

**OTOMATISASI SISTEM PENJADWALAN PERKULIAHAN
MENGUNAKAN ALGORITMA GENETIKA**

SKRIPSI



**Disusun Oleh :
ARIF CAHYONO
08.18.043**

**PRODI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN

**OTOMATISASI SISTEM PENJADWALAN PERKULIAHAN
MENGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Informatika Strata Satu (S-1)*

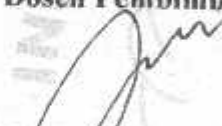
Disusun Oleh :


**ARIF CAHYONO
NIM : 08.18.043**

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Dr. Eng. Aryuanto S, ST, MT
NIP. P.1030800417


Ahmad Faisol, ST
NIP. P. 1031000431

Mengetahui

Ketua Program Studi Teknik Informatika S-1




Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

**PRODI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

OTOMATISASI SISTEM PENJADWALAN PERKULIAHAN MENGUNAKAN ALGORITMA GENETIKA

Arif Cahyono (08.18.043)

Program Studi Teknik Informatika S-1,
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Email: arif_cahyana90@yahoo.com

Dosen Pembimbing: I. Dr. Eng. Aryuanto S, ST , MT
II. Ahmad Faisol, ST

Abstrak

Penyusunan penjadwalan mata kuliah secara optimal sangatlah penting bagi kelancaran proses belajar mengajar. Sering terjadinya bentrok, baik bentrokan jadwal yang terjadi pada mata kuliah yang diambil maupun pada dosen yang mengajar, serta pada keterbatasan ruang mengakibatkan tidak efektifnya proses belajar mengajar. Oleh karena itu dalam makalah ini penulis, melakukan riset penerapan metode Algoritma Genetika pada permasalahan penjadwalan perkuliahan di ITN MALANG. Selama ini di ITN MALANG untuk penjadwalan mata kuliahnya masih dilakukan secara manual, sedangkan untuk membagi dosen keahlian dengan bidang mata kuliahnya membutuhkan proses yang cukup rumit. Apalagi jika proses penjadwalan dilakukan secara manual dengan data yang besar maka permasalahan akan jauh lebih kompleks nantinya. Maka dari itu diperlukan sebuah solusi yang bersifat heuristik dan terotomatisasi sehingga dapat mempersingkat waktu yang ada dalam proses penyusunan penjadwalan perkuliahan.

Hasil dari pengujian, dengan menggunakan metode GA penulis berhasil meminimalisir kemungkinan terjadinya bentrokan jadwal dengan tingkat keberhasilan >96% dengan variasi pembangkitan populasi sebanyak 10 populasi, pembangkitan jumlah induk/parent sebanyak 100 parent, pembangkitan generasi sebanyak 50 generasi, dengan durasi waktu proses <5 menit.

Kata kunci: algoritma genetika, sistem penjadwalan..

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : ARIF CAHYONO

NIM : 08.18.043

Program Studi : Teknik Informatika S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, Februari 2013

Yang membuat Pernyataan,



Arif Cahyono
NIM. 08.18.043

KATA PENGANTAR

Puji syukur penulis panjatkan ke Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan skripsi dengan judul **“OTOMATISASI SISTEM PENJADWALAN PERKULIAHAN MENGGUNAKAN ALGORITMA GENETIKA”** dengan lancar. Skripsi merupakan persyaratan kelulusan di Program Studi Teknik Informatika S-1 ITN Malang dan untuk mencapai gelar Sarjana Komputer.

Keberhasilan penyelesaian laporan skripsi tidak lepas dari dukungan dan bantuan berbagai pihak, untuk itu penyusun menyampaikan terima kasih kepada:

1. Bapak Ir. Soeparno Djiwo, MT. selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Joseph Dedi Irawan, ST., MT. selaku Ketua Program Studi Teknik Informatika S-1
4. Bapak Dr. Eng. Aryuanto S, ST, MT. selaku Dosen Pembimbing I
5. Bapak Ahmad Faisol, ST. selaku Dosen Pembimbing II
6. Orangtua yang selalu memberikan semangat dan motivasi selama menyusun skripsi
7. Teman-teman dan semua yang telah membantu dalam menyelesaikan penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Februari 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
SURAT PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Manfaat.....	2
1.5. Batasan Masalah.....	3
1.6. Metodologi.....	3
1.7. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	
2.1. Penjadwalan	5
2.1.1. Penjadwalan Perkuliahan.....	5
2.1.2. Penjadwalan Perkuliahan di Tehnik Informatika.....	7
2.1.3. Sampel Data.....	8
2.1.4. Faktor-Faktor Yang Mempengaruhi Sistem Penjadwalan.....	8
2.1.5. Batasan(Hard constrain), Aturan-Aturan Pada Sistem Penjadwalan.....	9
2.2. Kecerdasan Buatan (Articial Intelegence)	9
2.3. Metode Algoritma Genetika	12
2.3.1. Pengeruan Metode Algoritma Genetika	12

2.3.2.	Hal – Hal yang harus dilakukan dalam menggunakan metode algoritma genetika adalah:	13
2.3.3.	Pengertian Individu.....	13
2.3.4.	Nilai Fitness	14
2.3.5.	Siklus Algoritma Genetika.....	15
2.4.	Pemrograman Visual Basic 6.0	19
2.4.1.	Keistimewaan Microsoft Visual Basic.....	19
2.5.	Microsoft Access.....	20

BAB III ANALISA DAN PERANCANGAN

3.1.	Deskripsi Sistem Secara Umum	24
3.2.	Perancangan Sistem.....	24
3.2.1.	Database.....	25
3.2.1.1.	Relasi Tabel.....	25
3.2.1.2.	Struktur Tabel.....	26
3.2.2.	Dekomposisi Basis Data.....	28
3.2.2.1.	Perancangan Basis Data.....	28
3.2.2.2.	Fitness.....	30
3.2.3.	Flowchart sistem.....	31
3.3.	Perancangan Desain Aplikasi.....	32
3.3.1.	Desain Tampilan Form Awal.....	32
3.3.2.	Tampilan Desain Form Utama.....	33
3.4.	Kebutuhan Sistem.....	34

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1.	Deskripsi Aplikasi.....	35
4.1.1.	Menu Input Data.....	35
4.1.2.	Menu GA Process.....	36
4.1.3.	Konfigurasi.....	37
4.2.	Penerapan Aplikasi.....	37

4.3.	Pengujian.....	38
4.3.1.	Pengujian Pertama.....	38
4.3.2.	Pengujian Kedua.....	39
BAB V PENUTUP		
5.1.	Kesimpulan	41
5.2.	Saran.....	42
DAFTAR PUSTAKA		43

DAFTAR TABEL

Tabel 2.1. Jam Perkuliahan Reguler.....	7
Tabel 3.1. Tabel Dosen.....	26
Tabel 3.2. Tabel Mata kuliah.....	26
Tabel 3.3. Tabel Ruang.....	26
Tabel 3.4. Tabel Jam Kuliah.....	26
Tabel 3.5. Tabel Hari.....	27
Tabel 3.6. Tabel Transaksi.....	27
Tabel 3.7. Kebutuhan Sistem.....	34
Tabel 4.1. Konfigurasi Nilai Default Pada Menu GA Process.....	37
Tabel 4.2. Pengujian Jadwal Semester Ganjil.....	39
Tabel 4.3. Pengujian Jadwal Semester Genap.....	40

DAFTAR GAMBAR

Gambar 2.1. Sampel Data.....	8
Gambar 2.2. Bidang-Bidang Tugas (Task Domains) Dari AI.....	11
Gambar 2.3. Penerapan Konsep Kecerdasan Buatan di Komputer.....	12
Gambar 2.4. Siklus Proses Evolusi.....	14
Gambar 2.5. Siklus Algoritma Genetika.....	15
Gambar 2.6. Pertukaran Gen 1 Titik.....	17
Gambar 2.7. Pertukaran Gen 2 Titik.....	17
Gambar 2.8. Mutasi Tingkat Kromosom.....	18
Gambar 2.9. Mutasi Gen.....	18
Gambar 2.10. Mutasi tingkat Bit.....	19
Gambar 3.1. Diagram Blok Sistem.....	25
Gambar 3.2. Relasi Sistem Penjadwalan.....	25
Gambar 3.3. Tabel Transaksi.....	28
Gambar 3.4. Proses Dekomposisi Basis Data.....	29
Gambar 3.5. Ilustrasi Perhitungan Total Fitness Ruang dan Dosen.....	30
Gambar 3.6. Flowchart sistem.....	31
Gambar 3.7. Tampilan Form Awal.....	33
Gambar 3.8. Tampilan Desain Form Utama.....	33
Gambar 4.1. Tampilan Halaman Utama.....	35
Gambar 4.2. Tampilan Menu Input Data.....	35
Gambar 4.3. Data Dosen.....	35
Gambar 4.4. Data Mata Kuliah.....	36
Gambar 4.5. Data Ruang.....	36
Gambar 4.6. Menu GA Process.....	36
Gambar 4.7. Jadwal Hasil Setelah Proses.....	38

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di setiap awal semester perkuliahan setiap universitas atau institusi pendidikan, seringkali membuat jadwal yang biasanya selalu berubah – ubah atau bisa dibilang dengan jadwal sementara yang nantinya akan diganti saat perkuliahan dimulai mengingat keterbatasan sumber daya yang ada, seperti kurangnya ruangan atau kelas, pengajar yang jumlahnya kurang dari mata kuliah yang ada sehingga 1 dosen bisa mengajar lebih dari 1 mata kuliah, sehingga membuat kurang optimalnya penyusunan jadwal perkuliahan. Penyusunan penjadwalan mata kuliah secara optimal sangatlah penting bagi kelancaran proses belajar mengajar. Sering terjadinya bentrok, baik bentrokan jadwal yang terjadi pada mata kuliah yang diambil maupun pada dosen yang mengajar, serta pada keterbatasan ruang mengakibatkan tidak efektifnya proses belajar mengajar.

Selama ini di ITN MALANG untuk penjadwalan mata kuliahnya masih dilakukan secara manual, sedangkan untuk membagi dosen keahlian dengan bidang mata kuliahnya membutuhkan proses yang cukup rumit. Apalagi jika proses penjadwalan dilakukan secara manual dengan data yang besar maka permasalahan akan jauh lebih kompleks nantinya.

Karena semua kegiatan belajar mengajar antara dosen dengan mahasiswa bergantung pada jadwal perkuliahan yang ada, maka dari permasalahan tersebut diperlukan solusi yang bersifat bisa mencari, menyelidiki sendiri/heuristik dan terotomatisasi sehingga dapat mempersingkat waktu yang ada dalam menyusun penjadwalan perkuliahan. Sehingga jadwal perkuliahan bisa disusun secara optimal pada awal semester, supaya nantinya tidak mengganggu proses belajar mengajar.

Berdasarkan uraian diatas penulis ingin melakukan riset penerapan metode *Genetic Algorithm* (GA) pada sebuah permasalahan sistem penjadwalan secara otomatisasi sebagai metode optimasi. Maka dari itu penulis mengambil topik skripsi

dengan judul "OTOMATISASI SISTEM PENJADWALAN PERKULIAHAN MENGGUNAKAN METODE ALGORITMA GENETIKA.

1.2 Perumusan Masalah

Berdasarkan uraian diatas, maka yang menjadi permasalahan adalah: bagaimana merancang dan membuat aplikasi otomatisasi sistem penjadwalan perkuliahan , serta merancang pengkodean/*decoding* attribute database untuk dapat diterapkan menggunakan metode GA sehingga bisa memperoleh jadwal perkuliahan yang optimal.

1.3 Tujuan

Tujuan penyusunan tugas akhir (skripsi) ini adalah :

- a. Menghasilkan suatu aplikasi yang berfungsi untuk menyusun penjadwalan perkuliahan secara optimal.
- b. Mengembangkan metode Algoritma Genetika (AG) pada permasalahan sistem penjadwalan secara otomatisasi sebagai optimasi.
- c. Merancang dan mengaplikasikan otomatisasi sistem penjadwalan dengan menggunakan bahasa pemrograman Visual Basic dan menggunakan Microsoft Access sebagai basis data, dan menggunakan metode Algoritma Genetika sebagai dasar metodenya.

1.4 Manfaat

Membantu untuk memberikan beberapa pertimbangan dalam proses pengubahan jadwal mata kuliah. Misal ada dosen yang menginginkan pergantian jadwal kuliahnya. Sistem ini akan memberikan beberapa pertimbangan seperti tersedianya jam mata kuliah yang ada atau ruang kelas yang tersedia, serta memperoleh jadwal perkuliahan yang optimal.

1.5 Batasan Masalah

Karena keterbatasan waktu dan pengetahuan penulis, maka ruang lingkup permasalahan dalam merancang aplikasi otomatisasi sistem penjadwalan perkuliahan ini antara lain :

- a. Mengambil studi kasus di jurusan Teknik Informatika INSTITUT TEKNOLOGI NASIONAL MALANG.
- b. Mata kuliah yang dijadwalkan hanya untuk jurusan Tehnik Informatika saja.
- c. Dosen tidak diperkenankan meminta jadwal mengajar dihari tertentu, dan ruangan tertentu.
- d. Dosen pengganti akan tetap berkedudukan sebagai dosen aslinya, misal dosen asli "A" dan dosen pengganti "B" maka dosen pengganti akan tetap berkedudukan sebagai dosen "A".
- e. Untuk pengecualian jam perkuliahan pada hari jum'at dilakukan secara manual, dengan cara menggeser jam perkuliahan yang seharusnya dimulai pada jam ke - 6 (11.30 - 12.20) dan jam ke - 7 (12.20 - 13.10) dipindahkan ke jam ke - 8 (13.10 - 14.00) dan jam ke - 9 (14.00 - 14.50).
- f. Studi kasus yang penulis angkat menggunakan data penjadwalan perkuliahan semester ganjil dan semester genap tahun akademik 2012/2013 di T.Informatika INSTITUT TEKNOLOGI NASIONAL MALANG.

1.6 Metodologi

Berbagai macam tahapan penyelesaian Tugas akhir ini meliputi:

- a. Pengambilan Sampel Data

Pengumpulan data yang dilakukan secara langsung dari sumbernya. Karena tugas akhir ini merupakan studi kasus Otomatisasi Sistem Penjadwalan Perkuliahan, maka data yang diperlukan adalah berupa data penjadwalan pada semester ini yang nantinya akan digunakan sebagai sampel.

b. Studi Literatur

Mencari referensi dari berbagai sumber baik berupa buku, paper, maupun data – data dari *digital library* yang terdapat di internet. Secara lebih spesifik kajian yang dipelajari pada penulisan tugas akhir ini adalah sebagai berikut :

- Kajian tentang Algoritma Genetika (AG) sebagai dasar metode yang diterapkan pada sebuah permasalahan sistem penjadwalan secara otomatisasi sebagai metode optimasi.
- Kajian mengenai bahasa pemrograman Visual Basic sebagai media perancang / pembuat aplikasi otomatisasi sistem penjadwalan perkuliahan dengan menggunakan Visual Basic 6.

c. Desain Aplikasi

Merancang sebuah sistem aplikasi yang nantinya dapat mengidentifikasi komponen-komponen sistem informasi yang akan dibuat secara rinci. Tahap ini dilakukan setelah tahap analisis sistem selesai dilaksanakan, karena hasil desain terinci akan percuma sia-sia bila tidak sesuai dengan apa yang diinginkan. Desain aplikasi ini meliputi flowchart atau teknik lain seperti sketsa dan lain-lain dalam mendukung desain system.

d. Implementasi

Secara garis besar implementasi pembuatan aplikasi ini nantinya menggunakan bahasa pemrograman Visual Basic 6, dimana aplikasi otomatisasi sistem penjadwalan ini nantinya dibuat dengan tujuan untuk mengantisipasi kemungkinan kendala yang muncul berkaitan dengan penjadwalan yang bentrok. Untuk basis datanya menggunakan Microsoft Access. Setelah sistem dianalisis dan didesain secara rinci, serta teknologi sudah diseleksi dan dipilih, maka implementasi sistem merupakan tahapan yang harus dilakukan selanjutnya. Implementasi sistem merupakan tahapan pembuatan aplikasi supaya nantinya siap untuk dioperasikan.

Tahapan implementasi ini meliputi :

- Perancangan basis data
- Menyesuaikan keterkaitan entitas – *attribute*

- Menycsuaikan basis data kedalam operator genetika

e. Ujicoba

Pada tahap ini akan dilakukan ujicoba sistem untuk mencari kekurangan atau kesalahan yang terjadi pada aplikasi ini. Semua fungsi harus diujicoba terlebih dahulu, apakah semua sistemnya sudah bisa berjalan dengan baik atau belum. Hal ini dilakukan supaya bisa memenuhi kebutuhan yang sudah didefinisikan sebelumnya.

1.7 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, makasistematika penulisan disusun sebagai berikut :

BAB I : PENDAHULUAN

Berisi latar belakang, rumusan masalah, tujuan dan manfaat, ruang lingkup masalah, metodologi dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang di lakukan.

BAB III : PERANCANGAN DAN DESAIN APLIKASI

Dalam bab ini berisi mengenai analisa kebutuhan system baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari system yang akan dibuat.

BAB IV : PENGUJIAN

Bab ini berisi tentang pengujian aplikasi, struktur, dan tampilan aplikasi.

BAB V : PENUTUP

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan mengenai teori-teori yang diperlukan dalam pengerjaan Skripsi ini. Teori-teori yang dibahas meliputi Kecerdasan Buatan(Artificial Intelegence), Algoritma Genetika, Microsoft Visual Basic 6.0, Microsoft access 2007.

2.1 Penjadwalan

Penjadwalan diperlukan jika terdapat keterbatasan sumber daya dalam suatu pekerjaan, sehingga diperlukan adanya pengaturan sumber-sumber daya tersebut secara efisien. Sehingga definisi penjadwalan adalah permasalahan dalam menentukan jadwal yang tepat atas suatu pekerjaan terhadap sumber daya yang tersedia sesuai dengan *constrain* yang harus dipenuhi. [5]

2.1.1 Penjadwalan Perkuliahan

Penjadwalan dibutuhkan pada aktivitas yang membutuhkan waktu dan memiliki keterbatasan sumber daya. Salah satu kegiatan yang membutuhkan penjadwalan adalah perkuliahan. Dalam perkuliahan terdapat sumber daya seperti mata kuliah, dosen ruangan dan waktu atau slot perkuliahan. Dari sumber daya yang tersedia ini dibuatlah jadwal pekerjaan, dalam hal ini kegiatan perkuliahan, sesuai dengan *constrain* (batasan) yang harus dipenuhi yaitu dosen tidak mengajar lebih dari satu mata kuliah yang sama, dan ruangan tidak boleh digunakan lebih dari satu perkuliahan pada waktu yang sama. [3]

Jadwal untuk perkuliahan biasanya mingguan dan berlaku untuk satu smester. Sebuah jadwal disebut lengkap jika setiap mata kuliah yang ditawarkan pada suatu smester memiliki slot kapan kuliah tersebut berlangsung, ruang, dan dosen yang mengajar. Jadwal yang lengkap belum tentu valid. Jadwal yang valid adalah jadwal yang sesuai dengan *constrain* yang harus dipenuhi agar kegiatan perkuliahan dapat benar-benar berlangsung. [3]

2.1.2 Penjadwalan Perkuliahan di Teknik Informatika

Kegiatan belajar mengajar di Teknik informatika ITN MALANG dilaksanakan enam hari dalam seminggu diantaranya hari senin sampai dengan hari sabtu, dimana dimulai pada pukul 07.00 sampai dengan pukul 17.30. Dan untuk proses penjadwalan perkuliahan pada awal semester diawali dengan pemasaran matakuliah melalui siacad. Lalu mata kuliah dibebankan kepada dosen keahlian. Menentukan jadwal mata kuliah berdasarkan jam, hari dan ruang yang tersedia dengan mempertimbangkan kesedian waktu mengajar dari dosen yang bersangkutan. Untuk hasil akhirnya berupa jadwal perkuliahan 1 semester yang disusun berdasarkan mata kuliah, kelas, hari, jam ke, dan ruang. Berikut tabel jam perkuliahan reguler yang berlaku saat ini: [ITN MALANG]

Tabel 2.1 Jam perkuliahan reguler

Jam ke	Mulai	Berakhir
1	07.00	07.50
2	07.50	08.40
3	08.50	09.40
4	09.40	10.30
5	10.40	11.30
6	11.30	12.20
7	12.30	13.10
8	13.10	14.00
9	14.10	14.50
10	15.10	15.50
11	15.50	16.40
12	16.40	17.30

2.1.3 Sampel data

Berikut merupakan sampel data yang diperoleh dari pengajaran yang akan didekomposisikan kedalam database.



DAFTAR JADUAL MENGAJAR DOSEN
PROGRAM STUDI / KONSENTRASI TEKNIK INFORMATIKA S 1
SEMESTER GANJIL, TAHUN AKADEMIK 2012/2013

HARI : Senin
TANGGAL :

NO	ID	DOSEN	MATA KULIAH	JAM	SKS	SM	KLS	R	JAM Dg Pg
1	0201	DRS. MELIK SUWANANDAR, MSED	PENDIDIKAN BAHASA INDONESIA	1-3	3	I	IF-A	ELA	
2	0201	DRS. MELIK SUWANANDAR, MSED	PENDIDIKAN BAHASA INDONESIA	1-3	3	I	IF-B	ELA	
3	0201	DRS. MELIK SUWANANDAR, MSED	PENDIDIKAN BAHASA INDONESIA	5-7	3	I	IF-C	ELA	
4	0204	DR. H. DATAL GUSTOPO, S.T	PENGANTAR	7-8	2	VI	IF-C	ELA	
5	0204	DR. H. DATAL GUSTOPO, S.T	PENGANTAR	7-8	2	VI	IF-D	ELA	
6	0600	DRS. BUDIANTO, MSF	FISIKA ALAM	4-8	2	I	IF-A	ELA	
7	0600	DRS. BUDIANTO, MSF	FISIKA ALAM	4-8	2	I	IF-B	ELA	
8	0600	DRS. BUDIANTO, MSF	FISIKA ALAM	6-8	2	I	IF-C	ELA	

Gambar 2.1 Sampel data jadwal semester sebelumnya

2.1.4 Faktor-faktor yang mempengaruhi sistem penjadwalan

Supaya suatu jadwal bias dibuat dengan benar dan optimal, berikut terdapat sejumlah faktor yang mempengaruhi prancangan sistem penjadwalan perkuliahan meliputi: [8]

1. Dosen

Dosen tidak bisa mengajar mata kuliah lebih dari satu dalam satu waktu. Untuk mensiasati ketersediaan waktu dosen diperkenankan untuk meminta waktu sesuai dengan ketersediaan waktu yang ada (yang dikehendaki).

2. Mata kuliah

Mata kuliah memiliki semester di tiap mata kuliah itu dijadwalkan, maka perlu adanya aturan yang membatasi penjadwalan mata kuliah, supaya nantinya mata kuliah tersebut sesuai dengan aturan-aturan penjadwalan. Disini penulis

menggunakan data penjadwalan Teknik Informatika ITN Malang semester ganjil, tahun akademik 2012/2013.

3. Ruang

Jadwal hanya mengakomodasi, memaksimalkan ruangan yang ada, kapasitas semua ruangan dianggap sama.

4. Waktu

Waktu yang tersedia harus bisa dimanfaatkan dengan baik sehingga dapat tercipta sebuah penjadwalan perkuliahan yang optimal.

2.1.5 Batasan (*Hard Constraint*), aturan-aturan pada sistem penjadwalan

1. Dosen tidak diperbolehkan mengajar di ruang yang berbeda pada waktu yang sama. [8]
2. Satu ruangan tidak boleh dipergunakan oleh lebih dari 1 mata kuliah pada waktu atau jam yang sama. [8]
3. Satu mata kuliah bisa diampu lebih dari 1 dosen keahlian.

Untuk hari jum'at perkuliahan pada jam ke 6 (11.30-12.20) sampai jam ke 7 (12.20-13.10) ditiadakan karena adanya waktu shalat jum'at. Perkuliahan dimulai lagi pada jam ke 8 (13.10). [8]

2.2 Kecerdasan Buatan (*Artificial Intelligence*)

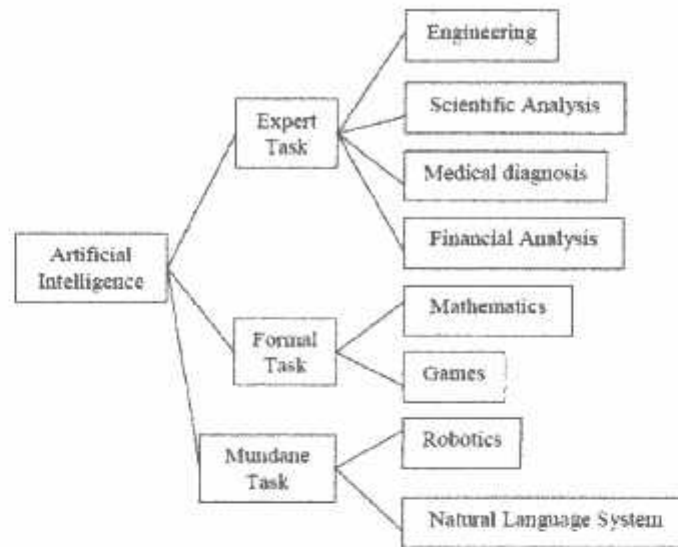
Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang berhubungan dengan pengautomatisasian tingkah laku cerdas. Pernyataan tersebut juga dapat dijadikan definisi dari AI. Definisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada *sound theoretical* (teori suara) dan prinsip-prinsip aplikasi dari bidangnya. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk mengaplikasikan pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya. [6]

Dari beberapa perspektif, AI dapat dipandang sebagai:[6]

1. Dari perspektif kecerdasan, AI adalah bagaimana membuat mesin yang cerdas dan dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan manusia
2. Dari perspektif bisnis, AI adalah sekelompok alat bantu (*tools*) yang berdayaguna dan metodologi yang menggunakan alat-alat bantu tersebut untuk menyelesaikan masalah- masalah bisnis.
3. Dari perspektif pemrograman, AI meliputi studi tentang pemrograman simbolik, pemecahan masalah, dan proses pencarian (*search*).
4. Dari perspektif penelitian :
 - a. Riset tentang AI dimulai pada awal tahun 1960-an, percobaan pertama adalah membuat program permainan catur, membuktikan teori, dan *general problem solving*.
 - b. *Artificial intelligence* adalah nama pada akar dari studi area.

Ada dua hal yang sangat mendasar mengenai penelitian-penelitian AI, yaitu *knowledge representation* (representasi pengetahuan) dan *search* (pelacakan). Para peneliti AI terus mengembangkan berbagai jenis teknik baru dalam menangani sejumlah permasalahan yang tergolong ke dalam AI seperti *vision* dan percakapan, pemrosesan bahasa alami, dan permasalahan khusus seperti diagnosa medis. [6]

AI seperti bidang ilmu lainnya juga memiliki sejumlah sub-disiplin ilmu yang sering digunakan untuk pendekatan yang esensial bagi penyelesaian suatu masalah dan dengan aplikasi bidang AI yang berbeda. Gambar II.1 merupakan sejumlah bidang-bidang tugas (*task domains*) dari AI. [6]



Gambar 2.2 Bidang-bidang tugas (task domains) dari AI [6]

Aplikasi penggunaan AI dapat dibagi ke dalam tiga kelompok, yaitu : [6]

a. Mundane task

Secara harfiah, arti *mundane* adalah keduniaan. Di sini, AI digunakan untuk melakukan hal-hal yang sifatnya duniawi atau melakukan kegiatan yang dapat membantu manusia. Contohnya : [6]

1. Persepsi (*vision & speech*).
2. Bahasa alami (*understanding, generation & translation*).
3. Pemikiran yang bersifat *commonsense*.
4. *Robot control*.

b. Format Task

AI digunakan untuk melakukan tugas-tugas formal yang selama ini manusia biasa lakukan dengan lebih baik. Contohnya : [6]

1. Permainan games
2. Matematika (geometri, logika, kalkulus, integral, pembuktian).

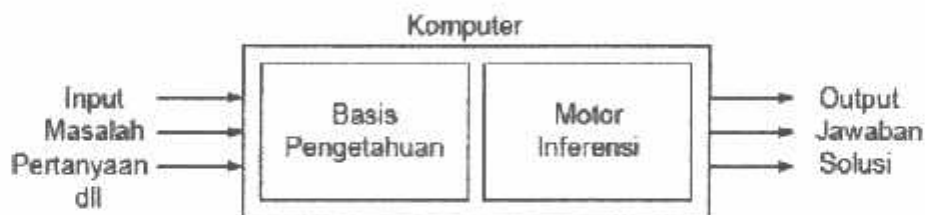
c. *Expert task*

AI dibentuk berdasarkan pengalaman dan pengetahuan yang dimiliki oleh para ahli. Penggunaan ini dapat membantu para ahli untuk menyampaikan ilmu-ilmu yang mereka miliki. Contohnya : [6]

1. Analisis financial
2. Analisis medical
3. Analisis ilmu pengetahuan
4. Rekayasa (desain, pencarian, kegagalan, perencanaan, manufaktur)

Aplikasi *Artificial Inteleget* memiliki dua bagian utama, yaitu : [6]

- a. Basis Pengetahuan (*Knowledge Base*) : berisi fakta - fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Motor Inferensi (*Inference Engine*) : kemampuan menarik kesimpulan berdasarkan pengalaman .



Gambar 2.3 Penerapan konsep kecerdasan buatan di computer [6]

2.3 Metode Algoritma Genetika

2.3.1 Pengertian Algoritma Genetika

Algoritma genetika merupakan metode pencarian yang disesuaikan dengan proses genetika dari organisme-organisme biologi yang berdasarkan pada teori evolusi Charles Darwin. (Algoritma genetic pertama kali ditemukan oleh John Holland, itu dapat dilihat dalam bukunya yang berjudul *Adaption in Natural and Artificial Systems* pada tahun 1960-an dan kemudian dikembangkan bersama murid dan rekan kerjanya di Universitas Michigan pada tahun 1960-an sampai 1970-an. Tujuan Holland mengembangkan Algoritma Genetika saat itu bukan untuk mendesain suatu algoritma yang dapat memecahkan suatu masalah, namun lebih

mengarah ke studi mengenai fenomena adaptasi di alam dan mencoba menerapkan mekanisme adaptasi alam tersebut ke dalam sistem komputer. [7]

Algoritma Genetika yang dibuat Holland merupakan sebuah metode untuk memisahkan satu populasi kromosom (terdiri dari bit-bit 1 dan 0) ke populasi baru dengan menggunakan “seleksi alam” dan operator genetik seperti *crossover*, *mutation*, *inversion*. *Crossover* menukar bagian kecil dari dua kromosom, *mutation* mengganti secara acak nilai gen di beberapa lokasi pada kromosom, *inversion* membalikkan urutan beberapa gen yang berurutan dalam kromosom. Dasar teori inilah yang menjadi dasar kebanyakan program yang menggunakan algoritma genetika pada saat ini. [7]

2.3.2 Hal-hal yang harus dilakukan dalam menggunakan algoritma genetika adalah: [7]

1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat
2. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
3. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
4. Menentukan proses seleksi yang akan digunakan.
5. Menentukan proses perkawinan silang (*cross-over*).
6. Mutasi gen yang akan digunakan.

2.3.3 Pengertian individu

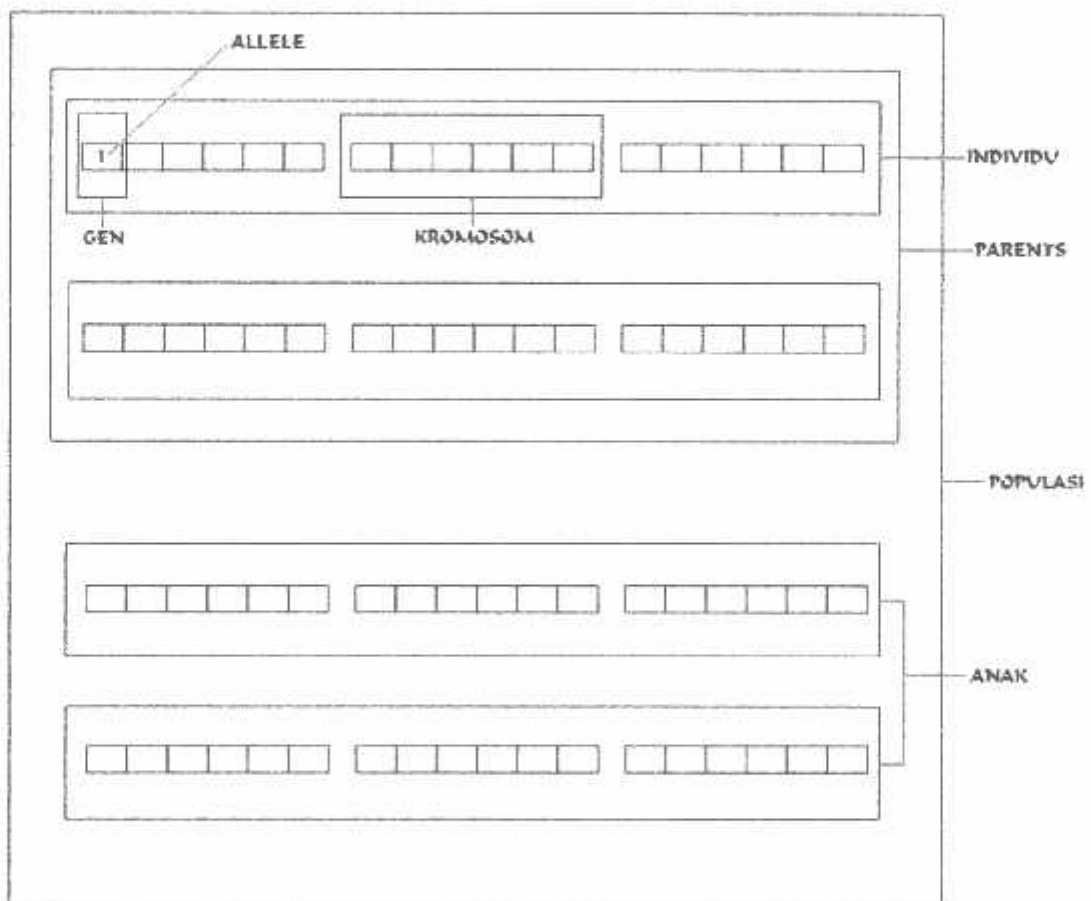
Individu menyatakan salah satu solusi yang mungkin. Individu bisa dikatakan sama dengan kromosom, yang merupakan kumpulan gen. Gen ini bisa berupa biner, float, dan kombinatorial. [7]

Beberapa definisi penting dalam mendefinisikan individu untuk membangun penyelesaian permasalahan pada algoritma genetika adalah sebagai berikut : [7]

1. 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, atau kombinatorial.

2. Allele, nilai dari gen.
3. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
4. Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. Generasi, menyatakan satu-satuan siklus proses evolusi.



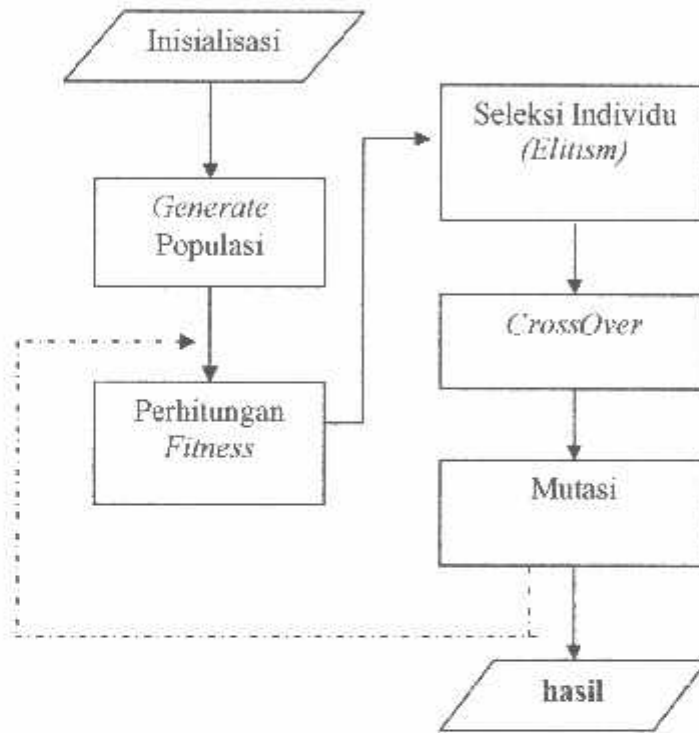
Gambar 2.4 Siklus proses evolusi [7]

7. Nilai Fitness, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

2.3.4 Nilai Fitness

Nilai fitness adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai fitness inilah yang dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika. Algoritma genetika bertujuan mencari individu dengan nilai fitness yang paling tinggi. [7]

2.3.5 Siklus Algoritma Genetika



Gambar 2.5 Siklus Algoritma Genetika [7]

1. Membangkitkan Populasi Awal

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap individunya. [7]

2. Seleksi

Seleksi dilakukan untuk mendapatkan calon induk yang baik. "Induk yang baik akan menghasilkan keturunan yang baik". Semakin tinggi nilai fitness suatu individu semakin besar kemungkinannya untuk terpilih. Seleksi dapat dilakukan dengan menggunakan dua macam teknik, yaitu mesin roulette, dan turnamen. Seleksi dilakukan dengan menggunakan prosentasi fitness setiap individu, dimana setiap individu mendapatkan luas bagian sesuai dengan prosentase nilai fitnessnya. [7]

3. Cross Over

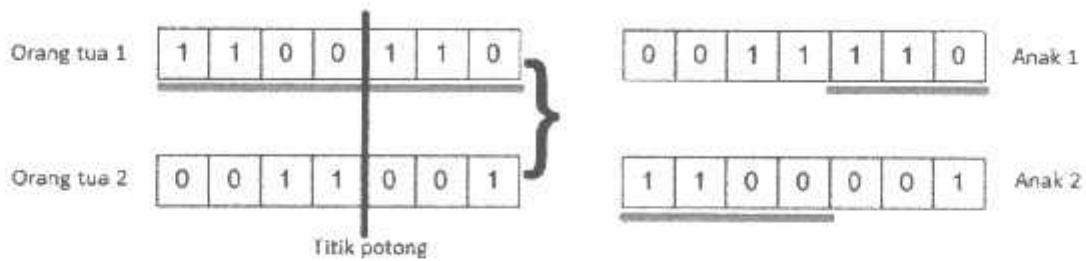
Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Cross over dilakukan dengan melakukan pertukaran gen dari dua induk secara acak. Proses cross over dilakukan pada setiap individu dengan probabilitas cross-over yang ditentukan. [7]

Crossover adalah operator Algoritma Genetika yang utama karena beroperasi pada dua kromosom pada suatu waktu dan membentuk offspring dengan mengkombinasikan dua bentuk kromosom. Cara sederhana untuk memperoleh crossover adalah dengan memilih suatu titik yang dipisahkan secara random dan kemudian membentuk offspring dengan cara mengkombinasikan segmen dari satu induk ke sebelah kiri dari titik yang dipisahkan dengan segmen dari induk yang lain ke sebelah kanan dari titik yang dipisahkan. Metode ini akan berjalan normal dengan representasi bit string. Performa dari Algoritma Genetika bergantung pada performa dari operator crossover yang digunakan. [7]

Crossover rate merupakan rasio antara jumlah offspring yang dihasilkan pada setiap generasi terhadap luas populasinya. Semakin tinggi crossover rate akan memungkinkan eksplorasi ruang solusi yang lebih luas dan mereduksi kemungkinan jatuh pada kondisi optimum yang salah. Namun memberikan rate yang memberikan konsekuensi makin lamanya waktu perhitungan yang diperlukan sebagai akibat eksplorasi pada luas

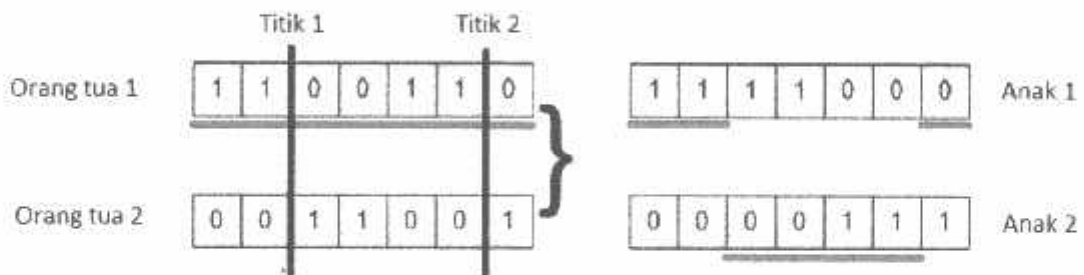
populasi yang ada. Macam-macam Cross-Over yang banyak digunakan antara lain: [7]

a. Pertukaran gen 1 titik



Gambar 2.6. Pertukaran Gen 1 titik [7]

b. Pertukaran gen 2 titik



Gambar 2.7. Pertukaran Gen 2 titik [7]

4. Mutasi Gen

Mutasi Gen merupakan operator yang menukar nilai gen dengan nilai inversinya, mialnya gennya bernilai 0 menjadi 1. Setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan. Mutasi dilakukan dengan memberikan nilai inversi atau menggeser nilai gen pada gen yang terpilih untuk dimutasikan. [7]

Mutasi dapat dikatakan sebagai operasi pendukung yang menghasilkan perubahan secara acak dan seketika pada berbagai jenis kromosom. Cara mudah untuk mendapatkan mutasi dengan mengubah satu atau lebih genes. Pada Algoritma Genetika, mutasi memainkan peran penting, yaitu pertama,

menggantikan genes yang hilang dari populasi selama proses seleksi, sehingga dapat diujikan pada suatu kondisi yang baru. Kedua, menyediakan genes yang tidak ditampilkan pada populasi awal. [7]

Mutation rate menyatakan presentase dari total jumlah genes dalam populasi. Mutation rate ini melakukan kontrol dimana genes baru dalam populasi dapat diuji seleksi. Jika rate terlalu kecil akan banyak genes yang sebenarnya bermanfaat tetapi tidak pernah diuji seleksi. Namun jika rate terlalu tinggi akan terjadi random perturbation, yang berakibat offspring mulai kehilangan kemiripan dengan induknya dan Algoritma Genetika akan kehilangan kemampuan untuk melihat urutan langkah observasinya. [7]

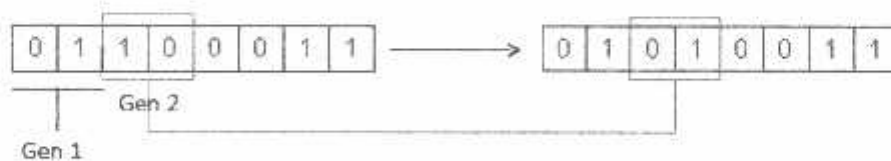
Mutasi diperlukan untuk mengembalikan informasi bit yang hilang akibat *crossover*. Mutasi diterapkan dengan probabilitas yang sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi individu yang unggul akan dirusak. Berdasarkan bagian yang dimutasi dapat dibagikan atas 3 bagian: [7]

1. Mutasi pada tingkat kromosom: semua gen dalam kromosom berubah. Contoh:



Gambar 2.8 Mutasi tingkat kromosom [7]

2. Mutasi pada tingkat gen: semua bit dalam satu gen akan berubah. Misal gen 2 yang akan mengalami mutasi maka semua bit yang ada dalam gen 2 tersebut akan berubah. Contoh:



Gambar 2.9 Mutasi gen [7]

3. Mutasi pada tingkat: Hanya satu bit yang berubah



Gambar 2.10 Mutasi tingkat bit [7]

5. Elitisme

Proses seleksi dilakukan secara random sehingga tidak ada jaminan bahwa suatu individu yang bernilai fitness tertinggi akan selalu terpilih. Walaupun individu bernilai fitness tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai fitnessnya menurun) karena proses pindah silang. Oleh karena itu, untuk menjaga agar individu bernilai fitness tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa kopinya. Prosedure ini dikenal sebagai elitisme. [7]

2.4 Pemrograman Visual Basic 6.0

Microsoft Visual Basic (sering disingkat sebagai VB) merupakan sebuah bahasa pemrograman yang menawarkan Integrated Development Environment (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows. Basis bahasa pemrograman yang digunakan dalam Visual Basic adalah bahasa BASIC (*Beginners All-Purpose Symbolic Instruction Code*) yang merupakan salah satu bahasa pemrograman tingkat tinggi yang sederhana. Dengan Visual Basic, kita bias membuat program aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pengguna komputer berkomunikasi dengan computer tersebut menggunakan grafik atau gambar. [2]

Microsoft Visual Basic 6.0 menyediakan berbagai perangkat kontrol yang dapat digunakan untuk membuat program aplikasi dalam sebuah form baik aplikasi kecil, sederhana hingga ke aplikasi pengolahan database yang kompleks. Kemudahan ini masih ditambah dengan tersedianya sarana dan peranti yang lengkap. [2]

2.4.1 Keistimewaan Microsoft Visual Basic 6.0

Microsoft Visual basic 6.0 memiliki banyak keistimewaan dalam penggunaannya yaitu: [1]

- a) Memiliki perangkat yang otomatis.
- b) Dapat mengembangkan Database dengan mudah dan kemampuan dengan menghadirkan banyak fasilitas baru untuk aplikasi Database.
- c) Perlengkapan untuk merancang aplikasi web tersedia sangat banyak

2.5 Microsoft Access

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007. [4]

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek. [4]

Microsoft merilis Microsoft Access 1.0 pada bulan November 1992 dan dilanjutkan dengan merilis versi 2.0 pada tahun 1993. Microsoft menentukan spesifikasi minimum untuk menjalankan Microsoft Access 2.0 adalah sebuah komputer dengan sistem operasi Microsoft Windows 3.0, RAM berkapasitas 4 *megabyte* (6 *megabyte* lebih disarankan) dan ruangan kosong hard disk yang dibutuhkan 8 *megabyte* (14 *megabyte* lebih disarankan). Versi 2.0 dari Microsoft Access ini datang dengan tujuh buah disket *floppy* 3½ inci berukuran 1.44 *megabyte*.

Perangkat lunak tersebut bekerja dengan sangat baik pada sebuah basis data dengan banyak *record* tapi terdapat beberapa kasus di mana *data* mengalami kerusakan. Sebagai contoh, pada ukuran basis data melebihi 700 *megabyte* sering mengalami masalah seperti ini (pada saat itu, memang *hard disk* yang beredar masih berada di bawah 700 *megabyte*). Buku manual yang dibawanya memperingatkan bahwa beberapa kasus tersebut disebabkan oleh *driver* perangkat yang kuno atau konfigurasi yang tidak benar. [4]

Nama kode (*codename*) yang digunakan oleh Access pertama kali adalah Cirrus yang dikembangkan sebelum Microsoft mengembangkan Microsoft Visual Basic, sementara mesin pembuat *form* antarmuka yang digunakannya dinamakan dengan Ruby. Bill Gates melihat purwarupa (*prototype*) tersebut dan memutuskan bahwa komponen bahasa pemrograman BASIC harus dikembangkan secara bersamaan sebagai sebuah aplikasi terpisah tapi dapat diperluas. Proyek ini dinamakan dengan Thunder. Kedua proyek tersebut dikembangkan secara terpisah, dan mesin pembuat *form* yang digunakan oleh keduanya tidak saling cocok satu sama lainnya. Hal tersebut berakhir saat Microsoft merilis Visual Basic for Applications (VBA). [4]

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para programmer untuk membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. Access juga dapat digunakan sebagai sebuah basis data untuk aplikasi Web dasar yang disimpan di dalam *server* yang menjalankan Microsoft Internet Information Services (IIS) dan menggunakan Microsoft Active Server Pages (ASP). Meskipun demikian, penggunaan Access kurang disarankan, mengingat telah ada Microsoft SQL Server yang memiliki kemampuan yang lebih tinggi. [4]

Beberapa pengembang aplikasi profesional menggunakan Microsoft Access untuk mengembangkan aplikasi secara cepat (digunakan sebagai *Rapid Application Development/RAD tool*), khususnya untuk pembuatan purwarupa untuk sebuah program yang lebih besar dan aplikasi yang berdiri sendiri untuk para *salesman*. [4]

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server. Meskipun demikian, tampilan muka Access (*form, report, query*, dan kode Visual Basic) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya Microsoft Jet Database Engine (yang secara *default* digunakan oleh Microsoft Access), Microsoft SQL Server, Oracle Database, dan beberapa produk lainnya yang mendukung ODBC. [4]

Salah satu keunggulan Microsoft Access dilihat dari perspektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statemen-statemen SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek. [4]

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini dari Microsoft SQL Server 2000, dimasukkan ke dalam Office XP Developer Edition dan dapat digunakan oleh Microsoft Access sebagai alternatif dari Microsoft Jet Database Engine. [4]

Tidak seperti sebuah sistem manajemen basis data relasional yang komplit, Microsoft JET Database Engine tidak memiliki fitur *trigger* dan *stored procedure*. Dimulai dari Microsoft Access 2000 yang menggunakan Microsoft Jet Database Engine versi 4.0, ada sebuah sintaksis yang mengizinkan pembuatan kueri dengan beberapa parameter, dengan sebuah cara seperti halnya sebuah *stored procedure*, meskipun prosesur tersebut dibatasi hanya untuk sebuah pernyataan tiap prosedurnya. Access juga mengizinkan *form* untuk mengandung kode yang dapat dieksekusi ketika terjadi sebuah perubahan terhadap tabel basis data, seperti halnya *trigger*, selama modifikasi dilakukan hanya dengan menggunakan *form* tersebut, dan merupakan sesuatu hal yang umum untuk menggunakan kueri yang akan diteruskan (*passthrough*

dan teknik lainnya di dalam Access untuk menjalankan *stored procedure* di dalam RDBMS yang mendukungnya. [4]

Dalam berkas Access Database Project (ADP) yang didukung oleh Microsoft Access 2000 dan yang selanjutnya, fitur-fitur yang berkaitan dengan basis data berbeda dari versi format/struktur data yang digunakan Access (*.MDB), karena jenis berkas ini dapat membuat koneksi ke sebuah basis data MSDE atau Microsoft SQL Server, ketimbang menggunakan Microsoft Jet Database Engine. Sehingga, dengan menggunakan ADP, adalah mungkin untuk membuat hampir semua objek di dalam server yang menjalankan mesin basis data tersebut (tabel basis data dengan *constraints* dan *trigger*, *view*, *stored procedure*, dan *UDF*). Meskipun demikian, yang disimpan di dalam berkas ADP hanyalah *form*, *report*, *macro*, dan modul, sementara untuk tabel dan objek lainnya disimpan di dalam server basis data yang membelakangi program tersebut. [4]

Access mengizinkan pengembangan yang relatif cepat karena semua table basis data, kueri, form, dan report disimpan di dalam berkas basis data miliknya (*.MDB). Untuk membuat Query, Access menggunakan Query Design Grid, sebuah program berbasis grafis yang mengizinkan para penggunanya untuk membuat query tanpa harus mengetahui bahasa pemrograman SQL. Di dalam Query Design Grid, para pengguna dapat memperlihatkan tabel basis data sumber dari *query*, dan memilih *field-field* mana yang hendak dikembalikan oleh proses dengan mengklik dan menyeretnya ke dalam *grid*. *Join* juga dapat dibuat dengan cara mengklik dan menyeret *field-field* dalam tabel ke dalam *field* dalam tabel lainnya. Access juga mengizinkan pengguna untuk melihat dan memanipulasi kode SQL jika memang diperlukan. [4]

Bahasa pemrograman yang tersedia di dalam Access adalah Microsoft Visual Basic for Applications (VBA), seperti halnya dalam beberapa aplikasi Microsoft Office. Dua buah pustaka komponen Component Object Model (COM) untuk mengakses basis data pun disediakan, yakni Data Access Object (DAO), yang hanya terdapat di dalam Access 97, dan ActiveX Data Objects (ADO) yang tersedia dalam versi-versi Access terbaru. [4]

BAB III

PERANCANGAN DAN DESAIN APLIKASI

Bab ini berisi tentang perancangan dan desain aplikasi otomatisasi sistem penjadwalan perkuliahan dengan menggunakan Algoritma Genetika sebagai metodenya. Perancangan aplikasi ini meliputi deskripsi sistem secara umum, perancangan sistem dan perancangan desain aplikasi.

3.1 Deskripsi Sistem Secara Umum

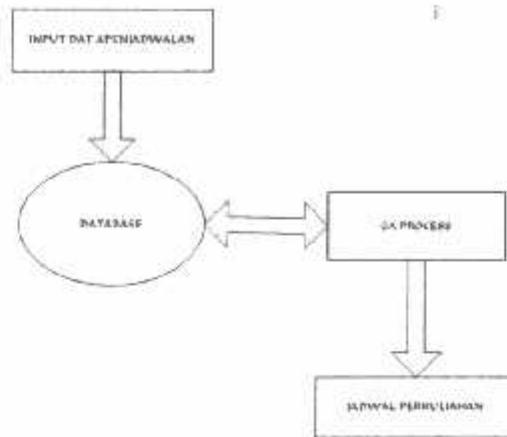
Secara umum aplikasi ini adalah sebuah sistem yang dibuat untuk menyediakan penjadwalan mata kuliah secara otomatis/terotomatisasi dengan menggunakan Algoritma Genetika sebagai metodenya, sehingga dapat mengefisiensi waktu didalam pembuatan jadwal perkuliahan nantinya.

Penjadwalan perkuliahan yang dimaksudkan adalah untuk mengantisipasi kemungkinan terjadinya kendala yang muncul berkaitan dengan penjadwalan yang bentrok. Seperti terjadinya crash jadwal ruang, mata kuliah, dosen mengajar, dll. Permasalahan akan menjadi kompleks ketika proses penjadwalan dilakukan secara manual dengan jumlah data yang besar, data tersebut diantaranya adalah data dosen, data mata kuliah, data ruangan yang digunakan dalam perkuliahan, serta data waktu yang dipakai dalam perkuliahan.

Aplikasi ini membantu untuk memberikan beberapa pertimbangan dalam proses pengubahan jadwal mata kuliah. Misal ada dosen yang menginginkan pergantian jadwal kuliahnya. Sistem ini akan memberikan beberapa pertimbangan seperti tersedianya jam mata kuliah yang ada atau ruang kelas yang tersedia.

3.2 Perancangan Sistem

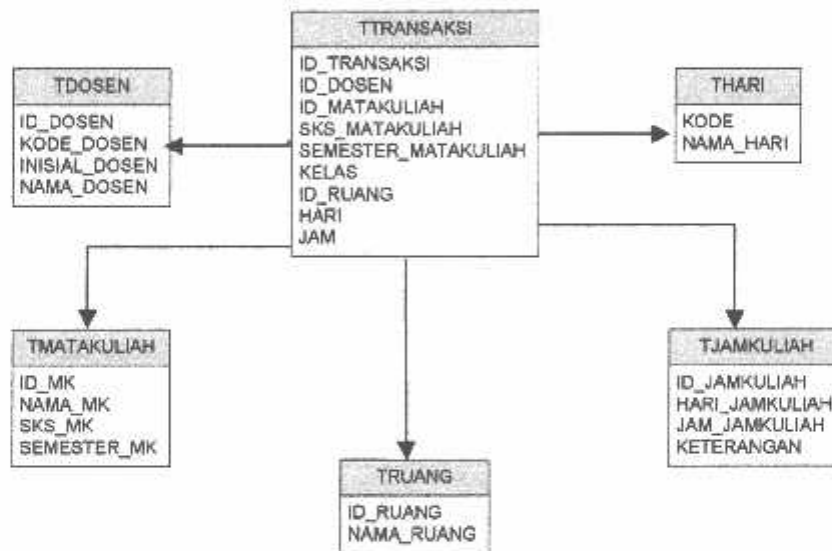
Tahapan perancangan sistem aplikasi ini adalah dimulai dengan mengolah data yang didapat dari data penjadwalan smester sebelumnya, kemudian merancangnya kedalam sebuah basis data lalu disesuaikan dengan keterkaitan antar entitas - *attribute*. Kemudian langkah selanjutnya adalah dengan menyesuaikan basis data kedalam operator GA, diproses dengan parameter *crossover* dan *mutasi* sehingga dapat menghasilkan jadwal perkuliahan. Seperti yang tertera pada gambar 3.1.



Gambar 3.1 Diagram Blok sistem

3.2.1 Database

3.2.1.1 Relasi Tabel



Gambar 3.2 Relasi sistem penjadwalan

Gambar 3.2 tersebut merupakan skema Relasi tabel dari sistem penjadwalan perkuliahan. Terdapat enam tabel dimana 5 tabel berisi tabel murni dan satu tabel yang menghubungkan antara TDOSEN, TMATAKULIAH, TRUANG, TRUANG, TJAMKULIAH, dan THARI.

3.2.1.2 Struktur Tabel

Berikut berupa rincian tabel-tabel yang terdapat pada relasi tabel yang ada di gambar 3.2 yang nantinya akan diproses kedalam algoritma genetika:

1. Tabel Dosen

Pada tabel dosen terdapat daftar nama dosen, inisial dosen, kode dosen dan juga id dosen. Adapun desainnya tertera pada tabel 3.1 sebagai berikut :

Tabel 3.1 Tabel Dosen

Kolom	Type data
ID_DOSEN	Number (Primary key)
KODE DOSEN	Text
INISIAL_DOSEN	Text
NAMA_DOSEN	Text

2. Tabel mata kuliah

Pada tabel mata kuliah berisikan nama mata kuliah, sks mata kuliah, semester mata kuliah, dan juga id mata kuliah. Dimana untuk desain tabelnya tertera pada tabel 3.2 sebagai berikut :

Tabel 3.2 Tabel mata kuliah

Kolom	Type data
ID_MK	Number (Primary key)
NAMA_MK	Text
SKS_MK	Number
SEMESTER_MK	Text

3. Tabel ruang

Pada tabel ruang terdapat beberapa nama ruang, dan juga id ruang. Untuk desain tabelnya seperti yang tertera pada tabel 3.3 sebagai berikut :

Tabel 3.3 Tabel ruang

Kolom	Type data
ID_RUANG	Number (Primary key)
NAMA_RUANG	Text

4. Tabel jam kuliah

Pada tabel jam kuliah terdapat id jam kuliah, hari, jam, dan keterangan hari. Untuk desain tabelnya seperti yang tertera pada tabel 3.4 sebagai berikut

Tabel 3.4 Tabel Jam kuliah

Kolom	Type data
ID_JAMKULIAH	Number (Primary key)
HARI_JAMKULIAH	Number
JAM_JAMKULIAH	Number
KETERANGAN	Text

5. Tabel hari

Pada tabel hari terdapat nama hari, dan juga kode hari. Untuk desain tabelnya seperti yang tertera pada tabel 3.5 sebagai berikut :

Tabel 3.5 Tabel hari

Kolom	Type data
KODE	Number (Primary key)
NAMA_HARI	Text

6. Tabel transaksi

Tabel transaksi merupakan kumpulan dari masing – masing tabel yang nantinya akan didekodensikan ke dalam kromosom GA. Untuk desain tabelnya seperti yang tertera pada tabel 3.6 sebagai berikut :

Tabel 3.6 Tabel transaksi

Kolom	Type data
ID_TRANSAKSI	Number (Primary key)
ID_DOSEN	Number
ID_MATAKULIAH	Number
SKS_MATAKULIAH	Number
SEMESTER_MATAKULIAH	Text
KELAS	Text
ID_RUANG	Number
HARI	Number
JAM	Number

3.2.2 Dekomposisi Basis Data

3.2.2.1 Perancangan Basis Data

Basis data yang dibuat berfungsi untuk menampung data sebuah tabel. Attribute yang ada berupa ID_DOSEN, ID_MATAKULIAH, SKS_MATAKULIAH, SEMESTER_MATAKULIAH, KELAS, ID_RUANG, HARI, JAM. Attribute ini memiliki index *unique* sebagai ID_TRANSAKSI terdapat dalam sebuah tabel yang diberi nama TTRANSAKSI. Data yang ada sebanyak 104 index *unique* untuk penjadwalan semester ganjil, dan terdapat 92 index *unique* untuk penjadwalan semester genap.

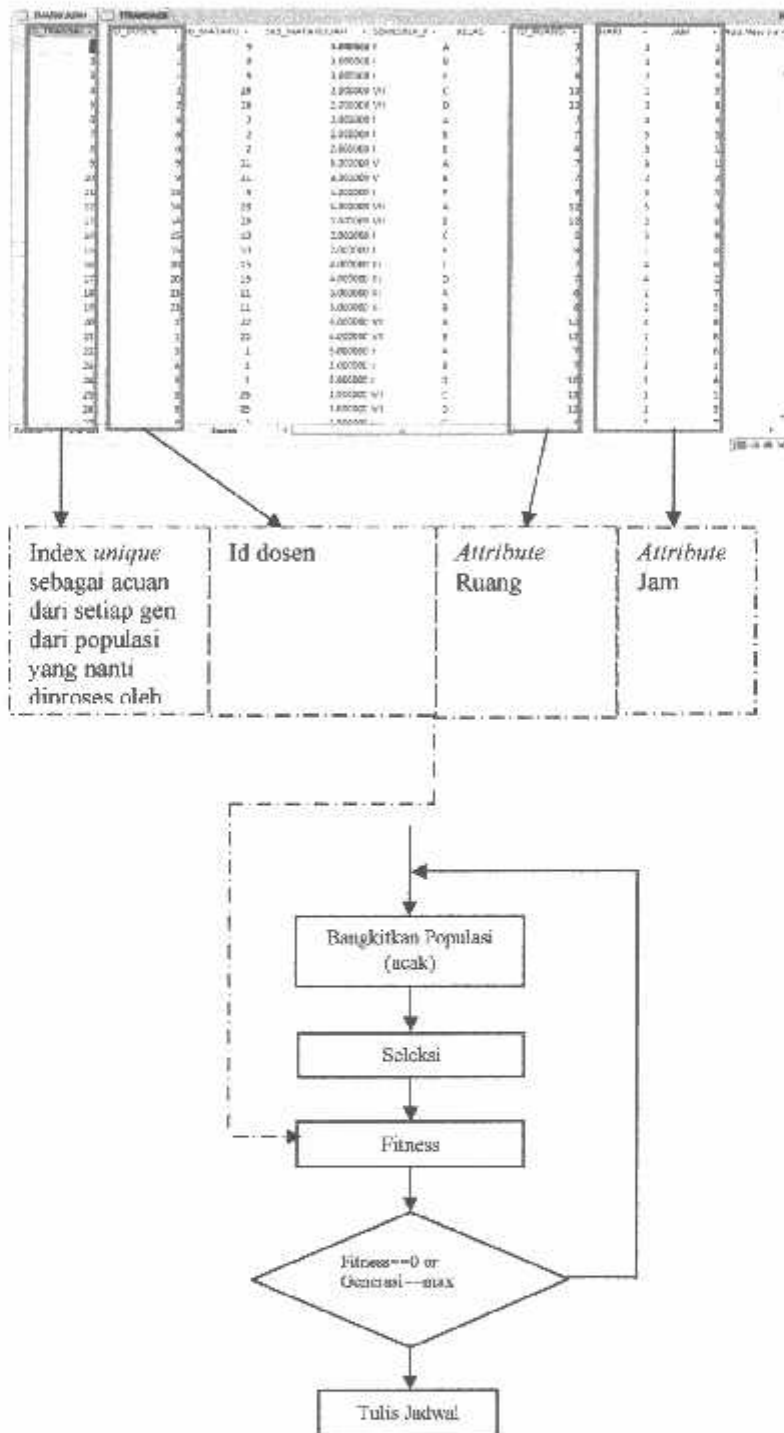
Dalam dekomposisinya tabel yang akan dikodensi sebagai kromosom GA adalah seperti yang tertera pada gambar 3.3.

ID_TRANSAKSI	ID_DOSEN	ID_MATAKULIAH	SKS_MATAKULIAH	SEMESTER_MATAKULIAH	KELAS	ID_RUANG	HARI	JAM
1	1	9	3.000000	I	A	7	9	1
2	1	9	3.000000	I	B	7	9	8
3	1	9	3.000000	I	F	9	2	5
4	2	26	2.000000	VII	C	12	1	9
5	2	26	2.000000	VII	D	12	2	8
6	6	2	2.000000	I	A	7	4	3
7	6	2	2.000000	I	B	7	5	3
8	6	2	2.000000	I	E	4	5	1
9	9	21	1.000000	V	A	7	9	1
10	9	21	3.000000	V	B	7	2	2
11	13	3	4.000000	I	F	9	4	5
12	14	28	1.300000	VII	A	12	5	9
13	14	28	3.000000	VII	B	12	2	8
14	15	10	2.000000	I	C	5	3	9
15	15	10	3.000000	I	F	9	1	4
16	20	15	4.000000	III	C	7	4	6
17	20	15	4.000000	III	D	7	4	2
18	23	11	3.000000	III	A	6	1	7
19	23	11	3.000000	III	B	6	1	5
20	1	22	4.000000	VIII	A	12	4	8
21	1	22	4.000000	VIII	B	12	1	8
22	3	1	5.000000	I	A	7	3	8
23	3	1	5.000000	I	B	7	3	1
24	3	1	5.000000	I	C	10	1	4
25	5	25	3.000000	VIII	C	12	1	1
26	5	25	3.000000	VIII	D	12	1	5

Gambar 3.3 Tabel Transaksi

Dimana dari seluruh tabel yang ada di dekomposisikan ke dalam sebuah tabel yang diberi nama tabel transaksi. Dan nantinya akan dikodensi sebagai kromosom GA. Pengelompokan keterikatan antar attribute ditentukan oleh jam perkuliahan, ruang, dan dosen. Terdapat 12 jam pertemuan untuk hari senin-sabtu, terdapat 12 ruangan yang tersedia untuk melakukan kegiatan belajar mengajar, dan juga terdapat 23 dosen yang tersedia untuk mengajar. Nantinya dari ketiga attribute inilah yang dijadikan acuan formulasi nilai fitness.

Berikut Proses dekomposisi basis data kedalam GA diilustrasikan seperti pada gambar 3.4 dimana diambil kolom index *unique*, id dosen, atribut ruang, dan jam yang akan diproses kedalam pencarian fitness.



Gambar 3.4 Proses dekomposisi basis data.

3.2.2.2 Fitness

Setelah individu – individu dalam populasi terbentuk, maka dilakukan perhitungan nilai fitness terhadap setiap individu dengan aturan sebagai berikut:

$$TF = TFD - TFR$$

Dimana TF mewakili nilai Total Fitness, TFD mewakili nilai Total Fitness Dosen, dan TFR mewakili Total Fitness Ruang. Untuk penjelasan penghitungan total fitness dosen/ fitness ruang dapat diilustrasikan sebagai berikut:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN
1	Fitness Jam & Ruang													Fitness Jam & Dosen																										
2	Hari, Jam ke-													Hari, Jam ke-																										
3	Senin													Senin																										
4	1	1	2	3	4	5	6	7	8	9	10	11	12		1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
5	2	1	2	3	4	5	6	7	8	9	10	11	12		2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
6	3	1	2	3	4	5	6	7	8	9	10	11	12		3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
7	4	1	2	3	4	5	6	7	8	9	10	11	12		4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
8	5	1	2	3	4	5	6	7	8	9	10	11	12		5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
9	6	1	2	3	4	5	6	7	8	9	10	11	12		6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
10	7	1	2	3	4	5	6	7	8	9	10	11	12		7	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
11	8	1	2	3	4	5	6	7	8	9	10	11	12		8	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
12	9	1	2	3	4	5	6	7	8	9	10	11	12		9	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
13	10	1	2	3	4	5	6	7	8	9	10	11	12		10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
14	11	1	2	3	4	5	6	7	8	9	10	11	12		11	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
15	12	1	2	3	4	5	6	7	8	9	10	11	12		12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
16	Selasa													Selasa																										
17	1	1	2	3	4	5	6	7	8	9	10	11	12		1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
18	2	1	2	3	4	5	6	7	8	9	10	11	12		2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
19	3	1	2	3	4	5	6	7	8	9	10	11	12		3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
20	4	1	2	3	4	5	6	7	8	9	10	11	12		4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
21	5	1	2	3	4	5	6	7	8	9	10	11	12		5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
22	6	1	2	3	4	5	6	7	8	9	10	11	12		6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
23	7	1	2	3	4	5	6	7	8	9	10	11	12		7	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
24	8	1	2	3	4	5	6	7	8	9	10	11	12		8	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
25	9	1	2	3	4	5	6	7	8	9	10	11	12		9	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
26	10	1	2	3	4	5	6	7	8	9	10	11	12		10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
27	11	1	2	3	4	5	6	7	8	9	10	11	12		11	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
28	12	1	2	3	4	5	6	7	8	9	10	11	12		12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
29	Rabu													Rabu																										

Gambar 3.5 ilustrasi perhitungan total fitness ruang dan dosen

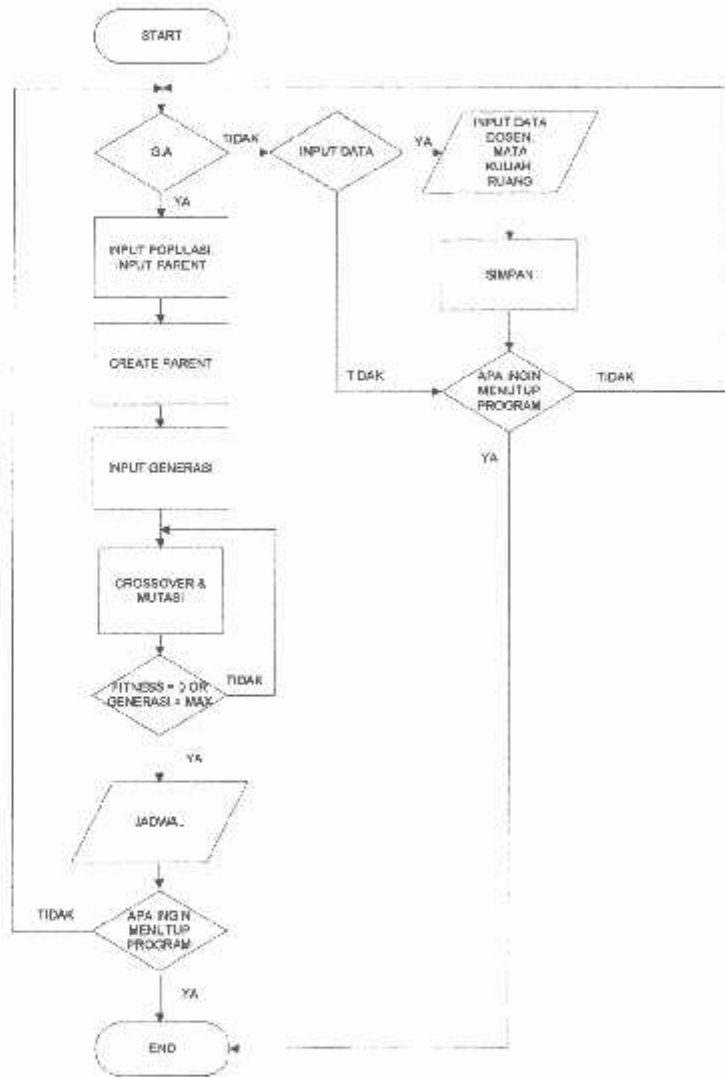
Pada gambar 3.5 terdapat 2 tabel Fitness Jam & Ruang, Fitness Jam & Dosen. Tabel tersebut berisi dua kolom yaitu kolom Hari, Jam ke- dan Ruang. Kolom Hari, Jam ke- terdiri dari hari aktif dalam perkuliahan mulai dari hari senin sampai hari sabtu beserta jam perkuliahan yang berlaku dimulai dari jam ke-1 sampai dengan jam ke-12. Untuk kolom Ruang terdiri dari angka 1 – sampai dengan 12. 12 merupakan jumlah ruangan yang digunakan dalam proses belajar mengajar di Teknik Informatika ITN Malang. Begitu juga untuk Tabel Fitness Jam dan Dosen terdapat angka 1- sampai dengan 23 pada kolom ruang dimana 23 merupakan jumlah dosen yang mengajar.

Proses perhitungan tabel Fitness Jam & Ruang. Pada baris ke – 4 kolom pertama terdapat kolom hari, jam ke yakni hari senin jam ke – 1 dimana pada kolom berikutnya terdapat kolom ruang yang berisi angka 1 – 12, nantinya pada kolom ruang tersebut akan dilakukan proses *randomize* dimana dari ke – 12 itu

akan diisi atau ditandai. Setelah ditandai kemudian dihitung dengan menggunakan rumus – 1, dimana untuk hasilnya terdapat pada kolom N. Proses *randomize* terus dilakukan sampai ke baris akhir yakni pada hari sabtu jam ke – 12. Lalu hasil dari keseluruhan perhitungan pada kolom N tersebut ditotal sehingga terbentuklah total nilai fitness ruang. Untuk proses perhitungan tabel nilai fitness dosen sama dengan proses perhitungan tabel fitness jam dan ruang.

3.2.3 Flowchart Sistem

Flowchart merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *Flowchart* menjelaskan tentang urutan-urutan dari prosedur yang ada di dalam sistem dengan menggunakan simbol-simbol.



Gambar 3.6 Flowchart sistem

Gambaran flowchart sistem yang terdapat pada gambar 3.6 adalah sebagai berikut:

1. Start
Langkah awal sistem akan dimulai.
2. GA
Masuk ke menu utama GA process.
3. Input Populasi dan Parent
Menginputkan batasan berapa kali Populasi dan parent akan dibangkitkan.
4. Create parent
Manjalankan proses setelah batasan populasi yang akan dibangkitkan telah diinputkan.
5. Input generasi
Setelah proses pembangkitan populasi dan parent selesai, dilanjutkan dengan menginputkan batasan untuk generasi yang akan dibangkitkan.
6. Crossover dan mutasi
Memproses dengan parameter crossover dan mutasi berdasarkan batasan generasi yang telah dibangkitkan.
7. Jadwal
Hasil
8. Data
Berisi tentang data yang berupa data dosen, mata kuliah, dan ruang.

3.3 Perancangan Desain Aplikasi

3.3.1 Desain Tampilan Form Awal

Form awal berisikan dua menu bar yaitu menu GA Process dan menu Data. Menu GA Process berfungsi untuk masuk ke menu utama penjadwalan perkuliahan, menu Data berfungsi untuk mengupdate data dosen, data mata kuliah, data ruang. Berikut contoh desain tampilan form awal terdapat pada gambar 3.7.

GA	Data

Gambar 3.7 Tampilan form awal.

3.3.2 Tampilan Desain Form Utama

Bangkitkan Populasi ke	1	B1	Jumlah Generasi	8	B3
Jumlah Induk Parent	2		3	9	
			B2		
4	5	6	7		

Gambar 3.8 Tampilan desain form utama.

Keterangan :

1. B1 = Create parent. Setelah kolom bangkitkan populasi dan jumlah induk parent diinputkan Button ini berfungsi untuk memprosesnya.
2. B2 = Crossover&Mutasi. Setelah kolom 3 (jumlah generasi) di inputkan batasannya Button ini berfungsi untuk memprosesnya dengan parameter mutasi dan crossover.

3. B3 = Setelah semua proses selesai mulai dari create parent (B1) dan crossover&mutasi (B2) selesai button ini berfungsi untuk melihat output dari sistem penjadwalan ini.
4. Kolom 1 = Bangkitkan populasi. Menginputkan batasan nilai untuk populasi yang akan dibangkitkan.
5. Kolom 2 = Jumlah Induk/Parent. Menginputkan batasan untuk jumlah induk / parent yang akan dibangkitkan.
6. Kolom 3 = Jumlah generasi. Setelah proses di kolom 4 dan 5 selesai di run menggunakan B1(Button 1) maka dilanjutkan dengan menginputkan jumlah generasi dengan batasan yang telah ditentukan pada kolom 3 ini.
7. Kolom 4 = kolom yang berisi nilai fitness dosen, fitness ruang, fitness total setelah dibangkitkan dengan batasan yang telah ditentukan.
8. Kolom 5 = sama dengan kolom 4
9. Kolom 6 = sama dengan kolom 4
10. Kolom 7 = Hasil jadwal.
11. Kolom 8 = Total Generasi yang dibangkitkan.
12. Kolom 9 = Lama waktu running.

3.4 Kebutuhan Sistem

Kebutuhan sistem pada aplikasi ini meliputi *hardware* dan *software* yang akan digunakan untuk membangun aplikasi sistem penjadwalan. Spesifikasi beserta *hardware* dan *software* yang digunakan diantaranya terdapat pada tabel 3.7 dibawah ini.

Tabel 3.7 kebutuhan sistem

Hardware	Processor	Intel® Core™2 Duo processor P8600 2,40 Ghz
	Kartu VGA	NVIDIA GeForce 9600m GS 512 MB
	Memory/RAM	2GB DDR 2
Software	OS	Windows 7 Ultimate 32 bit
	Bahasa Pemrograman	Visual Basic 6
	Developer tools	Microsoft office Access 2007 Notepad

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Deskripsi aplikasi

Otomatisasi Sistem Penjadwalan Menggunakan Algoritma Genetika merupakan aplikasi yang difungsikan untuk membantu pembuatan jadwal perkuliahan, dimana Input data yang digunakan adalah data dari jadwal perkuliahan di semester sebelumnya. Tampilan halaman utama aplikasi ini bisa dilihat pada gambar 4.1 dan 4.2 berikut.



Gambar 4.1 Tampilan halaman utama



Gambar 4.2 Tampilan menu input data

4.1.1 Menu input data

Berikut merupakan tampilan dari form input data yang ada pada aplikasi ini.

a. Input data dosen

ID DOSEN	KODE DOSEN	INISIAL DOSEN	NAMA DOSEN
8	1054	M.Ae	M. ASHAR BR. MT
9	1070	S.N.M	SANDY NATALI HANTJA, S.KOM
10	1083	Ay	DR. ARYUANTO, ST., MT
11	1086	F.S/W	FEBRIANA SANTI W., S.KOM., M.KOM
12	1087	K.A.S	KARINA ALIHA SARI, ST
13	1102	A.M	ALI MAHMUDI, B.Eng., PhD
14	1103	S.P	SOMMY PRASETIO, ST., MT
15	1104	S.A.W	SURYO ACH W/BOWU, ST
16	1105	Y.A.P	YUSEP AGUS PRANOTO, ST
17	1107	M.Ar	MICHAEL ARDITA, ST., MT
18	1110	Ke	KARONO, SS, MA
19	1111	A.P	AMIN PRASADO, SAG
20	1115	N.V	NURLAILY VENDYANSYAH, ST
21	1131	E.S	EKO SUHARVANTO, ST
22	1136	D.S	DOMINGGUS SIMBOLINGL.S TH
23	1145	A.T.E	DRS. AGUS TRI EDY, M.PD

Gambar 4.3 Data dosen

b. Input data mata kuliah



ID_MK	NAMA_MK	SKS_MK	SEMESTER_MK
13	PROGRAMAN BERBASIS WEB 4		III
14	PROGRAMAN WEB 4		III
15	SISTEM OPERASI + PR 4		III
16	KECEPDASAN BUAHA 3		V
17	KOMPUTER VISION 3		V
18	MANAJEMEN JARINGAN 3		V
19	PEMROSESAN DAN SIM 3		V
20	REKAYASA PERANGKAT 4		V
21	SISTEM PENDUKUNG 3		V
22	BAHASA INGGRIS PR 4		VII
23	EMBEDDED SYSTEM 3		VII
24	KOMPUTAS BERGAYA 3		VII
25	PENDIRIAN KEWAJIBAN 3		VII
26	PENANGANAN TEKNIS 2		VII
27	PERENCANAAN SUKSES 3		VII
28	ROBOTIKA 3		VII

Gambar 4.4 Data mata kuliah

c. Input data ruang



ID_RUANG	NAMA_RUANG
1	14
2	15
3	17
4	8.2.2
5	8.2.3
6	8.2.4
7	8.2.4
8	8.3.1
9	8.3.2
10	8.3.3
11	8.3.4
12	8.3.A

Gambar 4.5 Data ruang

4.1.2 Menu GA Process

Menu GA Process merupakan menu utama dari aplikasi ini. Berikut merupakan tampilan dari menu GA Process terdapat pada gambar 4.6.



Populasi Awal	10	Crossover	Jumlah Generasi	50	Total Generasi	0	JADWAL JADI
Jumlah Parent	100		CROSSOVER DAN MUTASI		0		

Gambar 4.6 Menu GA Process

4.1.3 Konfigurasi

Pada gambar 4.6 menu GA Process terdapat inputan data berupa nilai pada kolom populasi awal, jumlah parent, jumlah generasi. Nilai tersebut berupa nilai batas dari jumlah populasi yang akan dibangkitkan pada kolom populasi, jumlah parent yang akan dibangkitkan pada kolom jumlah parent, serta jumlah generasi yang akan dibangkitkan pada kolom jumlah generasi. Seperti yang tertera digambar menu GA Process inputan nilai sudah terdefault dengan nilai sebagai berikut:

Tabel 4.1 Konfigurasi nilai default pada menu GA Process

Jumlah Generasi	50
Jumlah Populasi	10
Jumlah Parent	100

Dari tabel tersebut bisa disimpulkan dari jumlah generasi yang dihasilkan adalah 50. Setiap generasi menghasilkan 10 populasi, dan setiap populasi menghasilkan 100 parents. Sebenarnya untuk penentuan nilai inputan jumlah populasi, parents, generasi bisa diisi terserah oleh user. Namun setelah dilakukan percobaan beberapa kali untuk inputan nilai tersebut seperti pada tabel 4.1 bisa dibilang lebih efisien dari segi waktu dan hasil, karena jika jumlah generasi yang dimasukkan lebih besar maka butuh waktu *running* yang lama.

4.2 Penerapan aplikasi

Aplikasi ini digunakan untuk menyelesaikan masalah penjadwalan dengan menggunakan paramer algoritma genetika. Besar masalah yang harus diselesaikan adalah sebanyak 104 kelas mata kuliah, dimana konfigurasi nilai pembangkitan sudah di *default* seperti pada tabel 4.2. Maka langkah selanjutnya adalah proses pembuatan jadwal dengan tahapan – tahapan button proses seperti create parent, crossover&mutasi, dan jadwal hasil. Berikut gambar 4.7 merupakan jadwal hasil setelah proses create parent, crossover&mutasi, jadwal jadi selesai dijalankan.

Dan setelah melakukan pengujian sebanyak 7 kali didapatkan hasil pengujian seperti yang tertera pada tabel 4.2 berikut :

Tabel 4.2 Pengujian jadwal semester ganjil

Pengujian ke -	Total Generasi	Bentrok atau crash		Durasi	Total Fitness
		Dosen	Ruang		
1	50	0	0	1m 53s	3
2	50	0	0	1m 53s	2
3	50	1	1	1m 56s	6
4	50	0	0	1m 51s	4
5	50	0	0	1m 52s	1
6	50	0	0	1m 52s	3
7	50	1	1	2m 1s	6

Berdasarkan data pengujian pada tabel 4.2 dapat disimpulkan dari 7 kali pengujian pada jadwal semester ganjil terjadi 2 kali kegagalan dengan total fitness maksimal yang dihasilkan 7. Total fitness terbaik yang dihasilkan adalah 1, terdapat pada pengujian ke-5. Dengan catatan total generasi yang dibangkitkan sebesar 50 total generasi dimana sebenarnya setelah nilai total fitness keluar masih bisa diproses lagi dengan proses crossover dan mutasi lagi sehingga bisa menghasilkan nilai ideal total fitness = 0. Dengan begitu dengan menggunakan metode GA penulis berhasil meminimalisir kemungkinan terjadinya bentrokan jadwal dengan tingkat keberhasilan >96% dengan variasi pembangkitan populasi sebanyak 10 populasi, pembangkitan jumlah induk/parent sebanyak 100 parent, pembangkitan generasi sebanyak 50 generasi, dengan durasi waktu proses <5 menit.

4.3.2 Pengujian Kedua

Pengujian kedua menggunakan data konfigurasi sebagai berikut :

- Data yang digunakan adalah daftar jadwal mengajar dosen Tahun akademik 2012/2013
- Semester Genap

- Jumlah populasi : 10
- Jumlah parent : 100
- Jumlah generasi : 50

Dan setelah melakukan pengujian sebanyak 7 kali didapatkan hasil pengujian seperti yang tertera pada tabel 4.3 berikut :

Tabel 4.3 Pengujian jadwal semester genap

Pengujian ke -	Total Generasi	Bentrok atau crash		Durasi	Total Fitness
		Dosen	Ruang		
1	50	0	0	1m 55s	2
2	50	0	0	1m 56s	1
3	50	0	0	1m 56s	1
4	50	0	0	1m 51s	3
5	50	0	0	1m 56s	2
6	50	1	1	2m 8s	7
7	50	0	0	1m 55s	3

Berdasarkan data pengujian pada tabel 4.3 dapat disimpulkan dari 7 kali pengujian pada tiap masing-masing tabel hanya terjadi 1 kali pada jadwal semester genap dengan total nilai fitness maksimal 7. Total fitness terbaik yang dihasilkan adalah 1, terdapat pada pengujian ke-2 dan ke-3. Dengan catatan total generasi yang dibangkitkan sebesar 50 total generasi dimana sebenarnya setelah nilai total fitness keluar masih bisa diproses lagi dengan proses crossover dan mutasi lagi sehingga bisa menghasilkan nilai ideal total fitness = 0. Dengan begitu dengan menggunakan metode GA penulis berhasil meminimalisir kemungkinan terjadinya bentrokan jadwal dengan tingkat keberhasilan >96% dengan variasi pembangkitan populasi sebanyak 10 populasi, pembangkitan jumlah induk/parent sebanyak 100 parent, pembangkitan generasi sebanyak 50 generasi, dengan durasi waktu proses <5 menit.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang diperoleh dari pengujian yang dilakukan :

1. Berdasarkan data pengujian penjadwalan semester ganjil dapat disimpulkan dari 7 kali pengujian pada jadwal semester ganjil terjadi 2 kali kegagalan dengan total fitness maksimal yang dihasilkan 7. Total fitness terbaik yang dihasilkan adalah 1, terdapat pada pengujian ke-5. Dengan catatan total generasi yang dibangkitkan sebesar 50 total generasi, jumlah populasi awal = 10, dan jumlah parent = 100.
2. Berdasarkan data pengujian penjadwalan semester genap dapat disimpulkan dari 7 kali pengujian pada semester genap hanya terjadi 1 kali kegagalan pada jadwal semester genap dengan total nilai fitness maksimal 7. Total fitness terbaik yang dihasilkan adalah 1, terdapat pada pengujian ke-2 dan ke-3. Dengan catatan total generasi yang dibangkitkan sebesar 50 total generasi, jumlah populasi awal = 10, dan jumlah parent = 100.
3. Berdasarkan hasil dari seluruh data pengujian yang dilakukan dapat disimpulkan dari 7 kali pengujian pada tiap masing-masing tabel terjadi 3 kali kegagalan diantaranya 2 kali pada pengujian semester ganjil dan 1 kali pada pengujian semester genap. Total fitness terbaik pada masing-masing pengujian adalah 1, dan total fitness maksimal adalah 7. Dengan catatan pengujian hanya dilakukan dengan 1 kali proses crossover dan mutasi sehingga total fitness yang dihasilkan masih tinggi. Setelah nilai total fitness keluar masih bisa diproses lagi dengan proses crossover dan mutasi sehingga bisa menghasilkan nilai ideal total fitness = 0. Dengan begitu dengan menggunakan metode GA penulis berhasil meminimalisir kemungkinan terjadinya bentrokan jadwal dengan tingkat keberhasilan >96% dengan variasi pembangkitan populasi sebanyak 10 populasi, pembangkitan jumlah induk/parent sebanyak 100 parent, pembangkitan generasi sebanyak 50 generasi, dengan durasi waktu proses <5 menit.

5.2 Saran

Dari melihat hasil yang diperoleh dari pengujian maka disarankan :







1. Konstrain bisa ditambah dan disesuaikan dengan kebutuhan dan persyaratan pembuatan jadwal yang berlaku.
2. Sistem penjadwalan ini masih sangat jauh dari sempurna dan resiko terjadinya jadwal bentrok juga masih bisa terjadi, untuk kedepannya penulis berharap aplikasi ini bisa lebih disempurnakan lagi seperti untuk output hasilnya bisa langsung di cetak tanpa harus mengcopy ulang, serta untuk ruang lingkup permasalahannya tidak hanya di jurusan tehnik informatika saja namun bisa untuk semua jurusan yang ada di ITN MALANG.

DAFTAR PUSTAKA

- 1) ANDI (2005). *Mahir Dalam 7 Hari Pemrograman Visual Basic 6.0 Untuk Pemula*. Madiun: Madcom
- 2) ANDI (2008). *Microsoft Visual Basic 6.0 Untuk Pemula*. Madiun: Madcoms
- 3) Ariani,Dian. 2011. *Optimasi Penjadwalan Mata Kuliah di Jurusan Teknik Informatika PENS Dengan Menggunakan Algoritma Particle Swarm Optimization(PSO)*. Skripsi. Surabaya: Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh November Surabaya.
- 4) Kadir,Abdul 2010. *Mudah Mempelajari Database Access*. Yogyakarta: Andi.
- 5) Maria, Lina, 2011. *Optimasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi kasus: Jurusan Tehnik Informatika UIN Maulana Malik Ibrahim)*. Skripsi. Malang: Jurusan Tehnik Informatika UIN Maulana Malik Ibrahim
- 6) Suyanto,ST.MSc. 2011 (edisi revisi). *Artificial Intelligence Searching, Reasoning, Planning dan Learning*. Bandung: Informatika.
- 7) Suyanto. 2010. *Algoritma Optimasi Deterministik atau probabilitik*.Yogyakarta: Graha ilmu,
- 8) Wahyu,G.Dan Andi,K. 2012. *Otomatisasi Sistem Penjadwalan Dengan Dekomposisi Attribute Basis Data Menggunakan Algoritma Genetika*.Mataram: SNTEI Fakultas Teknik UNRAM.

FORMULIR BIMBINGAN SKRIPSI

Nama : Arif Cahyono
NIM : 0818043
Masa Bimbingan : 17 November 2012 s/d 17 Mei 2013
Judul Skripsi : Otomatisasi Sistem Penjadwalan Perkuliahan Menggunakan
Algoritma Genetika

NO	TANGGAL	URAIAN	PARAF
1	12/11/2012	Revisi Bab I & II	
2	20/11/2012	Revisi Bab III, ACC Bab I & II	
3	23/1/2013	Revisi Bab IV & V	
4	26/1/2013	Revisi makalah seminar hasil	
5	27/1/2013	Presentasi seminar hasil	
6	12/2/2013	ACC Kompre	

Malang, 12 Februari 2013

Dosen Pembimbing I


Dr. Eng. Aryanto S, ST, MT

NIP. P. 1030800417

FORMULIR BIMBINGAN SKRIPSI

Nama : Arif Cahyono
NIM : 0818043
Masa Bimbingan : 17 November 2012 s/d 17 Mei 2013
Judul Skripsi : Otomatisasi Sistem Penjadwalan Perkuliahan Menggunakan
Algoritma Genetika

NO	TANGGAL	URAIAN	PARAF
1	12/11/2012	Revisi Bab I & Bab II	
2	19/11/2012	Acc Bab I & II, Revisi Bab III	
3	20/11/2012	Acc Bab III	
4	23/1/2013	Revisi Bab IV & V	
5	24/1/2013	Acc Bab IV & V, Revisi Makalah Seminar Hasil	
6	26/1/2013	Acc Makalah Seminar Hasil	
7	27/1/2013	Presentasi Seminar Hasil	
8	12/2/2013	Acc Kompre	

Malang, 12 Februari 2013

Dosen Pembimbing II

Ahmad Faisal, ST

NIP. P. 1031000431





INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo, KM 2 Malang


**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : ARIF CAHYONO
NTM : 0818043
JURUSAN : Teknik Informatika S-1
JUDUL : OTOMATISASI SISTEM PENJADWALAN PERKULIAHAN
MENGUNAKAN ALGORITMA GENETIKA

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Selasa
Tanggal : 19 Februari 2013
Nilai : A

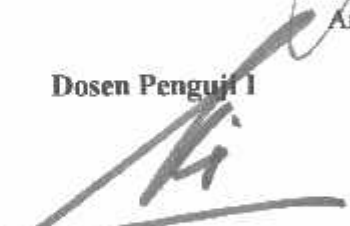
Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST, MT
NIP. 19740416 200501 1 002

Anggota Penguji :

Dosen Penguji I


Karina Auliasari, ST, M. Eng
NIP.P. 1031000426

Dosen Penguji II


Febriana Santi W, S.Kom, M.Kom
NIP.P. 1031000425

Dari gambar blok diagram 3.1 dapat dijelaskan cara kerjanya sebagai berikut :

Saklar ditekan sesuai sandi morse dengan bunyi misalnya :

- --- .-.. --- -. --.
T O L O N G

Kemudian dikirimkan melalui walky talky dengan frekuensi carrier sebesar 400 Mhz, dan di terima oleh walky talky tujuan kemudian tone tersebut disalin menggunakan tone decoder, setelah disalin diproses di mikrokontroller untuk diterjemahkan kedalam huruf yang hasilnya sebuah kalimat TOLONG, ditampilkan di LCD. Bunyi terjadi karena adanya perubahan data digital, untuk logika 0 yang mewakili dot (bunyi pendek) perubahan data digital terjadi setiap $0,5 \text{ s} = 500\text{ms}$ dengan frekuensi tones 1 KHz. Untuk logika 1 yang mewakili dash (bunyi panjang) perubahan data digital terjadi setiap $1,0 \text{ s} = 1000\text{ms}$, dengan frekuensi tones 1 KHz.

Keterangan dari diagram blok :

- Mikrokontroller AT89S51
Berfungsi sebagai pengendali kerja sistem keseluruhan
- Walky Talky
Berfungsi sebagai pengirim dan penerima sandi *morse*.
- Switch Push Button
Berfungsi untuk inputan sandi *morse*.