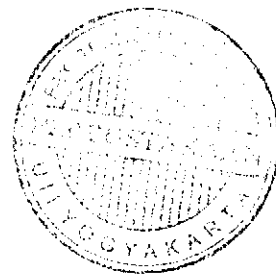


**PENYELESAIAN KNAPSACK PROBLEM DENGAN
ALGORITMA GENETIKA**

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika



Oleh :

Nama : Indra Gunawan

NIM : 02 523 203

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2007

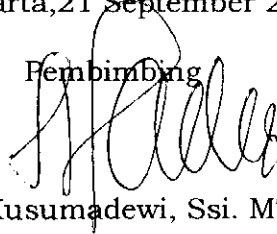
LEMBAR PENGESAHAN PEMBIMBING
PENYELESAIAN KNAPSACK PROBLEM DENGAN
ALGORITMA GENETIKA



Oleh :
Nama : Indra Gunawan
NIM : 02 523 203

Yogyakarta, 21 September 2006

Pembimbing



- Sri Kusumadewi, Ssi. MT

LEMBAR PENGESAHAN PENGUJI

**PENYELESAIAN KNAPSACK PROBLEM DENGAN
ALGORITMA GENETIKA**

TUGAS AKHIR

Oleh :

Nama : **Indra Gunawan**
NIM : **02 523 203**

Telah Dipertahankan di Depan Sidang Pengujian sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas Teknik Industri Universitas Islam Indonesia

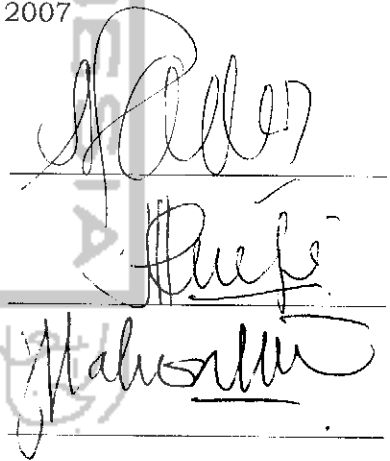
Yogyakarta, 27 Maret 2007

Tim Penguji,

Sri Kusumadewi, S.si., M.T.
Ketua

Taufik Hidayat, S.T., MCS.
Anggota I

Affan Mahtarami, S.Kom.
Anggota II



Yuni Prayudi, S.Si, M.Kom.,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

PERSEMBAHAN

Skripsi ini kupersembahkan untuk:

*Allah SWT Sebagai dzat yang Maha Tinggi
Rasulallah Muhammad SAW keluarga dan para sahabatnya
serta semua pengikutnya sampai akhir zaman*

Kedua Orang Tua ku

*Bapak dan Ibu tercinta yang selama ini selalu mencurahkan cinta dan kasih
sayangnya tanpa pamrih*

*Kakak ku yang telah banyak memberi support, doa dan masukan sehingga saya
menjadi lebih dewasa*

HALAMAN MOTTO

Sesungguhnya sesudah kesulitan itu ada kemudahan, Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain

(Q.S. Alam Nasrah: 6,7)

Tidak ada yang menimpa seorang mukmin berupa sakit, kebingungan, kesedihan, bahkan duri yang menancap kepadanya, kecuali Allah akan menghapus dosa-dosa orang tersebut dengannya.

(HR Bukhari)

Sesungguhnya jika kamu bersyukur pasti akan kami tambahkan kenikmatan, dan jika kamu ingkar sesungguhnya azab-Ku sangatlah pedih

(Q.S. 14:7)

Dan Jadikanlah sabar dan shalat sebagai penolongmu, dan sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusyu.

(Q.S. Al-Baqoroh : 45)

LEMBAR PERNYATAAN KEASLIAN

HASIL TUGAS AKHIR

Saya yang bertandatangan dibawah :

Nama : Indra Gunawan

NIM : 02 523 203

Menyatakan bahwa seluruh komponen dan isi dalam Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, Maret 2007



INDRA GUNAWAN

KATA PENGANTAR

السلام عليكم ورحمة الله وبركاته

Assalamu'alaikum wr. wb

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT, berkat rahmat dan hidayah-NYA lah penulis dapat menyelesaikan Tugas Akhir ini dengan baik.

Tugas Akhir ini selain merupakan salah satu bagian dari mata kuliah di Teknik Informatika Teknik Industri Universitas Islam Indonesia, juga merupakan sarana bagi penulis untuk menambah wawasan baik berupa pedalaman materi dari bidang kecerdasan buatan pada umumnya dan sistem pakar pada khususnya yang telah penulis dapatkan ataupun pengetahuan baru yang tak ternilai harganya.

Selesainya penelitian tugas akhir ini pastilah tidak terlepas dari peranan pihak-pihak yang ikut serta membantu dan mendukung secara moril dan materil, oleh karena itu penyusun ucapkan terimakasih yang terdalam kepada semua pihak yang telah membantu terselesaikannya penelitian tugas akhir ini :

1. Bapak Prof. Dr. Drs. Edy Suandi Hamid, M.Ec, selaku rektor Universitas Islam Indonesia.
2. Bapak Fathul Wahid, ST., M.Sc, selaku dekan FTI UII.
3. Bapak Yudi Prayudi, S.Si. M.Kom, selaku ketua jurusan Teknik Informatika.
4. Ibu Sri Kusumadewi, S.Si., MT, selaku dosen pembimbing.

5. Bapak dan Ibu tercinta, yang telah mencurahkan kasih sayangnya sepanjang hidup ananda.
6. Teman-teman angkatan 2002 semuanya.
7. Semua pihak yang memberikan kontribusi dalam penelitian tugas akhir ini.

Selama suatu karya adalah dari hasil berfikir manusia, maka tidaklah akan lepas dari cacat dan salah, oleh karena itu sumbangsih saran dan kritik akan selalu terbuka demi kesempurnaan penelitian ini selanjutnya.

Walhasil, semoga penelitian tugas akhir ini dapat memberikan wacana baru bagi pembaca dan dunia pemrograman khususnya dibidang *Artificial Intelligence*. Amien...

والسلام عليكم ورحمة الله وبركاته

Wassalamu'aikum wr. wb.

Yogyakarta, September 2006

Penyusun

SARI

Dalam suatu perusahaan dibutuhkan suatu perangkat lunak untuk menentukan barang apa saja yang seharusnya diangkut ke suatu kontainer dengan kapasitas yang terbatas agar keuntungan menjadi optimal. Disini Perangkat lunak ini dapat membantu para pengambil keputusan untuk menentukan barang apa saja yang seharusnya diangkut.

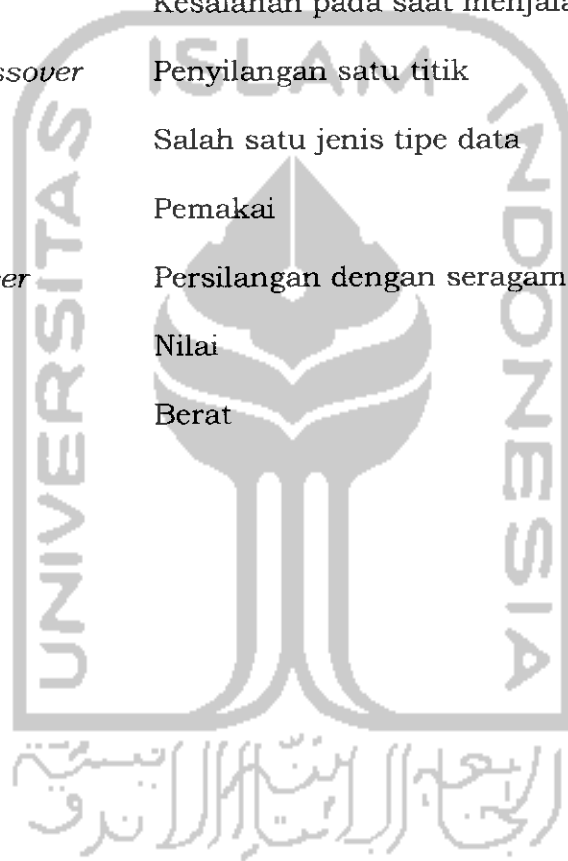
Perangkat lunak ini juga dapat menampung hingga 100 barang dan variabel-variabel genetika yang bisa diubah-ubah disesuaikan dengan permasalahan yang ada, terdapat juga menu cetak sehingga pemakai bisa melakukan proses optimasi berulang ulang dan menyimpannya pada suatu file dengan format .doc untuk digunakan sewaktu waktu. Diharapkan dengan dikembangkannya perangkat lunak ini bisa menambah khasanah dalam bidang Kecerdasan buatan.



TAKARIR

<i>Algoritma Genetika</i>	Algoritma pencarian yang berdasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam.
<i>Database</i>	Tempat untuk menyimpan data digital
<i>Delete</i>	Proses menghapus
<i>Developer</i>	Pihak yang mengembangkan/membangun suatu produk
<i>Fitness</i>	Kemampuan untuk bertahan hidup
<i>Flow chart</i>	Diagram alir
<i>Form</i>	Formulir
<i>GUI</i>	Tampilan grafik yang memudahkan pemakai berinteraksi dengan perangkat lunak
<i>Information Teknologi</i>	Ilmu yang mempelajari seputar teknologi informasi
<i>Insert</i>	Memasukkan data
<i>Knapsack</i>	Karung, kantung, atau buntelan
<i>Konvergen</i>	Konvergen
<i>Language error</i>	Kesalahan bahasa
<i>Library Research</i>	Penelitian berdasar informasi pada suatu perpustakaan
<i>logical error</i>	Kesalahan logika
<i>Multi Point Crossover</i>	Persilangan banyak titik

<i>Mutation</i>	Perubahan genetic pada suatu individu untuk menyesuaikan dengan lingkungannya
<i>Permutation Crossover</i>	Persilangan dengan permutasi
<i>Probabilitas</i>	Peluang terjadinya sesuatu
<i>Roulette wheel</i>	Roda Rolet
<i>Run time error</i>	Kesalahan pada saat menjalankan program
<i>Single Point Crossover</i>	Penyilangan satu titik
<i>String</i>	Salah satu jenis tipe data
<i>User</i>	Pemakai
<i>Uniform Crossover</i>	Persilangan dengan seragam
<i>Value</i>	Nilai
<i>Weight</i>	Berat



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING	ii
LEMBAR PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
HALAMAN PERNYATAAN KEASLIAN HASIL TUGAS AKHIR	vi
KATA PENGANTAR	vii
SARI.....	ix
TAKARIR	x
DAFTAR ISI	xii
DAFTAR TABEL	xvi
DAFTAR GAMBAR.....	xvii
BAB I	
PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian.....	4
1.6.1 Metode Pengumpulan Data	4
1.6.2 Metode Pengembangan Sistem.....	4
1.7 Sistematika Penulisan	5
BAB II.....	7
LANDASAN TEORI.....	7
2.1 Algoritma Genetika	7
2.1.1 Sejarah	7
2.1.2 Definisi	8
2.1.3 Deskripsi.....	9
2.1.3.1 Representasi Kromosom	9

2.1.3.2 Fungsi Evaluasi	9
2.1.3.2 Seleksi	10
2.2.3.3 Operator Genetik	15
2.2 Knapsack Problem	19
2.2.1 Definisi	19
2.2.2 Deskripsi	19
2.2.3 Data Barang	20
BAB III	21
ANALISIS KEBUTUHAN PERANGKAT LUNAK	21
3.1 Metode Analisis	21
3.2 Hasil Analisis	21
3.2.1 Kebutuhan masukan	21
3.2.2 Proses	22
3.2.3 Keluaran	23
3.3 Fungsi-fungsi yang Dibutuhkan	24
3.3.1 Basis data	24
3.3.2 Algoritma genetika	24
BAB IV	25
PERANCANGAN PERANGKAT LUNAK	25
4.1 Metode Perancangan	25
4.2 Hasil Perancangan	26
4.2.1 Perancangan flow chart	26
4.2.1.1 Proses utama	26
4.2.1.2 Pembangkitan populasi awal	28
4.2.1.3 Penghitungan fitnes	30
Keterangan Gambar 4.3 :	31
4.2.1.4 Seleksi roda rolet	32
4.2.2 Perancangan representasi kromosom	52
4.2.3 Penentuan nilai fitnes	53
4.2.4 Struktur data	55
4.2.5 Perancangan basis data	55
4.2.5.1 Struktur tabel basis data	55
4.2.6 Perancangan antarmuka	55
4.2.6.1 Perancangan antarmuka menu	55
4.2.6.2 Perancangan antarmuka form utama	56
4.2.6.3 Perancangan antarmuka form report	57
4.2.6.4 Perancangan antarmuka form grafik	58
4.2.6.5 Perancangan antarmuka form hasil optimasi	59
4.2.6.6 Perancangan antarmuka form insert data	60
4.2.6.7 Perancangan antarmuka form edit data	61
4.2.6.8 Perancangan antarmuka form delete data	61
4.2.6.9 Perancangan antarmuka form about	61
4.2.7 Perancangan operator algoritma genetika	63
4.2.7.1 Seleksi	63
4.2.7.2 Crossover	63

2.1.3.2 Fungsi Evaluasi	9
2.1.3.2 Seleksi.....	10
2.2.3.3 Operator Genetik.....	15
2.2 Knapsack Problem	19
2.2.1 Definisi.....	19
2.2.2 Deskripsi.....	19
2.2.3 Data Barang.....	20
BAB III	21
ANALISIS KEBUTUHAN PERANGKAT LUNAK	21
3.1 Metode Analisis	21
3.2 Hasil Analisis	21
3.2.1 Kebutuhan masukan.....	21
3.2.2 Proses.....	22
3.2.3 Keluaran.....	23
3.3 Fungsi-fungsi yang Dibutuhkan	24
3.3.1 Basis data.....	24
3.3.2 Algoritma genetika.....	24
BAB IV	25
PERANCANGAN PERANGKAT LUNAK	25
4.1 Metode Perancangan	25
4.2 Hasil Perancangan	26
4.2.1 Perancangan flow chart	26
4.2.1.1 Proses utama	26
4.2.1.2 Pembangkitan populasi awal	28
4.2.1.3 Penghitungan fitness	30
Keterangan Gambar 4.3 :	31
4.2.1.4 Seleksi roda rolet	32
4.2.2 Perancangan representasi kromosom.....	52
4.2.3 Penentuan nilai fitness.....	53
4.2.4 Struktur data	55
4.2.5 Perancangan basis data.....	55
4.2.5.1 Struktur tabel basis data	55
4.2.6 Perancangan antarmuka.....	55
4.2.6.1 Perancangan antarmuka menu.....	55
4.2.6.2 Perancangan antarmuka form utama.....	56
4.2.6.3 Perancangan antarmuka form report.....	57
4.2.6.4 Perancangan antarmuka form grafik	58
4.2.6.5 Perancangan antarmuka form hasil optimasi.....	59
4.2.6.6 Perancangan antarmuka form insert data	60
4.2.6.7 Perancangan antarmuka form edit data	61
4.2.6.8 Perancangan antarmuka form delete data	61
4.2.6.9 Perancangan antarmuka form about	61
4.2.7 Perancangan operator algoritma genetika.....	63
4.2.7.1 Seleksi.....	63
4.2.7.2 Crossover	63

4.2.7.3 Mutasi	63
4.2.8 Perancangan file	64
4.2.8.1 Perancangan file input	64
4.2.8.2 Perancangan file output.....	64
BAB V	65
IMPLEMENTASI PERANGKAT LUNAK.....	65
5.1 Batasan Implementasi	65
5.2 Implementasi Perangkat Lunak.....	65
5.2.1 Perancangan antarmuka perangkat lunak	65
5.2.1.1 Perancangan antarmuka form utama.....	65
5.2.1.2 Perancangan antarmuka form report.....	66
5.2.1.3 Perancangan antarmuka form grafik	67
5.2.1.4 Perancangan antarmuka form hasil optimasi.....	68
5.2.1.5 Perancangan antarmuka form insert data	68
5.2.1.6 Perancangan antarmuka form edit data	69
5.2.1.7 Perancangan antarmuka form delete data.....	69
5.2.1.8 Perancangan antarmuka form about	70
BAB VI	71
ANALISIS KINERJA PERANGKAT LUNAK.....	71
6.1 Proses analisa sistem.....	71
6.1.1 Kesalahan dalam basis data.....	71
6.1.1.1 Kesalahan dalam memasukkan nilai string	71
6.1.2 Kesalahan dalam algoritma genetika.....	72
6.1.2.1 Kesalahan dalam memasukkan nilai max generasi	72
6.1.2.2 Kesalahan dalam memasukkan nilai max populasi	72
6.1.2.3 Kesalahan dalam memasukkan nilai probabilitas	72
crossover	72
6.1.2.4 Kesalahan dalam memasukkan nilai probabilitas	73
mutasi	73
6.1.2.5 Kesalahan dalam memasukkan nilai kapasitas	73
max peti	73
6.1.2.6 Kesalahan dalam memasukkan nilai variabel	73
breeder	73
6.1.2.7 Kesalahan dalam memasukkan nilai string.....	74
6.2 Pengujian Sistem.....	74
6.2.1 Pengujian dengan basis data.....	74
6.2.2 Pengujian dengan variabel genetika.....	76
6.2.2.1 Pengujian dengan variabel Pc (probabilitas	76
crossover).....	76
6.2.2.2 Pengujian dengan variabel Pm (probabilitas mutasi).	77
.....	77
6.2.2.3 Pengujian dengan variabel BGA (Breeder GA).	78
6.2.2.4 Pengujian dengan variabel populasi.	79

6.2.2.5 Pengujian dengan variabel generasi.....	80
6.2.2.6 Pengujian dengan type crossover.....	81
6.2.2.7 Kesimpulan.....	82
BAB VII.....	83
KESIMPULAN DAN SARAN.....	83
7.1 Kesimpulan.....	83
7.2 Saran.....	83
DAFTAR PUSTAKA.....	84



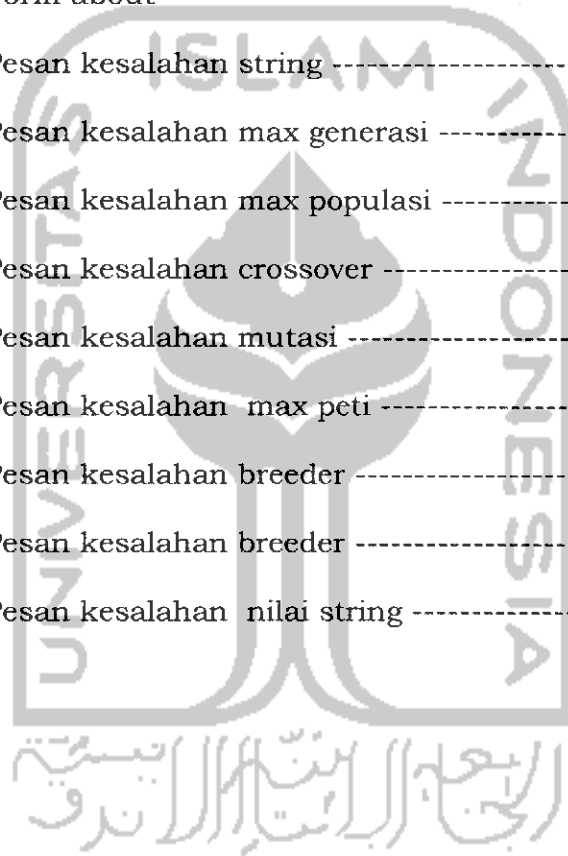
DAFTAR TABEL

Tabel 2.1	Data Barang-----	18
Tabel 4.1	Daftar Barang-----	54
Tabel 4.2	Struktur Basis data -----	55
Tabel 6.1	Variabel genetika-----	75
Tabel 6.2	Hasil optimasi-----	76
Tabel 6.3	Variabel genetika-----	76
Tabel 6.4	Hasil optimasi-----	77
Tabel 6.5	Variabel genetika-----	77
Tabel 6.6	Hasil optimasi-----	78
Tabel 6.7	Variabel genetika-----	78
Tabel 6.8	Hasil optimasi-----	79
Tabel 6.9	Variabel genetika-----	79
Tabel 6.10	Hasil optimasi-----	80
Tabel 6.11	Variabel genetika-----	80
Tabel 6.12	Hasil optimasi-----	81
Tabel 6.13	Variabel genetika-----	81
Tabel 6.14	Hasil optimasi-----	82
Tabel 6.15	Keadaan Optmal-----	82

DAFTAR GAMBAR

Gambar 2.1	Roulette wheel selection-----	12
Gambar 2.2	Pemetaan individu-----	12
Gambar 2.3	Penyilangan satu titik -----	15
Gambar 4.1	Flow chart Proses utama -----	16
Gambar 4.2	Flowchart populasi awal -----	28
Gambar 4.3	Flowchart penghitungan fitness -----	30
Gambar 4.4	Seleksi roda rolet -----	33
Gambar 4.5	Crossover satu titik -----	37
Gambar 4.6	Crossover banyak titik -----	41
Gambar 4.7	Crossover seragam -----	45
Gambar 4.8	Crossover 2 titik -----	49
Gambar 4.9	Representasi kromosom -----	52
Gambar 4.10	Representasi kromosom -----	53
Gambar 4.11	Perancangan form utama -----	57
Gambar 4.12	Perancangan form report -----	58
Gambar 4.13	Perancangan form grafik -----	59
Gambar 4.14	Perancangan form optimasi -----	60
Gambar 4.15	Perancangan form insert data -----	60
Gambar 4.16	Perancangan form edit data -----	61
Gambar 4.17	Perancangan form hapus data -----	61
Gambar 4.18	Perancangan form about -----	62
Gambar 5.1	Form utama -----	66
Gambar 5.2	Form report -----	66

Gambar 5.3	Form Grafik -----	67
Gambar 5.4	Form optimasi -----	68
Gambar 5.5	Form insert data -----	69
Gambar 5.6	Form edit data -----	69
Gambar 5.7	Form delete data -----	70
Gambar 5.8	Form about -----	70
Gambar 6.1	Pesan kesalahan string -----	71
Gambar 6.2	Pesan kesalahan max generasi -----	72
Gambar 6.3	Pesan kesalahan max populasi -----	72
Gambar 6.4	Pesan kesalahan crossover -----	72
Gambar 6.5	Pesan kesalahan mutasi -----	73
Gambar 6.6	Pesan kesalahan max peti -----	74
Gambar 6.7	Pesan kesalahan breeder -----	74
Gambar 6.8	Pesan kesalahan breeder -----	74
Gambar 6.9	Pesan kesalahan nilai string -----	74



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan perangkat keras dan perangkat lunak sekarang ini sangat mempengaruhi pola pemakaian komputer di segala bidang. Kebutuhan informasi yang sifatnya terkini (*up to date*) sangat penting, hal ini dapat dilihat dengan semakin cepatnya perkembangan teknologi, khususnya di bidang *IT (Information Technology)*.

Sebuah informasi pada proses pengangkutan barang pada perusahaan sangatlah berpengaruh dalam setiap proses pengambilan keputusan manajer pada perusahaan. Oleh karena itu dibutuhkan suatu perangkat lunak yang dapat membantu memutuskan jenis barang apa yang akan diangkut oleh suatu mesin pengangkut / kontainer sehingga dapat dicapai keuntungan yang maksimal dengan biaya pengangkutan minimal.

Peran Algoritma genetika dalam hal ini digunakan sebagai suatu metode untuk membantu menentukan kombinasi barang yang akan diangkut ke suatu pengangkut/kontainer sehingga didapat keuntungan yang maksimal.

1.2 Rumusan Masalah

Dari uraian di atas dapat ditarik suatu rumusan masalah yaitu: Bagaimana merancang dan membangun serta mengimplemetasikan *Knapsack Problem* ke dalam suatu aplikasi perangkat lunak yang dinamis, di mana aplikasi perangkat lunak tersebut berfungsi sebagai

mesin untuk melakukan optimasi pada *Knapsack Problem* dengan metode *Algoritma Genetika*.

1.3 Batasan Masalah

Implementasi yang dilakukan pada penelitian tugas akhir ini mempunyai batasan masalah sebagai berikut:

- a. Ukuran maksimal jumlah barang yang bisa diinputkan oleh *User* adalah 100 barang.
- b. Optimasi peramalan dilakukan oleh Perangkat Lunak dengan menggunakan metode *Algoritma Genetika*.
- c. Jumlah Populasi dalam satu Generasi yang bisa ditangani pada proses *Algoritma Genetika* adalah sampai 100 Populasi.
- d. Jumlah Generasi dalam satu Iterasi yang bisa ditangani pada proses *Algoritma Genetika* adalah sampai 100 Generasi.
- e. Nilai Maksimal kapasitas per kontainer yang bisa ditangani pada proses *Algoritma Genetika* adalah mencapai 1.000 satuan.
- f. Perangkat Lunak ini melakukan proses Seleksi Alam pada proses *Algoritma Genetika* dengan menggunakan metode *Roulette Wheel Selection (Seleksi Roda Roulette)*.
- g. *User* bisa memilih metode *Crossover* yang digunakan pada proses *Algoritma Genetika* dengan memilih pada menu yang tersedia yang terdiri 4 item:
 1. *Single Point Crossover (Penyilangan Satu Titik)*.
 2. *Multi Point Crossover (Penyilangan Banyak Titik)*.
 3. *Uniform Crossover (Penyilangan Seragam)*.

4. *Permutation Crossover (Penyilangan dengan Permutasi).*

- h. Perangkat lunak ini tidak mempertimbangkan Aspek Ekonomi, Sosial maupun manajerial dalam proses optimasi peramalan melainkan berdasarkan nilai Fitness yang didapat dari pengalihan variabel berat tiap barang dikali nilai / value / keuntungan pada tiap barang.

1.4 Tujuan Penelitian

Tujuan utama penelitian tugas akhir ini adalah untuk mengimplementasikan Knapsack Problem ke dalam komputer. Secara lebih spesifik, tujuan penelitian adalah :

1. Membuat aplikasi perangkat Lunak yang bisa memecahkan permasalahan Knapsack Problem.
2. Penggunaan Artificial Intellegence untuk memecahkan permasalahan *Knapsack Problem*

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian tugas akhir ini adalah terbentuknya suatu aplikasi perangkat lunak yang dapat memberikan alternatif yang lebih mudah untuk memecahkan masalah yang berhubungan dengan Knapsack Problem dengan Metode Algoritma Genetika.

1.6 Metodologi Penelitian

Metode penelitian adalah metode yang digunakan untuk menyusun penelitian ini dalam yaitu digunakan dalam mengumpulkan informasi dan mengembangkan perangkat lunak sehingga didapatkan hasil yang maksimal.

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data adalah metode yang digunakan dalam mengumpulkan semua informasi dari berbagai sumber yang berguna dalam menyusun penelitian ini.

a. Metode Kepustakaan (*Library Research*)

Metode Kepustakaan (*Library Research*) adalah dengan mengumpulkan data lewat buku-buku referensi yang relevan dengan permasalahan yang dihadapi.

b. Telaah Dokumen

Telaah dokumen adalah mempelajari dokumen, artikel, dan catatan lain yang juga masih berkaitan dengan bidang permasalahan yang dihadapi.

1.6.2 Metode Pengembangan Sistem

Pada penelitian ini untuk mencapai hasil yang baik dalam merancang program, maka metodologi yang digunakan adalah :

a. Obyek Penelitian

Dalam hal ini yang menjadi obyek penelitian adalah Proses Optimasi dengan Algoritma Genetika.

b. Analisis Kebutuhan Perangkat Lunak

Melakukan analisis terhadap permasalahan yang ada untuk lebih mendapatkan gambaran yang luas mengenai sistem yang dibutuhkan dalam mengatasi permasalahan tersebut. Sistem yang dibutuhkan tersebut lebih terarah ke perangkat lunak yang melibatkan pemakai (*user*) dan pengembang (*developer*).

c. Perancangan Sistem

Metode yang digunakan dalam perancangan sistem pada perangkat lunak ini adalah dengan menggunakan diagram alir (*flowchart*) dari tingkat umum menjadi tingkat yang lebih spesifik.

d. Implementasi dan Pengujian Perangkat Lunak

Pada tahap ini digunakan untuk pengembangan aplikasi perangkat lunak dilanjutkan dengan pengujian aplikasi perangkat lunak tersebut.

1.7 Sistematika Penulisan

a. Bab 1 Pendahuluan

Pada bab Pendahuluan ini diawali dengan penjelasan mengenai latar belakang masalah, kemudian dilanjutkan dengan rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

b. Bab 2 Landasan teori

Pada bab Landasan Teori ini membahas lebih dalam mengenai landasan teori yang dapat membantu para pembaca dalam memahami implementasi yang akan dilakukan. Landasan teori tersebut mencakup *Knapsack Problem*, *Artificial Intelligence (Kecerdasan Buatan)*, *Algoritma Genetika*, *Database*.

c. Bab 3 Analisis kebutuhan perangkat lunak

Pada bab Analisis kebutuhan perangkat lunak ini membahas tentang metode analisis yang dipakai pada analisis kebutuhan perangkat lunak dan pemilihan kebutuhan-kebutuhan dalam pembuatan perangkat lunak yang meliputi: kebutuhan masukan, proses dan keluaran serta fungsi-fungsi yang dibutuhkan dalam aplikasi yang akan dibuat.

d. Bab 4 Perancangan perangkat lunak

Pada bab Perancangan perangkat lunak ini membahas tentang metode yang dipakai dalam perancangan perangkat lunak dan hasil perancangan perangkat lunak yang meliputi: struktur data, arsitektur perangkat lunak, prosedur-prosedur dan antarmuka.

e. Bab 5 Implementasi perangkat lunak

Pada bab Implementasi perangkat lunak ini membahas tentang batasan implementasi perangkat lunak yang meliputi: asumsi-asumsi yang dipakai, lingkungan pengembangan, bahasa dan kompilator yang dipakai disertai alasan pemilihannya, dan batasan lain yang dibuat dan ditemui selama pengembangan perangkat lunak. Bab ini juga berisi mengenai dokumen implementasi perangkat lunak yang meliputi: implementasi struktur data, prosedur-prosedur dalam bahasa pemrograman yang dipilih serta antarmuka. Khusus untuk prosedur-prosedur disertai keterangan untuk setiap proses yang ada.

f. Bab 6 Analisis kinerja perangkat lunak

Pada bab Analisis kinerja perangkat lunak ini berisi mengenai dokumentasi hasil pengujian terhadap perangkat lunak yang

dibandingkan kebenaran dan kesesuaiannya dengan kebutuhan perangkat lunak untuk kemudian dianalisa.

g. Bab 7 Kesimpulan dan saran

Pada bab kesimpulan dan saran ini sebagai bab terakhir diisi dengan kesimpulan dan saran dari penelitian tugas akhir ini.



BAB II

LANDASAN TEORI

Bab ini membahas mengenai landasan teori yang menjadi dasar pengetahuan untuk tahap implementasi. Teori-teori yang akan diuraikan di sini adalah *Algoritma Genetika, Knapsack Problem Basis Data*.

2.1 Algoritma Genetika

2.1.1 Sejarah

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 evolusi, yaitu :

- a. Kemampuan organisme untuk melakukan reproduksi.
- b. Keberadaan populasi organisme yang bisa melakukan reproduksi.
- c. Keberagaman organisme dalam suatu populasi.
- d. Perbedaan kemampuan untuk survive.

Individu yang lebih kuat (fit) akan memiliki tingkat survival dan tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang fit. Pada kurun waktu tertentu (sering dikenal dengan

istilah generasi), populasi secara keseluruhan akan lebih banyak memuat organisme yang fit.

Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John hollan mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma Genetika adalah sirmulasi dari proses evolusi Darwin dan operasi genetika kromosom[SRI03].

2.1.2 Definisi

Algoritma genetik merupakan evolusi/perkembangan dunia komputer dalam bidang kecerdasan buatan (*artificial intellegence*). Sebenarnya algoritma genetik ini terinspirasi oleh teori evolusi Darwin (walaupun pada kenyatanya teori tersebut terbukti keliru). Jadi Definisi dari Algoritma genetik adalah algoritma pencarian yang berdasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. Dalam aplikasi algoritma genetik, variabel solusi dikodekan kedalam struktur string yang merepresentasikan barisan gen, yang merupakan karakteristik dari solusi problem.

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

berdasarkan suatu fungsi evaluasi (Gen,1997). Setelah beberapa generasi maka algoritma genetik akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal (Goldberg,1989).

2.1.3 Deskripsi

2.1.3.1 Representasi Kromosom

Cara merepresentasikan permasalahan dalam kromosom merupakan suatu hal yang penting dalam algoritma genetik. Ada beberapa model representasi kromosom yang dapat dipergunakan untuk menyelesaikan suatu masalah, salah satunya adalah *operationbased representation*.

Prinsip dari *Operation-Base Representation* adalah semua operasi pada suatu pekerjaan akan dikodekan dengan simbol yang sama, kemudian diinterpretasikan menurut urutannya dalam sequence. Sehingga untuk problem dengan a-pekerjaan b-mesin, kromosom akan terdiri dari $a \times b$ gen. Dimana setiap pekerjaan akan mengalami pengulangan sebanyak m kali. Contoh kromosom untuk hal tersebut yaitu 3x3 yang dibentuk dengan representasi ini adalah {1 2 2 3 1 3 2 1 3}.

2.1.3.2 Fungsi Evaluasi

Fungsi evaluasi merupakan dasar untuk proses seleksi. Langkah-langkahnya yaitu string dikonversi ke parameter fungsi, fungsi objective-nya dievaluasi, kemudian mengkonvert fungsi objektif tersebut ke dalam *fitness*. Dimana untuk maksimasi problem, *fitness* sama dengan fungsi objective-nya. Output dari fungsi *fitness* dipergunakan sebagai dasar

untuk menyeleksi individu pada generasi berikutnya. Invers dari makespan dapat digunakan untuk menentukan *fitness* pada tiap kromosom. Misalnya Ck_{max} menunjukkan makespan untuk kromosom ke- k , *fitness* dihitung dengan menggunakan (Gen, 1997) :

$$Eval(Vk) = 1 / Ck_{max}$$

2.1.3.2 Seleksi

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit.

Roulette wheel merupakan salah satu metode seleksi yang banyak dipergunakan. Roulette wheel menyeleksi populasi baru dengan distribusi probabilitas yang berdasarkan nilai *fitness*.

Ada beberapa metode seleksi dari induk, antara lain :

a. Rank-based fitness assignment.

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

Misalkan N adalah jumlah individu dalam suatu populasi. Pos adalah posisi individu dalam populasi tersebut (posisi terendah suatu individu adalah $Pos=1$, dan posisi tertingginya adalah $Pos=N$). Sedangkan SP adalah selective pressure. Nilai *fitness* suatu individu dapat dihitung sebagai:

1. Linear ranking :

$$Fitness(Pos) = 2 - SP + 2 (SP-1) (Pos-1) ; I = 1 \dots N.$$

Nilai $SP \in [1,2]$.

2. Non -linear rangking :

$$\text{Fitness}(\text{Pos}) = N_{\text{ind}} * x^{(\text{Pos}-1)} / \sum (X^{(i-1)}); i=1 \dots N.$$

Sedangkan x dihitung sebagai akar polynomial:

$$(\text{SP}-1) * x^{(N-1)} + \text{SP} * x^{(N-2)} + \dots + \text{SP} * x + \text{SP} = 0$$

Nilai SP $\in [1, N-2]$.

b. Roulette wheel Selection.

Metode seleksi roda roulette ini merupakan metode yang paling sederhana, dan sering juga dikenal dengan nama stochastic sampling with replacement. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran fitnessnya.

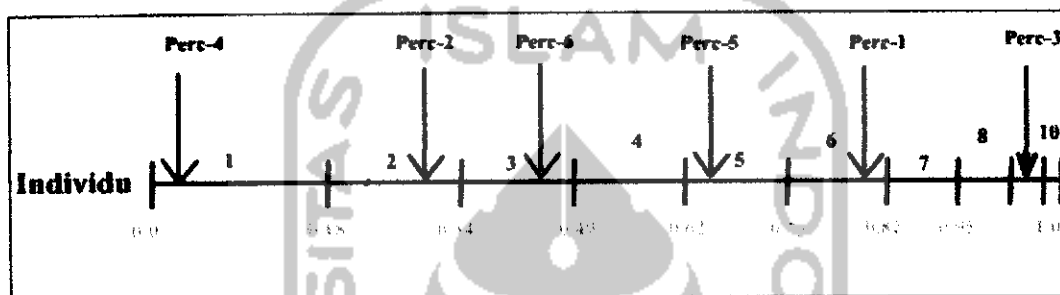
Sebuah bilangan random dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.

Pada gambar 2.1 menunjukkan probabilitas seleksi dari 11 individu. Individu pertama memiliki fitness terbesar, dengan demikian dia juga memiliki interval terbesar. Sedangkan individu ke-10 memiliki fitness terkecil kedua. Individu ke -11 memiliki fitness terkecil(=0), interval terkecil sehingga tidak memiliki kesempatan untuk melakukan reproduksi.

Gambar 2.1

Individu ke -	1	2	3	4	5	6	7	8	9	10	11
Nilai fitness	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Probabilitas	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

Gambar 2.2



Setelah dilakukan seleksi, maka individu-individu yang terpilih adalah :

1 2 3 5 6 9

c. Stochastic universal sampling.

Stochastic universal sampling memiliki nilai bias nol dan penyebaran yang minimum. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki urutan yang sama dengan ukuran fitnessnya seperti halnya pada seleksi roda roulette. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi pada garis tersebut. Andaikan N adalah jumlah individu yang akan diseleksi, maka jarak antar pointer adalah $1/N$, dan posisi pointer pertama diberikan secara acak pada range $[1, 1/N]$.

Apabila ada 6 individu yang akan diseleksi, maka jarak antar pointer adalah $1/6 = 0,167$. Sehingga, misalkan bilangan random yang dibangkitkan pada $[0,0.0167]$ adalah 0,1, maka yang diperoleh setelah seleksi adalah:

d. Local selection.

Pada seleksi local, setiap individu yang berada di dalam konstrain tertentu disebut dengan nama lingkungan lokal. Interaksi antar individu hanya dilakukan di dalam wilayah tersebut. Lingkungan tersebut ditetapkan sebagai struktur dimana populasi tersebut terdistribusi. Lingkungan tersebut juga dapat dipandang sebagai kelompok pasangan-pasangan yang potensial.

Langkah pertama adalah menyeleksi separo pertama dari populasi yang berpasangan secara random (atau menggunakan salah satu metode yang sudah dijelaskan sebelumnya, seperti *stochastic universal sampling*). Kemudian lingkungan baru tersebut diberikan pada setiap individu yang terseleksi. Pada lingkungan yang baru tersebut kita bias menyeleksi pasangan-pasangan yang cocok (pasangan yang terbaik, pasangan yang memiliki fitness proporsional, atau pasangan yang seragam). Struktur lingkungan pada seleksi lokal ini dapat berbentuk :

1. Linear : Full ring dan half ring
2. Dimensi-2 :
 - full cross dan half cross
 - full star dan half star
3. Dimensi-3 dan struktur yang lebih omplek yang merupakan kombinasi dari kedua struktur di atas.

Jarak antara individu dengan struktur tersebut akan sangat menentukan ukuran lingkungan (jumlah tetangga dari suatu individu). Jarak antara satu individu dengan individu yang lainnya tentu saja akan terbatas oleh ukuran lingkungan suatu individu. Individu yang terdapat dalam lingkungan dengan ukuran yang lebih kecil, tentu saja akan lebih terisolasi jika dibandingkan dengan individu yang terletak pada lingkungan dengan ukuran yang lebih besar.

e. Truncation selection.

Pada metode-metode seleksi yang telah dijelaskan terdahulu, seleksi dilakukan secara alami. Pada seleksi dengan pemotongan ini, lebih berkesan sebagai seleksi buatan. Seleksi ini biasanya digunakan oleh populasi yang jumlahnya sangat besar. Pada metode ini, individu-individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang trunc yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk berkisar antara 50%-10%. Individu-individu yang ada di bawah nilai ambang ini tidak akan menghasilkan keturunan.

f. Tournament selection.

Pada metode seleksi dengan turnamen ini, akan ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara random dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran tour yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi).

2.2.3.3 Operator Genetik

Operator genetik dipergunakan untuk mengkombinasi (modifikasi) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya. Ada tiga operator genetik yaitu crossover dan mutasi dan BGA.

a. Crossover

Crossover membangkitkan offspring baru dengan mengganti sebagian informasi dari parents (Orang tua/induk).

Ada beberapa metode crossover:

1. Penyilangan satu titik (Single-point crossover)

Pada penyilangan satu titik, posisi penyilangan k ($k=1,2,\dots,N-1$) dengan N =panjang kromosom diseleksi secara random. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak.

Gambar 2.3



Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : 0 1 1 1 0 | 0 1 0 1 1 1 0

Induk 2 : 1 1 0 1 0 | 0 0 0 1 1 0 1

Posisi penyilangan yang terpilih : misalkan 5

Setelah penyilangan , diperoleh kromosom-kromosom baru :

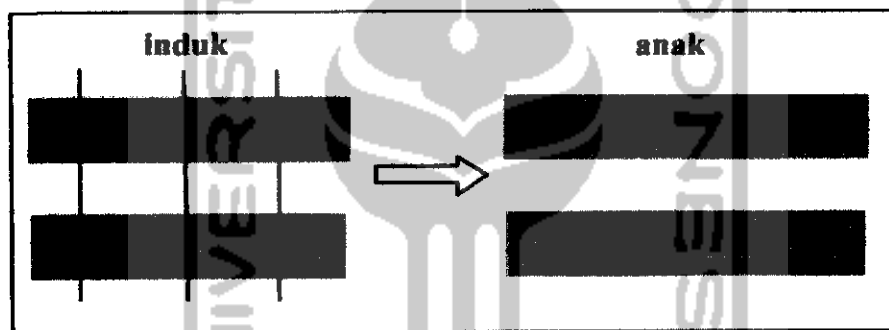
Induk 1 : 0 1 1 1 0 | 0 0 0 1 1 0 0

Induk 2 : 1 1 0 1 0 | 0 1 0 1 1 1 0

2. Penyilangan banyak titik (Multi-point crossover)

Pada penyilangan banyak titik, m posisi penyilangan k_i ($k=1,2,\dots,N-1$, $i=1,2,\dots,m$) dengan N =panjang kromosom diseleksi secara random dan tidak diperbolehkan ada posisi yang sama, serta diurutkan naik. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak.

Gambar 2.4



Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 0 1 1 1 0 0 1 0 1 1 1 0

Induk 2: 1 1 0 1 0 0 0 0 1 1 0 1

Posisi penyilangan yang terpilih:

Misalkan ($m=3$): 2 6 10

Setelah penyilangan, diperoleh kromosom-kromosom baru:

Anak 1: 0 1 | 0 1 0 0 | 1 0 1 1 | 0 1

Anak 2: 1 1 | 1 1 0 0 | 0 0 1 1 | 1 0

3. Penylangan seragam (Uniform crossover)

Pada penylangan seragam, setiap lokasi memiliki potensi sebagai tempat penylangan. Sebuah mask penylangan dibuat sepanjang panjang kromosom secara random yang menunjukkan bit bit dalam mask yang mana induk akan mensupply anak dengan bit bit yang ada.

Induk mana yang akan menyumbangkan bit ke anak dipilih secara random dengan probability yang sama. Disini anak₁ akan dihasilkan dari induk₁ Jika bit mask bernilai 1, atau sebaliknya anak₁ akan dihasilkan dari induk₂ jika bit mask bernilai 0, Sedangkan anak₂ dihasilkan dari kebalikan mask.

Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : 0 1 1 1 0 0 1 0 1 1 1 0

Induk 2 : 1 1 0 1 0 0 0 0 1 1 0 1

Mask bit:

Sampel 1 : 1 0 0 1 1 1 0 0 1 1 0 1

Sampel 2 : 0 1 1 0 0 0 1 1 0 0 1 0

4. Penylangan dengan permutasi (Permutation crossover)

Pada penylangan dengan permutasi ini, kromosom-kromosom anak diperoleh dengan cara memilih sub-barisan suatu tour dari satu induk dengan tetap menjaga urutan dan posisi sejumlah.

Misal :

Induk 1 : (1 2 3 | 4 5 6 7 | 8 9)

Induk 2 : (4 5 3 | 1 8 7 6 | 9 2)

Anak 1 : (x x x | 1 8 7 6 | x x)

Anak 2 : (x x x | 4 5 6 7 | x x)

Dari sini kita memperoleh pemetaan :

1-4, 8-5, 7- 6 , 6-7

Kemudian kita copy sisa gen di induk-1 ke anak anak-1 dengan menggunakan pemetaan yang sudah ada.

Anak 1 : (1-4 2 3 | 1 8 7 6 | 6-5 10)

Anak 1 : (4 2 3 | 1 8 7 6 | 5 9)

Lakukan hal yang sama untuk anak-2.

Anak 2: (4-1 5-8 2 | 4 5 6 7 | 9 2)

Anak 2: (1 8 3 | 4 5 6 7 | 9 2)

b. Mutasi

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

c. BGA(Breeder GA)

Pada metode Algoritma Genetika, populasi awal dibangkitkan secara random sehingga ada kemungkinan individu yang mempunyai nilai fitness baik tidak ikut di bangkitkan, untuk itulah seorang ahli bernama Muhlenbein mengusulkan adanya variabel BGA (*Breeder Genetic Algorithm*), BGA ini mempunyai parameter r yang berisi individu individu terbaik. Kromosom kromosom ini akan tetap dipertahankan pada generasi berikutnya dengan cara menggantikan sebanyak r kromosom pada generasi tersebut secara acak[SR03].

2.2 Knapsack Problem

2.2.1 Definisi

Knapsack adalah karung, kantung, atau buntelan. Karung digunakan untuk memuat sesuatu. Dan tentunya tidak semua objek dapat ditampung di dalam karung. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (weight) lebih kecil atau sama dengan ukuran kapasitas karung.

2.2.2 Deskripsi

Knapsack problem merupakan suatu masalah bagaimana menentukan suatu item (barang) dari sekumpulan barang dimana setiap barang mempunyai berat dan keuntungan yang akan dimasukkan dalam sebuah tempat (knapsack) yang mempunyai kapasitas yang terbatas sehingga didapat keuntungan yg maksimal[WKP06].

2.2.3 Data Barang

Data barang adalah suatu basis data yang berfungsi untuk menyimpan data barang, contoh tabelnya seperti tertera pada **Tabel 2.1**.

Tabel 2.1

No	Nama Barang	Berat Barang	Keuntungan
1.	Barang 1	500	10000
2.	Barang 2	200	20000
3.	Barang 3	230	3000
4.	Barang 4	153	5500
5.	Barang 5	250	15000
6.	Barang 6	350	17000
7.	Barang 7	400	3500
8.	Barang 8	100	650
9.	Barang 9	50	850
10.	Barang 10	650	53000

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

3.1 Metode Analisis

Metode analisis yang dipakai dalam penelitian tugas akhir ini diawali dengan metode pengumpulan data kemudian menganalisa data yang berhasil dikumpulkan dengan metode analisis berarah-obyek.

3.2 Hasil Analisis

Setelah dianalisis, maka dapat diketahui apa yang menjadi masukan, proses dan keluaran sistem sesuai dengan yang diharapkan dan direncanakan, serta kebutuhan perangkat lunak dan perangkat kerasnya. Hasil analisis secara umum adalah penterjemahan permasalahan knapsack problem menjadi kode-kode kromosom, hasil akhir program, serta input yang dibutuhkan program.

3.2.1 Kebutuhan masukan

Kebutuhan masukan pada penelitian tugas akhir ini dikelompokkan menjadi dua bagian, yaitu:

1. Kebutuhan masukan untuk basis data pada data barang.

Basis data ini berisi item item barang yang akan dioptimasi, kebutuhan input untuk basis data ini yaitu :

- a. index, merupakan no urut dari barang.
- b. nama barang.
- c. berat, merupakan nilai berat tiap item barang.

- d. Nilai, merupakan keuntungan tiap barang dalam satu satuan mata uang.

2. Kebutuhan masukan untuk variabel algoritma genetika.

Variabel genetika ini sangat menentukan hasil dari optimasi yang dikerjakan oleh program, kebutuhan input untuk variabel genetika ini yaitu :

- a. Populasi, yaitu ukuran populasi tiap generasi.
- b. Generasi, yaitu maksimum generasi pada tiap iterasi.
- c. Probabilitas *crossover*, yaitu menunjukkan banyaknya induk yang terkena *crossover* pada tiap generasi.
- d. Probabilitas mutasi, yaitu menunjukkan banyaknya induk yang terkena mutasi pada tiap generasi.
- e. Jenis *crossover*, yaitu metode yang digunakan pada *crossover*.
- f. Breeder, yaitu menunjukkan besarnya induk terbaik yang di lestarikan pada tiap generasi.

3.2.2 Proses

Proses yang dimaksud adalah pengolahan data masukan baik itu pada basis data maupun optimasi dengan Algoritma genetika.

1. Basis Data

- a. Proses memasukkan data baru.
- b. Proses mengubah data yang sudah ada.
- c. Proses menghapus data yang sudah ada.

2. Algoritma Genetika

- a. Proses pembentukan populasi awal.

- b. Proses menghitung nilai fitness.
- c. Proses seleksi *roda roulette*.
- d. Proses crossover.
- e. Proses Mutasi
- f. Proses melestarikan individu terbaik (*Breeder GA*).
- g. Proses menghitung fitness akhir.

3.2.3 Keluaran

Berikut adalah keluaran dari aplikasi Knapsack Problem menggunakan Algoritma genetika:

1. Basis data
 - a. Daftar item barang.
2. Algoritma Genetika
 - a. Jenis-jenis item barang hasil optimasi
 - b. Grafik nilai fitness pada semua generasi.
 - c. Grafik nilai fitness pada generasi terbaik.
 - d. Report program mengenai fitness rata-rata, fitness terbaik, fitness terburuk, serta berat tiap item individu pada tiap iterasi.
 - e. Catatan tentang individu terbaik yaitu meliputi jumlah barang, generasi ke-, nilai fitness, berat barang, jumlah keuntungan.

3.3 Fungsi-fungsi yang Dibutuhkan

3.3.1 Basis Data

Fungsi-fungsi yang ditangani oleh Basis Data barang adalah :

1. Fungsi memasukkan data baru.
2. Fungsi mengubah data yang sudah ada.

3. Fungsi menghapus data.

3.3.2 Algoritma Genetika

Fungsi-fungsi yang ditangani oleh proses algoritma genetika adalah sebagai berikut:

1. Fungsi menghitung jumlah item barang pada basis data.
2. Fungsi membentuk populasi awal.
3. Fungsi menghitung nilai fitness.
4. Fungsi seleksi roda roulette.
5. Fungsi melestarikan individu terbaik (breeder GA).
6. Fungsi crossover.
7. Fungsi mutasi.
8. Fungsi menghitung fitness akhir
9. Fungsi decode dari kode kode kromosom menjadi suatu bentuk permasalahan yang nyata.
10. Fungsi menggambar grafik fitness.
11. Fungsi untuk menampilkan report program.
12. Fungsi untuk mencatat generasi terbaik.

BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

Untuk memudahkan penggambaran suatu sistem yang ada sistem yang baru yang akan dikembangkan secara logika tanpa memperhatikan lingkungan fisik dimana data tersebut mengalir atau link fisik dimana data tersebut akan disimpan, maka digunakan *Flow chart*.

Flow chart merupakan teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output. *Flow chart* juga digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap tingkat abstraksi.

4.2 Hasil Perancangan

Hasil perancangan adalah tahap pembuatan suatu rancangan perangkat lunak yang terdiri dari : perancangan Diagram alir, perancangan Basis data, perancangan Antar muka, perancangan Operator genetika, perancangan file.

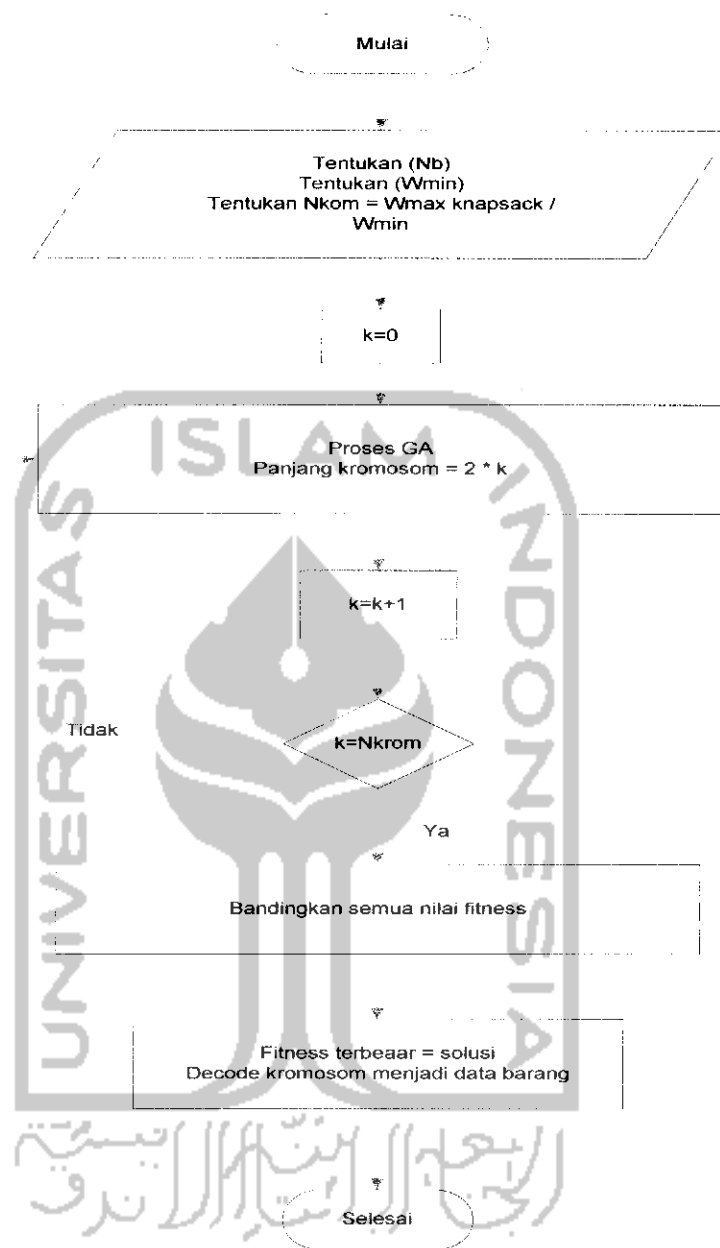
4.2.1 Perancangan flow chart

Perancangan flowchart adalah tahap dimana dibuat suatu diagram alir yang dimulai dari pokok bahasan yang umum sampai ke spesifik yang terdiri dari : Proses utama, Pembangkitan populasi awal, Penghitungan fitness, Seleksi roda rolet, Crossover satu titik, Crossover dua titik, Crossover banyak titik. Crossover seragam.

4.2.1.1 Proses utama

Proses utama adalah proses yang menggambarkan jalannya algoritma suatu program secara umum, flow chart untuk proses utama dapat dilihat pada **Gambar 4.1**, langkah langkahnya adalah sebagai berikut :

1. Tentukan jumlah barang pada tabel barang (N_b).
2. Tentukan berat terkecil dari tabel barang (W_{min}).
3. Tentukan banyak kombinasi $N_{kom} = W_{max} \text{ knapsack} / W_{min}$.
4. Isi k dengan 0 ($k=0$).
5. Lakukan optimasi GA dengan panjang kromosom = $2*k$.
6. $k=k+1$.
7. Lakukan langkah ke-5 sampai $k=N_{kom}$.
8. Bandingkan nilai fitness individu-individu terbaik pada optimasi GA pada semua kombinasi barang k .
9. Tentukan fitness terbesar menjadi solusi.
10. Decode kromosom menjadi data barang.



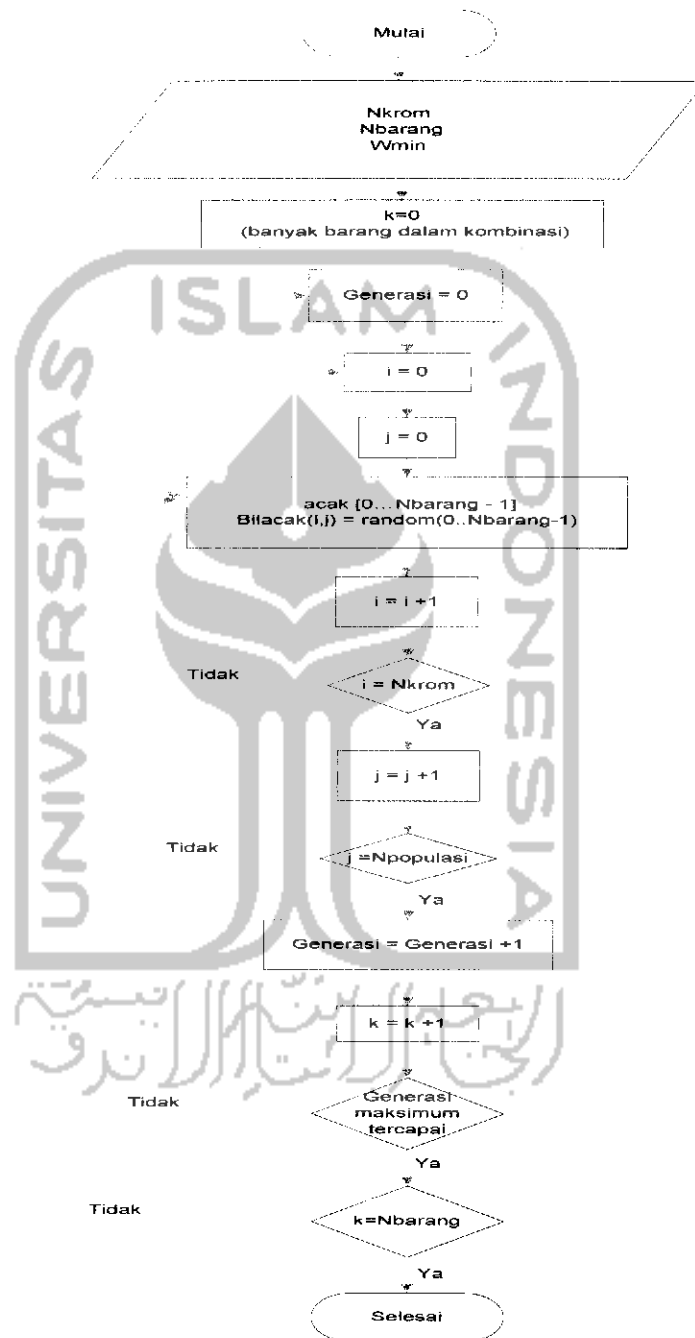
Gambar 4.1 Flowchart proses utama

4.2.1.2 Pembangkitan populasi awal

Pembangkitan populasi awal berfungsi untuk mendapatkan individu-individu terbaik, flow chart untuk pembangkitan populasi awal dapat dilihat pada **Gambar 4.2**, langkah langkahnya adalah sebagai berikut :

1. Tentukan panjang kromosom (N_{krom}).
2. Tentukan jumlah barang (N_{barang}).
3. Tentukan jumlah barang terkecil (W_{min}).
4. $k=0$.
5. Generasi=0.
6. $i = 0$.
7. $j = 0$.
8. Bangkitkan bilangan acak $[0 \dots N_{barang} - 1]$.
9. $i = i+1$.
10. Lakukan langkah ke-8 sampai $i = N_{krom}$.
11. $j = j+1$.
12. Lakukan langkah ke-8 sampai $j = N_{pop}$.
13. Generasi = Generasi +1
14. $k = k+1$.
15. Lakukan langkah ke-5 sampai $k=N_{barang}$.

15. Lakukan langkah ke-5 sampai $k=N\text{barang}$.

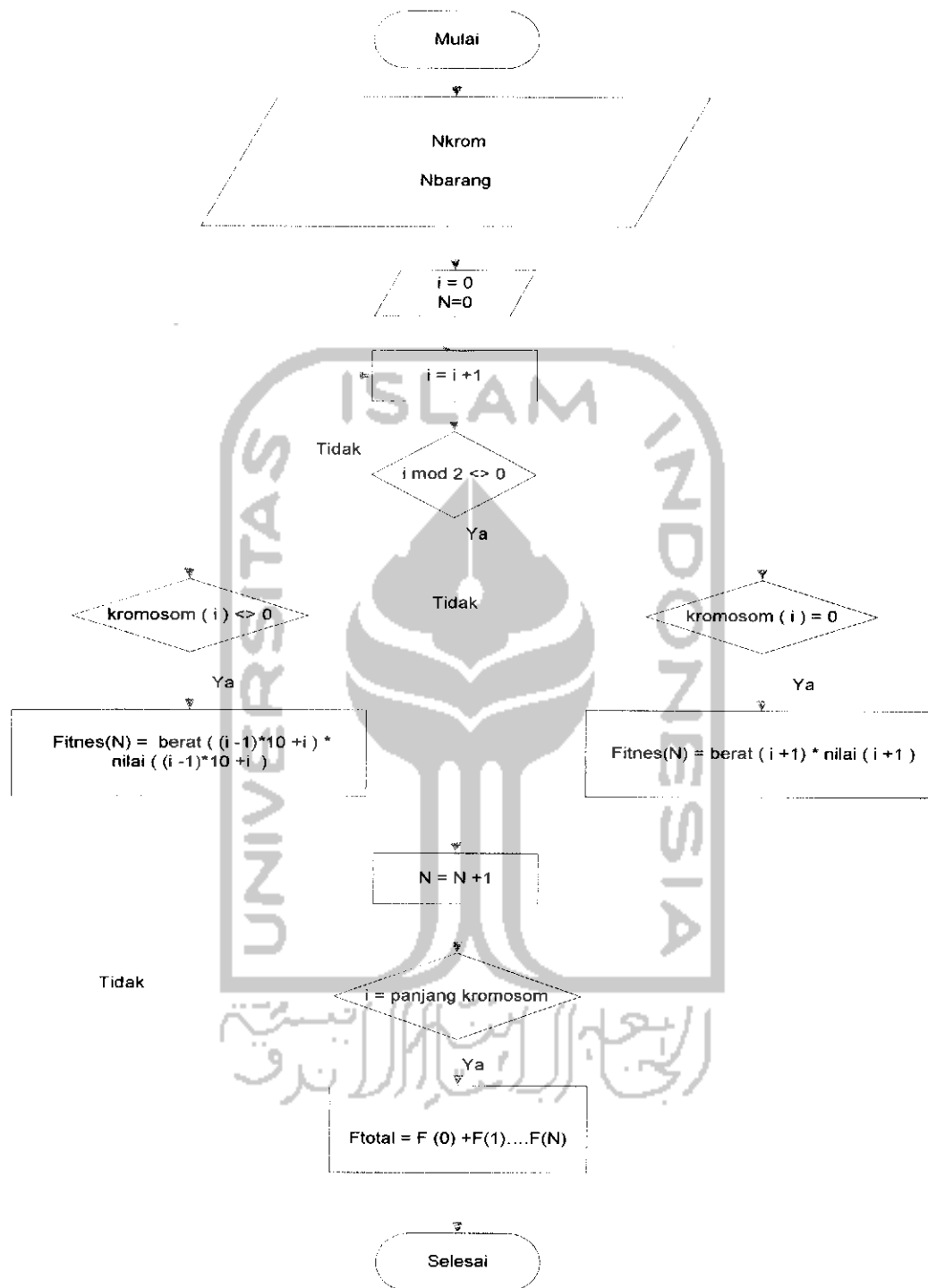


Gambar 4.2 Flowchart populasi

4.2.1.3 Penghitungan fitness

Penghitungan fitness digunakan untuk menghitung bobot tiap individu pada suatu populasi yang menggambarkan kemampuan bertahan hidup individu tersebut, flow chart untuk penghitungan fitness dapat dilihat pada **Gambar 4.3**, langkah-langkahnya adalah sebagai berikut :

1. Menentukan panjang kromosom (N_{krom}).
2. Menentukan jumlah barang (N_{barang}).
3. $I=0$.
4. $N=0$.
5. $I = i+1$.
6. Jika ($I \bmod 2 \neq 0$) lakukan langkah 7, jika tidak lakukan langkah 5.
7. Jika kromosom (I) =0 maka lakukan langkah 8, jika tidak lakukan langkah 9 .
8. $Fitnes(N) = berat (i+1) * nilai (i+1)$.
9. $Fitnes(N) = berat ((i-1)*10+i) * nilai ((i-1)*10+i)$.
10. $N=N+1$.
11. Lakukan langkah 5 sampai $I = N_{krom}$.
12. Hitung fitness total pada tiap individu, $F_{total} = F(0) + F(1) \dots F(N)$



Gambar 4.3 Flowchart penghitungan fitness

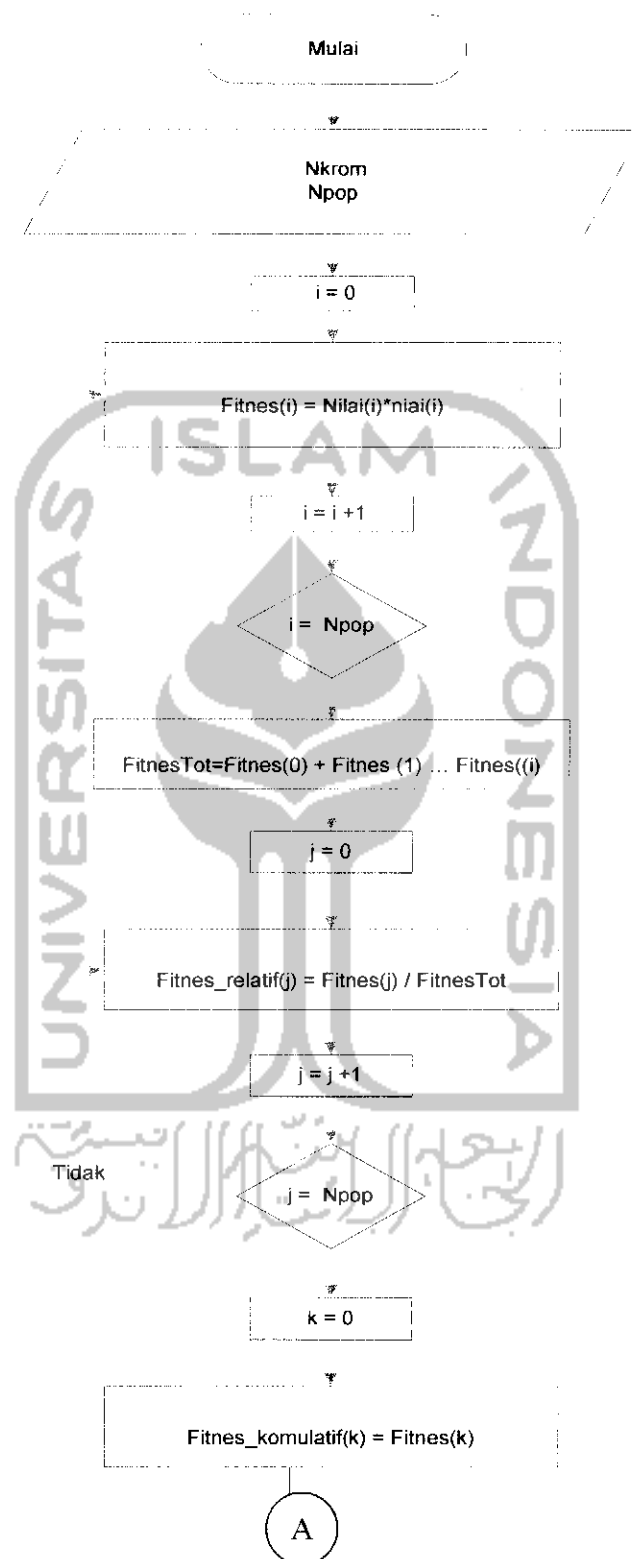
4.2.1.4 Seleksi Roda rolet

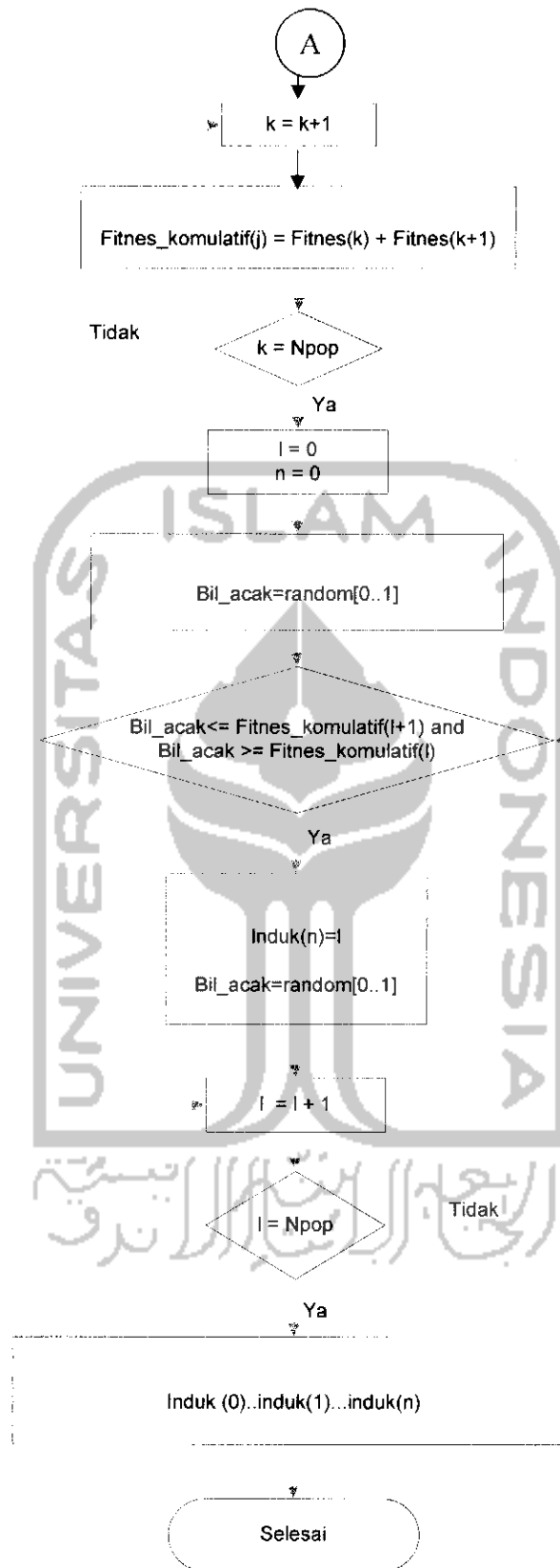
Seleksi Roda rolet digunakan untuk menseleksi individu mana saja yang akan menjadi individu terbaik, flow chart untuk seleksi Roda rolet dapat dilihat pada **Gambar 4.4**, langkah langkahnya adalah sebagai berikut :

1. Tentukan panjang kromosom (N_{krom}).
2. Tentukan jumlah populasi (N_{pop}).
3. $I = 0$.
4. Hitung fitness tiap individu, $Fitness(i) = Nilai(i) * niai(i)$.
5. $I = i + 1$.
6. Lakukan langkah 4 sampai N_{pop} .
7. Hitung fitness total, $Fitness_{Tot} = Fitness(0) + Fitness(1) \dots Fitness(i)$.
8. $j = 0$.
9. Hitung fitness relatif tiap individu, $Fitness_{relatif}(j) = Fitness(j) / Fitness_{Tot}$.
10. $j = j + 1$.
11. Lakukan langkah 9 sampai $j = N = pop$.
12. $k = 0$.
13. Hitung fitness kumulatif tiap individu, $Fitness_{kumulatif}(k) = Fitness(k)$.
14. $k = k + 1$.
15. Hitung fitness kumulatif tiap individu, $Fitness_{kumulatif}(j) = Fitness(k) + Fitness(k+1)$.
16. Lakukan langkah 14 sampai $k = N_{pop}$.
17. $I = 0$.
18. $N = 0$.

19. Bangkitkan bilangan acak, $Bil_acak = \text{random}[0..1]$.
20. Jika $Bil_acak \leq \text{Fitnes_komulatif}(l+1)$ and $Bil_acak \geq \text{Fitnes_komulatif}(l)$ lakukan langkah 21, jika tidak lakukan langkah 23.
21. Catat induk yg terpilih, $\text{Induk}(n)=l$
22. Bangkitkan bilangan acak, $Bil_acak = \text{random}[0..1]$.
23. $l = l+1$.
24. Lakukan langkah 20 sampai $l = Npop$.
25. Catat n induk yg terpilih untuk operasi selanjutnya, Induk (0)..induk(1)...induk(n).







Gambar 4.4 Seleksi roda

4.2.1.5 Crossover

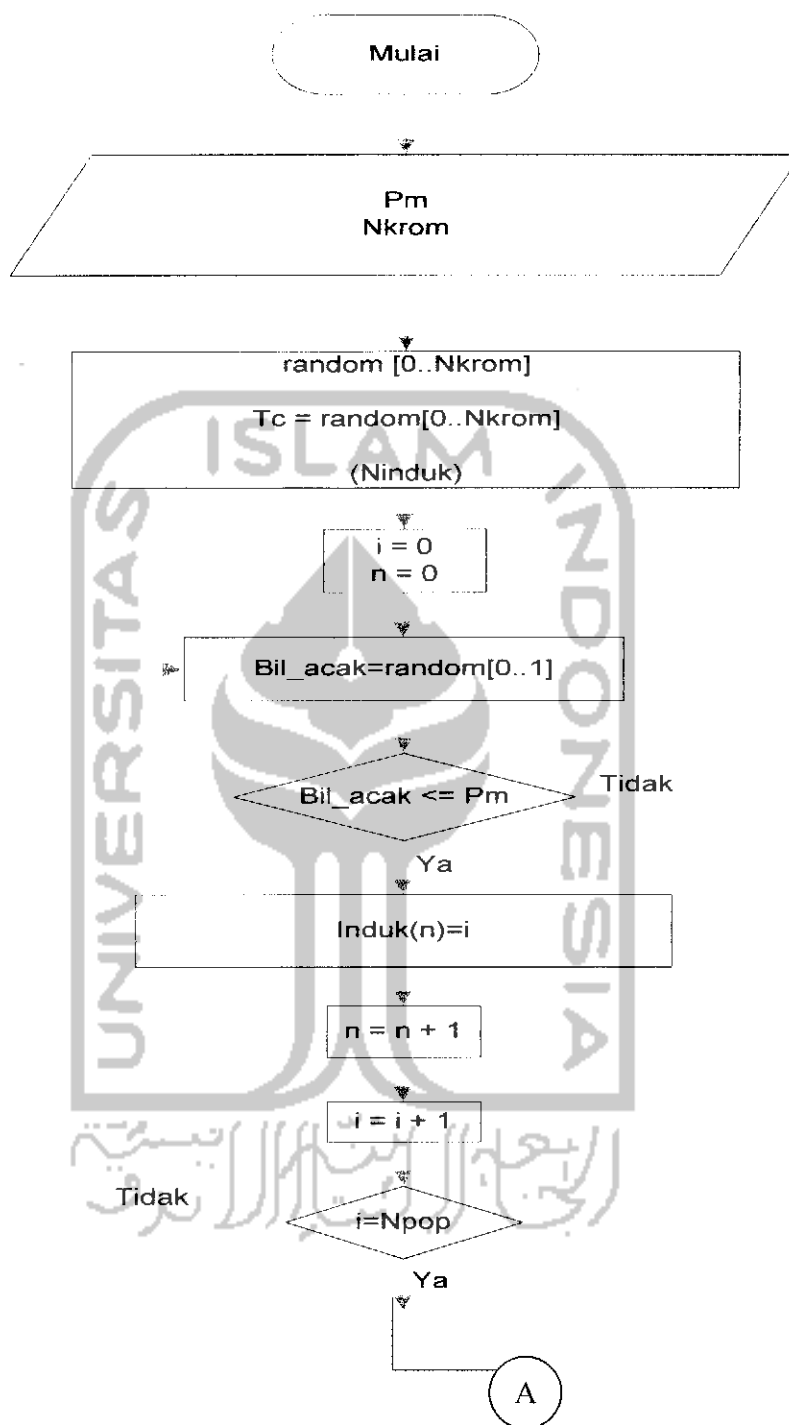
Crossover adalah proses persilangan antara 2 individu sehingga menghasilkan individu-individu yang baru yang mirip dengan induknya yang diharapkan mempunyai kemampuan bertahan hidup yang lebih tinggi, Crossover yang digunakan pada penelitian ini terdiri 4 macam : Crossover satu titik, Crossover dua titik, Crossover banya titik, Crossover seragam.

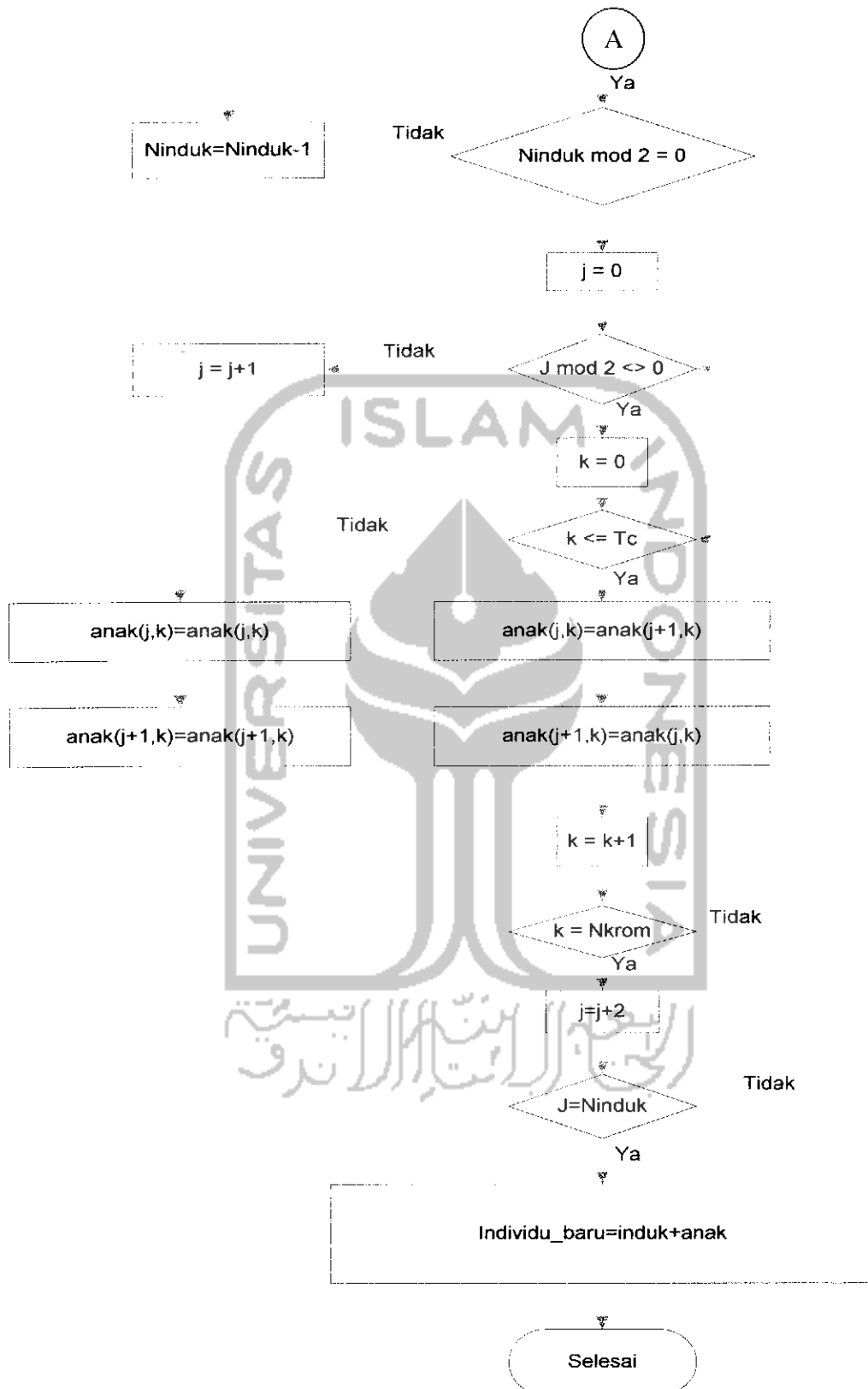
1. Crossover satu titik (*one point crossover*)

Crossover satu titik dilakukan dengan dengan melakukan persilangan antara induk dengan menggunakan satu titik acuan. Flow chart crossover satu titik dapat dilihat pada **Gambar 4.5** ,Berikut adalah langkah-langkah untuk proses persilangan satu titik (*one point crossover*):

1. Tentukan Probabilitas crossover(P_m).
2. Tentukan panjang kromosom(N_{krom}).
3. Tentukan titik crossover, $T_c = \text{random}[0..N_{krom}]$.
4. Tentukan jumlah induk yg tercrossover (N_{induk}).
5. $I = 0$.
6. $n = 0$.
7. Bangkitkan bil acak, $Bil_acak = \text{random}[0..1]$.
8. Jika $Bil_acak \leq P_m$ lakukan langkah 9, jika tidak lakukan langkah 11.
9. Catat induk yg terkena crossover $Induk(n)=i$.
10. $n = n+1$.
11. $I = i+1$.
12. Lakukan langkah 8 sampai $i = N_{pop}$.

13. jika $(N_{\text{induk}} \bmod 2 = 0)$ maka lakukan langkah 14, jika tidak lakukan langkah 15.
14. $N_{\text{induk}} = N_{\text{induk}} + 1$.
15. $j = 0$.
17. Jika $(N_{\text{induk}} \bmod 2 \neq 0)$ lakukan langkah 19, jika tidak lakukan langkah 18.
18. $j = j + 1$
19. $k = 0$.
20. Jika $k \leq T_c$ maka lakukan langkah 23, Jika tidak lakukan langkah 21.
21. Tukar induk 1 dengan induk 2, $\text{anak}(j, k) = \text{anak}(j, k)$.
22. Tukar induk 1 dengan induk 2, $\text{anak}(j+1, k) = \text{anak}(j+1, k)$, lanjutkan k langkah 25.
23. Tukar induk 1 dengan induk 2, $\text{anak}(j, k) = \text{anak}(j+1, k)$.
24. Tukar induk 1 dengan induk 2, $\text{anak}(j+1, k) = \text{anak}(j, k)$.
25. $k = k + 1$.
26. Lakukan langkah 20 sampai $k = N_{\text{krom}}$.
27. $j = j + 2$.
28. Lakukan langkah 17 sampai $j = N_{\text{induk}}$.
29. Bangkitkan individu baru sejumlah N_{pop} , $\text{Individu_baru} = \text{induk} + \text{anak}$.





Gambar 4.5 Crossover satu titik

2. Crossover banyak titik (*multi point crossover*)

Crossover banyak titik dilakukan dengan dengan melakukan persilangan antara induk dengan menggunakan banyak titik acuan yang dibangkitkan secara acak. Flow chart crossover banyak titik dapat dilihat pada **Gambar 4.6** ,Berikut adalah langkah-langkah untuk proses persilangan banyak titik (*multi point crossover*):

1. Tentukan Probabilitas crossover(P_m).
2. Tentukan panjang kromosom(N_{krom}).
3. Tentukan jumlah titik crossover (N_{Tc}), $N_{Tc} = \text{random}[0..N_{krom}]$
4. Tentukan titik crossover sebanyak N_{Tc} , $T_c = \text{random}[0..N_{krom}]$
5. Tentukan jumlah induk yg tercrossover(N_{induk})
6. $I = 0$.
7. $n = 0$.
8. Bangkitkan bil acak, $Bil_acak = \text{random}[0..1]$.
9. Jika $Bil_acak \leq P_m$ lakukan langkah 10, jika tidak lakukan langkah 11.
10. Catat induk yg terkena crossover $Induk(n)=i$.
11. $n = n+1$.
12. $I = i+1$.
13. Lakukan langkah 8 sampai $i = N_{pop}$.
14. jika ($N_{induk} \text{ mod } = 0$) maka lakukan langkah 15, jika tidak lakukan langkah 16.
15. $N_{induk} = N_{induk} + 1$.
16. $j=0$.

17. Jika $(N_{\text{induk}} \bmod 2 \neq 0)$ lakukan langkah 19, jika tidak lakukan langkah 18.

18. $j=j+1$

19. $k=0$.

20. $l=0$.

20. Jika $k \leq T_c(l)$ maka lakukan langkah 23, Jika tidak lakukan langkah 21.

21. Tukar induk 1 dengan induk 2, $\text{anak}(j,k)=\text{anak}(j,k)$.

22. Tukar induk 1 dengan induk 2, $\text{anak}(j+1,k)=\text{anak}(j+1,k)$, lanjutkan k langkah 25.

23. Tukar induk 1 dengan induk 2, $\text{anak}(j,k)=\text{anak}(j+1,k)$.

24. Tukar induk 1 dengan induk 2, $\text{anak}(j+1,k)=\text{anak}(j,k)$.

25. $k=k+1$.

26. Lakukan langkah 20 sampai $k=T_c(l)$.

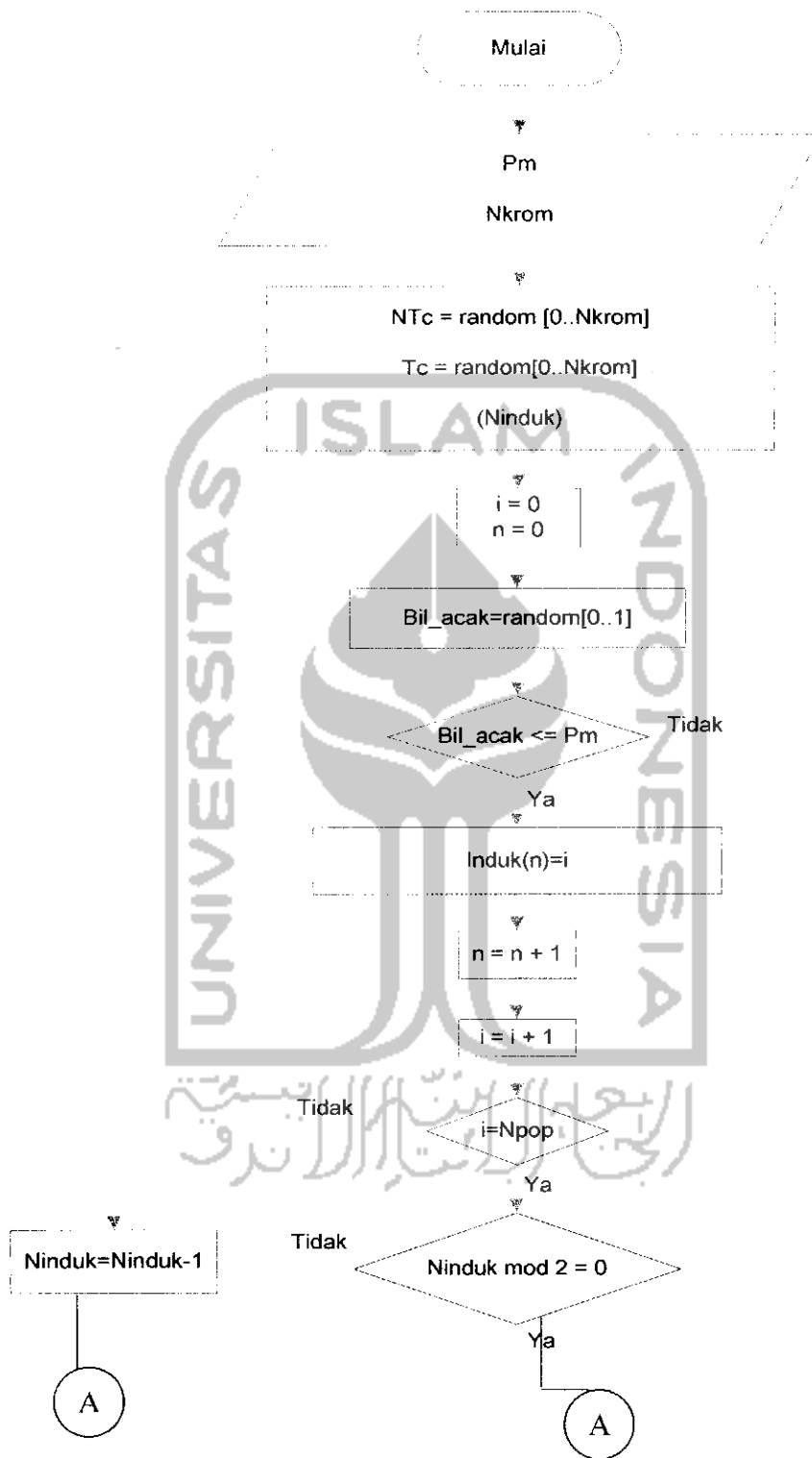
27. $l=l+1$.

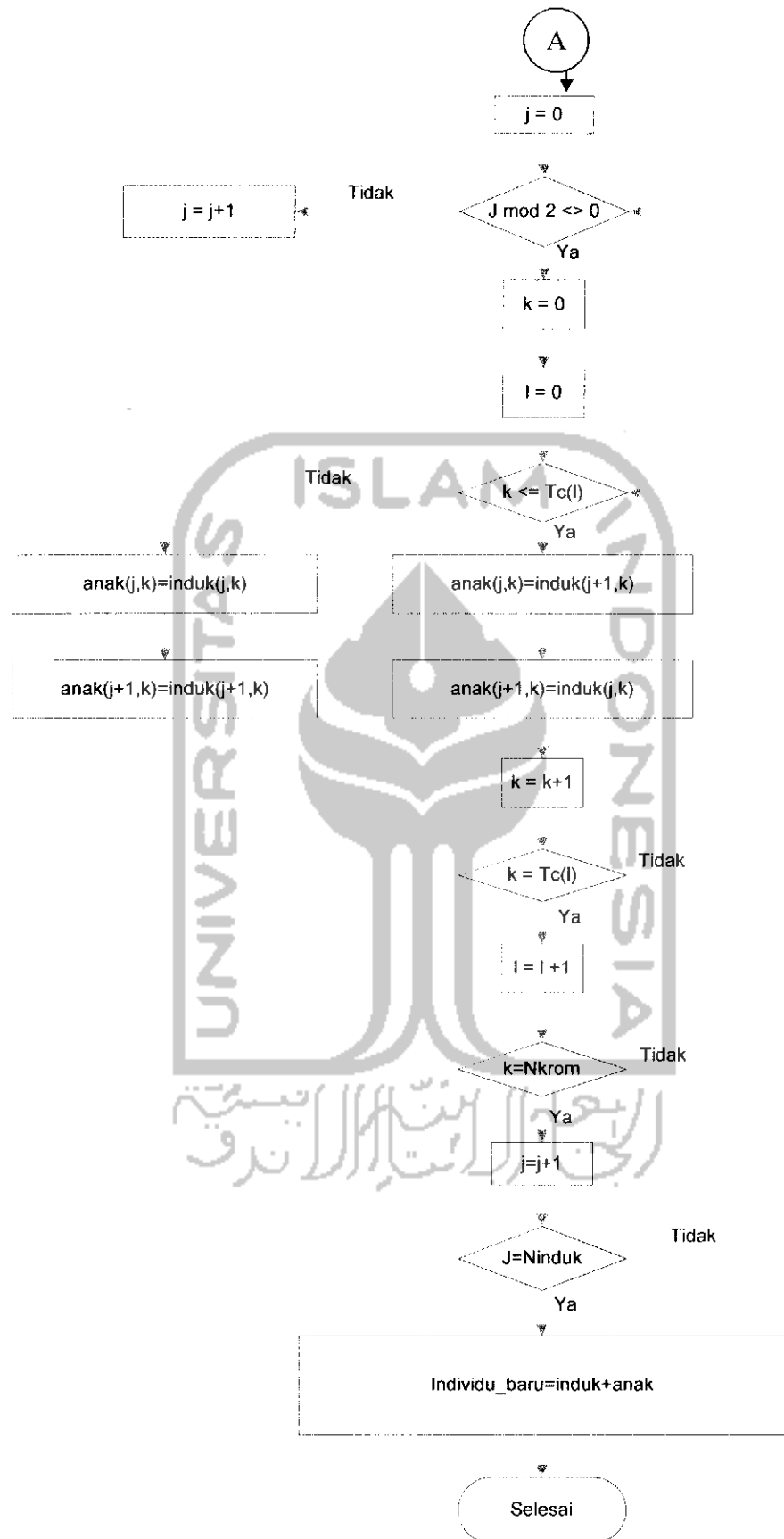
28. Lakukan langkah 26 sampai $k=N_{\text{krom}}$.

28. $j=j+1$.

29. Lakukan langkah 17 sampai $j=N_{\text{induk}}$.

30. Bangkitkan individu baru sejumlah N_{pop} , $\text{Individu_baru} = \text{induk} + \text{anak}$.





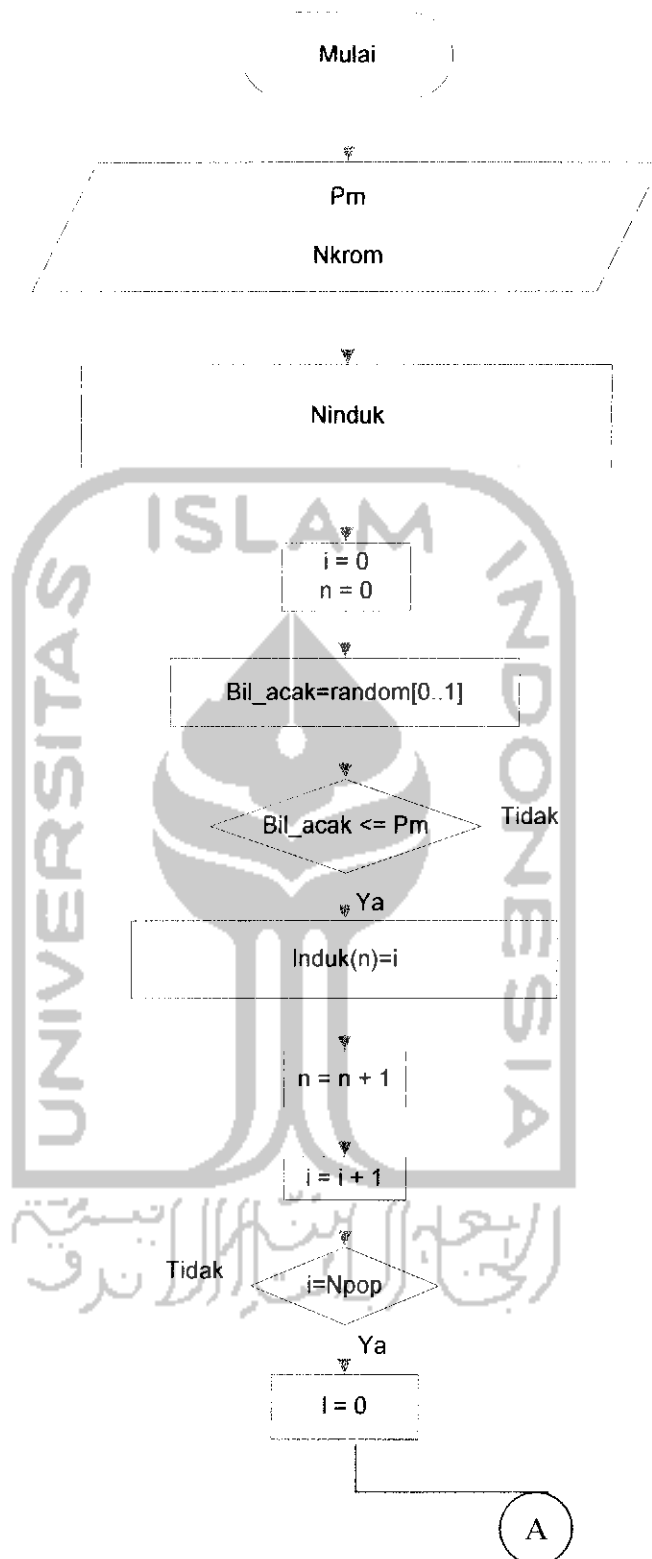
Gambar 4.6 Crossover banyak titi

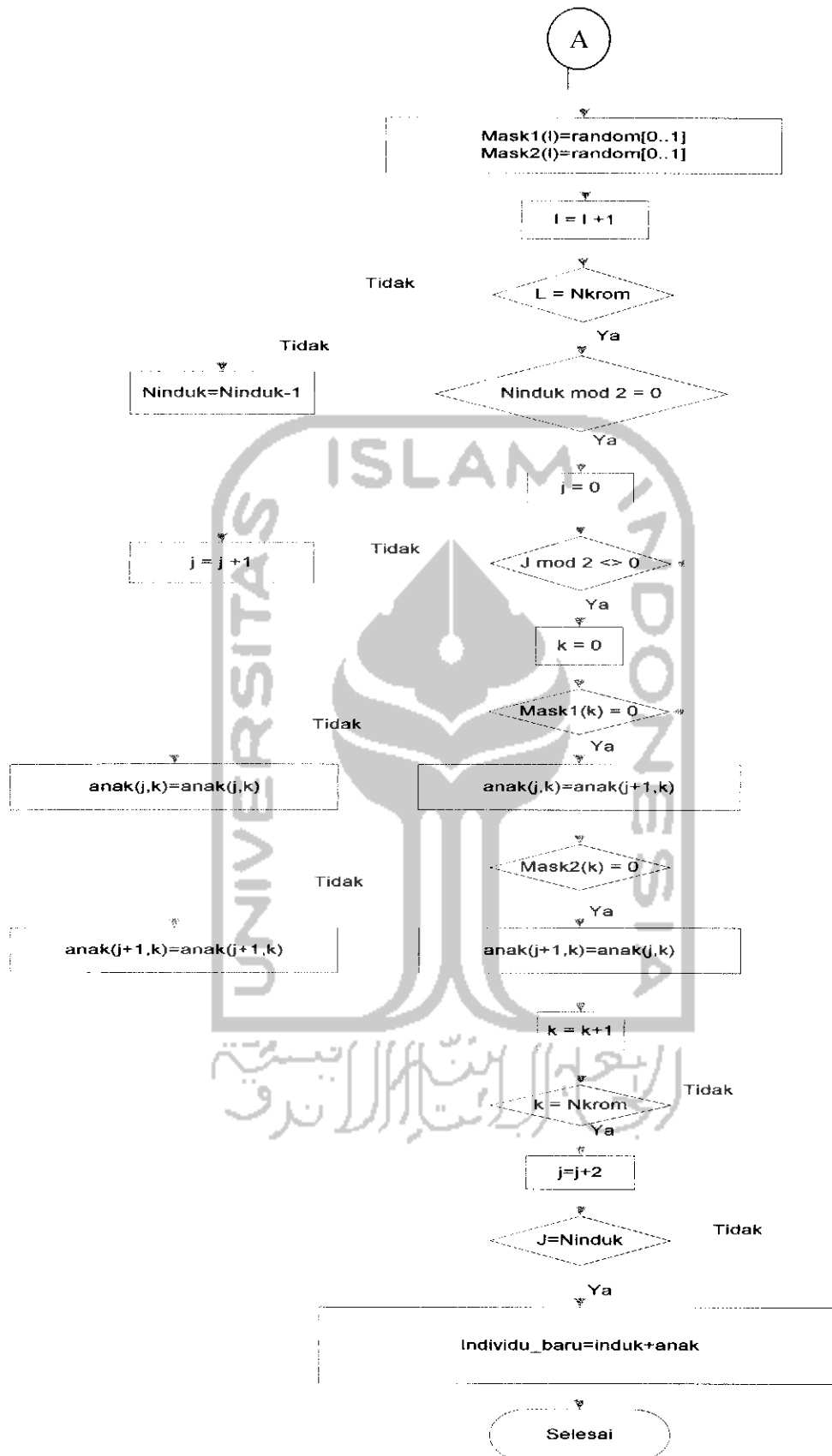
3. Crossover seragam (*uniform crossover*)

Crossover seragam dilakukan dengan dengan melakukan persilangan antara induk dengan menggunakan banyak titik acuan yang didasarkan pada variable mask1 dan mask2. Flow chart crossover seragam dapat dilihat pada **Gambar 4.7** ,Berikut adalah langkah-langkah untuk proses persilangan seragam (*uniform crossover*):

1. Tentukan Probabilitas crossover(P_m).
2. Tentukan panjang kromosom(N_{krom}).
3. Tentukan jumlah induk yg tercrossover (N_{induk}).
4. $I = 0$.
5. $n = 0$.
6. Bangkitkan bil acak, $Bil_acak = random[0..1]$.
7. Jika $Bil_acak \leq P_m$ lakukan langkah 8, jika tidak lakukan langkah 10.
8. Catat induk yg terkena crossover $Induk(n)=i$.
9. $n = n+1$.
10. $I = i+1$.
12. Lakukan langkah 8 sampai $I = N_{pop}$.
13. $l=0$.
14. Bangkitkan bil acak $Mask1$, $Mask2$, $Mask1(l)=random[0..1]$, $Mask2(l)=random[0..1]$.
15. $l=l+1$.
16. Lakukan langkah 14 sampai $l=N_{krom}$.
17. jika $(N_{induk} \bmod =0)$ maka lakukan langkah 14, jika tidak lakukan langkah 15.
18. $N_{induk}=N_{induk}+1$.

19. $j=0$.
20. Jika $(N_{\text{induk}} \bmod 2 \neq 0)$ lakukan langkah 19, jika tidak lakukan langkah 18.
21. $j=j+1$
22. $k=0$.
23. jika $\text{Mask1}(k) = 0$ maka lakukan langkah 25, jika tidak lakukan langkah 24.
24. Anak1 berasal dari induk2, $\text{anak}(j,k)=\text{anak}(j,k)$.
25. Anak1 berasal dari induk1, $\text{anak}(j+1,k)=\text{anak}(j+1,k)$.
26. jika $\text{Mask2}(k) = 0$ maka lakukan langkah 27 jika tidak lakukan langkah 28.
27. Anak2 berasal dari induk1, $\text{anak}(j,k)=\text{anak}(j,k)$.
28. Anak2 berasal dari induk2, $\text{anak}(j+1,k)=\text{anak}(j+1,k)$, lanjutkan k langkah 33.
29. $k=k+1$.
30. Lakukan langkah 23 sampai $k=N_{\text{krom}}$.
31. $j=j+2$.
32. Lakukan langkah 17 sampai $j=N_{\text{induk}}$.
33. Bangkitkan individu baru sejumlah N_{pop} , $\text{Individu_baru} = \text{induk} + \text{anak}$.





Gambar 4.7 Crossover se

3. Crossover dua titik (*two point crossover*)

Crossover dua titik dilakukan dengan dengan melakukan persilangan antara induk dengan menggunakan 2 titik acuan. Flow chart crossover dua titik dapat dilihat pada **Gambar 4.8** ,Berikut adalah langkah-langkah untuk proses persilangan dua titik (*two point crossover*):

1. Tentukan Probabilitas crossover (P_m).
2. Tentukan panjang kromosom (N_{krom}).
3. Tentukan titik crossover, $Tc1 = \text{random}[0..N_{krom}]$, $Tc2 = \text{random}[0..N_{krom}]$.
4. Tentukan jumlah induk yg tercrossover (N_{induk}).
5. $I = 0$.
6. $n = 0$.
7. Bangkitkan bil acak, $Bil_acak = \text{random}[0..1]$.
8. Jika $Bil_acak \leq P_m$ lakukan langkah 9, jika tidak lakukan langkah 11.
9. Catat induk yg terkena crossover $Induk(n)=i$.
10. $n = n+1$.
11. $I = i+1$.
12. Lakukan langkah 8 sampai $i = N_{pop}$.
13. jika $(N_{induk} \bmod 2 = 0)$ maka lakukan langkah 14, jika tidak lakukan langkah 15.
14. $N_{induk} = N_{induk} + 1$.
15. $j = 0$.
17. Jika $(N_{induk} \bmod 2 \neq 0)$ lakukan langkah 19, jika tidak lakukan langkah 18.

18. $j=j+1$

19. $k=0$.

20. Jika $k \leq Tc1$ maka lakukan langkah 24, Jika tidak lakukan langkah 23.

21. Tukar induk 1 dengan induk 2, $anak(j,k)=anak(j,k)$.

22. Tukar induk 1 dengan induk 2, $anak(j+1,k)=anak(j+1,k)$, lanjutkan k langkah 26.

23. Jika $k \leq Tc2$ maka lakukan langkah 24, Jika tidak lakukan langkah 21.

24. Tukar induk 1 dengan induk 2, $anak(j,k)=anak(j+1,k)$.

25. Tukar induk 1 dengan induk 2, $anak(j+1,k)=anak(j,k)$.

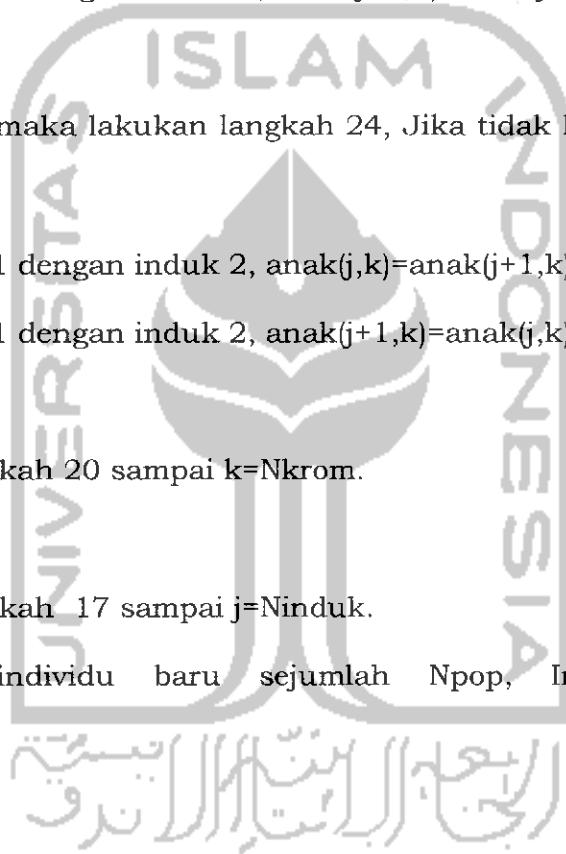
26. $k=k+1$.

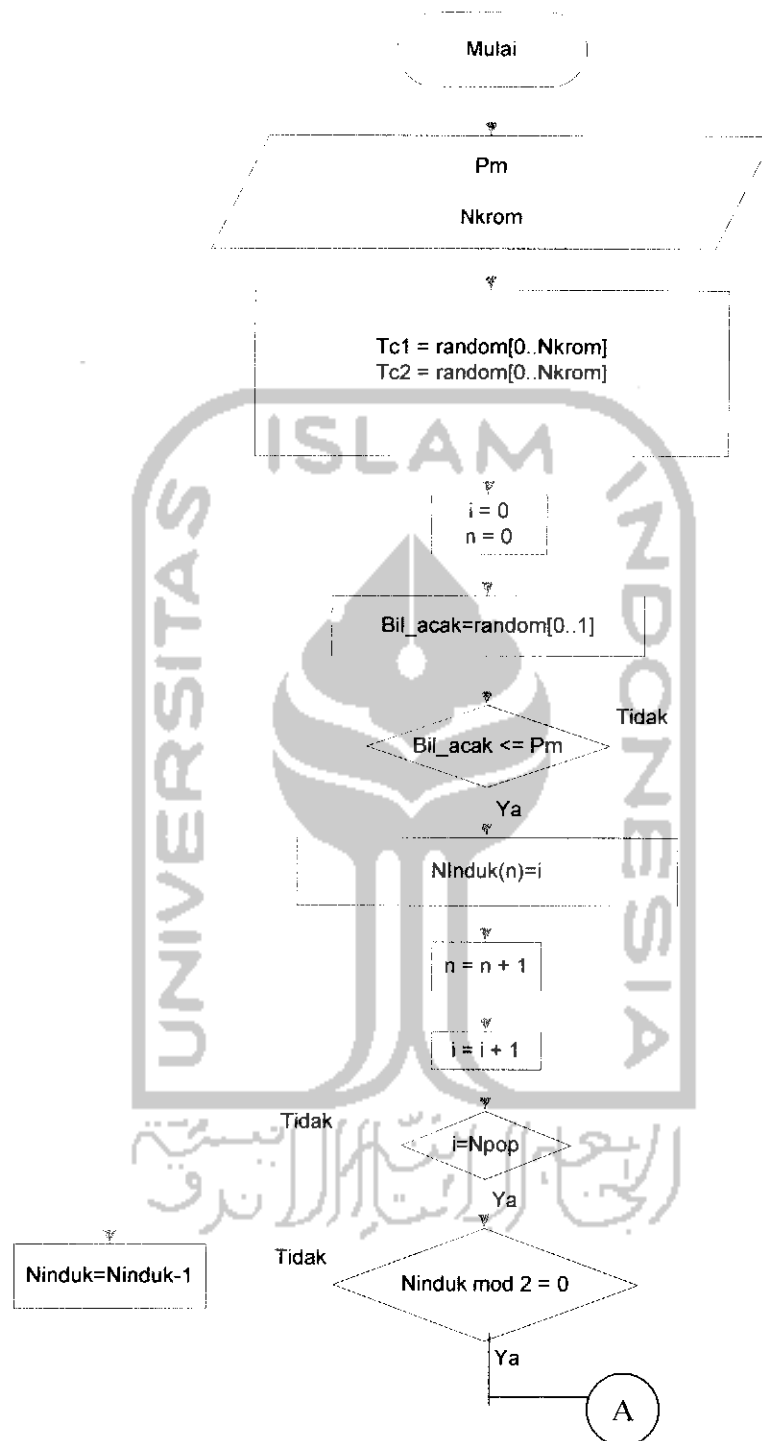
27. Lakukan langkah 20 sampai $k=Nkrom$.

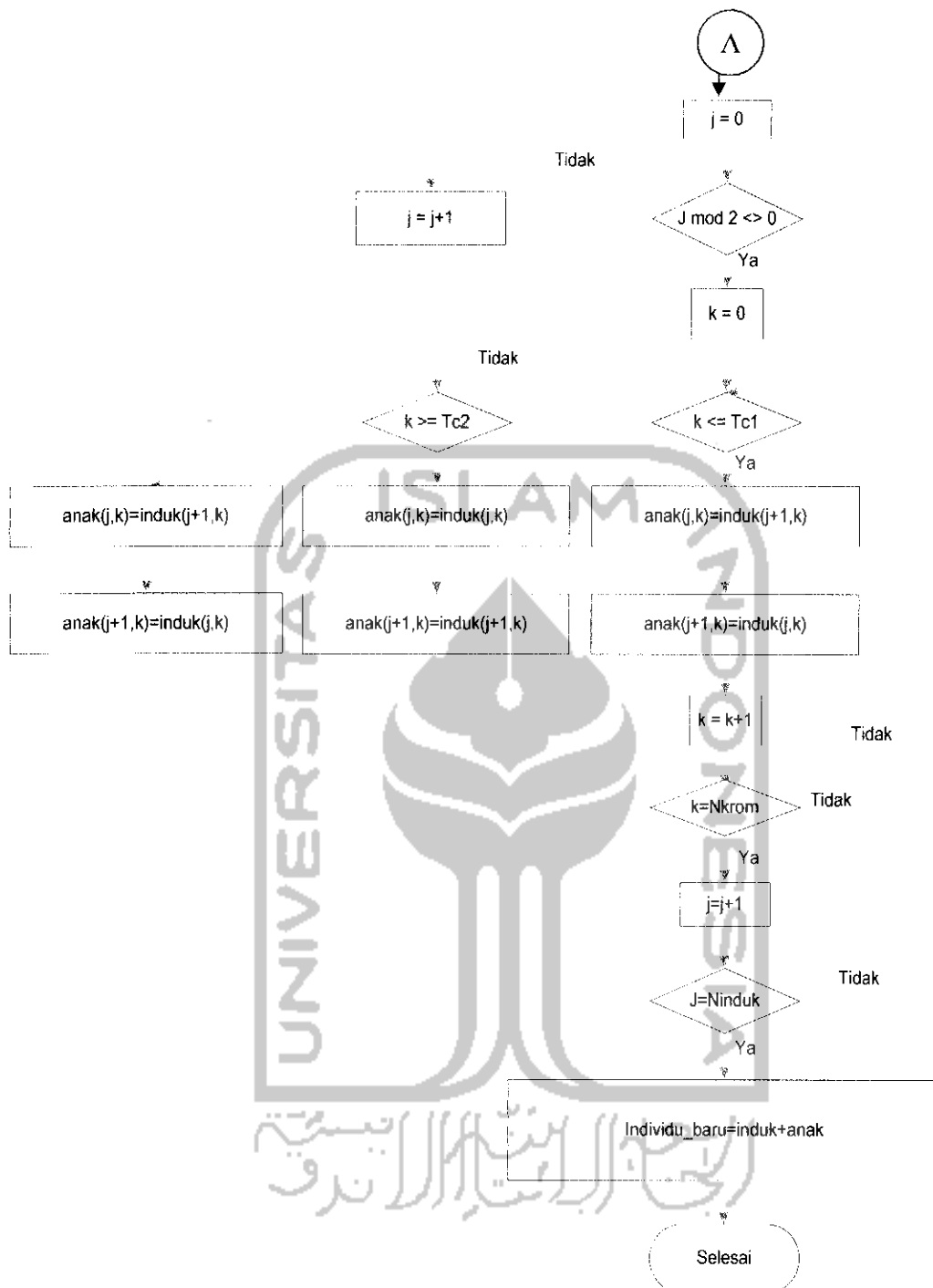
28. $j=j+2$.

29. Lakukan langkah 17 sampai $j=Ninduk$.

30. Bangkitkan individu baru sejumlah $Npop$, Individu_baru = induk+anak.







Gambar 4.8 Crossover 2 titik

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi

Aplikasi algoritma genetika dalam knapsack problem ini dikembangkan dengan perangkat lunak Visual Basic.Net 2003 keluaran Microsoft yang berjalan pada sistem operasi Microsoft Windows XP Professional Edition Service Pack 1, sedangkan aplikasi *database* yang digunakan adalah Microsoft Access 2003.

Perangkat keras yang digunakan dalam pengembangan aplikasi penelitian tugas akhir ini adalah komputer desktop standar dengan spesifikasi hardware sebagai berikut:

- a. Processor intel Pentium 4 1,8 Ghz
- b. 80GB hard drive
- c. 15" CRT monitor
- d. 128 Mb Gforce Fx 5200 video memory
- e. DVD CDR Drive

5.2 Implementasi Perangkat Lunak

Adalah proses membangun suatu perangkat lunak dengan menggunakan perangkat lunak lain mulai dari pembuatan antarmuka perangkat lunak sampai proses akhir.

5.2.1 Perancangan antarmuka perangkat lunak

Adalah proses pembuatan antarmuka suatu perangkat lunak sehingga *user* dapat berinteraksi dengan perangkat lunak tersebut

4.2.2 Perancangan representasi kromosom

Representasi kromosom dalam penelitian ini menggunakan tipe integer dimana kromosom berisi angka-angka integer yang melambangkan kode index barang pada table barang, tiap kromosom terdiri dari n bit dengan n bilangan genap, kemudian tiap kromosom dibagi menjadi 2 bit yang melambangkan kode index pada table barang. Panjang kromosom akan berganti-ganti sesuai dengan jumlah kombinasi barang dalam knapsack.

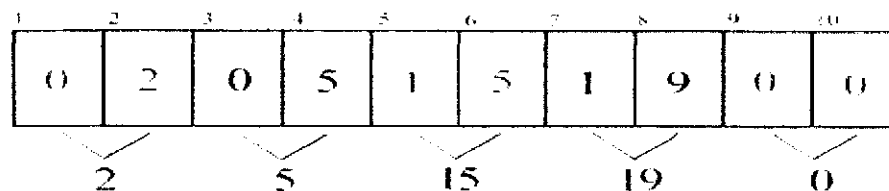
1.) Misal diketahui

- Jumlah barang dalam tabel barang = 20 barang
- Berat terkecil dari 20 barang adalah 20 kg
- Berat maksimum yg bisa diangkut knapsack 100 kg

Jika misalnya kombinasi barang dalam knapsack yg terpilih adalah 5 barang, maka panjang kromosom = $5 * 2 = 10$, dan akan dilakukan proses GA sebanyak 5 kali dengan rumus :

$$\begin{aligned} n_{GA} &= W_{\max} / \text{berat terkecil} \\ &= 100/20 = 5 \end{aligned}$$

Representasi kromosomnya adalah seperti pada **Gambar 4.9**



kromosom [0205151900]

kombinasi barang [2,5,15,19,0]

Gambar 4.9 Representasi kromosom

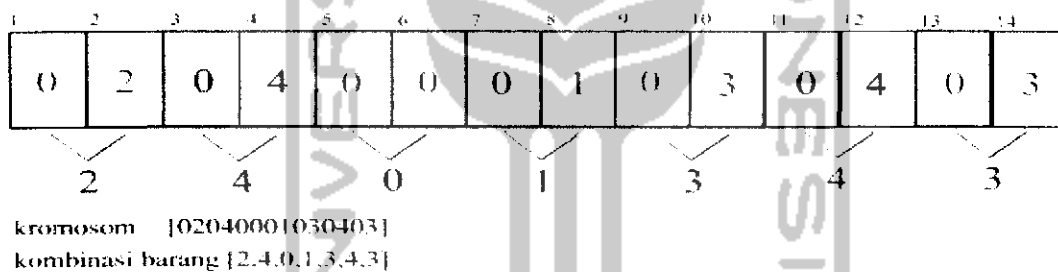
2.) Misal diketahui

- Jumlah barang dalam tabel barang = 7 barang
- Berat terkecil dari 7 barang adalah 10 kg
- Berat maksimum yg bisa diangkut knapsack 100 kg
- Jika misalnya kombinasi barang dalam knapsack yg

terpilih adalah 7 barang, maka panjang kromosom = $7 * 2 = 14$, dan akan dilakukan proses GA sebanyak 10 kali dengan rumus :

$$\begin{aligned} nGA &= W_{\max} / \text{berat terkecil} \\ &= 100 / 10 = 10 \end{aligned}$$

Representasi kromosomnya adalah seperti pada **Gambar 4.10**



Gambar 4.10 Representasi kromosom

4.2.3 Penentuan nilai fitness

Dalam aplikasi ini penghitungan nilai fitness secara umum menggunakan rumus :

$$F = \sum (\text{berat} * \text{nilai})$$

Misal terdapat kromosom sebagai berikut :

02040001030403

Dengan daftar barang seperti pada **Tabel 4.1**.

Tabel 4.1 Daftar barang

No	Nama barang	Berat(kg)	Nilai/Keuntungan(Rp)
0.	Barang 1	10	20
1.	Barang 2	11	21
2.	Barang 3	12	22
3.	Barang 4	13	23
4.	Barang 5	14	24

Perhitungan nilai fitnessnya adalah sebagai berikut :

$$\text{Rumus } F = \sum (\text{berat} \cdot \text{nilai})$$

$$\text{Kromosom} = 02-04-00-01-03-04-03$$

Fitnes masing masing gen :

$$02 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 02 = 11 \cdot 21 = 231$$

$$04 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 04 = 13 \cdot 23 = 299$$

$$00 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 00 = 10 \cdot 20 = 200$$

$$01 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 01 = 11 \cdot 21 = 231$$

$$03 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 03 = 13 \cdot 23 = 299$$

$$04 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 04 = 14 \cdot 24 = 336$$

$$03 = (\text{berat} \cdot \text{nilai}) \text{ pada barang ke } 03 = 13 \cdot 23 = 299$$

Fitnes tiap individu :

$$F_{\text{individu}} = F(00) + F(04) + F(00) + F(01) + F(03) + F(04) + F(03)$$

$$= 231 + 299 + 200 + 231 + 299 + 336 + 229$$

$$= 1685$$

4.2.4 Struktur data

Struktur data pada aplikasi perangkat lunak ini meliputi struktur data pada kromosom, kromosom yang dibuat berupa array n data integer dimana n tergantung dari jumlah barang pada basis data barang, struktur data pada kromosom adalah sebagai berikut :

Kromosom = Array[0..Nbarang] of integer

4.2.5 Perancangan basis data

4.2.5.1 Struktur tabel basis data

Berikut merupakan penjelasan mengenai struktur data tabel *basis data* yang digunakan dalam penelitian tugas akhir ini.

1. Tabel db

Tabel ini digunakan untuk menyimpan data barang yang akan dioptimasi. **Tabel 4.2** merupakan struktur dari tabel db.

Tabel 4.2

Fields	Tipe Data	Not Null	Key	Keterangan
index	Number	Yes	Primary Key	index nomor barang
nama	Text	no		Nama barang
Berat	Number	no		Berat barang
Nilai	Number	no		Keuntungan barang

4.2.6 Perancangan antarmuka

4.2.6.1 Perancangan antarmuka menu

Program aplikasi algoritma genetika dalam knapsack problem ini secara umum terdiri 3 menu utama

1. Menu File

Menu file ini terdiri 2 sub menu item yaitu

a. Menu Exit

Menu Exit ini berfungsi untuk keluar dari program

b. Menu Reset

Menu Reset ini berfungsi untuk mereset program dari keadaan ter optimasi kedalam keadaan awal sebelum dilakukan optimasi.

2. Menu Help

Menu Help ini berfungsi untuk menampilkan file help dalam format .chm yang akan membantu user dalam memakai software ini secara lengkap disertai pula dengan gambar dan penjelasan.

3. Menu About

Menu about ini berfungsi untuk menampilkan keterangan tentang software dan si pembuat software.

4.2.6.2 Perancangan antarmuka form utama

Form ini digunakan sebagai tampilan awal program saat dijalankan. Desain GUI form utama dapat dilihat pada **Gambar 4.11**

Data		Keuntungan		Hutang		Grafik	
Index	Nama	Berat	Nilai	Index	Nama	Berat	Nilai
1	Barang 1	20	150				
2	Barang 2	34	124				
3	Barang 3	12	950				
4	Barang 4	17	782				
5	Barang 5	25	75				
6	Barang 6	34	852				
7	Barang 6	56	274				
8	Barang 7	27	123				
9	Barang 8	11	127				
10	Barang 9	15	129				
11	Barang 10	66	256				
12	Barang 11	32	664				
13	Barang 12	76	541				
14	Barang 13	35	134				
15	Barang 14	42	555				

INSERT DELETE
UPDATE REFRESH

OPTIMASI

Nilai Optimal

Grafik Fitness

Monitor Proses

Gambar 4.11 Perancangan form utama

4.2.6.3 Perancangan antarmuka form report

Form report digunakan untuk menampilkan hasil report program setelah melakukan optimasi knapsack, Desain GUI form report dapat dilihat pada **Gambar 4.12**

Data	Report	Hasil	Grafik
Jumlah barang dalam satu individu : 1			
Generasi ke : 0			
Index	Fitness tiap individu	Berat tiap individu	Kromosom
0	12250	35	16
1	15504	34	49
2	15750	35	03
3	8400	24	00
4	13690	37	28
5	13690	37	28
6	15750	35	03
7	13690	37	28
8	13690	37	28
9	13690	37	28
10	13690	37	28
11	7200	32	23
12	13500	7	45
13	5590	12	21
14	13500	7	45
15	13590	37	28
16	5590	12	21
17	13590	37	28
18	13500	7	45
19	5590	12	21

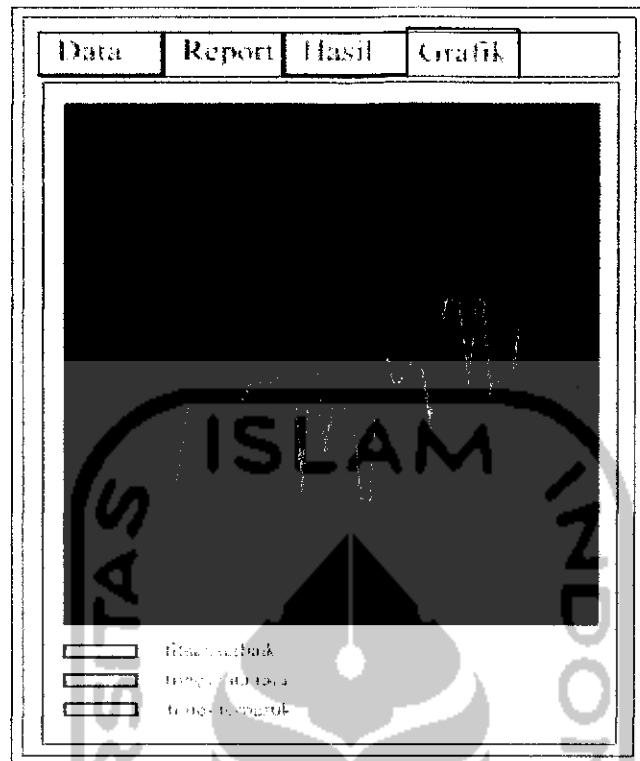
Print to File

Gambar 4.12 Perancangan form report

4.2.6.4 Perancangan antarmuka form grafik

Form grafik digunakan untuk menampilkan hasil report program berupa grafik chart 2 dimensi, Desain GUI form grafik dapat dilihat pada

Gambar 4.13



Gambar 4.13 Perancangan form grafik

4.2.6.5 Perancangan antarmuka form hasil optimasi

Form hasil optimasi digunakan untuk menampilkan hasil setelah dilakukan optimasi, Desain GUI form hasil optimasi dapat dilihat pada

Gambar 4.14

Data	Report	Hasil	Grafik
Nomor Terbit : 1131 Kode Geometri : 27 Jumlah Barang : 4 Berat Total Barang dalam Pkg : 50 Harga Total Barang dalam Pkg : 90 Kapasitas Pkg Mahkota : 50 Kontainer : -- 12140400 --			

Gambar 4.14 Perancangan form optimasi

4.2.6.6 Perancangan antarmuka form insert data

Form insert data digunakan untuk menambah data barang, Desain GUI form tambah data dapat dilihat pada **Gambar 4.15**

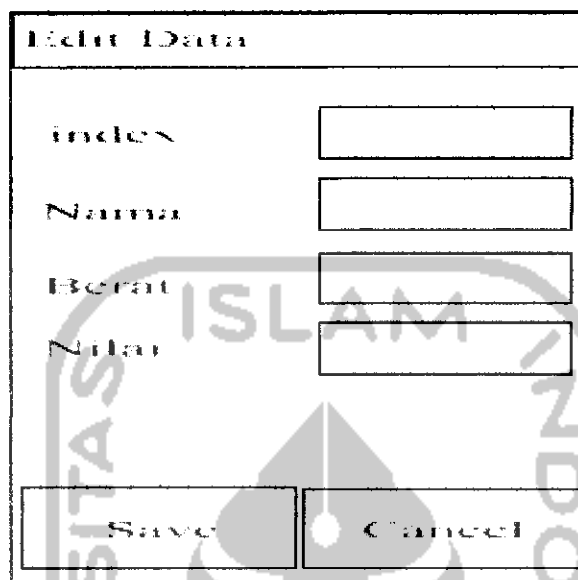
Tambah Data	
Index	<input type="text"/>
Nama	<input type="text"/>
Berat	<input type="text"/>
Zifar	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Gambar 4.15 Perancangan form insert data

4.2.6.7 Perancangan antarmuka form edit data



Form edit data digunakan untuk mengubah/edit data barang,
Desain GUI form edit data dapat dilihat pada **Gambar 4.16**

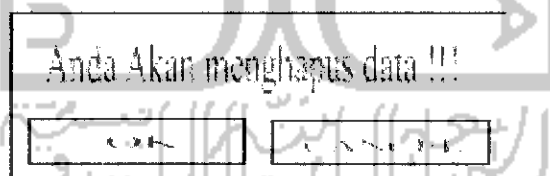


Edit Data	
index	<input type="text"/>
Nama	<input type="text"/>
Berat	<input type="text"/>
Nilai	<input type="text"/>
Save	Cancel

Gambar 4.16 Perancangan form edit data

4.2.6.8 Perancangan antarmuka form delete data

Form delete data digunakan untuk menghapus data barang,
Desain GUI form delete data dapat dilihat pada **Gambar 4.17**



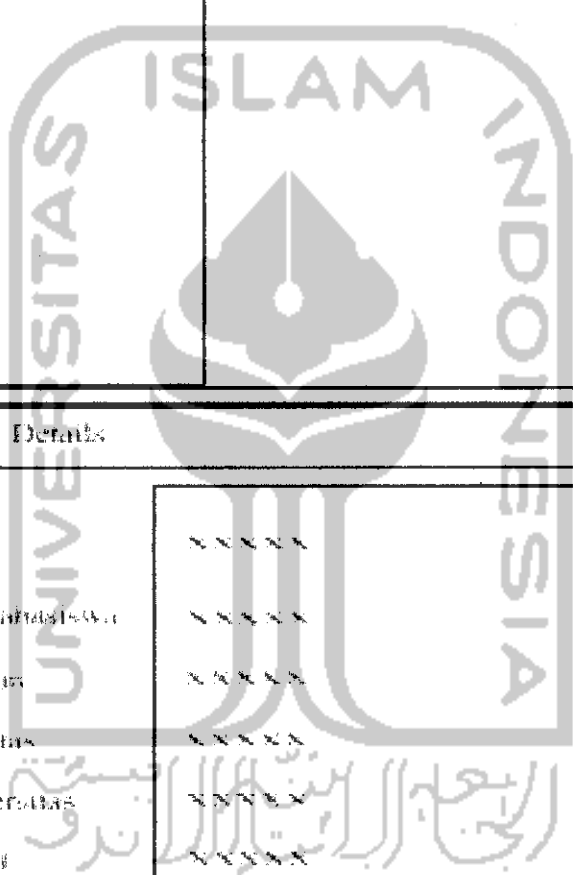
Anda Akan menghapus data !!!	
OK	Cancel

Gambar 4.17 Perancangan form hapus data

4.2.6.9 Perancangan antarmuka form about

Form about digunakan untuk menampilkan tentang keterangan software dan sekaligus menampilkan keterangan si pembuat software.

Desain GUI form about dapat dilihat pada **Gambar 4.18**

Author	
	
Author Details	
Nama	xxxxxx
No Mahasiswa	xxxxxx
Jurusan	xxxxxx
Fakultas	xxxxxx
Universitas	xxxxxx
Email	xxxxxx

Gambar 4.18 Perancangan form about

4.2.7 Perancangan operator algoritma genetika

4.2.7.1 Seleksi

Dalam penelitian ini seleksi yang digunakan adalah seleksi dengan metode roda rolet (*roulette wheel selection*) yaitu individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran fitnessnya.

Sebuah bilangan acak dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.

4.2.7.2 Crossover

Dalam aplikasi ini user bisa memilih sendiri metode *crossover* yang akan dipakai, aplikasi ini menyediakan 4 metode yaitu

1. *crossover* satu titik (*one point crossover*)
2. *crossover* banyak titik (*multi point crossover*)
3. *crossover* seragam (*uniform crossover*)
4. *crossover* 2 titik (*two point crossover*)

4.2.7.3 Mutasi

Pada penelitian ini metode mutasi yang digunakan adalah mutasi biner, yaitu mula mula kita tentukan Probabilitas mutasi (P_m) kemudian bangkitkan bilangan acak antara 0..1 sejumlah panjang kromosom * N populasi, kemudian nilai acak tadi dibandingkan dengan probabilitas mutasi, jika nilai acak $< P_m$ maka bit terkena mutasi (bit

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi

Aplikasi algoritma genetika dalam knapsack problem ini dikembangkan dengan perangkat lunak Visual Basic.Net 2003 keluaran Microsoft yang berjalan pada sistem operasi Microsoft Windows XP Professional Edition Service Pack 1, sedangkan aplikasi *database* yang digunakan adalah Microsoft Access 2003.

Perangkat keras yang digunakan dalam pengembangan aplikasi penelitian tugas akhir ini adalah komputer desktop standar dengan spesifikasi hardware sebagai berikut:

- a. Processor intel Pentium 4 1,8 Ghz
- b. 80GB hard drive
- c. 15" CRT monitor
- d. 128 Mb Gforce Fx 5200 video memory
- e. DVD CDR Drive

5.2 Implementasi Perangkat Lunak

Adalah proses membangun suatu perangkat lunak dengan menggunakan perangkat lunak lain mulai dari pembuatan antarmuka perangkat lunak sampai proses akhir.

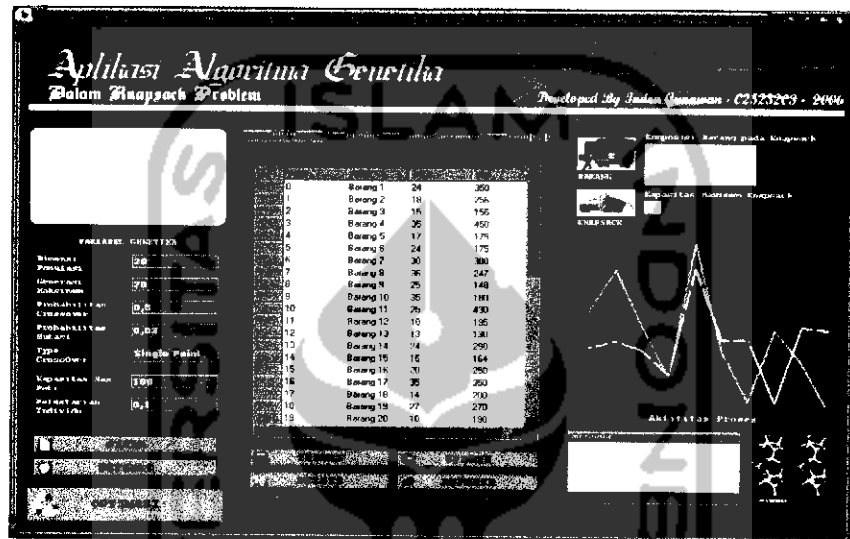
5.2.1 Perancangan antarmuka perangkat lunak

Adalah proses pembuatan antarmuka suatu perangkat lunak sehingga *user* dapat berinteraksi dengan perangkat lunak tersebut

dengan mudah dan cepat, perangkat lunak yang digunakan dalam pembuatan antarmuka ini yaitu Visual Basic.net 2003.

5.2.1.1 Perancangan antarmuka form utama

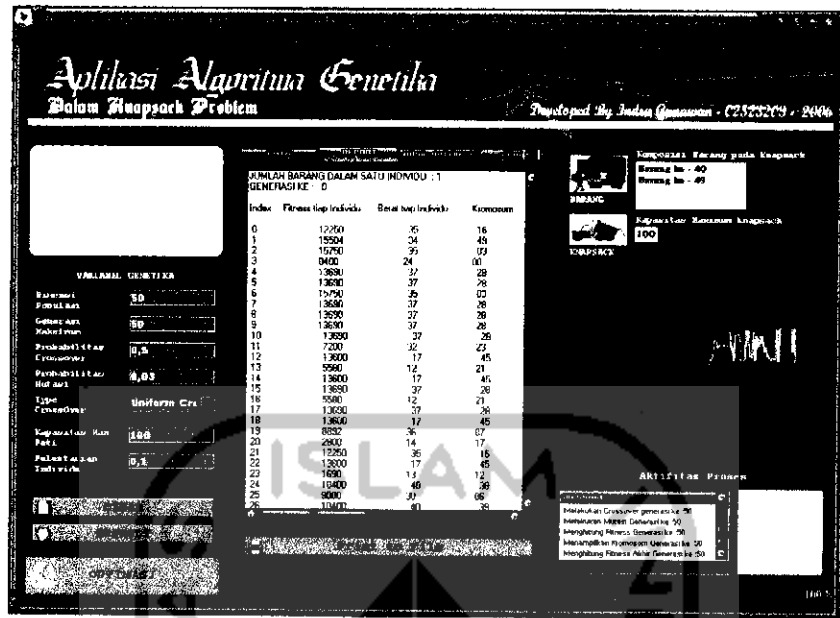
Form ini digunakan sebagai tampilan awal program saat dijalankan. Desain GUI form utama dapat dilihat pada **Gambar 5.1**



Gambar 5.1 Form utama

5.2.1.2 Perancangan antarmuka form report

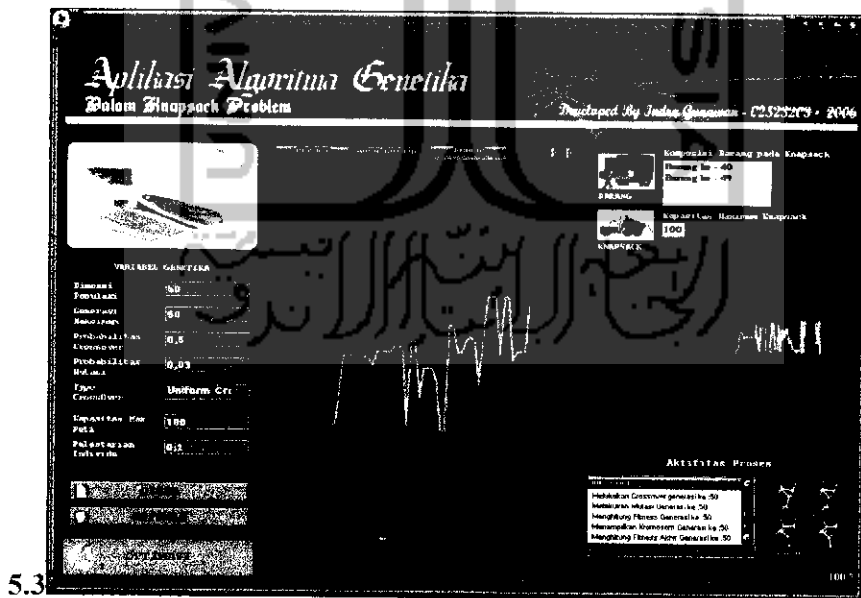
Form report digunakan untuk menampilkan hasil report program setelah melakukan optimasi knapsack, Desain GUI form report dapat dilihat pada **Gambar 5.2**



Gambar 5.2 Form report

5.2.1.3 Perancangan antarmuka form grafik

Form grafik digunakan untuk menampilkan hasil report program berupa grafik chart 2 dimensi. Desain GUI form grafik dapat dilihat pada Gambar

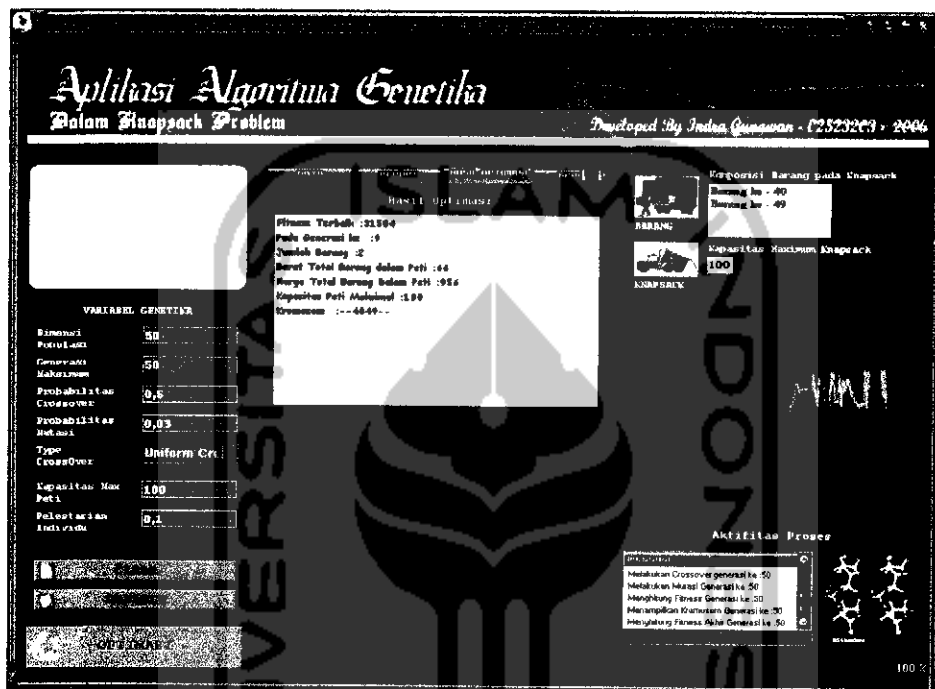


Gambar 5.3 Form Grafik

5.2.1.4 Perancangan antarmuka form hasil optimasi

Form hasil optimasi digunakan untuk menampilkan hasil setelah dilakukan optimasi, Desain GUI hasil optimasi dapat dilihat pada

Gambar 5.4



Gambar 5.4 Form optimasi

5.2.1.5 Perancangan antarmuka form insert data

Form insert data digunakan untuk menambah data barang, Desain GUI form insert data dapat dilihat pada **Gambar 5.5**

Gambar 5.5 Form insert data

5.2.1.6 Perancangan antarmuka form edit data

Form edit data digunakan untuk mengubah/edit data barang, Desain GUI form edit data dapat dilihat pada **Gambar 5.6**

Gambar 5.6 Form edit data

5.2.1.7 Perancangan antarmuka form delete data

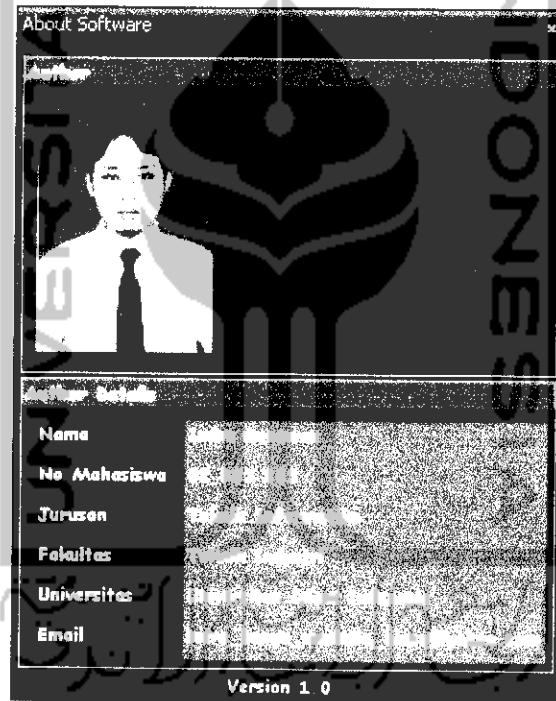
Form edit data digunakan untuk mengubah/edit data barang, Desain GUI form edit data dapat dilihat pada **Gambar 5.7**



Gambar 5.7 Form delete data

5.2.1.8 Perancangan antarmuka form about

Form about digunakan untuk menampilkan tentang keterangan software dan sekaligus menampilkan keterangan si pembuat software. Desain GUI form about dapat dilihat pada **Gambar 5.8**



Gambar 5.8 Form about

BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

Pengujian perangkat lunak merupakan suatu pengetesan program sebelum diterapkan. Program disini diuji dengan maksud untuk menemukan kesalahan kesalahan yang mungkin terjadi. Adapun beberapa kemungkinan kesalahan yang terjadi antara lain:

1. Kesalahan bahasa (*language error*) yang merupakan kesalahan dalam penulisan program.
2. Kesalahan sewaktu proses (*run time error*) dan dapat ditemukan pada saat program dijalankan atau dieksekusi.
3. Kesalahan logika (*logical error*) merupakan kesalahan yang sulit ditemukan karena kesalahan terjadi pada logika program yang dibuat. Untuk melakukan pengetesan dibutuhkan pengujian dengan menggunakan data.

6.1 Proses analisa sistem

Yaitu proses menganalisa perangkat lunak secara keseluruhan mulai dari antarmuka, input sampai report program sehingga bisa diketahui kesalahan kesalahan yang muncul pada tiap proses seingga bisa dicari penyelesaiannya.

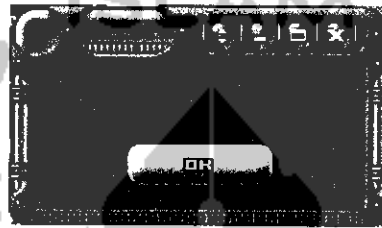
6.1.1 Kesalahan dalam basis data

Yaitu proses menganalisa suatu perangkat lunak dengan memasukkan berbagai macam input pada basis data melalui perangkat lunak tersebut sehingga bisa diketahui kesalahan-kesalahan yang akan

muncul pada saat user memasukkan data sehingga bisa memunculkan suatu peringatan agar user mengoreksi data yang dimasukkannya agar perangkat lunak berjalan normal.

6.1.1.1 Kesalahan dalam memasukkan nilai string

Kesalahan ini muncul apabila user salah dalam memasukkan nilai integer menjadi string. Pesan kesalahan seperti pada **Gambar 6.1**

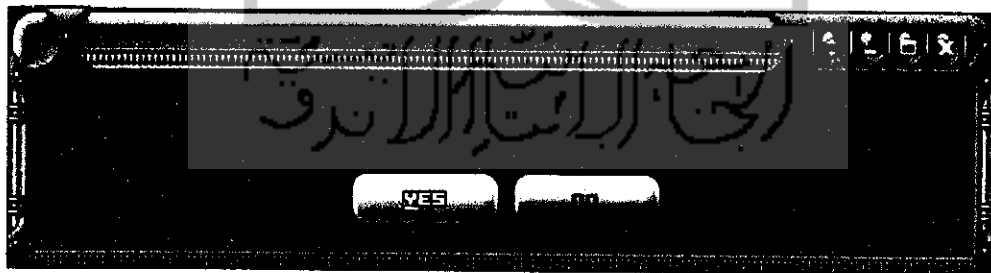


Gambar 6.1 Pesan kesalahan string

6.1.2 Kesalahan dalam algoritma genetika

6.1.2.1 Kesalahan dalam memasukkan nilai max generasi

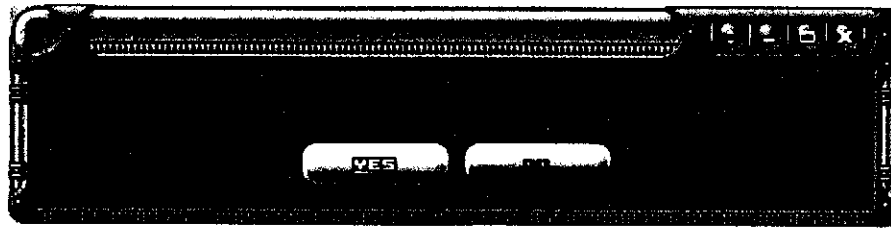
Kesalahan ini muncul apabila user salah dalam memasukkan nilai max generasi. Pesan kesalahan seperti pada **Gambar 6.2**



Gambar 6.2 Pesan kesalahan max generasi

6.1.2.2 Kesalahan dalam memasukkan nilai max populasi

Kesalahan ini muncul apabila user salah dalam memasukkan nilai max generasi. Pesan kesalahan seperti pada **Gambar 6.3**



Gambar 6.3 Pesan kesalahan max populasi

6.1.2.3 Kesalahan dalam memasukkan nilai probabilitas crossover

Kesalahan ini muncul apabila user salah dalam memasukkan nilai probabilitas crossover. Pesan kesalahan seperti pada **Gambar 6.4**



Gambar 6.4 Pesan kesalahan crossover

6.1.2.4 Kesalahan dalam memasukkan nilai probabilitas mutasi

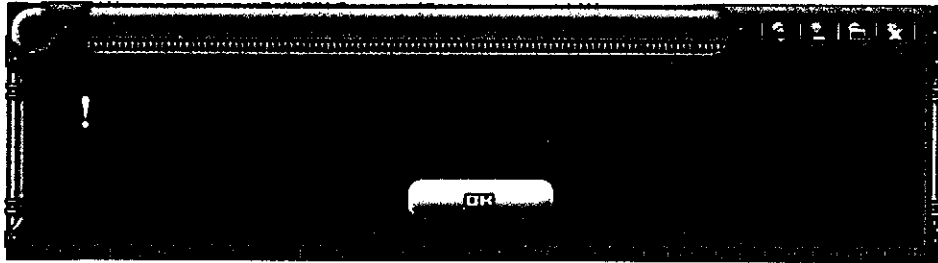
Kesalahan ini muncul apabila user salah dalam memasukkan nilai probabilitas mutasi. Pesan kesalahan seperti pada **Gambar 6.5**



Gambar 6.5 Pesan kesalahan mutasi

6.1.2.5 Kesalahan dalam memasukkan nilai kapasitas max peti

Kesalahan ini muncul apabila user salah dalam memasukkan nilai max peti / knapsack. Pesan kesalahan seperti pada **Gambar 6.6**



Gambar 6.6 Pesan kesalahan max peti

6.1.2.6 Kesalahan dalam memasukkan nilai variabel breeder

1. Kesalahan ini muncul apabila user salah dalam memasukkan nilai variabel breeder. Pesan kesalahan seperti pada **Gambar 6.7**



Gambar 6.7 Pesan kesalahan breeder

2. Kesalahan ini muncul apabila user salah dalam memasukkan range nilai variabel breeder. Pesan kesalahan seperti pada **Gambar 6.8**



Gambar 6.8 Pesan kesalahan breeder

6.1.2.7 Kesalahan dalam memasukkan nilai string

Kesalahan ini muncul apabila user salah dalam memasukkan nilai integer menjadi string. Pesan kesalahan seperti pada **Gambar 6.9**



Gambar 6.9 Pesan kesalahan nilai string

6.2 Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui kondisi optimal dari sistem dengan memberi berbagai macam bentuk pola data barang pada tabel barang sehingga didapat hasil optimasi yang paling optimal baik dari segi waktu maupun value.

6.2.1 Pengujian dengan basis data

Pengujian dengan variabel basis data dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.1**, dan hasil dari optimasi dengan variabel basis data dapat dilihat pada **Tabel 6.2**.

Tabel 6.1 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	50
2	Jumlah Generasi	50
3	Probabilitas crossover	0,5
4	Probabilitas mutasi	0,03
5	Type Crossover	single point
6	Kapasitas max knapsack	100 satuan barang
7	Breeder	0,1
8	Jumlah data	30 barang

Tabel 6.2 Hasil optimasi

No	Pola data barang	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	Kolom berat terurut naik, kolom nilai terurut turun.	58	980	98
2	Kolom berat terurut turun, Kolom nilai terurut naik.	58	1050	91
3	Kolom berat sama, Kolom nilai acak.	68	3270	100
4	Kolom berat dan nilai terurut naik.	55	768	76
5	Kolom berat dan nilai terurut turun.	58	970	97

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan waktu tercepat maka pola data barang yang paling sesuai adalah kolom berat dan nilai terurut naik.
2. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka pola data barang yang paling sesuai adalah kolom berat nilainya sama dan kolom nilai acak.

6.2.2 Pengujian dengan variabel genetika

Pengujian dengan variabel genetika ini terbagi menjadi 5 :

1. Pengujian dengan variabel P_c (probabilitas crossover).
2. Pengujian dengan variabel P_m (probabilitas mutasi).

3. Pengujian dengan variabel BGA (Breeder GA).
4. Pengujian dengan jumlah populasi.
5. Pengujian dengan generasi maksimal.

6.2.2.1 Pengujian dengan variabel Pc (probabilitas crossover).

Pengujian dengan variabel probabilitas crossover dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.3**, dan hasil dari optimasi dengan variabel probabilitas crossover dapat dilihat pada **Tabel 6.4**.

Tabel 6.3 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	50
2	Jumlah Generasi	50
3	Probabilitas crossover	dinamis
4	Probabilitas mutasi	0,03
5	Type Crossover	single point
6	Kapasitas max knapsack	100 satuan barang
7	Breeder	0,1
8	Jumlah data	30 barang

Tabel 6.4 Hasil optimasi

No	Probabilitas crossover	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	0,1	49	980	90
2	0,2	49	980	90
3	0,3	49	1285	94
4	0,4	49	900	80
5	0,5	49	1310	95
6	0,6	49	900	80
7	0,7	49	1110	96
8	0,8	49	900	90
9	0,9	51	900	90

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka probabilitas crossover yang paling sesuai adalah $P_c=0,5$.

6.2.2.2 Pengujian dengan variabel P_m (probabilitas mutasi).

Pengujian dengan variabel probabilitas mutasi dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.5**, dan hasil dari optimasi dengan variabel probabilitas mutasi dapat dilihat pada **Tabel 6.6**.

Tabel 6.5 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	50
2	Jumlah Generasi	50
3	Probabilitas crossover	0,5
4	Probabilitas mutasi	dinamis
5	Type Crossover	single point
6	Kapasitas max knapsack	100 satuan barang
7	Breeder	0,1
8	Jumlah data	30 barang

Tabel 6.6 Hasil optimasi

No	Probabilitas mutasi	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	0,01	49	1064	96
2	0,02	49	900	90
3	0,03	49	900	80
4	0,04	49	900	90
5	0,05	49	900	90
6	0,06	49	900	90
7	0,07	49	900	90
8	0,08	49	1365	92
9	0,09	49	900	80

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka probabilitas mutasi yang paling sesuai adalah $P_m=0,8$.

6.2.2.3 Pengujian dengan variabel BGA (Breeder GA).

Pengujian dengan variabel BGA dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.7**, dan hasil dari optimasi dengan variabel BGA dapat dilihat pada **Tabel 6.8**.

Tabel 6.7 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	50
2	Jumlah Generasi	50
3	Probabilitas crossover	0,5
4	Probabilitas mutasi	0,03
5	Type Crossover	single point
6	Kapasitas max knapsack	100 satuan barang
7	Breeder GA	dinamis
8	Jumlah data	30 barang

Tabel 6.8 Hasil optimasi

No	Breeder GA	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	0,1	35	1076	100
2	0,2	35	900	90
3	0,3	35	900	80
4	0,4	35	900	80
5	0,5	35	900	90
6	0,6	35	900	90
7	0,7	49	1156	98
8	0,8	49	1285	94
9	0,9	49	900	90

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka nilai BGA yang paling sesuai adalah BGA=0,8.

6.2.2.4 Pengujian dengan variabel populasi.

Pengujian dengan variabel populasi dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.9**, dan hasil dari optimasi dengan variabel populasi dapat dilihat pada **Tabel 6.10**.

Tabel 6.9 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	dinamis
2	Jumlah Generasi	20
3	Probabilitas crossover	0,5
4	Probabilitas mutasi	0,03
5	Type Crossover	single point
6	Kapasitas max knapsack	100 satuan barang
7	Breeder	0,1
8	Jumlah data	30 barang

Tabel 6.10 Hasil optimasi

No	Populasi	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	10	8	1594	100
2	20	8	1000	80
3	30	9	1650	77
4	40	10	1100	92
5	50	12	300	75

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka nilai populasi yang paling sesuai adalah Pop=30 yaitu sama dengan jumlah barang.
2. Semakin besar nilai populasi semakin lama pula waktu optimasinya.

6.2.2.5 Pengujian dengan variabel generasi.

Pengujian dengan variabel generasi dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.11**, dan hasil dari optimasi dengan variabel generasi dapat dilihat pada **Tabel 6.12**.

Tabel 6.11 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	50
2	Jumlah Generasi	dinamis
3	Probabilitas crossover	0,5
4	Probabilitas mutasi	0,03
5	Type Crossover	single point
6	Kapasitas max knapsack	100 satuan barang
7	Breeder	0,1
8	Jumlah data	30 barang

Tabel 6.12 Hasil optimasi

No	Populasi	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	10	5	1500	87
2	20	12	1450	84
3	30	19	950	67
4	40	27	1100	92
5	50	36	1300	90

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka nilai generasi yang paling sesuai adalah Gen=10 yaitu sama dengan sepertiga jumlah barang.
2. Semakin besar nilai generasi semakin lama pula waktu optimasinya.

6.2.2.6 Pengujian dengan type crossover.

Pengujian dengan variabel type crossover dilakukan dengan menggunakan variabel seperti terlihat pada **Tabel 6.13** , dan hasil dari optimasi dengan variabel type crossover dapat dilihat pada **Tabel 6.14**.

Tabel 6.13 Variabel genetika

No	Variabel genetika	nilai
1	Lebar populasi	50
2	Jumlah Generasi	50
3	Probabilitas crossover	0,5
4	Probabilitas mutasi	0,03
5	Type Crossover	dinamis
6	Kapasitas max knapsack	100 satuan barang
7	Breeder	0,1
8	Jumlah data	30 barang

Tabel 6.14 Hasil optimasi

No	Type crossover	Waktu optimasi(detik)	Keuntungan(Rp)	Berat (1 satuan barang)
1	Satu titik	1 menit 58 detik	300	75
2	Dua titik	1 menit 59 detik	300	300
3	Banyak titik	1 menit 59 detik	1400	89
4	Seragam	1 menit 22 detik	450	35

Dari hasil pengujian diatas dapat ditarik kesimpulan bahwa :

1. Untuk mendapatkan hasil optimasi dengan keuntungan yang paling maksimal maka type crossover yang paling sesuai adalah type crssover dengan banyak titik (*multipoint crossover*).

6.2.2.7 Kesimpulan

Dari kelima macam pengujian diatas dapat ditarik kesimpulan bahwa optimasi dengan GA pada aplikasi perangkat lunak ini dapat mencapai nilai optimal dalam keadaan seperti pada **Tabel 6.15** :

Tabel 6.15 Keadaan optimal

No	Variabel	Nilai
1	Probabilitas crossocer (Pc)	0,5
2	Probabilitas mutasi (Pm)	0,8
3	Breeder GA (BGA)	0,8
4	Jumlah populasi	Pop >= jumlah barang
5	Jumlah generasi	Gen >= 1/3 jumlah barang
6	Type Crossover	Banyak titik (<i>multipoint crossover</i>)

BAB VII

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan terhadap aplikasi Algoritma Genetika dalam Knapsack problem yang telah diimplementasikan serta saran dan pengembangannya lebih lanjut.

7.1 Kesimpulan

Kesimpulan penting yang dapat diambil dari penelitian yang telah dilakukan ini adalah dengan dibangunnya perangkat lunak ini diharapkan dapat membantu para pembuat keputusan pada suatu perusahaan secara khusus dan para pengguna komputer secara umum untuk membantu menyelesaikan masalah knapsack problem dan menentukan barang apa saja yang seharusnya diangkut pada suatu kontainer/peti dengan kapasitas yang terbatas agar dapat mencapai keuntungan yang paling maksimal.

7.2 Saran

Beberapa saran, kemungkinan pengembangan, dan penelitian lanjutan yang dapat dilakukan adalah:

1. Untuk menghasilkan pencarian yang optimal diperlukan variabel-variabel genetika yang tepat yaitu dengan melakukan uji coba dengan memakai semua kombinasi variabel variabel yang mungkin hingga ditemukan komposisi variabel genetika yang paling optimal.

2. Aplikasi ini hanya bisa menangani hingga maksimal 100 data barang, disini perlu dilakukan perbaikan sehingga aplikasi ini bisa menampung data yang lebih banyak lagi.



DAFTAR PUSTAKA

- [GAM04] *Anonimous*, <http://faculty.petra.ac.id>, di akses tanggal 1 Juli 2006.
- [ITS06] *Anonimous*, <http://web.si.its-sby.edu> , diakses tanggal 1 Juli 2006.
- [JAZ06] *Anonimous*, www.jazi.staff.ugm.ac.id, di akses tanggal 1 Juli 2006.
- [MED06] *Anonimous*, www.mediaindo.co.id, di akses tanggal 1 Juli 2006.
- [SRI03] Kusumadewi, Sri. 2003. *Artificial Intelligence*. Yogyakarta. Andi Publiser.
- [WKP06] *Anonimous*, www.id.wikipedia.org, di akses tanggal 1 Juli 2006.

