

**OPTIMASI PENJADWALAN DAFTAR JAGA PERAWAT  
MENGUNAKAN ALGORITMA GENETIKA  
STUDI KASUS PADA RUMAH SAKIT "UNDATA" PALU**

**Skripsi**

oleh:

**RICO ANDI TJATJO**

**0310960066-96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN  
ALAM  
UNIVERSITAS BRAWIJAYA MALANG  
2008**

UNIVERSITAS BRAWIJAYA



]

**OPTIMASI PENJADWALAN DAFTAR JAGA PERAWAT  
MENGUNAKAN ALGORITMA GENETIKA  
STUDI KASUS PADA RUMAH SAKIT "UNDATA" PALU**

**Skripsi**

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

oleh:

**RICO ANDI TJATJO**  
**0310960066-96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA MALANG  
2008**

UNIVERSITAS BRAWIJAYA



**LEMBAR PENGESAHAN TUGAS AKHIR**  
**OPTIMASI PENJADWALAN DAFTAR JAGA PERAWAT**  
**MENGGUNAKAN ALGORITMA GENETIKA**  
**STUDI KASUS PADA RUMAH SAKIT "UNDATA" PALU**

oleh:  
**RICO ANDI TJATJO**  
**0310960066-96**

Setelah dipertahankan di depan Majelis Penguji  
pada tanggal 17 Januari 2008  
dan dinyatakan memenuhi syarat untuk memperoleh gelar  
Sarjana Komputer dalam bidang Ilmu Komputer

**Pembimbing I,**

**Pembimbing II,**

**Dian Eka Ratnawati, S.Si., M.Kom**  
**NIP. 132 300 224**

**Drs. Marji, MT**  
**NIP. 131 993 386**

**Mengetahui,**  
**a.n Ketua Jurusan Matematika**  
**Fakultas MIPA Universitas Brawijaya**  
**Sekretaris,**

**Dra. Ani Budi Astuti, M.Si**  
**NIP. 131 993 385**

UNIVERSITAS BRAWIJAYA



## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

**Nama** : Rico Andi Tjatjo  
**NIM** : 0310960066-96  
**Jurusan** : Matematika  
**Program Studi** : Ilmu Komputer  
**Penulis tugas akhir berjudul** : Optimasi penjadwalan daftar jaga perawat menggunakan algoritma genetika studi kasus pada rumah sakit “UNDATA” Palu.

Dengan ini menyatakan bahwa :

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

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 17 Januari 2008

Yang menyatakan,

Rico Andi Tjatjo  
NIM. 0310960066-96

UNIVERSITAS BRAWIJAYA



# OPTIMASI PENJADWALAN DAFTAR JAGA PERAWAT MENGUNAKAN ALGORITMA GENETIKA STUDI KASUS PADA RUMAH SAKIT “UNDATA” PALU

## ABSTRAK

Penjadwalan daftar jaga perawat merupakan sebuah proses penjadwalan yang mengatur waktu kerja perawat di sebuah rumah sakit. Dalam proses penjadwalan daftar jaga perawat ada beberapa komponen yang digunakan yaitu perawat, ruang, hari, dan shift. Pada tugas akhir ini dibahas mengenai penjadwalan daftar jaga perawat menggunakan algoritma genetika. Penelitian ini bertujuan untuk membuat model genetika untuk masalah penjadwalan daftar jaga perawat dan menerapkan algoritma genetika sebagai alternatif solusi masalah penjadwalan daftar jaga perawat dengan menggunakan perangkat lunak. Pada algoritma genetika teknik pencarian solusi menggunakan prinsip seleksi alam, dimana individu yang lebih kuat (*fitness* tinggi) akan memiliki tingkat reproduksi yang lebih tinggi. Implementasi algoritma genetika dalam aplikasi ini hanya mengatur masalah perawat dan slot waktu (hari dan shift), sedangkan masalah pembagian ruangan dilakukan dengan prosedur tersendiri. Hasil akhir dari penjadwalan daftar jaga perawat diperoleh dari kromosom yang memiliki nilai *fitness* terbaik. Penjadwalan daftar jaga perawat yang diterapkan pada aplikasi ini adalah menggunakan data yang diperoleh dari studi kasus pada rumah sakit “UNDATA” Palu. Berdasarkan uji coba yang telah dilakukan, diperoleh nilai probabilitas perkawinan silang dan probabilitas mutasi yang optimal digunakan pada masalah penjadwalan mata kuliah adalah 0,6 dan 0,4.



UNIVERSITAS BRAWIJAYA



# **OPTIMALISASION NURSE WORK TIME TIME-TABLING USING GENETIC ALGORITHM STUDY CASE AT “UNDATA” HOSPITAL**

## **ABSTRACT**

Nurse work time time-tabling is a process that arrange the nurses work time in a hospital. Nurse work time time-tabling involves components such as nurses, rooms, days and shifts. This final project discuss nurse work time time-tabling that using genetic algorithm. This study aims at making a genetic model for problems of nurse work time time-tabling and applying genetic algorithm as an alternative solution to solve nurse work time time-tabling problems using software. In genetic algorithm, the technique of searching solution uses the principles of natural selection, in which stronger individuals (higher fitness) will have higher level of reproduction. The implementation of genetic algorithm in this application is only to arrange the problem of nurses and time slot (days and shifts), whereas the problem of rooms distribution is carried out using a separate procedure. The final result of this course time-tabling is obtained from chromosomes which have the best fitness value. The data source of nurse work time time-tabling that used in this application is taken from “UNDATA” hospital in Palu. Based on the try-out which was done, the optimum value of the probability of crossover and the probability of mutation which was used on lecture time-tabling was 0.6 and 0.4.

UNIVERSITAS BRAWIJAYA



## KATA PENGANTAR

Puji syukur Penulis panjatkan kepada Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga Penulis dapat menyelesaikan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer

Tugas akhir ini bertujuan untuk menerapkan algoritma genetika sebagai alternatif solusi masalah penjadwalan daftar jaga perawat dengan menggunakan perangkat lunak.

Pada penyusunan tugas akhir ini, Penulis mengucapkan terima kasih kepada :

1. Ibu Dian Eka R., S.Si, M.Kom selaku pembimbing utama penulisan tugas akhir.
2. Bapak Drs. Marji, MT, selaku pembimbing pendamping dalam penulisan tugas akhir.
3. Bapak Wayan Firdaus Mahmudy, S.Si., MT, selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
4. Bapak Drs. Achmad Ridok, M.Kom selaku penasehat akademik.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan tugas akhir ini.
7. Orang tua Penulis atas dukungan materi dan doa restunya kepada Penulis.
8. Dan semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak dapat Penulis sebutkan satu per satu.

Penulis sadari bahwa masih banyak kekurangan dalam laporan ini, oleh karena itu Penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan

mutu isi tugas akhir ini untuk kelanjutan penelitian serupa di masa mendatang.

Semoga laporan tugas akhir ini dapat bermanfaat bagi para pembaca sekalian.

Malang, Januari 2008

Penulis

UNIVERSITAS BRAWIJAYA



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN .....	iii
LEMBAR PERNYATAAN .....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
DAFTAR LAMPIRAN .....	xxi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Penjadwalan .....	5
2.1.1 Penjadwalan Secara umum .....	5
2.1.2 Penjadwalan Perawat .....	5
2.2 Algoritma Genetika .....	6
2.3 Pengkodean .....	9
2.3.1 Pengkodean Biner .....	9
2.3.2 Pengkodean Permutasi .....	9
2.3.3 Pengkodean Nilai .....	9
2.3.4 Pengkodean Pohon .....	10
2.4 Seleksi .....	10
2.4.1 Seleksi Ranging .....	11
2.4.2 Seleksi Roda <i>Roulette</i> .....	12
2.4.3 Seleksi Turnamen .....	13
2.5 Operator Genetika .....	13
2.5.1 Perkawinan Silang .....	14

2.5.1.1 Perkawinan Silang Satu Titik .....	15
2.5.1.2 Perkawinan Silang Banyak Titik .....	15
2.5.1.3 Perkawinan Silang Permutasi .....	16
2.5.2 Mutasi .....	16

### **BAB III METODOLOGI DAN PERANCANGAN**

3.1 Analisa Permasalahan .....	19
3.2 Faktor-faktor yang Mempengaruhi Penjadwalan .....	19
3.3 Aturan-Aturan Penjadwalan .....	20
3.4 Tahapan Penyelesaian .....	20
3.4.1 Pengkodean Kromosom .....	20
3.4.2 Alokasi Ruang .....	22
3.5 Inisialisasi Kromosom .....	22
3.6 Fungsi <i>Fitness</i> .....	23
3.6.1 Perawat Yang Cuti Mendapatkan Waktu Bertugas ....	24
3.6.2 Perawat bertugas lebih dari satu kali dalam 1 hari ....	25
3.6.3 Perawat mendapat shift malam lebih dari tiga kali berturut-turut .....	25
3.6.4 Perawat mendapatkan shift pagi setelah mendapatkan shift malam .....	25
3.6.5 Perawat yang bertugas dalam satu shift di setiap ruang kurang dari 3 orang .....	25
3.7 Seleksi .....	26
3.8 Perkawinan Silang .....	26
3.9 Mutasi .....	27
3.10 Perbaikan kromosom .....	28
3.11 Alokasi Ruangan .....	28
3.12 Basis Data .....	28
3.12.1 Perancangan Basis Data .....	28
3.12.2 Struktur Tabel .....	29

### **BAB IV IMPLEMENTASI DAN PEMBAHASAN**

4.1 Implementasi .....	31
4.1.1 Input Data .....	31
4.1.1.1 Data Perawat .....	32
4.1.1.2 Data Ruang .....	32
4.1.1.3 Data Shift .....	33
4.1.1.4 Data Cuti .....	34

4.1.2 Deskripsi Program .....	34
4.1.2.1 Struktur Data .....	34
4.1.2.2 Inisialisasi Kromosom.....	35
4.1.2.3 Seleksi .....	37
4.1.2.4 Perkawinan Silang.....	38
4.1.2.5 Mutasi.....	39
4.1.2.6 Perbaikan kromosom.....	39
4.1.2.7 Pembagian Ruang.....	41
4.2 Penerapan Aplikasi .....	43
4.2.1 Pencarian Berdasarkan Id Perawat .....	45
4.2.2 Pencarian Berdasarkan Tanggal.....	46
4.2.3 Pencarian Berdasarkan Tanggal dan Shift.....	47
4.2.4 Pencarian Berdasarkan Tanggal dan Ruang .....	47
4.2.5 Pencarian Berdasarkan Tanggal, Shift dan Ruang.....	48
4.2.6 Pencarian Berdasarkan Shift.....	49
4.2.7 Pencarian Berdasarkan Shift dan Ruang.....	49
4.2.8 Pencarian Berdasarkan Ruang .....	50
4.3 Analisa Hasil.....	51
4.3.1 Pengaruh Jumlah Generasi Terhadap Nilai Fitness .....	51
4.3.2 Pengaruh Nilai Probabilitas Kawin Silang Terhadap Nilai Fitness.....	53
4.3.3 Pengaruh Nilai Probabilitas Mutasi Terhadap Nilai Fitness.....	54
<b>BAB V KESIMPULAN DAN SARAN</b>	
5.1 Kesimpulan.....	57
5.2 Saran.....	57
<b>DAFTAR PUSTAKA .....</b>	<b>59</b>
<b>LAMPIRAN.....</b>	<b>61</b>

UNIVERSITAS BRAWIJAYA



## DAFTAR GAMBAR

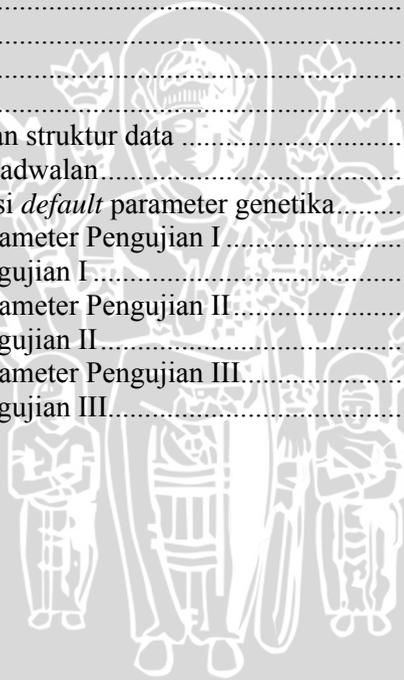
Gambar	2.1 Struktur Umum Algoritma Genetika .....	8
Gambar	2.2 <i>pseudo code</i> Algoritma Genetika.....	8
Gambar	2.3 Contoh kromosom pada pengkodean Biner.....	9
Gambar	2.4 Contoh kromosom dengan pengkodean Permutasi .....	9
Gambar	2.5 Contoh kromosom dengan pengkodean nilai.....	10
Gambar	2.6 Contoh kromosom dengan pengkodean pohon....	10
Gambar	2.7 Probabilitas suatu kromosom dalam roda <i>roulette</i>	12
Gambar	2.8 <i>Pseudo code</i> proses <i>crossover</i> .....	14
Gambar	2.9 Ilustrasi Penyilangan satu titik.....	15
Gambar	2.10 Ilustrasi Penyilangan banyak titik.....	15
Gambar	2.11 Ilustrasi Penyilangan permutasi.....	16
Gambar	2.12 <i>Pseudo code</i> proses mutasi.....	17
Gambar	3.1 Ilustrasi perkawinan silang.....	27
Gambar	3.2 Ilustrasi mutasi cara penukaran .....	27
Gambar	3.3 Skema basis data .....	28
Gambar	4.1 Tampilan utama aplikasi .....	31
Gambar	4.2 <i>Form</i> perawat .....	32
Gambar	4.3 <i>Form</i> ruang.....	33
Gambar	4.4 <i>Form</i> shift .....	33
Gambar	4.5 <i>Form</i> Cuti .....	34
Gambar	4.6 Struktur data .....	35
Gambar	4.7 Proses Inisialisasi.....	37
Gambar	4.8 Proses seleksi.....	38
Gambar	4.9 Proses perkawinan silang.....	39
Gambar	4.10 Proses Mutasi .....	39
Gambar	4.11 Proses perbaikan kromosom.....	41
Gambar	4.12 Proses pembagian ruang.....	42
Gambar	4.13 <i>Form</i> aktivitas genetika.....	44
Gambar	4.14 <i>Form</i> hasil akhir .....	45
Gambar	4.15 <i>Form</i> pencarian berdasarkan id perawat .....	46
Gambar	4.16 <i>Form</i> pencarian berdasarkan tanggal .....	46
Gambar	4.17 <i>Form</i> pencarian berdasarkan tanggal dan shift...	47
Gambar	4.18 <i>Form</i> pencarian berdasarkan tanggal dan ruang.	48
Gambar	4.19 <i>Form</i> pencarian berdasarkan tanggal, shift dan ruang.....	48

Gambar 4.20	<i>Form</i> pencarian berdasarkan shift .....	49
Gambar 4.21	<i>Form</i> pencarian berdasarkan shift dan ruang .....	50
Gambar 4.22	<i>Form</i> pencarian berdasarkan ruang .....	50
Gambar 4.23	Grafik Analisa Pengaruh Jumlah Generasi Terhadap Nilai Fitness .....	52
Gambar 4.24	Grafik Analisa Pengaruh Nilai Probabilitas Crossover Terhadap Nilai Fitness .....	54
Gambar 4.25	Grafik Analisa Pengaruh Nilai Probabilitas Mutasi Terhadap Nilai Fitness .....	56



## DAFTAR TABEL

Tabel 2.1 Keadaan sebelum di rangking .....	11
Tabel 2.2 keadaan setelah di rangking .....	11
Tabel 2.3 Contoh populasi dengan 5 kromosom .....	12
Tabel 3.1 Model Tahap 1 .....	21
Tabel 3.2 Model Tahap 2 .....	22
Tabel 3.3 Inisialisasi kromosom .....	23
Tabel 3.4 Aturan dan nilai pinalti .....	24
Tabel 3.5 Tperawat .....	29
Tabel 3.6 THari .....	29
Tabel 3.7 Tjadwal .....	29
Tabel 3.8 TShift .....	30
Tabel 3.9 TBulan .....	30
Tabel 3.10 TRuang .....	30
Tabel 3.11 Tcuti .....	30
Tabel 4.1 Keterangan struktur data .....	35
Tabel 4.2 Data Penjadwalan .....	43
Tabel 4.3 Kombinasi <i>default</i> parameter genetika .....	43
Tabel 4.4 Tabel Parameter Pengujian I .....	51
Tabel 4.5 Hasil pengujian I .....	51
Tabel 4.6 Tabel Parameter Pengujian II .....	53
Tabel 4.7 Hasil pengujian II .....	53
Tabel 4.8 Tabel Parameter Pengujian III .....	55
Tabel 4.9 Hasil pengujian III .....	55



UNIVERSITAS BRAWIJAYA



## DAFTAR LAMPIRAN

I.	Hasil Uji Coba 50 Generasi .....	61
II.	Hasil Uji Coba 100 Generasi .....	62
III.	Hasil Uji Coba 150 Generasi .....	63
IV.	Hasil Uji Coba 200 Generasi .....	64
V.	Hasil Uji Coba 250 Generasi .....	65
VI.	Hasil Uji Coba 300 Generasi .....	66
VII.	Jadwal Daftar Jaga Pada Instalasi Bedah Sentral Yang dihasilkan oleh aplikasi .....	68



UNIVERSITAS BRAWIJAYA



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Penjadwalan merupakan suatu kegiatan yang sangat erat kaitannya dengan suatu pekerjaan. Penjadwalan diperlukan untuk mengatur waktu kerja, sehingga menghasilkan sebuah jadwal yang efisien, dengan harapan dapat menghasilkan pekerjaan yang optimal. Sebuah penjadwalan akan tampak mudah jika komponen yang dijadwalkan dalam jumlah relatif sedikit, namun akan menjadi rumit jika komponen penyusunnya dalam jumlah yang besar.

Saat ini sudah banyak contoh penjadwalan yang digunakan dalam pekerjaan, antara lain penjadwalan produksi, penjadwalan pesawat, penjadwalan mata kuliah, penjadwalan pertandingan, penjadwalan daftar jaga perawat dan masih banyak lagi. Pada setiap jadwal yang terbentuk diharapkan dapat mengurangi konflik dan yang terpenting terjadi efisiensi sumber daya yang ada.

Pada penelitian ini akan dibahas penjadwalan daftar jaga perawat. Pembuatan jadwal daftar jaga perawat akan selalu muncul karena harus dilakukan pada setiap bulan. Umumnya jadwal daftar jaga perawat ini diselesaikan dengan membuat tabel jadwal secara manual. Cara ini membutuhkan waktu yang lama, karena pembuatan jadwal tersebut sangatlah kompleks yang terdiri dari beberapa komponen penyusun, seperti perawat, ruang, waktu bertugas dan waktu cuti perawat. Pada setiap komponen penyusun tersebut banyak terdapat aturan dan batasan-batasan yang telah ditentukan, oleh karena itu diperlukan penjadwalan otomatis yang dapat membuat jadwal dengan cepat, mudah dan tetap harus memperhatikan aturan-aturan.

Penjadwalan daftar jaga perawat bukanlah sebuah ide baru, saat ini telah banyak metode yang digunakan untuk menyelesaikan masalah penjadwalan daftar jaga perawat. Metode yang digunakan seperti, teknik *intelligent search*, metode *graph*, dan algoritma genetika. (Stéfan J. Darmoni, Alain Fajner, Nathalie Mahé, Arnaud Leforestier, Marc Vondracek, Olivier Stelian, Michel Baldenweck, 2002).

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme seleksi alami dan genetika alami (

Mitsui Gen, Cheng Runwei, 1997). Penulis memilih menggunakan algoritma genetika karena algoritma genetika berbeda dengan algoritma lain. Keunikan dari algoritma genetika adalah mengikuti pola evolusi makhluk hidup dan bilangan-bilangan yang dihasilkan secara random. Walaupun bilangan yang dihasilkan secara random, namun harus sesuai dengan aturan-aturan yang ditetapkan.

Selain itu, keunggulan algoritma generik dibandingkan algoritma lainnya adalah algoritma genetik sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan metode yang konvensional (Budi Sukmawan, 2003).

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan maka rumusan masalah yang dikerjakan adalah:

- Bagaimana membuat model genetika pada penjadwalan daftar jaga perawat yang mempertimbangkan waktu cuti perawat?
- Bagaimana mengimplementasikan algoritma genetika untuk pembuatan jadwal daftar jaga perawat dengan hasil yang optimal?

## **1.3 Batasan Masalah**

Pada tugas akhir ini dibatasi pada beberapa hal, antara lain:

- Studi kasus yang dipakai dalam aplikasi ini pada Rumah Sakit “UNDATA” Palu
- Setiap perawat hanya bertugas satu kali dalam satu hari.
- Jadwal daftar jaga perawat disusun tiap bulan.

## **1.4 Tujuan Penelitian**

Tujuan penelitian dalam tugas akhir ini adalah sebagai berikut :

1. Membuat model genetika untuk masalah penjadwalan daftar jaga perawat yang mempertimbangkan waktu cuti perawat.

2. Menerapkan algoritma genetika dalam pencarian alternatif solusi masalah penjadwalan daftar jaga perawat dengan menggunakan perangkat lunak.

### **1.5 Manfaat Penelitian**

Manfaat yang akan dicapai dari tugas akhir ini adalah untuk membantu menyelesaikan masalah penjadwalan daftar jaga perawat. Sehingga pembuatan jadwal daftar jaga akan lebih cepat, namun tetap memperhatikan aturan-aturan yang berlaku dalam aturan penjadwalan.

### **1.6 Sistematika Penulisan**

Pembuatan tugas akhir ini dilakukan dengan pembagian bab sebagai berikut:

#### **BAB I: PENDAHULUAN**

Pada bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan tugas akhir.

#### **BAB II: TINJAUAN PUSTAKA**

Bab ini menjelaskan tentang teori-teori yang berhubungan dengan penjadwalan daftar jaga perawat dan algoritma genetika. Adapun literatur yang digunakan meliputi buku referensi dan dokumentasi internet.

#### **BAB III: METODOLOGI DAN PERANCANGAN**

Pada bab ini dijelaskan metode-metode yang digunakan dalam menyelesaikan masalah penjadwalan daftar jaga perawat menggunakan algoritma genetika.

#### **BAB IV: HASIL DAN PEMBAHASAN**

Pada bab ini dijelaskan implementasi aplikasi, uji coba operator genetika dan analisa hasil.

## BAB V: PENUTUP

Bab lima berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan tugas akhir ini selanjutnya.

UNIVERSITAS BRAWIJAYA



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penjadwalan**

##### **2.1.1 Penjadwalan Secara Umum**

Penjadwalan adalah proses pembagian waktu berdasarkan rencana pengaturan urutan kerja dengan pembagian waktu pelaksanaan yang terperinci.

Perencanaan dan penjadwalan otomatis adalah cabang dari kecerdasan buatan yang mengacu pada realisasi dari strategi atau urutan pekerjaan. Tidak seperti pengendalian klasik dan masalah klasifikasi, solusinya kompleks, tidak diketahui dan harus ditemukan dan dioptimalkan pada ruang multidimensi (Wikipedia, 2000).

Ketepatan dalam penjadwalan merupakan kunci dari keefektifan dan keefesiansian ( Stéfán J. Darmoni, Alain Fajner, Nathalie Mahé, Arnaud Leforestier, Marc Vondracek, Olivier Stelian, Michel Baldenweck, 2002). Dalam matematika, masalah penjadwalan sering dipecahkan sebagai suatu masalah optimasi, dengan tujuan memaksimalkan mutu dari penjadwalan. Sebagai contoh, suatu perusahaan penerbangan ingin merancang rute penerbangan dari satu kota ke kota yang lain sehingga mendapatkan jarak minimum sehingga dapat mengurangi biaya yang dikeluarkan.

##### **2.1.2 Penjadwalan Perawat**

Penjadwalan perawat adalah suatu pekerjaan yang menentukan *shift-shift* para perawat setiap hari pada waktu tertentu agar dapat memenuhi kebutuhan terhadap perawat yang sesuai dengan persyaratan pada waktu tersebut. Keefektifan penjadwalan perawat sangat penting dalam mengontrol biaya perawatan kesehatan, karena mempengaruhi kualitas pelayanan pasien.

Penjadwalan perawat merupakan suatu pekerjaan yang sangat sulit dan menghabiskan banyak waktu, karena terdiri dari beberapa komponen penyusun, seperti perawat, ruang, waktu bertugas dan waktu cuti perawat. Semakin banyak komponen yang ada maka akan semakin banyak kombinasi dari komponen yang mungkin terjadi. Dan, yang terpenting dalam pemilihan kombinasi harus diperhatikan aturan-aturan yang telah ditetapkan ( Stéfán J.

Darmoni, Alain Fajner, Nathalie Mahé, Arnaud Leforestier, Marc Vondracek, Olivier Stelian, Michel Baldenweck, 2002).

Saat ini telah banyak metode yang digunakan untuk menyelesaikan masalah penjadwalan daftar jaga perawat. Metode yang digunakan seperti, teknik *intelligent search*, dan metode *graph* (Stéfan J. Darmoni, Alain Fajner, Nathalie Mahé, Arnaud Leforestier, Marc Vondracek, Olivier Stelian, Michel Baldenweck, 2002).

## 2.2 Algoritma Genetika

Algoritma genetik adalah algoritma pencarian yang berdasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. Dalam aplikasi algoritma genetik, variable solusi dikodekan kedalam struktur string yang merepresentasikan barisan gen, yang merupakan karakteristik dari solusi problem (Aries Syamsuddin, 2004).

Konsep dasar yang mengilhami timbulnya algoritma genetica adalah teori evolusi alam yang dikemukakan oleh Charles Darwin. Teori tersebut menjelaskan bahwa pada proses evolusi alami, setiap individu harus melakukan adaptasi terhadap lingkungan disekitarnya agar dapat bertahan hidup (Kuswara Setiawan, 2003).

Algoritma genetica pertama kali dikembangkan pada tahun 1975 oleh John Holland bersama rekan kerja dan murid-muridnya dari Universitas Michigan. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetica (Sri Kusumadewi, 2003).

Algoritma genetica sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode yang konvensional. Sebelum algoritma genetik dijalankan, masalah apa yang ingin dioptimalkan harus dinyatakan dalam fungsi tujuan, yang dikenal dengan fungsi fitness (Budi Sukmawan, 2003).

Berbeda dengan teknik pencarian konvensional, algoritma genetik berangkat dari himpunan solusi yang dihasilkan secara acak. Himpunan ini disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang

berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi dengan menggunakan alat ukur yang disebut fungsi *fitness* (Gen dan Cheng,1997). Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dari populasi tersebut (Sri Kusumadewi, 2003).

Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal,

Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Dalam membentuk generasi baru digunakan tiga operator, yaitu operator reproduksi/seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru.

3. Evaluasi solusi

Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti.

Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti yang sering digunakan antara lain:

- ❖ Berhenti pada generasi tertentu.
- ❖ Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi tidak berubah.
- ❖ Berhenti bila dalam  $n$  generasi berikut tidak didapatkan nilai *fitness* yang lebih tinggi (Budi Sukmawan, 2003).

Struktur umum algoritma genetik dapat dilihat pada gambar 2.1 (Koshimizu, Arai, 2003):



Gambar 2.1 Struktur Umum algoritma genetik

Apabila  $P(t)$  dan  $C(t)$  merupakan *parent* dan *offspring* pada generasi  $t$ , maka *pseudo code* dari algoritma genetika dapat dilihat pada Gambar 2.2: (Gen dan Cheng, 1997)

```

procedure AlgoritmaGenetika
begin
   $t \leftarrow 0$ ;
  inialisasi  $P(t)$ ;
  evaluasi  $P(t)$ ;
  while (bukan kondisi berhenti) do
    kombinasikan  $P(t)$  untuk menghasilkan  $C(t)$ ;
    evaluasi  $C(t)$ ;
    pilih  $P(t+1)$  dari  $P(t)$  dan  $C(t)$ ;
     $t \leftarrow t+1$ ;
  end
end
  
```

Gambar 2.2 *Pseudo Code* Algoritma Genetika

## 2.3 Pengkodean

Pengkodean adalah cara untuk merepresentasikan masalah ke dalam bentuk kromosom Berikut adalah beberapa jenis pengkodean yang umum digunakan (Kuswara Setiawan, 2003).

### 2.3.1 Pengkodean Biner

Pengkodean biner adalah pengkodean yang paling umum dan paling sederhana dalam merepresentasikan masalah pada algoritma genetika. Pada pengkodean biner setiap kromosom terdiri atas barisan string bit 0 atau 1. Sebagai contoh:

Kromosom A	0101101100010011
Kromosom B	1011010110110101

Gambar 2.3 Contoh kromosom pada pengkodean biner.

Contoh masalah yang sesuai menggunakan pengkodean biner adalah masalah nilai *maximize* pada sebuah fungsi matematika.

### 2.3.2 Pengkodean Permutasi

Pengkodean permutasi dapat digunakan pada masalah pengurutan data (*ordering problems*), seperti wiraniaga (*Travelling Salesman Problem*) atau masalah pengurutan tugas (*Task Ordering Problem*). Pada pengkodean permutasi, setiap kromosom terdiri dari barisan angka, yang merepresentasikan angka pada urutan.

Kromosom A	8549102367
Kromosom B	9102438576

Gambar 2.4 Contoh kromosom pada pengkodean permutasi.

### 2.3.3 Pengkodean Nilai

Pengkodean nilai dapat digunakan pada masalah yang sangat kompleks, dimana nilai yang dikodekan langsung merupakan representasi dari masalah. Penggunaan pengkodean biner pada tipe masalah yang kompleks akan menjadi lebih susah.

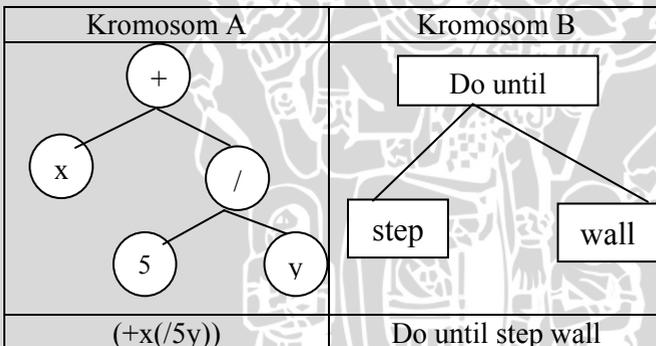
Pada pengkodean nilai, setiap kromosom adalah barisan dari beberapa nilai. Nilai dapat berupa apa saja, seperti bilangan biasa, bilangan riil, karakter sampai dengan obyek-obyek yang rumit.

Kromosom A	[red], [black], [blue], [yellow], [red]
Kromosom B	1.875, 3.9821, 9.1283, 6.8344, 4.116
Kromosom C	A, B, C, D, E, F, G, H, I, J, K, L, M, N

Gambar 2.5 Contoh kromosom pada pengkodean nilai.

### 2.3.4 Pengkodean Pohon

Pengkodean pohon lebih banyak digunakan untuk menyusun program atau ekspresi, bagi pemrograman genetika (*genetic programming*). Pada pengkodean pohon, setiap kromosom merupakan pohon dari sejumlah obyek, seperti fungsi atau perintah pada bahasa pemrograman.



Gambar 2.6 Contoh kromosom dengan pengkodean pohon.

## 2.4 Seleksi

Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu terpilih tersebut. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai fitness. Seleksi mempunyai tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai

nilai *fitness* terbaik. Ada beberapa metode seleksi dari induk, antara lain: (Sri Kusumadewi, 2003)

### 2.4.1 Seleksi Rangkings

Pada metode seleksi rangking, populasi diurutkan menurut nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya.

Proses dimulai dengan merangkings atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*nya kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan nilai terburuk akan memiliki *fitness* baru nilai 1, terburuk kedua bernilai 2 dan begitu seterusnya, sehingga kromosom yang memiliki *fitness* terbaik akan memiliki nilai *fitness* N, dimana N adalah jumlah kromosom di dalam populasi. Seperti dapat dilihat pada Gambar 2.6.

Tabel 2.1 Keadaan sebelum dirangkings

Kromosom	<i>Fitness</i>
A	25
B	10
C	15
D	10
E	10

Tabel 2.2 Keadaan setelah dirangkings

Kromosom	<i>Fitness</i>	<i>Fitness</i> baru
B	10	1
D	10	2
E	10	3
C	15	4
A	25	5

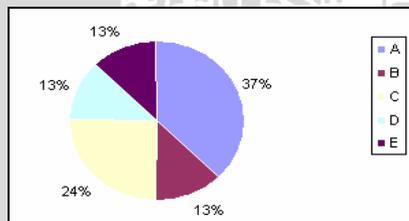
Metode ini memiliki kelebihan seluruh kromosom mempunyai kesempatan untuk dipilih. Akan tetapi, metode ini juga dapat menyebabkan konvergensi menjadi lambat, karena kromosom terbaik tidak terlalu berbeda dengan yang lainnya.

### 2.4.2 Seleksi Roda *Roulette*

Metode seleksi roda *roulette* merupakan metode yang paling sederhana, metode ini sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini individu-individu diurutkan berdasarkan ukuran *fitness*-nya, semakin besar nilai *fitness* maka akan semakin besar kemungkinannya untuk terpilih menjadi induk. Diandaikan semua kromosom diletakkan pada sebuah roda *roulette*, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*nya seperti pada contoh berikut:

Tabel 2.3 Contoh populasi dengan 5 kromosom

Kromosom	<i>Fitness</i>
A	25
B	10
C	15
D	10
E	10



Gambar 2.7 Probabilitas suatu kromosom dalam roda *roulette*

Pada Gambar 2.8 merupakan contoh dalam satu populasi terdiri dari lima kromosom. Pada tiap kromosom memiliki nilai *fitness* yang berbeda-beda. Pada Gambar 2.9 dapat diketahui

probabilitas terpilihnya masing-masing kromosom untuk menjadi induk. Pada kromosom A memiliki nilai *fitness* 25 dan nilai tersebut nilai *fitness* tertinggi pada populasi tersebut. Sehingga kromosom A memiliki probabilitas terbesar untuk terpilih menjadi induk.

Algoritma seleksi roulette wheel dapat dituliskan sebagai berikut:

1. Hitung total nilai *fitness* (F) :

$$TotFitness = \sum F_k$$

dimana  $k=1,2,\dots$  jumlah populasi

2. Hitung *fitness* relatif tiap individu :

$$P_k = \frac{F_k}{TotFitness}$$

3. Hitung *fitness* kumulatif :

$$q_1 = p_1$$

$$q_k = q_{k-1} + p_k$$

dimana  $k=2,3,\dots$  jumlah populasi

4. Pilih induk yang akan menjadi kandidat untuk di *crossover*.

### 2.4.3 Seleksi Turnamen

Seleksi turnamen merupakan variasi antara seleksi roda *roulette* dan seleksi rangking. Sejumlah  $k$  kromosom tertentu dari populasi beranggota  $n$  kromosom ( $k \leq n$ ) dipilih secara acak dengan probabilitas yang sama. Dari  $k$  kromosom yang terpilih kemudian akan dipilih satu kromosom dengan *fitness* terbaik, yang diperoleh dari hasil pengurutan rangking *fitness* semua kromosom terpilih. Perbedaan dengan seleksi roda *roulette* adalah pemilihan kromosom yang akan digunakan untuk berkembangbiak tidak berdasarkan skala *fitness* dari populasi.

### 2.5 Operator Genetika

Setelah proses seleksi dilakukan, proses selanjutnya adalah operasi genetika. Operator genetik dipergunakan untuk mengkombinasi (modifikasi) individu dalam aliran populasi guna

mencetak individu pada generasi berikutnya. Ada dua operator genetik yaitu crossover dan mutasi (Aries Syamsuddin, 2004).

### 2.5.1 Perkawinan Silang

Proses perkawinan silang (*crossover*) berfungsi untuk menghasilkan keturunan dari dua buah kromosom induk yang terpilih. Kromosom anak yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom induk (Kuswara Setiawan, 2003). Secara umum, mekanisme tukar silang adalah sebagai berikut:

1. memilih dua buah kromosom sebagai induk.
2. memilih secara acak posisi dalam kromosom, biasa disebut titik perkawinan silang, sehingga masing-masing kromosom induk terpecah menjadi dua segmen.
3. lakukan pertukaran antar segmen kromosom induk untuk menghasilkan kromosom anak.

Apabila  $v_k$  merupakan individu dalam populasi  $k$  maka *pseudo code* dari proses *crossover* dapat dilihat pada Gambar 2.6. (Gen dan Cheng, 1997).

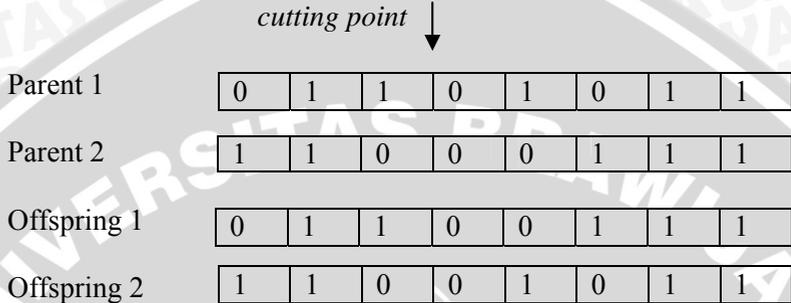
```
procedure Crossover
begin
  k ← 0;
  while (k ≤ jumlah populasi) do
     $r_k$  ← bangkitkan bilangan acak antara [0,1];
    if ( $r_k$  < probabilitas crossover) then
      pilih  $v_k$  sebagai parent untuk crossover
    end
    k ← k+1;
  end
end
```

Gambar 2.8 *Pseudo Code* Proses *Crossover*

Beberapa metode perkawinan silang antara lain : (Sri Kusumadewi, 2003)

### 2.5.1.1 Perkawinan silang satu titik (Single-point crossover)

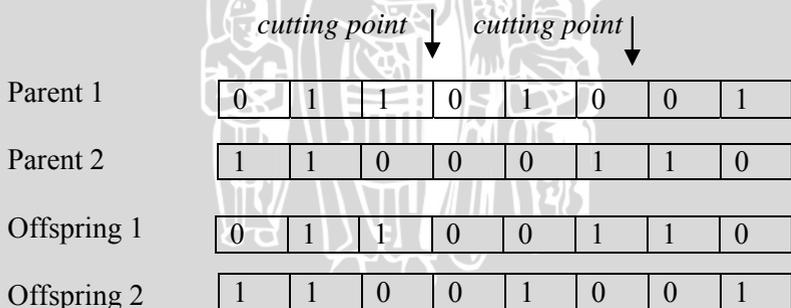
Pada metode ini , posisi penyilangan dengan panjang kromosom diseleksi secara acak. Variabel – variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak.



Gambar 2.9 Ilustrasi Penyilangan satu titik

### 2.5.1.2 Perkawinan silang banyak titik ( Multi-point crossover)

Pada penyilangan banyak titik, posisi penyilangan dengan panjang kromosom diseleksi secara acak dan tidak diperbolehkan ada posisi yang sama, serta diurutkan naik. Dimana jumlah titik tidak boleh lebih dari panjang kromosom. Variabel – variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan *offspring*.

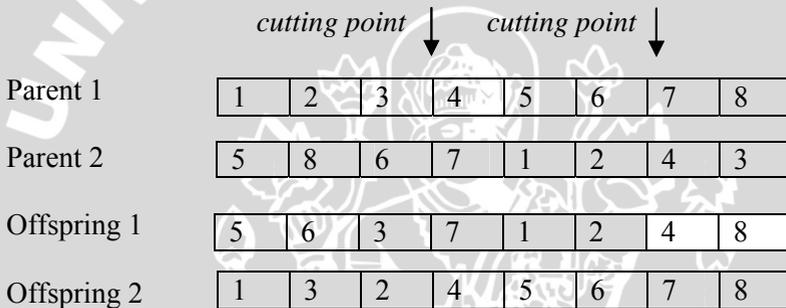


Gambar 2.10 Ilustrasi Penyilangan banyak titik

### 2.5.1.3 Perkawinan silang permutasi

Pada pengkodean permutasi, perkawinan silang yang sering digunakan adalah perkawinan silang satu titik, karena selain prosesnya yang sederhana juga dapat menjaga konsistensi urutan nilai pada kromosom.

Proses perkawinan silang satu titik dimulai dengan pemilihan satu titik perkawinan silang. Dari permutasi pertama sampai dengan titik perkawinan silang disalin dari induk pertama, sedangkan sisanya didapatkan dengan cara melihat satu persatu nilai pada orang tua kedua, jika belum ada pada kromosom keturunan, maka nilai tersebut ditambahkan (Sri Kusumadewi, 2003).



Gambar 2.11 Ilustrasi Penyilangan permutasi

### 2.5.2 Mutasi

Setelah melalui proses perkawinan silang, pada *offspring* dapat dilakukan proses mutasi. Variabel *offspring* dimutasi dengan menambahkan nilai random yang sangat kecil, dengan probabilitas yang rendah. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inialisasi populasi. Sehingga meningkatkan variasi populasi (Sri Kusumadewi, 2003).

Tujuan dari mutasi adalah agar individu-individu yang ada dalam populasi semakin bervariasi. Mutasi akan sangat berperan jika pada populasi awal hanya ada sedikit solusi yang mungkin terpilih. Sehingga, operasi itu sangat berguna dalam mempertahankan keanekaragaman individu dalam populasi meskipun dengan mutasi tidak dapat diketahui apa yang terjadi pada individu baru (Kuswara Setiawan, 2003).

*Pseudo Code* dari proses mutasi dapat dilihat pada Gambar 2.11 (Gen dan Cheng, 1997)

```
procedure Mutasi
begin
  i ← 0;
  while (i ≤ jumlah populasi) do
    pilih satu kromosom secara acak;
    ambil dua gen secara acak;
    tukar posisi kedua gen;
    i ← i+1;
  end
end
```

Gambar 2.12 *Pseudo Code* Proses Mutasi



UNIVERSITAS BRAWIJAYA



## **BAB III**

### **METODOLOGI DAN PERANCANGAN**

#### **3.1 Analisa Permasalahan**

Sistem penjadwalan daftar jaga perawat pada Rumah Sakit "UNDATA" Palu masih menggunakan sistem penjadwalan secara manual. Hal ini tentu saja memakan waktu yang sangat lama dan rumit. Dalam menentukan *shift* tersebut ada aturan-aturan yang harus dipenuhi agar perawat-perawat dapat bekerja secara optimal dan pasien mendapatkan kenyamanan selama dirawat di rumah sakit tersebut.

Mengingat pentingnya proses penjadwalan terhadap kegiatan keperawatan di rumah sakit, maka diperlukan suatu sistem penjadwalan daftar jaga perawat yang optimal, efektif dan efisien. Algoritma genetika dapat digunakan untuk menyelesaikan permasalahan tersebut karena algoritma genetika mempunyai banyak keunggulan dibandingkan metode-metode lain yang mungkin juga dapat menyelesaikan masalah penjadwalan. Karena dalam proses algoritma genetika, jadwal diolah dengan menggunakan teknik komputasi yang sederhana dan hanya didasarkan pada proses random. Selain itu, proses dalam algoritma genetika mempertahankan jadwal yang dianggap terbaik dimana tidak melanggar kriteria pembatas atau paling sedikit melanggar kriteria pembatas.

#### **3.2 Faktor-Faktor yang Mempengaruhi Penjadwalan**

Pada pembuatan jadwal daftar jaga perawat terdapat beberapa komponen utama, antara lain:

- Perawat  
Rumah Sakit "UNDATA" Palu memiliki 62 orang perawat.
- Ruang  
Rumah Sakit "UNDATA" Palu memiliki 6 ruangan perawatan. Ruangan tersebut adalah Instalasi Kamar Bersalin, Instalasi Gawat Darurat, Instalasi Bedah Sentral, dan Instalasi Rawat Inap Kelas I, II, dan III.
- Waktu

Waktu yang disediakan dibagi menjadi tiga *shift*. *Shift* pagi dimulai pukul 07.00 sampai dengan pukul 14.00, *shift* siang dimulai pukul 14.00 sampai dengan pukul 21.00, dan *shift* malam dimulai pukul 21.00 sampai dengan pukul 07.00. Selain itu, ada juga waktu yang disediakan untuk libur dan cuti.

### **3.3 Aturan-Aturan Penjadwalan**

Penjadwalan dibuat berdasarkan aturan-aturan yang telah ditetapkan. Agar penjadwalan daftar jaga perawat dapat tersusun dengan baik, maka terdapat beberapa aturan-aturan yang harus diperhatikan, antara lain:

1. Dalam satu hari minimal ada 3 perawat yang bertugas tiap *shift* di setiap instalasi.
2. Setiap Perawat hanya bertugas maksimal satu kali setiap hari.
3. Setiap Perawat tidak boleh mendapatkan *shift* pagi setelah bertugas pada *shift* malam di hari sebelumnya.
4. Setiap Perawat tidak boleh mendapat *shift* malam lebih dari tiga kali berturut-turut.
5. Setiap perawat mendapatkan waktu libur 4 hari dalam satu bulan.
6. Perawat yang mengajukan cuti tidak boleh mendapatkan waktu bertugas pada waktu cutinya.

### **3.4 Tahapan Penyelesaian**

Penyelesaian masalah penjadwalan daftar jaga perawat akan diselesaikan dengan dua tahap. Pembagian penyelesaian menjadi dua tahap karena perawat tidak dapat meminta instalasi atau ruang yang akan digunakan untuk bertugas. Dua tahap yang digunakan dalam menyelesaikan penjadwalan ini adalah :

1. Pengkodean Kromosom
2. Alokasi ruang

#### **3.4.1 Pengkodean Kromosom**

Tahap pertama ini akan diselesaikan dengan menggunakan algoritma genetika. Pengkodean yang akan digunakan pada

penjadwalan daftar jaga perawat ini adalah pengkodean nilai. Pemilihan pengkodean nilai karena jika dilakukan pengkodean biner akan membuat barisan bit semakin kompleks, dimana suatu barisan bit yang panjang sehingga diperlukan pengkodean ulang untuk mendapatkan nilai sebenarnya dari aturan yang direpresentasikan.

Komponen penjadwalan daftar jaga perawat yang dimasukkan dalam pengkodean kromosom adalah perawat dan waktu. Waktu memiliki dua komponen, yaitu tanggal dan *shift*. Sehingga model genetika dari penjadwalan daftar jaga perawat ini terdiri dari perawat, tanggal, dan *shift*.

Perawat disusun secara berurutan dalam kromosom, karena setiap perawat harus mendapat alokasi waktu. Pada tahap ini terdapat banyak aturan-aturan yang berhubungan dengan variabel waktu. Setelah seluruh perawat menempati slot waktu, maka langkah selanjutnya adalah pada tahap kedua yaitu mengalokasikan ruang.

Berikut contoh dari tahap pertama :

Tabel 3.1 Model tahap 1

P1				P2				P3				Pn			
T1	T2	...	Tm												

Keterangan:

P : Perawat yang bertugas  
 Contoh : P1 = Kasmiyati  
 P2 = Eni Titin Purwanti  
 P3 = Restu Widyarini  
 n = Jumlah perawat

T : Waktu bertugas ( tanggal, *shift*)  
 Contoh : T1 ( 3,2) = tanggal 3 *shift* siang  
 T2 (22,1) = tanggal 22 *shift* pagi  
 T3 (1,3) = tanggal 1 *shift* malam  
 m = jumlah kombinasi waktu

Pada Tabel 3.1 dapat dilihat tahap pertama, dimana dilakukan proses menggunakan algoritma genetika, sehingga menghasilkan slot waktu untuk masing-masing perawat (T1,T3,...,Tm).

### 3.4.2 Alokasi Ruang

Tahap kedua ini akan diselesaikan dengan menggunakan pemrograman biasa. Pada tahap ini dilakukan penempatan ruang pada slot waktu yang telah ada.

Berikut contoh dari tahap kedua :

Tabel 3.2 Model tahap 2

T1				T2				T3				Tn			
R1	R2	...	Rm												

Keterangan:

T : Waktu bertugas ( tanggal, *shift*)  
 Contoh : T1 ( 3,2) = tanggal 3 *shift* siang  
 T2 (22,1) = tanggal 22 *shift* pagi  
 T3 (1,3) = tanggal 1 *shift* malam  
 n = jumlah kombinasi waktu

R : Instalasi  
 Contoh: R1 = Instalasi Gawat Darurat  
 R2 = Instalasi Kamar Bersalin  
 R3 = Instalasi Rawat Inap Kelas I  
 m = jumlah ruangan

Pada Tabel 3.2 adalah proses tahap kedua. Pada tahap kedua ruangan yang tersedia ditempatkan pada slot waktu dari seluruh perawat.

### 3.5 Inisialisasi Kromosom

Inisialisasi kromosom direpresentasikan dalam bentuk larik dengan tipe data *record* yang berisi data yang mendukung proses penjadwalan. Panjang dari kromosom adalah sebanyak gen yang ada, dalam hal ini setiap gen mewakili waktu untuk bertugas tiap perawat..

Setiap kromosom adalah barisan gen yang terdiri dari dua nilai yaitu nilai tanggal dan *shift*. Nilai gen didapatkan dengan membangkitkan bilangan secara random. Bilangan yang dirandom adalah nilai tanggal dan nilai *shift*. Nilai tanggal dinyatakan dalam

bilangan *byte* 1 sampai *n* hari untuk mewakili jumlah hari dalam satu bulan. Sedangkan nilai *shift* dibangkitkan nilai 1 sampai 3 yang mewakili 3 slot *shift* yang tersedia dalam satu hari. Sehingga dalam satu bulan terdapat  $n \times 3$  slot waktu yang tersedia. Ilustrasi inialisasi kromosom:

Tabel 3.3 Inialisasi kromosom

Id_Perawat	1	2	3	Id ke- n
Kromosom 1	G1	G2	G3	Gn
Kromosom 2	G11	G12	G13	G1n
.....	...	...	...	...
Kromosom n	Gn1	Gn2	Gn3	Gnn

↓

T1	T2	T3	...	Tn
----	----	----	-----	----

Pada Tabel 3.3 merupakan ilustrasi dari inialisasi kromosom. Setelah terjadi proses genetika dihasilkan kromosom-kromosom yang terdiri dari beberapa gen (G) yang jumlahnya sesuai dengan jumlah perawat. Gen tersebut berisi slot waktu T (tanggal dan *shift*) untuk setiap perawat yang jumlahnya sesuai dengan jumlah hari bertugas. Jumlah hari bertugas adalah jumlah hari dalam satu bulan yang dikurangi dengan 4 hari libur. Sehingga di setiap inialisasi akan dihasilkan kromosom yang panjangnya adalah jumlah perawat dikalikan jumlah hari bertugas.

### 3.6 Fungsi *Fitness*

Setelah individu-individu dalam populasi telah terbentuk, maka langkah selanjutnya adalah menghitung nilai *fitness* setiap individu. Penghitungan dilakukan dengan memberikan pinalti untuk setiap aturan yang digunakan dalam penjadwalan. Semakin wajib aturan dilaksanakan, maka akan semakin besar nilai pinalti yang diberikan. Sebuah kromosom tidak layak untuk dijadikan jadwal apabila pinalti yang dihasilkan terlalu besar. Berikut aturan penghitungan fungsi *fitness* (Ross,1994):

$$f(g) = 1 / (1 + \sum P_i v_i(g));$$

dimana  $P_i$  adalah pinalti (pelanggaran) yang diberikan untuk aturan  $i$ ,  $v_i(g) = 1$  jika melanggar aturan  $i$ , dan sebaliknya jika tidak melanggar aturan  $i$  maka  $v_i(g) = 0$ .

Dari penghitungan nilai *fitness* dapat diketahui bahwa semakin sedikit aturan yang dilanggar, maka akan semakin besar nilai *fitness*-nya. Jadwal yang sempurna akan memiliki nilai *fitness* 1, karena nilai total aturan yang dilanggar adalah 0.

Pada penjadwalan perawat yang akan dibuat diberikan sejumlah aturan beserta pinalti dari setiap aturan sebagai berikut:

Tabel 3.4 Aturan dan nilai pinalti

Aturan	Nilai Pinalti
Perawat yang cuti mendapatkan waktu bertugas	4
Perawat bertugas lebih dari satu kali dalam 1 hari	4
Perawat mendapat shift malam lebih dari tiga kali berturut-turut	3
Perawat mendapatkan <i>shift</i> pagi setelah mendapatkan <i>shift</i> malam	3
Perawat yang bertugas dalam satu shift di setiap ruang kurang dari 3 orang	2

### 3.6.1 Perawat Yang Cuti Mendapatkan Waktu Bertugas

Perawat yang mengajukan waktu cuti tidak boleh bertugas di waktu di cuti. Jika hal ini dilanggar maka diberi pinalty 4.

1. **[Cek Perawat]** Mengecek waktu bertugas setiap Perawat.
2. **[Penilaian]** Membandingkan apakah perawat tersebut mendapatkan waktu bertugas saat perawat tersebut cuti.

### 3.6.2 Perawat bertugas lebih dari satu kali dalam 1 hari

Seorang perawat tidak dapat bertugas lebih dari satu kali dalam satu hari. Jika tidak maka akan dikenakan pinalti 4.

1. **[Cek Perawat]** Mengecek shift dari setiap perawat.
2. **[Penilaian]** Membandingkan apakah seorang perawat memiliki lebih dari satu kali penugasan dalam 1 hari.

### **3.6.3 Perawat Mendapat Shift Malam lebih dari tiga kali berturut-turut**

Seorang Perawat tidak boleh mendapatkan *shift* malam lebih dari 3 kali berturut-turut. Jika ditemukan perawat yang bertugas malam lebih dari 3 kali berturut-turut maka diberi pinalty 3.

1. **[Cek Perawat]** Mengecek banyak shift malam dari setiap Perawat.
2. **[Penilaian]** Membandingkan apakah perawat tersebut mendapatkan *shift* malam lebih dari tiga kali berturut-turut.

### **3.6.4 Perawat mendapatkan *shift* pagi setelah mendapatkan *shift* malam**

Seorang perawat tidak boleh mendapatkan *shift* pagi atau siang setelah mendapat *shift* malam. Jika hal ini terjadi maka akan diberi pinalty 3.

1. **[Cek Perawat]** Mengecek shift tiap Perawat.
2. **[Penilaian]** Membandingkan apakah perawat tersebut mendapatkan *shift* pagi setelah mendapat *shift* malam.

### **3.6.5 Perawat yang bertugas dalam satu shift di setiap ruang kurang dari 3 orang**

Jumlah perawat yang bertugas di setiap ruang di setiap shift minimal 3 orang. Jika tidak maka akan dikenakan pinalty 2.

1. **[Cek Kromosom]** Mengecek gen yang terdiri dari hari dan shift tiap kromosom.

2. **[Penilaian]** Membandingkan apakah di dalam kromosom jumlah gen yang kembar kurang dari 18 gen.

### 3.7 Seleksi

Seleksi mempunyai peranan penting dalam algoritma genetika, karena pada proses ini dipilih induk yang digunakan untuk menghasilkan individu baru. Seleksi yang digunakan pada penelitian ini adalah seleksi roda *roulette*. Pada seleksi roda *roulette*, semakin tinggi nilai *fitness* maka semakin besar kemungkinan untuk terpilih menjadi induk.

Langkah-langkah dari seleksi roda *roulette* pada penelitian ini adalah sebagai berikut :

1. **[Acak]** Bangkitkan bilangan acak
2. **[Pemilihan Parent]** Bandingkan nilai acak dengan peluang kumulatif individu ke-*i* bila nilai acak lebih kecil dari peluang kumulatifnya maka tetapkan individu ke-*i* sebagai *parent*.
3. **[Ulangi]** Kembali ke langkah pertama sampai terbentuk seluruh *parent*.

### 3.8 Perkawinan Silang

Apabila proses seleksi telah dilaksanakan dan sudah terpilih induk baru, maka operator berikutnya adalah perkawinan silang. Perkawinan silang adalah cara mengkombinasikan gen-gen induk untuk menghasilkan keturunan baru. Perkawinan silang yang digunakan adalah perkawinan silang satu titik. Pada perkawinan ini dilakukan dengan cara menukar nilai gen pada posisi gen yang sama dari kedua induk. Penukaran gen tersebut juga harus dilakukan pengecekan apakah individu baru yang terbentuk sesuai dengan aturan yang berlaku.

Kromosom 1 :

13,2	22,1	17,3	3,3	2,1	1,3	10,2
------	------	------	-----	-----	-----	------

Kromosom 2:

4,3	28,2	2,3	15,1	11,2	9,1	27,2
-----	------	-----	------	------	-----	------

Anak :

13,2	22,1	17,3	3,3	11,2	9,1	27,2
------	------	------	-----	------	-----	------

Gambar 3.1 Ilustrasi perkawinan silang

Langkah-langkah perkawinan silang ini adalah sebagai berikut :

1. **[Acak]** Bangkitkan bilangan acak antara 0 sampai dengan panjang kromosom sebagai titik perkawinan silang
2. **[Pembentukan Anak Pertama]** Salin gen dari titik kromosom awal sampai dengan titik perkawinan silang dari orang tua pertama, sisa gennya disalin dari orang tua kedua
3. **[Pembentukan Anak Kedua]** Salin gen dari titik kromosom awal sampai dengan titik perkawinan silang dari orang tua kedua, sisa gennya disalin dari orang tua pertama

### 3.9 Mutasi

Setelah dilakukan proses perkawinan silang maka langkah selanjutnya adalah melakukan mutasi. Mutasi dilakukan untuk mencegah terjadinya konvergensi prematur. Mutasi yang digunakan adalah dengan menukar langsung nilai dari gen. Gen yang digunakan pada proses mutasi dipilih secara random.

Ilustrasi mutasi cara penukaran:

Sebelum mutasi:

11,3	23,1	23,3	2,2	13,3	13,1	25,1
------	------	------	-----	------	------	------

Sesudah mutasi:

11,3	23,1	13,3	2,2	23,3	13,1	25,1
------	------	------	-----	------	------	------

Gambar 3.2 Ilustrasi mutasi cara penukaran

### **3.10 Perbaikan Kromosom**

Perbaikan kromosom dilakukan untuk mencegah agar tidak ada perawat yang bertugas lebih dari satu kali dalam satu hari. Pada perbaikan kromosom ini gen setiap perawat diamati apakah ada 2 atau lebih gen yang memiliki hari yang sama, apabila ada maka tukar salah satu gen dengan gen milik perawat lain.

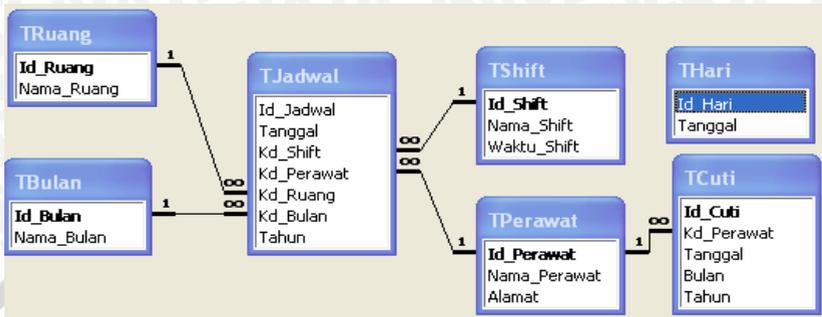
### **3.11 Alokasi Ruang**

Setelah semua slot waktu telah selesai di proses dengan menggunakan algoritma genetika, langkah selanjutnya adalah menentukan ruang jaga. Pada tahap ini ruang jaga langsung dimasukkan ke dalam masing – masing slot waktu yang telah tersedia secara random namun mengikuti aturan alokasi. Aturan alokasi antara lain pada ruang jaga yang berbeda tidak boleh ada perawat yang sama dan seluruh slot waktu pada setiap ruang jaga harus terisi.

### **3.12 Basis Data**

#### **3.12.1 Perancangan Basis Data**

Pada perancangan basis data, ada beberapa tabel yang saling berkaitan untuk menyediakan data yang dibutuhkan oleh sistem. Pada Gambar 3.3 terdapat 8 tabel yaitu tabel TPerawat, TRuang, THari, TJadwal, TSkill, TShift, Tcuti dan TBulan. Skema basis datanya dapat dilihat pada Gambar 3.3.



Gambar 3.3 Skema basis data

### 3.12.2 Struktur Tabel

#### 1. Tabel TPerawat.

Nama Tabel : TPerawat

Fungsi : Menyimpan data perawat

Tabel 3.5 TPerawat

No	Nama Field	Tipe Data	Keterangan
1	<b>Id_Perawat</b>	<b>Text</b>	<b>PK</b>
2	Nama_Perawat	Text	Not Null
3	Alamat	Text	-

#### 2. Tabel THari.

Nama Tabel : THari

Fungsi : Menyimpan data jumlah hari.

Tabel 3.6 THari

No	Nama Field	Tipe Data	Keterangan
1	Id Hari	Number	-
2	Tanggal	Text	-

#### 3. Tabel TJadwal.

Nama Tabel : TJadwal

Fungsi : Menyimpan data jadwal yang sudah terbentuk.

Tabel 3.7 TJadwal

No	Nama Field	Tipe Data	Keterangan
----	------------	-----------	------------

1	Id Jadwal	Number	Not Null
2	Tanggal	Number	Not Null
3	Kd Shift	Text	FK
4	Kd Perawat	Text	FK
5	Kd Ruang	Text	FK
6	Kd Bulan	Text	FK
7	Tahun	Text	Not Null

4. Tabel TShift.

Nama Tabel : TShift

Fungsi : Menyimpan data shift.

Tabel 3.8 TShift

No	Nama Field	Tipe Data	Keterangan
1	<b>Id_Shift</b>	<b>Text</b>	<b>PK</b>
2	Nama Shift	Text	Not Null
3	Waktu Shift	Text	Not Null

5. Tabel TBulan.

Nama Tabel : TBulan

Fungsi : Menyimpan data nama bulan.

Tabel 3.9 TBulan

No	Nama Field	Tipe Data	Keterangan
1	<b>Id_Bulan</b>	<b>Text</b>	<b>PK</b>
2	Nama Bulan	Text	Not Null

6. Tabel TRuang.

Nama Tabel : TRuang

Fungsi : Menyimpan data Ruang/instalasi.

Tabel 3.10 TRuang

No	Nama Field	Tipe Data	Keterangan
1	<b>Id_Ruang</b>	<b>Text</b>	<b>PK</b>
2	Nama Ruang	Text	Not Null

7. Tabel Tcuti

Nama Tabel : Tcuti  
Fungsi : Menyimpan data cuti perawat

Tabel 3.11 TCuti

No	Nama Field	Tipe Data	Keterangan
1	Id_Cuti	Text	PK
2	Kd_Perawat	Text	FK
3	Tanggal	Text	Not Null
4	Bulan	Text	Not Null
5	Tahun	Text	Not Null

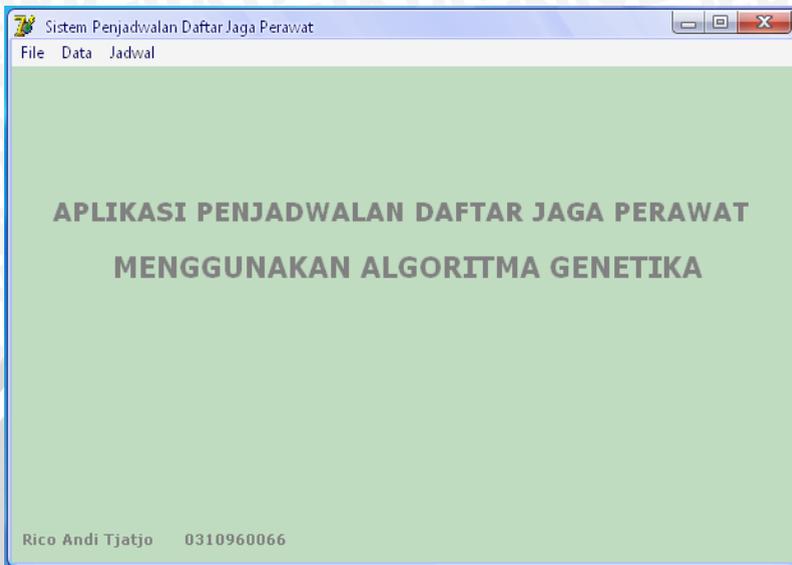
## BAB IV IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Implementasi

Implementasi perangkat lunak ini berupa aplikasi pemrograman yang menerapkan metode algoritma genetika untuk menentukan jadwal daftar jaga perawat. Perangkat lunak ini dibuat dengan menggunakan bahasa pemrograman *Borland Delphi 7.0* yang dijalankan pada Sistem Operasi Windows.

Tampilan utama dari aplikasi penjadwalan daftar jaga perawat menggunakan algoritma genetika dapat dilihat pada Gambar 4.1 berikut ini:





Gambar 4.1 Tampilan utama aplikasi

#### **4.1.1 Input Data**

Input data terdiri dari komponen utama penjadwalan yang meliputi garda depan, gerai, hari, shift, dan kesediaan waktu garda depan. Berikut penjelasan masing-masing komponen beserta tampilan:

##### **4.1.1.1 Data Perawat**

*Form* Perawat digunakan untuk menyimpan data Perawat yang terdiri dari id perawat, nama perawat dan alamat perawat.

Data Perawat

No Id :

Nama :

Alamat :

ID	Nama	Alamat
1	Kasmiyati	Jln. Kesehatan No. 7
10	Titik Emawati	Jln. Nuri No. 23
11	Defi Andriyanto	Jln. Diponegoro No. 3
12	Mila Faista	Jln. Ahmad Yani No. 37
13	Evi Ningrum	Jln. Wutukanjai No. 12
14	Winarti	Jln. Rajawali No. 23
15	Ekosih Widayati	Jln. Gatot Subroto No. 233
16	Nuning Dwi Wati	Jln. Raden Saleh No. 1
17	Mandawati	Jln. Kartini No. 34
18	Ninik Nailun Nusroh	Jln. Kakatua No. 4
19	Riza Dia Mayasari	Jln. Cendrawasih No. 13
2	Eri Titin Purwanti	Jln. W. Monginsidi No. 56
20	Ana Afifah	Jln. Tadulako No. 78
21	Denis Daryati	Jln. Ki Hajar Dewantara No. 33

Tambah

Edit

Hapus

Refresh

Exit

Gambar 4.2 *Form Perawat*

#### 4.1.1.2 Data Ruang

*Form Ruang* digunakan untuk menyimpan data ruang yang terdiri dari id ruang dan nama ruang.

Data Ruang

Id Ruang :

Nama Ruang :

ID	Ruang
1	Instalasi Bedah Sentral
2	Instalasi Gawat Darurat
3	Instalasi Kamar Bersalin
4	Instalasi Rawat Inap Kelas I
5	Instalasi Rawat Inap Kelas II
6	Instalasi Rawat Inap Kelas III

Tambah

Edit

Hapus

Refresh

Exit

Gambar 4.3 *Form Ruang*

#### 4.1.1.3 Data Shift

*Form Shift* digunakan untuk menyimpan data Shift yang terdiri dari id shift, nama shift dan waktu shift.

ID	Shift	Waktu
1	Pagi	07.00 - 14.00
2	Siang	14.00 - 21.00
3	Malam	21.00 - 07.00

Gambar 4.4 *Form Shift*

#### 4.1.1.4 Data Cuti

*Form cuti* digunakan untuk menyimpan data cuti perawat yang terdiri dari id cuti, kode perawat, nama perawat, tanggal cuti, bulan cuti dan tahun cuti.

Id_Cuti	Kd_Perawat	Nama	Tanggal	Bulan	Tahun
1	1	Kasmiyati	1	1	2007
2	1	Kasmiyati	2	1	2007
3	34	Ila Lutfia	21	1	2007

Gambar 4.5 *Form Cuti*

## 4.1.2 Deskripsi Program

### 4.1.2.1 Struktur Data

Struktur data pada aplikasi penjadwalan daftar jaga perawat ini dapat direpresentasikan sebagai berikut:

```

TIndividu = record
    kromosom: array of String;
    Cost :Real;
    Fitness :Real;
    P_Relatif :Real;
    P_Kom :integer;
end;

```

Individu: array of TIndividu;  
 ComHariShift: array of String;  
 idPerawat: array of String;  
 parental: array of integer;

Gambar 4.6 Struktur data

Struktur data pada Gambar 4.6 dapat dijelaskan pada Tabel 4.1 sebagai berikut:

Tabel 4.1 Keterangan struktur data

TIndividu	Menyimpan data yang terdiri dari kromosom, cost, fitness, P Relatif dan P Kom
Kromosom	Array yang menyimpan data hasil random slot waktu yang terdiri atas tanggal dan shift
Cost	Nilai pinalti yang timbul akibat pelanggaran terhadap aturan dari tiap individu
Fitness	Nilai fitness dari tiap individu
P Relatif	Nilai fitness relatif dari setiap individu
P Kom	Nilai fitness kumulatif dari setiap individu
Individu	Array yang digunakan untuk menyimpan data individu
ComHariShift	Array yang digunakan untuk menyimpan data bilangan awal slot waktu yang terdiri atas tanggal dan shift
idPerawat	Array yang menyimpan data perawat
Parental	Array yang menyimpan data individu yang menjadi parent.

#### 4.1.2.2 Inisialisasi Kromosom

Inisialisasi kromosom dalam program adalah dengan membangkitkan nilai random tanggal dan waktu shift sebanyak Id perawat yang ada. Cuplikan dari proses inisialisasi dapat dilihat pada Gambar 4.7.

```
ADOQueryPerawat.First;
for i:= 0 to ADOQueryPerawat.recordcount-1 do
```

```

begin
    jumhari := ADOqueryhari.RecordCount-4;
    ADOQuerycuti.First;
    for j := 0 to ADOquerycuti.RecordCount-1 do
        begin
            if adoquerycuti.FieldValues['Kd_Perawat']=
ADOqueryperawat.FieldValues['Id_Perawat'] then
                begin
                    jumhari := jumhari-1;
                end;
            ADOQuerycuti.Next;
        end;
        for tgl:= 0 to jumhari-1 do
            begin
                len:= len + 1;
                SetLength(idPerawat,len);
                idPerawat[len-1]:=
aDOQueryPerawat.FieldValues['Id_Perawat'];
            end;
            ADOQueryPerawat.Next;
        end;

        SetLength(Individu,SpinEditInd.Value);
        for jumIndividu:= 0 to SpinEditInd.Value-1 do
            begin
                len:= 0;
                ADOqueryperawat.First;
                for i:= 0 to ADOQueryPerawat.RecordCount - 1 do
                    begin
                        jumhari := ADOqueryhari.RecordCount-4;
                        ADOQuerycuti.First;
                        for j := 0 to ADOquerycuti.RecordCount-1 do
                            begin
                                if adoquerycuti.FieldValues['Kd_Perawat']=
ADOqueryperawat.FieldValues['Id_Perawat'] then
                                    begin
                                        jumhari := jumhari-1;
                                    end;
                            end;
                    end;
                end;
            end;
        end;
    end;

```

```

ADOquerycuti.Next;
end;
for tgl:= 0 to jumhari-1 do
begin
  len:= len + 1;
  SetLength(Individu[jumIndividu].kromosom,len);
  Randomize;
  index:= RandomRange(0,length(ComHariShift));
  Individu[jumIndividu].kromosom[len-1]:=
ComHariShift[index];
  DelKombinasi(index);
end;
Adoqueryperawat.Next;
end;
Individu[jumIndividu].Cost:=
TotalCost(Individu[jumIndividu].kromosom);
Individu[jumIndividu].Fitness:=
1/(1+Individu[jumIndividu].Cost);
getHariShift;
end;

```

Gambar 4.7 Proses Inisialisasi

### 4.1.2.3 Seleksi

Seleksi dimulai dengan menentukan jumlah parent. Kemudian lakukan penelusuran dimulai indek  $i=0$  sampai jumlah parent. Setelah itu bangkitkan sebuah nilai  $r$  secara random. Kemudian lakukan penelusuran dimulai dengan indek  $j=0$  sampai jumlah individu. Jika nilai  $r$  lebih kecil dari nilai peluang kumulatif individu ke- $j$  maka dapatkan individu ke- $j$  sebagai parent ke- $i$ . Cuplikan program untuk memilih induk dapat dilihat pada gambar 4.8.

```

SetLength(range,100);
for i:= 0 to 99 do
begin
  range[i]:= i;

```

```

end;

SetLength(parental,jumParent);
//pemilihan parent
for i:= 0 to jumParent - 1 do
begin
  Randomize;
  ran:= RandomRange(0,length(range));
  for j:= 0 to Length(Individu) - 1 do
  begin
    if range[ran] <= Individu[j].P_Kom then
    begin
      Parental[i]:= j;
      Break;
    end;
  end;
end;
end;

```

Gambar 4.8 Proses Seleksi

#### 4.1.2.4 Perkawinan silang

Pada *crossover* dimulai dengan memilih kromosom yang akan *dicrossover*. Kemudian menentukan titik potong secara random dengan range maksimum panjang kromosom. Kemudian dilakukan penukaran bagian kanan titik potong dari kedua parent untuk menghasilkan kromosom anak. Cuplikan dari prosedur perkawinan silang dapat dilihat pada gambar 4.9.

```

SetLength(potongan1,jumCrossOver,cutPoin);
SetLength(anak,jumCrossOver,Length(Individu[0].kromosom));
for i:= 0 to jumCrossOver - 1 do
begin
  for j:= 0 to cutPoin - 1 do
  begin
    anak[i,j]:= potongan1[i,j];
  end;
end;

len:= Length(Individu[0].kromosom)-cutPoin + 1;

```

```

SetLength(potongan2,jumCrossOver,len);
for j:= cutPoin to
    Length(Individu[0].kromosom) - 1 do
begin
    if i < (jumCrossOver - 1) then
        anak[i,j]:= potongan2[i+1,j-cutPoin]
    else
        anak[i,j]:= potongan2[0,j-cutPoin];
end;
end;
SetLength(tempAnak,length(Individu[0].kromosom));
for i:= 0 to jumCrossOver - 1 do
begin
    for j:= 0 to Length(Individu[0].kromosom) -
        1 do
begin
    tempAnak[j]:= anak[i,j];
end;
GetCostCrossover(tempAnak);
end;

```

Gambar 4.9 Proses Perkawinan Silang

#### 4.1.2.5 Mutasi

Mutasi yang dilakukan yaitu dengan cara menukar nilai dua gen. Langkah awal adalah menentukan dua gen yang akan dimutasi, kemudian langsung menukar nilai kedua gen tersebut. Cuplikan dari prosedur mutasi dapat dilihat pada gambar 4.10.

```

for i:= 0 to SpinEditInd.Value-1 do
begin
    for j:= 0 to length(Individu[0].kromosom) - 1 do
begin
    gen[j]:= Individu[i].kromosom[j];
end;
end;
end;

```

```

for i := 0 to jumMutasi-1 do
begin
s1:= RandomRange(0,Totalgen-1);
repeat
    S2:= RandomRange(0,Totalgen-1);
until (S2 <> s1);
temp:= gen[s1];
gen[s1]:= gen[s2];
gen[s2]:= temp;
end;

```

Gambar 4.10 Proses Mutasi

#### 4.1.2.6 Perbaikan Kromosom

Perbaikan kromosom yang dilakukan yaitu dengan cara menukar gen perawat A dengan gen perawat B hingga semua perawat tidak memiliki 2 atau lebih gen yang memuat hari yang sama. Cuplikan prosedur perbaikan kromosom dapat dilihat pada gambar 4.11

```

For i:= 0 to length(newpop.kromosom)-1 do
begin
gen[i]:=newpop.kromosom[i];
if idPer<>idPerawat[i] then
begin
idPer := idPerawat[i];
batas := i;
awal := i - (adoqueryhari.recordcount-4);
end;

if (batas>0) and (i mod (adoqueryhari.RecordCount-4) = 0) then
begin
for n := 0 to adoqueryhari.RecordCount-1 do
begin
jum := 0;
for j := awal to batas-1 do
begin

```

```

s1:="";
s1:= gen[j];
s2:= s1[1]+s1[2];
if s2[2] = ',' then
begin
  s2 :=s1[1];
end;
if strtoint(s2) = n+1 then
begin
  jum := jum+1
end;
end;
if jum>1 then
begin
  for k := awal to batas-1 do
  begin
    s1:="";
    s1:= newpop.kromosom[k];
    s2:= s1[1]+s1[2];
    if s2[2] = ',' then
    begin
      s2 :=s1[1];
    end;
    if strtoint(s2)=n+1 then
    begin
      titik1 := k;
      break;
    end;
  end;
end;
repeat
  y := randomrange(0,
                    length(newpop.kromosom)-1);
  s3:="";
  s3:= newpop.kromosom[y];
  s4:= s3[1]+s3[2];
  if s3[2] = ',' then
  begin
    s4 :=s3[1];
  end;
end;

```

```

end;
jum := 0;
for j := awal to batas-1 do
begin
  s1:= "";
  s1:= newpop.kromosom[j];
  s2:= s1[1]+s1[2];
  if s2[2] = ',' then
  begin
    s2 :=s1[1];
  end;
  if s2 = s4 then
  begin
    jum := jum+1
  end;
end;
until jum=0;
temp := newpop.kromosom[titik1];
newpop.kromosom[titik1] :=
  newpop.kromosom[y];
newpop.kromosom[y] :=temp;
end;
end;
end;
end;

```

Gambar 4.11 Proses Perbaikan Kromosom

#### 4.1.2.7 Pembagian Ruang

Pembagian ruangan dilakukan setelah tiap perawat memiliki slot waktu. Pemberian alokasi ruangan dilakukan urut mulai ruangan yang pertama sampai yang terakhir. Cuplikan prosedur pembagian ruang dapat dilihat pada gambar 4.12

```

Adoqjadwal.First;
for i:= 0 to adoqjadwal.RecordCount-1 do
begin

```

```

adoqueryshift.First;
for j := 0 to Adoqueryshift.RecordCount-1 do
begin
  tgl1:= "";
  cek1:= "";
  tgl1:= Adoqjadwal.Fieldvalues['Id_Hari'];
  cek1:= adoqueryshift.FieldValues['Id_Shift'];
  qcek :='SELECT * From Tjadwal Where Kd_Shift =
""+cek1+"" and Tanggal = '+tgl1+' and Kd_Bulan =
""+inttostr(Comboboxbulan.itemindex)+"" and Tahun =
""+comboboxtahun.Text+"" Order By Tanggal,Kd_Shift ';
  ADOquery1.close;
  adoquery1.SQL.Clear;
  adoquery1.SQL.Append(qcek);
  adoquery1.Open;
  Adoquery1.First;
  for k := 0 to Adoquery1.RecordCount-1 do
  begin
    ADOquery1.Edit;
    if (k mod 6 = 0) then
    begin
      ADOQueryRuang.First;
      ADOquery1.FieldName('Kd_Ruang').Value:=
ADOQueryRuang.FieldValues['Id_Ruang'];
    end
    else
    begin
      ADOQueryRuang.Next;
      ADOquery1.FieldName('Kd_Ruang').Value:=
ADOQueryRuang.FieldValues['Id_Ruang'];
    end;
    ADOquery1.Post;
    ADOquery1.Next;
  end;
  adoqueryshift.Next;
end;
adoqjadwal.Next;
end;

```

## Gambar 4.12 Proses Pembagian Ruang

### 4.2 Penerapan Aplikasi

Aplikasi diterapkan dengan memasukkan data penjadwalan sesuai dengan keinginan *user*. Dalam kasus ini data yang dimasukkan adalah sebagai berikut:

Tabel 4.2 Data penjadwalan

Perawat	62
Jumlah Hari	31
Ruang	6

Pada Tabel 4.2 dapat diketahui bahwa terdapat 62 perawat, 31 hari dan 6 ruang untuk bertugas.

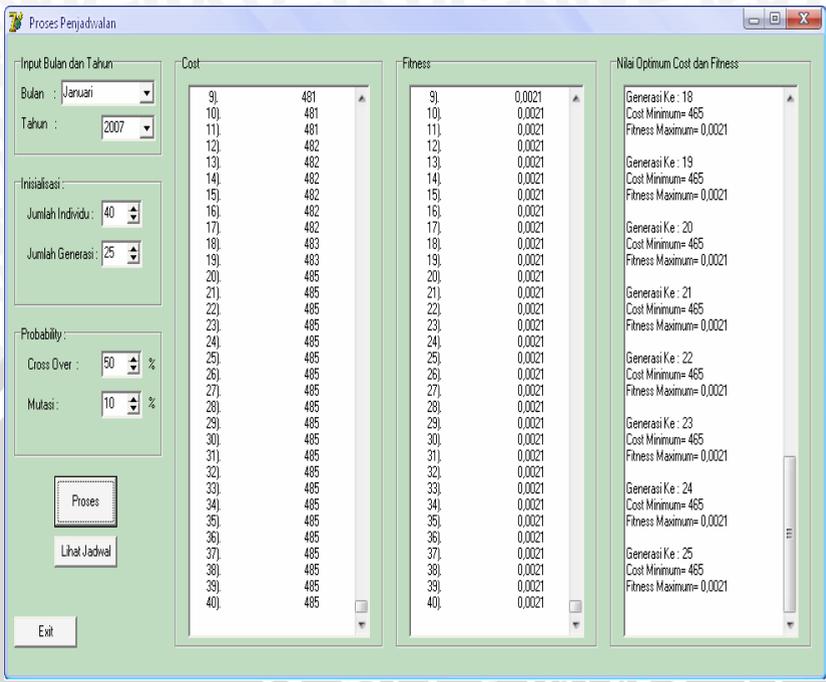
Langkah selanjutnya untuk pembuatan jadwal mata pelajaran adalah dengan mengisikan nilai parameter genetika. Secara *default* aplikasi memiliki kombinasi parameter seperti pada Tabel 4.3.

Tabel 4.3 Kombinasi *default* parameter genetika

Populasi	40
Generasi	25
Perkawinan silang	0,5
Mutasi	0,1

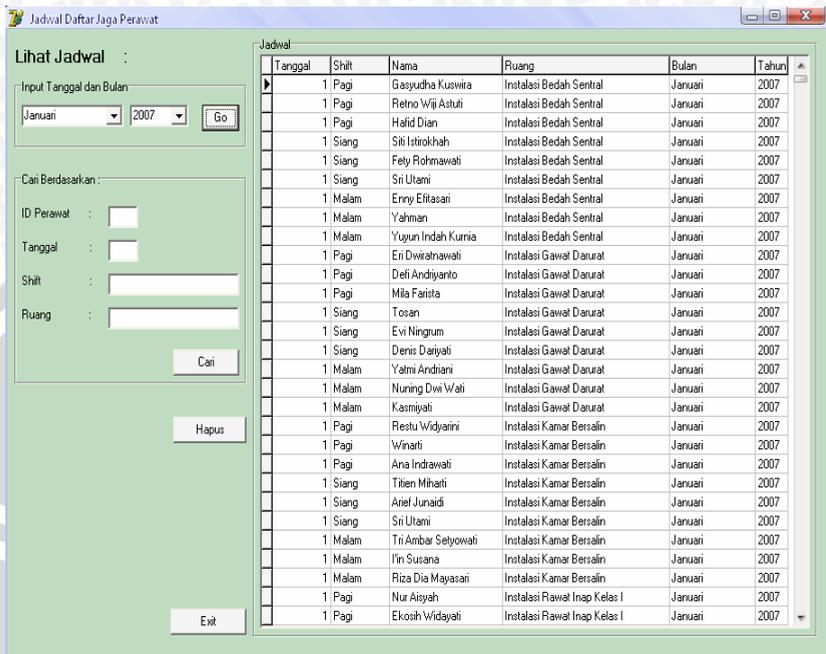
Pada Tabel 4.3 dapat dijelaskan bahwa jumlah generasi atau iterasi yang dilakukan adalah sebanyak 25 generasi. Setiap generasi menghasilkan 40 populasi atau 40 kromosom. Probabilitas *crossover* adalah sebesar 0.5. Diharapkan ada 20 kromosom dari 40 kromosom yang ada dalam populasi tersebut mengalami *crossover*.

Apabila nilai parameter sudah disimpan, maka proses selanjutnya adalah pembuatan jadwal. Pada Gambar 4.12 dapat dilihat proses pembuatan jadwal. Pada saat program dijalankan akan terlihat adanya perubahan nilai *fitness* mulai dari inisialisasi populasi sampai dengan generasi terakhir.



Gambar 4.13 Form aktivitas genetika

Jadwal daftar jaga perawat diperoleh dari kromosom yang memiliki nilai *fitness* terbaik pada sebuah generasi. Penjadwalan daftar jaga perawat yang diterapkan pada aplikasi diperoleh kromosom yang memiliki nilai *cost* terendah sebesar 465 dan nilai *fitness* tertinggi sebesar 0,0021. Kromosom tersebut terpilih untuk dijadikan jadwal daftar jaga perawat.



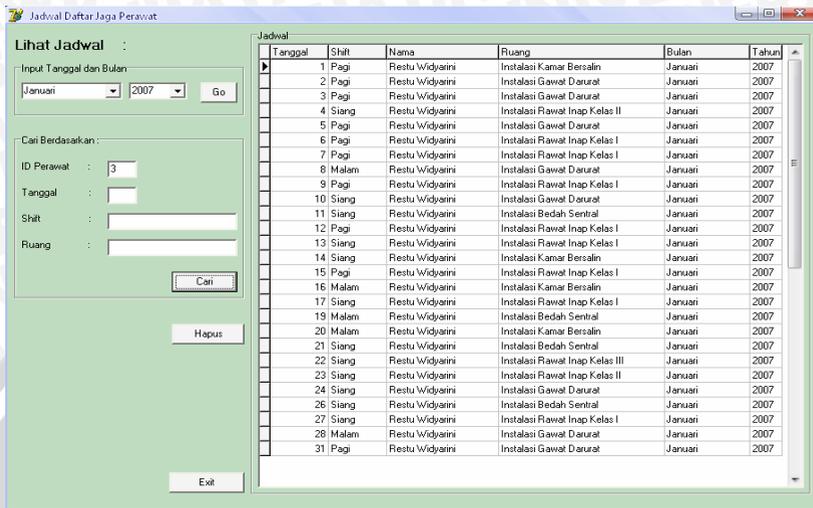
Gambar 4.14 Form hasil akhir

Pada Gambar 4.13 adalah hasil dari penjadwalan Perawat . Setiap perawat ditempatkan pada slot waktu (tanggal dan shift) dan ruang yang digunakan. Contohnya, perawat yang bernama Gasyudha Kuswira ditempatkan pada slot waktu Tanggal 1 shift pagi di Instalasi Bedah Sentral. Contoh jadwal daftar jaga perawat pada instalasi bedah sentral lebih lengkapnya dapat dilihat pada lampiran halaman 68.

Selain *Input Data*, pada aplikasi penjadwalan daftar jaga perawat ini terdapat juga proses untuk menampilkan jadwal berdasarkan tanggal bertugas, nama perawat, shift, ataupun berdasarkan ruang bertugas. Berikut penjelasan dari proses pencarian masing-masing komponen :

#### 4.2.1 Pencarian Berdasarkan Id Perawat

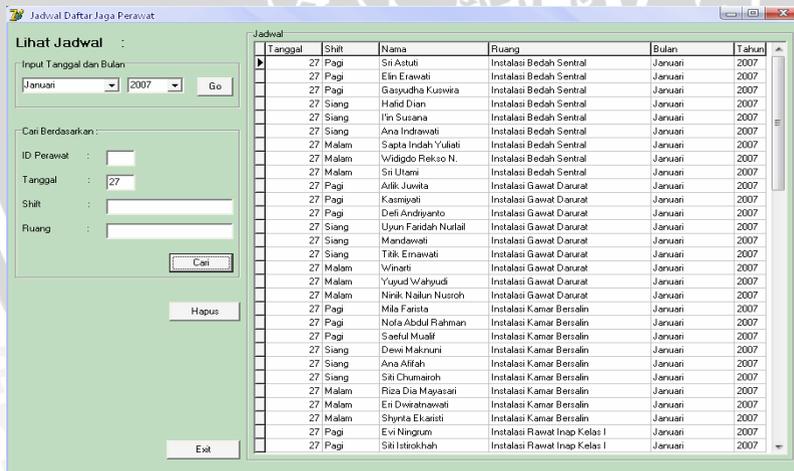
Pencarian ini akan menampilkan jadwal sesuai dengan Id perawat yang diinputkan oleh *user*.



Gambar 4.15 Form Pencarian Berdasarkan Id perawat

## 4.2.2 Pencarian Berdasarkan Tanggal

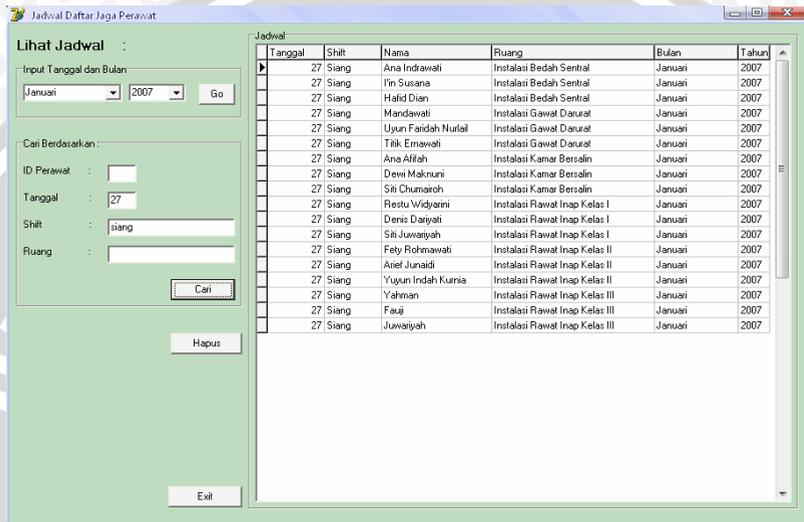
Pencarian ini akan menampilkan jadwal sesuai dengan tanggal yang diinginkan oleh *user*.



Gambar 4.16 Pencarian Berdasarkan Tanggal

### 4.2.3 Pencarian Berdasarkan Tanggal dan Shift

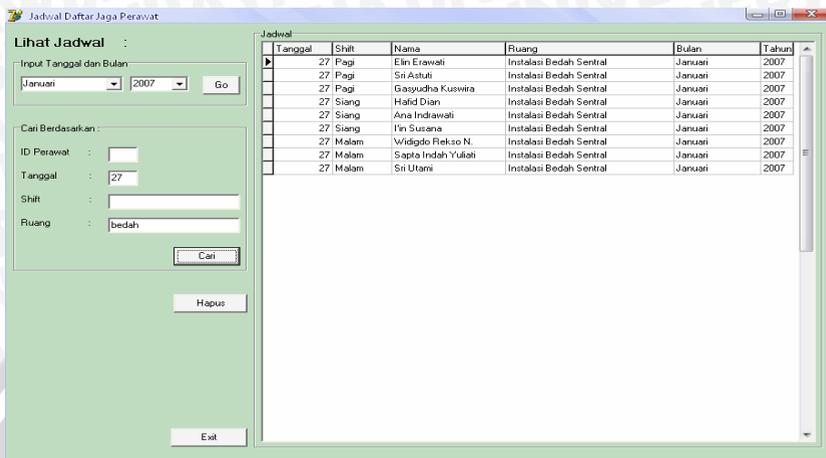
Pencarian ini akan menampilkan jadwal sesuai dengan tanggal dan shift yang diinginkan oleh *user*.



Gambar 4.17 Pencarian Berdasarkan Tanggal dan Shift

### 4.2.4 Pencarian Berdasarkan Tanggal dan Ruang

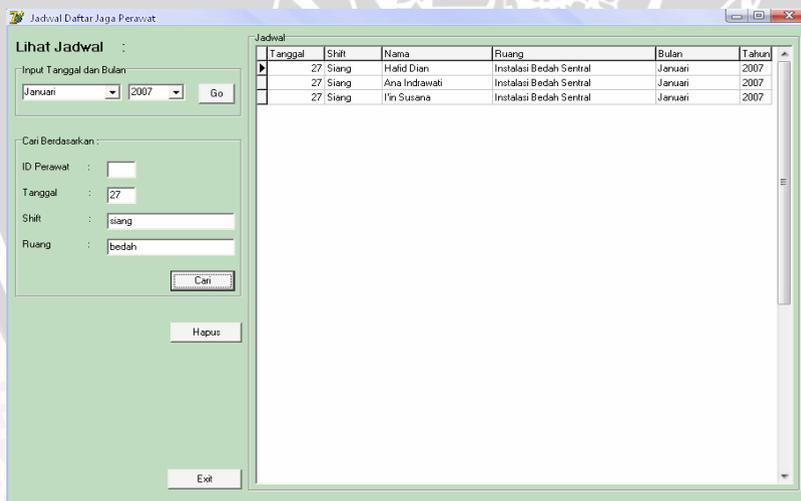
Pencarian ini akan menampilkan jadwal sesuai dengan tanggal dan ruang yang diinginkan oleh *user*.



Gambar 4.18 Pencarian Berdasarkan Tanggal dan Ruang

#### 4.2.5 Pencarian Berdasarkan Tanggal, Shift dan Ruang

Pada pencarian ini, jadwal yang ditampilkan akan diurutkan berdasarkan Tanggal, nama shift dan nama ruang yang diinputkan oleh *user*.



Gambar 4.19 Pencarian Berdasarkan Tanggal, Shift dan Ruang

## 4.2.6 Pencarian Berdasarkan Shift

Pada pencarian ini, jadwal yang ditampilkan akan diurutkan berdasarkan nama shift yang diinputkan oleh *user*.

The screenshot shows a software window titled "Jadwal Daftar Jaga Perawat". On the left, there is a search panel labeled "Lihat Jadwal" with the following fields:

- Input Tanggal dan Bulan: January 2007
- Cari Berdasarkan: ID Perawat, Tanggal, Shift (set to "malam"), Ruang
- Buttons: Cari, Hapus, Exit

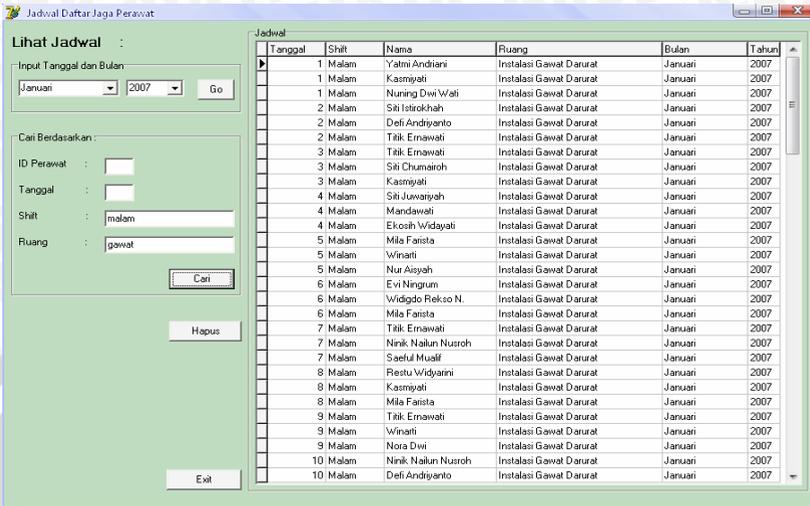
On the right, a table titled "Jadwal" displays the search results. The table has columns for Tanggal, Shift, Nama, Ruang, Bulan, and Tahun. The data is sorted by shift name.

Tanggal	Shift	Nama	Ruang	Bulan	Tahun
1 Malam	Enny Ertasari	Instalasi Bedah Sentral	Januari	2007	
1 Malam	Yuyun Indah Kurnia	Instalasi Bedah Sentral	Januari	2007	
1 Malam	Yahman	Instalasi Bedah Sentral	Januari	2007	
1 Malam	Kasmyati	Instalasi Gawat Darurat	Januari	2007	
1 Malam	Yanni Andiani	Instalasi Gawat Darurat	Januari	2007	
1 Malam	Nuning Dwi Wati	Instalasi Gawat Darurat	Januari	2007	
1 Malam	Iin Susana	Instalasi Kamar Bersalin	Januari	2007	
1 Malam	Riza Dia Mayasari	Instalasi Kamar Bersalin	Januari	2007	
1 Malam	Tri Ambar Setyowati	Instalasi Kamar Bersalin	Januari	2007	
1 Malam	Ana Afifah	Instalasi Rawat Inap Kelas I	Januari	2007	
1 Malam	Achmad Zamroni	Instalasi Rawat Inap Kelas I	Januari	2007	
1 Malam	Abdul Rauf	Instalasi Rawat Inap Kelas I	Januari	2007	
1 Malam	Masrucha	Instalasi Rawat Inap Kelas II	Januari	2007	
1 Malam	Syahnil Anawani	Instalasi Rawat Inap Kelas II	Januari	2007	
1 Malam	Fauz	Instalasi Rawat Inap Kelas II	Januari	2007	
1 Malam	Siti Chusainih	Instalasi Rawat Inap Kelas III	Januari	2007	
1 Malam	Setyowati	Instalasi Rawat Inap Kelas III	Januari	2007	
2 Malam	Erwin Wickacrono	Instalasi Bedah Sentral	Januari	2007	
2 Malam	Supriyanto	Instalasi Bedah Sentral	Januari	2007	
2 Malam	Widigdo Reksio N.	Instalasi Bedah Sentral	Januari	2007	
2 Malam	Defi Andiyanto	Instalasi Gawat Darurat	Januari	2007	
2 Malam	Siti Istirokiah	Instalasi Gawat Darurat	Januari	2007	
2 Malam	Titik Ernawati	Instalasi Gawat Darurat	Januari	2007	
2 Malam	Shyntia Ekaristi	Instalasi Kamar Bersalin	Januari	2007	
2 Malam	Nuning Dwi Wati	Instalasi Kamar Bersalin	Januari	2007	
2 Malam	Tri Ambar Setyowati	Instalasi Kamar Bersalin	Januari	2007	
2 Malam	Achmad Zamroni	Instalasi Rawat Inap Kelas I	Januari	2007	
2 Malam	Fauz	Instalasi Rawat Inap Kelas I	Januari	2007	
2 Malam	Abdul Rauf	Instalasi Rawat Inap Kelas I	Januari	2007	

Gambar 4.20 Pencarian Berdasarkan Shift

## 4.2.7 Pencarian Berdasarkan Shift dan Ruang

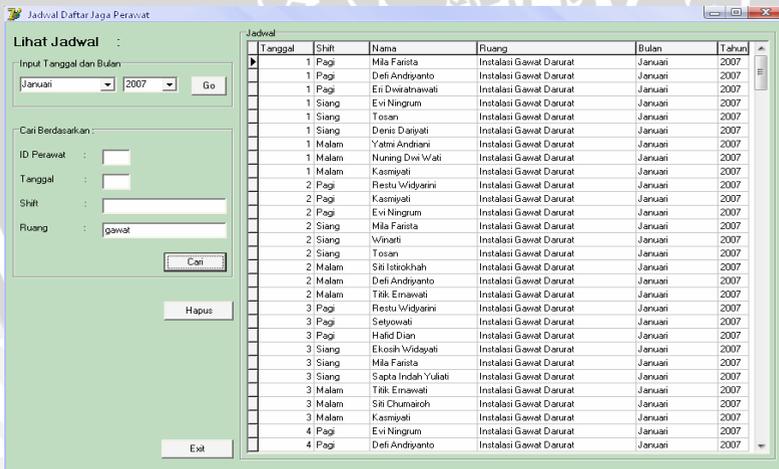
Pada pencarian ini, jadwal yang ditampilkan akan diurutkan berdasarkan nama shift yang diinputkan oleh *user*.



Gambar 4.21 Pencarian Berdasarkan Shift dan Ruang

#### 4.2.8 Pencarian Berdasarkan Ruang

Pada pencarian ini, *user* dapat menampilkan jadwal sesuai dengan nama ruangan yang akan dilihat jadwalnya.



Gambar 4.22 Pencarian Berdasarkan Ruang

### 4.3 Analisa Hasil

Pada penelitian ini akan dilakukan penelitian pengaruh beberapa parameter algoritma genetika terhadap nilai fitness. Parameter-parameter yang diujikan adalah Jumlah Generasi, Probabilitas Perkawinan Silang dan Probabilitas Mutasi. Hasil lengkap dari penelitian ini dapat dilihat pada lampiran halaman 61.

#### 4.3.1 Pengaruh Jumlah Generasi Terhadap Nilai *Fitness*

Pada pengujian pertama (I) ini akan diamati pengaruh parameter generasi dengan nilai fitness pada jumlah individu tertentu. Nilai fitness yang diamati adalah pada jumlah individu 50. Untuk nilai parameter ukuran probabilitas mutasi dan probabilitas kawin silang dipilih dari hasil uji coba yang menghasilkan nilai fitness tertinggi, sedangkan banyaknya generasi akan diset dengan nilai antara 10 sampai 300. Berikut ini tabel yang nilai parameter-parameter yang diset tetap yang digunakan pada pengujian I.

Tabel 4.4 Tabel parameter pengujian I

Parameter	Nilai Parameter
Probabilitas Kawin Silang	0,6
Probabilitas Mutasi	0,4
Individu	50

Hasil dari pengujian I dapat dilihat pada tabel 4.7.

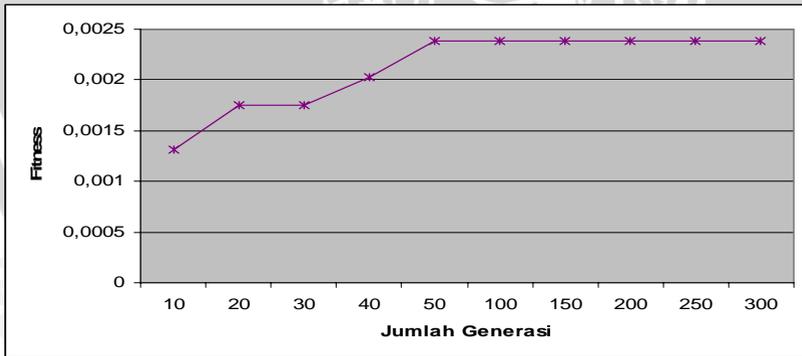
Tabel 4.5 Hasil pengujian I

Individu	Generasi	Prob Kawin Silang	Prob Mutasi	Nilai Fitness Maksimal
50	10	0,6	0,4	0,00131
50	20	0,6	0,4	0,00175
50	30	0,6	0,4	0,00175

50	40	0,6	0,4	0,00203
50	50	0,6	0,4	0,00238
50	100	0,6	0,4	0,00238
50	150	0,6	0,4	0,00238
50	200	0,6	0,4	0,00238
50	250	0,6	0,4	0,00238
50	300	0,6	0,4	0,00238

Pada Tabel 4.5 dapat diketahui nilai rata-rata *fitness* tertinggi dari setiap uji coba dengan nilai parameter generasi yang berbeda-beda. Tetapi pada generasi ke- 50 nilai *fitness* tidak mengalami perubahan hingga generasi ke-300. Pada uji coba dengan jumlah generasi 50 didapatkan hasil yang paling optimal dengan nilai rata-rata *cost* terendah adalah 419 dan rata-rata *fitness* tertinggi adalah 0,00238.

Pada gambar 4.23 dapat dilihat bahwa nilai *fitness* semakin meningkat dengan semakin bertambahnya jumlah generasi. Sehingga dapat ditarik kesimpulan bahwa parameter generasi berbanding lurus dengan nilai *fitness*. Semakin besar jumlah generasi maka *fitness* yang dihasilkan juga akan semakin besar tetapi pada kondisi tertentu nilai *fitness* akan mengalami konvergensi.



Gambar 4.23 Grafik Analisa Pengaruh Jumlah Generasi Terhadap Nilai Fitness

### 4.3.2 Pengaruh Nilai Probabilitas Perkawinan Silang Terhadap Nilai *Fitness*

Pada pengujian kedua (II) akan diamati pengaruh parameter kawin silang dengan nilai fitness pada jumlah individu tertentu. Nilai fitness yang diamati adalah pada jumlah individu 50. Untuk jumlah generasi dan nilai parameter ukuran probabilitas mutasi dipilih dari hasil uji coba yang menghasilkan nilai fitness tertinggi, sedangkan besarnya nilai probabilitas perkawinan silang akan diset dengan nilai antara 0,1 hingga 0,8. Berikut ini tabel yang nilai parameter-parameter yang diset tetap yang digunakan pada pengujian II.

Tabel 4.6 Tabel parameter pengujian II

Parameter	Nilai Parameter
Individu	50
Generasi	50
Probabilitas Mutasi	0,4

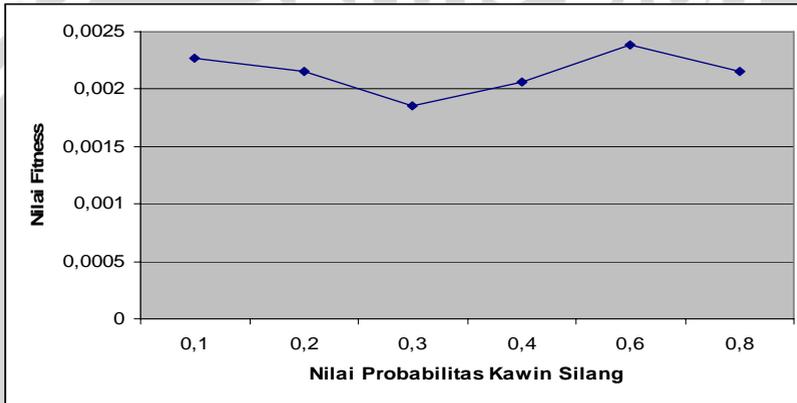
Hasil dari pengujian II adalah sebagai berikut :

Tabel 4.7 Hasil pengujian II

Individu	Generasi	Prob Kawin Silang	Prob Mutasi	Nilai Fitness Maksimal
50	50	0,1	0,4	0,00227
50	50	0,2	0,4	0,00216
50	50	0,3	0,4	0,00185
50	50	0,4	0,4	0,00206
50	50	0,6	0,4	0,00238
50	50	0,8	0,4	0,00216

Pada Tabel 4.7 dapat diketahui nilai rata-rata *fitness* tertinggi dari setiap uji coba dengan nilai parameter probabilitas perkawinan silang yang berbeda-beda. Pada uji coba dengan probabilitas perkawinan silang 0,6 didapatkan hasil yang paling optimal dengan nilai rata-rata *cost* terendah adalah 419 dan rata-rata *fitness* tertinggi adalah 0,00238.

Grafik pengaruh nilai probabilitas kawin silang dapat dilihat pada gambar 4.24



Gambar 4.24 Grafik Analisa Pengaruh Nilai Probabilitas Crossover Terhadap Nilai Fitness

### 4.3.3 Pengaruh Nilai Probabilitas Mutasi Terhadap Nilai *Fitness*

Pada pengujian ketiga (III) akan diamati pengaruh parameter mutasi dengan nilai *fitness* pada jumlah individu tertentu. Nilai *fitness* yang diamati adalah pada jumlah individu 50. Untuk jumlah generasi dan nilai parameter ukuran probabilitas kawin silang dipilih dari hasil uji coba yang menghasilkan nilai *fitness* tertinggi, sedangkan besarnya nilai probabilitas mutasi akan diset dengan nilai antara 0,1 hingga 0,8. Berikut ini tabel yang nilai parameter-parameter yang diset tetap yang digunakan pada pengujian III.

Tabel 4.8 Tabel parameter pengujian III

Parameter	Nilai Parameter
Individu	50
Generasi	50
Probabilitas Mutasi	0,4

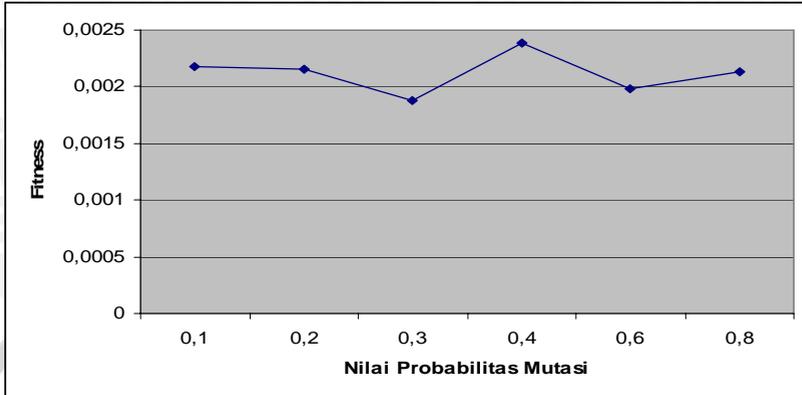
Hasil dari pengujian III adalah sebagai berikut :

Tabel 4.9 Hasil pengujian III

Individu	Generasi	Prob Kawin Silang	Prob Mutasi	Nilai Fitness Maksimal
50	50	0,6	0,1	0,00218
50	50	0,6	0,2	0,00215
50	50	0,6	0,3	0,00188
50	50	0,6	0,4	0,00238
50	50	0,6	0,6	0,00198
50	50	0,6	0,8	0,00213

Pada Tabel 4.9 dapat diketahui nilai rata-rata *fitness* tertinggi dari setiap uji coba dengan nilai parameter probabilitas mutasi yang berbeda-beda. Pada uji coba dengan nilai probabilitas mutasi 0,4 didapatkan hasil yang paling optimal dengan nilai rata-rata *cost* terendah adalah 419 dan rata-rata *fitness* tertinggi adalah 0,00238.

Hasil pengujian ke empat dapat juga dilihat pada gambar 4.25.



Gambar 4.25 Grafik Analisa Pengaruh Nilai Probabilitas Mutasi Terhadap Nilai Fitness



## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Kesimpulan yang diperoleh pada tugas akhir ini adalah :

1. Pada tugas akhir ini telah dibuat model genetika untuk masalah penjadwalan Daftar Jaga Perawat.
2. Algoritma genetika dapat digunakan sebagai alternatif solusi untuk menyelesaikan masalah penjadwalan Daftar Jaga Perawat. Jadwal Daftar jaga perawat diperoleh dari kromosom yang memiliki nilai *fitness* terbaik.
3. Semakin banyak jumlah generasi yang diberikan, maka nilai *fitness* akan semakin besar. Tetapi pada generasi ke 50, terjadi konvergensi sehingga nilai *fitness* tidak mengalami perubahan.
4. Perubahan nilai *fitness* dari inialisasi sampai menjadi *fitness* terbaik dikarenakan adanya parameter dasar yang penting yaitu probabilitas *crossover* dan mutasi. Probabilitas *crossover* menyatakan seberapa sering proses *crossover* akan terjadi diantara dua kromosom *parent*.. Pada penelitian ini nilai probabilitas *crossover* terbaik yang diperoleh adalah 0,6. Sedangkan probabilitas mutasi menyatakan seberapa sering bagian-bagian kromosom akan dimutasikan. Pada penelitian ini nilai probabilitas mutasi terbaik yang diperoleh adalah 0,4.

### 5.2 Saran

Aplikasi yang dibangun belum sempurna, berikut terdapat beberapa hal yang diharapkan dapat bermanfaat untuk pengembangan aplikasi ini:

1. Pada aplikasi yang dibangun masih dibutuhkan waktu yang lama untuk menyelesaikan lebih dari 500 iterasi, padahal seharusnya sebuah algoritma genetika harus melakukan iterasi yang banyak agar mencapai hasil yang lebih optimal.
2. Aplikasi penjadwalan daftar jaga perawat ini dapat dikembangkan menjadi sebuah sistem informasi yang

memiliki kemampuan membuat jadwal daftar jaga sekaligus sebagai sistem absen di sebuah rumah sakit.

**UNIVERSITAS BRAWIJAYA**



## DAFTAR PUSTAKA

- [1] Darmoni, S.J., Fajner, A., Mahé, N., Leforestier, A., Vondracek, M., Stelian, O., dan Baldenweck, M. *Computer-assisted Nurse Scheduling Using Constraint-based Programming*.  
<http://www.clinchem.org/cgi/reprint/47/1/118.pdf>.  
Tanggal akses : 5 April 2007
- [2] Gen, Mitsuo and Runwei, Cheng. 1997. *Genetic Algorithms And Engineering Design*. New York : John Wiley & Sons, Inc.
- [3] Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta : Graha Ilmu.
- [4] Makoto, Koshimizu and Makoto, Arai. *A Solution Of Nurse Scheduling Problem by Genetic Algorithm*.  
<http://www.kushiro-ct.ac.jp/library/kiyo/kiyo36/kosimizu.pdf>.  
Tanggal akses : 5 April 2007
- [5] Michalewicz , Zbigniew. 1999 . *Genetic Algorithms + Data Structures = Evolution Programs*. New York : Springer.
- [6] Setiawan, Kuswara. 2003. *Paradigma Sistem Cerdas (jaringan saraf tiruan, logika fazi dan algoritma genetik)*. Malang: Bayumedia publishing.
- [7] Sukmawan, Budi, 2003. *Sekilas Tentang Algoritma Genetika dan Aplikasinya pada Optimasi Jaringan Pipa Air Bersih*.  
<http://www.petra.ac.id/~puslit/journals/request.php?PublishedID=IND04060106>.  
Tanggal akses : 17 April 2007
- [8] Syamsudin, Aries, 2004, *Pengenalan Algoritma Genetik*,  
<http://www.ilmukomputer.com>.  
Tanggal akses : 17 April 2007
- [9] Wikipedia. *Scheduling*.  
<http://en.wikipedia.org/wiki/scheduling>.  
Tanggal akses: 9 April 2007