer	2015 B	1	2	Teori	HRY
27	ILK1205	Prak. Pengantar Ilmu Komputer	2015 B1	1	1	Praktikum	GUS
28	ILK1205	Prak. Pengantar Ilmu Komputer	2015 B2	1	1	Praktikum	GUS
29	ILK1203	Pengantar Ilmu Komputer	2015 C	1	2	Teori	RHA
30	ILK1205	Prak. Pengantar Ilmu Komputer	2015 C1	1	1	Praktikum	YSN
31	ILK1205	Prak. Pengantar Ilmu Komputer	2015 C2	1	1	Praktikum	YSN
32	UNI2206	Pancasila dan Kewarganegaraan	2014 A	3	3	Teori	SNU
33	UNI2206	Pancasila dan Kewarganegaraan	2014 B	3	3	Teori	ADR
34	UNI2206	Pancasila dan Kewarganegaraan	2014 C	3	3	Teori	ADR

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
35	ILK2208	Matematika Diskrit	2014 A	3	3	Teori	ASH
36	ILK2208	Matematika Diskrit	2014 B	3	3	Teori	PTS
37	ILK2208	Matematika Diskrit	2014 C	3	3	Teori	IRY
38	ILK2209	Pemrograman Berorientasi Objek	2014 A	3	3	Teori	VWJ
39	ILK2213	Prak. Pemrograman Berorientasi Objek	2014 A1	3	1	Praktikum	FQA
40	ILK2213	Prak. Pemrograman Berorientasi Objek	2014 A2	3	1	Praktikum	FQA
41	ILK2209	Pemrograman Berorientasi Objek	2014 B	3	3	Teori	HDL
42	ILK2213	Prak. Pemrograman Berorientasi Objek	2014 B1	3	1	Praktikum	DIK
43	ILK2213	Prak. Pemrograman Berorientasi Objek	2014 B2	3	1	Praktikum	DIK
44	ILK2209	Pemrograman Berorientasi Objek	2014 C	3	3	Teori	ADC
45	ILK2213	Prak. Pemrograman Berorientasi Objek	2014 C1	3	1	Praktikum	FAG
46	ILK2213	Prak. Pemrograman Berorientasi Objek	2014 C2	3	1	Praktikum	FAG
47	ILK2210	Sistem Manajemen Database	2014 A	3	2	Teori	MSL



**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
48	ILK2214	Prak. Sistem Manajemen Database	2014 A1	3	1	Praktikum	NPY
49	ILK2214	Prak. Sistem Manajemen Database	2014 A2	3	1	Praktikum	NPY
50	ILK2210	Sistem Manajemen Database	2014 B	3	2	Teori	AMS
51	ILK2214	Prak. Sistem Manajemen Database	2014 B1	3	1	Praktikum	JIP
52	ILK2214	Prak. Sistem Manajemen Database	2014 B2	3	1	Praktikum	JIP
53	ILK2210	Sistem Manajemen Database	2014 C	3	2	Teori	PLS
54	ILK2214	Prak. Sistem Manajemen Database	2014 C1	3	1	Praktikum	VAN
55	ILK2214	Prak. Sistem Manajemen Database	2014 C2	3	1	Praktikum	VAN
56	ILK2211	Probabilistik & Statistika	2014 A	3	2	Teori	DWD
57	ILK2215	Prak. Probabilitas & Statistika	2014 A1	3	1	Praktikum	DPS
58	ILK2215	Prak. Probabilitas & Statistika	2014 A2	3	1	Praktikum	DPS
59	ILK2211	Probabilistik & Statistika	2014 B	3	2	Teori	SDF
60	ILK2215	Prak. Probabilitas & Statistika	2014 B1	3	1	Praktikum	IDH
61	ILK2215	Prak. Probabilitas & Statistika	2014 B2	3	1	Praktikum	IDH
62	ILK2211	Probabilistik & Statistika	2014 C	3	2	Teori	OPD
63	ILK2215	Prak. Probabilitas & Statistika	2014 C1	3	1	Praktikum	TRH

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
64	ILK2215	Prak. Probabilitas & Statistika	2014 C2	3	1	Praktikum	TRH
65	ILK2212	Struktur Data dan Algoritma	2014 A	3	3	Teori	EMZ
66	ILK2216	Prak. Struktur Data dan Algoritma	2014 A1	3	1	Praktikum	RVP
67	ILK2216	Prak. Struktur Data dan Algoritma	2014 A2	3	1	Praktikum	RVP
68	ILK2212	Struktur Data dan Algoritma	2014 B	3	3	Teori	JTT
69	ILK2216	Prak. Struktur Data dan Algoritma	2014 B1	3	1	Praktikum	AKH
70	ILK2216	Prak. Struktur Data dan Algoritma	2014 B2	3	1	Praktikum	AKH
71	ILK2212	Struktur Data dan Algoritma	2014 C	3	3	Teori	PTS
72	ILK2216	Prak. Struktur Data dan Algoritma	2014 C1	3	1	Praktikum	MMH
73	ILK2216	Prak. Struktur Data dan Algoritma	2014 C2	3	1	Praktikum	MMH
74	ILK3220	Analisis Numerik	2013 A	5	2	Teori	SDF
75	ILK3229	Prak. Analisis Numerik	2013 A1	5	1	Praktikum	LJN
76	ILK3229	Prak. Analisis Numerik	2013 A2	5	1	Praktikum	LJN
77	ILK3220	Analisis Numerik	2013 B	5	2	Teori	SDF
78	ILK3229	Prak. Analisis Numerik	2013 B1	5	1	Praktikum	MAR
79	ILK3229	Prak. Analisis Numerik	2013 B2	5	1	Praktikum	MAR
80	ILK3220	Analisis Numerik	2013 C	5	2	Teori	SEF
81	ILK3229	Prak. Analisis Numerik	2013 C1	5	1	Praktikum	NPY
82	ILK3229	Prak. Analisis Numerik	2013 C2	5	1	Praktikum	NPY
83	ILK3221	Desain & Analisis Algoritma	2013 A	5	2	Teori	MAB

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
84	ILK3230	Prak. Desain & Analisis Algoritma	2013 A1	5	1	Praktikum	FAR
85	ILK3230	Prak. Desain & Analisis Algoritma	2013 A2	5	1	Praktikum	FAR
86	ILK3221	Desain & Analisis Algoritma	2013 B	5	2	Teori	MAB
87	ILK3230	Prak. Desain & Analisis Algoritma	2013 B1	5	1	Praktikum	KHN
88	ILK3230	Prak. Desain & Analisis Algoritma	2013 B2	5	1	Praktikum	KHN
89	ILK3221	Desain & Analisis Algoritma	2013 C	5	2	Teori	MAB
90	ILK3230	Prak. Desain & Analisis Algoritma	2013 C1	5	1	Praktikum	APY
91	ILK3230	Prak. Desain & Analisis Algoritma	2013 C2	5	1	Praktikum	APY
92	ILK3222	Mikroprosesor dan Assembly	2013 A	5	2	Teori	DLS
93	ILK3231	Prak. Mikroprosesor & Assembly	2013 A1	5	1	Praktikum	ARP
94	ILK3231	Prak. Mikroprosesor & Assembly	2013 A2	5	1	Praktikum	ARP
95	ILK3222	Mikroprosesor dan Assembly	2013 B	5	2	Teori	BSM
96	ILK3231	Prak. Mikroprosesor & Assembly	2013 B1	5	1	Praktikum	DDM
97	ILK3231	Prak. Mikroprosesor & Assembly	2013 B2	5	1	Praktikum	DDM
98	ILK3222	Mikroprosesor dan Assembly	2013 C	5	2	Teori	MHP
99	ILK3231	Prak. Mikroprosesor & Assembly	2013 C1	5	1	Praktikum	CLK

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
100	ILK3231	Prak. Mikroprosesor & Assembly	2013 C2	5	1	Praktikum	CLK
101	ILK3223	Pengolahan Citra	2013 A	5	2	Teori	SRS
102	ILK3232	Prak. Pengolahan Citra	2013 A1	5	1	Praktikum	VYN
103	ILK3232	Prak. Pengolahan Citra	2013 A2	5	1	Praktikum	VYN
104	ILK3223	Pengolahan Citra	2013 B	5	2	Teori	SRS
105	ILK3232	Prak. Pengolahan Citra	2013 B1	5	1	Praktikum	FFN
106	ILK3232	Prak. Pengolahan Citra	2013 B2	5	1	Praktikum	FFN
107	ILK3223	Pengolahan Citra	2013 C	5	2	Teori	HRY
108	ILK3232	Prak. Pengolahan Citra	2013 C1	5	1	Praktikum	LEN
109	ILK3232	Prak. Pengolahan Citra	2013 C2	5	1	Praktikum	LEN
110	ILK3224	Interaksi Manusia dan Komputer	2013 A	5	2	Teori	SUY
111	ILK3224	Interaksi Manusia dan Komputer	2013 B	5	2	Teori	HRY
112	ILK3224	Interaksi Manusia dan Komputer	2013 C	5	2	Teori	DWD
113	ILK3225	Metodologi Penelitian Informatika	2013 A	5	2	Teori	EMZ
114	ILK3225	Metodologi Penelitian Informatika	2013 B	5	2	Teori	ESN
115	ILK3225	Metodologi Penelitian Informatika	2013 C	5	2	Teori	MZL
116	ILK3226	Information Retrieval System	2013 A,B,C	5	3	Teori	PLS
117	ILK3227	Sistem Komputer Nirkabel dan Bergerak	2013 A,B,C	5	3	Teori	ANH
118	ILK3228	Jaringan Syaraf Tiruan	2013 A,B,C	5	2	Teori	AMS
119	ILK3233	Prak Jaringan Syaraf Tiruan	2013 A1	5	1	Praktikum	AMS

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
120	ILK3233	Prak Jaringan Syaraf Tiruan	2013 A2	5	1	Praktikum	AMS
121	ILK3233	Prak Jaringan Syaraf Tiruan	2013 B1	5	1	Praktikum	LEN
122	ILK3233	Prak Jaringan Syaraf Tiruan	2013 B2	5	1	Praktikum	LEN
123	ILK4201	Kecerdasan Buatan	2012 A	7	2	Teori	PLS
124	ILK4207	Prak. Kecerdasan Buatan	2012 A1	7	1	Praktikum	GUN
125	ILK4207	Prak. Kecerdasan Buatan	2012 A2	7	1	Praktikum	GUN
126	ILK4201	Kecerdasan Buatan	2012 B	7	2	Teori	EBN
127	ILK4207	Prak. Kecerdasan Buatan	2012 B1	7	1	Praktikum	GUN
128	ILK4207	Prak. Kecerdasan Buatan	2012 B2	7	1	Praktikum	GUN
129	ILK4201	Kecerdasan Buatan	2012 C	7	2	Teori	OSS
130	ILK4207	Prak. Kecerdasan Buatan	2012 C1	7	1	Praktikum	HYN
131	ILK4207	Prak. Kecerdasan Buatan	2012 C2	7	1	Praktikum	HYN
132	ILK4202	Sistem Terdistribusi	2012 A	7	2	Teori	RHA
133	ILK4202	Sistem Terdistribusi	2012 B	7	2	Teori	RHA
134	ILK4202	Sistem Terdistribusi	2012 C	7	2	Teori	DGN
135	ILK4203	Sistem Pendukung Keputusan	2012 A	7	3	Teori	MAR
136	ILK4203	Sistem Pendukung Keputusan	2012 B	7	3	Teori	MSL
137	ILK4203	Sistem Pendukung Keputusan	2012 C	7	3	Teori	SJD
138	ILK4204	Etika Profesi	2012 A	7	2	Teori	MZL
139	ILK4204	Etika Profesi	2012 B	7	2	Teori	ELW

**Tabel 3.9** Daftar Matakuliah dan Praktikum di Program Studi Ilmu Komputer  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
140	ILK4204	Etika Profesi	2012 C	7	2	Teori	ELW
141	ILK4205	Robotika	2012 A,B,C	7	2	Teori	DLS
142	ILK4206	Sistem Informasi Geografis	2012 A,B,C	7	2	Teori	ADC
143	ILK4208	Prak. Sistem Informasi Geografis	2012 A1	7	1	Praktikum	IRF
144	ILK4208	Prak. Sistem Informasi Geografis	2012 A2	7	1	Praktikum	IRF
145	ILK4208	Prak. Sistem Informasi Geografis	2012 B1	7	1	Praktikum	RAM
146	ILK4208	Prak. Sistem Informasi Geografis	2012 B2	7	1	Praktikum	RAM

Daftar tabel mata kuliah dan praktikum di prodi Teknologi Informasi dapat dilihat pada Tabel 3.10.

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
1	MAT1201	Matematika 1	2015 A	1	2	Teori	MAR
2	MAT1201	Matematika 1	2015 B	1	2	Teori	MAR
3	MAT1201	Matematika 1	2015 C	1	2	Teori	MAR
4	TIF1201	Pengantar Teknologi Informasi	2015 A	1	3	Teori	MFS
5	TIF1201	Pengantar Teknologi Informasi	2015 B	1	3	Teori	MFS
6	TIF1201	Pengantar Teknologi Informasi	2015 C	1	3	Teori	MFS
7	TIF1202	Dasar-Dasar Pemrograman	2015 A	1	3	Teori	OPM

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
8	TIF1204	Prak. Dasar-Dasar Pemrograman	2015 A1	1	1	Praktikum	IDR
9	TIF1204	Prak. Dasar-Dasar Pemrograman	2015 A2	1	1	Praktikum	IDR
10	TIF1202	Dasar-Dasar Pemrograman	2015 B	1	3	Teori	OPM
11	TIF1204	Prak. Dasar-Dasar Pemrograman	2015 B1	1	1	Praktikum	ATR
12	TIF1204	Prak. Dasar-Dasar Pemrograman	2015 B2	1	1	Praktikum	ATR
13	TIF1202	Dasar-Dasar Pemrograman	2015 C	1	3	Teori	OPM
14	TIF1204	Prak. Dasar-Dasar Pemrograman	2015 C1	1	1	Praktikum	EKA
15	TIF1204	Prak. Dasar-Dasar Pemrograman	2015 C2	1	1	Praktikum	EKA
16	TIF1203	Dasar-Dasar Web	2015 A	1	3	Teori	SFR
17	TIF1205	Prak. Dasar-Dasar Web	2015 A1	1	1	Praktikum	ANS
18	TIF1205	Prak. Dasar-Dasar Web	2015 A2	1	1	Praktikum	ANS
19	TIF1203	Dasar-Dasar Web	2015 B	1	3	Teori	SFR
20	TIF1205	Prak. Dasar-Dasar Web	2015 B1	1	1	Praktikum	YNI
21	TIF1205	Prak. Dasar-Dasar Web	2015 B2	1	1	Praktikum	YNI
22	TIF1203	Dasar-Dasar Web	2015 C	1	3	Teori	SFR
23	TIF1205	Prak. Dasar-Dasar Web	2015 C1	1	1	Praktikum	YNI
24	TIF1205	Prak. Dasar-Dasar Web	2015 C2	1	1	Praktikum	YNI
25	UNI1207	Bahasa Indonesia	2015 A	1	2	Teori	PAR
26	UNI1207	Bahasa Indonesia	2015 B	1	2	Teori	RDW
27	UNI1207	Bahasa Indonesia	2015 C	1	2	Teori	PAR
28	UNI1209	Ilmu Alamiah Dasar	2015 A	1	2	Teori	RYT
29	UNI1209	Ilmu Alamiah Dasar	2015 B	1	2	Teori	RYT
30	UNI1209	Ilmu Alamiah Dasar	2015 C	1	2	Teori	RYT

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
31	TIF2301	Pemrograman Internet 1	2014 A	3	3	Teori	DDA
32	TIF2306	Prak. Pemrograman Internet 1	2014 A1	3	1	Praktikum	GRC
33	TIF2306	Prak. Pemrograman Internet 1	2014 A2	3	1	Praktikum	GRC
34	TIF2301	Pemrograman Internet 1	2014 B	3	3	Teori	DDA
35	TIF2306	Prak. Pemrograman Internet 1	2014 B1	3	1	Praktikum	ERC
36	TIF2306	Prak. Pemrograman Internet 1	2014 B2	3	1	Praktikum	ERC
37	TIF2301	Pemrograman Internet 1	2014 C	3	3	Teori	DDA
38	TIF2306	Prak. Pemrograman Internet 1	2014 C1	3	1	Praktikum	ERC
39	TIF2306	Prak. Pemrograman Internet 1	2014 C2	3	1	Praktikum	ERC
40	TIF2302	Pengantar Teknologi Telekomunikasi	2014 A	3	2	Teori	BAI
41	TIF2302	Pengantar Teknologi Telekomunikasi	2014 B	3	2	Teori	BAI
42	TIF2302	Pengantar Teknologi Telekomunikasi	2014 C	3	2	Teori	BAI
43	TIF2303	Organisasi dan Arsitektur Komputer	2014 A	3	3	Teori	MRH
44	TIF2303	Organisasi dan Arsitektur Komputer	2014 B	3	3	Teori	MRH
45	TIF2303	Organisasi dan Arsitektur Komputer	2014 C	3	3	Teori	AML



**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
46	TIF2304	Struktur Data dan Algoritma	2014 A	3	3	Teori	SWL
47	TIF2307	Prak. Struktur Data dan Algoritma	2014 A1	3	1	Praktikum	FTM
48	TIF2307	Prak. Struktur Data dan Algoritma	2014 A2	3	1	Praktikum	FTM
49	TIF2304	Struktur Data dan Algoritma	2014 B	3	3	Teori	SWL
50	TIF2307	Prak. Struktur Data dan Algoritma	2014 B1	3	1	Praktikum	TAM
51	TIF2307	Prak. Struktur Data dan Algoritma	2014 B2	3	1	Praktikum	TAM
52	TIF2304	Struktur Data dan Algoritma	2014 C	3	3	Teori	MAB
53	TIF2307	Prak. Struktur Data dan Algoritma	2014 C1	3	1	Praktikum	FTM
54	TIF2307	Prak. Struktur Data dan Algoritma	2014 C2	3	1	Praktikum	FTM
55	TIF2305	Sistem Operasi	2014 A	3	3	Teori	DGW
56	TIF2308	Prak. Sistem Operasi	2014 A1	3	1	Praktikum	RZA
57	TIF2308	Prak. Sistem Operasi	2014 A2	3	1	Praktikum	RZA
58	TIF2305	Sistem Operasi	2014 B	3	3	Teori	DGW
59	TIF2308	Prak. Sistem Operasi	2014 B1	3	1	Praktikum	ATM
60	TIF2308	Prak. Sistem Operasi	2014 B2	3	1	Praktikum	ATM
61	TIF2305	Sistem Operasi	2014 C	3	3	Teori	SYT
62	TIF2308	Prak. Sistem Operasi	2014 C1	3	1	Praktikum	ATM
63	TIF2308	Prak. Sistem Operasi	2014 C2	3	1	Praktikum	ATM
64	TIS2301	Dasar-Dasar Kelistrikan dan Elektronika	2014 A	3	2	Teori	SNM

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
65	TIS2301	Dasar-Dasar Kelistrikan dan ElektriKA	2014 B	3	2	Teori	SNM
66	TIS2301	Dasar-Dasar Kelistrikan dan ElektriKA	2014 C	3	2	Teori	SNM
67	TIF3401	Rekayasa Perangkat Lunak	2013 A	5	3	Teori	MAM
68	TIF3401	Rekayasa Perangkat Lunak	2013 B	5	3	Teori	MAM
69	TIF3401	Rekayasa Perangkat Lunak	2013 C	5	3	Teori	MAM
70	TIF3402	Sistem Informasi Manajemen	2013 A	5	2	Teori	SRH
71	TIF3402	Sistem Informasi Manajemen	2013 B	5	2	Teori	SRH
72	TIF3402	Sistem Informasi Manajemen	2013 C	5	2	Teori	SRH
73	TIF3403	Sistem Administrasi Linux	2013 A	5	2	Teori	BAI
74	TIF3409	Prak. Sistem Administrasi Linux	2013 A1	5	1	Praktikum	STG
75	TIF4309	Prak. Sistem Administrasi Linux	2013 A2	5	1	Praktikum	STG
76	TIF3403	Sistem Administrasi Linux	2013 B	5	2	Teori	DGW
77	TIF3409	Prak. Sistem Administrasi Linux	2013 B1	5	1	Praktikum	SUR
78	TIF3409	Prak. Sistem Administrasi Linux	2013 B2	5	1	Praktikum	SUR
79	TIF3403	Sistem Administrasi Linux	2013 C	5	2	Teori	DGW
80	TIF3409	Prak. Sistem Administrasi Linux	2013 C1	5	1	Praktikum	SUR

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
81	TIF3409	Prak. Sistem Administrasi Linux	2013 C2	5	1	Praktikum	SUR
82	TIF3404	Pengantar Sistem Multimedia	2013 A	5	2	Teori	DGW
83	TIF3404	Pengantar Sistem Multimedia	2013 B	5	2	Teori	MAM
84	TIF3404	Pengantar Sistem Multimedia	2013 C	5	2	Teori	MAM
85	TIF3405	Manajemen Sistem Database	2013 A	5	2	Teori	DDA
86	TIF3407	Prak. Manajemen Sistem Database	2013 A1	5	1	Praktikum	ICA
87	TIF3407	Prak. Manajemen Sistem Database	2013 A2	5	1	Praktikum	ICA
88	TIF3405	Manajemen Sistem Database	2013 B	5	2	Teori	DDA
89	TIF3407	Prak. Manajemen Sistem Database	2013 B1	5	1	Praktikum	SIL
90	TIF3407	Prak. Manajemen Sistem Database	2013 B2	5	1	Praktikum	SIL
91	TIF3405	Manajemen Sistem Database	2013 C	5	2	Teori	DDA
92	TIF3407	Prak. Manajemen Sistem Database	2013 C1	5	1	Praktikum	ICA
93	TIF3407	Prak. Manajemen Sistem Database	2013 C2	5	1	Praktikum	ICA
94	TIF3406	Komunikasi dan Jaringan Nirkabel	2013 A	5	3	Teori	ROM
95	TIF3408	Prak. Komunikasi data dan Jaringan Nirkabel	2013 A1	5	1	Praktikum	RZA
96	TIF3408	Prak. Komunikasi data dan Jaringan Nirkabel	2013 A2	5	1	Praktikum	RZA
97	TIF3406	Komunikasi dan Jaringan Nirkabel	2013 B	5	3	Teori	ROM
98	TIF3408	Prak. Komunikasi data dan Jaringan Nirkabel	2013 B1	5	1	Praktikum	RZA
99	TIF3408	Prak. Komunikasi data dan Jaringan Nirkabel	2013 B2	5	1	Praktikum	RZA

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
100	TIF3406	Komunikasi dan Jaringan Nirkabel	2013 C	5	3	Teori	ROM
101	TIF3408	Prak. Komunikasi data dan Jaringan Nirkabel	2013 C1	5	1	Praktikum	RZA
102	TIF3408	Prak. Komunikasi data dan Jaringan Nirkabel	2013 C2	5	1	Praktikum	RZA
103	TIS3401	Manajemen Proyek TI	2013 A	5	2	Teori	SJD
104	TIS3401	Manajemen Proyek TI	2013 B	5	2	Teori	SJD
105	TIS3401	Manajemen Proyek TI	2013 C	5	2	Teori	SJD
106	TIF4501	Sistem Database Client-Server	2012 A	7	3	Teori	SRH
107	TIF4508	Prak. Sistem Database Client-Server	2012 A1	7	1	Praktikum	ADR
108	TIF4508	Prak. Sistem Database Client-Server	2012 A2	7	1	Praktikum	ADR
109	TIF4501	Sistem Database Client-Server	2012 B	7	3	Teori	SRH
110	TIF4508	Prak. Sistem Database Client-Server	2012 B1	7	1	Praktikum	ADR
111	TIF4508	Prak. Sistem Database Client-Server	2012 B2	7	1	Praktikum	ADR
112	TIF4502	Pemrograman Mobile	2012 A	7	3	Teori	ROM
113	TIF4507	Prak. Pemrograman Mobile	2012 A1	7	1	Praktikum	STG
114	TIF4507	Prak. Pemrograman Mobile	2012 A2	7	1	Praktikum	STG
115	TIF4502	Pemrograman Mobile	2012 B	7	3	Teori	ROM
116	TIF4507	Prak. Pemrograman Mobile	2012 B1	7	1	Praktikum	STG

**Tabel 3.10** Daftar Matakuliah dan Praktikum di Program Studi Teknologi Informasi  
(Lanjutan)

ID	Kode	Nama	Kelas	Semester	SKS	Jenis	Inisial
117	TIF4507	Prak. Pemrograman Mobile	2012 B2	7	1	Praktikum	STG
118	TIF4503	Administrasi dan Desain Jaringan	2012 A	7	2	Teori	BAI
119	TIF4503	Administrasi dan Desain Jaringan	2012 B	7	3	Teori	BAI
120	TIF4504	Desain Multimedia Digital	2012 A	7	3	Teori	MAM
121	TIF4504	Desain Multimedia Digital	2012 B	7	3	Teori	MAM
122	TIF4505	Sistem Pakar	2012 A	7	3	Teori	ERN
123	TIF4505	Sistem Pakar	2012 B	7	3	Teori	ERN
124	TIF4506	Pengantar Natural Language Processing	2012 A	7	3	Teori	MYD
125	TIF4506	Pengantar Natural Language Processing	2012 B	7	3	Teori	MYD
126	TIS4501	Metodologi Penelitian dan Survey	2012 A	7	2	Teori	ERN
127	TIS4501	Metodologi Penelitian dan Survey	2012 B	7	2	Teori	ERN
128	TIS4502	Keahlian Presentasi dan Komunikasi	2012 A	7	2	Teori	SRH
129	TIS4502	Keahlian Presentasi dan Komunikasi	2012 B	7	2	Teori	SRH
130	TIS4503	Pengantar Kewirausahaan	2012 A	7	2	Teori	MDN
131	TIS4503	Pengantar Kewirausahaan	2012 B	7	2	Teori	MDN

### 3.4 Proses Algoritma Genetika pada Penjadwalan

Dalam proses penjadwalan kuliah dan praktikum, ada beberapa hal penting yang harus dilakukan untuk penerapan algoritma genetika seperti pengkodean kromosom, membangkitkan populasi awal, proses seleksi, proses *crossover*, proses mutasi sampai dengan proses regenerasi yang memenuhi syarat atau tidak.

#### 3.4.1 Membangkitkan populasi awal

Teknik yang digunakan dalam membangkitkan populasi awal adalah dengan mengambil variabel dosen, kelas, SKS, ruangan, hari dan waktu, yang kemudian direpresentasikan ke dalam bentuk kromosom.

##### 3.4.1.1 Representasi kromosom

Dalam proses penjadwalan menggunakan algoritma genetika, representasi kromosom adalah hal sangat penting. Representasi kromosom akan mendefinisikan jumlah dari gen yang mewakili solusi dari masalah tersebut.

Dalam menentukan sebuah solusi permasalahan yang timbul, solusi diwakili dengan satu set parameter. Parameter ini berfungsi untuk mengatur gen yang bersatu untuk membentuk kromosom. Parameter yang digunakan adalah mata kuliah, ruang dan waktu.

Representasi kromosom pada prodi Ilmu Komputer dapat dijelaskan pada Gambar 3.1.

Kromosom 1:

ID_MK	ID_Ruang	ID_Waktu
1	5	5 8

Kromosom 2:

ID_MK	ID_Ruang	ID_Waktu
2	4	3 8

**Gambar 3.1** Representasi Kromosom

Berdasarkan gambar 3.1, dapat dijelaskan kromosom 1 berisikan ID\_MK = 1 (Pendidikan Agama Islam), di ID\_Ruangan = 5 (Visual Basic), pada ID\_Hari = 5 (Jumat) di mulai pada ID\_Waktu = 8 (13:50-14:40). Sedangkan kromosom 2 berisikan ID\_MK = 2 (Pendidikan Agama Katolik), di ID\_Ruangan = 4 (Java), pada ID\_Hari = 3 (Rabu) di mulai pada ID\_Waktu = 8 (13:50-14:40).

Sebuah individu diwakili oleh matriks  $m \times n$ , dimana  $m$  atau baris pada matriks merepresentasikan mata kuliah dan mata kuliah praktikum, sedangkan  $n$  atau kolom pada matriks merepresentasikan mata kuliah, ruang dan waktu. Pada program studi Ilmu Komputer, sebuah individu diwakili oleh matriks [146][3], sedangkan program studi Teknologi Informasi, sebuah individu diwakili oleh matriks [131][3].

Jumlah waktu yang tersedia sebanyak 58 waktu dalam seminggu. Jumlah hari yang tersedia sebanyak 6 hari, yaitu mulai Senin sampai dengan Sabtu. Jumlah waktu mulai perkuliahan dan praktikum sebanyak 10 waktu dalam sehari, mulai pukul 08.00 hingga pukul 15.30 WIB, sedangkan pada Jum'at di mulai dari pukul 12.10 WIB sampai pukul 13.50 WIB dipergunakan untuk sholat Jum'at.

Setiap Individu awal akan dibangkitkan secara acak untuk selanjutnya dilakukan proses evaluasi setiap kromosomnya. Representasi individu dapat dijelaskan pada Tabel 3.11 dan Tabel 3.12.

**Tabel 3.11** Representasi sebuah individu pada prodi Ilmu Komputer

ID_MK	Dosen	Kelas	SKS	Ruang	Waktu	
					Hari	Jam
1	54	37	2	5	5	8
2	55	37	2	1	5	6
3	56	37	2	2	5	1
4	57	37	2	1	2	1
145	16	33	1	7	1	7
146	16	34	1	10	1	6

Dari tabel 3.11 di atas, dapat dijelaskan pada matriks baris pertama yaitu : Kromosom 1 berisikan ID\_MK = 1 (Pendidikan Agama Islam), diampu oleh Dosen = 54 (Dra. Hj. Sahlia, M.Ag), kelas = 37 (2015 A,B,C), SKS = 2, Ruangan = 5 (Visual Basic), pada Hari = 5 (Jumat) di mulai pada ID\_Waktu = 8 (13:50 - 14:40).

**Tabel 3.12** Representasi sebuah individu pada Prodi Teknologi Informasi

ID_MK	Dosen	Kelas	SKS	Ruang	Waktu	
					Hari	Jam
1	2	1	2	1	5	9
2	2	2	2	4	6	2
3	2	3	2	1	2	7
4	4	1	3	2	4	2
130	22	10	2	9	2	8
131	22	11	2	9	6	4

Dari tabel 3.12 di atas, dapat dijelaskan pada matriks baris pertama yaitu : Kromosom 1 berisikan ID\_MK = 1 (Matematika 1), diampu oleh Dosen = 2 (Dra. Mardiningsih, M.Si.), Kelas = 1 (2015 A), SKS = 2, di Ruangan = 1 (TI 101), pada ID\_Hari 5 (Jumat) di mulai pada ID\_Waktu = 9 (14:40 - 15:30).

Representasi individu jadwal perkuliahan dan praktikum dibuat secara acak dengan cara berikut.

1. Buat nilai gen kromosom sesuai dengan jumlah mata kuliah dan praktikum.
2. Acak nilai gen ID\_MK dari 1 sampai Maks,
3. Acak nilai gen ID\_ruang dari 1 sampai Maks,
4. Acak nilai gen ID\_Waktu, terdiri [hari][jam], dari [1 sampai 6] [1 sampai 10].



5. Nilai ID\_Dosen, ID\_Kelas, dan SKS harus terikat kepada gen ID\_MK karena sesuai dengan mata kuliah yang diajarkan dosen.
6. Simpan nilai gen ID\_MK, ID\_ruang, dan ID\_Waktu sebagai individu awal.

#### **3.4.1.2 Inisialisasi populasi**

Proses inisialisasi populasi dibuat dengan cara membangkitkan populasi acak tanpa memperhatikan nilai fungsi *fitness*. Proses ini merupakan proses yang mengkodekan informasi data ke dalam slot kromosom. Adapun tahapan proses dari inisialisasi populasi adalah sebagai berikut.

1. Inisialisasi dibuat dengan memasukkan nilai parameter genetika yaitu populasi awal.
2. Populasi awal =  $n$  (jumlah individu).
3. Inisialisasi populasi dibuat dari representasi kromosom sebanyak  $n$  kali.

Misalkan jumlah populasi awal ( jumlah individu awal ) = 3. Inisialisasi populasi dalam prodi Ilmu Komputer dapat dilihat seperti pada Tabel 3.13.

**Tabel 3.13** Inisialisasi Populasi pada Prodi Ilmu Komputer

Individu-1 :

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	54	37	2	5	5	8
2	55	37	2	1	5	6
3	56	37	2	2	5	1
4	57	37	2	1	2	1
145	16	33	1	7	1	7
146	16	34	1	10	1	6

Individu-2 :

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	54	37	2	1	3	1
2	55	37	2	4	2	3
3	56	37	2	6	2	9
4	57	37	2	1	3	2
145	16	33	1	10	1	2
146	16	34	1	9	1	6

Individu-3 :

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	54	37	2	6	6	1
2	55	37	2	5	3	8
3	56	37	2	4	4	3
4	57	37	2	9	1	1
145	16	33	1	5	5	6
146	16	34	1	9	5	8

Inisialisasi populasi prodi Teknologi Informasi dapat dilihat pada Tabel 3.14.

**Tabel 3.14** Inisialisasi Populasi pada Prodi Teknologi Informasi

Individu-1 :

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	2	1	2	2	2	9
2	2	2	2	6	2	9
3	2	3	2	3	5	9
4	4	1	3	5	6	6
130	22	10	2	5	3	1
131	22	11	2	5	6	3

Individu-2 :

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	2	1	2	4	6	10
2	2	2	2	1	4	7
3	2	3	2	3	3	3
4	4	1	3	3	3	3
130	22	10	2	3	6	1
131	22	11	2	4	3	10

Individu-3 :

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	2	1	2	3	2	6
2	2	2	2	5	5	4
3	2	3	2	1	1	4
4	4	1	3	5	2	5
130	22	10	2	5	6	4
131	22	11	2	1	5	4

### 3.4.2 Evaluasi *fitness*

*Fitness* akan dihitung berdasarkan jumlah pelanggaran atau pinalti yang terjadi pada setiap slot kromosom dalam suatu individu. Suatu kromosom dikatakan terkena pinalti adalah bila melanggar salah satu dari syarat berikut.

1. Terdapat pengajar yang mengajar pada hari dan waktu yang sama.
2. Terdapat kelas yang masuk kuliah/praktikum pada hari dan waktu yang sama.
3. Terdapat ruangan yang digunakan pada hari dan waktu yang sama.
4. Terdapat dosen yang mengajar pada jam sibuknya.
5. Terdapat perkuliahan yang dilakukan pada ruangan Laboratorium.
6. Terdapat perkuliahan selama waktu pelaksanaan Sholat Jumat.

Kromosom yang terkena penalti akan dikelompokkan berdasarkan pelanggaran yang dilakukan. Dalam penelitian ini, sebuah kromosom yang terkena pinalti memiliki nilai sama dengan kromosom yang lainnya. Sehingga, nilai fitness dapat dihitung melalui Rumus 3.1.

$$Fitness = \frac{1}{1 + (\sum_{i=1}^6 p_i)} \quad (3.1)$$

Misalkan individu yang dipilih untuk proses evaluasi adalah salah satu individu dari prodi Ilmu Komputer seperti pada Tabel 3.15 hingga Tabel 3.17.

**Tabel 3.15** Evaluasi Fitness pada Individu-1

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	54	37	2	5	5	8
2	55	37	2	1	5	6
3	56	37	2	2	5	1
4	57	37	2	1	2	1
145	16	33	1	7	1	7
146	16	34	1	10	1	6

Berdasarkan pengecekan penalti individu-1 pada Tabel 3.15, dapat dilihat dari kromosom 2 terkena penalti ke-6, karena jadwalnya berada pada waktu sholat Jumat. Selain itu, kromosom 145 juga terkena penalti terhadap kromosom 146. Dimana dosen yang sama mengajar pada hari dan jam yang sama (penalti ke-1). Sehingga, didapat nilai fitness pada individu-1  $= \frac{1}{1+(1+0+0+0+0+1)} = \frac{1}{3} = 0.333$ .

**Tabel 3.16** Evaluasi Fitness pada Individu-2

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	54	37	2	1	3	1
2	55	37	2	4	2	3
3	56	37	2	6	2	9
4	57	37	2	1	3	2
145	16	33	1	10	1	2
146	16	34	1	9	1	6

Berdasarkan pengecekan penalti individu-2 pada Tabel 3.16, dapat dilihat dari kromosom 1 mengalami bentrok dengan kromosom 4, dimana ruangan dipakai pada waktu yang sama (penalti ke-3). Sehingga, didapat nilai fitness pada individu-2 =

$$\frac{1}{1+(0+0+1+0+0+0)} = \frac{1}{2} = 0.5.$$

**Tabel 3.17** Evaluasi Fitness pada Individu-3

ID_Mata kuliah	Dosen	Kelas	SKS	ID_Ruang	ID_Waktu	
					Hari	Jam
1	54	37	2	6	6	1
2	55	37	2	5	3	8
3	56	37	2	4	4	3
4	57	37	2	9	1	1
145	16	33	1	5	5	6
146	16	34	1	9	5	8

Berdasarkan pengecekan penalti individu-3 pada Tabel 3.17, dapat dilihat dari kromosom 4 terkena penalti ke-5, karena perkuliahan ID\_MK = 4 (Pend. Agama Buddha) dilakukan pada ruangan = 9 (Lab III). Selain itu, kromosom 145 juga terkena penalti terhadap kromosom 146. Dimana dosen yang sama mengajar pada hari dan jam yang sama (penalti-1). Selain itu, kromosom 145 juga terkena penalti ke-6, karena mengajar pada waktu sholat jumat. Sehingga, didapat nilai fitness pada individu-3 =

$$\frac{1}{1+(1+0+0+0+1+1)} = \frac{1}{4} = 0.25.$$

Tahapan proses dari evaluasi *fitness* adalah sebagai berikut:

1. Setiap kromosom dalam satu individu di cek masing-masing nilai gen ID\_MataKuliah, ID\_Dosen, ID\_Kelas, ID\_SKS, ID\_Ruang, ID\_Hari dan ID\_Waktu terkena penalti atau tidak dengan cara mengecek dari kromosom 1 sampai dengan kromosom terakhir.
2. Hitung jumlah kromosom yang terkena penalti, kemudian masukkan nilai ke dalam fungsi *fitness*.

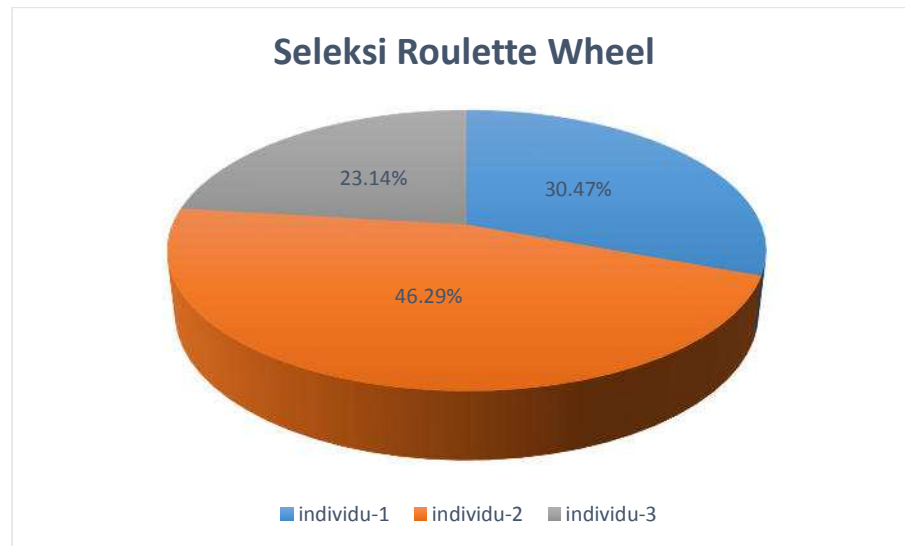
### 3.4.3 Seleksi

Dalam penelitian ini, metode yang digunakan adalah seleksi roda *roulette (roulette wheel selection)*. Pada seleksi ini, orang tua yang akan dipilih berdasarkan nilai *fitness* yang dimilikinya, semakin besar nilai *fitness*-nya akan mendapatkan kemungkinan yang lebih besar untuk terpilih sebagai induk. Nilai *fitness* pada Tabel 3.18, di dapat dari persamaan Rumus 3.1. Besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*-nya seperti pada Tabel 3.18.

**Tabel 3.18** Persentase *Fitness* pada prodi Ilmu Komputer

Individu	Fitness	Persentase
1	0.333	30.47%
2	0.50	46.29%
3	0.25	23.14%

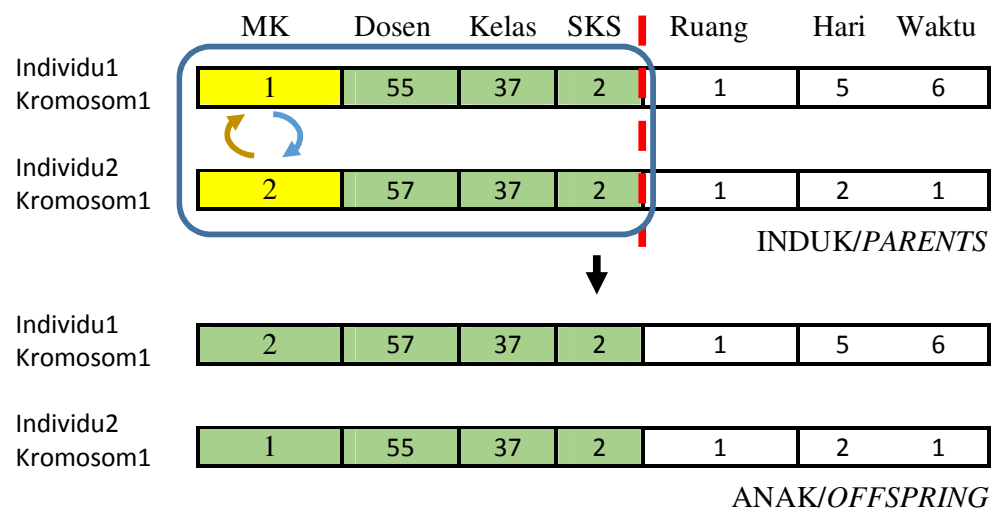
Pada Gambar 3.2 dapat dilihat probabilitas terpilihnya individu untuk dilakukan proses seleksi. Individu 2 memiliki probabilitas yang terbesar, yaitu 46.29%, terpilih sebagai induk pertama pada pemilihan individu untuk proses *crossover* kromosom. Dan individu 1 dengan probabilitas 30.47 % terpilih sebagai induk kedua untuk pembentukan keturunan baru pada pemilihan individu untuk proses *crossover* kromosom.



**Gambar 3.2** Probabilitas Seleksi *Roulette Wheel*

#### 3.4.4 *Crossover*

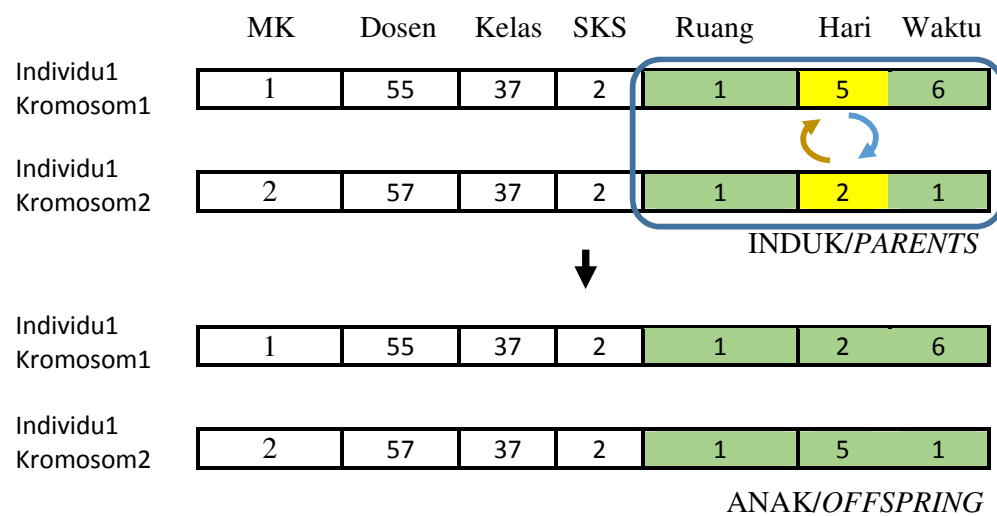
Apabila proses seleksi telah dilaksanakan dan sudah terpilih induk baru, maka tahapan selanjutnya dari operator algoritma genetika adalah *crossover*. *Crossover* adalah cara mengkombinasikan gen-gen induk untuk menghasilkan keturunan baru. *Crossover* yang digunakan pada penelitian ini adalah *crossover* satu titik. Pada *crossover* ini dilakukan dengan cara menukar nilai gen pada posisi gen yang sama dari kedua induk. Tahapan proses *crossover* kromosom pada prodi Ilmu Komputer dapat dilihat pada Gambar 3.3.



**Gambar 3.3** Ilustrasi *Crossover* Satu Titik

### 3.4.5 Mutasi

Tahapan selanjutnya setelah melakukan proses *crossover* yaitu mutasi. Mutasi yang digunakan pada penelitian ini adalah *swap mutation*. Setelah mendapatkan dua kromosom baru yang merupakan hasil *crossover*, mutasi akan dilakukan dengan cara memilih dua posisi gen dari kromosom dan kemudian nilainya saling dipertukarkan. Tahapan proses mutasi kromosom dua individu induk pada prodi Ilmu Komputer dapat dilihat pada Gambar 3.4.



Gambar 3.4 Ilustrasi *Swap Mutation*

### 3.5 Perancangan Sistem

Perancangan sistem bertujuan untuk menentukan hasil akhir dari *software* yang akan dibangun dengan melakukan beberapa proses untuk memperoleh hasil tersebut. Pada *software* penjadwalan perkuliahan dan praktikum dengan menggunakan algoritma genetika, tahap perancangan yang dilakukan mencakup perancangan data, rancangan arsitektur dalam bentuk DFD (*Data Flow Diagram*), skema relasi, rancangan antarmuka (*interface*), rancangan prosedural berupa rancangan bagan alir (*flowchart*).



### 3.5.1 Rancangan data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model data yang akan digunakan. Perancangan data terbagi menjadi dua yaitu skema relasi dan struktur tabel. Di dalam pembuatan aplikasi penjadwalan mata kuliah dan praktikum, *database* sangat diperlukan. Karena untuk menyusun jadwal mata kuliah dan praktikum, penulis membutuhkan komponen data penjadwalan kuliah dan praktikum yang disimpan dalam bentuk *database*.

#### 3.5.1.1 Struktur tabel

Didalam pembuatan program dibutuhkan suatu spesifikasi tabel yang dimaksudkan untuk dapat melakukan kegiatan-kegiatan dalam pengaturan pencarian data. Oleh karena itu, sistem pengolahan data ini membutuhkan spesifikasi tabel.

Tabel-tabel yang penulis gunakan yaitu tabel dosen, tabel hari, tabel kelas, tabel jadwal, tabel matkul, tabel ruang, tabel sibuk dan tabel waktu untuk menjalankan proses penjadwalan menggunakan algoritma genetika. Struktur tabel-tabel tersebut dapat dilihat pada Tabel 3.19 sampai 3.26.

**Tabel 3.19** Struktur Tabel Dosen

Nama Field	Jenis Field	Ukuran	Keterangan
ID	Int	10	
INISIAL	Text	3	
JENIS	Text	10	Teori/Praktikum
NAMA	Text	50	

**Tabel 3.20** Struktur Tabel Hari

Nama Field	Jenis Field	Ukuran	Keterangan
ID	Int	10	
HARI	Text	10	

**Tabel 3.21** Struktur Tabel Kelas

<b>Nama Field</b>	<b>Jenis Field</b>	<b>Ukuran</b>	<b>Keterangan</b>
ID	Int	10	
NAMA	Text	10	
JENIS	Text	10	Teori/Praktikum
SEMESTER	Int	3	

**Tabel 3.22** Struktur Tabel Jadwal

<b>Nama Field</b>	<b>Jenis Field</b>	<b>Ukuran</b>	<b>Keterangan</b>
ID	Int	10	
KODE	Text	10	
MATKUL	Text	50	
SEMESTER	Int	3	
SKS	Int	3	
DOSEN	Text	50	
KELAS	Text	10	
RUANG	Text	10	
HARI	Text	10	
WAKTU	Text	15	

**Tabel 3.23** Struktur Tabel Matkul

<b>Nama Field</b>	<b>Jenis Field</b>	<b>Ukuran</b>	<b>Keterangan</b>
ID	Int	10	
KODE	Text	10	
NAMA	Text	50	
KELAS	Text	10	
SEMESTER	Int	3	
SKS	Int	3	
JENIS	Text	10	Teori/Praktikum
DOSEN	Text	3	Inisial

**Tabel 3.24** Struktur Tabel Ruang

Nama Field	Jenis Field	Ukuran	Keterangan
ID	Int	10	
NAMA	Text	10	
KAPASITAS	Int	10	
JENIS	Text	10	Teori/Praktikum

**Tabel 3.25** Struktur Tabel Sibuk

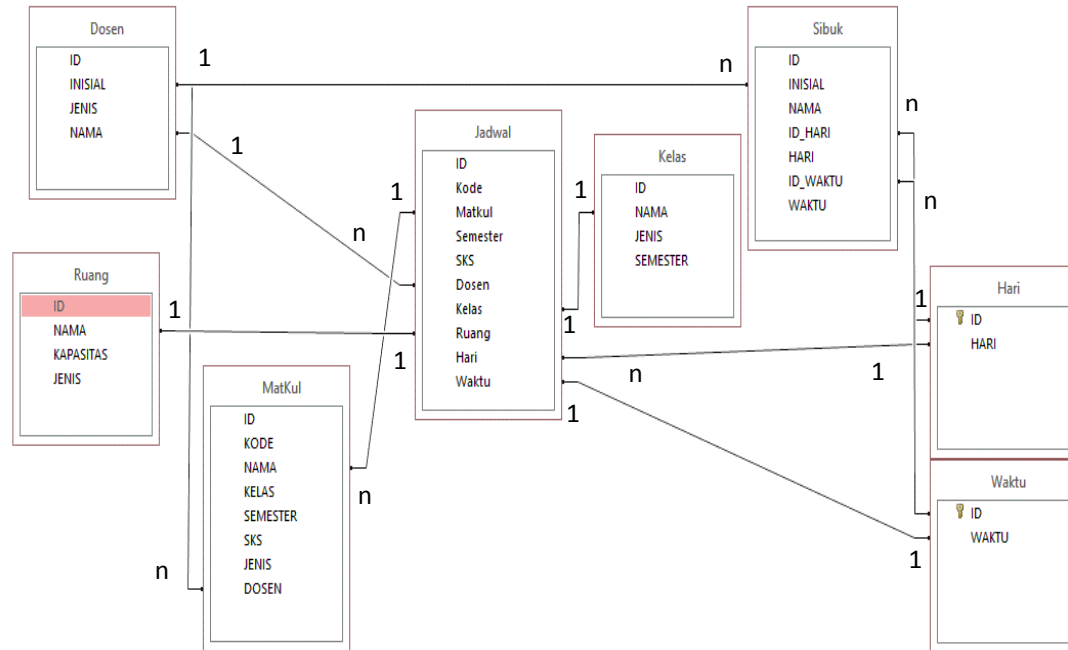
Nama Field	Jenis Field	Ukuran	Keterangan
ID	Int	10	
INISIAL	Text	3	
NAMA	Text	50	
ID_HARI	Int	10	
HARI	Text	10	
ID_WAKTU	Int	10	
WAKTU	Text	15	

**Tabel 3.26** Struktur Tabel Waktu

Nama Field	Jenis Field	Ukuran	Keterangan
ID	Int	10	
WAKTU	Text	15	

### 3.5.1.2 Skema relasi

Skema relasi antar tabel merupakan pengelompokkan data menjadi tabel-tabel yang menunjang entitas dan relasi yang berfungsi untuk mengakses data item dari setiap *database*. Relasi antar tabel untuk sistem yang akan di buat dapat di lihat pada Gambar 3.5.



**Gambar 3.5** Skema Relasi Database Penjadwalan

### 3.5.2 Rancangan arsitektural

Pada bagian ini akan digambarkan tahap perancangan arsitektur dengan metode perancangan terstruktur yaitu berupa *Data Flow Diagram* (DFD).

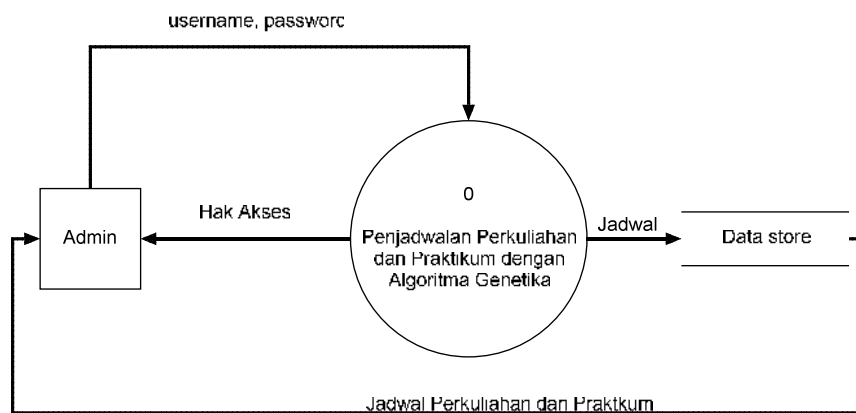
#### 3.5.2.1 Perancangan DFD (*data flow diagram*)

DFD (*Data Flow Diagram*) merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Model fungsional ini berfungsi membantu memahami cara kerja sistem dan hubungan setiap proses dalam sistem secara terstruktur dan logis.

Pada penelitian ini, konsep data flow diagram yang digunakan menurut Yourdan dan DeMarco.

### 3.5.2.2 Diagram Konteks

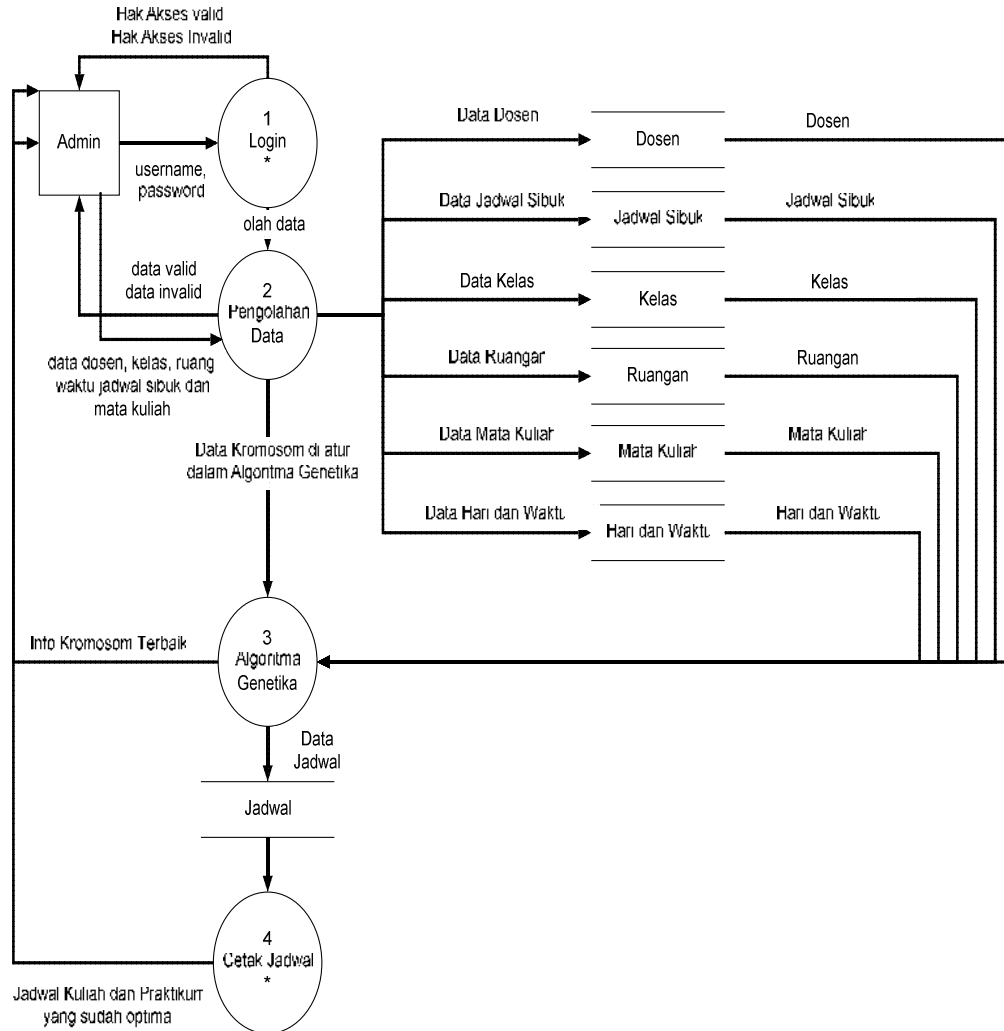
Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks disebut juga DFD level 0. Bentuk diagram konteks pada aplikasi penjadwalan ini dapat dilihat pada Gambar 3.6.



**Gambar 3.6** Diagram Konteks

### 3.5.2.3 DFD level 1

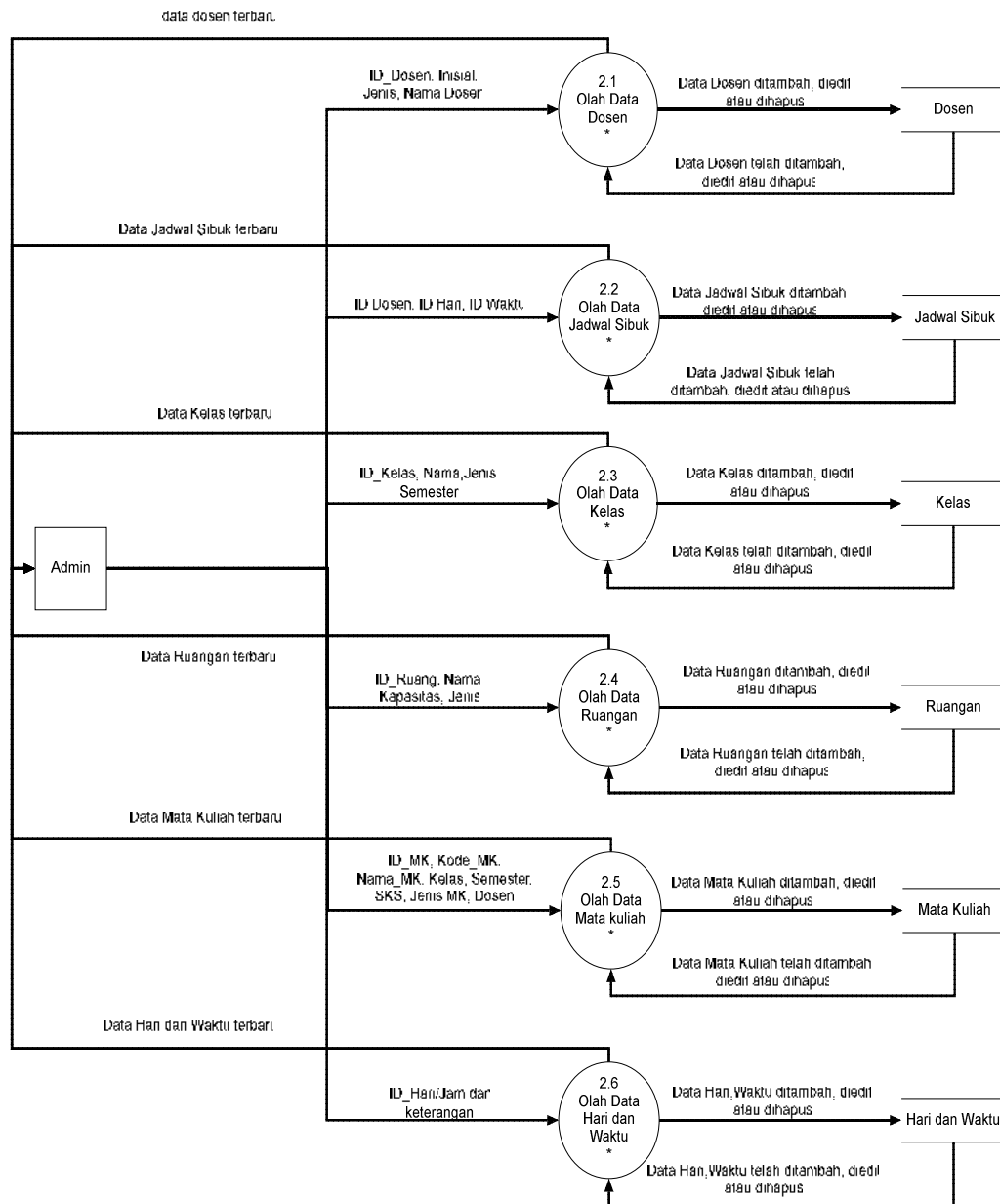
DFD level 1 menjelaskan proses utama didalam membuat keseluruhan sistem aplikasi. Diagram level 1 berfungsi menggambarkan hubungan antara entitas luar, berupa masukan dan keluaran sistem. Dari gambar dapat dilihat bahwa terdapat sebuah entitas luar yaitu *admin*. Pada aplikasi ini, *admin* memasukkan data-data utama yang berkaitan dengan penjadwalan seperti data dosen, data jadwal sibuk, data kelas, data ruangan, data mata kuliah, serta data hari dan waktu. Bentuk DFD level 1 pada aplikasi penjadwalan ini dapat dilihat pada Gambar 3.7.



**Gambar 3.7 DFD Level 1**

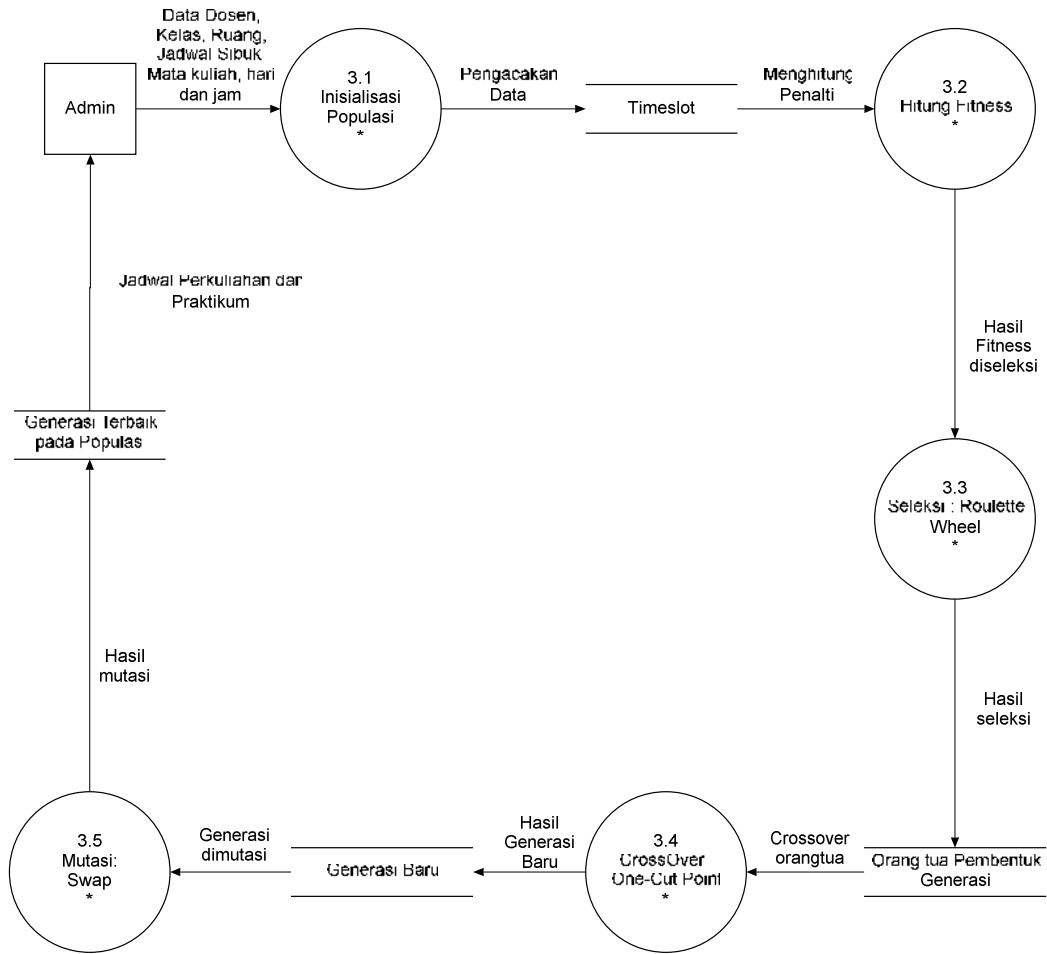
### 3.5.2.4 DFD level 2

Berdasarkan DFD level 1 pada gambar 3.7, maka proses 2 dapat digambarkan dalam Gambar 3.8.



**Gambar 3.8 DFD Level 2 Proses 2**

Berdasarkan DFD level 1 pada gambar 3.7, maka proses 3 dapat digambarkan dalam Gambar 3.9.



**Gambar 3.9** DFD Level 2 Proses 3

### 3.5.2.5 Kamus data

Kamus data merupakan sekumpulan data yang terdapat pada sistem, dimana data disusun untuk memudahkan selama proses analisis dan desain sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen data store. Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data didasarkan pada alur data yang terdapat pada DFD. Fungsi dari kamus data menjelaskan lebih detail mengenai diagram alir data yang mencakup proses, data flow, dan data store.



Kamus data yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.27 berikut.

**Tabel 3.27** Kamus Data penjadwalan

<b>Nama</b>	<b>Tipe Data</b>	<b>Keterangan</b>
<i>File Dosen</i>	<i>File</i>	<i>File dosen yang diinput</i>
<i>File Jadwal Sibuk</i>	<i>File</i>	<i>File jadwal sibuk yang diinput</i>
<i>File Kelas</i>	<i>File</i>	<i>File Kelas yang diinput</i>
<i>File Ruang</i>	<i>File</i>	<i>File Ruang yang diinput</i>
<i>File Mata Kuliah</i>	<i>File</i>	<i>File Mata Kuliah yang diinput</i>
<i>File Hari dan Waktu</i>	<i>File</i>	<i>File Hari dan Waktu yang diinput</i>
<i>Fitness</i>	<i>Double</i>	Hasil keseluruhan penalti dari setiap individu
Kromosom	<i>Integer</i>	Hasil dari Generate Random ID_MK, ID_Ruangan, ID_Waktu, <i>Crossover</i> gen ID_Ruangan, ID_Waktu, dan mutasi nilai gen
Inisialisasi Populasi	<i>Integer</i>	<i>Generate Random</i> ID_MK, ID_Ruangan, dan ID_Waktu
<i>Crossover</i>	<i>Integer</i>	Menyilangkan gen ID_Ruangan dan ID_Waktu pada kromosom induk
Mutasi	<i>Integer</i>	Menukarkan nilai gen ID_Dosen, ID_Sibuk, ID_Kelas, ID_Ruangan, ID_Waktu pada kromosom anak setelah proses <i>crossover</i>

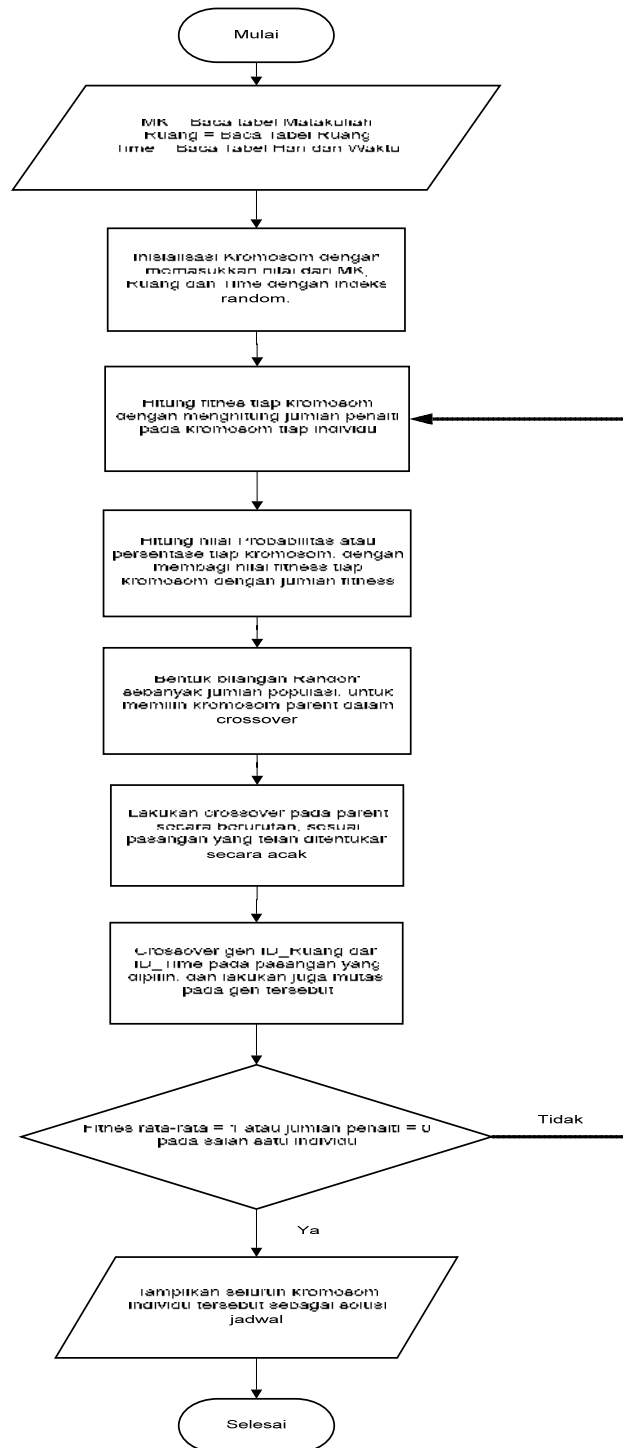
### **3.5.3 Rancangan prosedural**

Perancangan prosedural akan dibantu dengan menggunakan bagan alir (*flowchart*). *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari sebuah program. *Flowchart* menolong analisis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

#### **3.5.3.1 Pemodelan sistem**

Model dasar *flowchart* algoritma genetika yang akan dibuat pada penelitian ini dapat digambarkan pada Gambar 3.10.





**Gambar 3.11** Flowchart sistem penjadwalan

Selain menggunakan *flowchart*, pemodelan sistem juga dapat dilakukan dengan *pseudocode*. Bentuk *pseudocode* algoritma genetika adalah sebagai berikut.

```

Bangkitkan populasi awal P(0);
t=0;
while fitness<>1 or totalpenalti<>0 or maksgenerasi<>500
    Evaluasi P(t);
    I(t) = seleksi P(t);
    If random < Pc          %Pc = prob. Crossover
        A(t) = crossover (I(t));
        If random < Pm      %Pc = prob. Mutasi
            A(t) = mutasi (A(t));
        Endif
    Endif
t=t+1;
endwhile

output terbaik berada di dalam P(t)

```

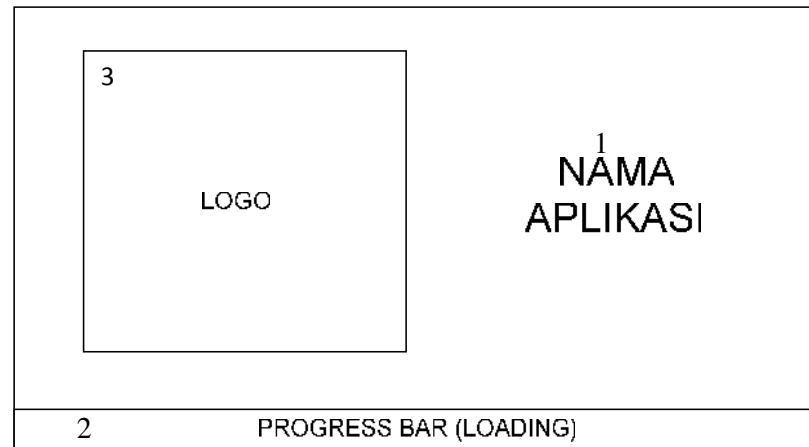
### 3.5.4 Rancangan *interface*

Perancangan *interface* atau antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi. Agar aplikasi dapat digunakan dengan baik oleh pengguna, perlu dibuat rancangan antarmuka yang *user friendly*.

#### 3.5.4.1 Rancangan *splash screen*

*Splash Screen* adalah tampilan yang umum muncul ketika awal aplikasi telah dijalankan. Tampilan ini berisikan berbagai macam, tergantung kreatifitas dan kemauan si perancang aplikasi. Tetapi, umumnya, tampilan ini berisikan logo dan nama dari aplikasi yang dibuat.

Rancangan tampilan *splash screen* pada aplikasi penjadwalan ini, dapat dilihat pada Gambar 3.12.



**Gambar 3.12** Rancangan Tampilan *Splash Screen*

Keterangan Gambar 3.12:

1. Label
2. Progress bar
3. PictureBox

#### 3.5.4.2 Rancangan *form login*

**Gambar 3.13** Rancangan Tampilan Form *Login*

Keterangan Gambar 3.13:

1. Label
2. Text Box
3. Linked Label
4. Picture Box

### 3.5.4.3 Rancangan *form* lupa password

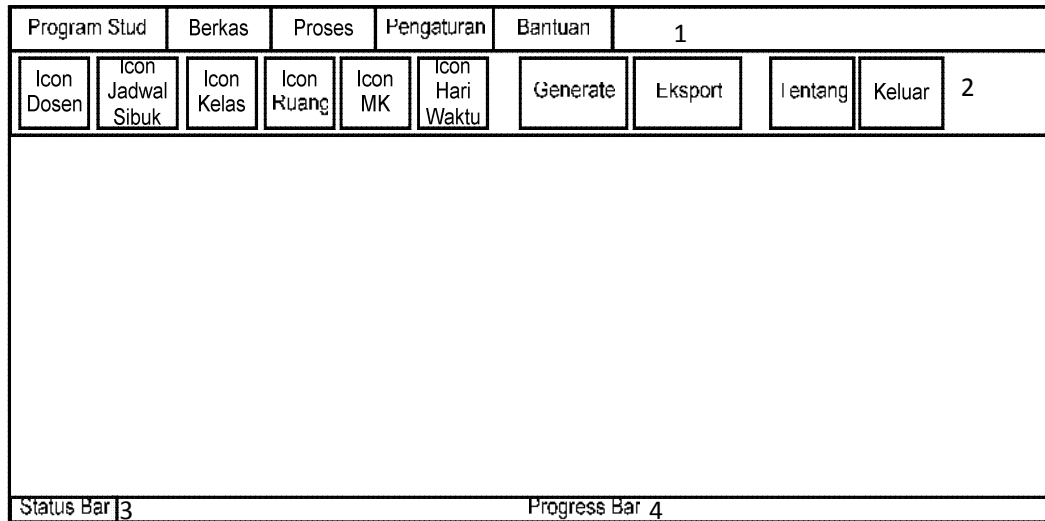
**Gambar 3.14** Rancangan Tampilan Form Lupa Password

Keterangan Gambar 3.14:

1. Label
2. Text Box
3. Combo Box
4. Picture Box

### 3.5.4.4 Rancangan menu utama

Rancangan *form* utama dalam program ini terdiri dari 5 menu utama, yaitu menu Program Studi, Berkas, Proses, Pengaturan dan Bantuan. Menu Program studi terdiri dari submenu Pilih program studi dan Keluar. Menu Berkas terdiri dari sub menu Dosen, Jadwal Sibuk, Kelas, Ruangan, Mata Kuliah dan Hari Waktu. Menu Proses terdiri dari submenu Genetika dan *Eksport*. Menu Pengaturan terdiri dari submenu *Cascade*, *Tile Horizontal* dan *Tile Vertical*. Sedangkan menu Bantuan terdiri dari submenu Tentang. Rancangan tampilan menu utama dapat dilihat pada Gambar 3.15.

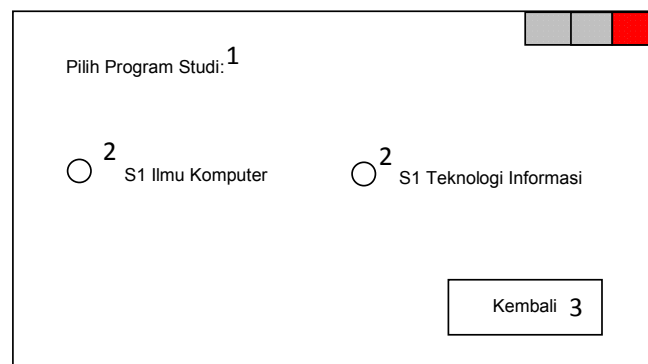


**Gambar 3.15** Rancangan Tampilan Menu Utama

Keterangan Gambar 3.15:

1. Menu Strip
2. Tool Strip
3. Status Strip
4. Progress Bar

#### 3.5.4.5 Rancangan submenu Pilih Program Studi



**Gambar 3.16** Rancangan Tampilan Pilih Program Studi

Keterangan Gambar 3.16:

1. Label
2. Radio Button
3. Button



### 3.5.4.6 Rancangan tampilan form Dosen

The diagram illustrates the layout of the 'Dosen' form. It features a main window with a standard Windows title bar. The primary data entry area is a 'Data Dosen' group box (1). Within this box, there are labels (2) for 'ID2', 'Inisial2', and 'Nama2', each followed by a text box (3) containing the number '3'. Below these, there are radio buttons (4) for 'Jenis2', with options 'Teori4' and 'Praktikum4'. At the bottom of the form, there are five buttons (5) for 'Tambah', 'Hapus', 'Ubah', 'Simpan', and 'Batal'. To the right of the data entry section is a large rectangular area (6) designated for the 'DATAGRID DOSEN'.

**Gambar 3.17** Rancangan Tampilan form isian Dosen

Keterangan Gambar 3.17:

1. GroupBox
2. Label
3. TextBox
4. Radio Button
5. Button
6. DataGried View

### 3.5.4.7 Rancangan tampilan form Jadwal Sibuk/Tak Sedia

The form is titled "Data Ketaksediaan Dosen 1". It contains the following elements:

- Fields:**
  - ID<sup>2</sup>: Text box containing "3"
  - Dosen<sup>2</sup>: Dropdown menu showing "4"
  - Hari<sup>2</sup>: Dropdown menu showing "4"
  - Waktu<sup>2</sup>: Dropdown menu showing "4"
- Buttons:**
  - Tambah (5)
  - Hapus (5)
  - Ubah (5)
  - Simpan (5)
  - Batal (5)
- DataGrid Area:**
  - Label: 6
  - Text: DATAGRID TAK SEDIA DOSEN/ SIBUK

**Gambar 3.18** Rancangan Tampilan form isian Jadwal Sibuk/Tak Sedia

Keterangan Gambar 3.18:

1. GroupBox
2. Label
3. TextBox
4. ComboBox
5. Button
6. DataGried View

### 3.5.4.8 Rancangan tampilan form Kelas

**Gambar 3.19** Rancangan Tampilan form isian Kelas

Keterangan Gambar 3.19:

- |                 |                  |
|-----------------|------------------|
| 1. GroupBox     | 5. Button        |
| 2. Label        | 6. DataGrid View |
| 3. TextBox      |                  |
| 4. Radio Button |                  |

### 3.5.4.9 Rancangan tampilan form Ruangan

**Gambar 3.20** Rancangan Tampilan form isian Ruangan

Keterangan Gambar 3.20:

1. GroupBox
2. Label
3. TextBox
4. Radio Button
5. Button
6. DataGried View

#### 3.5.4.10 Rancangan tampilan form Mata Kuliah

Form isian Mata Kuliah dengan elemen-elemen berikut:

- GroupBox 1:** Data Mata Kuliah
- Input Fields:** ID2 (3), Kode (2), Nama (2), Kelas (2), Semester (2), SKS (2), Dosen (2), Jenis (2).
- Radio Buttons:** Teori (5), Praktikum (5).
- Buttons:** Tambah (6), Hapus (6), Ubah (6), Simpan (6), Batal (6), Simpan (6).
- DataGrid View 7:** DATAGRID MATAKULIAH

**Gambar 3.21** Rancangan Tampilan form isian Mata Kuliah

Keterangan Gambar 3.21:

1. GroupBox
2. Label
3. TextBox
4. ComboBox
5. Radio Button
6. Button
7. DataGrid View

### 3.5.4.11 Rancangan tampilan form Hari dan Waktu

**Gambar 3.22** Rancangan Tampilan form isian Hari dan Waktu

Keterangan Gambar 3.22:

1. GroupBox
2. Label
3. TextBox
4. Radio Button
5. Button
6. DataGrid View

### 3.5.4.12 Rancangan tampilan form Genetika

The diagram illustrates the layout of the Genetic Form. It features a main container with a title bar (gray and red buttons). The interface is organized into several functional areas:

- Parameter Input Section (Top Left):** A group box (1) contains labels (2) for 'Panjang Kromosom' and 'Jumlah Kromosom', both set to 2. Below these are three input fields (3) for 'Populasi', 'Crossover %', and 'Mutasi %', each set to 3.
- Action Buttons (Bottom Left):** Two buttons (4) labeled 'Export' and 'Proses' are positioned below the input fields.
- Data Display Areas (Right and Bottom):**
  - A large area (5) on the top right is designated for 'DATAGRID JADWAL'.
  - An area (5) on the bottom right is designated for 'DATAGRID KROMOSOM'.
  - A text box (3) at the bottom center is labeled 'TEXTBOX (Menampilkan Keterangan Proses)'.

**Gambar 3.23** Rancangan Tampilan form Genetika

Keterangan Gambar 3.23:

1. GroupBox
2. Label
3. TextBox
4. Button
5. DataGrid View

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN**

Setelah melalui tahap analisis dan perancangan sistem, maka tahap selanjutnya adalah tahap implementasi dan pengujian sistem. Implementasi dan pengujian pada bab 4 dikerjakan sesuai dengan perancangan yang telah dijabarkan di bab 3. Untuk mengetahui apakah implementasi berhasil atau tidak, diperlukan sebuah pengujian. Berikut ini hasil implementasi dan pengujian dari aplikasi yang telah dibangun.

#### **4.1 Implementasi Sistem**

Implementasi sistem adalah penerapan hasil perancangan yang telah diuraikan pada bab sebelumnya. Sistem dirancang dengan menggunakan bahasa pemrograman *Visual Basic 2010* dan *database* yang digunakan adalah Microsoft Access 2013.

##### **4.1.1 Form splash screen**

Splash screen adalah *form* pembuka yang biasanya muncul ketika memulai sebuah aplikasi. Tampilan *form* splash screen pada program penjadwalan perkuliahan dan praktikum dapat dilihat pada Gambar 4.1.



**Gambar 4.1** Tampilan Splash Screen

Pada tampilan splash screen, hanya menampilkan judul aplikasi, logo serta proses *loading* yang terjadi.

#### 4.1.2 Form Login

Pada *form login*, *user* harus mengisi username dan password yang tepat untuk dapat masuk ke *form* berikutnya. Tampilan *form login* dapat dilihat seperti Gambar 4.2.



**Gambar 4.2** Tampilan Login



#### 4.1.3 Form Lupa Password

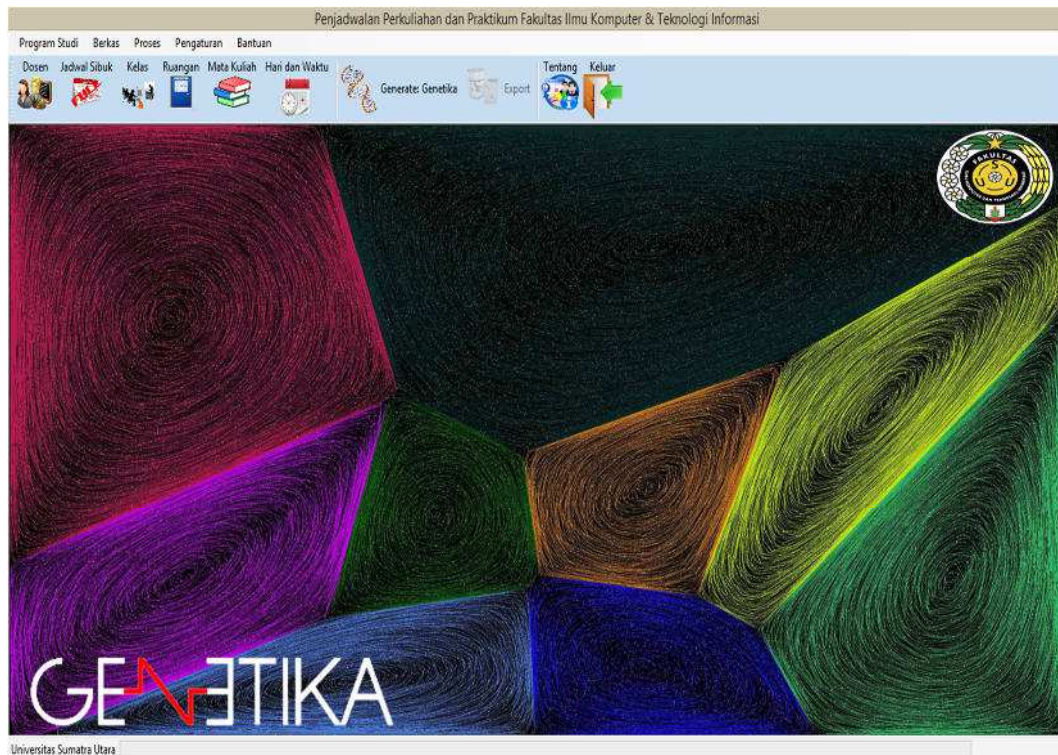
Pada *form* lupa password, *user* harus mengisi username, pertanyaan dan jawaban yang tepat untuk mendapatkan password kembali. Tampilan *form* lupa password dapat dilihat seperti Gambar 4.3.



**Gambar 4.3** Tampilan Lupa Password

#### 4.1.4 Form Menu Utama

*Form* Menu Utama merupakan *form* yang dapat diakses setelah berhasil melakukan proses *login*. Pada *form* menu utama terdapat 5 menu yang akan digunakan, yaitu menu Program Studi, menu Berkas, menu Proses, menu Pengaturan dan menu Bantuan. Selain menu, pada *form* utama juga terdapat beberapa icon, yaitu Dosen, Jadwal Sibuk, Kelas, Ruangan, Mata Kuliah, Hari dan Waktu, *Generate: Genetika*, *Export*, Tentang dan Keluar. Tampilan *form* Menu Utama dapat dilihat seperti Gambar 4.4.



**Gambar 4.4** Tampilan Menu Utama

#### 4.1.5 Form Program Studi

Pada *form* Program Studi, *user* harus memilih salah satu dari program studi di Fasilkom-TI. Hal tersebut berfungsi untuk pemilihan database yang akan digunakan. Secara default, sistem akan membaca program studi sebagai Ilmu Komputer. Tampilan *form* Program Studi dapat dilihat seperti Gambar 4.5.

**Gambar 4.5** Tampilan Program Studi

#### 4.1.6 Form Input Dosen

*Form* Input Dosen merupakan *form* pengolahan untuk data dosen. User dapat menambah, menghapus, mengedit data dosen. Tampilan *form* input dosen dapat dilihat seperti Gambar 4.6.

ID	INISIAL	JENIS	NAMA
1	AES	Praktikum	Angga Eriansyah Setiawan
2	APY	Praktikum	Agung Putu Yoga
3	DTR	Praktikum	Dias Tia Rahmadhani
4	AKH	Praktikum	Akhinuddin Nur
5	DMR	Praktikum	Dina Meladya Rizki Sujiono
6	FQA	Praktikum	Furqan Alatas
7	MMH	Praktikum	Muhammad Miftahul Huda
8	AIS	Praktikum	Anhar Ismail
9	GEU	Praktikum	Geubrina Rizky
10	GUS	Praktikum	Gusra Algeri
11	ATR	Praktikum	Atika Rahayu
12	AMU	Praktikum	Andika Mula Utama
13	YSN	Praktikum	Yogi Sulaiman
14	APR	Praktikum	Ahmad Pratama Ramadhan
15	NUR	Praktikum	Nurhasanah
16	RAM	Praktikum	Ramudin, S.Kom
17	AMZ	Praktikum	Al Mizfar

**Gambar 4.6** Tampilan *Form* Input Dosen

#### 4.1.7 Form Input Jadwal Sibuk

*Form* Input Jadwal Sibuk merupakan *form* pengolahan untuk data jadwal ketaktersediaan dosen. User dapat menambah, menghapus, mengedit data jadwal sibuk dosen tersebut. Tampilan *form* input jadwal sibuk dosen dapat dilihat seperti Gambar 4.7.

ID	INISIAL	NAMA	ID_HARI	HARI	ID_WAKTU	WAKTU
1	SEF	Dr. Syahril Effendi, S.Si, MIT	2	Selasa	8	13:50 - 14:40
2	SEF	Dr. Syahril Effendi, S.Si, MIT	2	Selasa	9	14:40 - 15:30
3	SEF	Dr. Syahril Effendi, S.Si, MIT	2	Selasa	10	15:30 - 16:20
4	OSS	Prof. Dr. Opim S Sitompul	2	Selasa	8	13:50 - 14:40
5	OSS	Prof. Dr. Opim S Sitompul	2	Selasa	9	14:40 - 15:30
6	OSS	Prof. Dr. Opim S Sitompul	3	Rabu	8	13:50 - 14:40
7	SEF	Dr. Syahril Effendi, S.Si, MIT	3	Rabu	9	14:40 - 15:30
8	SEF	Dr. Syahril Effendi, S.Si, MIT	3	Rabu	10	15:30 - 16:20
9	OSS	Prof. Dr. Opim S Sitompul	4	Kamis	8	13:50 - 14:40
10	OSS	Prof. Dr. Opim S Sitompul	4	Kamis	9	14:40 - 15:30
11	SEF	Dr. Syahril Effendi, S.Si, MIT	5	Jumat	8	13:50 - 14:40
12	SEF	Dr. Syahril Effendi, S.Si, MIT	5	Jumat	9	14:40 - 15:30
13	SJD	Sajadin Sembiring, S.Si, M.Comp.Sc	3	Rabu	8	13:50 - 14:40
14	SJD	Sajadin Sembiring, S.Si, M.Comp.Sc	3	Rabu	9	14:40 - 15:30
15	SJD	Sajadin Sembiring, S.Si, M.Comp.Sc	2	Selasa	8	13:50 - 14:40
16	SJD	Sajadin Sembiring, S.Si, M.Comp.Sc	2	Selasa	9	14:40 - 15:30
17	PLS	Dr. Poltak Sihombing, M.Kom	2	Selasa	10	15:30 - 16:20

**Gambar 4.7** Tampilan *Form* Input Jadwal Sibuk

#### 4.1.8 *Form* Input Kelas

*Form* Input Kelas merupakan *form* pengolahan untuk data kelas. User dapat menambah, menghapus, mengedit data kelas. Tampilan *form* input kelas dapat dilihat seperti Gambar 4.8.

ID	NAMA	JENIS	SEMESTER
1	2015 A	Teori	1
2	2015 B	Teori	1
3	2015 C	Teori	1
4	2014 A	Teori	3
5	2014 B	Teori	3
6	2014 C	Teori	3
7	2013 A	Teori	5
8	2013 B	Teori	5
9	2013 C	Teori	5
10	2012 A	Teori	7
11	2012 B	Teori	7
12	2012 C	Teori	7
13	2015 A1	Praktikum	1
14	2015 A2	Praktikum	1
15	2015 B1	Praktikum	1
16	2015 B2	Praktikum	1
17	2015 C1	Praktikum	1

**Gambar 4.8** Tampilan *Form* Input Kelas

#### 4.1.9 Form Input Ruangan

*Form* Input Ruangan merupakan *form* pengolahan untuk data ruang. User dapat menambah, menghapus, mengedit data ruang. Tampilan *form* input ruang dapat dilihat seperti Gambar 4.9.

ID	NAMA	KAPASITAS	JENIS
1	Pascal	50	Teori
2	Fortran	50	Teori
3	Basic	50	Teori
4	Java	50	Teori
5	Visual Basic	50	Teori
6	Delphi	50	Teori
7	Lab I	20	Praktikum
8	Lab II	20	Praktikum
9	Lab III	20	Praktikum
10	Lab IV	20	Praktikum
*			

**Gambar 4.9** Tampilan *Form* Input Ruangan

#### 4.1.10 Form Input Matakuliah

*Form* Input Matakuliah merupakan *form* pengolahan untuk data matakuliah. User dapat menambah, menghapus, mengedit data matakuliah. Tampilan *form* input matakuliah dapat dilihat seperti Gambar 4.10.



**Data Mata Kuliah**

ID: 33  
 Kode: UNIZ206  
 Nama: Pancasila dan Kewarganegaraan  
 Kelas: 2014 B  
 Semester: 3  
 SKS: 3  
 Dosen: Abdul Rahman, SH, MH  
 Jenis Kelas: ☒ Teori ☐ Praktikum

Hapus Ubah  
 Tambah Simpan Batal

ID	KODE	NAMA	KELAS	SEMESTER	SKS	JENIS	DOSEN
33	UNIZ206	Pancasila dan Kewarganegaraan	2014 B	3	3	Teori	ADR
34	UNIZ206	Pancasila dan Kewarganegaraan	2014 C	3	3	Teori	ADR
35	ILK2208	Matematika Diskrit	2014 A	3	3	Teori	ASH
36	ILK2208	Matematika Diskrit	2014 B	3	3	Teori	PTS
37	ILK2208	Matematika Diskrit	2014 C	3	3	Teori	IRY
38	ILK2209	Penrograman Berorientasi Objek	2014 A	3	3	Teori	VWJ
39	ILK2213	Prak. Penrograman Berorientasi Objek	2014 A1	3	1	Praktikum	FQA
40	ILK2213	Prak. Penrograman Berorientasi Objek	2014 A2	3	1	Praktikum	FQA
41	ILK2209	Penrograman Berorientasi Objek	2014 B	3	3	Teori	HDL
42	ILK2213	Prak. Penrograman Berorientasi Objek	2014 B1	3	1	Praktikum	DIK
43	ILK2213	Prak. Penrograman Berorientasi Objek	2014 B2	3	1	Praktikum	DIK
44	ILK2209	Penrograman Berorientasi Objek	2014 C	3	3	Teori	ADC
45	ILK2213	Prak. Penrograman Berorientasi Objek	2014 C1	3	1	Praktikum	FAG
46	ILK2213	Prak. Penrograman Berorientasi Objek	2014 C2	3	1	Praktikum	FAG
47	ILK2210	Sistem Manajemen Database	2014 A	3	2	Teori	MSL
48	ILK2214	Prak. Sistem Manajemen Database	2014 A1	3	1	Praktikum	NPY
49	ILK2214	Prak. Sistem Manajemen Database	2014 A2	3	1	Praktikum	NPY
50	ILK2210	Sistem Manajemen Database	2014 B	3	2	Teori	AMS
51	ILK2214	Prak. Sistem Manajemen Database	2014 B1	3	1	Praktikum	JIP
52	ILK2214	Prak. Sistem Manajemen Database	2014 B2	3	1	Praktikum	JIP

**Gambar 4.10** Tampilan *Form* Input Matakuliah

#### 4.1.11 *Form* Input Hari dan Waktu

*Form* Input Hari dan Waktu merupakan *form* pengolahan untuk data hari dan waktu. User dapat menambah, menghapus, mengedit data hari dan waktu. Tampilan *form* input hari dan waktu dapat dilihat seperti Gambar 4.11.

**Input Hari dan Waktu**

**Hari**  
 ID: 1  
 Keterangan: Senin

**Waktu**  
 ID:  
 Keterangan:

☒ Hari ☐ Waktu Hapus Ubah  
 Tambah Simpan Batal

ID	HARI
1	Senin
2	Selasa
3	Rabu
4	Kamis
5	Jumat
6	Sabtu
*	

ID	WAKTU
1	08:00 - 08:50
2	08:50 - 09:40
3	09:40 - 10:30
4	10:30 - 11:20
5	11:20 - 12:10
6	12:10 - 13:00
7	13:00 - 13:50
8	13:50 - 14:40
9	14:40 - 15:30
10	15:30 - 16:20
*	

**Gambar 4.11** Tampilan *Form* Input Hari dan Waktu

#### 4.1.12 Form Algoritma Genetika

*Form* Algoritma Genetika merupakan *form* proses pembentukan jadwal perkuliahan dan praktikum dilakukan. Pada *form* ini, user harus menginput jumlah populasi, jumlah generasi, besar crossover rate, dan besar mutasi rate. Tampilan *form* Algoritma Genetika dapat dilihat seperti Gambar 4.12.

**Gambar 4.12** Tampilan *Form* Algoritma Genetika

Hasil proses yang dilakukan algoritma genetika akan disimpan di database dan user dapat mengeksport jadwal tersebut ke dalam file *excel*. Tampilan laporan yang telah dibuat dapat dilihat seperti pada Gambar 4.13.

ID	Kode	Mata Kuliah	Semester	SKS	Dosen	Kelas	Ruang	Hari	Waktu
1	UN1201	Pendid. Agama Islam	1	2	Dra. Hj. Sahlia, M.Ag	2015 A,B,C	Visual Basic	Selasa	13:50 - 14:40
2	UN1202	Pendid. Agama Kristen Katolik	1	2	Drs. BM Sembiring	2015 A,B,C	Basic	Rabu	12:10 - 12:50
3	UN1203	Pendid. Agama Kristen Protestan	1	2	Linda Zenita Simanjuntak, STP, M.Obv, M.Th.	2015 A,B,C	Basic	Rabu	13:00 - 13:50
4	UN1204	Pendid. Agama Budha	1	2	Meng Lue, S.Ag.	2015 A,B,C	Passel	Senin	11:20 - 12:10
5	ILK1200	Matematika Dasar	1	2	Open Darius, M.Sc.	2015 A	Passel	Kamis	14:40 - 15:30
6	ILK1200	Matematika Dasar	1	2	Open Darius, M.Sc.	2015 B	Fortran	Rabu	12:10 - 13:00
7	ILK1200	Matematika Dasar	1	2	Adartina A. Sipayung	2015 C	Delphi	Kamis	09:40 - 10:30
8	UN1222	Fisika I	1	2	Drs. Achiruddin, M.Si.	2015 A	Jawa	Kelapa	08:00 - 09:40
9	UN1222	Fisika I	1	2	Drs. Achiruddin, M.Si.	2015 B	Jawa	Jumel	08:50 - 09:40
10	UN1222	Fisika I	1	2	Syahrul Humaidi, S.Si, M.Kom.	2015 C	Basic	Selasa	08:00 - 08:50
11	ILK1201	Konsep Dasar Pemrograman	1	3	Dian Wirdasari, S.Si, M.Kom.	2015 A	Passel	Selasa	15:30 - 16:20
12	ILK1204	Prak. Konsep Dasar Pemrograman	1	1	Akhiruddin Nur	2015 A1	Lab III	Jumat	09:40 - 10:30
13	ILK1204	Prak. Konsep Dasar Pemrograman	1	1	Akhiruddin Nur	2015 A2	Lab III	Kamis	08:00 - 08:50
14	ILK1201	Konsep Dasar Pemrograman	1	3	Dian Rachmawati, S.Si, M.Kom.	2015 B	Passel	Kamis	08:50 - 09:40
15	ILK1204	Prak. Konsep Dasar Pemrograman	1	1	Angga Erlansyah Setiawan	2015 D1	Lab I	Rabu	13:50 - 14:40
16	ILK1204	Prak. Konsep Dasar Pemrograman	1	1	Angga Erlansyah Setiawan	2015 D2	Lab IV	Rabu	13:50 - 14:40
17	ILK1201	Konsep Dasar Pemrograman	1	3	Handrizal, S.Si, M.Comp.Sc.	2015 C	Jawa	Senin	15:30 - 16:20
18	ILK1204	Prak. Konsep Dasar Pemrograman	1	1	Dias Tia Rahmadhani	2015 C3	Lab III	Rabu	08:50 - 09:40
19	ILK1204	Prak. Konsep Dasar Pemrograman	1	1	Dias Tia Rahmadhani	2015 C2	Lab I	Selasa	12:10 - 13:00
20	ILK1202	Logika Komputer	1	2	Siti Dara Fadilla, S.Si, MT	2015 A	Visual Basic	Rabu	08:50 - 09:40
21	ILK1202	Logika Komputer	1	3	Dr. Syahril Chendi, S.Si, MT	2015 B	Passel	Selasa	13:00 - 13:50
22	ILK1202	Logika Komputer	1	2	Prof. Dr. Iryanto, M.Si.	2015 C	Delphi	Subtu	13:00 - 13:50

**Gambar 4.13** Tampilan Laporan Algoritma Genetika

## 4.2 Pengujian Sistem

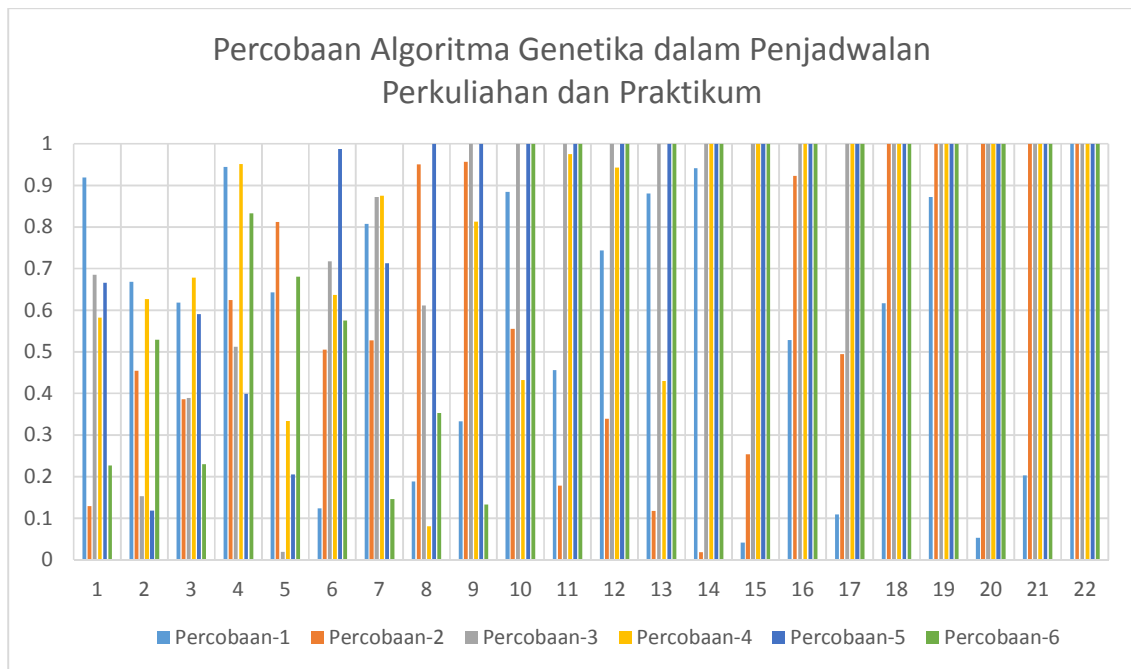
Pengujian sistem ini dilakukan untuk melihat apakah algoritma genetika dapat menentukan jadwal yang paling optimal dengan memasukkan nilai parameter genetika yang berbeda-beda. Pada tahap pengujian, untuk dapat menghasilkan pengujian dilakukan dengan cara memasukkan nilai inputan parameter genetika yang sama ataupun berbeda. Hasil perbandingan pengujian, dapat dilihat pada tabel 4.1.

**Tabel 4.1** Pengujian Algoritma Genetika dalam penjadwalan perkuliahan

Pengujian ke-	Populasi	Crossover Rate	Mutasi Rate	Fitness Rata-Rata	Fitness Akhir	Solusi	
						Individu	Generasi
1	60	0.3	0.2	0.106 229	1	46	22
2	40	0.8	0.3	0.200 656	1	21	18
3	45	0.5	0.5	0.151 774	1	3	9
4	20	0.6	0.8	0.267 56	1	14	14
5	20	0.4	0.4	0.210 139	1	9	8
6	20	0.4	0.4	0.120 96	1	6	10



Untuk mendapatkan jadwal dan hasil generasi yang optimal, nilai dari hasil fitness akhir harus bernilai 1. Dari tabel 4.1 dapat dijelaskan bahwa pengujian menggunakan algoritma genetika akan menghasilkan solusi optimal. Perolehan solusi yang didapat. Untuk mendapatkan jadwal, algoritma genetika harus mengalami perubahan nilai fitness dalam beberapa generasi. Demikian pula dalam pengujian sistem di atas, perubahan nilai fitness dalam contoh pengujian dapat dilihat dalam gambar 4.14 berikut.



**Gambar 4.14** Grafik perubahan nilai fitness dalam pengujian Algoritma Genetika

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis dan pengujian yang dilakukan pada bab sebelumnya, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Implementasi penjadwalan perkuliahan dan praktikum yang dibuat oleh penulis berhasil menghasilkan sebuah jadwal yang baik dengan menggunakan algoritma genetika.
2. Perbandingan hasil akhir dari sistem yang dibangun dengan menggunakan algoritma Genetika akan menghasilkan sebuah solusi jadwal yang baik tetapi waktu yang diperlukan untuk mencapai solusi berbeda-beda. Hal tersebut dikarenakan perbedaan nilai parameter dan bilangan random yang digunakan di setiap pengujian.

#### **5.2 Saran**

Beberapa saran yang dapat dijadikan pertimbangan dalam mengembangkan penelitian ini adalah :

1. Penelitian selanjutnya disarankan untuk membangun sistem yang lebih *user friendly* untuk memudahkan *user* dalam penggunaan system.
2. Implementasi diharapkan dapat diterapkan dalam algoritma selain Genetika.

## DAFTAR PUSTAKA

- [ADH03] Adhy, S. & Kushartantya. 2003. Penyelesaian Masalah Job Shop menggunakan Algoritma Genetika. *Jurnal Masyarakat Informatika*. 1(1) : 31-42.
  
- [APR12] Apriani, R. 2012. Algoritma Genetika untuk Menyelesaikan Permasalahan Penjadwalan Perkuliahan dan Praktikum. Skripsi. Universitas Sumatera Utara.
  
- [BAN12] Bangun, P.B.J., Octarina, S. & Virgo, G.H. 2012. Penerapan Konsep Algoritma Genetika untuk Penjadwalan Kegiatan Perkuliahan Semester Ganjil Kurikulum 2012 di Jurusan Matematika FMIPA UNSRI. *Jurnal Penelitian Sains*. 2(15) : 55-59.
  
- [BER10] Berlianty, I. & Arifin, M. 2010. *Teknik-Teknik Optimasi Heuristik*. Graha Ilmu : Yogyakarta.
  
- [KOZ92] Koza, J.R. 1992. *Genetic Programming on the Programming of Computers by Means of Natural Selection*. The MIT Press: London.
  
- [KUS05] Kusumadewi, S. & Purnomo, H. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Graha Ilmu : Yogyakarta.
  
- [LAL03] Lalescu, L. & Badica, C. 2003. Timetabling Experiments Using Genetic Algorithms. *Proceedings 14th International Conference on Control Systems and Computer Science, vol. II*, pp. 114.
  
- [NUG08] Nugraha, I. 2008. Aplikasi Algoritma Genetik untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar. *Jurnal Strategi Algoritmik* : Program Studi Teknik Informatika, Institut Teknologi Bandung, Bandung.
  
- [SIH10] Sihombing, P. 2010. Keyword Competition Approach Ranked Document Retrieval. Disertasi Ph.D. Universiti Sains Malaysia.

- [SYA14] Syarif, E. A. 2014. *Algoritma Genetika: Teori dan Aplikasi*. Graha Ilmu : Bandar Lampung.
- [WAH09] Wahyuni, S. 2009. Metode Pencarian Langsung Untuk Menyelesaikan Problema Knapsack. Skripsi. Universitas Sumatera Utara.
- [ZUK13] Zukhri, Z. 2013. *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Penerbit Andi : Yogyakarta.

## LISTING PROGRAM

### Form Generate

```

Public Class frmGenetika

    Dim R As Random = New Random
    Dim T, N As String
    Dim Log As String
    Dim TotalR_Teori, TotalR_Lab, TotalRuangan As Integer

    Dim PjgKrom, JlhKrom, JlhTime As Integer
    Dim PopSize, GenSize As Integer
    Dim CR, MR, ER As Double

    Dim Individu(,), IndividuB(,), Solved() As Kromosom
    Dim Pnl() As Penalti
    Dim TotalPenalti As Integer
    Dim Fitness() As Double
    Dim TotalFitness As Double
    Dim Prob(), ProbK() As Double
    Dim TotalProb As Integer
    Dim BilanganAcak() As Double
    Dim Induk(,), Anak(,) As Kromosom
    Dim TotalInduk As Integer
    Dim UrutanInduk() As Integer

    Public Structure MataKuliah
        Dim ID As Integer
        Dim Dosen As Integer
        Dim Kelas As Integer
        Dim SKS As Integer
    End Structure

    Public Structure Kromosom
        Dim MK As MataKuliah
        Dim Ruang As Integer
        Dim Time As Integer
    End Structure

    Public Structure Penalti
        Dim Dosen, Sibuk, Kelas, Ruang, Lab, SholJum, Total As Integer
    End Structure

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Inisialisasi()
    End Sub

```

```

Private Sub Main()
    btnProses.Text = "Mohon Tunggu"
    btnProses.Enabled = False

    Inisialisasi()
    InisialisasiDG()
    InitVar()

    For i = 1 To GenSize
        TotalPenalti = 0
        TotalFitness = 0
        TotalProb = 0

        For j = 1 To PopSize
            Dim P As Integer
            P = HitungPenalti(j)
            If P = 0 Then
                isSolved = True
                frmMDI.ExportToolStripMenuItem.Enabled = True
                frmMDI.tsbExport.Enabled = True
                frmMDI.tsbProgress.Value = GenSize
                frmMDI.tsb.Refresh()
                btnExport.Enabled = True
                CetakPopulasi(Solved, dgvKrom)
                SolvedIndividu(Solved, dgvReport)
                MessageBox.Show("Proses Genetika Selesai Pada Generasi Ke " & i & ",
Individu Ke " & j, "Selamat")
                Exit For
            End If
            TotalPenalti = TotalPenalti + P
        Next

        If isSolved Then
            Exit For
        End If

        For j = 1 To PopSize
            TotalFitness = TotalFitness + HitungFitness(j)
        Next
        For j = 1 To PopSize
            TotalProb = TotalProb + HitungProb(j)
        Next
        For j = 1 To PopSize
            HitungProbK(j)
        Next

        SeleksiRouletteWheel(True)
        Crossover(True)
    
```

```

MutasiKelas()
MutasiRuang()
MutasiLab()
MutasiDosen()
MutasiSibuk()
MutasiSholat()

Log = Log & "===== Akhir Generasi Ke " & i & "
===== " & N & N

frmMDI.tsbProgress.Value = i
'frmMDI.tsbProgress.

If i = GenSize And isSolved = False Then
    MessageBox.Show("Proses Genetika Tidak Membuahkan Hasil", "Mohon
Maaf")
End If
Next
CetakLog()
btnProses.Text = "Proses"
btnProses.Enabled = True
End Sub

Sub IsiKromosom(ByVal Pop As Integer)
    ' Input Data Ke Kromosom
    For i = 1 To JlhKrom
        Individu(Pop, i).MK.ID = dgvMatkul.Item(0, i - 1).Value
        Individu(Pop, i).MK.Dosen = AmbilID(dgvDosen, dgvMatkul.Item(7, i -
1).Value, 1)
        Individu(Pop, i).MK.Kelas = AmbilID(dgvKelas, dgvMatkul.Item(3, i -
1).Value, 1)
        Individu(Pop, i).MK.SKS = dgvMatkul.Item(5, i - 1).Value
        Individu(Pop, i).Ruang = AcakAngka(1, dgvRuang.Rows.Count - 1)
        Individu(Pop, i).Time = AcakAngka(10, JlhTime) 'Math.Floor(AcakAngka(10,
60) / 10) + (AcakAngka(10, 60) Mod 7) + 1
    Next
End Sub

Public Function AcakAngka(ByVal a As Integer, ByVal x As Integer) As Integer
    Dim Result As Integer = 0

    While Result < a
        Result = R.Next Mod (x + 1)
    End While
    Return Result
End Function

```

```

Function AmbilID(ByVal Tabel As DataGridView, ByVal Field As String, ByVal
Pos As Integer) As Integer
    Dim Result As Integer = 0

```

```

    For i = 1 To Tabel.Rows.Count - 1
        If Field = Tabel.Item(Pos, i - 1).Value Then
            Return Tabel.Item(0, i - 1).Value
        End If
    Next
    Return Result
End Function

```

```

Function AmbilWaktu(ByVal Tabel As DataGridView, ByVal Field1 As Integer,
ByVal Field2 As Integer, ByVal Baris As Integer) As Integer
    Return (Tabel.Item(Field1, Baris - 1).Value * 10) + (Tabel.Item(Field2, Baris -
1).Value - 1)
End Function

```

```

Sub CetakPopulasi(ByVal K() As Kromosom, ByVal DG As DataGridView)
    ' Cetak Kromosom Ke DataGridView dgvKrom
    DG.Rows.Clear()
    For i = 1 To K.Length - 1
        DG.Rows.Add(K(i).MK.ID, K(i).MK.Dosen, K(i).MK.Kelas, K(i).MK.SKS,
K(i).Ruang, Math.Floor(K(i).Time / 10), (K(i).Time Mod 10) + 1)
    Next
End Sub

```

```

Sub CetakLog()
    txtLog.Text = Log
End Sub

```

```

Sub Inisialisasi()
    T = vbTab
    N = vbCrLf
    PjgKrom = 3

```

```

    CetakGrid("Matkul", dgvMatkul)
    CetakGrid("Dosen", dgvDosen)
    CetakGrid("Kelas", dgvKelas)
    CetakGrid("Ruang", dgvRuang)
    HitungRuangan()
    CetakGrid("Hari", dgvHari)
    CetakGrid("Waktu", dgvWaktu)
    CetakGrid("Sibuk", dgvSibuk)

```

```

    JlhKrom = dgvMatkul.Rows.Count - 1
    JlhTime = ((dgvHari.Rows.Count - 1) * 10) + 9

```



```

    btnExport.Enabled = False
    frmMDI.ExportToolStripMenuItem.Enabled = False
    frmMDI.tsbExport.Enabled = False
End Sub

```

```

Sub InisialisasiDG()
    dgvKrom.RowCount = JlhKrom + 1
    'dgvKrom.ColumnCount = 7

    'dgvKrom.Columns(0).Name = "Matkul"
    'dgvKrom.Columns(1).Name = "Dosen"
    'dgvKrom.Columns(2).Name = "Kelas"
    'dgvKrom.Columns(3).Name = "SKS"
    'dgvKrom.Columns(4).Name = "Ruang"
    'dgvKrom.Columns(5).Name = "Hari"
    'dgvKrom.Columns(6).Name = "Waktu"

    'For i = 0 To 6
    'dgvKrom.Columns(i).Width = 50
    'dgvKrom.Columns(i).Resizable = True
    'Next

    'dgvReport.ColumnCount = 10
    dgvReport.RowCount = JlhKrom + 1

    'dgvReport.Columns(0).Name = "ID"
    'dgvReport.Columns(0).Width = 30
    'dgvReport.Columns(1).Name = "KODE"
    'dgvReport.Columns(1).Width = 70
    'dgvReport.Columns(2).Name = "MATA KULIAH"
    'dgvReport.Columns(2).Width = 200
    'dgvReport.Columns(3).Name = "SEMESTER"
    'dgvReport.Columns(3).Width = 30
    'dgvReport.Columns(4).Name = "SKS"
    'dgvReport.Columns(4).Width = 30
    'dgvReport.Columns(5).Name = "DOSEN"
    'dgvReport.Columns(5).Width = 250
    'dgvReport.Columns(6).Name = "KELAS"
    'dgvReport.Columns(6).Width = 70
    'dgvReport.Columns(7).Name = "RUANG"
    'dgvReport.Columns(7).Width = 70
    'dgvReport.Columns(8).Name = "HARI"
    'dgvReport.Columns(8).Width = 70
    'dgvReport.Columns(9).Name = "WAKTU"
    'dgvReport.Columns(9).Width = 100
End Sub

```

```

Sub InitVar()
    Log = ""

```

```

txtLog.Text = ""
isSolved = False
PopSize = Val(txtPopSize.Text)
GenSize = Val(txtGenSize.Text)
frmMDI.tsbProgress.Maximum = GenSize

CR = txtProb.Text
MR = txtMutasi.Text

txtPjgKrom.Text = 3
txtJlhKrom.Text = JlhKrom
frmMDI.tsbProgress.Value = 0

ReDim Individu(PopSize, JlhKrom)
ReDim IndividuB(PopSize, JlhKrom)
ReDim Solved(JlhKrom)
ReDim Pnl(PopSize)
ReDim Fitness(PopSize)
ReDim Prob(PopSize)
ReDim ProbK(PopSize)
ReDim BilanganAcak(PopSize)
ReDim Induk(PopSize, JlhKrom)
ReDim Anak(PopSize, JlhKrom)
ReDim UrutanInduk(PopSize)

For i = 1 To PopSize
    IsiKromosom(i)
Next

End Sub

Sub HitungRuangan()
    TotalR_Teori = 0
    TotalR_Lab = 0
    For i = 0 To dgvRuang.Rows.Count - 2
        If dgvRuang.Item(3, i).Value = "Teori" Then
            TotalR_Teori = TotalR_Teori + 1
        ElseIf dgvRuang.Item(3, i).Value = "Praktikum" Then
            TotalR_Lab = TotalR_Lab + 1
        End If
    Next
    TotalRuangan = TotalR_Teori + TotalR_Lab
End Sub

Function HitungPenalti(ByVal Pop As Integer)
    Dim SKS As Integer

    Pnl(Pop).Dosen = 0
    Pnl(Pop).Sibuk = 0

```

```

Pnl(Pop).Kelas = 0
Pnl(Pop).Ruang = 0
Pnl(Pop).Lab = 0
Pnl(Pop).SholJum = 0

'Penalti Dosen Ngajar Pada Hari & Jam Yang Bersamaan
For i = 1 To JlhKrom - 1
  For j = i + 1 To JlhKrom
    If (Individu(Pop, i).MK.Dosen = Individu(Pop, j).MK.Dosen) And
      (Individu(Pop, i).Time = Individu(Pop, j).Time) Then
      Pnl(Pop).Dosen = Pnl(Pop).Dosen + 1
      Log = Log & "Bentrok Dosen Pas Kromosom " & i & " <-> Kromosom "
      & j & N
    ElseIf (Individu(Pop, i).MK.Dosen = Individu(Pop, j).MK.Dosen) And
      (Math.Floor(Individu(Pop, i).Time / 10) = Math.Floor(Individu(Pop, j).Time / 10))
      Then
        If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then
          SKS = Individu(Pop, i).MK.SKS
          If SKS = 1 Then SKS = 2
          If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then
            Pnl(Pop).Dosen = Pnl(Pop).Dosen + 1
            Log = Log & "Bentrok Dosen " & Individu(Pop, i).MK.SKS & "
            SKS Ke Sesudahnya Kromosom " & i & " <-> Kromosom " & j & N
          End If
        Else
          SKS = Individu(Pop, j).MK.SKS
          If SKS = 1 Then SKS = 2
          If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then
            Pnl(Pop).Dosen = Pnl(Pop).Dosen + 1
            Log = Log & "Bentrok Dosen " & Individu(Pop, j).MK.SKS & "
            SKS Ke Sebelumnya Kromosom " & i & " <-> Kromosom " & j & N
          End If
        End If
      End If
    End If
  Next
Next
Log = Log & N

'Penalti Kesibukan Dosen
For i = 1 To JlhKrom
  For j = 1 To dgvSibuk.Rows.Count - 1
    'Test Apakah Waktu Dosen Di Kromosom Bentrok Dengan Jadwal Pada
    Tabel Sibuk Dosen
    If (Individu(Pop, i).MK.Dosen = AmbilID(dgvDosen, dgvSibuk.Item(1,
      j).Value, 1)) And (Individu(Pop, i).Time = AmbilWaktu(dgvSibuk, 3, 5, j)) Then
      Pnl(Pop).Sibuk = Pnl(Pop).Sibuk + 1
      Log = Log & "Bentrok Kromosom " & i & " Dengan Tabel Sibuk Dosen
      Ke " & j & N
    End If
  End If
End For

```

```

    Next
  Next
  Log = Log & N

  'Penalti Kelas Belajar Pada Hari & Jam Yang Bersamaan
  For i = 1 To JlhKrom - 1
    For j = i + 1 To JlhKrom
      If (Individu(Pop, i).MK.Kelas = Individu(Pop, j).MK.Kelas) And
        (Individu(Pop, i).Time = Individu(Pop, j).Time) Then
        Pnl(Pop).Kelas = Pnl(Pop).Kelas + 1
        Log = Log & "Bentrok Kelas Pas Kromosom " & i & " <-> Kromosom "
        & j & N
      ElseIf (Individu(Pop, i).MK.Kelas = Individu(Pop, j).MK.Kelas) And
        (Math.Floor(Individu(Pop, i).Time / 10) = Math.Floor(Individu(Pop, j).Time / 10))
        Then
          If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then
            SKS = Individu(Pop, i).MK.SKS
            If SKS = 1 Then SKS = 2
            If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then
              Pnl(Pop).Kelas = Pnl(Pop).Kelas + 1
              Log = Log & "Bentrok Kelas " & Individu(Pop, i).MK.SKS & "
              SKS Ke Sesudahnya Kromosom " & i & " <-> Kromosom " & j & N
            End If
          Else
            SKS = Individu(Pop, j).MK.SKS
            If SKS = 1 Then SKS = 2
            If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then
              Pnl(Pop).Kelas = Pnl(Pop).Kelas + 1
              Log = Log & "Bentrok Kelas " & Individu(Pop, j).MK.SKS & "
              SKS Ke Sebelumnya Kromosom " & i & " <-> Kromosom " & j & N
            End If
          End If
        End If
      End If
    Next
  Next
  Log = Log & N

  'Penalti Ruang Belajar Pada Hari & Jam Yang Bersamaan
  For i = 1 To JlhKrom - 1
    For j = i + 1 To JlhKrom
      If (Individu(Pop, i).Ruang = Individu(Pop, j).Ruang) And (Individu(Pop,
i).Time = Individu(Pop, j).Time) Then
        Pnl(Pop).Ruang = Pnl(Pop).Ruang + 1
        Log = Log & "Bentrok Ruang Pas Kromosom " & i & " <-> Kromosom "
        & j & N
      ElseIf (Individu(Pop, i).Ruang = Individu(Pop, j).Ruang) And
        (Math.Floor(Individu(Pop, i).Time / 10) = Math.Floor(Individu(Pop, j).Time / 10))
        Then
          If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then

```

```

    SKS = Individu(Pop, i).MK.SKS
    If SKS = 1 Then SKS = 2
    If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then
        Pnl(Pop).Ruang = Pnl(Pop).Ruang + 1
        Log = Log & "Bentrok Ruang " & Individu(Pop, i).MK.SKS & "
    SKS Ke Sesudahnya Kromosom " & i & " <-> Kromosom " & j & N
    End If
Else
    SKS = Individu(Pop, j).MK.SKS
    If SKS = 1 Then SKS = 2
    If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then
        Pnl(Pop).Ruang = Pnl(Pop).Ruang + 1
        Log = Log & "Bentrok Ruang " & Individu(Pop, j).MK.SKS & "
    SKS Ke Sebelumnya Kromosom " & i & " <-> Kromosom " & j & N
    End If
End If
End If

Next
Next
Log = Log & N

'Penalti Ruangan Lab Dipakai 2 SKS Teori
For i = 1 To JlhKrom
    If (Individu(Pop, i).MK.SKS > 1 And Individu(Pop, i).Ruang >= 7) Then
        Pnl(Pop).Lab = Pnl(Pop).Lab + 1
    End If
Next

'Penalti Sholat Jum'at Pada Jam Ke 5, 6 dan 7
For i = 1 To JlhKrom
    If (Individu(Pop, i).Time = 54) Or (Individu(Pop, i).Time = 55) Or
    (Individu(Pop, i).Time = 56) Then
        Pnl(Pop).SholJum = Pnl(Pop).SholJum + 1
        Log = Log & "Bentrok Sholat Jum'at Pada Kromosom " & i & N
    End If
Next
Log = Log & N

Pnl(Pop).Total = Pnl(Pop).Dosen + Pnl(Pop).Sibuk + Pnl(Pop).Kelas +
Pnl(Pop).Ruang + Pnl(Pop).Lab + Pnl(Pop).SholJum
Log = Log & "Indv [" & Pop & "]" --> D : " & Pnl(Pop).Dosen & T & "S : " &
Pnl(Pop).Sibuk & T & "K : " & Pnl(Pop).Kelas & T
Log = Log & "R : " & Pnl(Pop).Ruang & T & "L : " & Pnl(Pop).Lab & T & "SH
: " & Pnl(Pop).SholJum & T & "T : " & Pnl(Pop).Total & N

'Jika Fitness = 1, Maka Masukkan Kromosom ke Solved
If Pnl(Pop).Total = 0 Then
    For i = 1 To JlhKrom

```

```

        Solved(i) = Individu(Pop, i)
    Next
End If
Return Pnl(Pop).Total
End Function

Function HitungFitness(ByVal Pop As Integer)
    Fitness(Pop) = 1 / (1 + Pnl(Pop).Total)
    Log = Log & "Fitness [" & Pop & "] --> " & Fitness(Pop) & N
    Return Fitness(Pop)
End Function

Function HitungProb(ByVal Pop As Integer)
    Prob(Pop) = Fitness(Pop) / TotalFitness
    Log = Log & "Prob [" & Pop & "] --> " & Prob(Pop) & N
    Return Prob(Pop)
End Function

Sub HitungProbK(ByVal Pop As Integer)
    ProbK(Pop) = 0
    For i = 1 To Pop
        ProbK(Pop) = ProbK(Pop) + Prob(i)
    Next
    Log = Log & "ProbK [" & Pop & "] --> " & ProbK(Pop) & N
End Sub

Sub SeleksiRouletteWheel(ByVal Print As Boolean)
    For p = 1 To PopSize
        BilanganAcak(p) = Rnd()
        If Print Then
            Log = Log & "Random [" & p & "] = " & BilanganAcak(p) & vbCrLf
        End If
    Next
    If Print Then
        Log = Log & vbCrLf
    End If

    For p = 1 To PopSize
        For j = 1 To PopSize
            If BilanganAcak(p) < ProbK(j) Then
                For k = 1 To JlhKrom
                    IndividuB(p, k) = Individu(j, k)
                Next
                If Print Then
                    Log = Log & "Individu Baru [" & p & "] <=> Individu [" & j & "] " &
vbCrLf
                End If
            End If
        Exit For
    Next
End Sub

```

```

        End If
    Next
Next
If Print Then
    Log = Log & vbCrLf
End If

For p = 1 To PopSize
    For i = 1 To JlhKrom
        Individu(p, i) = IndividuB(p, i)
    Next
Next
End Sub

Sub Crossover(ByVal Print As Boolean)
    'Menciptakan Crossover
    TotalInduk = 0

    'Membangkitkan Bilangan Acak Antara 0 dan 1
    For p = 1 To PopSize
        BilanganAcak(p) = Rnd()
        If Print Then
            Log = Log & "Random [" & p & "] = " & BilanganAcak(p)
        End If

        ' Mendapatkan Parent
        If BilanganAcak(p) < CR Then
            TotalInduk = TotalInduk + 1
            UrutanInduk(TotalInduk) = p
            For i = 1 To JlhKrom
                Induk(TotalInduk, i) = Individu(p, i)
            Next
        End If
        If Print Then
            Log = Log & "]" & vbCrLf
        End If
    Next

    ' Proses Crossover
    For i = 1 To TotalInduk
        For j = 1 To JlhKrom
            Anak(i, j) = Induk(i, j)
            If i < TotalInduk Then
                Anak(i, j).MK = Induk(i + 1, j).MK
            Else
                Anak(i, j).MK = Induk(1, j).MK
            End If
        Next
    Next
Next

```

Log = Log & vbCrLf

'Masukkan Offspring Ke Individu

For i = 1 To TotalInduk

For j = 1 To JlhKrom

Individu(UrutanInduk(i), j) = Anak(i, j)

Next

Next

End Sub

Sub MutasiDosen()

Dim SKS As Integer

Dim idx As Integer

'Mutasi Dosen

For Pop = 1 To Math.Floor(MR \* PopSize) Mod PopSize

For i = 1 To JlhKrom - 1

For j = i + 1 To JlhKrom

idx = AcakAngka(10, JlhTime)

If (Individu(Pop, i).MK.Dosen = Individu(Pop, j).MK.Dosen) And  
(Individu(Pop, i).Time = Individu(Pop, j).Time) Then

Individu(Pop, j).Time = idx

ElseIf (Individu(Pop, i).MK.Dosen = Individu(Pop, j).MK.Dosen) And  
(Math.Floor(Individu(Pop, i).Time / 10) = Math.Floor(Individu(Pop, j).Time / 10))  
Then

If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then

SKS = Individu(Pop, i).MK.SKS

If SKS = 1 Then SKS = 2

If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then

Individu(Pop, j).Time = idx

End If

Else

SKS = Individu(Pop, j).MK.SKS

If SKS = 1 Then SKS = 2

If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then

Individu(Pop, j).Time = idx

End If

End If

End If

Next

Next

Next

End Sub

Sub MutasiSibuk()

Dim idx As Integer

For Pop = 1 To Math.Floor(MR \* PopSize) Mod PopSize

For i = 1 To JlhKrom



```

        idx = AcakAngka(10, JlhTime)
        For j = 1 To dgvSibuk.Rows.Count - 1
            'Test Apakah Waktu Dosen Di Kromosom Bentrok Dengan Jadwal Pada
            Tabel Sibuk Dosen
            If (Individu(Pop, i).MK.Dosen = AmbilID(dgvDosen, dgvSibuk.Item(1,
            j).Value, 1)) And (Individu(Pop, i).Time = AmbilWaktu(dgvSibuk, 3, 5, j)) Then
                Individu(Pop, j).Time = idx
            End If
        Next
    Next
Next
End Sub

```

```

Sub MutasiKelas()
    Dim SKS As Integer
    Dim idx As Integer
    Dim KelasT, KelasP As String

    'Mutasi Ruangan
    For Pop = 1 To Math.Floor(MR * PopSize) Mod PopSize
        'Mutasi Kelas Belajar Pada Hari & Jam Yang Bersamaan
        For i = 1 To JlhKrom - 1
            For j = i + 1 To JlhKrom
                idx = AcakAngka(10, JlhTime)
                If (Individu(Pop, i).MK.Kelas = Individu(Pop, j).MK.Kelas) And
                (Individu(Pop, i).Time = Individu(Pop, j).Time) Then
                    Individu(Pop, j).Time = idx
                ElseIf (Individu(Pop, i).MK.Kelas = Individu(Pop, j).MK.Kelas) And
                (Math.Floor(Individu(Pop, i).Time / 10) = Math.Floor(Individu(Pop, j).Time / 10))
                Then
                    If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then
                        SKS = Individu(Pop, i).MK.SKS
                        If SKS = 1 Then SKS = 2
                        If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then
                            Individu(Pop, j).Time = idx
                        End If
                    Else
                        SKS = Individu(Pop, j).MK.SKS
                        If SKS = 1 Then SKS = 2
                        If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then
                            Individu(Pop, j).Time = idx
                        End If
                    End If
                End If
            Next
        Next

        'Mutasi Jika Kelas Teori dan Praktek Pada Hari dan Jam Yang Bersamaan
        KelasT = AmbilField(dgvKelas, Individu(Pop, i).MK.Kelas, 0, 1)
        KelasT = KelasT.Substring(0, 6)
    Next
End Sub

```

```

For j = i + 1 To JlhKrom
  idx = AcakAngka(10, JlhTime)
  KelasP = AmbilField(dgvKelas, Individu(Pop, j).MK.Kelas, 0, 1)
  KelasP = KelasP.Substring(0, 6)

  If (KelasT = KelasP) And (Individu(Pop, j).MK.SKS = 1) Then
    If (Individu(Pop, i).Time = Individu(Pop, j).Time) Then
      Individu(Pop, j).Time = idx
    ElseIf (Math.Floor(Individu(Pop, i).Time / 10) =
Math.Floor(Individu(Pop, j).Time / 10)) Then
      If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then
        SKS = Individu(Pop, i).MK.SKS
        If SKS = 1 Then SKS = 2
        If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then
          Individu(Pop, j).Time = idx
        End If
      Else
        SKS = Individu(Pop, j).MK.SKS
        If SKS = 1 Then SKS = 2
        If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then
          Individu(Pop, j).Time = idx
        End If
      End If
    End If
  End If
Next
Next
Next
End Sub
Sub MutasiRuang()
  Dim SKS As Integer
  Dim idx As Integer

  'Mutasi Ruangan
  For Pop = 1 To Math.Floor(MR * PopSize) Mod PopSize
    For i = 1 To JlhKrom - 1
      For j = i + 1 To JlhKrom
        idx = AcakAngka(10, JlhTime)
        If (Individu(Pop, i).Ruang = Individu(Pop, j).Ruang) And (Individu(Pop,
i).Time = Individu(Pop, j).Time) Then
          Individu(Pop, j).Time = idx
        ElseIf (Individu(Pop, i).Ruang = Individu(Pop, j).Ruang) And
(Math.Floor(Individu(Pop, i).Time / 10) = Math.Floor(Individu(Pop, j).Time / 10))
Then
          If (Individu(Pop, j).Time > Individu(Pop, i).Time) Then
            SKS = Individu(Pop, i).MK.SKS
            If SKS = 1 Then SKS = 2

```

```

        If (SKS > Individu(Pop, j).Time - Individu(Pop, i).Time) Then
            Individu(Pop, j).Time = idx
        End If
    Else
        SKS = Individu(Pop, j).MK.SKS
        If SKS = 1 Then SKS = 2
        If (SKS > Individu(Pop, i).Time - Individu(Pop, j).Time) Then
            Individu(Pop, j).Time = idx
        End If
    End If
End If
Next
Next
Next
End Sub

Sub MutasiSholat()
    Dim idx As Integer

    'Mutasi Sholat
    For Pop = 1 To Math.Floor(MR * PopSize) Mod PopSize
        For i = 1 To JlhKrom
            idx = AcakAngka(10, JlhTime)

            If (Individu(Pop, i).Time = 54) Or (Individu(Pop, i).Time = 55) Or
(Individu(Pop, i).Time = 56) Then
                Individu(Pop, i).Time = idx
            End If
        Next
    Next
End Sub

Sub MutasiLab()
    For Pop = 1 To Math.Floor(MR * PopSize) Mod PopSize
        For i = 1 To JlhKrom - 1
            'Jika SKS 2 Ruang Lab
            If (Individu(Pop, i).MK.SKS > 1 And Individu(Pop, i).Ruang >
TotalRuangan - TotalR_Lab) Then
                Individu(Pop, i).Ruang = AcakAngka(1, TotalR_Teori)
            'Jika SKS 1 Ruang Teori
            ElseIf (Individu(Pop, i).MK.SKS = 1 And Individu(Pop, i).Ruang <=
TotalR_Teori) Then
                Individu(Pop, i).Ruang = AcakAngka(TotalRuangan - TotalR_Lab + 1,
TotalRuangan)
            End If
        Next
    Next
End Sub

```

```

Sub SimpanKeDatabase(ByVal DG As DataGridView)
    Dim tmp As Integer = 0

    Using con = New OleDb.OleDbConnection(DBASE)
        Try
            con.Open()
            cmd = New OleDb.OleDbCommand
            cmd.CommandType = CommandType.Text
            cmd.Connection = con

            'Hapus Table Jadwal
            strSQL = "DROP TABLE Jadwal"
            cmd.CommandText = strSQL
            tmp = cmd.ExecuteNonQuery()
            'Buat Tabel Jadwal
            strSQL = "CREATE TABLE Jadwal (ID number, Kode char(30), Matkul
char(100), Semester number, SKS number, Dosen char(100), Kelas char(40), Ruang
char(40), Hari char(10), Waktu char(35))"
            cmd.CommandText = strSQL
            tmp = cmd.ExecuteNonQuery()
            Catch ex As Exception
                MsgBox(ex.Message)
            End Try
        End Using

        frmMDI.tsbProgress.Maximum = DG.Rows.Count - 1
        frmMDI.tsbProgress.Value = 0

        Using con = New OleDb.OleDbConnection(DBASE)
            Try
                con.Open()
                cmd = New OleDb.OleDbCommand
                cmd.CommandType = CommandType.Text
                cmd.Connection = con

                For i = 0 To DG.Rows.Count - 2
                    strSQL = "INSERT INTO Jadwal (ID, Kode, Matkul, Semester, SKS,
Dosen, Kelas, Ruang, Hari, Waktu) VALUES ('" & DG.Item(0, i).Value & "', '" &
DG.Item(1, i).Value & "', '" & DG.Item(2, i).Value & "', '" & DG.Item(3, i).Value &
"', '" & DG.Item(4, i).Value & "', '" & DG.Item(5, i).Value & "', '" & DG.Item(6,
i).Value & "', '" & DG.Item(7, i).Value & "', '" & DG.Item(8, i).Value & "', '" &
DG.Item(9, i).Value & "')"
                    cmd.CommandText = strSQL
                    tmp = cmd.ExecuteNonQuery()
                    frmMDI.tsbProgress.Value = frmMDI.tsbProgress.Value + 1
                    Application.DoEvents()
                Next
            Catch ex As Exception

```

```

        MsgBox(ex.Message)
    End Try
End Using

End Sub

Sub SolvedIndividu(ByVal K() As Kromosom, ByVal DG As DataGridView)
    Dim Hari As Integer
    For i = 0 To JlhKrom - 1
        Hari = Math.Floor(K(i + 1).Time / 10) - 1
        DG.Item(0, i).Value = K(i + 1).MK.ID
        DG.Item(1, i).Value = dgvMatkul.Item(1, i).Value
        DG.Item(2, i).Value = dgvMatkul.Item(2, i).Value
        DG.Item(3, i).Value = dgvMatkul.Item(4, i).Value
        DG.Item(4, i).Value = dgvMatkul.Item(5, i).Value
        DG.Item(5, i).Value = dgvDosen.Item(3, K(i + 1).MK.Dosen - 1).Value
        DG.Item(6, i).Value = dgvMatkul.Item(3, i).Value
        DG.Item(7, i).Value = dgvRuang.Item(1, K(i + 1).Ruang - 1).Value
        DG.Item(8, i).Value = dgvHari.Item(1, Hari).Value
        DG.Item(9, i).Value = dgvWaktu.Item(1, ((K(i + 1).Time Mod 10) + 1) -
1).Value
    Next

    'Simpan Ke Dalam Database
    If MsgBox("Simpan Hasil Proses Ke Dalam Database?", vbYesNo) = 6 Then
        SimpanKeDatabase(dgvReport)
    End If
End Sub

Public Function DGV2Excel(ByVal DGV As DataGridView) As Boolean
    Dim NamaFile As String = ""

    If MsgBox("Apakah Anda Ingin Menyimpan Hasil Dalam Bentuk File Excel?",
vbYesNo) = MsgBoxResult.No Then
        Return False
    End If

    If sfdExport.ShowDialog() = DialogResult.Cancel Then
        Return False
    End If

    NamaFile = sfdExport.FileName

    Try
        Dim DTB = New DataTable, RWS As Integer, CLS As Integer

        For CLS = 0 To DGV.ColumnCount - 1 ' COLUMNS OF DTB
            DTB.Columns.Add(DGV.Columns(CLS).HeaderText)

```

```

Next

Dim DRW As DataRow

For RWS = 0 To DGV.Rows.Count - 1 ' FILL DTB WITH DATAGRIDVIEW
    DRW = DTB.NewRow

    For CLS = 0 To DGV.ColumnCount - 1
        Try
            DRW(DTB.Columns(CLS).ColumnName.ToString) =
DGV.Rows(RWS).Cells(CLS).Value.ToString
        Catch ex As Exception

        End Try
    Next

    DTB.Rows.Add(DRW)
Next

DTB.AcceptChanges()

Dim DST As New DataSet
DST.Tables.Add(DTB)
Dim FLE As String = NamaFile ' PATH AND FILE NAME WHERE THE
XML WIL BE CREATED (EXEMPLE: C:\REPS\XML.xml)
DTB.WriteXml(FLE)
Dim EXL As String = "C:\Program Files\Microsoft
Office\Office15\Excel.exe" ' PATH OF/ EXCEL.EXE IN YOUR MICROSOFT
OFFICE
Shell(Chr(34) & EXL & Chr(34) & " " & Chr(34) & FLE & Chr(34),
vbNormalFocus) ' OPEN XML WITH EXCEL

Catch ex As Exception
    MsgBox(ex.ToString)
End Try
Return True
End Function

Private Sub btnProses_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnProses.Click
    Main()
End Sub

Private Sub btnExport_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExport.Click
    DGV2Excel(dgvReport)
End Sub
End Class

```

## DAFTAR RIWAYAT HIDUP



### Data Pribadi

---

Nama : YAKHDI PERARI PINEM  
 Jenis Kelamin : Laki-Laki  
 Tempat Tanggal Lahir: Medan, 30 Juni 1992  
 Kewarganegaraan : Indonesia  
 Status Perkawinan : Belum Kawin  
 Tinggi, Berat Badan : 181 cm, 81 kg  
 Agama : Islam  
 Alamat : Jl. Tembakau Raya No. 12 Lk. XVI Perumnas Simalingkar  
 Medan  
 Handphone : 083198662892  
 E-mail : yaqdee.frarie@gmail.com

### Pendidikan

---

1997 – 1998	R.A. Melati Perumnas Simalingkar Medan
1998 - 2004	SD Al-Azhar Medan
2004 - 2007	SMP Negeri 10 Medan
2007 - 2010	SMA Negeri 17 Medan
2010 - 2013	Diploma III Teknik Informatika di Universitas Sumatera Utara

### Pengalaman/Kegiatan

---

2012	Praktek Kerja Lapangan (PKL) di PT. Perkebunan Nusantara III (Persero)
2014	Seminar Nasional Literasi Informasi (SENARAI), Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara