

**OPTIMASI PENJADWALAN MATA PELAJARAN  
MENGUNAKAN ALGORITMA GENETIKA  
( STUDI KASUS: SMPN 1 GONDANG MOJOKERTO )**

**SKRIPSI**

**KONSENTRASI KOMPUTASI CERDAS DAN VISUAL**

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

**DIANITA DWI PERMATA SARI**

**115060801111007**

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI**

**UNIVERSITAS BRAWIJAYA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER**

**MALANG**

**2015**

## LEMBAR PERSETUJUAN

### OPTIMASI PENJADWALAN MATA PELAJARAN MENGUNAKAN ALGORITMA GENETIKA ( STUDI KASUS: SMPN 1 GONDANG MOJOKERTO )

#### SKRIPSI

#### KONSENTRASI KOMPUTASI CERDAS DAN VISUAL

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh:

**DIANITA DWI PERMATA SARI**

**NIM. 115060801111007**

Skrripsi ini telah disetujui oleh dosen pembimbing pada tanggal 5 Mei 2015

**Dosen Pembimbing I,**

**Dosen Pembimbing II,**

**Wayan Firdaus Mahmudy, S.Si., MT, Ph.D**

**Dian Eka Ratnawati, S.Si., M.Kom**

**NIP. 19720919 199702 1 001**

**NIP. 19730619 200212 2 001**

## LEMBAR PENGESAHAN

### OPTIMASI PENJADWALAN MATA PELAJARAN MENGUNAKAN ALGORITMA GENETIKA ( STUDI KASUS: SMPN 1 GONDANG MOJOKERTO )

#### SKRIPSI

#### KONSENTRASI KOMPUTASI CERDAS DAN VISUAL

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

**DIANITA DWI PERMATA SARI**

**NIM. 115060801111007**

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 22 Mei 2015

**Penguji I**

**Penguji II**

**Drs. Marji, M.T.**

**NIP. 19670801 199203 1 001**

**Muhammad Tanzil Furgon, S.Kom., M.CompSc**

**NIP. 19820930 200801 1 004**

**Penguji III**

**Indriati, S.T., M.Kom**

**NIK. 83101306120035**

**Mengetahui**

**Ketua Program Studi Informatika/Illmu Komputer**

**Drs. Marji, M.T.**

**NIP. 19670801 199203 1 001**

## PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 28 Mei 2015

Mahasiswa,

**Dianita Dwi Permata Sari**

**115060801111007**

## KATA PENGANTAR

Syukur dan alhamdulillah penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul: **”OPTIMASI PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA GENETIKA (STUDI KASUS: SMPN 1 GONDANG MOJOKERTO)”**.

Skripsi ini diajukan sebagai syarat ujian skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi Dan Ilmu Komputer (PTI IK), Program Studi Informatika/Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, penulis mengucapkan terima kasih kepada:

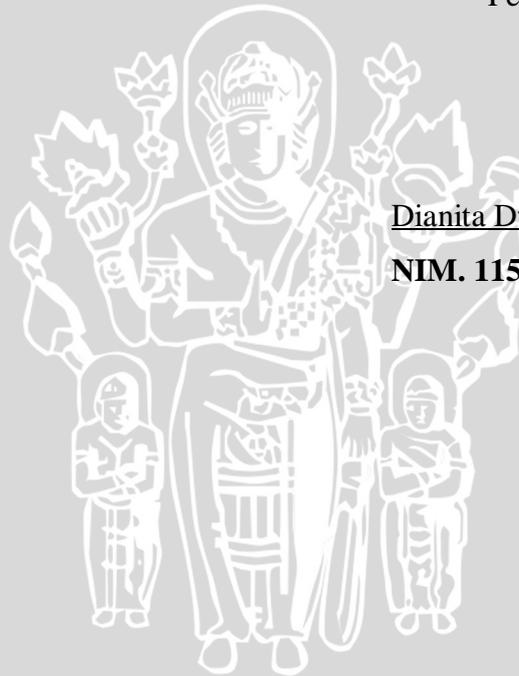
1. Bapak Wayan Firdaus Mahmudy, S.Si, MT, Ph.D, selaku Dosen Pembimbing Skripsi pertama yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
2. Ibu Dian Eka Ratnawati, S.Si., M.Kom, selaku Dosen Pembimbing Skripsi kedua yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
3. Bapak Ir. Sutrisno, MT., selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Drs. Marji, MT, selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya Malang.
5. Orang tua dan saudara Penulis yang telah selalu mendukung dan mendoakan sehingga terselesaikannya skripsi ini.
6. Sahabat Penulis serta teman – teman Teknik Informatika angkatan 2011 yang memotivasi dan saling menyemangati.
7. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.

8. Segenap staff dan karyawan di Program Studi Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu dalam hal administrasi Penulis dalam pelaksanaan penyusunan skripsi ini.
9. Penulis menyadari bahwa skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat Penulis harapkan dari berbagai pihak demi penyempurnaan penulisan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, 28 Mei 2015

Penulis,



Dianita Dwi Permata Sari

**NIM. 115060801111007**

## ABSTRAK

Penjadwalan mata pelajaran merupakan hal yang sangat kompleks dan bagian yang sangat penting dalam suatu sekolah karena dengan penyusunan jadwal maka proses belajar mengajar dapat berjalan dengan baik. Proses penyusunan jadwal ini dilakukan di SMPN 1 Gondang yang masih melakukan penjadwalan mata pelajaran secara manual dan sering mengalami kendala karena proses penyusunan tersebut membutuhkan waktu lama dan hasil yang tidak akurat. Sebagai contoh permasalahan yang kerap terjadi adalah kesulitan untuk menempatkan jadwal supaya tidak terjadi tabrakan, efisien waktu dengan jumlah guru yang terbatas dan kemungkinan jumlah jam mengajar guru yang berlebih. Salah satu metode optimasi untuk permasalahan ini adalah melalui pendekatan algoritma genetika. Pada penelitian ini menggunakan representasi permutasi berbasis kode guru. Dengan menerapkan algoritma genetika diharapkan dapat menghasilkan solusi penjadwalan yang optimal. Metode *crossover* yang digunakan yaitu *one cut point* dan metode mutasinya *reciprocal exchange mutation* serta seleksinya menggunakan *elitism*. Dari hasil pengujian didapatkan hasil terbaik dengan nilai *fitness* tertinggi pada jumlah generasi 80, jumlah populasi 60 dan kombinasi *crossover rate* dan *mutation rate* yaitu 0.4:0.6. Hasil akhir yaitu kromosom terbaik yang menghasilkan penjadwalan dengan nilai *fitness* tertinggi.

Kata Kunci : Algoritma genetika, penjadwalan mata pelajaran

## ABSTRACT

*Scheduling subject is very complex and very important part in a school because by preparing the schedule, learning process can run well. This scheduling process is done in SMPNI Gondang which still do it manually. But it often faces problems because the drafting process takes a long time and the results are inaccurate. As an example of the problems that often occur is difficult to put a schedule without getting any collision, efficient time with a limited number of teachers, and the possibility of excess teaching hours. One method of optimization for this problem is through the genetic algorithm. In this research using a permutation representation based teacher code. By applying genetic algorithm, it is expected to produce optimal scheduling solutions. Crossover method used is the one cut point and method of reciprocal exchange mutation of the mutation itself, and for the selection is using elitism. From the tests' results obtained the best results with the highest fitness value generation number 80, number of population is 60, and combined population crossover rate and mutation rate is 0.4: 0.6. The end of the result is the best chromosome that produces optimal scheduling with the highest fitness value.*

*Kata Kunci : Genetic algorithm, scheduling subject*

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	5
1.5 Manfaat.....	5
1.6 Sistematika Penulisan.....	5
<b>BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....</b>	<b>7</b>
2.1 Kajian Pustaka.....	7
2.2 Penjadwalan Sekolah.....	10
2.3 Algoritma Genetika.....	13
2.4 Struktur Umum Algoritma Genetika.....	14
2.5 Komponen – Komponen Utama Algoritma Genetika.....	17
<b>2.5.1 Generasi Awal.....</b>	<b>17</b>
<b>2.5.2 Representasi Kromosom.....</b>	<b>17</b>
<b>2.5.3 Fungsi <i>Fitness</i>.....</b>	<b>18</b>
<b>2.5.4 Seleksi.....</b>	<b>19</b>
2.6 Operator Genetika.....	20
<b>2.6.1 Persilangan (<i>Crossover</i>).....</b>	<b>20</b>
<b>2.6.2 Mutasi (<i>Mutation</i>).....</b>	<b>21</b>
2.7 Parameter Genetika.....	22

2.8	Kondisi Berhenti ( <i>Termination Condition</i> ).....	23
BAB III METODOLOGI PENELITIAN .....		24
3.1	Tahapan Penelitian.....	24
3.1.1	<b>Studi Literatur</b> .....	24
3.1.2	<b>Metode Pengambilan Data</b> .....	25
3.1.3	<b>Analisis Kebutuhan</b> .....	25
3.1.4	<b>Perancangan</b> .....	26
3.1.5	<b>Implementasi</b> .....	26
3.1.6	<b>Uji Coba</b> .....	26
3.1.7	<b>Evaluasi</b> .....	26
3.2	Kebutuhan Sistem.....	27
3.2.1	<b>Deskripsi Umum Sistem</b> .....	27
3.2.2	<b>Batasan Implementasi</b> .....	28
3.2.3	<b>Spesifikasi Kebutuhan Perangkat</b> .....	29
3.3	Formulasi Permasalahan .....	29
3.4	Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika.....	33
3.4.1	<b>Representasi Kromosom dan Perhitungan <i>Fitness</i></b> .....	37
3.4.2	<b>Inisialisasi Populasi Awal</b> .....	42
3.4.3	<b>Reproduksi</b> .....	44
3.4.4	<b>Evaluasi dan Seleksi</b> .....	48
BAB IV PERANCANGAN .....		51
4.1	Perancangan Database .....	51
4.1.1	<b>Tabel Guru</b> .....	51
4.1.2	<b>Tabel Mapel</b> .....	51
4.1.3	<b>Tabel Penugasan</b> .....	51
4.1.4	<b>Perancangan Database Keseluruhan</b> .....	52
4.2	Perancangan <i>User Interface</i> .....	52
4.2.1	<b>Rancangan <i>User Interface</i> Halaman Data Guru</b> .....	53
4.2.2	<b>Rancangan <i>User Interface</i> Halaman Data Mapel</b> .....	54
4.2.3	<b>Rancangan <i>User Interface</i> Halaman Penugasan</b> .....	55
4.2.4	<b>Rancangan <i>User Interface</i> Halaman Parameter</b> .....	56
4.3	Perancangan Uji Coba dan Evaluasi.....	57
4.3.1	<b>Uji Coba Banyaknya Generasi</b> .....	57
4.3.2	<b>Uji Coba Jumlah Populasi</b> .....	58

4.3.3	Uji Coba Kombinasi Probabilitas <i>Crossover</i> dan Mutasi.....	59
BAB V	IMPLEMENTASI.....	61
5.1	Implementasi Basis Data.....	61
5.1.1	Tabel Guru.....	61
5.1.2	Tabel Mapel.....	61
5.1.3	Tabel Penugasan.....	62
5.2	Implementasi Algoritma.....	62
5.2.1	Proses Penugasan.....	62
5.2.2	Inisialisasi Populasi Awal.....	65
5.2.3	Perhitungan Fitness.....	66
5.2.4	Proses <i>Crossover</i> .....	67
5.2.5	Proses Mutasi.....	68
5.2.6	Proses Seleksi <i>Elitism</i> .....	69
5.2.7	Proses Pemilihan Kromosom Terbaik.....	70
5.3	Implementasi <i>User Interface</i> .....	70
5.3.1	Halaman Data Guru.....	70
5.3.2	Halaman Data Mata Pelajaran.....	71
5.3.3	Halaman Data Penugasan.....	72
5.3.4	Halaman Proses Genetika.....	73
5.3.5	Halaman Hasil Penjadwalan.....	74
BAB VI	PENGUJIAN DAN ANALISIS.....	76
6.1	Contoh Hasil Penjadwalan Data Sebagian.....	76
6.2	Sistematika Pengujian.....	80
6.3	Analisa Hasil dan Pembahasan.....	80
6.3.1	Hasil Pengujian Terhadap Banyaknya Generasi.....	80
6.3.2	Hasil Pengujian Terhadap Jumlah Populasi.....	82
6.3.3	Hasil Pengujian Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i> .....	84
6.3.4	Hasil Penjadwalan.....	87
BAB VII	PENUTUP.....	91
7.1	Kesimpulan.....	91
7.2	Saran.....	92
DAFTAR PUSTAKA	.....	93

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Pseudocode siklus algoritma genetika.....	15
<b>Gambar 2.2</b> Contoh kromosom dengan pengkodean permutasi .....	17
<b>Gambar 2.3</b> <i>Pseudocode</i> seleksi <i>elitism</i> .....	19
<b>Gambar 2.4</b> Contoh <i>one cut point crossover</i> pada pengkodean permutasi .....	20
<b>Gambar 3.1</b> Diagram Alir Metodologi Penelitian.....	24
<b>Gambar 3.2</b> Flowchart Proses Algoritma Genetika .....	34
<b>Gambar 3.3</b> Algoritma Evaluasi <i>Fitness</i> .....	40
<b>Gambar 3.4</b> Algoritma Inisialisasi Populasi Awal.....	42
<b>Gambar 3.5</b> Algoritma <i>Crossover</i> .....	45
<b>Gambar 3.6</b> Algoritma Mutasi .....	47
<b>Gambar 3.7</b> Algoritma Metode Seleksi <i>Elitism</i> .....	49
<b>Gambar 4.1</b> Tabel Guru .....	51
<b>Gambar 4.2</b> Tabel Mapel .....	51
<b>Gambar 4.3</b> Tabel Penugasan .....	52
<b>Gambar 4.4</b> Relasi Antar Tabel Database.....	52
<b>Gambar 4.5</b> Rancangan <i>User Interface</i> Halaman Data Guru .....	53
<b>Gambar 4.6</b> Rancangan <i>User Interface</i> Halaman Data Mapel .....	54
<b>Gambar 4.7</b> Rancangan <i>User Interface</i> Halaman Penugasan .....	55
<b>Gambar 4.8</b> Rancangan <i>User Interface</i> Halaman Parameter.....	56
<b>Gambar 5.1</b> <i>Source Code</i> Proses Penugasan .....	65
<b>Gambar 5.2</b> <i>Source Code</i> Inisialisasi Populasi Awal.....	66
<b>Gambar 5.3</b> <i>Source Code</i> Perhitungan <i>Fitness</i> .....	66
<b>Gambar 5.4</b> <i>Source Code</i> Proses <i>Crossover</i> .....	68
<b>Gambar 5.5</b> <i>Source Code</i> Proses Mutasi .....	69
<b>Gambar 5.6</b> <i>Source Code</i> Proses Seleksi <i>Elitism</i> .....	69
<b>Gambar 5.7</b> <i>Source Code</i> Proses Pemilihan Kromosom Terbaik .....	70
<b>Gambar 5.8</b> Implementasi <i>User Interface</i> Halaman Data Guru .....	71
<b>Gambar 5.9</b> Implementasi <i>User Interface</i> Halaman Data Mapel.....	72
<b>Gambar 5.10</b> Implementasi <i>User Interface</i> Halaman Data .....	73
<b>Gambar 5.11</b> Implementasi <i>User Interface</i> Halaman Proses Genetika.....	74
<b>Gambar 5.12</b> Implementasi <i>User Interface</i> Halaman Hasil Penjadwalan .....	75
<b>Gambar 6.1</b> Implementasi <i>User Interface</i> .....	77
<b>Gambar 6.2</b> Implementasi <i>User Interface</i> Hasil Proses Genetika Pada Data Sebagian .....	78
<b>Gambar 6.3</b> Implementasi <i>User Interface</i> Hasil Penjadwalan Pada Data Sebagian .....	79
<b>Gambar 6.4</b> Implementasi <i>User Interface</i> Hasil Penjadwalan Optimal Pada Data Sebagian .....	79
<b>Gambar 6.5</b> Grafik Hasil Pengujian Pengaruh Banyaknya Generasi.....	82
<b>Gambar 6.6</b> Grafik Hasil Pengujian Pengaruh Jumlah Populasi .....	83
<b>Gambar 6.7</b> Grafik Hasil Pengujian Kombinasi Cr : Mr.....	85

Gambar 6.8 Hasil Penjadwalan Mata Pelajaran..... 89



## DAFTAR TABEL

<b>Tabel 2.1</b> Istilah Ilmu genetik dan Algoritma genetik .....	15
<b>Tabel 3.1</b> Representasi Algoritma Genetika Dalam Penjadwalan Mata Pelajaran .....	27
<b>Tabel 3.2</b> Daftar Mata Pelajaran dan Jumlah Porsi Jam Pelajaran .....	29
<b>Tabel 3.3</b> Daftar Guru Pengajar dan Mata Pelajaran .....	30
<b>Tabel 3.4</b> Porsi Waktu Pengajaran .....	32
<b>Tabel 3.5</b> Jam Pengajaran .....	32
<b>Tabel 3.6</b> Data Guru Pengajar .....	35
<b>Tabel 3.7</b> Penugasan Guru .....	36
<b>Tabel 3.8</b> Komponen Kromosom .....	37
<b>Tabel 3.9</b> Representasi Kromosom .....	38
<b>Tabel 3.10</b> Penyusunan Jadwal Hasil Representasi Kromosom .....	38
<b>Tabel 3.11</b> Contoh Evaluasi <i>Fitness</i> .....	40
<b>Tabel 3.12</b> Inisialisasi Populasi Awal .....	43
<b>Tabel 3.13</b> Proses <i>Crossover</i> .....	46
<b>Tabel 3.14</b> Proses Mutasi .....	48
<b>Tabel 3.15</b> Contoh Nilai <i>Fitness</i> Semua Kromosom .....	49
<b>Tabel 3.16</b> Hasil dari Proses Seleksi <i>Elitism</i> .....	50
<b>Tabel 3.17</b> Hasil Solusi Jadwal Individu Terbaik .....	50
<b>Tabel 4.1</b> Rancangan Uji Coba Banyaknya Generasi .....	58
<b>Tabel 4.2</b> Rancangan Uji Coba Jumlah Populasi .....	59
<b>Tabel 4.3</b> Rancangan Uji Coba Terhadap Probabilitas <i>Crossover</i> Dan Mutasi ...	60
<b>Tabel 5.1</b> Tipe data pada Tabel Guru .....	61
<b>Tabel 5.2</b> Tipe data pada Tabel Mapel .....	61
<b>Tabel 5.3</b> Tipe data pada Tabel Penugasan .....	62
<b>Tabel 6.1</b> Hasil Pengujian Banyaknya Generasi ( $cr = 0,5$ ; $mr = 0,5$ ) .....	81
<b>Tabel 6.2</b> Hasil Pengujian Jumlah Populasi ( $cr = 0,5$ ; $mr = 0,5$ ) .....	83
<b>Tabel 6.3</b> Hasil Pengujian Kombinasi <i>Crossover Rate</i> dan <i>Mutastion Rate</i> .....	85

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Masalah penjadwalan dalam institusi pendidikan, masih menjadi isu yang menarik dan secara luas masih diteliti oleh banyak peneliti di dunia. Masalah penjadwalan dapat diklasifikasikan ke dalam dua kategori utama, yaitu: masalah penjadwalan kuliah dan masalah penjadwalan ujian (Gunawan, 2004). Klasifikasi dari masalah penjadwalan ini juga dapat dibagi ke dalam tiga klasifikasi utama, yaitu: penjadwalan sekolah, penjadwalan kuliah dan penjadwalan ujian (Schaerf dalam S'andor 2005). Untuk penjadwalan di sekolah yaitu penyusunan mata pelajaran pada umumnya dilakukan secara manual dengan menggunakan tabel, namun proses penjadwalan dengan cara tersebut merupakan hal yang cukup merepotkan dan rentan terhadap human error serta membutuhkan waktu cukup lama.

Penjadwalan kegiatan belajar mengajar merupakan hal yang sangat kompleks dan bagian yang sangat penting dalam suatu sekolah dikarenakan penyusunan jadwal harus ada supaya proses belajar mengajar dapat berjalan dengan baik sehingga siswa ataupun guru dapat menikmati kegiatan belajar mengajar tersebut. Permasalahan penyusunan jadwal sering dihadapi di sekolah menengah pertama khususnya SMPN 1 Gondang. Tujuan penyusunan jadwal sendiri adalah untuk menempatkan mata pelajaran sesuai dengan klasifikasinya dan memenuhi ketersediaan guru dalam mengajar serta memenuhi kendala lainnya. Pada tahun sebelumnya khususnya di SMPN 1 Gondang dalam menyusun jadwal mata pelajaran masih menggunakan software Ms.Excel sehingga sering mengalami kendala dikarenakan proses penyusunan tersebut membutuhkan waktu lama dan hasil yang tidak akurat. Sebagai contoh permasalahan yang kerap terjadi adalah kesulitan untuk menempatkan jadwal supaya tidak terjadi tabrakan, efisien waktu dengan jumlah guru yang terbatas dan kemungkinan jumlah jam mengajar guru yang berlebih.

Dalam proses penyusunan jadwal di SMPN 1 Gondang masih dilakukan secara manual yaitu dengan memadukan data – data yang berasal dari data guru, data mata pelajaran yang akan diajarkan, data ketersediaan kelas dan waktu pengajaran yang tepat. Penyusunan jadwal manual seperti itu sangat tidak mudah dan diperlukan pengaturan yang cukup rumit. Alokasi dan penentuan guru merupakan salah satu elemen penting dalam menyusun jadwal pelajaran di suatu sekolah. Untuk menyusun jadwal belajar mengajar yang baik dan optimal maka harus memperhatikan beberapa aspek diantaranya yaitu dari aspek guru dimana pembuat jadwal harus mempertimbangkan frekuensi mengajar guru dalam sehari dan lamanya waktu yang diperlukan guru dalam sekali mengajar serta ketersediaan jumlah guru yang menguasai masing – masing bidang. Dan juga dari aspek mata pelajaran yang diajarkan, maka pembuat jadwal harus mengatur jadwal mata pelajaran yang baik berdasarkan waktu pengajaran yang tepat agar murid dapat menjalankan kegiatan belajar mengajar dengan optimal. Sebagai contoh, mata pelajaran olahraga atau sering dikenal dengan Penjaskes harus dilakukan pada jam pertama dimulainya proses belajar supaya murid dapat melakukan aktifitas tersebut pada pagi hari dan manfaat yang didapat lebih maksimal. Di samping aspek – aspek di atas, dalam penyusunan jadwal belajar mengajar ini pun terdapat sangat banyak kemungkinan yang selayaknya dicoba untuk menemukan penjadwalan yang terbaik. Oleh karena itu dibutuhkan adanya optimasi untuk merancang sistem penjadwalan dengan meminimalisir error jadwal sehingga kegiatan belajar dapat terlaksana dengan optimal. Berdasarkan hasil observasi pada sekolah tersebut terdapat beberapa komponen yang harus dijadwalkan yaitu mata pelajaran, ketersediaan guru, ruang kelas dan waktu pengajaran.

Salah satu metode optimasi untuk permasalahan ini adalah melalui pendekatan Algoritma Genetika. Algoritma Genetika telah banyak diaplikasikan untuk penyelesaian masalah dan permodelan dalam bidang teknologi, bisnis, dan entertainment, seperti optimasi penjadwalan, pemrograman otomatis, machine learning, model ekonomi, model sistem imunisasi, model ekologis, interaksi antara evolusi dan belajar (Sam'ani, 2012). Penjadwalan bersifat unik terhadap kasus yang diselesaikan. Skripsi ini akan memfokuskan pada penyusunan

representasi kromosom yang paling efisien. Algoritma genetika dapat menjadi solusi masalah optimasi yang model matematikanya kompleks atau bahkan sulit dibangun (Mahmudy, 2013). Diharapkan dengan digunakannya algoritma genetika akan diperoleh optimasi penjadwalan yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan mata pelajaran dan guru secara keseluruhan, tidak ada permasalahan bentrokan jadwal serta ketersediaan kelas yang sesuai secara fasilitas untuk seluruh mata pelajaran yang ada.

Penggunaan algoritma genetika untuk menyelesaikan masalah optimasi dan penjadwalan telah dibahas dalam beberapa penelitian sebelumnya. Misalnya Salimi (2006) yang menerapkan metode algoritma genetika pada sistem informasi penjadwalan mata pelajaran pada sekolah menengah umum dengan menggunakan salah satu dari metode algoritma genetika yaitu *breeder genetic algorithm* dimana metode ini hanya memilih individu terbaik untuk diseleksi dan dimasukkan ke *gen pool* untuk direkombinasi dan dimutasi. Penetapan kriteria pembatas yang digunakan dalam penyelesaian masalah penjadwalan mata pelajaran ini yaitu mata pelajaran yang mempunyai prioritas sama dalam satu hari tidak boleh dijadwalkan berurutan dan dalam satu minggu suatu mata pelajaran tidak boleh dijadwalkan melebihi bobot yang telah ditetapkan misal mata pelajaran agama mempunyai bobot 2 jam pelajaran jadi mata pelajaran agama tidak boleh dijadwalkan lebih dari 2 kali serta mata pelajaran untuk kelas XI dan XII sudah diterapkan sistem penjurusan sehingga ada perbedaan dalam mendapatkan pengajaran mata pelajaran berdasarkan jurusan yang diminati. Penelitian yang sejenis dilakukan oleh Nugraha (2008) pada obyek yang berbeda yaitu masalah penjadwalan pada perkuliahan. Kedua penelitian menjelaskan bahwa algoritma genetika sebagai algoritma yang baik dalam melakukan optimasi penyusunan penjadwalan dan menghasilkan optimasi penjadwalan yang optimal serta diperoleh kondisi kombinasi terbaik.

Berdasarkan latar belakang yang telah dipaparkan di atas berkaitan dengan penjadwalan mata pelajaran dan penggunaan metode algoritma genetika, maka penulis mengambil topik skripsi dengan judul **“Optimasi Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika (Studi Kasus : SMPN 1 Gondang Mojokerto)”**. Perbedaan dengan penelitian sebelumnya adalah pada

penetapan kriteria pembatas yang nantinya akan mempengaruhi representasi kromosom dari penjadwalan mata pelajaran. Kriteria pembatasnya adalah beban belajar dalam 1 minggu dibutuhkan 40 jam pembelajaran dengan durasi 40 menit, standar pengajaran bagi guru yang telah bersertifikasi minimal 24 jam per minggu sedangkan standarisasi tersebut tidak berlaku bagi guru tidak tetap sehingga guru tidak tetap hanya memenuhi jam kerja sesuai kebutuhan sekolah.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan yang akan dijadikan objek penelitian untuk skripsi ini, yaitu:

1. Bagaimana menerapkan algoritma genetika untuk mengoptimalkan jadwal mata pelajaran di SMPN 1 Gondang.
2. Bagaimana bentuk kromosom dan *fitness* yang dapat digunakan untuk menyelesaikan masalah optimasi penjadwalan mata pelajaran.
3. Bagaimana menentukan parameter algoritma genetika yang tepat untuk mendapatkan hasil penjadwalan yang optimal pada masalah optimasi penjadwalan mata pelajaran.
4. Bagaimana kualitas solusi yang dihasilkan oleh algoritma genetika terhadap hasil dari optimasi penjadwalan mata pelajaran.

### 1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah dijabarkan, penelitian ini mempunyai batasan – batasan masalah sebagai berikut:

1. Objek data yang digunakan untuk penelitian ini diperoleh dari jadwal kegiatan belajar mengajar yang dilakukan di SMPN 1 Gondang Mojokerto.
2. Parameter – parameter Algoritma Genetika yang digunakan adalah jumlah individu, probabilitas *crossover*, dan probabilitas *mutation*.
3. Parameter – parameter dalam penjadwalan yang digunakan adalah mata pelajaran yang diajarkan, ketersediaan guru, jumlah ruang kelas dan waktu pengajaran.

#### 1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah :

1. Mengetahui penerapan algoritma genetika pada masalah optimasi penjadwalan mata pelajaran sehingga diperoleh kombinasi terbaik.
2. Menentukan bentuk kromosom dan *fitness* yang dapat digunakan untuk menyelesaikan masalah optimasi penjadwalan mata pelajaran.
3. Mengetahui parameter algoritma genetika yang tepat terhadap hasil dari optimasi penjadwalan mata pelajaran.
4. Mengetahui kualitas solusi penjadwalan terbaik yang dihasilkan algoritma genetika pada optimasi penjadwalan mata pelajaran.

#### 1.5 Manfaat

Manfaat yang diharapkan dengan adanya penulisan skripsi ini adalah:

1. Mendapatkan pemecahan dalam masalah optimasi penjadwalan mata pelajaran dengan beberapa parameter yang telah ada.
2. Meningkatkan efisiensi dan efektifitas dalam pembuatan jadwal mata pelajaran.
3. Meningkatkan pemahaman tentang penggunaan algoritma genetik dalam memperoleh optimasi penjadwalan sehingga diperoleh kombinasi terbaik untuk pasangan mata pelajaran dan guru secara keseluruhan, tidak ada permasalahan bentrok jam kegiatan belajar mengajar, waktu dan kelas.

#### 1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

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

Berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan skripsi.

##### **BAB II                KAJIAN PUSTAKA DAN DASAR TEORI**

Berisi kajian teori yang berhubungan dengan penelitian skripsi ini yaitu teori dasar tentang algoritma genetika dan referensi yang mendasari pembuatan aplikasi penjadwalan mata pelajaran serta teori-teori yang berhubungan dalam skripsi ini.

### **BAB III      METODOLOGI PENELITIAN**

Dalam bab metodologi ini akan dibahas tentang tahapan-tahapan yang dilakukan pada saat penelitian yang terdiri dari studi literatur, metode pengambilan data, analisis kebutuhan, perancangan, implementasi, pengujian dan evaluasi. Terdapat juga penjelasan mengenai kebutuhan sistem, formulasi permasalahan dan siklus penyelesaian masalah menggunakan algoritma genetika.

### **BAB IV      PERANCANGAN**

Pada tahap perancangan akan dibahas tentang rancangan yang akan dibuat untuk penjadwalan mata pelajaran seperti perancangan database, perancangan *user interface*, dan perancangan uji coba dan evaluasi.

### **BAB V      IMPLEMENTASI**

Berisi tentang penjelasan implementasi dari sistem, bagaimana *user interface* sistem dan *source code* untuk mengembangkan sistem.

### **BAB VI      PENGUJIAN DAN ANALISIS**

Berisi tentang penjelasan proses pengujian dan hasil pengujian serta analisis dari pengujian tersebut.

### **BAB VII      PENUTUP**

Berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran untuk pengembangan.

## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

#### 2.1 Kajian Pustaka

Berdasarkan judul skripsi yang dibahas, penulis menemukan beberapa hasil penelitian yang relevan untuk mendukung penelitian dalam skripsi ini, antara lain: Nugraha (ITB:2008), dalam tugas akhirnya yang berjudul Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar menjelaskan bahwa algoritma genetika sebagai algoritma yang baik dalam melakukan optimasi penyusunan penjadwalan mata kuliah dan menghasilkan optimasi penjadwalan yang optimal serta diperoleh kondisi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan, tidak ada permasalahan bentrokan jadwal pada sisi mahasiswa, ketersediaan ruang yang cukup serta sesuai secara fasilitas untuk seluruh mata kuliah yang ada. Penjadwalan mata kuliah ini merepresentasikan kromosomnya berdasarkan pada mata kuliah yang akan dijadwalkan, ketersediaan dosen, kelas maupun ruangan yang digunakan untuk perkuliahan, jumlah mahasiswa serta alokasi waktu perkuliahannya. Penetapan kriteria pembatas yang terkait dengan penjadwalan tersebut diantaranya tidak boleh adanya jadwal kuliah yang beririsan dengan jadwal kuliah angkatan sebelumnya maupun sesudahnya sehingga mahasiswa dapat mengambil mata kuliah angkatan sebelumnya maupun sesudahnya dan distribusi jadwal perkuliahan diharapkan dapat merata tiap harinya untuk setiap kelas. Dalam merepresentasikan kromosomnya juga dipengaruhi dengan adanya batasan dalam menyusun penjadwalan mata kuliah tersebut. Untuk pembuatan kromosom dan populasi berdasarkan urutan dari Tabel Prioritas Mata Kuliah, setiap mata kuliah akan dijadwalkan ke dalam Tabel Jadwal Mata Kuliah secara acak. Agar diketahui apakah pada waktu tersebut dosen, kelas, maupun ruangan dapat digunakan untuk melaksanakan perkuliahan, maka Tabel Dosen, Tabel Ruang, dan Tabel Kelas untuk setiap mata kuliah serta Tabel Mata Kuliah harus dipetakan terlebih dahulu dalam Tabel Jadwal Mata Kuliah. Sebelum menjadwalkan suatu mata kuliah pada Tabel Jadwal Mata Kuliah, algoritma akan

mengecek terlebih dahulu kepada Tabel Jadwal Mata Kuliah untuk mengetahui apakah pada waktu tersebut dapat digunakan untuk perkuliahan atau tidak. Jika tidak maka algoritma didesain untuk mencari alokasi waktu lainnya. Proses penjadwalan mata kuliah menggunakan algoritma genetika ini dapat diterapkan pada kasus-kasus penjadwalan dengan multi angkatan dan multi ruangan. Untuk mendapatkan solusi yang terbaik, maka program harus menyeleksi solusi yang memiliki nilai *fitness* yang tergolong rendah. Seleksi menggunakan metode *good fitness* dan hasil penelitian menunjukkan bahwa algoritma genetika yang menggunakan metode *best fitness* akan selalu menunjukkan kenaikan *fitness* atau dengan kata lain generasi selanjutnya lebih baik atau minimal sama dengan generasi sebelumnya.

Salimi (STIKOM:2006) dalam tugas akhirnya yang berjudul Sistem Informasi Penjadwalan Mata Pelajaran pada Sekolah Menengah Umum Dengan Metode Algoritma Genetika menjelaskan bahwa proses penjadwalan mata pelajaran dengan algoritma genetika menghasilkan output yang lebih optimal (lebih sedikit jadwal yang berbenturan) dibandingkan dengan proses manual dan diperoleh hasil berupa mata pelajaran yang efektif dan efisien. Metode algoritma genetika yang digunakan yaitu *breeder genetic algorithm* dimana metode ini hanya memilih individu terbaik untuk diseleksi dan dimasukkan ke *gen pool* untuk direkombinasi dan dimutasi. Pada algoritma genetika, teknik penentuan jadwal berdasarkan sejumlah solusi yang dikenal sebagai kromosom dimana kromosom terdiri dari beberapa individu yang terdapat dalam suatu populasi. Setiap individu merepresentasikan jadwal – jadwal yang tersusun berdasarkan hari, ruang kelas, dan guru mata pelajaran. Dalam masalah penjadwalan mata pelajaran pada sekolah menengah umum, gen yang menyusun sebuah kromosom terdiri dari kode mata pelajaran, waktu, kode kelas dan kode guru. Penetapan kriteria pembatas yang digunakan dalam penyelesaian masalah penjadwalan mata pelajaran ini yaitu mata pelajaran yang mempunyai prioritas sama dalam satu hari tidak boleh dijadwalkan berurutan dan dalam satu minggu suatu mata pelajaran tidak boleh dijadwalkan melebihi bobot yang telah ditetapkan misal mata pelajaran agama mempunyai bobot 2 jam pelajaran jadi mata pelajaran agama tidak boleh dijadwalkan lebih dari 2 kali serta mata pelajaran untuk kelas XI dan

XII sudah diterapkan sistem penjurusan sehingga ada perbedaan dalam mendapatkan pengajaran mata pelajaran berdasarkan jurusan yang diminati. Penggunaan algoritma genetika ini juga mempermudah pengolahan data akademik pada sekolah menengah umum sehingga segala pengolahan data dapat diproses dengan lebih cepat dan efisien.

Mawaddah (UB:2006) dalam tugas akhirnya yang berjudul Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika menjelaskan bahwa algoritma genetika dapat digunakan sebagai alternatif solusi untuk menyelesaikan masalah penjadwalan ujian yang hasil akhirnya berupa kromosom yang memiliki nilai fitness terbaik untuk dijadikan sebuah penjadwalan ujian. Dalam penjadwalan ujian ini diperlukan beberapa komponen yaitu dosen, waktu (hari dan jam) ujian dan ruang. Untuk pengkodeannya menggunakan pengkodean nilai dan inisialisasi kromosom yang direpresentasikan dalam bentuk larik yang berisi mata kuliah dan diurutkan berdasarkan id dosen untuk memudahkan penghitungan nilai *cost*. Dalam menghitung *fitness* dilakukan dengan memberikan pinalti terhadap aturan yang digunakan pada penjadwalan ini seperti kesediaan waktu dosen dengan nilai pinalti 1, konflik waktu dosen dengan nilai 2, dan sebagainya. Kemudian pada proses *crossover* menggunakan metode silang satu titik dengan menukar nilai gen pada posisi gen yang sama. Untuk proses mutasi menggunakan dua cara yaitu *random* dan *swap*. Pada hasil akhir untuk mendapatkan kromosom terbaik menggunakan seleksi roda *roulette* dengan memilih nilai *fitness* terbaik. Berdasarkan hasil uji coba disimpulkan bahwa rata-rata nilai *cost* akan semakin rendah dengan semakin besar probabilitas perkawinan silang dan probabilitas perkawinan silang yang optimal digunakan adalah 0.6, karena pada probabilitas tersebut nilai rata-rata *cost* terendah.

Berdasarkan jurnal tersebut maka kemudian penelitian dalam skripsi ini dilakukan. Perbedaannya adalah jika ketiga jurnal tersebut menjelaskan proses penyusunan jadwal pada perkuliahan, sekolah menengah umum dan penjadwalan ujian serta memiliki batasan kriteria berbeda yang nantinya berpengaruh saat merepresentasikan kromosom. Dalam skripsi ini melakukan proses penjadwalan pada sekolah menengah pertama sehingga terdapat perbedaan objek penelitian terhadap ketiga jurnal tersebut dan penetapan kriteria batasan dalam penyusunan

jadwal mata pelajaran. Kriteria pembatasnya adalah beban belajar dalam 1 minggu dibutuhkan 40 jam pembelajaran dengan durasi 40 menit, standar pengajaran bagi guru yang telah bersertifikasi minimal 24 jam per minggu sedangkan standarisasi tersebut tidak berlaku bagi guru tidak tetap sehingga guru tidak tetap hanya memenuhi jam kerja sesuai kebutuhan sekolah. Pada ketiga jurnal tersebut menggunakan algoritma genetika untuk melakukan proses penyusunan jadwal agar mendapatkan optimasi penjadwalan yang optimal dan memperoleh jadwal mata kuliah maupun mata pelajaran yang lebih efektif dan efisien. Oleh karena itu, skripsi ini menggunakan algoritma genetika karena algoritma ini sangat relevan untuk menyelesaikan kasus penjadwalan dan mempunyai penyusunan representasi kromosom yang paling efisien.

## 2.2 Penjadwalan Sekolah

Penjadwalan adalah sebuah proses dari penempatan tugas untuk sebuah kumpulan data jadwal. Penjadwalan mata pelajaran sekolah adalah suatu informasi yang menentukan mata pelajaran, hari dan jam dimulainya, ruang yang akan digunakan dimana akan membentuk suatu jadwal yang terstruktur dan tertata rapi sehingga kegiatan belajar dan mengajar dapat terselenggara dengan baik (Salimi,2006).

Penjadwalan merupakan proses untuk menyusun suatu jadwal atau urutan proses yang diperlukan dalam sebuah persoalan. Persoalan penjadwalan biasanya berhubungan dengan penjadwalan kelas dalam sekolah atau perkuliahan dan juga dalam lingkup yang tidak jauh berbeda seperti penjadwalan mata kuliah, penjadwalan ujian, atau bisa juga penjadwalan karyawan, baik dalam suatu perusahaan ataupun dalam rumah sakit. Dalam penjadwalan pelajaran sekolah akan dibahas tentang pembagian jadwal pelajaran untuk tiap-tiap kelas yang ada beserta guru pengajar pelajaran tersebut (Ariani, 2007).

Penjadwalan sekolah merupakan hal yang sangat penting dilakukan agar kegiatan belajar mengajar dapat berlangsung dengan baik. Banyaknya mata pelajaran yang diselenggarakan serta tenaga pengajar yang terbatas ditambah dengan sistem pendidikan sekolah yang telah ditetapkan menggunakan sistem semester maka sistem penjadwalan akan terasa sangat dibutuhkan untuk membuat jadwal mata pelajaran dengan mudah. Dengan adanya sistem penjadwalan mata

pelajaran ini akan mempermudah bagian kurikulum dalam mengontrol kegiatan belajar mengajar yang diselenggarakan dan mempermudah untuk memberikan informasi akademik kepada guru pengajar maupun kepada siswa serta memberikan laporan yang tepat dan akurat kepada kepala sekolah.

Pembuatan jadwal mata pelajaran yang dilakukan secara manual terdapat beberapa hal yang harus diperhatikan dimana akan sangat berpengaruh dalam proses pembuatan jadwal antara lain : jumlah guru pengajar, jumlah kelas tiap jenjang, jumlah hari, jumlah dan kombinasi jam pelajaran, jumlah bobot tiap mata pelajaran per minggu, jumlah mata pelajaran, dan beban tiap mata pelajaran.

Dalam mengatasi masalah tersebut maka langkah – langkah yang dapat diambil yaitu untuk mata pelajaran yang mempunyai bobot lebih dari 2 jam per minggu dibutuhkan catatan yang mencatat sisa jam pelajaran yang belum terjadwal karena dalam 1 hari tidak boleh terdapat lebih dari 2 jam mata pelajaran yang sama. Sedangkan untuk membedakan beban mata pelajaran yang sulit dan yang tidak berdasarkan survey pada siswa maka dalam sistem juga akan dibedakan menjadi 2 sehingga nantinya dalam 1 hari tidak terdapat mata pelajaran yang sulit diselenggarakan secara berturut – turut (Salimi, 2006).

Terdapat batasan-batasan dalam penyusunan penjadwalan mata pelajaran. Menurut Burke (2002) mengemukakan bahwa batasan-batasan dalam penjadwalan dibagi ke dalam dua kategori yaitu “*hard*” dan “*soft*” constraints. *Hard constraints* merupakan batas-batas yang harus diterapkan pada penjadwalan mata pelajaran dan harus dipenuhi. Solusi yang tidak melanggar *hard constraints* disebut solusi layak. *Hard constraints* yang umum dalam penjadwalan mata pelajaran adalah sebagai berikut (Burke, 2002):

- a. Seorang guru hanya dapat mengajarkan mata pelajaran untuk satu lokasi pada waktu tertentu.
- b. Seorang siswa hanya dapat mengikuti pelajaran untuk satu lokasi pada waktu tertentu.
- c. Sebuah lokasi (ruangan) hanya dapat digunakan untuk satu mata pelajaran pada waktu tertentu.
- d. Mata pelajaran dengan bobot 2 jam pelajaran dijadwalkan dengan satu kali pertemuan dalam seminggu.

- e. Hari aktif untuk kegiatan belajar mengajar adalah hari Senin sampai dengan Sabtu.

*Soft constraints* didefinisikan sebagai batas-batas mengenai alokasi sumber daya yang jika dilanggar masih dapat menghasilkan solusi yang layak tetapi sedapat mungkin untuk dipenuhi. Dalam kenyataannya, masalah penjadwalan mata pelajaran biasanya tidak mungkin untuk memenuhi semua *soft constraints*. Kualitas jadwal yang layak dapat dinilai berdasarkan seberapa baik *soft constraints* dapat dipenuhi. Namun beberapa masalah yang kompleks sulit menemukan solusi yang layak. Sebagai contoh, *soft constraints* yang mungkin ingin dicapai dalam jadwal sehubungan dengan aspek mata pelajaran adalah guru dapat meminta waktu mengajar tertentu yang diinginkan dan penempatan jadwal untuk waktu yang telah diminta guru disesuaikan dengan prioritas guru (Burke, 2002).

Dalam penyusunan penjadwalan mata pelajaran juga harus diperhatikan penyusunan program mengajar yang ada di SMPN 1 Gondang yaitu dilihat dari isi kurikulum sekolah yang bersangkutan, yang dimaksud isi disini terutama adalah jumlah atau macam pokok bahasan dan sub pokok bahasan dari setiap bidang studi masing-masing. Kurikulum SMPN 1 Gondang disusun agar sekolah memiliki pedoman penyelenggaraan kegiatan pembelajaran dan pemenuhan 8 Standar Nasional Pendidikan (SNP) dalam rangka mencapai tujuan pendidikan nasional. Pengembangan Kurikulum SMP Negeri 1 Gondang mengacu kepada karakteristik Kurikulum 2013 yang strukturnya disusun berdasarkan Standar Kompetensi Lulusan (SKL) dan Kompetensi Inti (KI), serta Kompetensi Dasar (KD) yang sesuai untuk semua mata pelajaran. Kurikulum 2013 memuat 11 mata pelajaran yang terdiri dari 8 mata pelajaran kelompok A yaitu Pendidikan Agama dan Budi Pekerti, Pendidikan Pancasila dan Kewarganegaraan, Bahasa Indonesia, Matematika, Ilmu Pengetahuan Alam, Ilmu Pengetahuan Sosial, Bahasa Inggris, Pendidikan Jasmani dan 3 mata pelajaran kelompok B yaitu Seni Budaya, Prakarya dan Bahasa Jawa. Pada kurikulum 2013 ini juga terdapat beban belajar yang merupakan keseluruhan kegiatan yang harus diikuti peserta didik dalam satu minggu, satu semester, dan satu tahun pembelajaran. Beban belajar ini termasuk

dalam aturan sebelum penjadwalan dibuat pada SMPN 1 Gondang sesuai dengan kurikulum 2013 adalah sebagai berikut:

- 1) Beban belajar satu minggu adalah 40 jam pembelajaran. Durasi setiap satu jam pembelajaran adalah 40 menit.
- 2) Standar pengajaran bagi guru yang telah bersertifikasi minimal 24 jam per minggu sedangkan standarisasi tersebut tidak berlaku bagi guru tidak tetap sehingga guru tidak tetap hanya memenuhi jam kerja sesuai kebutuhan sekolah.

Pada penjadwalan ini hanya terdapat *hard constraints* yang nantinya diproses oleh algoritma genetika yaitu tidak adanya bentrok pada penjadwalan dimana satu guru mengajar pada kelas yang berbeda dalam waktu yang sama. Oleh karena itu solusi penjadwalan yang dihasilkan tidak boleh melanggar *hard constraints* yang sudah ditentukan.

### 2.3 Algoritma Genetika

Algoritma genetika (AG) adalah sebuah teknik optimalisasi dan pencarian yang berdasar pada prinsip genetika dan seleksi alami (evolusi biologi). Metode ini dikembangkan pertama kali oleh John Holland (1975) dan muridnya yang bernama DeJong (1975) (Haupt, 2004).

*Genetic Algorithms* atau algoritma genetika adalah algoritma pencarian yang didasarkan atas mekanisme dari seleksi alam yang lebih dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. Hanya individu-individu yang kuat yang mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika, proses perkembangbiakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: “Bagaimana mendapatkan keturunan yang lebih baik” (Basuki, 2003).

Algoritma Genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap

hidup. Pada dasarnya ada 4 kondisi yang sangat mempengaruhi proses evaluasi yaitu (Kusumadewi, 2003):

1. kemampuan organisme untuk melakukan reproduksi
2. keberadaan populasi organisme yang bisa melakukan reproduksi
3. keberagaman organisme dalam suatu populasi
4. perbedaan kemampuan untuk *survive*

Secara umum, dalam algoritma genetika terdapat 5 (lima) proses, yaitu: pembentukan populasi awal, perhitungan nilai *fitness*, seleksi, regenerasi (*crossover* dan mutasi), penciptaan populasi baru hasil regenerasi. Tetapi dalam kasus ini, dalam proses regenerasi, sebuah kromosom hanya dilakukan proses *crossover* atau mutasi saja. Hal ini dilakukan dengan asumsi bahwa dengan cara ini akan didapatkan keturunan yang lebih baik. Algoritma genetika yang digunakan disini berfungsi untuk mendefinisikan permasalahan ke dalam suatu kumpulan solusi yang disebut dengan kromosom, kemudian melalui proses-proses yang ada dalam algoritma genetika, dilakukan pencarian solusi yang paling optimal yang didasarkan atas nilai *fitness*. Nilai *fitness* dihitung berdasarkan jumlah bentrokan yang terjadi pada jadwal yang dibuat. Semakin besar nilai *fitness* yang dihasilkan dari proses genetika maka semakin sedikit jumlah bentrokan atau pelanggaran yang terjadi dari solusi jadwal yang dihasilkan. Hal ini dapat dikatakan, bahwa dalam proses algoritma genetika untuk masalah penjadwalan adalah dicari nilai *fitness* sama dengan satu.

#### 2.4 Struktur Umum Algoritma Genetika

Algoritma genetika menggunakan mekanisme seleksi alam dan ilmu genetik. Hal ini berarti bahwa istilah-istilah yang terdapat pada algoritma genetik akan bersesuaian dengan istilah-istilah pada seleksi alam dan ilmu genetik. Dalam ilmu genetik kromosom terdiri dari susunan gen- gen. Tiap gen mengandung nilai atau sifat tertentu yang disebut allele, sedangkan posisi gen dalam kromosom disebut *locus*. Selanjutnya, satu atau beberapa kromosom bergabung membangun paket genetik yang disebut genotif. Interaksi genotif dengan lingkungannya disebut fenotif. Istilah ilmu genetik dan algoritma genetik pada biologi dapat ditunjukkan pada Tabel 2.1.

**Tabel 2.1** Istilah Ilmu genetik dan Algoritma genetik

Natural	Algoritma genetik
Kromosom	String
Gen	Karakter, feature
Allele	Nilai karakter
<i>Locus</i>	Posisi dalam string
Genotif	Struktur
Fenotif	Parameter
Populasi	Kumpulan string
<i>Fitness function</i>	Fungsi tujuan

Sumber : (Fitri R, 2004)

Berikut merupakan struktur umum pada Algoritma Genetika menurut :

1. Representasi Kromosom
2. Evaluasi dengan menghitung *fitness*.
3. Proses *crossover* untuk mendapatkan individu baru.
4. Proses mutasi yang berfungsi untuk meningkatkan variasi dalam populasi.
5. Proses seleksi untuk membentuk populasi baru.

Dari struktur ini bisa disusun siklus algoritma genetika yang ditunjukkan pada Gambar 2.1 berikut :

```

procedure Algoritma Genetika
begin
  t = 0
  inisialisasi P(t)
  while (bukan kondisi berhenti) do
    reproduksi C(t) dari P(t)
    evaluasi P(t) dan C(t)
    seleksi P(t+1) dari P(t) dan C(t)
    t = t + 1
  end while
end

```

**Gambar 2.1** Pseudocode siklus algoritma genetika

Beberapa definisi penting dalam algoritma genetika adalah sebagai berikut

(Basuki, 2003) :

- a. *Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan

- kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter.
- Allele, merupakan nilai dari gen.
  - Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
  - Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
  - Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
  - Generasi, menyatakan satu-satuan siklus proses evolusi.
  - Nilai *fitness*, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

Proses dalam algoritma genetika diawali dengan inisialisasi, yaitu menciptakan individu - individu secara acak yang memiliki susunan gen (*chromosome*) tertentu. *Chromosome* ini mewakili solusi dari permasalahan yang akan dipecahkan. Reproduksi dilakukan untuk menghasilkan keturunan (*offspring*) dari individu-individu yang ada di populasi. Evaluasi digunakan untuk menghitung *fitness* setiap *chromosome*. Semakin besar *fitness* maka semakin baik *chromosome* tersebut untuk dijadikan calon solusi. Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan hidup pada generasi berikutnya. Fungsi probabilitas digunakan untuk memilih individu yang dipertahankan hidup. Individu yang lebih baik (mempunyai nilai *fitness* lebih besar) mempunyai peluang lebih besar untuk terpilih.

Setelah melewati sekian iterasi (generasi) akan didapatkan individu terbaik. Individu terbaik ini mempunyai susunan *chromosome* yang bisa dikonversi menjadi solusi yang terbaik (paling tidak mendekati optimum). Dari sini bisa disimpulkan bahwa algoritma genetika menghasilkan suatu solusi optimum dengan melakukan pencarian di antara sejumlah alternatif titik optimum berdasarkan fungsi probabilitas (Mahmudy, 2013).

## 2.5 Komponen – Komponen Utama Algoritma Genetika

Penerapan algoritma genetika yaitu membangun generasi awal, representasi kromosom, fungsi *fitness*, seleksi, operator genetika, parameter genetika (Suyanto, 2011).

### 2.5.1 Generasi Awal

Langkah pertama dalam algoritma ini adalah membentuk sejumlah populasi awal yang digunakan untuk mencari penyelesaian optimal. Populasi awal yang dibangun dalam tugas akhir ini dengan menggunakan bilangan *random* (acak) dengan range bilangan yang telah ditentukan.

### 2.5.2 Representasi Kromosom

Representasi kromosom merupakan proses pengkodean dari penyelesaian asli suatu permasalahan. Solusi dari suatu masalah harus dipetakan (*encoding*) menjadi string kromosom. String kromosom ini tersusun atas sejumlah gen yang menggambarkan variabel – variabel keputusan yang digunakan dalam solusi (Mahmudy, 2013). Dalam berbagai macam kasus, representasi sebuah solusi menjadi kromosom sangat menentukan kualitas dari solusi yang dihasilkan (Mahmudy, 2012).

Terdapat banyak skema pengkodean dalam algoritma genetika dengan kelebihan dan kekurangannya masing-masing diantaranya adalah representasi biner, representasi integer, representasi *real* dan representasi permutasi. Pengkodean permutasi digunakan dalam masalah yang memerlukan 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 pada urutan seperti pada Gambar 2.2. berikut (Adamanti, 2002).

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

**Gambar 2.2** Contoh kromosom dengan pengkodean permutasi

Pengkodean permutasi hanya berguna pada masalah pengurutan. Dalam kasus optimasi penjadwalan mata pelajaran ini menggunakan representasi

permutasi berbasis kode guru. Kasus ini dapat dienkodkan dengan pengkodean permutasi dimana setiap angka dalam kromosom mewakili setiap kode guru dan memasukkan ke penjadwalannya sesuai dengan urutan angka dalam kromosomnya.

**2.5.3 Fungsi Fitness**

Fungsi *fitness* adalah fungsi yang digunakan untuk mengukur tingkat *fitness* suatu kromosom dalam populasi. Semakin besar nilai *fitness* maka semakin besar kemungkinan kromosom tersebut dapat tetap bertahan pada generasi berikutnya.

Individu-individu dalam populasi telah terbentuk, maka langkah selanjutnya adalah menghitung nilai *fitness* setiap individu. Perhitungan dilakukan dengan memberikan pinalti untuk setiap aturan yang digunakan dalam penjadwalan. Semakin wajib aturan dilaksanakan, maka akan semakin besar nilai pinalti yang diberikan. Berikut aturan perhitungan fungsi *fitness* yang ditunjukkan pada Persamaan 2-1 dan Persamaan 2-2 :

$$Fitness = \frac{1}{1+pinalti} \dots\dots\dots (2-1)$$

dimana  $Pinalti = \sum B_p N_p \dots\dots\dots (2-2)$

Dari Persamaan 2-1 dan 2-2 nilai *fitness* ditentukan oleh nilai pinalti. Pinalti tersebut menunjukkan jumlah pelanggaran terhadap *hard constraint* pada suatu kromosom. Jika ada satu guru yang mengajar pada kelas berbeda dalam waktu yang sama termasuk *hard constraint* yang diberikan pinalti sebesar 1. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai pinalti berbanding terbalik dengan nilai *fitness*, semakin kecil nilai pinalti (jumlah pelanggaran) semakin besar nilai *fitness*-nya seperti ditunjukkan pada Persamaan 2-3.

$$Fitness = \frac{1}{1+\sum B_p N_p} \dots\dots\dots (2-3)$$

Keterangan :

- Bp = Bobot pelanggaran
- Np = Indikator pelanggaran
- Fitness* = Fungsi *fitness*



### 2.5.4 Seleksi

Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan pada generasi berikutnya (Mahmudy, 2013). Individu yang lebih (individu yang mempunyai *fitness* lebih besar) mempunyai peluang besar untuk terpilih (Gen & Cheng, 1997 dalam Mahmudy, 2013). Semakin besar nilai *fitness* sebuah *chromosome* maka semakin besar peluangnya untuk terpilih. Hal ini dilakukan agar terbentuk generasi berikutnya yang lebih baik dari generasi sekarang. Metode seleksi yang sering digunakan adalah *roulette wheel*, *binary tournament*, dan *elitism*.

Metode yang digunakan pada penelitian ini adalah metode seleksi *elitism*. Metode seleksi *elitism* merupakan metode yang memilih *popSize* individu terbaik dari kumpulan individu di populasi (*parent*) dan *offspring*. Metode seleksi *elitism* bekerja dengan mengumpulkan semua individu dalam populasi (*parent*) dan *offspring* dalam satu penampungan. *PopSize* individu terbaik dalam penampungan ini akan lolos untuk masuk dalam generasi selanjutnya. Metode seleksi ini menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2013). *Pseudo-code* seleksi elitism ditunjukkan pada Gambar 2.3 sebagai berikut (Mahmudy, 2013):

#### PROCEDURE Elitism Selection

##### Input:

POP: himpunan individu pada populasi

pop\_size: ukuran populasi

OS: himpunan individu anak (*offspring*) hasil reproduksi menggunakan *crossover* dan mutasi

##### Output :

POP: himpunan individu pada populasi setelah proses seleksi selesai

/\* gabungkan individu pada POP dan OS ke dalam TEMP \*/

TEMP <- Merge (POP,OS)

/\* urutkan individu berdasarkan *fitness* secara ascending \*/

OrderAscending (Temp)

/\* copy pop\_size individu terbaik ke POP \*/

POP <- CopyBest (Temp, pop\_size)

END PROCEDURE

**Gambar 2.3** Pseudocode seleksi elitism

Salah satu kelemahan dari *elitism selection* adalah tidak memberikan kesempatan kepada individu dengan nilai *fitness* rendah untuk bereproduksi. Dalam beberapa kasus, solusi optimum justru bisa dicapai dari hasil reproduksi individu dengan nilai *fitness* rendah.

## 2.6 Operator Genetika

Operator genetika dipergunakan untuk mengkombinasi (modifikasi) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya. Ada dua operator genetika yaitu *crossover* dan *mutation* (Juniawati, 2003).

### 2.6.1 Persilangan (*Crossover*)

Operator persilangan merupakan operasi yang bekerja untuk menggabungkan dua kromosom orang tua (*parent*) menjadi kromosom baru (*offspring*). Tidak semua kromosom mengalami persilangan. Jumlah kromosom dalam populasi yang mengalami persilangan ditentukan oleh parameter yang disebut dengan *crossover rate* (probabilitas persilangan) (Juniawati, 2003).

*Crossover* dilakukan dengan memilih dua induk (*parent*) secara acak dari populasi. Pada pengkodean permutasi, jenis perkawinan silang yang sering dipergunakan adalah perkawinan silang 1-titik karena kesederhanaan prosesnya dapat menjaga konsistensi urutan nilai pada kromosom. Jenis perkawinan silang yang terdapat pada pengkodean biner tidak dapat dipergunakan pada pengkodean ini karena tidak dapat menjamin terjaganya konsistensi urutan nilai pada kromosom (Adamanti, 2002).

Metode *crossover* yang digunakan pada penelitian ini adalah perkawinan silang 1-titik (*one cut point*) dimulai dengan pemilihan satu titik perkawinan silang. Dari permutasi pertama sampai dengan titik perkawinan silang disalin dari orang tua pertama, sedangkan sisanya didapatkan dengan cara melihat satu persatu nilai pada orang tua kedua, jika belum ada pada kromosom keturunan, maka nilai tersebut ditambahkan seperti pada Gambar 2.4.

Kromosom orangtua 1	1 2 3 4 5 6 7 8 9
Kromosom orangtua 2	4 5 3 6 8 9 7 2 1
Keturunan (AND)	1 2 3 4 5 6 8 7 9

**Gambar 2.4** Contoh *one cut point crossover* pada pengkodean permutasi

### 2.6.2 Mutasi (*Mutation*)

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Beberapa cara operasi mutasi diterapkan dalam algoritma genetika menurut jenis pengkodean terhadap *phenotype*, antara lain (Adamanti, 2002):

#### 1. Mutasi dalam pengkodean biner

Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang dipilih secara acak (atau dengan menggunakan skema tertentu) pada kromosom, yang disebut inversi bit. Melakukan inversi pada bit yang terpilih, 0 menjadi 1 dan sebaliknya, 1 menjadi 0.

Contoh mutasi pada pengkodean biner

Kromosom sebelum mutasi : 1 0 0 1 0 **1** 1 1

Kromosom sesudah mutasi : 1 0 0 1 0 **0** 1 1

#### 2. Mutasi dalam pengkodean permutasi

Proses mutasi yang dilakukan dalam pengkodean biner dengan mengubah langsung bit-bit pada kromosom tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan permutasi harus diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (*locus*) dari kromosom dan kemudian nilainya saling dipertukarkan. Contoh mutasi dalam pengkodean permutasi.

Kromosom sebelum mutasi : 1 2 **3** 4 6 5 8 **7** 9

Kromosom sesudah mutasi : 1 2 **7** 4 6 5 8 **3** 9

Mutasi dilakukan dengan memilih satu induk secara acak dari populasi. Metode mutasi yang digunakan pada penelitian ini adalah *reciprocal exchange mutation* yang telah dimodifikasi dengan menggunakan pengkodean permutasi. Metode ini bekerja dengan memilih dua posisi secara random kemudian menukarkan dua nilai pada posisi tersebut (Mahmudy, 2013).

## 2.7 Parameter Genetika

Parameter-parameter genetik berguna dalam pengendalian operator-operator genetik. Pemilihan penggunaan nilai-nilai parameter genetik sangat berpengaruh terhadap kinerja algoritma genetik dalam menyelesaikan suatu masalah. Parameter-parameter genetik yang digunakan antara lain :

### 1. Ukuran populasi

Ukuran populasi mempengaruhi ukuran efektivitas dan kinerja algoritma genetik. Tidak ada aturan yang pasti tentang berapa nilai ukuran populasi. Apabila ukuran populasi kecil berarti hanya tersedia sedikit pilihan untuk *crossover* dan sebagian kecil dari domain solusi saja yang dieksplorasi untuk setiap generasinya. Sedangkan apabila terlalu besar, kinerja algoritma genetik akan menurun (Juniawati, 2003).

### 2. Jumlah generasi

Jumlah generasi berpengaruh terhadap banyaknya iterasi yang akan dikerjakan dan domain solusi yang akan dieksplorasi untuk setiap generasinya. Semakin besar jumlah generasi berarti semakin banyak iterasi yang dilakukan, dan semakin besar solusi yang dieksplorasi. Sedangkan semakin kecil jumlah generasinya, maka akan semakin kecil iterasi yang akan dilakukan, dan semakin kecil pula solusi yang dieksplorasi untuk tiap generasinya (Juniawati, 2003).

### 3. Probabilitas *crossover* ( $P_c$ )

Probabilitas *crossover* akan mengendalikan operator *crossover* dalam setiap generasi dalam populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover*, akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Sedangkan apabila nilai probabilitas *crossover* terlalu besar, individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya (Juniawati, 2003).

#### 4. Probabilitas mutasi ( $P_m$ )

Probabilitas mutasi akan mengendalikan operator mutasi pada setiap generasi. Peluang mutasi yang digunakan biasanya lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul. Oleh karena itu, operator mutasi pada algoritma genetik juga tidak selalu terjadi. Untuk itulah nilai peluang mutasi dibuat lebih kecil untuk setiap generasi (Juniawati, 2003).

### 2.8 Kondisi Berhenti (*Termination Condition*)

*Offspring* merupakan kromosom baru yang dihasilkan setelah melalui proses-proses di atas. Kemudian pada *offspring* tersebut dihitung fitnessnya apakah sudah optimal atau belum, jika sudah optimal berarti *offspring* tersebut merupakan solusi optimal, tetapi jika belum optimal maka akan diseleksi kembali, begitu seterusnya sampai terpenuhi kriteria berhenti.

Iterasi GAs diulang terus sampai kondisi berhenti tercapai. Beberapa kriteria bisa dipakai untuk hal ini sebagai berikut (Mahmudy, 2013):

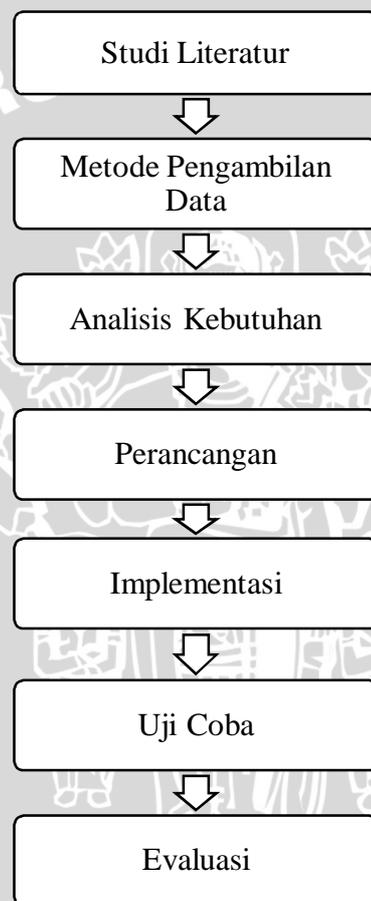
1. Iterasi berhenti sampai generasi  $n$ . Nilai  $n$  ditentukan sebelumnya berdasarkan beberapa eksperimen pendahuluan. Semakin tinggi ukuran dan kompleksitas masalah maka nilai  $n$  semakin besar. Nilai  $n$  ditentukan sedemikian rupa sehingga konvergensi populasi tercapai dan akan sulit didapatkan solusi yang lebih baik setelah  $n$  iterasi.
2. Iterasi berhenti setelah  $n$  generasi berurutan tidak dijumpai solusi yang lebih baik. Kondisi ini menunjukkan bahwa GAs sulit mendapatkan solusi yang lebih baik dan penambahan iterasi hanya membuang waktu.
3. Iterasi berhenti setelah  $t$  satuan waktu tercapai. Ini biasa digunakan jika diinginkan untuk membandingkan performa dari beberapa algoritma.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

Pada bab metodologi penelitian ini akan dibahas metode yang digunakan dalam penjadwalan mata pelajaran dan dijelaskan pula langkah – langkah dalam mengimplementasikan metode pada penjadwalan mata pelajaran. Adapun diagram alir metodologi yang akan digunakan seperti pada Gambar 3.1 sebagai berikut :



**Gambar 3.1** Diagram Alir Metodologi Penelitian

##### 3.1.1 Studi Literatur

Tahap ini dilakukan untuk mendapatkan dasar – dasar teori dan sumber acuan untuk merancang jadwal mata pelajaran menggunakan algoritma genetika. Informasi dan pustaka yang berkaitan dengan skripsi ini didapat dari buku, situs internet, penjelasan dari dosen pembimbing, dan rekan – rekan mahasiswa serta referensi lain yang dapat digunakan untuk menyelesaikan skripsi ini. Adapun

teori – teori yang dipelajari yaitu tentang penjadwalan sekolah dan juga penggunaan metode algoritma genetika untuk menyelesaikan suatu kasus sampai dengan cara perhitungannya dalam penjadwalan mata pelajaran di sekolah.

### 3.1.2 Metode Pengambilan Data

Pada penelitian ini metode yang digunakan dalam penjadwalan mata pelajaran adalah algoritma genetika. Lokasi penelitian skripsi ini adalah SMPN 1 Gondang lebih tepatnya di kota Mojokerto. Variabel penelitian skripsi ini adalah bagaimana penyusunan jadwal mata pelajaran yang baik bagi murid maupun guru sesuai kriteria. Hipotesis dari penelitian ini adalah algoritma genetika bisa diimplementasikan pada penyusunan jadwal mata pelajaran dengan optimal.

Berdasarkan cara pengumpulan data untuk kegiatan penelitian terdapat 2 jenis data yaitu data sekunder dan data primer. Data sekunder adalah data yang telah dikumpulkan oleh orang lain dan tidak dipersiapkan untuk kegiatan penelitian, tetapi dapat digunakan untuk tujuan penelitian. Data primer adalah data didapatkan langsung dari responden penelitian. Metode pengumpulan data primer yang bersifat kuantitatif dapat menggunakan instrument kuesioner dan wawancara.

Data yang digunakan pada penelitian ini ada dua yaitu data sekunder untuk mendapatkan jadwal mata pelajaran yang nantinya akan diproses dengan algoritma genetika dan data primer untuk mengetahui kriteria jadwal mata pelajaran yang baik dengan metode wawancara. Penentuan kebutuhan data yang diperlukan pada penelitian ini akan memudahkan dalam proses pencarian data untuk membangun sisten penjadwalan mata pelajaran.

### 3.1.3 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk menganalisis dan mendapatkan kebutuhan yang diperlukan dalam sistem penjadwalan mata pelajaran. Analisis kebutuhan bertujuan untuk menganalisis dan mendapatkan kebutuhan yang diperlukan dalam sistem penjadwalan mata pelajaran. Pada tahap ini merupakan tahap menganalisis hal- hal yang dibutuhkan untuk menjalankan sistem sehingga sistem dapat berjalan secara optimal. Beberapa analisis kebutuhan yaitu deskripsi sistem, data yang digunakan dan spesifikasi kebutuhan perangkat yang digunakan guna pembangunan sistem.

### 3.1.4 Perancangan

Setelah data terkumpul dan dianalisis, tahap selanjutnya adalah melakukan perancangan sistem. Perancangan ini bertujuan untuk mengetahui komponen jadwal mata pelajaran apa saja yang dibutuhkan untuk memasuki proses dalam algoritma genetika serta mempermudah identifikasi masalah, perancangan, implementasi dan pengujian sistem.

### 3.1.5 Implementasi

Implementasi dilakukan untuk menerapkan metode dalam penjadwalan mata pelajaran berdasarkan perancangan sistem yang telah dirancang sebelumnya. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman yaitu dengan menggunakan bahasa pemrogramana PHP. Pada pembuatan database digunakan database managemen system (DBMS) mysql dan *tools* pendukung lainnya. Implementasi penjadwalan mata pelajaran meliputi penerapan algoritma genetika dalam program yang dibuat dengan Bahasa PHP, pembuatan antar muka program dan memasukkan data penelitian ke database MySQL.

### 3.1.6 Uji Coba

Pada tahap ini dilakukan proses pengujian sistem dengan menggunakan data yang telah tersedia. Data akan diinputkan ke sistem dan diproses oleh sistem menggunakan perhitungan dengan algoritma genetika. Pengujian perangkat lunak pada penelitian ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhannya dan mengetahui seberapa baik metode yang digunakan dalam menyelesaikan kasus penjadwalan mata pelajaran.

### 3.1.7 Evaluasi

Langkah terakhir dalam penelitian ini adalah mengevaluasi hasil uji coba metode yang telah diterapkan. Tujuannya adalah mengetahui nilai – nilai parameter metode yang terbaik dalam penjadwalan mata pelajaran. Tahapan evaluasi ini juga dilakukan pada perangkat lunak yang digunakan untuk mengukur performasi dari sistem yang telah dibuat dan guna mengetahui kekurangan – kekurangan sementara dari sistem tersebut. Dari sini dapat

diketahui apakah sistem yang dibuat telah sesuai dengan tujuan awal dari dilakukannya penelitian.

### 3.2 Kebutuhan Sistem

Tahap analisis kebutuhan pada subbab ini merupakan tahap menganalisis segala hal yang berhubungan dengan proses pembuatan sistem implementasi metode algoritma genetika pada penjadwalan mata pelajaran. Pada analisa kebutuhan ini, diawali dengan deskripsi umum dari sistem, batasan implementasi dan apa saja yang dibutuhkan dalam proses implementasi. Analisis kebutuhan bertujuan untuk menggambarkan setiap kebutuhan yang memang diperlukan untuk memenuhi kebutuhan pengguna nantinya.

#### 3.2.1 Deskripsi Umum Sistem

Aplikasi yang dibangun adalah sebuah sistem yang menerapkan algoritma genetika dalam penyelesaian penjadwalan mata pelajaran. Untuk mendapatkan jadwal mata pelajaran yang paling efisien dan mempunyai kombinasi terbaik maka dibutuhkan bermacam – macam penjadwalan dengan beberapa kemungkinan. Data masukan dalam sistem ini yaitu mata pelajaran, guru pengajar, waktu pengajaran, dan kelas sebagai tempat belajar mengajar. Pada sistem ini data masukan akan dikodekan menjadi populasi awal sebagai calon solusi jadwal mata pelajaran. Populasi awal tersebut terdiri dari sejumlah solusi jadwal mata pelajaran. Representasi algoritma genetika dalam penjadwalan mata pelajaran ditunjukkan pada Tabel 3.1.

**Tabel 3.1** Representasi Algoritma Genetika Dalam Penjadwalan Mata Pelajaran

Populasi	Kumpulan sejumlah solusi jadwal
Individu	1 solusi jadwal
Kromosom	1 jadwal yang terdiri dari 1 mata pelajaran dalam 1 hari dan jam tertentu.
Gen	Atribut dari jadwal yaitu nama guru pengajar, mata pelajaran, kelas, porsi jam mata pelajaran.
<i>Fitness</i>	Jika nilai <i>fitness</i> suatu jadwal sama dengan 1 maka kualitas jadwal tersebut baik dan berpeluang menjadi solusi tetapi jika nilai <i>fitness</i> kurang dari 1

maka kualitas jadwal tersebut masih belum baik.
---

Setelah representasi kromosom dari data masukan telah dikodekan lalu dibuat populasi awal yang terdiri dari solusi jadwal yang dibentuk secara random. Kemudian dilakukan *crossover* dengan memilih 2 solusi jadwal secara random. Langkah berikutnya adalah mutasi yang juga dilakukan secara random dari solusi jadwal yang ada. Lalu solusi jadwal yang sudah melalui proses *crossover* dan mutasi dikumpulkan menjadi satu untuk dilakukan proses seleksi. Metode seleksi yang digunakan yaitu *elitism selection*. Metode ini dilakukan dengan memilih *popSize* individu terbaik dari kumpulan individu di populasi (*parent*) dan *offspring* berdasarkan nilai *fitness* yang terbesar. Semakin tinggi nilai *fitness* pada suatu solusi jadwal, maka jadwal tersebut memiliki kualitas yang semakin baik. Kriteria kualitas jadwal yang baik adalah jadwal yang tidak melanggar *constraint* yang sudah ditentukan.

Dalam penelitian ini terdapat aturan sebelum penjadwalan dibuat yaitu standar pengajaran bagi guru yang telah bersertifikasi minimal 24 jam per minggu sedangkan standarisasi tersebut tidak berlaku bagi guru tidak tetap sehingga guru tidak tetap hanya memenuhi jam kerja sesuai kebutuhan sekolah dan bagi Kepala Sekolah memiliki standar pengajaran minimal 6 jam agar memenuhi standar dari sertifikasi serta beban belajar pada sekolah dalam 1 minggu dibutuhkan 40 jam pembelajaran dengan durasi 40 menit dan seorang guru hanya dapat mengajarkan mata pelajaran untuk satu lokasi pada waktu tertentu. Untuk *hard constraint* pada penjadwalan ini yaitu tidak adanya bentrok guru pengajar pada solusi penjadwalan.

### 3.2.2 Batasan Implementasi

Batasan dari perangkat lunak yang akan dikembangkan adalah sebagai berikut:

1. Data yang digunakan adalah data Jadwal Mata Pelajaran SMPN 1 Gondang Mojokerto Tahun Ajaran 2014 / 2015.
2. Metode seleksi yang digunakan dalam penelitian ini adalah metode *elitism selection*.

3. Metode *crossover* yang digunakan dalam penelitian ini adalah *one cut point crossover*.
4. Metode mutasi yang digunakan dalam penelitian ini adalah *reciprocal exchange mutation*.

### 3.2.3 Spesifikasi Kebutuhan Perangkat

Spesifikasi kebutuhan perangkat yang digunakan dalam pembuatan sistem ini meliputi:

- a. Spesifikasi Kebutuhan *Hardware*
  - Prosesor Inter (R) Core™ 15-2410M CPU @ 2.30 GHz.
  - Memori 4.00 GB.
  - Hardisk dengan kapasitas 500GB.
- b. Spesifikasi Kebutuhan *Software*
  - Sistem operasi menggunakan Windows (Windows 8.1 Enterprise).
  - Bahasa pemrograman PHP.
  - Database untuk pengolahan data menggunakan database management system (DBMS) MySQL dengan XAMPP 3.2.1.

### 3.3 Formulasi Permasalahan

Sebagaimana telah dijelaskan sebelumnya pada deskripsi umum sistem bahwa dalam menyelesaikan kasus optimasi penjadwalan mata pelajaran dibutuhkan beberapa data masukan yang nantinya digunakan untuk merepresentasikan data penjadwalan mata pelajaran dengan algoritma genetika. Beberapa data masukan yang akan digunakan yaitu daftar mata pelajaran dan jumlah porsi jam pelajaran, daftar guru pengajar dan mata pelajaran, porsi waktu pengajaran serta jam pengajaran setiap harinya.

Pada penelitian ini terdapat 11 mata pelajaran. Untuk daftar mata pelajaran dan jumlah porsi jam pelajaran dalam seminggu ditunjukkan pada Tabel 3.2 berikut.

**Tabel 3.2** Daftar Mata Pelajaran dan Jumlah Porsi Jam Pelajaran

	Mata Pelajaran	Porsi Jam
1.	Pendidikan Agama dan Budi Pekerti	3

2.	Pendidikan Pancasila dan Kewarganegaraan	3
3.	Bahasa Indonesia	6
4.	Matematika	5
5.	Ilmu Pengetahuan Alam	5
6.	Ilmu Pengetahuan Sosial	4
7.	Bahasa Inggris	4
8.	Seni Budaya	3
9.	Pendidikan Jasmani, Olah Raga, dan Kesehatan	3
10.	Prakarya	2
11.	Bahasa Jawa	2
Jumlah Alokasi Waktu per Minggu		40

Daftar guru pengajar dan mata pelajaran yang diajarkan beserta status untuk membedakan porsi jam mengajarnya ditunjukkan pada Tabel 3.3 berikut.

**Tabel 3.3** Daftar Guru Pengajar dan Mata Pelajaran

No	Nama Guru	Mata Pelajaran	Status
1	Bahtiar Mashudi, S.Pd	Matematika	Kepala Sekolah
2	Sri Hastuti, S.Pd.	IPA	Sertifikasi
3	Hj.Luluk Muflikhatin, M PdI	Pend. Agama	Sertifikasi
4	Drs. Musta'in	Penjas	Sertifikasi
5	Siti Kareni, S.Pd.	Bhs. Indonesia	Sertifikasi
6	Dra.Eveline, M.Pd	Bhs. Indonesia	Sertifikasi
7	Setyo Rakhmad, S.Pd.	Matematika	Sertifikasi
8	Meineni Prihartini, S.Pd.	IPA	Sertifikasi
9	Hj. Rahma Murwani, S.Pd	Bhs. Indonesia	Sertifikasi
10	Rohimah, S.Pd	IPA	Sertifikasi
11	Dian Estusih Wilujeng, S.Pd.	PKn	Sertifikasi
12	H. Makudi, S.Ag.	Pend. Agama	Sertifikasi
13	Warnadi, S.Pd.	IPS	Sertifikasi
14	Andik Susanto, S.Pd.	Bhs.Jawa	Sertifikasi
15	Erni Puji Lestari, S.Pd.	IPS	Sertifikasi
16	H.Purnomo, S.Pd	Penjas	Sertifikasi

17	Hj.Sumartiyah,S.Pd	Matematika	Sertifikasi
18	Ummu Hanik, S.Pd.	IPS	Sertifikasi
19	Betty Endang Pramuni,S.Pd	IPA	Sertifikasi
20	Nurbudi Hariono	Seni Budaya	Sertifikasi
21	Yayuk Purwantini, S.Pd.	Matematika	Sertifikasi
22	Ilyas, S.Pd.	Bhs.Indonesia	Sertifikasi
23	Setyo Purwantini, S.Pd.	Matematika	Sertifikasi
24	Dewi Nenitriana, S.Pd.	Bhs.Inggris	Sertifikasi
25	Iwan Winardi, S.Pd	Penjas	Sertifikasi
26	Kasminingsih, S.Pd.	Bhs.Inggris	Sertifikasi
27	Kuwat Setyobudi, S.Pd.	Bhs. Inggris	Sertifikasi
28	Retno Anggraeni, S.Pd.	Bhs. Inggris	Sertifikasi
29	Maria Ulfah, S.Pd.	Bhs.Indonesia	Sertifikasi
30	Tutik Herawati, S.Pd.	Matematika	Sertifikasi
		IPA	Sertifikasi
31	Misnan	IPS	Sertifikasi
32	Trisia Santi Parlina, S.Pd.	Seni Budaya	Sertifikasi
33	Sopii,S.Pd	Bhs.Jawa	Sertifikasi
34	Oktriyana Ricke V, S.Pd.	Prakarya	Sertifikasi
35	Masripah Farikah, S.Pd.	Pkn	Sertifikasi
36	Nur Farida, S.PdI.	Pend. Agama	GTT
37	Andriyati, S.Kom.	Prakarya	GTT
38	Arismoyo Tri Utomo	PKn	GTT
		Seni Budaya	

Keterangan :

- Guru yang punya status sertifikasi mempunyai porsi pengajaran minimal 24 jam per minggu.
- Guru yang termasuk GTT ( Guru Tidak Tetap) mempunyai porsi pengajaran tergantung kebutuhan.
- Status Kepala Sekolah mempunyai porsi pengajaran minimal 6 jam per minggu.

Untuk ketersediaan kelas yang digunakan sebagai tempat pengajaran terdapat 24 kelas yaitu 7A – 7H, 8A – 8H, dan 9A – 9H. Sedangkan porsi waktu pengajaran bisa dilihat pada Tabel 3.4 berikut.

**Tabel 3.4** Porsi Waktu Pengajaran

JAM KE-	PUKUL
1	07.00 - 07.40
2	07.40 - 08.20
3	08.20 - 09.00
4	09.20 - 10.00
5	10.00 - 10.40
6	10.40 - 11.20
7	11.40 - 12.20
8	12.20 - 13.00

Untuk jam pengajarannya terdapat perbedaan pada hari – hari tertentu yang bisa dilihat pada Tabel 3.5 berikut :

**Tabel 3.5** Jam Pengajaran

Hari	Jam ke -							
	1	2	3	4	5	6	7	8
Senin								
Selasa								
Rabu								
Kamis								
Jumat								
Sabtu								

Pada Tabel 3.5 menjelaskan tentang jam pengajaran setiap harinya dengan tanda warna kuning yang menunjukkan jam aktif mengajar pada masing – masing hari tersebut.

Dari data masukan yang ada nantinya akan digunakan untuk merepresentasikan kromosom dengan algoritma genetika. Sebelum membuat representasi kromosom, peneliti harus menyusun penjadwalan dengan

memperhatikan beberapa ketentuan berdasarkan jumlah porsi jam pengajaran guru dan mata pelajarannya. Kemudian hasil dari penugasan jadwal tersebut dirubah menjadi kode sebagai representasi kromosomnya. Selanjutnya akan terbentuk beberapa representasi kromosom yang nantinya akan dihitung nilai *fitness* berdasarkan hasil pelanggaran terhadap *hard constraint* yang telah ditentukan lalu hasilnya diseleksi dengan memilih nilai *fitness* tertinggi. Setelah itu akan didapatkan representasi kromosom dengan kualitas jadwal yang baik dan optimal dengan ketentuan bahwa solusi jadwal tersebut tidak terjadi bentrokan pada jadwal gurunya. Oleh karena itu digunakan algoritma genetika untuk mempermudah dalam mendapatkan solusi jadwal dengan kualitas jadwal yang baik.

### 3.4 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

Pada penelitian ini akan membahas tentang masalah optimasi penjadwalan mata pelajaran pada sekolah menengah pertama dengan menggunakan metode algoritma genetika. Berdasarkan analisis yang telah dilakukan, maka pada subbab ini akan dibahas mengenai proses pada metode yang akan diterapkan pada perangkat lunak. Data yang digunakan pada penelitian ini adalah data mata pelajaran, data guru, data waktu dan tempat pengajaran. Proses algoritma genetika terdiri dari beberapa fungsi yaitu representasi kromosom, hitung *fitness*, *crossover*, mutasi, dan seleksi. Flowchart untuk proses algoritma genetika yang digunakan dapat dilihat pada Gambar 3.2 berikut.



**Gambar 3.2** Flowchart Proses Algoritma Genetika

Dari Gambar 3.2 telah dijelaskan proses algoritma genetika yang akan digunakan pada penelitian ini. Langkah pertama yaitu membuat inisialisasi parameter awal dengan menentukan jumlah individu pada sebuah populasi, jumlah generasi atau iterasi yang akan dilakukan, *crossover rate* ( $cr$ ), dan *mutation rate* ( $mr$ ). Langkah kedua yaitu membangkitkan populasi awal sebanyak individu yang telah diinisialisasikan. Langkah ketiga yaitu menghitung nilai *fitness* dari populasi awal berdasarkan jumlah nilai pelanggaran. Langkah keempat yaitu membuat populasi baru dengan melakukan proses *crossover* menggunakan metode *one cut point* pada induk secara acak dan lakukan secara berulang berdasarkan  $cr$  yang telah ditentukan kemudian melakukan proses mutasi menggunakan metode

*reciprocal exchange mutation* dengan induk secara acak juga serta dihitung *fitness* pada masing – masing kromosom yang telah didapatkan. Langkah terakhir yaitu melakukan proses seleksi dengan metode *elitism selection* untuk menentukan kromosom induk pada generasi selanjutnya. Jika kondisi akhir terpenuhi, berhenti dan hasilnya adalah solusi terbaik pada seluruh generasi.

Pada tahap ini terdapat beberapa parameter awal antara lain menentukan *popSize* untuk menetapkan banyaknya populasi pada tiap generasi, *crossover rate* untuk menentukan banyaknya *offspring* yang dihasilkan pada proses *crossover* terhadap ukuran populasi sehingga menghasilkan *offspring* dengan rumus  $cr \times popSize$ , *mutation rate* untuk menentukan banyaknya *offspring* yang dihasilkan pada proses mutasi terhadap ukuran populasi sehingga menghasilkan *offspring* dengan rumus  $mr \times popSize$ .

Sebelum dilakukan representasi kromosom untuk penjadwalan mata pelajaran maka diperlukan beberapa pendekatan untuk menyusun kromosom tabel penugasan guru. Ada beberapa pendekatan untuk menyusun kromosom jadwal mata pelajaran yaitu memasukkan kode guru pengajar terlebih dahulu ke dalam jadwal dengan menyesuaikan porsi jam mengajarnya atau memasukkan kode mata pelajaran dahulu dengan menyesuaikan porsi jam mata pelajarannya. Pada penelitian ini digunakan pendekatan untuk menyusun kromosom dengan memasukkan semua data guru pengajar terlebih dahulu berdasarkan porsi jam mengajarnya dalam seminggu diikuti dengan kode mata pelajaran, kelas dan porsi jam mata pelajarannya.

Dari penjelasan yang telah ada maka dibuatlah suatu penyelesaian optimasi penjadwalan mata pelajaran dengan data yang tersedia yaitu terdapat 4 guru pengajar beserta porsi jam mengajar masing –masing yaitu 20 jam dengan ketersediaan kelas sebanyak 3 kelas. Data guru pengajar bisa dilihat pada Tabel 3.6 berikut.

**Tabel 3.6** Data Guru Pengajar

No	Nama Guru	Mata Pelajaran
1	Sri Hastuti, S.Pd.	IPA
2	Hj.Luluk Muflikhatin,M PdI	Pend. Agama

3	Drs. Musta'in	Penjas
4	Siti Kareni, S.Pd.	Bhs. Indonesia

Untuk menjelaskan perhitungan manual dengan algoritma genetika diperlukan penugasan guru mata pelajaran untuk kelas tertentu seperti ditunjukkan pada Tabel 3.7 berikut.

**Tabel 3.7** Penugasan Guru

Kode	Nama Guru	Mata Pelajaran	Kelas	Porsi Jam Mapel
1	Sri Hastuti, S.Pd.	IPA	7A	2
2	Sri Hastuti, S.Pd.	IPA	7A	2
3	Sri Hastuti, S.Pd.	IPA	7A	1
4	Sri Hastuti, S.Pd.	IPA	7B	2
5	Sri Hastuti, S.Pd.	IPA	7B	2
6	Sri Hastuti, S.Pd.	IPA	7B	1
7	Sri Hastuti, S.Pd.	IPA	7C	2
8	Sri Hastuti, S.Pd.	IPA	7C	2
9	Sri Hastuti, S.Pd.	IPA	7C	1
10	Hj.Luluk Muflikhatin,M PdI	Pend. Agama	7A	2
11	Hj.Luluk Muflikhatin,M PdI	Pend. Agama	7A	1
12	Hj.Luluk Muflikhatin,M PdI	Pend. Agama	7B	2
13	Hj.Luluk Muflikhatin,M PdI	Pend. Agama	7B	1
14	Hj.Luluk Muflikhatin,M PdI	Pend. Agama	7C	2
15	Hj.Luluk Muflikhatin,M PdI	Pend. Agama	7C	1
16	Drs. Musta'in	Penjas	7A	2
17	Drs. Musta'in	Penjas	7A	1
18	Drs. Musta'in	Penjas	7B	2
19	Drs. Musta'in	Penjas	7B	1
20	Drs. Musta'in	Penjas	7C	2
21	Drs. Musta'in	Penjas	7C	1

22	Siti Kareni, S.Pd.	Bhs. Indonesia	7A	2
23	Siti Kareni, S.Pd.	Bhs. Indonesia	7A	2
24	Siti Kareni, S.Pd.	Bhs. Indonesia	7A	2
25	Siti Kareni, S.Pd.	Bhs. Indonesia	7B	2
26	Siti Kareni, S.Pd.	Bhs. Indonesia	7B	2
27	Siti Kareni, S.Pd.	Bhs. Indonesia	7B	2
28	Siti Kareni, S.Pd.	Bhs. Indonesia	7C	2
29	Siti Kareni, S.Pd.	Bhs. Indonesia	7C	2
30	Siti Kareni, S.Pd.	Bhs. Indonesia	7C	2

Pada Tabel 3.7 merupakan penugasan guru dari data yang telah tersedia dengan ketersediaan kelas sebanyak 3 yaitu kelas 7A, 7B dan 7C. Karena ada 30 penugasan maka panjang kromosom juga sebesar 30. Setelah itu menentukan ukuran populasi, *crossover rate* dan *mutation rate* untuk proses selanjutnya. Untuk menyelesaikan kasus ini menggunakan ukuran populasi = 2,  $cr = 1$  dan  $mr = 0.5$ .

### 3.4.1 Representasi Kromosom dan Perhitungan *Fitness*

Representasi kromosom yang digunakan pada penelitian ini adalah representasi permutasi berupa bilangan bulat. Dalam merepresentasikan kromosomnya berupa *array* yang nantinya berisi kode dari Tabel 3.7 dengan panjang kromosom sesuai dengan jumlah kode penugasan guru yang telah ditentukan sebelumnya. Contoh komponen kromosom ditunjukkan pada Tabel 3.8 berikut.

**Tabel 3.8** Komponen Kromosom

Gen 1	Gen 2	Gen 3	....
Kode penugasan guru	Kode penugasan guru	Kode penugasan guru	....

Langkah selanjutnya yaitu membuat inisialisasi kromosom yang dilakukan secara random terhadap kode penugasan guru pada Tabel 3.7. Contoh representasi kromosom dapat dilihat pada Tabel 3.9 berikut.

**Tabel 3.9** Representasi Kromosom

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	10	3	15	26	11	20	1	8	2	16	22	4	19	28	12	27	18	29	6

21	22	23	24	25	26	27	28	29	30
13	25	17	7	21	9	14	23	30	24

Tabel 3.9 merupakan contoh representasi kromosom dengan panjang kromosom 30. Kromosom tersebut disusun secara random berdasarkan pada kode yang ada pada Tabel 3.7. Untuk penjelasannya sebagai contoh pada gen ke – 1 yang berisi angka 5 merupakan kode penugasan guru atas nama Sri Hastuti, S.Pd. yang mengajar IPA, gen ke – 2 yaitu Hj.Luluk Muflikhatin, M PdI dan seterusnya. Oleh karena itu satu kromosom tersebut dapat menyelesaikan suatu masalah penjadwalan.

Setelah menghasilkan representasi kromosom kemudian kromosom tersebut dirubah menjadi sebuah jadwal yang ditunjukkan pada Tabel 3.10 berikut.

**Tabel 3.10** Penyusunan Jadwal Hasil Representasi Kromosom

Kelas	Senin								Selasa							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
7A			10	10	3	11	1	1	2	2	16	16	22	22	17	23
7B			5	5	26	26	4	4	19	12	12	27	27	18	18	6
7C			15	20	20	8	8	28	28	29	29	7	7	21	9	14

Kelas	Rabu		
	1	2	3
7A	23	24	24
7B	13	25	25
7C	14	30	30

Pada Tabel 3.10 merupakan hasil dari proses perubahan kromosom menjadi sebuah jadwal dimana kromosomnya berasal dari representasi kromosom pada Tabel 3.9 dan terdapat juga kolom yang mewakili porsi waktu pengajaran seperti yang ada pada Tabel 3.4. Untuk memasukkan kode yang ada pada representasi kromosom ke sebuah penjadwalan dilihat dari Tabel 3.7 dengan memperhatikan kelas dan porsi jam pelajarannya serta harus berurutan dalam memasukkan sesuai dengan representasi kromosomnya. Misalnya pada gen ke – 1

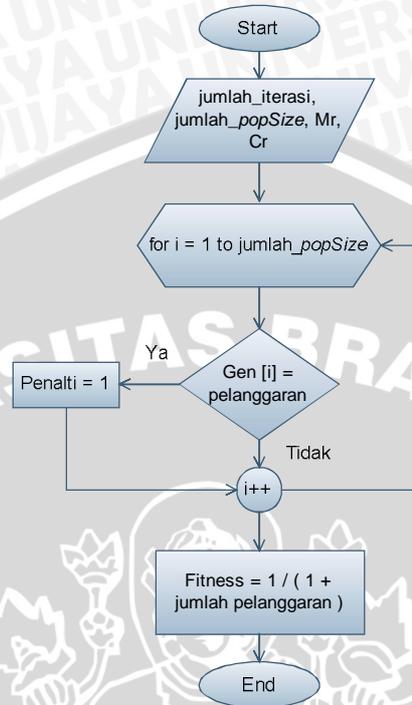
yaitu angka 5 atas nama Sri Hastuti, S.Pd. yang mengajar IPA pada kelas 7B dengan porsi jam pelajarannya ada 2 jam maka angka tersebut dimasukkan pada slot kelas 7B sebanyak dua karena setiap angka 5 mewakili 1 jam porsi pengajaran. Selain itu pada penjadwalan tersebut hanya menampilkan representasi kromosom pada hari Senin, Selasa dan Rabu dikarenakan data yang digunakan hanya sebagian dari data masukan yang sebenarnya yaitu data mata pelajaran. Pada data mata pelajaran sebenarnya ada 11 mata pelajaran tetapi pada perhitungan ini hanya menggunakan 4 mata pelajaran sehingga tidak sesuai dengan porsi waktu yang tersedia. Untuk memasukkan penjadwalan maka dimulai dari hari Senin jam ke 3 yang dapat dilihat juga pada Tabel 3.5 untuk penjelasan jam aktif mengajar masing – masing harinya.

Setelah menyusun representasi kromosom secara acak seperti contoh pada Tabel 3.9 selanjutnya dilakukan proses perhitungan *fitness*. Nilai *fitness* dihitung berdasarkan jumlah pelanggaran terhadap *hard constraint*. Pada masalah optimasi penjawalan ini, solusi yang akan dicari adalah meminimalkan fungsi pinalti (masalah minimasi) maka fungsi pinalti tidak bisa digunakan secara langsung. Karena ada aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu, nilai *fitness* yang digunakan adalah  $f = \frac{1}{\text{pinalti}}$ , artinya semakin kecil nilai pinalti maka semakin besar nilai *fitness*. Tetapi hal ini akan menjadi masalah jika nilai pinalti bernilai 0 yang mengakibatkan *fitness* bisa bernilai tak terhingga. Untuk mengatasinya pinalti perlu ditambahi sebuah bilangan yang dianggap kecil (0-1) sehingga rumus perhitungan nilai *fitness* dalam penelitian ini seperti pada Persamaan 3-1 berikut.

$$\text{Fitness} = \frac{1}{1+\Sigma P} \dots\dots\dots (3-1)$$

Keterangan : P = Pelanggaran yang dilakukan jika 1 guru berada pada lebih dari 1 kelas dalam satu hari dan jam tertentu dikenakan penalti 1 (*hard constraint*).

Flowchart algoritma yang digunakan untuk menghitung nilai *fitness* ditampilkan pada Gambar 3.3 berikut.



**Gambar 3.3** Algoritma Evaluasi *Fitness*

Pada proses ini dilakukan penghitungan nilai *fitness* menggunakan Persamaan 3-1. Nilai *fitness* akan menunjukkan kualitas dari solusi jadwal yang dihasilkan jika nilai *fitness* semakin tinggi maka semakin baik kualitas jadwalnya. Kualitas jadwal ditentukan berdasarkan pelanggaran yang dilakukan pada solusi jadwal tersebut. Proses perhitungan berdasarkan *hard constraint* yaitu jika 1 guru berada pada lebih dari 1 kelas dalam satu hari dan jam tertentu. Contoh evaluasi *fitness* dari penyusunan jadwal hasil representasi kromosom dapat dilihat pada Tabel 3.11 sebagai berikut:

**Tabel 3.11** Contoh Evaluasi *Fitness*

Kelas	Senin								Selasa							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
7A			10	10	3	11	1	1	2	2	16	16	22	22	17	23
7B			5	5	26	26	4	4	19	12	12	27	27	18	18	6
7C			15	20	20	8	8	28	28	29	29	7	7	21	9	14
∑ Pelanggaran			1	-	-	-	2	1	-	-	-	-	1	1	1	-
∑ Total	4								3							

Kelas	Rabu		
	1	2	3
7A	23	24	24
7B	13	25	25
7C	14	30	30
$\sum$ Pelanggaran	1	2	2
$\sum$ Total	5		

Untuk melihat pelanggaran yang ada pada solusi jadwal bisa dilihat berdasarkan nama guru yang ada di Tabel 3.7. Pada Tabel 3.11 menggambarkan terjadinya bentrokan jam antara kelas 7A, 7B dan 7C karena ada 1 guru yang mengisi pada jam yang sama di kelas berbeda. Misalnya pada hari Senin terjadi bentrokan jam atas nama Hj.Luluk Muflikhatin,M PdI dengan kode 10 dan 15. Terjadinya bentrokan jam pengajaran karena guru pengajar atas nama Hj.Luluk Muflikhatin,M PdI. mempunyai kode 10 – 15 yang bisa dilihat pada Tabel 3.7 berada pada waktu yang sama dengan kelas yang berbeda pada solusi jadwal maka nilai pelanggaran dalam fungsi *fitness* yaitu bernilai 1. Oleh karena itu bentrokan tersebut termasuk *hard constraint* yang dilakukan pada solusi jadwal sehingga nilai pelanggaran tersebut bernilai 1. Untuk menghitung nilai pelanggaran dari Tabel 3.11 sebagai berikut:

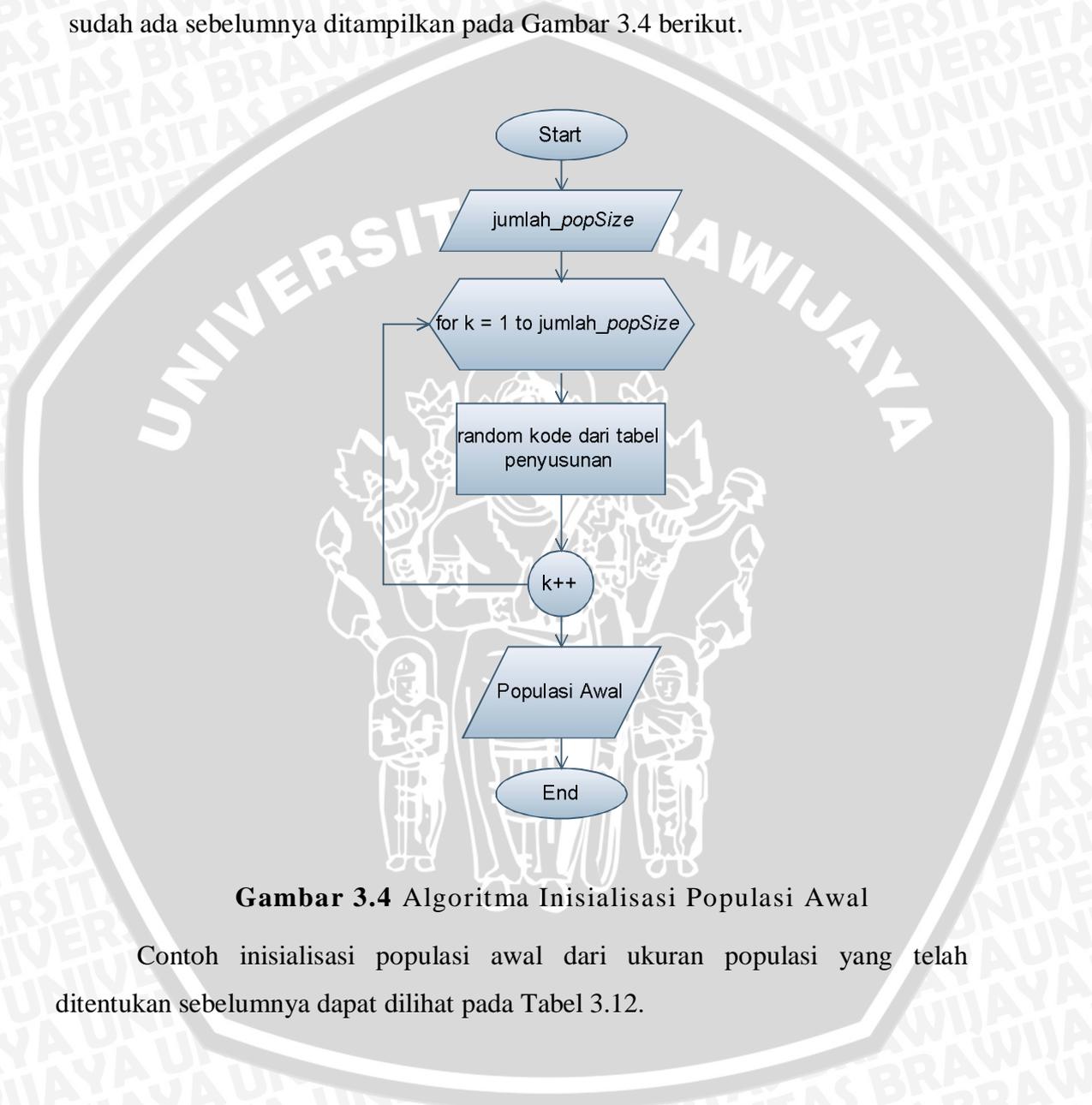
$$\sum \text{Total} = 4 + 3 + 5 = 12$$

Pada perhitungan nilai pelanggaran di atas didapatkan nilai 12 dimana nilai tersebut berasal dari pelanggaran yang dihitung dari penjumlahan pelanggaran setiap harinya. Telah dijelaskan sebelumnya bahwa pelanggaran dihitung dari bentrokan jam yang sama dengan kelas yang berbeda namun hanya satu guru yang mengisi sehingga setiap terjadi bentrokan jam akan bernilai 1. Contoh perhitungan nilai *fitness* pada Tabel 3.11 yaitu:

$$\text{Fitness} = \frac{1}{1 + \sum P} = \frac{1}{1 + 12} = 0.0769$$

### 3.4.2 Inisialisasi Populasi Awal

Pada tahap ini adalah membuat populasi awal sebesar `popSize` yang telah ditentukan dan inisialisasi jadwal ini dilakukan secara *random*. Flowchart algoritma yang digunakan untuk membuat kromosom populasi awal atau yang sudah ada sebelumnya ditampilkan pada Gambar 3.4 berikut.



**Gambar 3.4** Algoritma Inisialisasi Populasi Awal

Contoh inisialisasi populasi awal dari ukuran populasi yang telah ditentukan sebelumnya dapat dilihat pada Tabel 3.12.

**Tabel 3.12** Inisialisasi Populasi Awal

P1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	10	3	15	26	11	20	1	8	2	16	22	4	19	28	12	27	18	29	6

21	22	23	24	25	26	27	28	29	30
13	25	17	7	21	9	14	23	30	24

P2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
11	6	20	7	19	1	10	22	29	16	8	23	2	9	18	27	4	12	21	30

21	22	23	24	25	26	27	28	29	30
5	14	26	13	25	3	15	17	24	28

Inisialisasi populasi awal yang terdapat pada Tabel 3.12 merupakan kode guru yang berasal dari Tabel 3.7 kemudian disusun secara *random* sehingga membentuk sebuah representasi kromosom dengan panjang 30. Untuk ukuran populasi sudah ditentukan sebelumnya sebanyak 2 sehingga kromosom yang terbentuk adalah 2 seperti yang bisa dilihat pada Tabel 3.12.

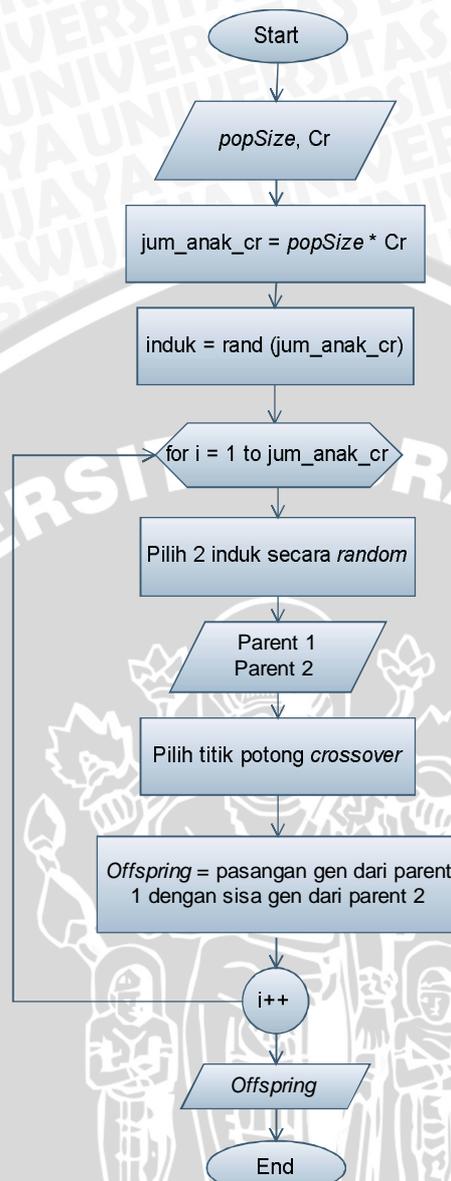
### 3.4.3 Reproduksi

Pada tahap reproduksi ini dilakukan untuk menghasilkan keturunan (*offspring*) dari beberapa induk kromosom yang dipilih secara *random*. Ada dua operator genetika yang digunakan dalam tahap ini yang terdiri *crossover* dan mutasi.

#### 3.4.3.1 Crossover

Pada tahap selanjutnya setelah pembangunan populasi awal adalah membuat populasi untuk generasi selanjutnya dengan memasuki tahap reproduksi yang diawali dengan proses *crossover*. Pada proses *crossover* ini, populasi awal diambil untuk dijadikan sebagai calon induk yang dipilih secara *random*. Untuk mengetahui berapa banyak anak yang akan dibentuk dapat dihitung berdasarkan nilai *cr* (*crossover rate*) yang telah diinisialisasikan. Metode *crossover* yang digunakan dalam penelitian ini adalah *one cut point crossover*. Pada metode ini dipilih 2 induk secara *random* yang nantinya akan menghasilkan 1 anak dari hasil *crossover* tersebut. Metode ini bekerja dengan cara memilih satu titik potong secara acak pada kromosom kemudian mendapatkan nilai dari bagian kiri *parent* 1 dengan sisa gen dari *parent* 2 untuk dijadikan *offspring*. Proses *crossover* ini bertujuan untuk menghasilkan kromosom baru dari penggabungan 2 induk. Pemilihan induk untuk melakukan *crossover* dipilih secara *random* dari populasi awal.

Flowchart algoritma yang digunakan untuk *crossover* ditampilkan pada Gambar 3.5 berikut.



**Gambar 3.5** Algoritma *Crossover*

Pada penelitian ini sudah ditentukan  $cr = 1$  maka ada  $1 \times 2 = 2$  *offspring* yang dihasilkan dari proses *crossover* dan setiap proses *crossover* menghasilkan satu anak sehingga butuh dua kali operasi *crossover*. Misalkan dipilih secara *random* yaitu P1 = individu 1 dan P2 = individu 2 yang akan dijadikan induk untuk melakukan proses *crossover*, selanjutnya yaitu membangkitkan nilai *random* untuk memilih titik potongnya. Titik potongnya = 12, maka setelah gen

ke-12 akan dipotong untuk dilakukan proses *crossover*. Untuk contoh dari proses *crossover* dapat dilihat pada Tabel 3.13 berikut.

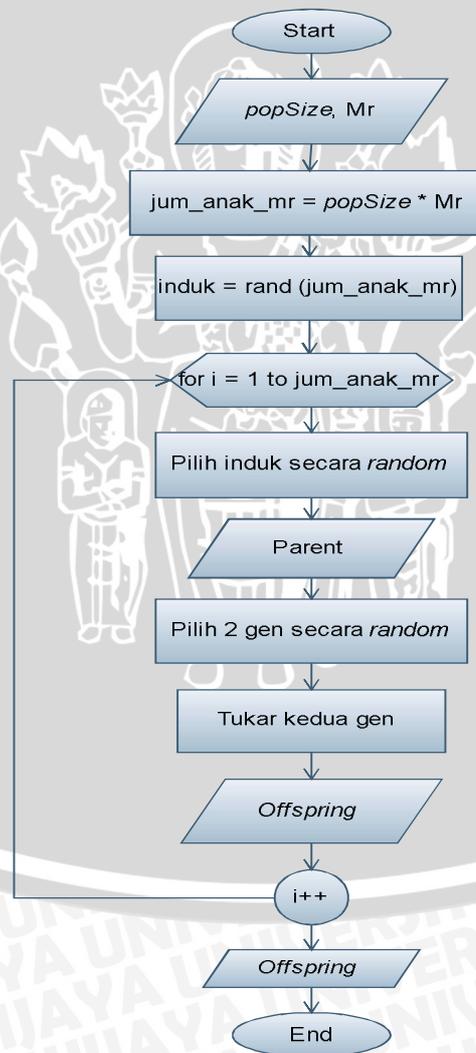
**Tabel 3.13** Proses *Crossover*

<i>Parent 1</i>	5	10	3	15	26	11	20	1	8	2	16	22	4	19	28
	12	27	18	29	6	13	25	17	7	21	9	14	23	30	24
<i>Parent 2</i>	11	6	20	7	19	1	10	22	29	16	8	23	2	9	18
	27	4	12	21	30	5	14	26	13	25	3	15	17	24	28
<i>Child 1</i>	5	10	3	15	26	11	20	1	8	2	16	22	6	7	19
	29	23	9	18	27	4	12	21	30	14	13	25	17	24	28
<i>Parent 2</i>	11	6	20	7	19	1	10	22	29	16	8	23	2	9	18
	27	4	12	21	30	5	14	26	13	25	3	15	17	24	28
<i>Parent 1</i>	5	10	3	15	26	11	20	1	8	2	16	22	4	19	28
	12	27	18	29	6	13	25	17	7	21	9	14	23	30	24
<i>Child 2</i>	11	6	20	7	19	1	10	22	29	16	8	23	2	9	18
	27	4	12	21	30	5	3	15	26	28	13	25	17	14	24

Pada Tabel 3.13 menggambarkan proses *crossover* menggunakan metode *one cut point* dengan menghasilkan 1 anak dari 2 parent. Untuk *child 1* mendapatkan nilai dari bagian kiri pada *parent 1* dan sisanya diambil dari *parent 2*. Sedangkan untuk *child 2* menggunakan *parent 2* sebagai individu 1 dan *parent 1* sebagai individu 2 dengan titik potong ke 21 sehingga dihasilkan nilai dari bagian kiri *parent 2* dan sisa gen dari *parent 1*.

### 3.4.3.2 Mutasi

Proses kedua dalam *membuat* populasi baru setelah melakukan *crossover* adalah proses mutasi. Proses mutasi adalah proses mengubah gen dari keturunan secara random. Untuk calon induknya didapatkan dari kromosom awal atau hasil dari proses sebelumnya. Dalam menentukan banyaknya anak hasil mutasi berdasarkan nilai *mr* (*mutation rate*) yang telah diinisialisasikan sebelumnya. Metode mutasi yang digunakan adalah *reciprocal exchange mutation*. Metode ini bekerja dengan memilih dua posisi secara *random* pada satu kromosom, kemudian menukar kedua posisi tersebut. Flowchart algoritma yang digunakan untuk mutasi ditampilkan pada Gambar 3.6 berikut.



Gambar 3.6 Algoritma Mutasi

Pada penelitian ini digunakan probabilitas mutasi yaitu 0,5 maka akan dilakukan 1 kali mutasi karena  $2 \times 0,5 = 1$  sehingga hanya membutuhkan 1 kali mutasi untuk menghasilkan 1 anak. Untuk contoh proses mutasi bisa dilihat pada Tabel 3.14 berikut.

**Tabel 3.14** Proses Mutasi

<i>Parent 1</i>	5	10	3	15	26	11	20	1	8	2	16	22	4	19	28
	12	27	18	29	6	13	25	17	7	21	9	14	23	30	24
<i>Child 3</i>	5	10	3	15	26	21	20	1	8	2	16	22	4	19	28
	12	27	18	29	6	13	25	17	7	11	9	14	23	30	24

Pada Tabel 3.14 merupakan proses mutasi dengan memilih *parent* secara *random* dan yang terpilih adalah *parent 1*. Pada kromosom tersebut dilakukan pemilihan titik gen yang akan ditukar secara *random* juga dan titik gen yang dipilih yaitu pada gen ke – 6 dan ke – 25. Lalu kedua gen tersebut ditukar posisinya sehingga menghasilkan *child 3* yang berisi nilai 21 pada gen ke – 6 dan nilai 11 pada gen ke – 25.

#### 3.4.4 Evaluasi dan Seleksi

Metode seleksi yang digunakan pada penelitian ini adalah *elitism selection*. Metode ini bekerja dengan cara memilih kromosom yang terbaik dari populasi sebelumnya berdasarkan nilai *fitness*-nya. Semakin besar nilai *fitness* maka solusi jadwal tersebut mempunyai peluang untuk dipilih. Flowchart algoritma yang digunakan untuk membuat kromosom selanjutnya dengan metode seleksi *elitism* ditampilkan pada Gambar 3.7 berikut.



**Gambar 3.7** Algoritma Metode Seleksi *Elitism*

Sebelum dilakukan proses seleksi maka dilakukan penggabungan hasil semua kromosom pada proses sebelumnya ditambah dengan kromosom hasil dari proses *crossover* dan mutasi kemudian dihitung nilai *fitness*-nya. Contoh nilai *fitness* dari hasil penggabungan semua kromosom yang ada ditunjukkan pada Tabel 3.15.

**Tabel 3.15** Contoh Nilai *Fitness* Semua Kromosom

Individu	Nilai <i>Fitness</i>
P1	0.0769
P2	0.0714
C1	0.0714
C2	0.0833
C3	0.0667

Tahapan selanjutnya yaitu melakukan proses seleksi *elitism* dengan mengambil kromosom yang mempunyai nilai *fitness* terbesar sebanyak *popSize* yang telah ditentukan sebelumnya. Untuk melihat hasil dari proses seleksi *elitism* ditunjukkan pada Tabel 3.16.

**Tabel 3.16** Hasil dari Proses Seleksi *Elitism*

Individu	Nilai <i>Fitness</i>
C2	0.0833
P1	0.0769

Setelah dilakukan proses seleksi maka dilanjutkan dengan pemilihan individu terbaik berdasarkan nilai *fitness* yang paling besar dari Tabel 3.16. Dari Tabel 3.16 didapatkan individu yang mempunyai nilai *fitness* terbesar yaitu C2. Untuk mengetahui solusi jadwal dari C2 yang merupakan individu terbaik pada iterasi pertama dapat dilihat pada Tabel 3.17 berikut.

**Tabel 3.17** Hasil Solusi Jadwal Individu Terbaik

Kelas	Senin								Selasa							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
7A			11	1	1	10	10	22	22	16	16	23	23	2	2	3
7B			6	19	18	18	27	27	4	4	12	12	5	5	26	26
7C			20	20	7	7	29	29	8	8	9	21	30	30	15	28

Kelas	Rabu		
	1	2	3
7A	17	24	24
7B	13	25	25
7C	28	14	14

## BAB IV

### PERANCANGAN

#### 4.1 Perancangan Database

Dalam penelitian ini menggunakan database MySQL untuk menyimpan data – data yang nantinya digunakan untuk menyelesaikan masalah penjadwalan. Struktur tabel database untuk optimasi penjadwalan mata pelajaran menggunakan algoritma genetika adalah sebagai berikut :

##### 4.1.1 Tabel Guru

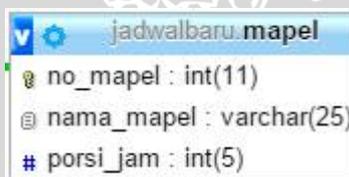
Tabel guru digunakan untuk menyimpan data guru dengan atribut yang terdiri dari no guru, nama guru, no mapel, status dan porsi total. Perancangan tabel guru dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tabel Guru

##### 4.1.2 Tabel Mapel

Tabel mapel digunakan untuk menyimpan data mata pelajaran dengan atribut yang terdiri dari no mapel, nama mapel dan porsi jam. Perancangan tabel mata pelajaran dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tabel Mapel

##### 4.1.3 Tabel Penugasan

Tabel penugasan digunakan untuk menyimpan data penugasan guru dengan atribut yang terdiri dari kode, no guru, no mapel, kelas dan porsi mapel. Perancangan tabel penugasan dapat dilihat pada Gambar 4.3.

```

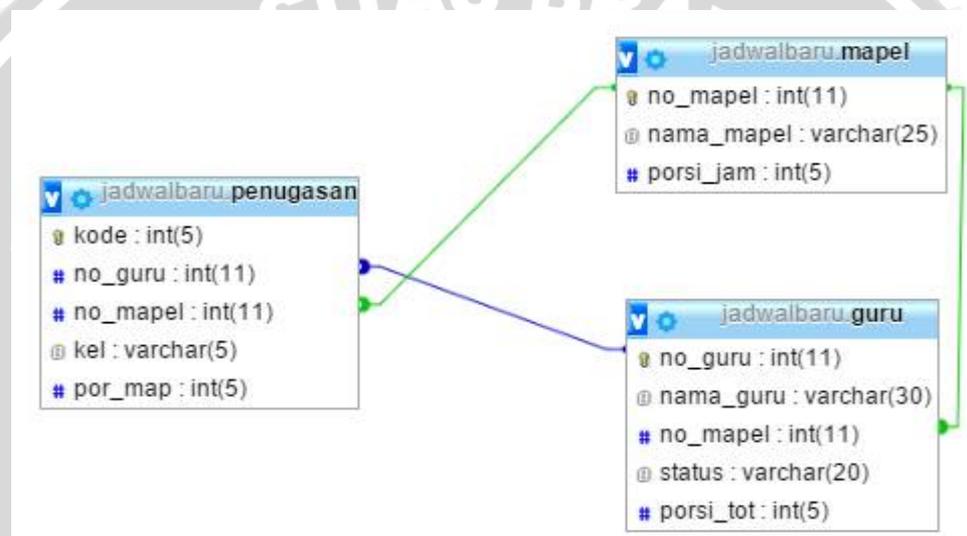
jadwalbaru.penugasan
┌───┴───┐
│ kode : int(5)          │
│ # no_guru : int(11)   │
│ # no_mapel : int(11)  │
│ @ kel : varchar(5)    │
│ @ por_map : int(5)    │
└───┴───┘

```

**Gambar 4.3** Tabel Penugasan

#### 4.1.4 Perancangan Database Keseluruhan

Relasi antar tabel-tabel database yang ada pada aplikasi penjadwalan ini dapat dilihat pada Gambar 4.4.



**Gambar 4.4** Relasi Antar Tabel Database

Gambar 4.4 merupakan relasi antar tabel di database pada aplikasi penjadwalan mata pelajaran yang menghubungkan 3 buah tabel yaitu mapel, guru dan penugasan. Pada tabel mapel dengan atribut yaitu no mapel berelasi dengan tabel guru dan penugasan serta di tabel penugasan dengan atribut no guru berelasi dengan tabel guru sehingga dari ketiga tabel tersebut saling memiliki relasi satu sama lain.

#### 4.2 Perancangan *User Interface*

Pada perancangan *user interface* dari aplikasi penjadwalan mata pelajaran ini terdiri dari beberapa halaman yaitu halaman data guru, halaman data mata pelajaran, halaman penugasan dan halaman parameter dari proses algoritma genetika.

#### 4.2.1 Rancangan *User Interface* Halaman Data Guru

**APLIKASI PENJADWALAN MATA PELAJARAN**

Data Guru	Data Mapel	Penugasan	Parameter
-----------	------------	-----------	-----------

No	Nama Guru	Nama Mata Pelajaran	Status

**Gambar 4.5** Rancangan *User Interface* Halaman Data Guru

Keterangan Gambar 4.5 :

1. Judul sistem
2. Daftar menu pada aplikasi penjadwalan yaitu data guru, data mata pelajaran, penugasan, dan parameter.
3. Tabel untuk menampilkan data guru yang berisi no, nama guru, nama mata pelajaran dan status.

#### 4.2.2 Rancangan *User Interface* Halaman Data Mapel

APLIKASI PENJADWALAN MATA PELAJARAN ①

Data Guru	Data Mapel	Penugasan	Parameter	②
-----------	------------	-----------	-----------	---

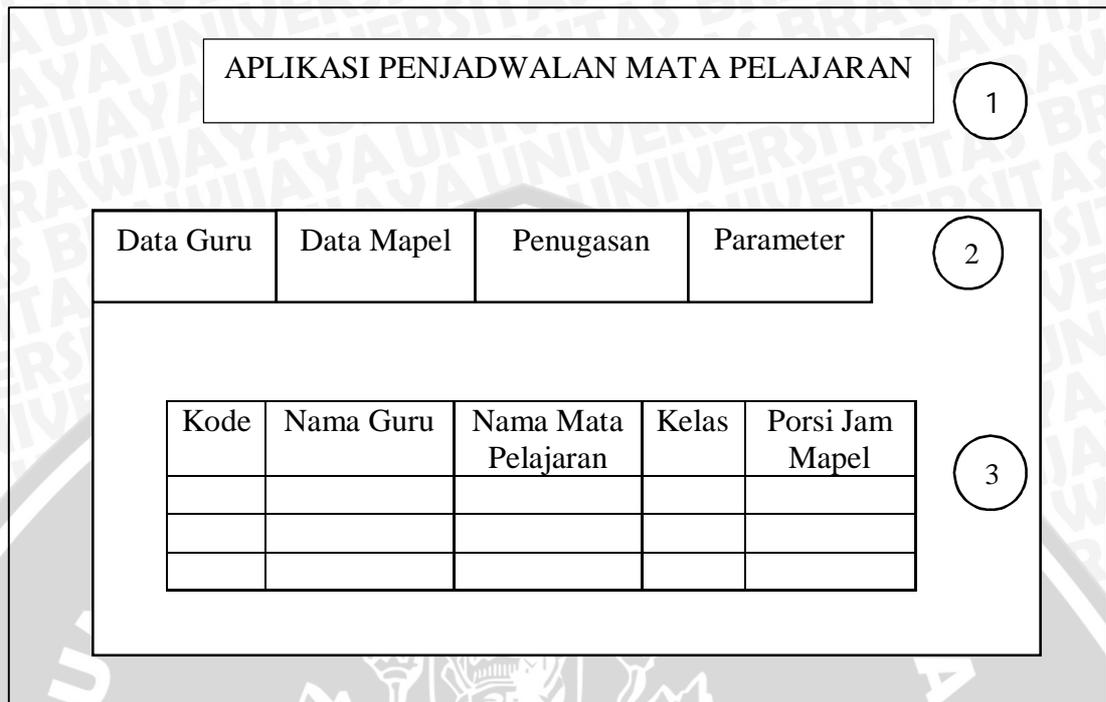
No	Nama Mata Pelajaran	Porsi Jam /Minggu	③

**Gambar 4.6** Rancangan *User Interface* Halaman Data Mapel

Keterangan Gambar 4.6 :

1. Judul sistem
2. Daftar menu pada aplikasi penjadwalan yaitu data guru, data mata pelajaran, penugasan, dan parameter.
3. Tabel untuk menampilkan data mapel yang berisi no, nama mata pelajaran dan porsi jam/ minggu.

### 4.2.3 Rancangan *User Interface* Halaman Penugasan



**Gambar 4.7** Rancangan *User Interface* Halaman Penugasan

Keterangan Gambar 4.7 :

1. Judul sistem
2. Daftar menu pada aplikasi penjadwalan yaitu data guru, data mata pelajaran, penugasan, dan parameter.
3. Tabel untuk menampilkan data penugasan jadwal yang berisi kode, nama guru, nama mata pelajaran, kelas dan porsi jam mapel.

#### 4.2.4 Rancangan *User Interface* Halaman Parameter

The user interface design for the parameter page consists of the following elements:

- 1. Title Bar:** A box containing the text "APLIKASI PENJADWALAN MATA PELAJARAN".
- 2. Menu Bar:** A horizontal bar with four menu items: "Data Guru", "Data Mapel", "Penugasan", and "Parameter".
- 3. Input Form:** A section titled "Masukkan Perhitungan" containing four input fields: "Jumlah Iterasi", "Jumlah PopSize", "Cr", and "Mr".
- 4. Action Button:** A button labeled "Proses" located below the input fields.
- 5. Results Table:** A table with the following structure:
 

Generasi		
Kromosom	Bentrok	Fitness

**Gambar 4.8** Rancangan *User Interface* Halaman Parameter

Keterangan Gambar 4.8 :

1. Judul sistem
2. Daftar menu pada aplikasi penjadwalan yaitu data guru, data mata pelajaran, penugasan, dan parameter.
3. Form input jumlah iterasi.

4. Form input jumlah populasi.
5. Form input *crossover rate* (*cr*).
6. Form input *mutation rate* (*mr*).
7. Tombol untuk memproses hasil parameter.
8. Kolom untuk menampilkan generasi.
9. Kolom untuk menampilkan kromosom.
10. Kolom untuk menampilkan bentrok.
11. Kolom untuk menampilkan nilai *fitness*.

### 4.3 Perancangan Uji Coba dan Evaluasi

Pada penelitian ini dilakukan pengujian dengan mengetahui nilai parameter algoritma genetika dalam menemukan jadwal mata pelajaran yang paling optimal, akan dilakukan beberapa percobaan antara lain:

1. Uji coba untuk menentukan banyaknya generasi atau iterasi yang optimal untuk proses algoritma genetika penjadwalan.
2. Uji coba untuk menentukan jumlah populasi (*PopSize*) yang optimal untuk proses algoritma genetika penjadwalan.
3. Uji coba untuk mencari kombinasi probabilitas mutasi dan probabilitas *crossover* untuk menghasilkan solusi yang paling optimum.

#### 4.3.1 Uji Coba Banyaknya Generasi

Pada tahap ini dilakukan uji coba terhadap program untuk mengetahui banyaknya generasi yang optimal dalam menemukan solusi jadwal yang terbaik. Dalam pengujian ini terdapat hipotesa dimana semakin tinggi jumlah generasinya maka semakin tinggi pula kemampuan algoritma genetika untuk mencari solusi terbaik. Pengujian ini bertujuan untuk mengetahui pengaruh perubahan jumlah generasi terhadap nilai *fitness* yang dihasilkan. Uji coba akan dilakukan sebanyak 10 kali dan digunakan generasi dengan kelipatan 20. Parameter yang digunakan pada uji coba generasi adalah :

- a. Jumlah iterasi / generasi = 10 - 100
- b. Jumlah populasi = 15
- c. *Cr* = 0.5

$$d. Mr = 0.5$$

Setiap pengujian akan diambil nilai *fitness* masing – masing percobaan kemudian akan dibandingkan dengan setiap nilai generasi yang lain untuk dianalisa. Tabel hasil uji coba terhadap banyaknya generasi bisa dilihat pada Tabel 4.1.

**Tabel 4.1** Rancangan Uji Coba Banyaknya Generasi

Percobaan ke-	Banyaknya Generasi									
	10	20	30	40	50	60	70	80	90	100
	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
Rata-rata <i>Fitness</i>										
Rata-rata Waktu										

#### 4.3.2 Uji Coba Jumlah Populasi

Pada tahap ini dilakukan uji coba jumlah individu dalam populasi (*popSize*) untuk menemukan solusi jadwal mata pelajaran yang paling optimum. Hipotesa pada pengujian jumlah populasi ini adalah semakin besar nilai populasi maka semakin bervariasi solusi jadwal yang ada sehingga akan menghasilkan solusi jadwal yang lebih baik dengan mengetahui nilai *fitness* yang dihasilkan. Jumlah populasi yang digunakan dengan kelipatan 10. Parameter yang digunakan pada uji coba populasi adalah :

- a. Jumlah iterasi / generasi = 15
- b. Jumlah populasi = 10 - 100

$$c. Cr = 0.5$$

$$d. Mr = 0.5$$

Tabel hasil uji coba terhadap jumlah populasi bisa dilihat pada Tabel 4.2.

**Tabel 4.2** Rancangan Uji Coba Jumlah Populasi

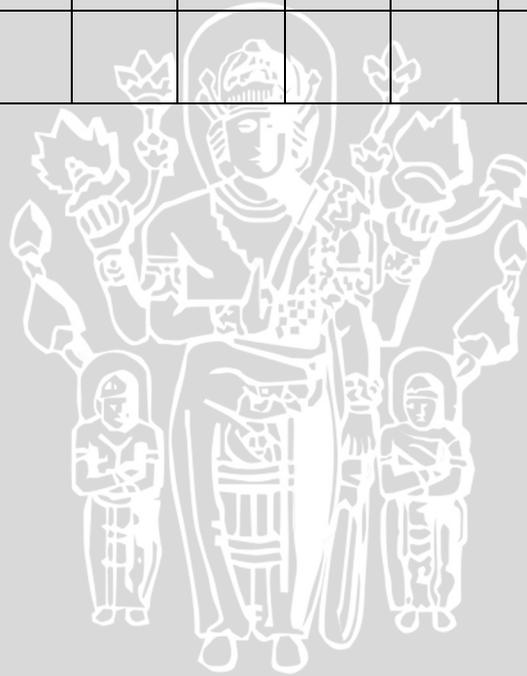
Percobaan ke-	Jumlah Populasi									
	10 <i>fitness</i>	20 <i>fitness</i>	30 <i>fitness</i>	40 <i>fitness</i>	50 <i>fitness</i>	60 <i>fitness</i>	70 <i>fitness</i>	80 <i>fitness</i>	90 <i>fitness</i>	100 <i>fitness</i>
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
Rata-rata <i>Fitness</i>										
Rata-rata Waktu										

#### 4.3.3 Uji Coba Kombinasi Probabilitas *Crossover* dan Mutasi

Pada tahap ini dilakukan uji coba terhadap nilai *crossover rate* ( $cr$ ) dan nilai *mutation rate* ( $mr$ ) yang sesuai agar menemukan solusi jadwal yang optimal. Hipotesa dari pengujian ini adalah semakin tinggi nilai  $cr$  dan  $mr$  maka semakin optimal hasil yang diperoleh. Nilai *crossover rate* dan *mutation rate* yang diuji yaitu 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. Pada penelitian ini jumlah generasi dan populasi yaitu generasi 70 dan populasi 20. Setiap pengujian dilakukan 10 kali percobaan kemudian diambil nilai *fitness*-nya. Tabel hasil uji coba terhadap probabilitas *crossover* dan mutasi bisa dilihat pada Tabel 4.3.

**Tabel 4.3** Rancangan Uji Coba Terhadap Probabilitas *Crossover* Dan Mutasi

Percobaan ke-	<i>Crossover rate : Mutation rate</i>										
	0:1	0.1:0.9	0.2:0.8	0.3:0.7	0.4:0.6	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2	0.9:0.1	1:0
	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
Rata-rata <i>Fitness</i>											
Rata – rata Waktu											



## BAB V

### IMPLEMENTASI

#### 5.1 Implementasi Basis Data

Pada sistem penjadwalan mata pelajaran ini diperlukan database dengan inputan berupa data – data yang telah diperoleh sebelumnya. Database yang digunakan yaitu database MySQL. Dalam database ini terdapat tiga tabel yaitu tabel guru, tabel mapel dan tabel penugasan.

##### 5.1.1 Tabel Guru

Tabel guru merupakan tabel yang berisi data – data guru dengan atribut yang terdiri dari no guru, nama guru, no mapel, status dan porsi total. Untuk atribut status terdapat tiga status yaitu sertifikasi, kepala sekolah dan GTT dan pada atribut porsi total berisi jumlah porsi pengajaran berdasarkan statusnya. Tipe data pada atribut tabel guru dapat dilihat pada Tabel 5.1.

**Tabel 5.1** Tipe data pada Tabel Guru

No.	Atribut	Tipe Data	Ukuran
1.	no_guru	integer	11
2.	nama_guru	varchar	30
3.	no_mapel	integer	11
4.	status	varchar	20
5.	porsi_tot	integer	5

##### 5.1.2 Tabel Mapel

Tabel mapel merupakan tabel yang berisi data – data mata pelajaran yang digunakan untuk membuat penjadwalan dengan atribut yang terdiri dari no mapel, nama mapel dan porsi jam. Untuk atribut porsi jam berisi porsi jam mata pelajaran dalam seminggunya. Tipe data pada atribut tabel mapel dapat dilihat pada Tabel 5.2.

**Tabel 5.2** Tipe data pada Tabel Mapel

No.	Atribut	Tipe Data	Ukuran
1.	no_mapel	integer	11
2.	nama_mapel	varchar	25
3.	porsi_jam	integer	5

### 5.1.3 Tabel Penugasan

Tabel penugasan merupakan tabel yang berisi kode guru hasil dari proses penugasan guru dengan memperhatikan porsi jam mengajar berdasarkan statusnya dan porsi jam mata pelajarannya serta kesediaan kelasnya. Atribut pada tabel ini terdiri dari kode, no guru, no mapel, kelas dan porsi mapel. Untuk atribut porsi mapel berisi porsi mata pelajaran yang diperoleh dari tabel mapel kemudian dipecah menjadi beberapa bagian sesuai porsinya dengan ketentuan mata pelajaran yang diberikan maksimal 2 jam untuk setiap pengajaran. Misalnya Bahasa Indonesia mempunyai porsi 6 jam seminggunya sehingga dipecah menjadi 2, 2 dan 2 dalam seminggu. Tipe data pada atribut tabel penugasan dapat dilihat pada Tabel 5.3.

**Tabel 5.3** Tipe data pada Tabel Penugasan

No.	Atribut	Tipe Data	Ukuran
1.	kode	integer	5
2.	no_guru	integer	11
3.	no_mapel	integer	11
4.	kel	varchar	5
5.	por_map	integer	5

## 5.2 Implementasi Algoritma

Berdasarkan analisa dan perancangan sistem yang terdapat pada bab sebelumnya maka pada subbab ini akan dijelaskan implementasi proses – proses tersebut ke dalam sistem dengan menggunakan bahasa pemrograman PHP dan database MySql.

### 5.2.1 Proses Penugasan

Proses penugasan ini dilakukan berdasarkan data guru dan mata pelajaran dengan memperhatikan jumlah porsi jam mengajar, porsi jam pelajaran dan kesediaan kelas yang ada. Hasil dari proses penugasan ini berupa kromosom dengan panjang 552 kromosom yang diperoleh dari jumlah porsi jam mata pelajaran yang telah dipecah menjadi 23 bagian dan jumlah kelas sebanyak 24 sehingga dihasilkan  $23 \times 24 = 552$ . Untuk jumlah porsi jam mata pelajaran

menjadi 23 bagian sesuai dengan jumlah mata pelajaran yaitu 11 mata pelajaran yang setiap mata pelajarannya dipecah porsi jam mengajarnya maksimal 2 jam.

Implementasi algoritma proses penugasan dapat dilihat pada Gambar 5.1.

```

<?php
$kode = array();
$id_guru = array();
$na_guru = array();
$na_mapel = array();
$na_kelas = array();
$jum_porsi = array();
$no_mapel = array();
$query=mysql_query("SELECT g.no_guru,g.nama_guru,
p.nama_mapel,g.status,g.porsi_tot,p.no_mapel, p.porsi_jam
FROM guru g left join mapel p on (g.no_
mapel=p.no_mapel) order by g.no_guru");
$no=1;
$kelas = array('7A','7B','7C','7D','7E','7F','7G','7H','8A',
'8B','8C','8D','8E','8F','8G','8H','9A','9B','9C','9D','9E',
'9F','9G','9H');
$mapel = array();
while ($data = mysql_fetch_array($query)){
    $nama_mapel = $data['nama_mapel'];
    if(empty($mapel[$nama_mapel])) $mapel[$nama_mapel] = 0;
    $porosi_total = $data['porosi_tot'];
    $porosi_jam = $data['porosi_jam'];
    $jum_porsi = 0;
    do{
        $porosi = $porosi_jam;
        do{
            if($porosi>2) {
                $dporosi = 2;
                $porosi = $porosi - 2;
            } else {$dporosi = $porosi;
                $porosi =0;
            }
            $kode[] = $no;
            $id_guru[] = $data['no_guru'];
            $na_guru[] = $data['nama_guru'];
            $na_mapel[] = $data['nama_mapel'];
            $no_mapel[] = $data['no_mapel'];
            if(!empty($kelas[$mapel[$nama_mapel]]))
$na_kelas[] = $kelas[$mapel[$nama_mapel]];
            else $na_kelas[] = 'no_guru';
            $jum_porsi[] = $dporosi;
            $no++;
        } while($porosi > 0);
        $mapel[$nama_mapel]++;
        $jum_porsi += $porosi_jam;
    } while ($jum_porsi < $porosi_total);
}

```

```

$query=mysql_query("SELECT * FROM mapel");
while($data = mysql_fetch_array($query))
{
    $nama_mapel = $data['nama_mapel'];
    $data_guru = array();
    $data_idguru = array();
    $data_idmapel = array();
    $porsi_jam = 0;
    $d_query=mysql_query("SELECT
g.no_guru,g.nama_guru,p.nama_mapel,g.status,g.porsi_tot, p.no
_mapel, p.porsi_jam FROM guru g, mapel p where p.nama_mapel =
'".$nama_mapel."' and g.status = 'sertifikasi' order by
g.no_guru");
    while($guru = mysql_fetch_array($d_query))
    {
        $porsi_jam = $guru['porsi_jam'];
        $data_guru[] = $guru['nama_guru'];
        $data_idguru[] = $guru['no_guru'];
        $data_idmapel[] = $guru['no_mapel'];
    }
    $i=0;
    while($mapel[$nama_mapel]<=23) {
        $porsi = $porsi_jam;
        if($i>=count($data_guru)) $i=0;
        do{
            if($porsi>2) {
                $dporsi = 2;
                $porsi = $porsi - 2;
            } else {
                $dporsi = $porsi;
                $porsi =0;
            }
            $kode[] = $no;
            $id_guru[] = $data_idguru[$i];
            $no_mapel[] = $data_idmapel[$i];
            $na_guru[] = $data_guru[$i];
            $na_mapel[] = $nama_mapel;
            $na_kelas[] = $kelas[$mapel[$nama_mapel]];
            $jm_porsi[] = $dporsi;
            $no++;
        } while($porsi > 0);
        $i++;
        $mapel[$nama_mapel]++;
    }
}
array_multisort($id_guru, SORT_ASC, $na_guru, $na_mapel, $na_kela
s, $jm_porsi, SORT_DESC);

```

```

$query_insert = " DELETE FROM penugasan";
mysql_query($query_insert);
$query_insert = '';
for($i=0; $i<count($kode); $i++){
    echo '<tr><td style="text-align:center">'.$kode[$i].'\<br
/></td>';
    echo '<td style="text-align:left">'.$na_guru[$i].'\<br
/></td>';
    echo '<td style="text-align:left">'.$na_mapel[$i].'\<br
/></td>';
    echo '<td style="text-align:left">'.$na_kelas[$i].'\<br
/></td>';
    echo '<td style="text-align:center">'.$jm_porsi[$i].'\<br
/></td></tr>';
    if($i==0) $query_insert .= " INSERT INTO penugasan
(kode, na_guru, na_mapel, kel, por_map,kode_guru) VALUES
('".$kode[$i]."', '".$na_guru[$i]."', '".$na_mapel[$i]."',
'".$na_kelas[$i]."', '".$jm_porsi[$i]."', '".$id_guru[$i]."'
";
    else $query_insert .= " ,('".$kode[$i]."',
'".$na_guru[$i]."', '".$na_mapel[$i]."', '".$na_kelas[$i]."',
'".$jm_porsi[$i]."', '".$id_guru[$i]."' ) ";
}

```

**Gambar 5.1** Source Code Proses Penugasan

### 5.2.2 Inisialisasi Populasi Awal

Proses inisialisasi populasi awal dilakukan berdasarkan data yang diperoleh dari proses penugasan yang disimpan pada tabel penugasan. Dalam membangkitkan populasi awal dengan cara mengacak kode guru yang ada pada tabel penugasan yaitu 552 kromosom. Implementasi algoritma inisialisasi populasi awal dapat dilihat pada Gambar 5.2.

```

for($k=1;$k<=$popsiz;$k++){
    $kromosom[$k] = '';
    $ceknomor=array();
    $jadwal = array();
    $kelas = array();
    $kro[$k] = '';
    for($i=0;$i<$pindividu;$i++){
        $individu=rand(1,$pindividu);
        while (in_array($individu,$ceknomor)){
            $individu=rand(1,$pindividu);
        }
        $ceknomor[$i] = $individu;
    }
}

```

```

        if($i%20 == 19) $kro[$k]=$kro[$k].$individu." <br>";
        else $kro[$k]=$kro[$k].$individu." ";
        $kromosom[$k]=$kromosom[$k].$individu." ";
        // proses perubahan kromosom menjadi sebuah jadwal dengan
        merubah kode guru menjadi id guru
        $tmp_kelas = $na_kelas[$individu-1];
        $tmp_porsi = $jm_porsi[$individu-1];
        if(empty($kelas[$tmp_kelas])) $kelas[$tmp_kelas] =0;
        for($j=0; $j<$tmp_porsi; $j++){
            $jadwal[$tmp_kelas][$kelas[$tmp_kelas]]=$individu;
            $kelas[$tmp_kelas]++;
        }
    }
}
<?php
echo "P" . $k. " = " . $kro[$k];
?>

```

**Gambar 5.2** Source Code Inisialisasi Populasi Awal

### 5.2.3 Perhitungan Fitness

Proses perhitungan nilai fitness ini dimulai dengan menghitung nilai pinalti atau nilai bentrok pada setiap solusi jadwal pelajaran. Nilai bentrok diperoleh jika terdapat satu guru yang mengajar pada kelas yang berbeda dengan waktu yang bersamaan maka akan dikenakan pinalti sebanyak 1. Untuk nilai *fitness* dihitung dengan Persamaan 2-1. Implementasi algoritma perhitungan *fitness* dapat dilihat pada Gambar 5.3.

```

<?php
$jadwal_ajar[] = $jadwal;
//hitung bentrok
$bentrok[$k] = 0;
for($i=0; $i<$kelas[$kls[0]]; $i++){
    $in =1
    $kode_guru = array();
    $tmp_btr =0;
    for($j=0; $j<count($kls); $j++){
        $individu = $jadwal[$kls[$j]][$i];
        $tmp_idguru = $na_guru[$individu-1];
        if(array_search($tmp_idguru,$kode_guru)>0) {$tmp_btr++;
    }
    else {$kode_guru[$in] = $tmp_idguru; $in ++;
    }
    $bentrok[$k]+=$tmp_btr;
}
$bentrok[$k];
$fitness[$k] = 1/(1+$bentrok[$k]);
echo $bentrok[$k];
$data_fit[$k] = round($fitness[$k],4);
echo $data_fit[$k];

```

**Gambar 5.3** Source Code Perhitungan Fitness

#### 5.2.4 Proses Crossover

Metode *crossover* yang digunakan pada penjadwalan ini adalah *one – cut point*. Tahapan dalam proses ini diawali dengan menentukan jumlah anak yang akan dihasilkan kemudian memilih dua *parent* secara *random*. Tahap selanjutnya yaitu memilih titik potong secara *random* pada *parent* 1 sehingga didapatkan gen dari *parent* 1 dan sisanya didapatkan dari *parent* 2. Pada proses ini diperoleh 1 anak setiap tahapannya. Implementasi algoritma proses *crossover* dapat dilihat pada Gambar 5.4.

```
<?php
//crossover
$jum_cr = ceil($cr * $popsize); //pembulatan ke atas
for($k=$popsize+1; $k<=$popsize+$jum_cr; $k++){
    $kromosom[$k]="";
    $ceknomor=array();
    $jadwal = array();
    $kelas = array();
    $parent_1 = rand(1, $popsize);
    $parent_2 = rand(1, $popsize);
    do{
        $parent_2 = rand(1,$popsize);
    }
    while($parent_1 == $parent_2);

    $titik = rand(1,$pindividu-1);
    $p1 = explode(" ", $kromosom[$parent_1]);
    $p2 = explode(" ", $kromosom[$parent_2]);

    $d=0;
    $kro[$k]='';
    for($i=0;$i<$pindividu;$i++){
        if($i<$titik){
            $schild1=$p1[$i];
        } else {
            $cek = 0;
        }
    }
}
```

```

do{
    if(array_search($p2[$d],$ceknomor)>0){}
    else {
        $child1=$p2[$d];
        $cek++;
    }
    $d++;
} while($cek==0);
}

$ceknomor[$i+1]=$child1;
if($i%20 == 19) $kro[$k]=$kro[$k].$child1." <br>"; else
$kro[$k]=$kro[$k].$child1." ";
$kromosom[$k] = $kromosom[$k].$child1." ";
echo "C".($k - $popsiz). " = ".$kro[$k]. "----
".$titik."P".$parent_1.$parent_2;
?>

```

**Gambar 5.4** Source Code Proses Crossover

### 5.2.5 Proses Mutasi

Metode mutasi yang digunakan adalah *reciprocal exchange mutation*. Proses ini juga diawali dengan menentukan jumlah anak berdasarkan inputan *mr* dan jumlah populasi kemudian memilih satu *parent* secara *random*. Selanjutnya memilih dua nilai gen pada satu populasi secara *random* dan ditukar posisinya. Implementasi algoritma proses mutasi dapat dilihat pada Gambar 5.5.

```

<?php
//mutasi
$jum_mr = ceil($mr * $popsiz); //pembulatan keatas
for($k=$popsiz+$jum_cr+1; $k<=$jum_cr+$popsiz+$jum_mr;
$k++) {
    $ceknomor=array();
    $jadwal = array();
    $kelas = array();

    $parent = rand(1, $popsiz);
    $xp1 = rand(0,$pindividu-1);
    $xp2 = rand(0,$pindividu-1);
    do {
        $xp2 = rand(0,$pindividu-1);
    }while($xp2 ==$xp1);

    $p = explode(" ",$kromosom[$parent]);
    $ceknomor=array();
    $i=1;
    $kromosom[$k]="";

```

```

for($d=0;$d<$pindividu;$d++){
    $child[$d]=$p[$d];
}

$child[$xp1]=$p[$xp2];
$child[$xp2]=$p[$xp1];
$kro[$k] = '';
for($d=0;$d<$pindividu;$d++){
    $ceknomor[$i]=$child[$d];
    $i++;
    if($d%20 == 19) $kro[$k]=$kro[$k].$child[$d]."  
<br>"; else $kro[$k]=$kro[$k].$child[$d]." ";
    $kromosom[$k]=$kromosom[$k]. $child[$d]." ";

echo "C".($k - $popsiize). " = ".$kro[$k]."----".($xp1+1).",
".($xp2+1);
?>

```

**Gambar 5.5** Source Code Proses Mutasi

### 5.2.6 Proses Seleksi *Elitism*

Metode seleksi yang digunakan pada penjadwalan mata pelajaran ini adalah seleksi *elitism*. Pada metode ini dilakukan proses pengurutan nilai *fitness* dari yang terbesar berdasarkan hasil penggabungan kromosom populasi, anak *crossover* dan anak mutasi. Setelah itu dipilih nilai *fitness* terbesar untuk dijadikan sebagai generasi berikutnya sesuai dengan jumlah populasi awal. Implementasi algoritma seleksi dapat dilihat pada Gambar 5.6.

```

<?php
//seleksi elitism
$gabung = $jum_cr+$jum_mr+$popsiize;
for ($k=1; $k <=$gabung; $k++) {
    for ($j=1; $j<=$gabung-$k; $j++) {
        if ($data_fit[$j] < $data_fit[$j+1]) {
            $temp = $data_fit[$j];
            $data_fit[$j] = $data_fit[$j+1];
            $data_fit[$j+1] = $temp;
            $temp2 = $kromosom[$j];
            $kromosom[$j] = $kromosom[$j+1];
            $kromosom[$j+1]=$temp2;
            $temp3 = $bentrok[$j];
            $bentrok[$j] = $bentrok[$j+1];
            $bentrok[$j+1]=$temp3;
        }
    }
}
?>

```

**Gambar 5.6** Source Code Proses Seleksi *Elitism*

### 5.2.7 Proses Pemilihan Kromosom Terbaik

Proses ini dilakukan dengan mengambil kromosom terbaik yang berasal dari generasi terakhir yang mempunyai nilai *fitness* terbesar. Implementasi algoritma proses pemilihan kromosom terbaik dapat dilihat pada Gambar 5.7.

```
<?php
//solusi
if ($h = $iterasi){
    $gabung = $popsize[$h]+$jum_cr[$h]+$jum_mr[$h];
    for ($k=1; $k <=$gabung; $k++) {
        for ($j=1; $j<=$gabung-$k; $j++) {
            if ($data_fit[$j] < $data_fit[$j+1]) {
                $temp = $data_fit[$j];
                $data_fit[$j] = $data_fit[$j+1];
                $data_fit[$j+1] = $temp;
                $temp2 = $kromosom[$j];
                $kromosom[$j] = $kromosom[$j+1];
                $kromosom[$j+1]=$temp2;
                $temp3 = $bentrok[$j];
                $bentrok[$j] = $bentrok[$j+1];
                $bentrok[$j+1]=$temp3;
                $temp4 = $kro[$j];
                $kro[$j] = $kro[$j+1];
                $kro[$j+1]=$temp4;
            }
        }
    }
    echo "<center><h4>Hasil Kromosom Terbaik</h4></center>
    ".$kromosom[$k]."<br>". "<center><h4>Fitness</h4><b>". $data_
    fit[$k]."</b></center><br>";
}
?>
```

**Gambar 5.7** Source Code Proses Pemilihan Kromosom Terbaik

## 5.3 Implementasi User Interface

Halaman *user interface* pada penjadwalan mata pelajaran ini terdapat 4 halaman yaitu halaman data guru, halaman data mata pelajaran, halaman data penugasan, halaman proses genetika dan halaman hasil penjadwalan.

### 5.3.1 Halaman Data Guru

Halaman data guru berisikan no, nama guru, nama mata pelajaran dan statusnya. Implementasi *user interface* halaman data guru dapat dilihat pada Gambar 5.8.

**APLIKASI PENJADWALAN MATA PELAJARAN**

Data Guru    Data Mapel    Penugasan    Parameter

**Data Guru**

No	Nama Guru	Nama Mata Pelajaran	Status
1	Bahliar Mashudi, S.Pd	Matematika	Kepala Sekolah
2	Sri Hastuti, S.Pd	IPA	Sertifikasi
3	Hj.Luluk Mutikhatin, M.Pd	Pendidikan Agama	Sertifikasi
4	Drs.Mustain	Penjas	Sertifikasi
5	Siti Kareni, S.Pd	Bahasa Indonesia	Sertifikasi
6	Dra.Eveline, M.Pd	Bahasa Indonesia	Sertifikasi
7	Setyo Rakhmad, S.Pd	Matematika	Sertifikasi
8	Melneni Prihartini, S.Pd	IPA	Sertifikasi
9	Hj. Rahma Murwani, S.Pd	Bahasa Indonesia	Sertifikasi
10	Rohimah, S.Pd	IPA	Sertifikasi
11	Dian Estusih Wilujeng, S.Pd	Pkn	Sertifikasi
12	H. Makudi, S.Ag	Pendidikan Agama	Sertifikasi
13	Warnadi, S.Pd	IPS	Sertifikasi
14	Andik Susanto, S.Pd	Bahasa Jawa	Sertifikasi
15	Erni Puji Lestari, S.Pd	IPS	Sertifikasi
16	H.Purnomo, S.Pd	Penjas	Sertifikasi
17	Hj.Sumartiyah, S.Pd	Matematika	Sertifikasi
18	Ummu Hanik, S.Pd	IPS	Sertifikasi
19	Betty Endang Pramuni, S.Pd	IPA	Sertifikasi
20	Nurbudi Hariono	Seni Budaya	Sertifikasi
21	Yayuk Punwantini, S.Pd	Matematika	Sertifikasi
22	Ilyas, S.Pd	Bahasa Indonesia	Sertifikasi
23	Setyo Purwantini, S.Pd	Matematika	Sertifikasi
24	Dewi Nenitriana, S.Pd	Bahasa Inggris	Sertifikasi
25	Iwan Winardi, S.Pd	Penjas	Sertifikasi
26	Kasminingsih, S.Pd	Bahasa Inggris	Sertifikasi
27	Kuwat Setyobudi, S.Pd	Bahasa Inggris	Sertifikasi
28	Retno Anggraeni, S.Pd	Bahasa Inggris	Sertifikasi
29	Maria Ulfah, S.Pd	Bahasa Indonesia	Sertifikasi
30	Tutik Herawati, S.Pd	Matematika	Sertifikasi
31	Tutik Herawati, S.Pd	IPA	Sertifikasi
32	Milshan	IPS	Sertifikasi
33	Trisla Santi Parlina, S.Pd	Seni Budaya	Sertifikasi
34	Sopli, S.Pd	Bahasa Jawa	Sertifikasi
35	Oktriyana Riko V, S.Pd	Prakarya	Sertifikasi
36	Masripah Farikah, S.Pd	Pkn	Sertifikasi
37	Nur Farida, S.Pd	Pendidikan Agama	GTT
38	Andriyati, S.Kom	Prakarya	GTT
39	Arismojo Tri Utomo	Pkn	GTT
40	Arismojo Tri Utomo	Seni Budaya	GTT

**Gambar 5.8** Implementasi *User Interface* Halaman Data Guru

### 5.3.2 Halaman Data Mata Pelajaran

Halaman data mata pelajaran berisi no, nama mata pelajaran dan porsi jam/minggu. Implementasi *user interface* halaman data mapel dapat dilihat pada Gambar 5.9.

APLIKASI PENJADWALAN MATA  
PELAJARAN

Data Guru

Data Mapel

Penugasan

Parameter

## Data Mata Pelajaran

No	Nama Mata Pelajaran	Porsi Jam/Minggu
1	Pendidikan Agama	3
2	PKn	3
3	Bahasa Indonesia	6
4	Matematika	5
5	IPA	5
6	IPS	4
7	Bahasa Inggris	4
8	Seni Budaya	3
9	Penjas	3
10	Prakarya	2
11	Bahasa Jawa	2

**Gambar 5.9** Implementasi *User Interface* Halaman Data Mapel

### 5.3.3 Halaman Data Penugasan

Halaman data penugasan ini berisi kode, nama guru, nama mata pelajaran, kelas dan porsi jam mata pelajaran. Pada halaman ini sudah ditentukan setiap guru akan mengajar pada beberapa kelas yang ada dengan memperhatikan jumlah porsi jam mata pelajaran dan porsi jam mengajar berdasarkan status. Data penugasan ini tersimpan pada tabel penugasan di dalam database MySQL. Implementasi *user interface* halaman data mapel dapat dilihat pada Gambar 5.10.

## APLIKASI PENJADWALAN MATA PELAJARAN

Data Guru    Data Mapel    **Penugasan**    Parameter

Kode	Nama Guru	Nama Mata Pelajaran	Kelas	Porasi Jam Mapel
1	Bahliar Mashudi, S.Pd	Matematika	7A	2
2	Bahliar Mashudi, S.Pd	Matematika	7A	2
3	Bahliar Mashudi, S.Pd	Matematika	7A	1
4	Bahliar Mashudi, S.Pd	Matematika	7B	2
5	Bahliar Mashudi, S.Pd	Matematika	7B	2
6	Bahliar Mashudi, S.Pd	Matematika	7B	1
7	Sri Hastuti, S.Pd.	IPA	7A	2
8	Sri Hastuti, S.Pd.	IPA	7A	2
9	Sri Hastuti, S.Pd.	IPA	7A	1
10	Sri Hastuti, S.Pd.	IPA	7B	2
11	Sri Hastuti, S.Pd.	IPA	7B	2
12	Sri Hastuti, S.Pd.	IPA	7B	1
13	Sri Hastuti, S.Pd.	IPA	7C	2
14	Sri Hastuti, S.Pd.	IPA	7C	2
15	Sri Hastuti, S.Pd.	IPA	7C	1
16	Sri Hastuti, S.Pd.	IPA	7D	2
17	Sri Hastuti, S.Pd.	IPA	7D	2
18	Sri Hastuti, S.Pd.	IPA	7D	1
19	Sri Hastuti, S.Pd.	IPA	7E	2
20	Sri Hastuti, S.Pd.	IPA	7E	2
21	Sri Hastuti, S.Pd.	IPA	7E	1
22	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7A	2
23	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7A	1
24	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7B	2
25	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7B	1
26	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7C	2
27	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7C	1
28	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7D	2
29	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7D	1
30	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7E	2
31	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7E	1
32	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7F	2
33	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7F	1
34	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7G	2
35	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7G	1
36	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7H	2
37	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	7H	1
38	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	9H	2
39	Hj.Luluk Muflikhatin, M Pdi	Pendidikan Agama	9H	1
40	Drs.Mustain	Penjas	7A	2
41	Drs.Mustain	Penjas	7A	1
42	Drs.Mustain	Penjas	7B	2
43	Drs.Mustain	Penjas	7B	1
44	Drs.Mustain	Penjas	7C	2
45	Drs.Mustain	Penjas	7C	1
46	Drs.Mustain	Penjas	7D	2
47	Drs.Mustain	Penjas	7D	1
48	Drs.Mustain	Penjas	7E	2
49	Drs.Mustain	Penjas	7E	1
50	Drs.Mustain	Penjas	7F	2
51	Drs.Mustain	Penjas	7F	1
52	Drs.Mustain	Penjas	7G	2
53	Drs.Mustain	Penjas	7G	1
54	Drs.Mustain	Penjas	7H	2
55	Drs.Mustain	Penjas	7H	1

**Gambar 5.10** Implementasi *User Interface* Halaman Data Penugasan

### 5.3.4 Halaman Proses Genetika

Pada halaman proses genetika ini terdapat form yang digunakan untuk menginputkan nilai parameter genetika. Lalu *user* diminta untuk menginputkan jumlah iterasi, jumlah populasi, *cr* dan *mr*. Selanjutnya terdapat tombol proses

untuk melakukan proses genetika. Implementasi *user interface* halaman data mapel dapat dilihat pada Gambar 5.11.

The screenshot shows the 'Masukkan Perhitungan' (Enter Calculation) interface of the 'APLIKASI PENJADWALAN MATA PELAJARAN' (Subject Scheduling Application). It features a navigation bar with 'Data Guru', 'Data Mapel', 'Penugasan', and 'Parameter'. The main area contains input fields for 'Jumlah Oras', 'Jumlah PopBaz', 'Cr', and 'Mlr', along with a 'Proses' button. Below this is a table for 'Generasi 0' with columns 'Kromosom', 'Bentrok', and 'Fitness'. The table displays a large volume of numerical data. Underneath the table is the 'Hasil Kromosom Terbaik' (Best Chromosome Result) section, which shows a single line of numerical data and a 'Fitness' value of 0.0047. At the bottom, there is a 'Lihat Jadwal' (View Schedule) button and a note indicating the page loading time: 'Waktu menampilkan halaman 0.21 seconds'.

**Gambar 5.11** Implementasi *User Interface* Halaman Proses Genetika

### 5.3.5 Halaman Hasil Penjadwalan

Halaman hasil penjadwalan berisi hasil penjadwalan dari hasil kromosom yang terbaik pada generai terakhir. Implementasi *user interface* halaman hasil penjadwalan dapat dilihat pada Gambar 5.12.

Hasil Penjadwalan Mata Pelajaran

Jumlah bentrok: 214

Hari	7A	7B	7C	7D	7E	7F	7G	7H
Semua								
07.00-07.40	-	-	-	-	-	-	-	-
07.40-08.20	-	-	-	-	-	-	-	-
08.20-09.00	Sri Hastuti, S.Pd	Hj.Luluk.Muflikhatun.MPd	Andik.Susanto.S.Pd	Warnadi.S.Pd	Dra.Eveline.MPd	Dra.Eveline.MPd	Drs.Mustain	Andik.Susanto.S.Pd
09.20-10.00	Sri Hastuti, S.Pd	Warnadi.S.Pd	Andik.Susanto.S.Pd	Warnadi.S.Pd	Dra.Eveline.MPd	Dra.Eveline.MPd	Drs.Mustain	Andik.Susanto.S.Pd
10.00-10.40	Bahzar.Mashudi.S.Pd	Warnadi.S.Pd	Dian.Estusih.Wilijeng.S.	Andik.Susanto.S.Pd	Sri.Hastuti.S.Pd	Dewi.Neutriana.S.Pd	Drs.Mustain	Dra.Eveline.MPd
10.40-11.20	Hj.Luluk.Muflikhatun.MPd	Hj.Luluk.Muflikhatun.MPd	Dian.Estusih.Wilijeng.S.	Andik.Susanto.S.Pd	Sri.Hastuti.S.Pd	Dewi.Neutriana.S.Pd	Oltriyana.Racke.V.S.Pd	Dra.Eveline.MPd
11.40-12.20	Hj.Luluk.Muflikhatun.MPd	Hj.Luluk.Muflikhatun.MPd	Sri.Hastuti.S.Pd	Setyo.Rahmad.S.Pd	Sri.Hastuti.S.Pd	Drs.Mustain	Oltriyana.Racke.V.S.Pd	Hj.Luluk.Muflikhatun.MPd
12.20-13.00	Sri.Hastuti.S.Pd	Dewi.Neutriana.S.Pd	Warnadi.S.Pd	Setyo.Rahmad.S.Pd	Setyo.Rahmad.S.Pd	Dra.Eveline.MPd	Dra.Eveline.MPd	Dian.Estusih.Wilijeng.S.
Selasa								
07.00-07.40	Bahzar.Mashudi.S.Pd	Dewi.Neutriana.S.Pd	Warnadi.S.Pd	Nurbudi.Hariono	Setyo.Rahmad.S.Pd	Dra.Eveline.MPd	Dra.Eveline.MPd	Dian.Estusih.Wilijeng.S.
07.40-08.20	Bahzar.Mashudi.S.Pd	Dian.Estusih.Wilijeng.S.	Setyo.Rahmad.S.Pd	Nurbudi.Hariono	Setyo.Rahmad.S.Pd	Setyo.Rahmad.S.Pd	Setyo.Rahmad.S.Pd	Memeni.Prihartim.S.Pd
08.20-09.00	Bahzar.Mashudi.S.Pd	Dian.Estusih.Wilijeng.S.	Dian.Estusih.Wilijeng.S.	Dian.Estusih.Wilijeng.S.	Dewi.Neutriana.S.Pd	Setyo.Rahmad.S.Pd	Nurbudi.Hariono	Memeni.Prihartim.S.Pd
09.20-10.00	Bahzar.Mashudi.S.Pd	Sri.Hastuti.S.Pd	Dewi.Neutriana.S.Pd	Dian.Estusih.Wilijeng.S.	Dewi.Neutriana.S.Pd	Nurbudi.Hariono	Nurbudi.Hariono	Memeni.Prihartim.S.Pd
10.00-10.40	Siti.Karen.S.Pd	Sri.Hastuti.S.Pd	Dewi.Neutriana.S.Pd	Dian.Estusih.Wilijeng.S.	Oltriyana.Racke.V.S.Pd	Nurbudi.Hariono	Erni.Puji.Lestari.S.Pd	Kasmunguh.S.Pd
10.40-11.20	Siti.Karen.S.Pd	Sri.Hastuti.S.Pd	Nurbudi.Hariono	Sri.Hastuti.S.Pd	Oltriyana.Racke.V.S.Pd	Nurbudi.Hariono	Erni.Puji.Lestari.S.Pd	Kasmunguh.S.Pd
11.40-12.20	Nurbudi.Hariono	Drs.Mustain	Hj.Luluk.Muflikhatun.MPd	Hj.Luluk.Muflikhatun.MPd	Dra.Eveline.MPd	Dra.Eveline.MPd	Dra.Eveline.MPd	Hj.Sumarthyah.S.Pd
12.20-13.00	Nurbudi.Hariono	Drs.Mustain	Siti.Karen.S.Pd	Dra.Eveline.MPd	Dra.Eveline.MPd	Dra.Eveline.MPd	Dra.Eveline.MPd	Hj.Sumarthyah.S.Pd
Rabu								
07.00-07.40	Nurbudi.Hariono	Drs.Mustain	Siti.Karen.S.Pd	Siti.Karen.S.Pd	Warnadi.S.Pd	Dra.Eveline.MPd	Dian.Estusih.Wilijeng.S.	Dian.Estusih.Wilijeng.S.
07.40-08.20	Dewi.Neutriana.S.Pd	Bahzar.Mashudi.S.Pd	Nurbudi.Hariono	Siti.Karen.S.Pd	Warnadi.S.Pd	Dra.Eveline.MPd	Dian.Estusih.Wilijeng.S.	Erni.Puji.Lestari.S.Pd
08.20-09.00	Dewi.Neutriana.S.Pd	Bahzar.Mashudi.S.Pd	Nurbudi.Hariono	Nurbudi.Hariono	Nurbudi.Hariono	Setyo.Rahmad.S.Pd	Dian.Estusih.Wilijeng.S.	Erni.Puji.Lestari.S.Pd
09.20-10.00	Drs.Mustain	Siti.Karen.S.Pd	Sri.Hastuti.S.Pd	Warnadi.S.Pd	Dra.Eveline.MPd	Setyo.Rahmad.S.Pd	Hj.Luluk.Muflikhatun.MPd	Dra.Eveline.MPd
10.00-10.40	Siti.Karen.S.Pd	Siti.Karen.S.Pd	Sri.Hastuti.S.Pd	Hj.Luluk.Muflikhatun.MPd	Setyo.Rahmad.S.Pd	Memeni.Prihartim.S.Pd	Hj.Luluk.Muflikhatun.MPd	Dra.Eveline.MPd
10.40-11.20	Oltriyana.Racke.V.S.Pd	Siti.Karen.S.Pd	Setyo.Rahmad.S.Pd	Hj.Luluk.Muflikhatun.MPd	Setyo.Rahmad.S.Pd	Warnadi.S.Pd	Nurbudi.Hariono	Erni.Puji.Lestari.S.Pd
11.40-12.20	Oltriyana.Racke.V.S.Pd	Siti.Karen.S.Pd	Setyo.Rahmad.S.Pd	Siti.Karen.S.Pd	Dian.Estusih.Wilijeng.S.	Warnadi.S.Pd	Hj.Luluk.Muflikhatun.MPd	Erni.Puji.Lestari.S.Pd
12.20-13.00	Drs.Mustain	Dian.Estusih.Wilijeng.S.	Dewi.Neutriana.S.Pd	Siti.Karen.S.Pd	Dewi.Neutriana.S.Pd	Oltriyana.Racke.V.S.Pd	Memeni.Prihartim.S.Pd	Oltriyana.Racke.V.S.Pd
Kamis								
07.00-07.40	Hj.Luluk.Muflikhatun.MPd	Siti.Karen.S.Pd	Dewi.Neutriana.S.Pd	Sri.Hastuti.S.Pd	Dian.Estusih.Wilijeng.S.	Oltriyana.Racke.V.S.Pd	Memeni.Prihartim.S.Pd	Oltriyana.Racke.V.S.Pd
07.40-08.20	Andik.Susanto.S.Pd	Siti.Karen.S.Pd	Siti.Karen.S.Pd	Sri.Hastuti.S.Pd	Dra.Eveline.MPd	Setyo.Rahmad.S.Pd	Hj.Luluk.Muflikhatun.MPd	Hj.Luluk.Muflikhatun.MPd
08.20-09.00	Andik.Susanto.S.Pd	Dewi.Neutriana.S.Pd	Warnadi.S.Pd	Setyo.Rahmad.S.Pd	Dra.Eveline.MPd	Dian.Estusih.Wilijeng.S.	Setyo.Rahmad.S.Pd	Hj.Luluk.Muflikhatun.MPd
09.20-10.00	Dian.Estusih.Wilijeng.S.	Dewi.Neutriana.S.Pd	Sri.Hastuti.S.Pd	Hj.Luluk.Muflikhatun.MPd	Andik.Susanto.S.Pd	Dian.Estusih.Wilijeng.S.	Nurbudi.Hariono	Nurbudi.Hariono
10.00-10.40	Warnadi.S.Pd	Nurbudi.Hariono	Sri.Hastuti.S.Pd	Dewi.Neutriana.S.Pd	Andik.Susanto.S.Pd	Andik.Susanto.S.Pd	Setyo.Rahmad.S.Pd	Nurbudi.Hariono
10.40-11.20	Warnadi.S.Pd	Nurbudi.Hariono	Dian.Estusih.Wilijeng.S.	Dewi.Neutriana.S.Pd	Dian.Estusih.Wilijeng.S.	Memeni.Prihartim.S.Pd	Kasmunguh.S.Pd	Kasmunguh.S.Pd
11.40-12.20	Sri.Hastuti.S.Pd	Andik.Susanto.S.Pd	Siti.Karen.S.Pd	Oltriyana.Racke.V.S.Pd	Dian.Estusih.Wilijeng.S.	Warnadi.S.Pd	Memeni.Prihartim.S.Pd	Kasmunguh.S.Pd
12.20-13.00	Sri.Hastuti.S.Pd	Andik.Susanto.S.Pd	Siti.Karen.S.Pd	Oltriyana.Racke.V.S.Pd	Dra.Eveline.MPd	Warnadi.S.Pd	Erni.Puji.Lestari.S.Pd	Hj.Sumarthyah.S.Pd
Jumat								
07.00-07.40	-	-	-	-	-	-	-	-
07.40-08.20	Siti.Karen.S.Pd	Bahzar.Mashudi.S.Pd	Siti.Karen.S.Pd	Siti.Karen.S.Pd	Dra.Eveline.MPd	Dian.Estusih.Wilijeng.S.	Erni.Puji.Lestari.S.Pd	Hj.Sumarthyah.S.Pd
08.20-09.00	Siti.Karen.S.Pd	Bahzar.Mashudi.S.Pd	Siti.Karen.S.Pd	Siti.Karen.S.Pd	Warnadi.S.Pd	Hj.Luluk.Muflikhatun.MPd	Kasmunguh.S.Pd	Hj.Sumarthyah.S.Pd
09.20-10.00	Siti.Karen.S.Pd	Bahzar.Mashudi.S.Pd	Hj.Luluk.Muflikhatun.MPd	Dewi.Neutriana.S.Pd	Warnadi.S.Pd	Hj.Luluk.Muflikhatun.MPd	Kasmunguh.S.Pd	Nurbudi.Hariono
10.00-10.40	Siti.Karen.S.Pd	Oltriyana.Racke.V.S.Pd	Hj.Luluk.Muflikhatun.MPd	Dewi.Neutriana.S.Pd	Hj.Luluk.Muflikhatun.MPd	Memeni.Prihartim.S.Pd	Memeni.Prihartim.S.Pd	Dra.Eveline.MPd
10.40-11.20	-	-	-	-	-	-	-	-
11.40-12.20	-	-	-	-	-	-	-	-
12.20-13.00	-	-	-	-	-	-	-	-

Gambar 5.12 Implementasi User Interface Halaman Hasil Penjadwalan

Gambar 5.12 merupakan halaman hasil penjadwalan dimana halaman tersebut menampilkan hari, jam, ruang kelas dan nama guru yang mengajar. Pada halaman tersebut terdapat tanda merah yang menunjukkan bahwa hasil penjadwalan tersebut masih terdapat bentrok dengan jumlah bentrok yang ditampilkan di atas tabel penjadwalan.

## BAB VI

### PENGUJIAN DAN ANALISIS

Pada bab ini akan membahas mengenai hasil pengujian yang telah dilakukan terhadap sistem penjadwalan mata pelajaran menggunakan algoritma genetika seperti yang telah dirancang sebelumnya. Sebelum menjelaskan tentang hasil pengujian maka terlebih dahulu akan dijelaskan contoh dari hasil penjadwalan yang menggunakan data sebagian pada sistem penjadwalan ini. Data sebagian ini digunakan untuk mempermudah melihat hasil penjadwalan jika penjadwalan tersebut tidak mempunyai bentrok sedangkan jika menggunakan data secara keseluruhan maka untuk menemukan hasil penjadwalan yang optimal sangatlah tidak mudah sehingga dengan adanya data sebagian ini dapat juga membuktikan bahwa aplikasi ini dapat digunakan secara optimal. Dan untuk hasil pengujian yang dilakukan terhadap parameter jumlah generasi, populasi, *crossover rate* (*cr*) dan *mutation rate* (*mr*) akan dijelaskan masing - masing pada bab berikut.

#### 6.1 Contoh Hasil Penjadwalan Data Sebagian

Pada sistem penjadwalan ini diperlukan data sebagian yang nantinya juga diolah menggunakan algoritma genetika untuk mendapatkan hasil penjadwalan namun hasil penjadwalan tersebut tidak seperti yang didapatkan menggunakan data keseluruhan dari penjadwalan mata pelajaran. Data sebagian diperoleh dari Tabel 3.6 dan menghasilkan tabel penugasan seperti pada Tabel 3.7 dengan panjang kromosom yaitu 30 kromosom. Untuk lebih jelasnya berikut merupakan hasil dari tabel penugasan yang dapat dilihat pada Gambar 6.1.

## APLIKASI PENJADWALAN MATA PELAJARAN

Kode	Nama Guru	Nama Mata Pelajaran	Kelas	Porsi Jam Mapel
1	Sri Hastuti, S.Pd.	IPA	7A	2
2	Sri Hastuti, S.Pd.	IPA	7A	2
3	Sri Hastuti, S.Pd.	IPA	7A	1
4	Sri Hastuti, S.Pd.	IPA	7B	2
5	Sri Hastuti, S.Pd.	IPA	7B	2
6	Sri Hastuti, S.Pd.	IPA	7B	1
7	Sri Hastuti, S.Pd.	IPA	7C	2
8	Sri Hastuti, S.Pd.	IPA	7C	2
9	Sri Hastuti, S.Pd.	IPA	7C	1
10	Hj.Luluk Muflikhatin,M Pdl	Pendidikan Agama	7A	2
11	Hj.Luluk Muflikhatin,M Pdl	Pendidikan Agama	7A	1
12	Hj.Luluk Muflikhatin,M Pdl	Pendidikan Agama	7B	2
13	Hj.Luluk Muflikhatin,M Pdl	Pendidikan Agama	7B	1
14	Hj.Luluk Muflikhatin,M Pdl	Pendidikan Agama	7C	2
15	Hj.Luluk Muflikhatin,M Pdl	Pendidikan Agama	7C	1
16	Drs. Musta'in	Penjas	7A	2
17	Drs. Musta'in	Penjas	7A	1
18	Drs. Musta'in	Penjas	7B	2
19	Drs. Musta'in	Penjas	7B	1
20	Drs. Musta'in	Penjas	7C	2
21	Drs. Musta'in	Penjas	7C	1
22	Siti Kareni, S.Pd.	Bahasa Indonesia	7A	2
23	Siti Kareni, S.Pd.	Bahasa Indonesia	7A	2
24	Siti Kareni, S.Pd.	Bahasa Indonesia	7A	2
25	Siti Kareni, S.Pd.	Bahasa Indonesia	7B	2
26	Siti Kareni, S.Pd.	Bahasa Indonesia	7B	2
27	Siti Kareni, S.Pd.	Bahasa Indonesia	7B	2
28	Siti Kareni, S.Pd.	Bahasa Indonesia	7C	2
29	Siti Kareni, S.Pd.	Bahasa Indonesia	7C	2
30	Siti Kareni, S.Pd.	Bahasa Indonesia	7C	2

**Gambar 6.1** Implementasi *User Interface* Tabel Penugasan Pada Data Sebagian

Gambar 6.1 merupakan hasil proses penugasan untuk menempatkan setiap guru pada beberapa kelas sesuai dengan mata pelajaran yang diajar, jumlah porsi jam pelajaran dan jumlah porsi jam mengajar. Pada data sebagian ini hanya menggunakan 3 kelas yaitu kelas 7A, 7B dan 7C serta porsi jam pelajaran terdapat 10 sehingga kromosom yang dihasilkan yaitu  $3 \times 10 = 30$  kromosom. Untuk hasil proses genetiknya bisa dilihat pada Gambar 6.2.

## APLIKASI PENJADWALAN MATA PELAJARAN

Data Guru    Data Mapel    Penugasan    **Parameter**

### Masukkan Perhitungan

Jumlah Iterasi
Jumlah PopSize
Cr
Mr
<input type="button" value="Proses"/>

Generasi 0		
Kromosom	Bentrok	Fitness
P1 = 1 2 7 7 3 6 16 25 8 12 28 30 17 18 22 5 21 24 23 19 20 15 29 4 14 13 11 9 10 26 2	11	0.0833
P2 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 14 15 25 5 10 2 27 24 8 26 6 13 19 9 30 17	14	0.0667
Anak-anak 0		
Kromosom	Bentrok	Fitness
C1 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 14 27 6 25 8 30 17 5 24 19 15 13 9 10 26 2 ---15P21	12	0.0769
C2 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 14 15 25 5 10 2 27 24 8 26 6 30 17 19 13 9 ---25P21	13	0.0714
C3 = 4 29 22 20 1 7 21 12 16 18 3 11 23 28 14 15 25 5 10 2 27 24 8 26 6 13 19 9 30 17 ----11, 3	13	0.0714
C4 = 4 29 3 20 1 7 21 12 16 18 22 2 23 28 14 15 25 5 10 11 27 24 8 26 6 13 19 9 30 17 ----12, 20	12	0.0769
Generasi 1		
Kromosom	Bentrok	Fitness
P1 = 1 2 7 7 3 6 16 25 8 12 28 30 17 18 22 5 21 24 23 19 20 15 29 4 14 13 11 9 10 26 2	11	0.0833
P2 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 14 27 6 25 8 30 17 5 24 19 15 13 9 10 26 2	12	0.0769
Anak-anak 1		
Kromosom	Bentrok	Fitness
C1 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 14 27 6 25 8 30 17 5 24 19 15 13 9 10 26 2 ---23P21	12	0.0769
C2 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 14 27 6 25 8 30 17 5 24 19 15 13 9 10 26 2 ---23P21	12	0.0769
C3 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 6 27 14 25 8 30 17 5 24 19 15 13 9 10 26 2 ----17, 15	12	0.0769
C4 = 4 29 3 20 1 7 21 12 16 18 22 11 23 28 8 27 6 25 14 30 17 5 24 19 15 13 9 10 26 2 ----15, 19	10	0.0909

### Hasil Kromosom Terbaik

4 29 3 20 1 7 21 12 16 18 22 11 23 28 8 27 6 25 14 30 17 5 24 19 15 13 9 10 26 2

**Fitness**

0.0909

Waktu menampilkan halaman 0.00 seconds

**Gambar 6.2** Implementasi *User Interface* Hasil Proses Genetika Pada Data Sebagian

Gambar 6.2 merupakan hasil proses genetika menggunakan data sebagian dengan inputan jumlah iterasi = 2, jumlah populasi = 2,  $cr = 1$ , dan  $mr = 1$ . Setelah dilakukan proses genetika maka akan dihasilkan penjadwalan dari data sebagian yang bisa dilihat pada Gambar 6.3.

## Hasil Penjadwalan Mata Pelajaran

Jumlah bentrok :10

Hari	7A	7B	7C
<b>Senin</b>			
07.00-07.40	-	-	-
07.40-08.20	-	-	-
08.20-09.00	Sri Hastuti, S.Pd.	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
09.20-10.00	Sri Hastuti, S.Pd.	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
10.00-10.40	Sri Hastuti, S.Pd.	Hj.Luluk Muflikhatin,M Pd	Drs. Mustain
10.40-11.20	Drs. Mustain	Hj.Luluk Muflikhatin,M Pd	Drs. Mustain
11.40-12.20	Drs. Mustain	Drs. Mustain	Sri Hastuti, S.Pd.
12.20-13.00	Siti Kareni, S.Pd.	Drs. Mustain	Sri Hastuti, S.Pd.
<b>Selasa</b>			
07.00-07.40	Siti Kareni, S.Pd.	Siti Kareni, S.Pd.	Drs. Mustain
07.40-08.20	Hj.Luluk Muflikhatin,M Pd	Siti Kareni, S.Pd.	Siti Kareni, S.Pd.
08.20-09.00	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
09.20-10.00	Siti Kareni, S.Pd.	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.
10.00-10.40	Drs. Mustain	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.
10.40-11.20	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.	Hj.Luluk Muflikhatin,M Pd
11.40-12.20	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.	Hj.Luluk Muflikhatin,M Pd
12.20-13.00	Hj.Luluk Muflikhatin,M Pd	Drs. Mustain	Siti Kareni, S.Pd.
<b>Rabu</b>			
07.00-07.40	Hj.Luluk Muflikhatin,M Pd	Hj.Luluk Muflikhatin,M Pd	Siti Kareni, S.Pd.
07.40-08.20	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd
08.20-09.00	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.
09.20-10.00	-	-	-
10.00-10.40	-	-	-
10.40-11.20	-	-	-
11.40-12.20	-	-	-
12.20-13.00	-	-	-

**Gambar 6.3** Implementasi *User Interface* Hasil Penjadwalan Pada Data Sebagian

Gambar 6.3 merupakan hasil penjadwalan mata pelajaran dengan menggunakan data sebagian yang masih terdapat bentrok dan ditandai dengan tanda merah. Untuk melihat hasil penjadwalan mata pelajaran yang optimal dapat dilihat pada Gambar 6.4.

## Hasil Penjadwalan Mata Pelajaran

Jumlah bentrok :1

Hari	7A	7B	7C
<b>Senin</b>			
07.00-07.40	-	-	-
07.40-08.20	-	-	-
08.20-09.00	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.	Drs. Mustain
09.20-10.00	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd	Sri Hastuti, S.Pd.
10.00-10.40	Siti Kareni, S.Pd.	Drs. Mustain	Sri Hastuti, S.Pd.
10.40-11.20	Siti Kareni, S.Pd.	Drs. Mustain	Sri Hastuti, S.Pd.
11.40-12.20	Sri Hastuti, S.Pd.	Drs. Mustain	Siti Kareni, S.Pd.
12.20-13.00	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd	Siti Kareni, S.Pd.
<b>Selasa</b>			
07.00-07.40	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd	Sri Hastuti, S.Pd.
07.40-08.20	Hj.Luluk Muflikhatin,M Pd	Siti Kareni, S.Pd.	Sri Hastuti, S.Pd.
08.20-09.00	Hj.Luluk Muflikhatin,M Pd	Siti Kareni, S.Pd.	Drs. Mustain
09.20-10.00	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.	Drs. Mustain
10.00-10.40	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd
10.40-11.20	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd
11.40-12.20	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.	Hj.Luluk Muflikhatin,M Pd
12.20-13.00	Hj.Luluk Muflikhatin,M Pd	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
<b>Rabu</b>			
07.00-07.40	Drs. Mustain	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
07.40-08.20	Drs. Mustain	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
08.20-09.00	Drs. Mustain	Sri Hastuti, S.Pd.	Siti Kareni, S.Pd.
09.20-10.00	-	-	-
10.00-10.40	-	-	-
10.40-11.20	-	-	-
11.40-12.20	-	-	-
12.20-13.00	-	-	-

**Gambar 6.4** Implementasi *User Interface* Hasil Penjadwalan Optimal Pada Data Sebagian

Gambar 6.4 merupakan hasil penjadwalan mata pelajaran dengan menggunakan data sebagian yang paling optimal walaupun masih ada satu bentrok. Untuk mendapatkan hasil penjadwalan tersebut dimasukkan parameter genetika dengan jumlah populasi = 50, jumlah iterasi = 200,  $cr = 1$ , dan  $mr = 1$ . Pada penjadwalan tersebut masih terdapat bentrok sebanyak satu karena kesalahan pada data yang hanya menggunakan sebagian kecil dari data guru maupun data kelasnya. Dengan adanya hasil penjadwalan tersebut maka dapat dibuktikan bahwa sistem yang digunakan sudah sesuai dengan algoritma genetika yang ada dan dapat menghasilkan penjadwalan mata pelajaran yang optimal.

## 6.2 Sistematika Pengujian

Pada pengujian sistem penjadwalan mata pelajaran menggunakan algoritma genetika ini dilakukan terhadap parameter jumlah populasi (*popsize*), *crossover rate* ( $cr$ ), *mutation rate* ( $mr$ ) dan jumlah iterasi. Tujuan dari proses pengujian ini untuk mengetahui nilai *fitness* yang dihasilkan terhadap pengaruh parameter sehingga didapatkan solusi penjadwalan mata pelajaran yang optimal dengan meminimalisir nilai *pinalti* dalam menyelesaikan masalah penjadwalan mata pelajaran. Untuk percobaan pengujian ini dilakukan sebanyak 10 kali terhadap masing – masing pengujian parameter kemudian dihitung rata – rata nilai *fitness* dari percobaan yang telah dilakukan.

## 6.3 Analisa Hasil dan Pembahasan

Pada sub bab ini akan dibahas mengenai hasil pengujian terhadap masing-masing parameter algoritma genetika serta pengaruhnya terhadap nilai *fitness* yang dihasilkan.

### 6.3.1 Hasil Pengujian Terhadap Banyaknya Generasi

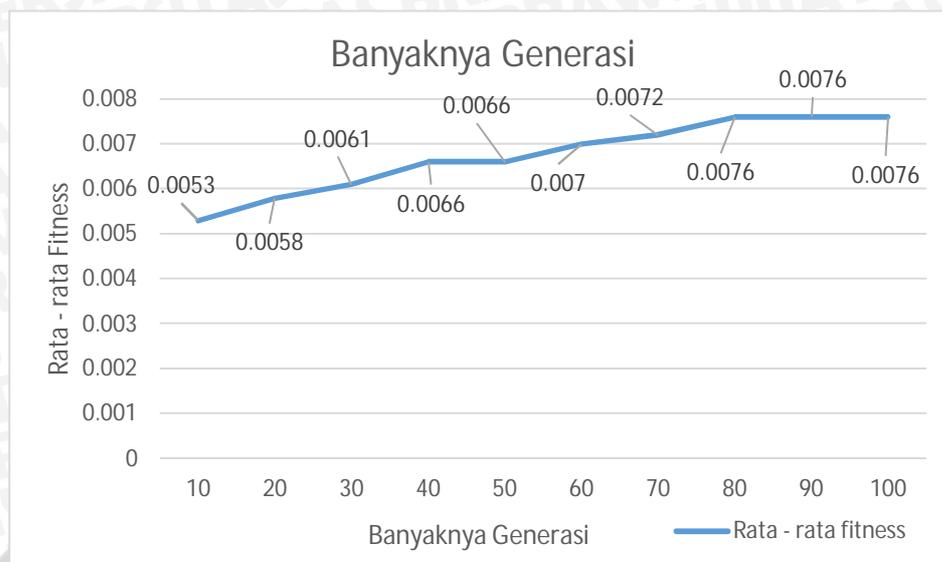
Pada pengujian ini dilakukan terhadap parameter banyaknya generasi yang bertujuan untuk mengetahui jumlah generasi yang optimal pada masalah penjadwalan mata pelajaran. Nilai jumlah generasi yang digunakan pada pengujian ini yaitu 10 – 100 dengan kelipatan 10 serta masing – masing 10 kali percobaan. Jumlah populasi yang digunakan yaitu 15 populasi dengan kombinasi

$cr$  dan  $mr$  yaitu 0,5 : 0,5. Hasil pengujian terhadap banyaknya generasi dapat dilihat pada Tabel 6.1.

**Tabel 6.1** Hasil Pengujian Banyaknya Generasi (  $cr = 0,5$  ;  $mr = 0,5$  )

Perco baan ke-	Banyaknya Generasi									
	10	20	30	40	50	60	70	80	90	100
	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>
1	0.0054	0.0063	0.0061	0.0065	0.0071	0.0074	0.0077	0.0075	0.0079	0.0078
2	0.0051	0.0057	0.0056	0.0069	0.0070	0.0068	0.0068	0.0081	0.0076	0.0072
3	0.0052	0.0059	0.0061	0.0066	0.0065	0.0074	0.0069	0.0070	0.0076	0.0079
4	0.0054	0.0056	0.0063	0.0062	0.0065	0.0071	0.0073	0.0076	0.0079	0.0076
5	0.0053	0.0055	0.0064	0.0064	0.0071	0.0070	0.0079	0.0081	0.0069	0.0078
6	0.0054	0.0059	0.0060	0.0067	0.0064	0.0067	0.0074	0.0070	0.0076	0.0072
7	0.0054	0.0063	0.0058	0.0067	0.0063	0.0074	0.0069	0.0076	0.0076	0.0079
8	0.0054	0.0060	0.0064	0.0068	0.0062	0.0064	0.0074	0.0081	0.0079	0.0076
9	0.0053	0.0054	0.0064	0.0067	0.0067	0.0070	0.0063	0.0070	0.0069	0.0078
10	0.0054	0.0053	0.0060	0.0069	0.0065	0.0067	0.0070	0.0076	0.0076	0.0072
<b>Rata-rata Fitness</b>	0.0053	0.0058	0.0061	0.0066	0.0066	0.0070	0.0072	0.0076	0.0076	0.0076
<b>Rata-rata Waktu</b>	3.03	5.47	8.11	10.65	13.02	15.98	18.14	21.11	23.77	25.02

Berdasarkan Tabel 6.1 maka didapatkan nilai *fitness* terbesar pada jumlah generasi 80, 90 dan 100 dengan nilai *fitness* 0.0076 dan waktu komputasi 21.11 detik pada generasi 80, 23.77 detik pada generasi 90 serta 25.02 detik pada generasi 100. Untuk nilai *fitness* terkecil didapatkan pada jumlah generasi 10 dengan nilai 0.0053 dan waktu komputasinya 3.03 detik. Adapun grafik hasil pengujian dari pengaruh banyaknya generasi yang dapat dilihat pada Gambar 6.5.



**Gambar 6.5** Grafik Hasil Pengujian Pengaruh Banyaknya Generasi

Pada Gambar 6.5 merupakan grafik dari pengaruh banyaknya generasi yang dapat dilihat bahwa rata – rata *fitness* yang dihasilkan mulai dari jumlah generasi 10 hingga 100 selalu mengalami kenaikan hanya pada jumlah generasi 40 dan 50 memiliki rata – rata nilai *fitness* yang sama yaitu 0.0066 serta jumlah generasi 80, 90 dan 100 juga memiliki nilai *fitness* sama yaitu 0.0076. Dari hasil pengujian maka didapatkan kesimpulan bahwa semakin banyak generasi maka akan menghasilkan nilai *fitness* yang lebih baik tetapi akan membutuhkan waktu komputasi yang lebih lama. Pola kenaikan *fitness* seperti ini juga didapatkan oleh Sundarniningsih (2015) yang menerapkan algoritma genetika untuk optimasi pencarian rute.

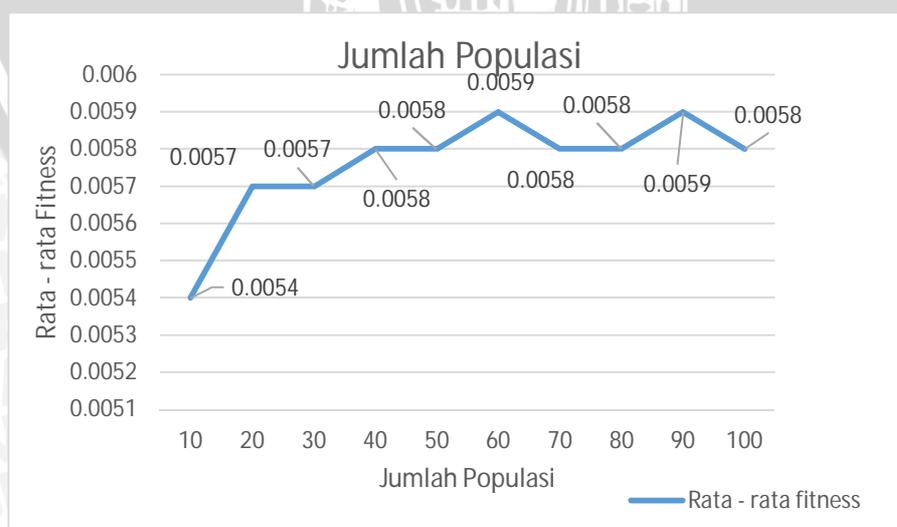
### 6.3.2 Hasil Pengujian Terhadap Jumlah Populasi

Pada pengujian ini dilakukan terhadap parameter jumlah populasi yang bertujuan untuk mengetahui jumlah populasi yang optimal pada masalah penjadwalan mata pelajaran. Nilai jumlah populasi yang digunakan pada pengujian ini yaitu 10 – 100 dengan kelipatan 10 serta masing – masing 10 kali percobaan. Jumlah generasi yang digunakan yaitu 15 dengan kombinasi *cr* dan *mr* yaitu 0,5 : 0,5. Hasil pengujian terhadap jumlah populasi dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Jumlah Populasi ( $cr = 0,5$  ;  $mr = 0,5$ )

Perco baan ke-	Banyaknya Generasi									
	10	20	30	40	50	60	70	80	90	100
	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>	<i>fitness</i>
1	0.0054	0.0056	0.0059	0.0058	0.0056	0.0061	0.0058	0.0056	0.0062	0.0060
2	0.0055	0.0059	0.0058	0.0057	0.0056	0.0059	0.0057	0.0058	0.0062	0.0057
3	0.0053	0.0058	0.0059	0.0056	0.0059	0.0059	0.0058	0.0058	0.0059	0.0061
4	0.0051	0.0058	0.0055	0.0060	0.0057	0.0058	0.0060	0.0058	0.0058	0.0055
5	0.0053	0.0057	0.0058	0.0059	0.0056	0.0056	0.0059	0.0061	0.0057	0.0060
6	0.0056	0.0054	0.0058	0.0057	0.0058	0.0063	0.0058	0.0056	0.0061	0.0057
7	0.0058	0.0055	0.0057	0.0055	0.0056	0.0059	0.0062	0.0058	0.0059	0.0061
8	0.0053	0.0061	0.0058	0.0057	0.0061	0.0058	0.0057	0.0058	0.0057	0.0055
9	0.0052	0.0058	0.0057	0.0060	0.0062	0.0058	0.0058	0.0058	0.0058	0.0060
10	0.0052	0.0056	0.0056	0.0058	0.0058	0.0057	0.0056	0.0058	0.0060	0.0057
<b>Rata-rata Fitness</b>	0.0054	0.0057	0.0057	0.0058	0.0058	<b>0.0059</b>	0.0058	0.0058	<b>0.0059</b>	0.0058
<b>Rata-rata Waktu</b>	2.58	5.06	7.77	10.58	12.92	16.49	18.82	21.23	23.85	26.59

Berdasarkan Tabel 6.2 maka didapatkan nilai *fitness* terbesar pada jumlah populasi 60 dan 90 dengan nilai *fitness* 0.0059 dan waktu komputasi detik 16,49 detik untuk populasi 60 serta 23,85 untuk populasi 90. Untuk nilai *fitness* terkecil didapatkan pada jumlah populasi 10 dengan nilai 0.0054 dan waktu komputasinya 2,58. Adapun grafik hasil pengujian dari pengaruh jumlah populasi yang dapat dilihat pada Gambar 6.6.



Gambar 6.6 Grafik Hasil Pengujian Pengaruh Jumlah Populasi

Pada Gambar 6.6 rata – rata *fitness* yang dihasilkan mulai dari jumlah populasi 10 hingga 60 selalu mengalami kenaikan namun pada jumlah generasi 70 dan 80 mengalami penurunan lalu meningkat lagi pada jumlah generasi 90 serta penurunan lagi pada jumlah generasi 100. Nilai *fitness* ukuran 20 dan 30 memiliki nilai yang sama yaitu 0.0057 dan nilai *fitness* yang memiliki nilai yaitu 0.0058 terdapat pada ukuran 40, 50, 70 dan 80. Secara umum dengan meningkatnya jumlah populasi maka nilai *fitness* yang dihasilkan semakin baik karena akan menghasilkan keragaman individu yang lebih banyak. Namun dengan meningkatnya jumlah populasi maka akan berpengaruh dalam proses genetiknya yang akan membutuhkan waktu komputasi lebih lama seperti dapat dilihat pada Tabel 6.2. Tetapi dari hasil grafik didapatkan bahwa meningkatnya jumlah populasi tidak menjamin akan menghasilkan nilai *fitness* lebih baik karena nilai *fitness* yang dihasilkan masih fluktuatif dan kebanyakan nilainya hampir sama. Dari hasil pengujian maka didapatkan kesimpulan bahwa semakin besar jumlah populasi maka rata – rata *fitness* yang dihasilkan cenderung makin besar karena dengan penambahan ukuran populasi maka memungkinkan algoritma genetika untuk mengeksplorasi area pencarian dan sekaligus mendapatkan solusi yang lebih baik. Tetapi pada batas tertentu ukuran populasi yang terlalu besar akan membebani waktu komputasi dan kenaikan *fitness* yang didapatkan tidak terlalu signifikan. Oleh karena itu dengan penambahan populasi maka tidak menjamin akan menghasilkan nilai *fitness* lebih baik. Perubahan yang tidak begitu besar ini terjadi karena anak yang dihasilkan pada saat proses reproduksi mirip dengan induknya (Mahmudy, 2014).

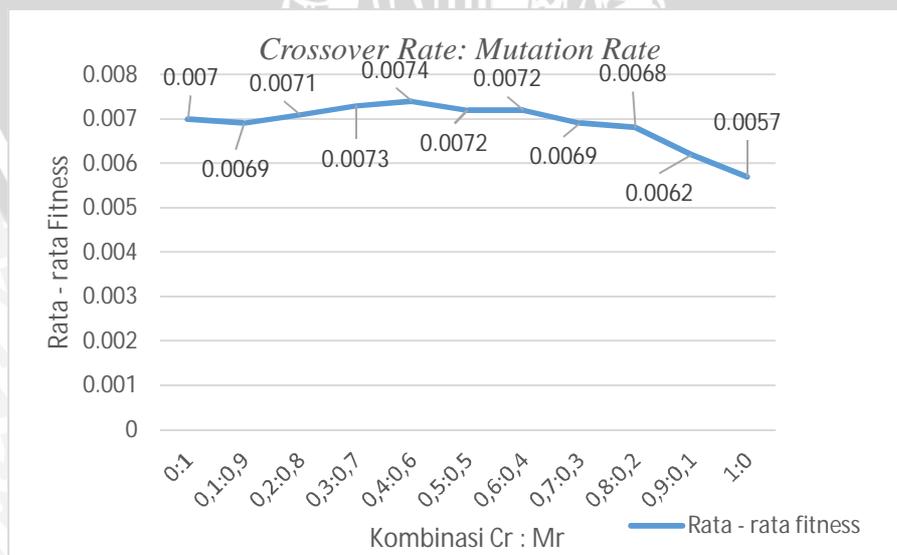
### 6.3.3 Hasil Pengujian Kombinasi *Crossover Rate* dan *Mutation Rate*

Pada pengujian ini bertujuan untuk mengetahui kombinasi *crossover rate* dan *mutation rate* yang optimal pada masalah penjadwalan mata pelajaran. Jumlah generasi dan populasi yang digunakan adalah generasi 70 dan populasi 20. Sedangkan kombinasi yang digunakan yaitu nilai 0 – 1. Nilai dari kombinasi *cr* dan *mr* yang digunakan pada pengujian ini yaitu 0:1, 0.1:0.9, 0.2:0.8, 0.3:0.7, 0.4:0.6, 0.5:0.5, 0.6:0.4, 0.7:0.3, 0.8:0.2, 0.9:0.1 dan 1:0. Hasil pengujian terhadap kombinasi *cr* dan *mr* dapat dilihat pada Tabel 6.3.

**Tabel 6.3** Hasil Pengujian Kombinasi *Crossover Rate* dan *Mutation Rate*

Perobaan ke-	Crossover rate : Mutation rate										
	0:1	0.1:0.9	0.2:0.8	0.3:0.7	0.4:0.6	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2	0.9:0.1	1:0
	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>	<i>Fitness</i>
1	0.0071	0.0064	0.0072	0.0076	0.0068	0.0067	0.0074	0.0076	0.0071	0.0059	0.0056
2	0.0068	0.0072	0.0072	0.0075	0.0074	0.0068	0.0070	0.0069	0.0067	0.0065	0.0056
3	0.0068	0.0070	0.0070	0.0074	0.0072	0.0076	0.0078	0.0072	0.0062	0.0060	0.0055
4	0.0070	0.0069	0.0071	0.0067	0.0072	0.0073	0.0064	0.0068	0.0076	0.0059	0.0057
5	0.0069	0.0068	0.0074	0.0066	0.0080	0.0072	0.0078	0.0070	0.0071	0.0063	0.0058
6	0.0070	0.0070	0.0070	0.0076	0.0068	0.0070	0.0068	0.0065	0.0064	0.0063	0.0057
7	0.0071	0.0070	0.0065	0.0075	0.0078	0.0071	0.0074	0.0071	0.0068	0.0062	0.0058
8	0.0072	0.0068	0.0077	0.0076	0.0076	0.0070	0.0072	0.0062	0.0071	0.0059	0.0059
9	0.0069	0.0071	0.0070	0.0073	0.0076	0.0071	0.0071	0.0068	0.0061	0.0070	0.0056
10	0.0070	0.0069	0.0067	0.0071	0.0075	0.0084	0.0070	0.0071	0.0070	0.0063	0.0057
Rata-rata <i>Fitness</i>	0.0070	0.0069	0.0071	0.0073	0.0074	0.0072	0.0072	0.0069	0.0068	0.0062	0.0057
Rata - rata Waktu	5.92	9.25	13.19	16.18	20.21	24.13	30.47	34.10	37.62	40.66	48.10

Berdasarkan Tabel 6.3 maka didapatkan nilai *fitness* terbesar pada kombinasi *cr* : *mr* yaitu 0,4 : 0,6 dengan nilai *fitness* 0.0074 dan waktu komputasi 20,21 detik. Untuk nilai *fitness* terkecil didapatkan pada kombinasi *cr* : *mr* yaitu 1:0 dengan nilai 0.0057 dan waktu komputasinya. 48,10 detik. Adapun grafik hasil pengujian dari pengaruh kombinasi *crossover rate* dan *mutation rate* yang dapat dilihat pada Gambar 6.7.

**Gambar 6.7** Grafik Hasil Pengujian Kombinasi Cr : Mr

Pada Gambar 6.7 rata – rata *fitness* pada kombinasi *cr* : *mr* 0:1 ke 0,1:0,9 sudah mengalami penurunan namun ke kombinasi selanjutnya mengalami kenaikan hingga kombinasi 0,4:0.6. Kemudian mengalami penurunan kembali pada *cr* : *mr* 0,5:0,5 dengan menghasilkan nilai rata – rata *fitness* yang tetap yaitu 0.0072 yang sama dengan kombinasi 0,6:0,4. Dan selanjutnya mengalami penurunan kembali hingga kombinasi *cr* : *mr* yaitu 1:0 dengan menghasilkan nilai rata – rata *fitness* terendah yaitu 0.0057. Kombinasi *crossover rate* dan *mutation rate* yang menghasilkan nilai optimal memang sulit untuk dicari karena proses algoritma genetika ini menghasilkan nilai yang tidak pasti tergantung hasil *random*. Jika nilai *crossover rate* lebih tinggi daripada nilai *mutation rate* maka akan menghasilkan nilai yang hampir sama atau disebut juga konvergensi dini sehingga akan menurunkan kemampuan algoritma genetika untuk mengeksplorasi daerah pencarian dan hanya berada pada daerah optimum lokal. Namun apabila nilai *crossover rate* yang rendah maka algoritma genetika akan sangat tergantung pada proses mutasi karena *crossover rate* yang rendah akan menyebabkan menurunnya kemampuan algoritma genetika untuk belajar dari generasi sebelumnya dan tidak mampu mengeksplorasi daerah pencarian secara efektif (Mahmudy, WF, Marian, RM & Luong, LHS 2014).



### 6.3.4 Hasil Penjadwalan

Dengan menggunakan parameter – parameter terbaik ini maka algoritma genetika dijalankan lagi sehingga akan menampilkan hasil penjadwalan secara lengkap yang ditunjukkan pada Gambar 6.8.

#### Hasil Penjadwalan Mata Pelajaran

Jumlah baris: 123

Hari	7A	7B	7C	7D	7E	7F	7G	7H
<b>Senin</b>								
07:00-07:40	-	-	-	-	-	-	-	-
07:40-08:20	-	-	-	-	-	-	-	-
08:20-09:00	Behar Masduki, S.Pd	Dian Estash Wihjeng, S.	Nurbudi Hariono	Dewi Neutrisna, S.Pd	Dra Eveline M Pd	Setyo Rakhmad, S.Pd	Dian Estash Wihjeng, S.	Dian Estash Wihjeng, S.
09:00-10:00	Behar Masduki, S.Pd	Dian Estash Wihjeng, S.	Nurbudi Hariono	Dewi Neutrisna, S.Pd	Dra Eveline M Pd	Setyo Rakhmad, S.Pd	Dian Estash Wihjeng, S.	Dian Estash Wihjeng, S.
10:00-10:40	Okriyana Ricke V, S.Pd	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Dian Estash Wihjeng, S.	Dewi Neutrisna, S.Pd	Meisani Prihartini, S.Pd	Erni Puji Lestari, S.Pd	Hj Luhk Muflikhatin, M Pd
10:40-11:20	Okriyana Ricke V, S.Pd	Warnadi, S.Pd	Sri Hastuti, S.Pd	Dian Estash Wihjeng, S.	Dewi Neutrisna, S.Pd	Meisani Prihartini, S.Pd	Erni Puji Lestari, S.Pd	Hj Luhk Muflikhatin, M Pd
11:40-12:20	Dian Estash Wihjeng, S.	Hj Luhk Muflikhatin, M Pd	Sri Hastuti, S.Pd	Nurbudi Hariono	Hj Luhk Muflikhatin, M Pd	Meisani Prihartini, S.Pd	Meisani Prihartini, S.Pd	Hj Sumartiyah, S.Pd
12:20-13:00	Siti Kareni, S.Pd	Warnadi, S.Pd	Warnadi, S.Pd	Nurbudi Hariono	Sri Hastuti, S.Pd	Dian Estash Wihjeng, S.	Meisani Prihartini, S.Pd	Hj Sumartiyah, S.Pd
<b>Selasa</b>								
07:00-07:40	Siti Kareni, S.Pd	Behar Masduki, S.Pd	Warnadi, S.Pd	Dra Mustain	Dra Eveline M Pd	Dian Estash Wihjeng, S.	Dra Eveline M Pd	Erni Puji Lestari, S.Pd
07:40-08:20	Dian Estash Wihjeng, S.	Behar Masduki, S.Pd	Dra Mustain	Nurbudi Hariono	Dra Eveline M Pd	Okriyana Ricke V, S.Pd	Dra Eveline M Pd	Erni Puji Lestari, S.Pd
08:20-09:00	Dian Estash Wihjeng, S.	Okriyana Ricke V, S.Pd	Dewi Neutrisna, S.Pd	Siti Kareni, S.Pd	Hj Luhk Muflikhatin, M Pd	Okriyana Ricke V, S.Pd	Meisani Prihartini, S.Pd	Nurbudi Hariono
09:00-10:00	Hj Luhk Muflikhatin, M Pd	Okriyana Ricke V, S.Pd	Dewi Neutrisna, S.Pd	Siti Kareni, S.Pd	Hj Luhk Muflikhatin, M Pd	Warnadi, S.Pd	Meisani Prihartini, S.Pd	Dra Mustain
10:00-10:40	Hj Luhk Muflikhatin, M Pd	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Siti Kareni, S.Pd	Andik Susanto, S.Pd	Warnadi, S.Pd	Setyo Rakhmad, S.Pd	Dra Eveline M Pd
10:40-11:20	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Siti Kareni, S.Pd	Siti Kareni, S.Pd	Andik Susanto, S.Pd	Setyo Rakhmad, S.Pd	Dra Eveline M Pd	Dra Eveline M Pd
11:40-12:20	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Siti Kareni, S.Pd	Setyo Rakhmad, S.Pd	Sri Hastuti, S.Pd	Dewi Neutrisna, S.Pd	Erni Puji Lestari, S.Pd	Hj Sumartiyah, S.Pd
12:20-13:00	Siti Kareni, S.Pd	Dian Estash Wihjeng, S.	Siti Kareni, S.Pd	Setyo Rakhmad, S.Pd	Nurbudi Hariono	Dewi Neutrisna, S.Pd	Erni Puji Lestari, S.Pd	Hj Sumartiyah, S.Pd
<b>Rabu</b>								
07:00-07:40	Siti Kareni, S.Pd	Siti Kareni, S.Pd	Dra Mustain	Dian Estash Wihjeng, S.	Nurbudi Hariono	Andik Susanto, S.Pd	Setyo Rakhmad, S.Pd	Hj Sumartiyah, S.Pd
07:40-08:20	Hj Luhk Muflikhatin, M Pd	Siti Kareni, S.Pd	Dra Mustain	Dra Mustain	Setyo Rakhmad, S.Pd	Andik Susanto, S.Pd	Setyo Rakhmad, S.Pd	Kecaminingsih, S.Pd
08:20-09:00	Sri Hastuti, S.Pd	Siti Kareni, S.Pd	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Setyo Rakhmad, S.Pd	Dra Mustain	Kecaminingsih, S.Pd	Kecaminingsih, S.Pd
09:00-10:00	Dra Mustain	Siti Kareni, S.Pd	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Sri Hastuti, S.Pd	Dra Eveline M Pd	Kecaminingsih, S.Pd	Nurbudi Hariono
10:00-10:40	Dra Mustain	Behar Masduki, S.Pd	Siti Kareni, S.Pd	Hj Luhk Muflikhatin, M Pd	Dian Estash Wihjeng, S.	Dra Eveline M Pd	Hj Luhk Muflikhatin, M Pd	Nurbudi Hariono
10:40-11:20	Nurbudi Hariono	Behar Masduki, S.Pd	Siti Kareni, S.Pd	Dewi Neutrisna, S.Pd	Dian Estash Wihjeng, S.	Meisani Prihartini, S.Pd	Hj Luhk Muflikhatin, M Pd	Dra Mustain
11:40-12:20	Behar Masduki, S.Pd	Nurbudi Hariono	Dewi Neutrisna, S.Pd	Dewi Neutrisna, S.Pd	Okriyana Ricke V, S.Pd	Dra Mustain	Hj Luhk Muflikhatin, M Pd	Dra Mustain
12:20-13:00	Behar Masduki, S.Pd	Andik Susanto, S.Pd	Dewi Neutrisna, S.Pd	Sri Hastuti, S.Pd	Okriyana Ricke V, S.Pd	Dra Mustain	Kecaminingsih, S.Pd	Erni Puji Lestari, S.Pd
<b>Kamis</b>								
07:00-07:40	Nurbudi Hariono	Andik Susanto, S.Pd	Dian Estash Wihjeng, S.	Sri Hastuti, S.Pd	Warnadi, S.Pd	Dra Eveline M Pd	Kecaminingsih, S.Pd	Erni Puji Lestari, S.Pd
07:40-08:20	Nurbudi Hariono	Andik Susanto, S.Pd	Dian Estash Wihjeng, S.	Setyo Rakhmad, S.Pd	Warnadi, S.Pd	Dra Eveline M Pd	Kecaminingsih, S.Pd	Dra Eveline M Pd
08:20-09:00	Warnadi, S.Pd	Nurbudi Hariono	Dian Estash Wihjeng, S.	Setyo Rakhmad, S.Pd	Sri Hastuti, S.Pd	Dra Eveline M Pd	Dra Eveline M Pd	Hj Luhk Muflikhatin, M Pd
09:00-10:00	Warnadi, S.Pd	Behar Masduki, S.Pd	Nurbudi Hariono	Setyo Rakhmad, S.Pd	Sri Hastuti, S.Pd	Dewi Neutrisna, S.Pd	Meisani Prihartini, S.Pd	Dra Eveline M Pd
10:00-10:40	Dra Mustain	Behar Masduki, S.Pd	Setyo Rakhmad, S.Pd	Warnadi, S.Pd	Nurbudi Hariono	Dewi Neutrisna, S.Pd	Meisani Prihartini, S.Pd	Dra Eveline M Pd
10:40-11:20	Dewi Neutrisna, S.Pd	Siti Kareni, S.Pd	Sri Hastuti, S.Pd	Warnadi, S.Pd	Hj Luhk Muflikhatin, M Pd	Dian Estash Wihjeng, S.	Nurbudi Hariono	Okriyana Ricke V, S.Pd
11:40-12:20	Dewi Neutrisna, S.Pd	Siti Kareni, S.Pd	Setyo Rakhmad, S.Pd	Warnadi, S.Pd	Warnadi, S.Pd	Meisani Prihartini, S.Pd	Dra Eveline M Pd	Okriyana Ricke V, S.Pd
12:20-13:00	Andik Susanto, S.Pd	Sri Hastuti, S.Pd	Setyo Rakhmad, S.Pd	Siti Kareni, S.Pd	Dewi Neutrisna, S.Pd	Meisani Prihartini, S.Pd	Dra Eveline M Pd	Kecaminingsih, S.Pd
<b>Jumat</b>								
07:00-07:40	-	-	-	-	-	-	-	-
07:40-08:20	-	-	-	-	-	-	-	-
08:20-09:00	Andik Susanto, S.Pd	Sri Hastuti, S.Pd	Sri Hastuti, S.Pd	Siti Kareni, S.Pd	Dra Mustain	Warnadi, S.Pd	Okriyana Ricke V, S.Pd	Kecaminingsih, S.Pd
09:00-10:00	Dewi Neutrisna, S.Pd	Sri Hastuti, S.Pd	Sri Hastuti, S.Pd	Andik Susanto, S.Pd	Dra Mustain	Warnadi, S.Pd	Okriyana Ricke V, S.Pd	Meisani Prihartini, S.Pd
10:00-10:40	Behar Masduki, S.Pd	Sri Hastuti, S.Pd	Setyo Rakhmad, S.Pd	Andik Susanto, S.Pd	Dian Estash Wihjeng, S.	Hj Luhk Muflikhatin, M Pd	Dian Estash Wihjeng, S.	Andik Susanto, S.Pd
10:40-11:20	-	-	-	-	-	-	-	-
11:40-12:20	-	-	-	-	-	-	-	-
12:20-13:00	-	-	-	-	-	-	-	-
<b>Sabtu</b>								
07:00-07:40	Sri Hastuti, S.Pd	Dewi Neutrisna, S.Pd	Andik Susanto, S.Pd	Dra Mustain	Warnadi, S.Pd	Dra Eveline M Pd	Nurbudi Hariono	Meisani Prihartini, S.Pd
07:40-08:20	Sri Hastuti, S.Pd	Dewi Neutrisna, S.Pd	Andik Susanto, S.Pd	Sri Hastuti, S.Pd	Dra Eveline M Pd	Dra Eveline M Pd	Dra Mustain	Meisani Prihartini, S.Pd
08:20-09:00	Siti Kareni, S.Pd	Dra Mustain	Okriyana Ricke V, S.Pd	Sri Hastuti, S.Pd	Dra Eveline M Pd	Nurbudi Hariono	Andik Susanto, S.Pd	Dra Eveline M Pd
09:00-10:00	Siti Kareni, S.Pd	Dra Mustain	Okriyana Ricke V, S.Pd	Setyo Rakhmad, S.Pd	Setyo Rakhmad, S.Pd	Nurbudi Hariono	Andik Susanto, S.Pd	Dra Eveline M Pd
10:00-10:40	Sri Hastuti, S.Pd	Dewi Neutrisna, S.Pd	Siti Kareni, S.Pd	Okriyana Ricke V, S.Pd	Setyo Rakhmad, S.Pd	Setyo Rakhmad, S.Pd	Dra Mustain	Meisani Prihartini, S.Pd
10:40-11:20	Sri Hastuti, S.Pd	Dewi Neutrisna, S.Pd	Siti Kareni, S.Pd	Okriyana Ricke V, S.Pd	Setyo Rakhmad, S.Pd	Setyo Rakhmad, S.Pd	Dra Mustain	Meisani Prihartini, S.Pd
11:40-12:20	-	-	-	-	-	-	-	-
12:20-13:00	-	-	-	-	-	-	-	-

SA	SB	SC	SD	SE	SF	SG	SH
Hj Sumartiyah, S Pd Meinemi Prihartini, S Pd Meinemi Prihartini, S Pd H. Makedi, S Ag Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd	Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Kasminingsih, S Pd Kasminingsih, S Pd Oktriyanas Ricke V, S Pd Oktriyanas Ricke V, S Pd	Rohimah, S Pd Rohimah, S Pd Trisia Santi Perline, S P Andik Susanto, S Pd Andik Susanto, S Pd Andik Susanto, S Pd H Purnomo, S Pd	Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd Andik Susanto, S Pd Andik Susanto, S Pd Trisia Santi Perline, S P Hj Sumartiyah, S Pd	Iyaa, S Pd Iyaa, S Pd Masripah Farikah, S Pd Masripah Farikah, S Pd H. Makedi, S Ag H. Makedi, S Ag	Rohimah, S Pd Rohimah, S Pd Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd Ummu Hanik, S Pd Ummu Hanik, S Pd	Masripah Farikah, S Pd Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd Ummu Hanik, S Pd Ummu Hanik, S Pd Kuwat Setyobudi, S Pd	Yayuk Purwantini, S Pd Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd H. Purnomo, S Pd Masripah Farikah, S Pd Masripah Farikah, S Pd
H. Purnomo, S Pd Masripah Farikah, S Pd Masripah Farikah, S Pd Kasminingsih, S Pd Kasminingsih, S Pd Masripah Farikah, S Pd Meinemi Prihartini, S Pd Meinemi Prihartini, S Pd	Meinemi Prihartini, S Pd Meinemi Prihartini, S Pd H. Purnomo, S Pd H. Purnomo, S Pd Masripah Farikah, S Pd Hj Sumartiyah, S Pd Hj Sumartiyah, S Pd Meinemi Prihartini, S Pd	Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Masripah Farikah, S Pd Masripah Farikah, S Pd Oktriyanas Ricke V, S Pd Oktriyanas Ricke V, S Pd H. Makedi, S Ag H. Makedi, S Ag	Hj Sumartiyah, S Pd H. Makedi, S Ag Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd Trisia Santi Perline, S P Trisia Santi Perline, S P Masripah Farikah, S Pd Masripah Farikah, S Pd	Rohimah, S Pd Rohimah, S Pd Ummu Hanik, S Pd Ummu Hanik, S Pd H. Purnomo, S Pd H. Purnomo, S Pd Trisia Santi Perline, S P Rohimah, S Pd	Masripah Farikah, S Pd Rohimah, S Pd Rohimah, S Pd Sopii, S Pd Sopii, S Pd Yayuk Purwantini, S Pd Iyaa, S Pd Iyaa, S Pd	Kuwat Setyobudi, S Pd Trisia Santi Perline, S P Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd H. Makedi, S Ag H. Makedi, S Ag Rohimah, S Pd Rohimah, S Pd	Sopii, S Pd Sopii, S Pd Andriyani, S Koom Andriyani, S Koom Iyaa, S Pd Iyaa, S Pd Betty Endang Prammni, S Pd Trisia Santi Perline, S P
Trisia Santi Perline, S P Trisia Santi Perline, S P Hj Sumartiyah, S Pd Hj Sumartiyah, S Pd Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd H. Purnomo, S Pd H. Purnomo, S Pd	Meinemi Prihartini, S Pd H. Purnomo, S Pd Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Masripah Farikah, S Pd Masripah Farikah, S Pd H. Makedi, S Ag H. Makedi, S Ag	Rohimah, S Pd Hj Sumartiyah, S Pd H. Purnomo, S Pd H. Purnomo, S Pd Hj Sumartiyah, S Pd Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd	Rohimah, S Pd Masripah Farikah, S Pd H. Makedi, S Ag H. Makedi, S Ag Kasminingsih, S Pd Kasminingsih, S Pd H. Purnomo, S Pd H. Purnomo, S Pd	Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd Ummu Hanik, S Pd Ummu Hanik, S Pd Iyaa, S Pd Iyaa, S Pd Sopii, S Pd Sopii, S Pd	Andriyani, S Koom Andriyani, S Koom Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd H. Makedi, S Ag H. Purnomo, S Pd Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd	Iyaa, S Pd Iyaa, S Pd Iyaa, S Pd Iyaa, S Pd H. Makedi, S Ag Ummu Hanik, S Pd Ummu Hanik, S Pd Masripah Farikah, S Pd	H. Makedi, S Ag H. Makedi, S Ag H. Makedi, S Ag Betty Endang Prammni, S Pd Betty Endang Prammni, S Pd Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd Ummu Hanik, S Pd
Trisia Santi Perline, S P Kasminingsih, S Pd Kasminingsih, S Pd Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd Hj Sumartiyah, S Pd Hj Sumartiyah, S Pd Andik Susanto, S Pd	Hj Sumartiyah, S Pd H. Makedi, S Ag Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Meinemi Prihartini, S Pd Meinemi Prihartini, S Pd Andik Susanto, S Pd Andik Susanto, S Pd Trisia Santi Perline, S P	Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd H. Makedi, S Ag Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Meinemi Prihartini, S Pd Trisia Santi Perline, S P Trisia Santi Perline, S P Hj Sumartiyah, S Pd	H. Purnomo, S Pd Hj Sumartiyah, S Pd Hj Sumartiyah, S Pd Hj Sumartiyah, S Pd Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Oktriyanas Ricke V, S Pd	Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd Trisia Santi Perline, S P Trisia Santi Perline, S P Andriyani, S Koom Andriyani, S Koom Yayuk Purwantini, S Pd Iyaa, S Pd	H. Purnomo, S Pd H. Purnomo, S Pd Iyaa, S Pd Iyaa, S Pd Yayuk Purwantini, S Pd Rohimah, S Pd Iyaa, S Pd	Masripah Farikah, S Pd Rohimah, S Pd Rohimah, S Pd Andriyani, S Koom Andriyani, S Koom Yayuk Purwantini, S Pd Iyaa, S Pd H. Purnomo, S Pd	Ummu Hanik, S Pd Ummu Hanik, S Pd Iyaa, S Pd Iyaa, S Pd Iyaa, S Pd H. Purnomo, S Pd Trisia Santi Perline, S P
Andik Susanto, S Pd Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Meinemi Prihartini, S Pd	Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd Trisia Santi Perline, S P Trisia Santi Perline, S P	Hj Sumartiyah, S Pd Masripah Farikah, S Pd Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd	Rohimah, S Pd Rohimah, S Pd Kasminingsih, S Pd Kasminingsih, S Pd	Iyaa, S Pd Masripah Farikah, S Pd Rohimah, S Pd Rohimah, S Pd	Iyaa, S Pd Ummu Hanik, S Pd Ummu Hanik, S Pd H. Makedi, S Ag	Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd H. Purnomo, S Pd H. Purnomo, S Pd	Trisia Santi Perline, S P Yayuk Purwantini, S Pd Yayuk Purwantini, S Pd Betty Endang Prammni, S Pd
Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Oktriyanas Ricke V, S Pd Oktriyanas Ricke V, S Pd H. Makedi, S Ag H. Makedi, S Ag	Erni Puji Lestari, S Pd Erni Puji Lestari, S Pd Kasminingsih, S Pd Kasminingsih, S Pd Hj Sumartiyah, S Pd Hj Sumartiyah, S Pd	Kasminingsih, S Pd Kasminingsih, S Pd Rohimah, S Pd Rohimah, S Pd Kasminingsih, S Pd Kasminingsih, S Pd	Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd Rohimah, S Pd Rohimah, S Pd Hj Rahma Murwani, S Pd Hj Rahma Murwani, S Pd	Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd H. Makedi, S Ag Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd H. Purnomo, S Pd	H. Makedi, S Ag Trisia Santi Perline, S P Trisia Santi Perline, S P Masripah Farikah, S Pd Masripah Farikah, S Pd	Rohimah, S Pd Yayuk Purwantini, S Pd Sopii, S Pd Sopii, S Pd Trisia Santi Perline, S P Trisia Santi Perline, S P	Betty Endang Prammni, S Pd Masripah Farikah, S Pd Iyaa, S Pd Iyaa, S Pd Kuwat Setyobudi, S Pd Kuwat Setyobudi, S Pd

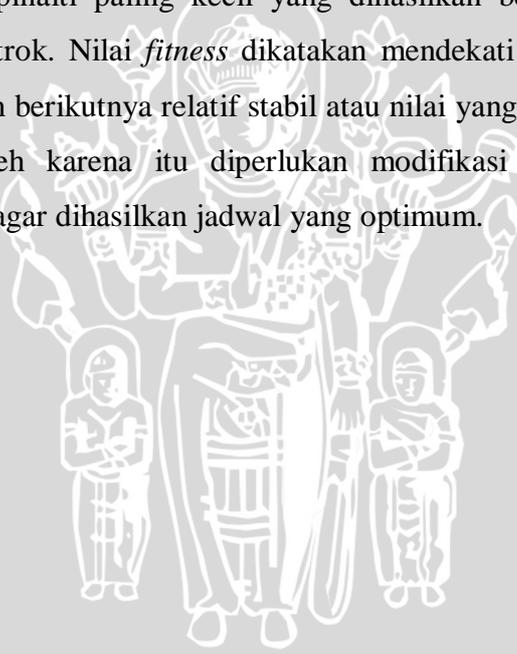
9A	9B	9C	9D	9E	9F	9G	9H
Ummah Hanik, S Pd	Setyo Purwastanti, S Pd	Bety Endang Pranamni, S Pd	Arisanoyo Tri Utomo	Sopii, S Pd	Trisni Senti Perlina, S P	Nur Fadila, S Pd	Tutik Herawati, S Pd
Ummah Hanik, S Pd	Setyo Purwastanti, S Pd	Bety Endang Pranamni, S Pd	Iwan Winardi, S Pd	Sopii, S Pd	Ratno Anggreani, S Pd	Hj Rahma Murwani, S Pd	Iyaa, S Pd
Ummah Hanik, S Pd	Maria Uifah, S Pd	Bety Endang Pranamni, S Pd	Arisanoyo Tri Utomo	Iwan Winardi, S Pd	Setyo Purwastanti, S Pd	Hj Rahma Murwani, S Pd	Iyaa, S Pd
Nur Farida, S Pd	Maria Uifah, S Pd	Nur Farida, S Pd	Arisanoyo Tri Utomo	Siti Kereni, S Pd	Misnan	Hj Rahma Murwani, S Pd	Tutik Herawati, S Pd
Nur Farida, S Pd	Setyo Purwastanti, S Pd	Nur Farida, S Pd	Misnan	Siti Kereni, S Pd	Misnan	Hj Rahma Murwani, S Pd	Tutik Herawati, S Pd
Nur Farida, S Pd	Setyo Purwastanti, S Pd	Arisanoyo Tri Utomo	Misnan	Siti Kereni, S Pd	Misnan	Tutik Herawati, S Pd	Tutik Herawati, S Pd
Yayuk Purwastanti, S Pd	Bety Endang Pranamni, S Pd	Maria Uifah, S Pd	Misnan	Setyo Purwastanti, S Pd	Andriyati, S Koon	Tutik Herawati, S Pd	Hj Luhuk Mufidhah, M Pd
Yayuk Purwastanti, S Pd	Bety Endang Pranamni, S Pd	Maria Uifah, S Pd	Arisanoyo Tri Utomo	Setyo Purwastanti, S Pd	Andriyati, S Koon	Nur Farida, S Pd	Tutik Herawati, S Pd
Iwan Winardi, S Pd	Ummah Hanik, S Pd	Maria Uifah, S Pd	Ratno Anggreani, S Pd	Iwan Winardi, S Pd	Dra Eveline, M Pd	Sopii, S Pd	Tutik Herawati, S Pd
Iwan Winardi, S Pd	Ummah Hanik, S Pd	Misnan	Ratno Anggreani, S Pd	Setyo Purwastanti, S Pd	Dra Eveline, M Pd	Sopii, S Pd	Iyaa, S Pd
Arisanoyo Tri Utomo	Iwan Winardi, S Pd	Misnan	Setyo Purwastanti, S Pd	Nurbudi Harsono	Nur Farida, S Pd	Nur Farida, S Pd	Ratno Anggreani, S Pd
Arisanoyo Tri Utomo	Andriyati, S Koon	Setyo Purwastanti, S Pd	Maria Uifah, S Pd	Ratno Anggreani, S Pd	Nur Farida, S Pd	Tutik Herawati, S Pd	Ratno Anggreani, S Pd
Yayuk Purwastanti, S Pd	Kunvat Setyobudi, S Pd	Setyo Purwastanti, S Pd	Maria Uifah, S Pd	Ratno Anggreani, S Pd	Nur Farida, S Pd	Tutik Herawati, S Pd	Nur Farida, S Pd
Bety Endang Pranamni, S Pd	Kunvat Setyobudi, S Pd	Iwan Winardi, S Pd	Maria Uifah, S Pd	Setyo Purwastanti, S Pd	Tutik Herawati, S Pd	Nur Farida, S Pd	Masripah Farikh, S Pd
Arisanoyo Tri Utomo	Maria Uifah, S Pd	Iwan Winardi, S Pd	Setyo Purwastanti, S Pd	Nurbudi Harsono	Misnan	Dian Estash Winjeng, S	Tutik Herawati, S Pd
Andriyati, S Koon	Maria Uifah, S Pd	Setyo Purwastanti, S Pd	Iwan Winardi, S Pd	Dian Estash Winjeng, S	Tutik Herawati, S Pd	Oktriyana Rickle V, S Pd	Tutik Herawati, S Pd
Andriyati, S Koon	Sopii, S Pd	Arisanoyo Tri Utomo	Bety Endang Pranamni, S Pd	Misnan	Tutik Herawati, S Pd	Iwan Winardi, S Pd	Trisni Senti Perlina, S P
Kunvat Setyobudi, S Pd	Setyo Purwastanti, S Pd	Arisanoyo Tri Utomo	Bety Endang Pranamni, S Pd	Nur Farida, S Pd	Dra Eveline, M Pd	Tutik Herawati, S Pd	Ratno Anggreani, S Pd
Kunvat Setyobudi, S Pd	Iwan Winardi, S Pd	Arisanoyo Tri Utomo	Andriyati, S Koon	Nur Farida, S Pd	Dra Eveline, M Pd	Ratno Anggreani, S Pd	Ratno Anggreani, S Pd
Kunvat Setyobudi, S Pd	Iwan Winardi, S Pd	Maria Uifah, S Pd	Andriyati, S Koon	Ratno Anggreani, S Pd	Dra Eveline, M Pd	Ratno Anggreani, S Pd	Trisni Senti Perlina, S P
Sopii, S Pd	Bety Endang Pranamni, S Pd	Maria Uifah, S Pd	Nur Farida, S Pd	Ratno Anggreani, S Pd	Nur Farida, S Pd	Misnan	Oktriyana Rickle V, S Pd
Arisanoyo Tri Utomo	Arisanoyo Tri Utomo	Sopii, S Pd	Bety Endang Pranamni, S Pd	Siti Kereni, S Pd	Iwan Winardi, S Pd	Misnan	Oktriyana Rickle V, S Pd
Arisanoyo Tri Utomo	Ratno Anggreani, S Pd	Sopii, S Pd	Bety Endang Pranamni, S Pd	Siti Kereni, S Pd	Iwan Winardi, S Pd	Ratno Anggreani, S Pd	Tutik Herawati, S Pd
Maria Uifah, S Pd	Ummah Hanik, S Pd	Arisanoyo Tri Utomo	Setyo Purwastanti, S Pd	Siti Kereni, S Pd	Ratno Anggreani, S Pd	Iwan Winardi, S Pd	Masripah Farikh, S Pd
Maria Uifah, S Pd	Ummah Hanik, S Pd	Arisanoyo Tri Utomo	Setyo Purwastanti, S Pd	Siti Kereni, S Pd	Ratno Anggreani, S Pd	Tutik Herawati, S Pd	Iwan Winardi, S Pd
Kunvat Setyobudi, S Pd	Arisanoyo Tri Utomo	Maria Uifah, S Pd	Maria Uifah, S Pd	Nur Farida, S Pd	Setyo Purwastanti, S Pd	Tutik Herawati, S Pd	Iwan Winardi, S Pd
Kunvat Setyobudi, S Pd	Arisanoyo Tri Utomo	Maria Uifah, S Pd	Misnan	Nur Farida, S Pd	Setyo Purwastanti, S Pd	Ratno Anggreani, S Pd	Tutik Herawati, S Pd
Nur Farida, S Pd	Kunvat Setyobudi, S Pd	Nur Farida, S Pd	Misnan	Tutik Herawati, S Pd	Masripah Farikh, S Pd	Ratno Anggreani, S Pd	Tutik Herawati, S Pd
Maria Uifah, S Pd	Bety Endang Pranamni, S Pd	Ratno Anggreani, S Pd	Nur Farida, S Pd	Tutik Herawati, S Pd	Masripah Farikh, S Pd	Misnan	Sopii, S Pd
Maria Uifah, S Pd	Bety Endang Pranamni, S Pd	Ratno Anggreani, S Pd	Nur Farida, S Pd	Tutik Herawati, S Pd	Nur Farida, S Pd	Misnan	Sopii, S Pd
Bety Endang Pranamni, S Pd	Maria Uifah, S Pd	Ratno Anggreani, S Pd	Sopii, S Pd	Tutik Herawati, S Pd	Nur Farida, S Pd	Nurbudi Harsono	Misnan
Yayuk Purwastanti, S Pd	Maria Uifah, S Pd	Ratno Anggreani, S Pd	Sopii, S Pd	Tutik Herawati, S Pd	Trisni Senti Perlina, S P	Nurbudi Harsono	Misnan
Bety Endang Pranamni, S Pd	Arisanoyo Tri Utomo	Iwan Winardi, S Pd	Ratno Anggreani, S Pd	Andriyati, S Koon	Trisni Senti Perlina, S P	Dian Estash Winjeng, S	Misnan
Bety Endang Pranamni, S Pd	Arisanoyo Tri Utomo	Andriyati, S Koon	Ratno Anggreani, S Pd	Andriyati, S Koon	Setyo Purwastanti, S Pd	Dian Estash Winjeng, S	Misnan
Arisanoyo Tri Utomo	Nur Farida, S Pd	Andriyati, S Koon	Maria Uifah, S Pd	Misnan	Setyo Purwastanti, S Pd	Tutik Herawati, S Pd	Hj Luhuk Mufidhah, M Pd
Maria Uifah, S Pd	Arisanoyo Tri Utomo	Setyo Purwastanti, S Pd	Maria Uifah, S Pd	Misnan	Iwan Winardi, S Pd	Nur Farida, S Pd	Hj Luhuk Mufidhah, M Pd
Maria Uifah, S Pd	Nur Farida, S Pd	Bety Endang Pranamni, S Pd	Arisanoyo Tri Utomo	Dian Estash Winjeng, S	Sopii, S Pd	Nur Farida, S Pd	Iyaa, S Pd
Iwan Winardi, S Pd	Nur Farida, S Pd	Bety Endang Pranamni, S Pd	Arisanoyo Tri Utomo	Dian Estash Winjeng, S	Sopii, S Pd	Hj Rahma Murwani, S Pd	Iyaa, S Pd
Iwan Winardi, S Pd	Nur Farida, S Pd	Nur Farida, S Pd	Arisanoyo Tri Utomo	Dian Estash Winjeng, S	Sopii, S Pd	Hj Rahma Murwani, S Pd	Iyaa, S Pd

Gambar 6.8 Hasil Penjadwalan Mata Pelajaran

Dari gambar ini terlihat bahwa masih ada beberapa guru yang terjadwal di dua kelas berbeda pada saat yang sama. Hal ini menunjukkan bahwa penjadwalan yang melibatkan banyak kelas dan sekaligus banyak guru merupakan permasalahan yang rumit dan tidak mudah dicari solusinya. Telah dibahas sebelumnya jika ada guru yang mengajar di kelas berbeda pada waktu yang sama termasuk bentrok dan dikenakan nilai pinalti sebanyak 1 kemudian dihitung nilai *fitness* menggunakan Persamaan 2-1.

Pada penelitian ini nilai *fitness* terbaik yang didapatkan dari pengujian ulang menggunakan parameter terbaik yaitu 0.0081 dengan jumlah bentrok 123. Solusi penjadwalan ini didapatkan dengan nilai parameter ukuran populasi = 60, ukuran generasi = 80, dan kombinasi *cr* dan *mr* yaitu 0.4 dan 0.6. Hasil solusi penjadwalan dari parameter terbaik ini masih memiliki jumlah bentrok yang banyak dikarenakan proses algoritma genetika memiliki sifat *stochastic* dimana setiap kali program dijalankan menghasilkan solusi yang berbeda-beda dan juga mendapatkan pengaruh dari ukuran parameter yang dimasukkan.

Dalam menentukan kualitas solusi penjadwalan terbaik dilihat dari nilai *fitness* jika nilai *fitness* yang dihasilkan sudah lebih besar dari sebelumnya maka nilai tersebut sudah mendekati optimal. Nilai *fitness* tertinggi adalah nilai *fitness* yang memiliki nilai pinalti paling kecil yang dihasilkan berdasarkan jumlah pelanggaran atau bentrok. Nilai *fitness* dikatakan mendekati optimal jika nilai *fitness* yang dihasilkan berikutnya relatif stabil atau nilai yang dihasilkan hampir mendekati sama. Oleh karena itu diperlukan modifikasi pada mekanisme *crossover* dan mutasi agar dihasilkan jadwal yang optimum.



## BAB VII PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian dalam menerapkan algoritma genetika untuk penjadwalan mata pelajaran, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Algoritma genetika dapat diterapkan pada permasalahan optimasi penjadwalan dengan menggunakan representasi permutasi berbasis kode guru yang sudah diatur sesuai dengan porsi jam pelajaran dan porsi jam mengajar guru. Metode *crossover* yang digunakan yaitu *one-cut point* dan untuk proses mutasinya menggunakan metode *reciprocal exchange mutation*. Sedangkan metode seleksi yang digunakan yaitu *elitism* yang menghasilkan nilai *fitness* paling optimal dari generasi terakhir.
2. Bentuk representasi kromosom yang digunakan memiliki panjang kromosom pada interval [1...552] yang nantinya hasil dari representasi kromosomnya berupa *random*. Untuk menghitung nilai *fitness* berdasarkan nilai bentrok pada penjadwalan jika terdapat satu guru yang mengajar pada waktu yang sama dengan kelas berbeda maka mendapatkan nilai pinalti 1 kemudian dihitung menggunakan rumus *fitness*. Jika nilai *fitness* yang dihasilkan semakin besar maka solusi yang dihasilkan semakin baik sedangkan jika semakin kecil nilai *fitness*nya maka semakin buruk solusi yang dihasilkan.
3. Dalam menentukan parameter algoritma genetika yang digunakan pada penjadwalan mata pelajaran maka dilakukan pengujian parameter yang terdiri dari jumlah generasi, jumlah populasi (*popsize*) dan kombinasi *crossover rate* (*cr*) dan *mutation rate* (*mr*). Hasil dari pengujian merupakan parameter dengan nilai rata-rata *fitness* tertinggi dari percobaan yang dilakukan sebanyak 10 kali. Hasilnya didapatkan jumlah generasi 80, jumlah populasi 60 dan kombinasi *crossover rate* dan *mutation rate* 0.4:0.6 yang memiliki nilai *fitness* 0.0081 dengan jumlah bentrok 123. Namun nilai *fitness* yang dihasilkan masih belum optimal karena dengan

panjang kromosom hingga 552 membutuhkan waktu komputasi yang lama dan hasil *random* dari proses algoritma genetika yang lebih bervariasi sehingga sulit untuk mendapatkan nilai *fitness* yang optimal. Berdasarkan hasil pengujian maka didapatkan kesimpulan bahwa nilai parameter algoritma genetika berpengaruh terhadap hasil optimasi. Dengan ukuran parameter yang kecil maka akan menyebabkan area pencarian algoritma genetika semakin sempit sedangkan jika ukuran parameter terlalu besar maka akan membutuhkan waktu komputasi lebih lama dan tidak menjamin akan menghasilkan nilai yang optimal.

4. Untuk menentukan kualitas dari solusi penjadwalan yang dihasilkan diukur dari nilai *fitness*-nya. Berdasarkan hasil pengujian maka didapatkan nilai *fitness* tertinggi yaitu 0.0081 dari parameter terbaik tetapi solusi yang dihasilkan masih belum optimal karena algoritma genetika memiliki sifat *stochastic* dimana setiap kali program dijalankan menghasilkan solusi yang berbeda-beda dan juga mendapatkan pengaruh dari ukuran parameter yang dimasukkan.

## 7.2 Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa saran yang diberikan untuk memperbaiki penelitian ini :

1. Bila menggunakan aplikasi ini diperlukan pengujian parameter lebih banyak lagi dan dapat digunakan metode lain sebagai *pre-processing* agar bentrok dapat diminimalkan sehingga dapat menghasilkan solusi penjadwalan mata pelajaran paling optimal. Selain itu juga dapat ditambahkan strategi perbaikan (*repair*) terhadap kromosom ketika solusi yang dihasilkan memiliki nilai *fitness* yang jauh dari optimal.
2. Jika ingin menggunakan algoritma genetika pada penjadwalan bisa mengaplikasikan dengan bahasa pemrograman lain yang mampu melakukan proses komputasi yang lebih cepat.
3. Dapat menggunakan metode *crossover*, mutasi atau seleksi lainnya dalam menerapkan algoritma genetika untuk mendapatkan solusi yang lebih bervariasi dalam kasus penjadwalan mata pelajaran.

## DAFTAR PUSTAKA

- Adamanti, Justina. 2002. *Penyelesaian Masalah Penjadwalan Mata Kuliah Di Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Gadjah Mada Dengan Menggunakan Algoritma Genetika*. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Gadjah Mada. Yogyakarta
- Ariani, dkk. 2007. *Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika PENS Dengan Menggunakan Algoritma Particle Swarm Optimization (PSO)*. Politeknik Elektronika Negeri Surabaya. Institut Teknologi Sepuluh Nopember. Surabaya
- Bangun, dkk. 2012. *Penerapan Konsep Algoritma Genetika untuk Penjadwalan Kegiatan Perkuliahan Semester Ganjil Kurikulum 2012 di Jurusan Matematika FMIPA UNSRI*. Jurusan Matematika. Universitas Sriwijaya. Sumatera Selatan
- Basuki, Ahmad. 2003. *Algoritma Genetika, Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning*. Politeknik Elektronika Negeri Surabaya PENS-ITS. Surabaya
- E.K. Burke, S. Petrovic. 2002. *Recent Research Directions In Automated Timetabling*. European Journal of Operational Research, 140(2), pp 266-280
- Fitri R, dkk. 2004. *Penjadwalan Perkuliahan Dengan Pengujian Tabel Waktu (Time-Table) Menggunakan Algoritma Genetika*. Jurusan Teknik informatika. Universitas Komputer Indonesia. Bandung
- Gunawan, dkk. 2004. *Applying Metaheuristics For The Course Scheduling Problem*. Department of Industrial Engineering. University of Surabaya. Indonesia
- Haupt, Randy L., dan Haupt, Sue E. 2004: *Practical Genetic Algorithms*. Second edition. John Wiley & Sons, Inc., New Jersey
- Juniawati. 2003. *Implementasi Algoritma Genetika untuk Mencari Volume Terbesar Bangun Kotak Tanpa Tutup Dari Suatu Bidang Datar Segi Empat*. Jurnal Ilmiah LPPM. Universitas Surabaya. Surabaya

- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta
- Mahmudy, Wayan Firdaus. 2013. *Modul Algoritma Evolusi*. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang
- Mahmudy, Wayan Firdaus. 2014. *Algoritma Evolusi*. Program Teknologi Informasi dan Ilmu Komputer. Malang : Universitas Brawijaya.
- Mahmudy, WF, Marian, RM & Luong, LHS 2012b. *Solving Part Type Selection And Loading Problem In Flexible Manufacturing System Using Real Coded Genetic Algorithms – Part I: Modelling*. International Conference on Control, Automation and Robotics, Singapore, 12-14 September, World Academy of Science, Engineering and Technology, pp. 699-705.
- Mahmudy, WF, Marian, RM & Luong, LHS. 2014. *Hybrid Genetic Algorithms For Part Type Selection And Machine Loading Problems With Alternative Production Plans In Flexible Manufacturing System*. ECTI Transactions on Computer and Information Technology (ECTI-CIT), vol. 8, no. 1, pp. 80-93.
- Mawaddah, NK & Mahmudy, WF. 2006. *Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika*. Kursor, vol. 2, no. 2, pp. 1-8. Jurusan Matematika. FMIPA. Universitas Brawijaya. Malang
- Nugraha, Ivan. 2008. *Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar*. Program Studi Teknik Informatika. Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Bandung
- Salimi, M.Alifie. 2006. *Sistem Informasi Penjadwalan Mata Pelajaran pada Sekolah Menengah Umum Dengan Metode Algoritma Genetika*. Jurusan Sistem Informasi. Sekolah Tinggi Manajemen Informatika dan Teknik Komputer. Surabaya
- Sam'ani. 2012. *Rancang Bangun Sistem Penjadwalan Perkuliahan Dan Ujian Akhir Semester Dengan Pendekatan Algoritma Genetika*. Program Studi Magister Sistem Informasi. Program Pascasarjana. Universitas Diponegoro. Semarang

- S'andor, dkk. 2005. *Genetic Algorithms in Timetabling , A New Approach*. Budapest University of Technology and Economics. Department of Measurement and Information Systems. Budapest. Hungary
- Sundarningsih, D, Mahmudy, WF & Sutrisno. 2015. *Penerapan algoritma genetika untuk optimasi vehicle routing problem with time window (VRPTW) : Studi kasus air minum kemasan*. DORO : Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 9.
- Suyanto. 2011. *Artificial intelligence*. Informatika. Bandung

