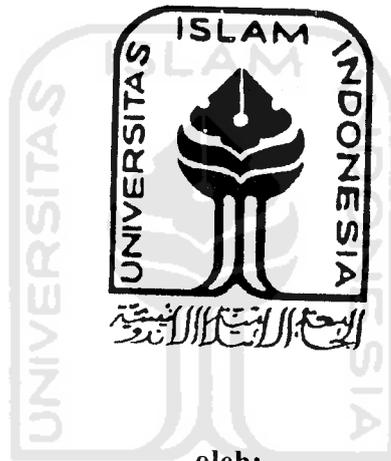


**PENENTUAN SUDUT KEMIRINGAN PUNGGUNG
PADA PENGANGKATAN BEBAN KARUNG MENGGUNAKAN
ALGORITMA GENETIKA**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



oleh:

**Nama : Abriyanti Widaningtyas Sihtyowarini
No. Mahasiswa : 01523193
NIRM : 01523193**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**



PERNYATAAN KEASLIAN TUGAS AKHIR

Saya yang bertanda tangan di bawah ini,

Nama : Abriyanti Widaningtyas Sihtyowarini

No. Mahasiswa : 01523193

Menyatakan bahwa Tugas Akhir dengan judul:

**Penentuan Sudut Kemiringan Punggung Pada Pengangkatan Beban Karung
Menggunakan Algoritma Genetika**

yang diajukan untuk diuji pada tanggal 29 Mei 2007 adalah hasil karya saya.

Dengan ini saya menyatakan dengan bahwa seluruh komponen dan isi di dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

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

Yogyakarta, 29 Mei 2007

(Abriyanti Widaningtyas Sihtyowarini)

LEMBAR PENGESAHAN PEMBIMBING

**PENENTUAN SUDUT KEMIRINGAN PUNGGUNG
PADA PENGANGKATAN BEBAN KARUNG MENGGUNAKAN
ALGORITMA GENETIKA**

TUGAS AKHIR



oleh:

Nama : Abriyanti Widaningtyas Sihtyowarini
No. Mahasiswa : 01523193
NIRM : 01523193

Yogyakarta, 15 Mei 2007

Pembimbing,

A handwritten signature in black ink, appearing to be 'Sri Kusumadewi', is written over the name of the supervisor.

Sri Kusumadewi, S.Si., MT

LEMBAR PENGESAHAN PENGUJI

PENENTUAN SUDUT KEMIRINGAN PUNGGUNG
PADA PENGANGKATAN BEBAN KARUNG MENGGUNAKAN
ALGORITMA GENETIKA

TUGAS AKHIR

oleh:

Nama : Abriyanti Widaningtyas Sihtyowarini
No. Mahasiswa : 01523193
NIRM : 01523193

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

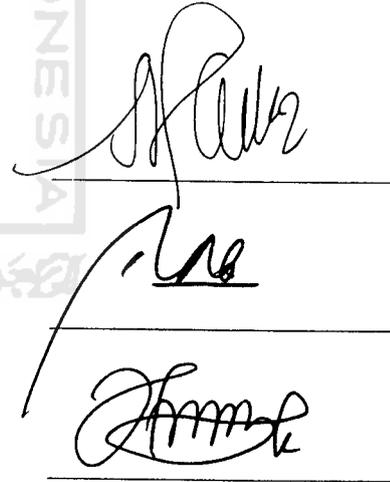
Yogyakarta, 29 Mei 2007

Tim Penguji,

Sri Kusumadewi, S.Si, MT
Ketua

Yudi Prayudi, S.Si, M.Kom
Anggota I

Hendrik, ST
Anggota II



Mengetahui,
Ketua Jurusan Teknik Informatika

Yudi Prayudi, S.Si, M.Kom

Thanks to...

ALLAH SWT, for every breath I take.. How can I not be grateful?

Ibu Bapakku, who never stop, to deadly Love me, pray 4 me, and understand this difficult daughter, more 'n more, in your every breath.. I Love U, mom, dad.. so much, of course!

AB 4256 KS, bestfriend for 7 years.... 'n my AX900 who often made me mad ☺

Beloved friends.. Lina, Diana, Spiet, Andre, Nonie..... and all people of Biner 2001.. guyz, u are the colours of my beautiful rainbow..

LaLa, thx 4 all vouchers, guest lists n laptop.. tanpamu aku bisa gila saat pendadaran..

Aad, Q-K, Dessy, Ino, yang telah mau, sudi, bersedia, dan rela dengan sepenuh hati menemaniku, without complain, menghabiskan 2 tahun gak jelas di kos bunga yang kini dah gak jelas juga.. Selamanya kita tak akan berhenti mengejar matahari ☺

Makhluk KKN.. Heru, kocokku sing ireng, Lemu + ora mutu ☹ Udin, for the weirdo movie + free talk + godniL + petuah-petuahmu sing wagu ☺ Ago (yang kadang kala membuatku terpana), Lita, Paklek, Ririn, Pakde, Muchlas "wayang", Dian, Amreeh, Yoga, Ike.. Nice to have spent the unforgettable 4 months++++ with u, guys

Belakang HJATT community.. Ria, the big fat devil a.k.a setan gudul.. u cut me inside...!!! ☹ Buzul, bwad doa di depan Ka'bah + all borrowed stuffs. Wita, with kostum "pembantu"nya yang....hm., nice! 'n selalu bikin gw pengen bersin ☺

All ITers Trans IV, who gave me chance, jadi pengurus en bikin gaduh @ 2nd floor trans tip.. QbaL, gw cuman jadi recycle bin neeh?! ☹ Vebly and Ina, teman bergosip di toilet + LG Bank Mega ☺ Oedoe, kita niy kepengen lulus aja kok susah? ☹

Gusti_79, yang telah memanjakan kegilaan + akal sehat gw, sekaligus! big tengkyu...

Sebuah sChatZt, nu world, nu start, nu day, nu plan.. di kala tiap2 sen duidku terus menguap dan berlatu, yang tersisa hanya dirimu + mie instan :) semoga perih terbang tinggi di awan.. GSE

Benx, my everlasting best friend.. ever after! (waksss, apaan seek?) Even though we've choosed our own different way, you have influenced most of the way I do

Tmonsterix, bwad doa, support, usuh, bete, wise words yang begitu "menampar" gw which push me to always try to stand up on my own feet, the bajakan CDs, flash disk abal-abal, etc

Amijoyo, temen maen petak umpet ☺

Other friends, all fans ☺ and big families, sebangsa dan setanah air.. you're all the beautiful people whose God has given to me.. Really..! I Love U

MY MOTTO

"Hai orang-orang yang beriman, mintalah pertolongan dari Allah dengan kesabaran dan salat. Sungguh Allah bersama orang-orang yang sabar"
(Q.S Al Baqarah: 153)

"Sungguh, bersama kesukaran pasti ada kemudahan. Dan bersama kesukaran pasti ada kemudahan. Karena itu, bila selesai suatu tugas, mulailah tugas lain dengan sungguh-sungguh"
(Q.S Asy-Syarah: 5-7)

"Sesungguhnya Allah tidak akan merubah nasib suatu kaum sampai mereka merubah keadaan yang ada pada diri mereka sendiri. Dan apabila Allah menghendaki keburukan terhadap suatu kaum, maka tidak ada yang dapat menolaknya, dan tidak ada pelindung bagi mereka kecuali Allah"
(Q.S. Ar-Ra'd: 11)

"Sesungguhnya apabila Ia menghendaki sesuatu, cukuplah Ia berkata, "Jadilah!" Maka terjadilah ia"
(Q.S. Yaasiin: 82)

"Ya Tuhanku, lapangkanlah dadaku. Lepaskan ikatan yang mengekang lidahku, agar mereka paham segala perkataanku"
(Q.S Thaahaa: 25-28)

*Ya Tuhan kami, janganlah Engkau bebankan kepada kami beban yang berat sebagaimana Engkau bebankan kepada orang-orang sebelum kami.
Ya Tuhan kami, janganlah Engkau pikulkan kepada kami apa yang tak sanggup kami memikulnya.*
(Q.S Al-Baqarah: 286)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb.

Alhamdulillah rabbi' alamin, puji syukur kehadiran Allah SWT, atas limpahan rahmat dan hidayat-Nya, sehingga Tugas Akhir dengan judul **“Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika”** ini dapat terselesaikan dengan baik.

Tugas Akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Jurusan Teknik Informatika, Universitas Islam Indonesia, dan atas apa yang telah diajarkan selama perkuliahan baik teori maupun praktek, di samping laporan sendiri yang merupakan rangkaian kegiatan yang harus dilakukan setelah tugas akhir ini selesai.

Pada kesempatan ini penyusun ingin mengucapkan banyak terima kasih kepada pihak-pihak yang mempunyai andil besar dalam pelaksanaan dan penyelesaian laporan tugas akhir ini, antara lain :

1. Bapak Fathul Wahid S.T., M.Sc sebagai Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi, S.Si, M.Kom selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Sri Kusumadewi, S.Si., MT selaku Dosen Pembimbing Tugas Akhir.
4. Bapak Irving Vitra Papatungan, ST selaku Dosen Pembimbing Tugas Akhir.

5. Bapak, Ibu dosen Teknik Informatika dan dosen-dosen Universitas Islam Indonesia.
6. Ayahanda Bambang Sulistyohadi dan Ibunda Titiek Safitri yang telah memberikan doa yang tidak pernah putus, semangat dan dukungan moril maupun materiil.
7. Teman-teman Teknik Informatika UII 2001 yang memberikan semangat, dorongan, bantuan sehingga penyusun dapat menyelesaikan tugas akhir dengan baik.
8. Semua pihak yang tidak dapat disebut satu persatu yang telah membantu sehingga laporan tugas akhir ini dapat terselesaikan.

Di tengah keterbatasan penyusun dalam laporan tugas akhir ini, semoga laporan ini dapat menambah khasaah pengetahuan di bidang teknologi informasi serta bermanfaat bagi pembaca. Semoga Allah SWT membimbing dan menyertai setiap langkah kita. Amin.

Wassalamu'alaikum Wr.Wb.

Yogyakarta, 12 Mei 2007

Penyusun

SARI

Algoritma genetika (*genetic algorithm*) merupakan suatu metode pembelajaran mesin yang menggunakan prinsip evolusi genetika, di mana spesies yang mampu beradaptasi dengan lingkungan akan dapat terus bertahan hidup, sedangkan yang tidak mampu akan tersingkir.

Pencarian sudut optimum pada *manual material handling* dalam aplikasi ini merupakan permasalahan optimasi dalam mencari sudut awal yang bernilai optimum pada pengangkatan beban dengan cara memanggul sehingga dapat meminimumkan resiko cedera pada tulang belakang.

Penerapan algoritma genetika digunakan untuk mendapatkan solusi yang optimum pada permasalahan optimasi ini. Kromosom dibangkitkan melalui representasi sudut-sudut yang akan dicari yaitu θ_H dan θ_T . Nilai *fitness* yang digunakan adalah nilai *fitness* terbaik yang didapat dengan cara meminimumkan gaya total *spinal erector*. Input merupakan masukan dari panjang segmen-segmen tubuh, sudut-sudut yang bekerja pada segmen-segmen tubuh tersebut saat beban diangkat, massa beban serta massa pekerja yang melakukan pengangkatan beban. Selain itu, input juga berupa masukan dari operator-operator algoritma genetika seperti jumlah populasi, jumlah generasi, probabilitas *crossover*, dan probabilitas mutasi.

Setelah sistem dijalankan, maka hasil yang didapatkan menunjukkan bahwa algoritma genetika memberikan respon sistem yang optimum. Respon sistem yang optimum bergantung pada penentuan jumlah generasi, probabilitas *crossover* serta probabilitas mutasi.

Kata Kunci: Algoritma Genetika, *fitness*, jumlah generasi, probabilitas *crossover*, probabilitas mutasi.

TAKARIR

<i>Manual material handling</i>	suatu aktivitas pemindahan benda tanpa bantuan alat
<i>Back injury</i>	nyeri pada tulang belakang
<i>Genotype</i>	interaksi sekumpulan kromosom
<i>Phenotype</i>	proses transformasi kode-kode genetika
<i>Feature</i>	karakter
<i>Alleles</i>	nilai karakter
<i>Locus</i>	letak gen di dalam kromosom
<i>Positions</i>	posisi atau letak
<i>Fitness</i>	nilai kebugaran
<i>Fitness fuction</i>	fungsi tujuan
<i>Population</i>	populasi
<i>Generation</i>	generasi
<i>Gen</i>	sebuah satuan unsur terkecil dari suatu nilai yang berupa bit, simbol, atau karakter
<i>Chromosom</i>	suatu struktur gen yang membentuk suatu kesatuan yang mewakili suatu nilai atau karakteristik
<i>Constrain</i>	batasan-batasan
<i>Random</i>	acak
<i>Crossover</i>	perkawinan silang

<i>Mutation</i>	mutasi
<i>Parent</i>	induk
<i>Offspring</i>	anak
<i>Generational update</i>	pembaharuan generasi yang dilakukan dengan mengganti keseluruhan populasi lama dengan populasi baru
<i>Continous update</i>	pembaharuan generasi dengan cara menggantikan sebagian kromosom pada populasi lama dengan kromosom-kromosom yang baru
<i>Steady state update</i>	pendekatan selektif dengan kecenderungan mempertahankan kromosom yang baik dari generasi lama ke generasi baru, serta hanya menggantikan kromosom yang memiliki nilai <i>fitness</i> yang kurang baik
<i>Input</i>	masukan
<i>Output</i>	keluaran
<i>Flowchart</i>	diagram alir untuk menggambarkan arus data sistem
<i>Software</i>	perangkat lunak
<i>Hardware</i>	perangkat keras
<i>User</i>	pengguna

DAFTAR ISI

JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO.....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
TAKARIR.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xviii
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Alasan Pemilihan Judul.....	2
1.3 Rumusan Masalah.....	3
1.4 Batasan Masalah.....	3
1.5 Tujuan Penelitian.....	4
1.6 Manfaat Penelitian.....	4

1.7	Metodologi Penelitian.....	4
1.8	Sistematika Penulisan.....	5
BAB II	LANDASAN TEORI.....	8
2.1	Ergonomi.....	8
2.1.1	Deskripsi Ergonomi.....	8
2.1.2	Penanganan Material Secara Manual (<i>Manual Material Handling</i>).....	8
2.1.3	Batasan Beban yang Boleh Diangkat.....	10
2.1.3.1	Batasan Fisiologi (<i>Physiological Limitations</i>).....	10
2.1.3.2	Batasan Biomekanika (<i>Biomechanical Limitations</i>).....	10
2.1.4	Mekanika Tubuh Manusia.....	11
2.2	Algoritma Genetika.....	20
2.2.1	Deskripsi Algoritma Genetika.....	20
2.2.2	Sejarah Algoritma Genetika.....	22
2.2.3	Latar Belakang Biologi.....	23
2.2.4	Mekanisme Kerja Algoritma Genetika	25
2.2.5	Parameter Algoritma Genetika	28
2.2.6	Prosedur Inisialisasi.....	28
2.2.7	Representasi Kromosom.....	29
2.2.8	Fungsi Evaluasi.....	30

2.2.9	Seleksi.....	34
2.2.9.1	Seleksi Roda Roulette (<i>Roulette Wheel Selection</i>)	34
2.2.10	<i>Crossover</i>	35
2.2.11	Mutasi.....	36
2.2.12	Pembaharuan Generasi.....	38
BAB III	ANALISIS KEBUTUHAN PERANGKAT LUNAK.....	39
3.1	Metode Analisis.....	39
3.2	Hasil Analisis	39
3.2.1	Input.....	39
3.2.2	Output.....	41
3.2.3	Kebutuhan Proses.....	42
3.2.4	Antarmuka yang Diinginkan.....	42
3.2.5	Kinerja yang Diharapkan.....	43
3.2.6	Kebutuhan Perangkat Lunak dan Perangkat Keras.....	43
BAB IV	PERANCANGAN PERANGKAT LUNAK.....	44
4.1	Metode Perancangan.....	48
4.2	Hasil Perancangan.....	44
4.2.1	Perancangan <i>Flowchart</i>	44
4.2.1.1	<i>Flowchart</i> Aplikasi Algoritma Genetika.....	45
4.2.1.2	<i>Flowchart</i> Proses Algoritma Genetika.....	46
4.2.1.3	<i>Flowchart</i> Proses Seleksi Roda Roulette.....	47

4.2.1.3.1	<i>Flowchart</i> Proses Perhitungan Total <i>Fitness</i>	48
4.2.1.3.2	<i>Flowchart</i> Proses Perhitungan <i>Fitness</i> Relatif dan <i>Fitness</i> Kumulatif.....	48
4.2.1.3.3	<i>Flowchart</i> Proses Pilih Induk.....	49
4.2.1.3.4	<i>Flowchart</i> Proses <i>Crossover</i>	50
4.2.1.3.5	<i>Flowchart</i> Proses Mutasi.....	51
4.2.2	Perancangan Variabel Input.....	52
4.2.2.1	Input Resultan Momen.....	52
4.2.2.2	Input Algoritma Genetika.....	57
4.3	Perancangan Antarmuka.....	61
4.3.1	Rancangan Antarmuka Halaman Utama.....	61
4.3.2	Rancangan Antarmuka Menu.....	62
4.3.2.1	Input.....	62
4.3.2.2	Proses.....	63
4.3.2.3	Report.....	65
4.2.2.4	About.....	66
BAB V	IMPLEMENTASI PERANGKAT LUNAK.....	67
5.1	Batasan Implementasi.....	67
5.1.1	Perangkat Keras yang Dibutuhkan.....	67
5.1.2	Perangkat Lunak yang Dibutuhkan.....	68
5.1.3	Bahasa dan <i>compiler</i> yang dipakai.....	68

5.2	Implementasi Perangkat Lunak.....	68
5.3	Implementasi Prosedural.....	69
5.3.1	Modul-modul Perangkat Lunak.....	69
5.3.2.1	Math.....	69
5.3.2.2	Chart.....	69
5.3.2	Prosedur-prosedur dalam Perangkat Lunak.....	69
5.3.2.1	Kode Program untuk Algoritma Genetika.....	70
5.3.3	Proses yang Digunakan.....	77
5.4.4	Implementasi Antarmuka.....	77
5.4.4.1	Tampilan <i>Form</i> Halaman Utama.....	78
5.4.4.2	Tampilan <i>Form</i> Input.....	78
5.4.4.3	Tampilan <i>Form</i> Proses.....	79
5.4.4.4	Tampilan <i>Form</i> Report.....	80
5.4.4.5	Tampilan <i>Form</i> About.....	81
BAB VI	ANALISIS KINERJA PERANGKAT LUNAK.....	83
6.1	Pengujian Perangkat Lunak.....	83
6.2	Pengujian Terhadap Sistem.....	83
6.2.1	Pengujian Tidak Normal.....	83
6.2.2	Pengujian Normal.....	85
6.3	Analisis Pengujian Perangkat Lunak.....	103

BAB VII	PENUTUP	105
7.1	Kesimpulan.....	105
7.2	Saran.....	106
DAFTAR PUSTAKA.....		107



DAFTAR GAMBAR

Gambar 2.1	Letak Pusat Massa Tiap Segmen Tubuh.....	12
Gambar 2.2	Proyeksi Sudut, Momen Gaya dan Gaya.....	13
Gambar 2.3	Proyeksi Sudut, Momen Gaya dan Gaya Pada Telapak Tangan.....	14
Gambar 2.4	Proyeksi Sudut, Momen Gaya dan Gaya Pada Lengan Bawah.....	15
Gambar 2.5	Proyeksi Sudut, Momen Gaya dan Gaya Pada Lengan Atas.....	16
Gambar 2.6	Proyeksi Sudut, Momen Gaya dan Gaya Pada Punggung	17
Gambar 2.7	Struktur Algoritma Genetika Menurut Mitsuo Gen.....	27
Gambar 2.8	Ilustrasi <i>One-Cut-Point Crossover</i>	35
Gambar 2.9	<i>Binary Mutation</i>	38
Gambar 4.1	<i>Flowchart</i> Aplikasi Algoritma Genetika.....	45
Gambar 4.2	<i>Flowchart</i> Proses Algoritma Genetika.....	46
Gambar 4.3	<i>Flowchart</i> Proses Seleksi Roda Roulette.....	47
Gambar 4.4	<i>Flowchart</i> Total Nilai <i>Fitness</i>	48
Gambar 4.5	<i>Flowchart</i> Proses Perhitungan <i>Fitness</i> Relatif dan <i>Fitness</i> Kumulatif.....	49
Gambar 4.6	<i>Flowchart</i> Proses Piih Induk.....	50

Gambar 4.7	<i>Flowchart</i> Proses <i>Crossover</i>	51
Gambar 4.8	<i>Flowchart</i> Proses Mutasi.....	52
Gambar 4.9	Rancangan Antarmuka Halaman Utama.....	62
Gambar 4.10	Rancangan Antarmuka Input.....	63
Gambar 4.11	Rancangan Antarmuka Proses.....	64
Gambar 4.12	Rancangan Antarmuka Report.....	65
Gambar 4.13	Rancangan Antarmuka Tentang Perangkat Lunak.....	66
Gambar 5.1	<i>Form</i> Halaman Utama.....	78
Gambar 5.2	<i>Form</i> Input.....	79
Gambar 5.3	<i>Form</i> Proses.....	80
Gambar 5.4	<i>Form</i> Report.....	81
Gambar 5.5	<i>Form</i> About.....	82
Gambar 6.1	Pesan Kesalahan Input SL_4 Belum Terisi.....	84
Gambar 6.2	Pesan Kesalahan Input Jumlah Generasi Belum Terisi	84
Gambar 6.3	Pesan Kesalahan Untuk Input θ_2	84
Gambar 6.4	Pesan Kesalahan Untuk Input <i>Crossover</i>	85
Gambar 6.5	Hasil Pengujian 1.....	87
Gambar 6.6	Laporan Hasil Pengujian 1.....	88
Gambar 6.7	Hasil Pengujian 2	89
Gambar 6.8	Laporan Hasil Pengujian 2.....	90
Gambar 6.9	Hasil Pengujian 3	91

Gambar 6.10	Laporan Hasil Pengujian 3.....	92
Gambar 6.11	Hasil Pengujian 4	93
Gambar 6.12	Laporan Hasil Pengujian 4.....	94
Gambar 6.13	Hasil Pengujian 5	95
Gambar 6.14	Laporan Hasil Pengujian 5.....	96
Gambar 6.15	Hasil Pengujian 6	97
Gambar 6.16	Laporan Hasil Pengujian 6.....	98
Gambar 6.17	Hasil Pengujian 7	99
Gambar 6.18	Laporan Hasil Pengujian 7.....	100
Gambar 6.19	Hasil Pengujian 8	101
Gambar 6.20	Laporan Hasil Pengujian 8.....	102



DAFTAR TABEL

Tabel 6.1	Tabel Hasil Pengujian Perangkat Lunak	103
------------------	---	-----



BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Penggunaan teknologi pada industri di Indonesia mempunyai tingkat keragaman yang tinggi, mulai dari teknologi yang sederhana sampai dengan teknologi yang cukup canggih. Dalam setiap aktivitas industri tersebut, penggunaan teknologi manusia masih dominan, terutama kegiatan penanganan material secara manual (*manual material handling*). Aktivitas *manual material handling* mempunyai peranan vital dalam suatu produksi manufaktur, yang bertujuan untuk mengalirkan material sesuai dengan aliran prosesnya dengan tepat waktu dan tujuannya. Kelebihan *manual material handling* bila dibandingkan dengan penanganan material dengan alat bantu adalah fleksibilitas gerakan yang dapat dilakukan untuk beban-beban ringan. Selain kelebihan tersebut, *manual material handling* juga memiliki kekurangan yaitu dalam hal keselamatan dan kesehatan kerja yang disebabkan kesalahan dalam penanganan material tersebut.

Pekerjaan-pekerjaan di bidang industri memiliki kecenderungan resiko cedera yang besar. Hal ini banyak dijumpai apabila terjadi kesalahan dalam penanganan beban, misal posisi tubuh yang salah (*awkward posture*) dalam bekerja, atau beban kerja yang berat (*forcefull exertions*).

Dalam sistem kerangka manusia terdapat beberapa titik rawan, yaitu pada ruas tulang leher, ruas tulang belakang, dan pangkal paha. Dari titik-titik rawan tersebut,

titik ruas tulang belakang merupakan titik paling rawan terhadap kecelakaan kerja sebab pada titik tersebut terdapat *disk* atau selaput berisi cairan yang berfungsi untuk meredam pergerakan antar ruas pada tulang belakang. Apabila tekanan yang diakibatkan pengangkatan beban kerja melebihi MPL (*Maximum Permissible Limit*) sebagai batas angkat maksimum, maka akan menyebabkan pecahnya *disk* tersebut sehingga pekerja akan mengalami kelumpuhan.

Pada industri pendistribusian beras maupun gula yang dikemas dalam karung, dapat ditentukan massa (berat) beban kerja yaitu maksimal seberat 50 kg. Untuk mengoptimalkan waktu kerja, maka metode kerja yang digunakan adalah dengan cara memanggul beban kerja di atas punggung sehingga terdapat sudut kemiringan punggung.

Input dan rumus-rumus gaya yang digunakan pada sistem ini berdasarkan penelitian yang telah dilakukan Totok Haryanto, mahasiswa Fakultas Teknik Industri UII bersama Hari Purnomo, dosen Fakultas Teknik Industri UII, pada tahun 2002 di Dolog DIY.

1.2 Alasan Pemilihan Judul

Alasan dipilihnya judul “Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika” dalam penelitian ini adalah untuk merepresentasikan algoritma genetika dalam pencarian sudut kemiringan punggung yang optimal pada aktifitas pengangkatan beban karung.

1.3 Rumusan Masalah

Perumusan masalahnya adalah bagaimana membuat sistem yang dapat menentukan sudut-sudut optimum pada pengangkatan awal pemanggulan beban dengan menggunakan algoritma genetika.

1.4 Batasan Masalah

Agar dapat mengarah sesuai dengan pokok permasalahan, maka ruang lingkup permasalahan diberikan beberapa batasan sebagai berikut.

1. Pekerjaan dikhususkan pada pengangkatan *material handling* yang dilakukan secara manual yang dilakukan oleh pekerja pria.
2. Metode pengangkatan beban kerja yang dilakukan dengan metode memanggul berat beban karung maksimal sebesar 50 kg.
3. Variabel pengangkatan meliputi berat beban kerja, berat badan pekerja, serta panjang masing-masing segmen tubuh pekerja yang digunakan untuk gaya kompresi pada ruas tulang belakang.
4. Pengoptimalan sudut hanya terbatas pada sudut pengangkatan awal yang bersifat statis.
5. Kondisi lingkungan kerja diasumsikan dalam keadaan normal.
6. Pekerja diasumsikan berada dalam kondisi fisik dan mental yang baik.
7. Faktor kesehatan dan usia pekerja diabaikan.

1.5 Tujuan Penelitian

Berdasarkan permasalahan-permasalahan yang ada, maka dapat ditentukan tujuan penelitian antara lain sebagai berikut.

1. Melakukan perumusan algoritma genetika untuk mengoptimalkan posisi atau sudut pengangkatan awal beban kerja pada aktivitas *manual material handling*, sehingga dapat meminimalkan cedera otot pada tulang belakang.
2. Menguji kemampuan algoritma genetika untuk menyelesaikan permasalahan optimasi.

1.6 Manfaat Penelitian

1. Memberikan pemahaman mengenai algoritma genetika.
2. Menyediakan suatu perangkat lunak untuk menyelesaikan permasalahan optimasi khususnya dalam mengoptimalkan posisi atau sudut pengangkatan awal beban kerja pada aktivitas *manual material handling*.
3. Algoritma genetika diharapkan dapat menjadi solusi yang optimum dalam menyelesaikan permasalahan optimasi.

1.7 Metodologi Penelitian

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode sekuensial linier, yang memiliki pendekatan sekuensial yang sistematis. Metode sekuensial linier meliputi aktifitas-aktifitas sebagai berikut [PRE97].

1. Tahap Analisis Perangkat Lunak

Pada tahap ini, dilakukan pengumpulan kebutuhan yang diintensifkan dan difokuskan pada sistem yang akan dibangun, meliputi identifikasi domain informasi, tingkah laku sistem, unjuk kerja dan antarmuka sistem.

2. Tahap Desain

Pada tahap ini, difokuskan pada proses desain struktur data, arsitektur sistem, representasi *interface* dan algoritma program.

3. Tahap Penulisan Program (*coding*)

Setelah tahap desain selesai dilakukan, maka hasilnya harus diterjemahkan ke dalam bentuk program komputer yang kemudian menghasilkan suatu sistem.

4. Tahap Pengujian

Tahap pengujian ini dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi pada proses pengkodean. Tahap ini juga berfungsi untuk memastikan bahwa input yang dibatasi akan memberikan hasil yang sesuai dengan kebutuhan.

1.8 Sistematika Penulisan

Sistematika penulisan laporan yang disusun sebagai upaya untuk mempermudah pembacaan yang lebih akurat, maka dibentuklah susunan sebagai berikut.

BAB I PENDAHULUAN

Membahas latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan.

BAB II LANDASAN TEORI

Berisi tentang teori-teori yang berhubungan dengan penelitian, yaitu teori-teori tentang Algoritma Genetika.

BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK

Menguraikan tentang analisis dan kebutuhan perangkat lunak serta hasil analisis mengenai tingkat kebutuhan perangkat lunak tersebut.

BAB IV PERANCANGAN PERANGKAT LUNAK

Menerangkan metode perancangan perangkat lunak yang digunakan serta hasil dari perancangan tersebut sehingga apa yang dirancang benar-benar sesuai dengan apa yang dibutuhkan, meliputi rancangan *flowchart* yang berisi aliran data dari informasi yang ada.

BAB V IMPLEMENTASI PERANGKAT LUNAK

Menjabarkan tentang implementasi perangkat lunak yang meliputi batasan implementasi dan implementasi perangkat lunak serta analisis fungsi yang digunakan.

BAB VI ANALISIS KINERJA PERANGKAT LUNAK

Membahas tentang analisis terhadap kinerja perangkat lunak terhadap sistem yang dibangun yang telah dibuat dan diuji dengan kebutuhan perangkat lunak yang sebenarnya.

BAB VII PENUTUP

Berisi beberapa kesimpulan dari sistem yang dirancang serta saran pengembangan dari keseluruhan tahapan pembuatan tugas akhir serta perangkat lunaknya.



BAB II

LANDASAN TEORI

2.1 Ergonomi

2.1.1 Deskripsi Ergonomi

Istilah ergonomi pertama kali muncul pada tahun 1949 di Inggris. Ergonomi berasal dari dua buah kata dalam bahasa Yunani, yaitu *ergo* yang berarti kerja, dan *nomos* yang berarti aturan, prinsip, atau kaidah. Di beberapa negara, ergonomi dikenal dengan beberapa nama, seperti *Arbeitswissenschaft* di Jerman, *Biotechnology* di negara-negara Skandinavia, serta *Human Engineering* atau *Human Factor Engineering* di Amerika Serikat [SUT79].

Inti dari ergonomi adalah suatu prinsip *human centered design* atau *fitting the task* atau *job the man*, yang artinya suatu pekerjaan harus disesuaikan dengan kemampuan dan keterbatasan yang dimiliki manusia. Ini berarti dalam merancang suatu jenis pekerjaan perlu diperhatikan faktor-faktor yang menjadi kelebihan dan keterbatasan manusia sebagai pelaku kerja.

2.1.2 Penanganan Material Secara Manual (*Manual Material Handling*)

Manual material handling merupakan suatu aktivitas pemindahan benda tanpa bantuan alat. Aktivitas ini sering dikombinasikan terhadap perputaran dan postur pengangkatan yang berbeda dan dapat menjadikan kontribusi terhadap terjadinya cedera.

Aktivitas penanganan material secara manual merupakan aktivitas yang masih banyak ditemui di dalam dunia industri. Meskipun pada beberapa bidang industri yang relatif modern telah banyak menggunakan mesin sebagai alat penanganan material, namun aktivitas *manual material handling* masih diperlukan. Hal tersebut dikarenakan *manual material handling* memiliki beberapa kelebihan antara lain,

- fleksibel dalam gerakan sehingga pemindahan beban dalam ruang terbatas akan lebih efisien;
- untuk beban yang ringan, penanganan material akan lebih murah dibandingkan dengan menggunakan alat bantu berupa mesin;
- tidak semua material dapat dipindahkan dengan alat atau mesin.

Penanganan material secara manual, termasuk pengangkatan beban dengan cara memanggul, apabila tidak dilakukan secara ergonomis akan menimbulkan kecelakaan dalam industri (*industrial accident*) yaitu kerusakan jaringan tubuh yang disebabkan oleh kelebihan beban angkatan.

Faktor yang banyak berpengaruh terhadap timbulnya cedera punggung adalah arah beban yang akan diangkat dan frekuensi aktivitas pemindahan. Beberapa parameter yang harus diperhatikan adalah sebagai berikut [NUR98].

1. Beban yang harus diangkat.
2. Perbandingan antara berat beban dan orangnya.
3. Jarak horizontal dari beban dan orangnya.
4. Ukuran beban yang akan diangkat.

2.1.3 Batasan Beban yang Boleh Diangkat

Berikut ini merupakan pendekatan terhadap batasan dari massa beban yang akan diangkat.

1. Batasan legal (*legal limitations*)
2. Batasan fisiologi (*physiological limitations*)
3. Batasan biomekanika (*biomechanical limitations*)
4. Batasan psiko-fisik (*psycho-physical limitations*)

2.1.3.1 Batasan Fisiologi (*Physiological Limitations*)

Fisiologi adalah ilmu yang mempelajari fungsi organ tubuh manusia yang dipengaruhi oleh tekanan pada otot. Metode pendekatan ini mempertimbangkan rata-rata beban metabolisme dari aktivitas angkat yang berulang (*repetitive lifting*), sebagaimana dapat juga ditentukan dari jumlah konsumsi oksigen. Hal ini harus benar-benar diperhatikan terutama dalam rangka untuk menemukan batasan angkat. Kelelahan kerja yang terjadi akibat aktivitas yang berulang-ulang akan meningkatkan resiko rasa nyeri pada tulang belakang (*back injury*).

2.1.3.2 Batasan Biomekanika (*Biomechanical Limitations*)

Biomekanika merupakan ilmu yang mempelajari tubuh manusia sebagai struktur yang dapat berfungsi mengikuti hukum-hukum mekanika Newton dan juga hukum biologi kehidupan. Biomekanika adalah kombinasi antara ilmu mekanika, antropometri dan dasar-dasar ilmu kedokteran (biologi dan fisiologi).

Pada *manual material handling*, hal yang menjadi perhatian adalah kekuatan kerja otot yang bergantung pada posisi anggota tubuh yang bekerja, arah gerakan kerja, perbedaan kekuatan antar bagian tubuh, usia, kecepatan, ketelitian serta daya tahan jaringan tubuh terhadap beban. Nilai dari analisis biomekanika adalah rentang postur atau posisi aktivitas kerja, ukuran beban dan ukuran manusia yang dievaluasi. Sedangkan kriteria keselamatan didasarkan pada beban tekan (*compression load*) pada tulang belakang.

2.1.4 Mekanika Tubuh Manusia

Mekanika dalam tubuh mengikuti hukum Newton mengenai gerak, kesetimbangan gaya dan kesetimbangan momen. Dalam hukum Newton disebutkan, apabila gaya resultan yang bereaksi pada suatu benda sama dengan nol (benda semula dalam keadaan diam), maka benda tersebut akan tetap diam. Apabila gaya resultan yang bereaksi pada suatu benda tidak sama dengan nol (benda dalam keadaan bergerak), maka benda tersebut akan bergerak dengan kelajuan tetap pada garis lurus.

Suatu benda dikatakan dalam keadaan setimbang apabila gaya eksternal yang bereaksi padanya membentuk sistem gaya ekuivalen dengan nol. Rumus kesetimbangan ditetapkan sebagai berikut.

$$\sum F_x = 0, \sum F_y = 0, \sum M = 0 \dots\dots\dots(2.1)$$

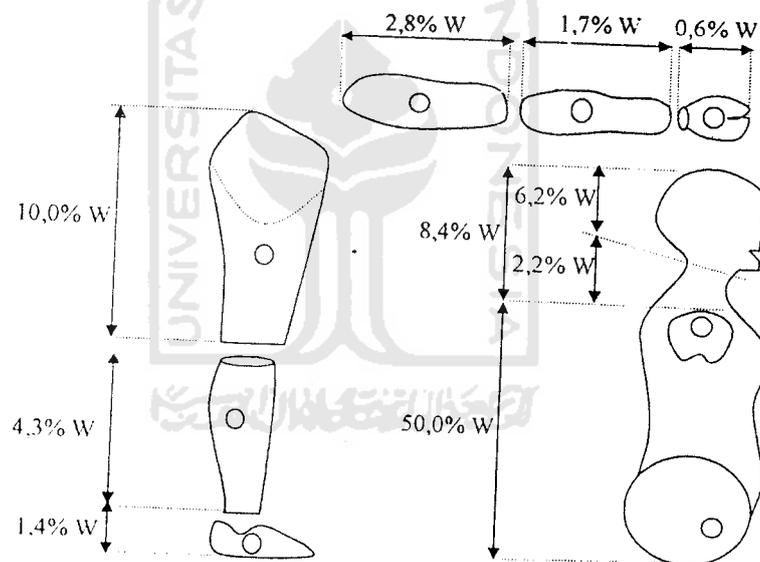
Keterangan:

ΣF_x = gaya horisontal

ΣF_y = gaya vertikal

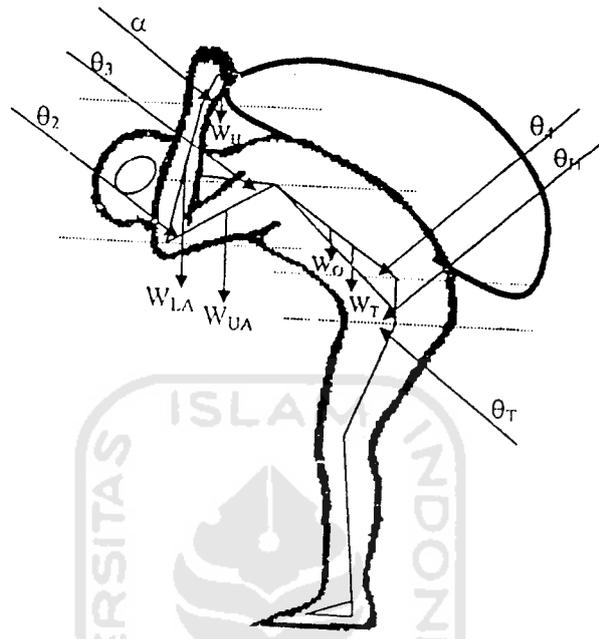
ΣM = momen gaya

Momen dan gaya dapat ditentukan dengan cara menghitung kesetimbangan gaya pada tiap segmen yang menyusun tubuh manusia. Gambar 2.1 di bawah ini menunjukkan besarnya massa yang dikenakan benda pada masing-masing anggota tubuh serta titik sentral di mana anggota tubuh tersebut menimbulkan gaya.



Gambar 2.1 Letak Pusat Massa Tiap Segmen Tubuh [WIN90]

Gambar di bawah ini menunjukkan proyeksi sudut, momen gaya serta gaya yang bekerja pada aktivitas pengangkatan beban dengan cara memanggul.



Gambar 2.2 Proyeksi Sudut, Momen Gaya dan Gaya

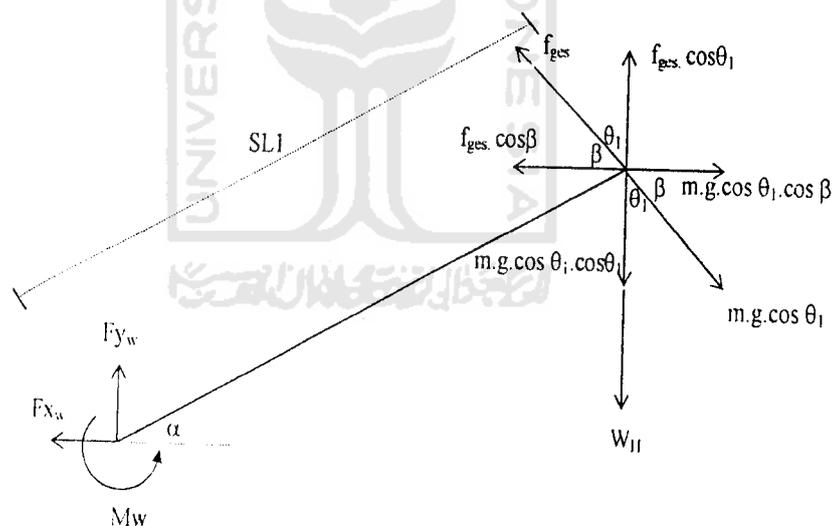
Keterangan:

- α = sudut pergelangan tangan dengan bidang horisontal
- θ_2 = sudut antara siku dengan bidang horisontal
- θ_3 = sudut antara bahu dengan bidang horisontal
- θ_H = sudut inklinasi badan terhadap bidang horisontal
- θ_T = sudut inklinasi kaki relatif terhadap bidang horisontal
- θ_4 = sudut antara punggung dengan bidang horisontal = $\theta_H - 5$

- W_{LA} = berat lengan bawah
 W_{UA} = berat lengan bawah
 W_0 = gaya normal beban
 W_t = berat segmen punggung

Dalam melakukan aktivitas pengangkatan beban karung dengan cara memanggul beban, bagian-bagian tubuh yang bekerja antara lain telapak tangan, lengan bawah, lengan atas, dan punggung. Gambar-gambar di bawah ini menunjukkan besarnya gaya serta momen gaya yang bekerja pada bagian-bagian tubuh tersebut.

1. Telapak tangan



Gambar 2.3 Proyeksi Sudut, Momen Gaya dan Gaya Pada Telapak Tangan

$$\Sigma F_x = 0, \Sigma F_y = 0, \Sigma M = 0$$

$$W_H = 0,6\% \times W_{\text{badan}}$$

$$F_{xw} = (m \cdot g \cdot \cos \theta_1 \cdot \cos \beta - f_{\text{ges}} \cdot \cos \beta) : 2$$

$$F_{yw} = W_H + [(m \cdot g \cdot \cos \theta_1 \cdot \cos \theta_1 - f_{\text{ges}} \cdot \cos \theta_1) : 2]$$

$$M_W = [W_H + (m \cdot g \cdot \cos \theta_1 \cdot \cos \theta_1 - f_{\text{ges}} \cdot \cos \theta_1) : 2] SL_1 \cdot \cos \alpha - (m \cdot g \cdot \cos \theta_1 \cdot \cos \beta - f_{\text{ges}} \cdot \cos \beta) SL_1 \cdot \sin \alpha$$

Keterangan:

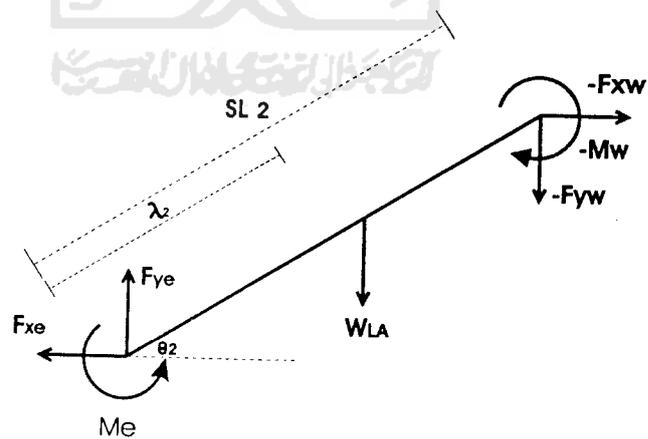
W_H = berat telapak tangan

F_{xw} = gaya pada genggaman telapak tangan sejajar sumbu x

F_{yw} = gaya pada genggaman telapak tangan sejajar sumbu y

M_W = momen gaya pada telapak tangan

2. Lengan bawah



Gambar 2.4 Proyeksi Sudut, Momen Gaya dan Gaya Pada Lengan Bawah

$$\Sigma F_x = 0, \Sigma F_y = 0, \Sigma M = 0$$

$$\lambda_2 = 43\%$$

$$W_{LA} = 1,7\% \cdot W_{\text{badan}}$$

$$F_{xe} = F_{xw}$$

$$F_{ye} = F_{yw} + W_{LA}$$

$$M_e = M_w + (W_{LA} \cdot \lambda_2 \cdot SL_2 \cdot \cos \theta_2) + (F_{yw} \cdot SL_2 \cdot \cos \theta_2)$$

Keterangan:

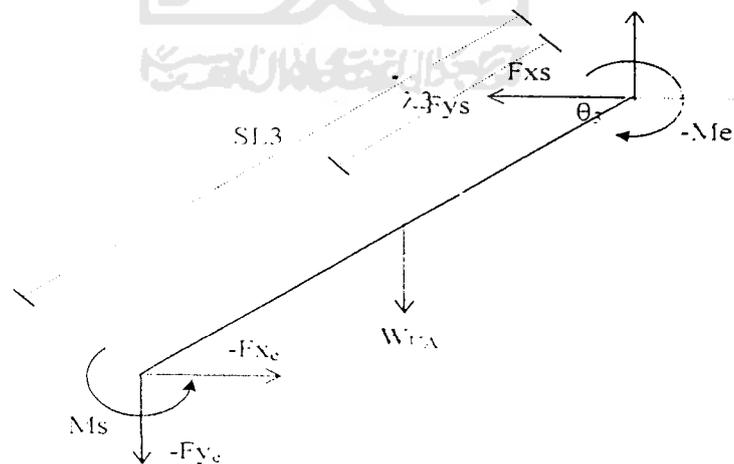
W_{LA} = berat lengan bawah

F_{xe} = gaya pada lengan bawah tangen sejajar sumbu x

F_{ye} = gaya pada lengan bawah sejajar sumbu y

M_w = momen gaya pada lengan bawah

3. Lengan atas



Gambar 2.5 Proyeksi Sudut, Momen Gaya dan Gaya Pada Lengan Atas

$$\Sigma F_x = 0, \Sigma F_y = 0, \Sigma M = 0$$

$$\lambda_3 = 43,6\%$$

$$W_{UA} = 2,8\% \times W_{\text{badan}}$$

$$F_{xs} = F_{xe}$$

$$F_{ys} = F_{ye} + W_{UA}$$

$$M_s = M_e + (W_{UA} \cdot \lambda_3 \cdot SL_3 \cdot \cos \theta_3) + (F_{ye} \cdot SL_3 \cdot \cos \theta_3)$$

Keterangan:

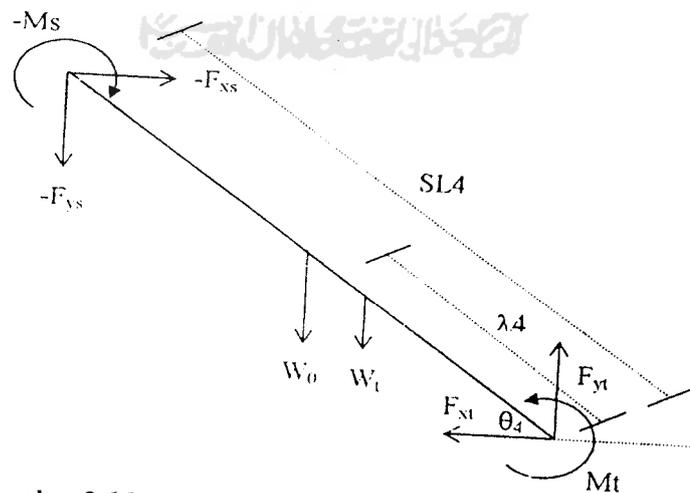
W_{UA} = berat lengan bawah

F_{xs} = gaya pada lengan atas sejajar sumbu x

F_{ys} = gaya pada lengan atas sejajar sumbu y

M_s = momen gaya pada lengan atas

4. Link punggung



Gambar 2.6 Proyeksi Sudut, Momen Gaya dan Gaya Pada Punggung

$$\Sigma F_y = 0, \Sigma F_x = 0, \Sigma M = 0$$

$$\lambda_4 = 67\%$$

$$W_t = 45\% \times W_{\text{badan}}$$

$$F_{xt} = 2F_{xs}$$

$$F_{yt} = 2F_{ys} + W_0 + W_t$$

$$M_t = 2M_s + (W_0 \cdot 0,5 \cdot SL_4 \cdot \cos \theta_4) + (W_t \cdot \lambda_4 \cdot SL_4 \cdot \cos \theta_4) + (2F_{ys} \cdot SL_4 \cdot \cos \theta_4)$$

Keterangan:

W_t = berat segmen punggung

F_{xt} = gaya pada punggung sejajar sumbu x

F_{yt} = gaya pada punggung sejajar sumbu y

M_t = momen gaya pada punggung

Dengan menggunakan teknik perhitungan keseimbangan gaya pada setiap segmen tubuh manusia, maka didapat momen resultan pada tulang belakang. Kemudian untuk mencapai keseimbangan tubuh pada aktivitas pengangkatan, momen pada tulang belakang tersebut harus diimbangi dengan gaya otot *spinal erector* (F_m) yang cukup besar, yang berfungsi untuk mempertahankan kesetimbangan statis. Gaya otot *spinal erector* dirumuskan sebagai berikut.

$$F_m = \frac{M_t - F_A \cdot D}{E} \dots\dots\dots(2.2)$$

Keterangan:

F_m = gaya otot *spinal erector* (Newton)

E = panjang lengan momen otot *spinal erector* = 0,05 m

M_t = momen resultan pada L5/S1

F_A = gaya perut (Newton)

D = jarak dari gaya perut ke L5/S1 = 0,11 m

Untuk mencari gaya perut (F_A), maka perlu dicari tekanan perut (P_A).

$$P_A = \frac{10^{-4} [43 = 0,36(\theta_H + \theta_T)] [M_t]^1,8}{75} \dots\dots\dots(2.3)$$

$$F_A = P_A \times AA \dots\dots\dots(2.4)$$

Keterangan:

P_A = tekanan perut (N/m^2)

F_A = gaya perut (Newton)

$AA = 0,0465 \text{ m}^2$

Berikut ini merupakan rumus untuk menghitung berat total segmen tubuh manusia.

$$W_{tot} = W_0 + 2W_H + 2W_{LA} + 2W_{UA} + W_t \dots\dots\dots(2.5)$$

Keterangan:

W_{tot} = berat total segmen tubuh manusia (Newton)

Kemudian gaya tekan atau gaya kompresi pada tulang belakang didapatkan dari rumus berikut.

$$F_c = W_{tot} \cdot \cos \theta_4 - F_A + F_m \dots\dots\dots(2.6)$$

Keterangan:

F_c = gaya kompresi (Newton)

W_{tot} = berat total segmen tubuh manusia (Newton)

F_A = gaya perut (Newton)

F_m = gaya otot *spinal erector* (Newton)

Sedangkan besarnya gaya kompresi pada tulang belakang (F_c) maksimum adalah 6370 Newton. Batas maksimum tersebut merupakan batas yang dapat ditolerir agar tidak terjadi kerusakan pada tulang belakang.

2.2 Algoritma Genetika

2.2.1 Deskripsi Algoritma Genetika

Algoritma Genetika merupakan suatu metode optimasi dan simulasi yang perkembangannya tidak bisa lepas dari paradigma evolusi yang ditunjukkan pertama kali oleh Charles Darwin (1809-1882) yang diteruskan dalam perkembangan teori genetika, walaupun pada kenyataannya teori tersebut terbukti keliru. Evolusi Darwinian yang berbasis pada konsep "*survival of the fittest*" menyatakan bahwa evolusi jenis-jenis spesies makhluk hidup dan ekosistemnya terjadi karena seleksi alam. Semakin adaptif sifat dari suatu individu dalam ekosistem maka kemungkinan untuk bertahan hidup dan menghasilkan keturunan semakin besar. Keturunan dari spesies yang bertahan hidup akan mewarisi sifat-sifat dari induknya dengan kemungkinan terjadi perubahan beberapa sifat antara induk dan keturunannya melalui perkawinan dan mutasi genetika.

Algoritma genetika merupakan teknik pencarian nilai optimum *stochastic* menggunakan gabungan dari mekanisme seleksi alam dengan teori genetika. Algoritma genetika berbeda dengan teknik konvergensi konvensional yang lebih bersifat deterministik [GEN97].

Algoritma genetika mempunyai perhitungan yang relatif sederhana, namun sangat kuat pada proses pencarian untuk melakukan peningkatan. Algoritma genetika juga tidak dibatasi oleh asumsi pembatas mengenai ruang pencarian, seperti asumsi mengenai kekontinuan, keberadaan turunan, dan sebagainya. Kekuatan algoritma genetika terletak pada beberapa hal yang membedakan algoritma genetika dengan algoritma pencarian lainnya [GOL89]. Letak perbedaan tersebut antara lain sebagai berikut.

1. Algoritma genetika bekerja dengan memanipulasi kode-kode set parameter, bukan dengan memanipulasi nilai parameter itu sendiri.
2. Algoritma genetika bebas untuk mengkodekan masalah dengan berbagai cara sehingga algoritma genetika tidak dibatasi oleh batasan metode lainnya.
3. Algoritma genetika bekerja dengan populasi titik, bukan hanya pada satu titik.
4. Algoritma genetika menggunakan informasi fungsi tujuan, bukan menggunakan informasi turunan atau yang lainnya.
5. Algoritma genetika menggunakan aturan perpindahan probabilistik, bukan deterministik.

2.2.2 Sejarah Algoritma Genetika

Ide tentang komputasi evolusioner pertama kali diperkenalkan oleh I. Rechenberg dalam karyanya yang berjudul “*Evolutionsstrategie*”, atau dalam versi bahasa Inggris yang berjudul “*Evolution Strategies*”. Ide tersebut kemudian dikembangkan oleh berbagai penelitian lain. Selanjutnya, algoritma genetika pertama kali dikemukakan oleh John Holland dan dikembangkan bersama dengan rekan-rekan serta murid-muridnya di Universitas Michigan. Penelitian tersebut menghasilkan dua buah kesimpulan sebagai berikut.

1. Meningkatkan pemahaman tentang proses adaptasi alami.
2. Mendesain perangkat lunak dari sistem tiruan (*artificial software systems*) yang mempunyai persamaan dengan sistem alami (*natural systems*), namun tetap mempertahankan mekanisme penting dari sistem alami tersebut.

John Holland memulai penelitiannya pada awal tahun 1960. Kemudian pada tahun 1975, John Holland mempublikasikan sebuah buku yang berjudul “*Adaptation in Natural and Artificial Systems*”. Holland tidak begitu tertarik pada permasalahan optimasi, namun ia memiliki ketertarikan yang besar terhadap proses adaptasi. Ia menekankan akan pentingnya penggabungan ulang (*recombination*) dalam suatu populasi yang besar. Dalam bukunya tersebut, Holland [HOL75] mengatakan “*solution to a problem solved by genetic algorithms is evolved*”.

2.2.3 Latar Belakang Biologi

Algoritma genetika meniru beberapa proses yang ditemukan pada evolusi alamiah. Evolusi terjadi pada kromosom, alat organik yang mengkodekan struktur makhluk hidup. Sebagian makhluk hidup terbentuk melalui proses pengkodean kromosom. Individu-individu yang kuat akan berhasil bertahan hidup, sedangkan individu-individu yang lemah akan punah. Individu-individu yang bertahan hidup akan membentuk individu-individu baru sehingga terjadi proses regenerasi. Berikut ini merupakan teori-teori dasar evolusi yang diadopsi oleh algoritma genetika.

1. Evolusi merupakan proses yang bekerja pada kromosom, bukan pada makhluk hidup yang dikodekannya.
2. Seleksi alamiah merupakan hubungan antara kromosom dan kinerja dari struktur yang dikodekannya. Proses seleksi alamiah menyebabkan kromosom yang mengkodekan struktur yang sesuai lebih banyak diproduksi dibandingkan dengan yang tidak sesuai.
3. Proses reproduksi merupakan titik saat terjadi evolusi. Mutasi dapat menyebabkan kromosom anak berbeda dengan induknya. Sedangkan rekombinasi akan menciptakan kromosom yang jauh berbeda, dengan cara mengkombinasikan bagian kromosom dari kedua induknya.
4. Evolusi biologis tidak mempunyai memori. Apapun yang berhubungan dengan fungsi individu dengan lingkungannya terdapat dalam gen (sekumpulan kromosom yang dibawa individu) dan struktur yang dikodekannya.

Teori dasar evolusi di atas, apabila secara sempurna digabungkan dalam sebuah algoritma pemrograman komputer mampu menghasilkan teknik pencarian untuk memecahkan permasalahan yang sulit dengan cara yang sama dengan yang dilakukan oleh alam melalui evolusi.

Berikut ini istilah-istilah yang diadopsi oleh algoritma genetika dari istilah-istilah yang terdapat dalam sistem biologi.

1. Kromosom dengan *string*

Kromosom merupakan tempat penyimpanan informasi genetika. Informasi yang disimpan merupakan representasi dari masalah yang dihadapi. Di dalam algoritma genetika, *string* dianalogikan sebagai kromosom.

2. *Genotype* dengan struktur

Kombinasi satu atau beberapa kromosom membentuk fungsi kerja suatu organisme. Interaksi sekumpulan tersebut disebut sebagai *genotype*. Di dalam algoritma genetika, struktur dianalogikan dengan *genotype*

3. *Phenotype* dengan set parameter (alternatif solusi)

Interaksi di dalam struktur terjadi karena adanya proses transformasi kode-kode genetika. Proses ini disebut *phenotype*. Jadi, *phenotype* merupakan representasi set parameter masalah yang dihadapi. Representasi kode dapat berupa numerik atau non numerik.

4. Gen dengan *feature* (karakter)

Suatu kromosom dibentuk oleh beberapa gen. Di dalam algoritma genetika, *string* dibentuk oleh beberapa *feature*.

5. *Alleles* dengan *feature value*

Suatu *feature* memiliki nilai tertentu. Di dalam sistem biologi, nilai *feature* disebut dengan *alleles*.

6. *Locus* dengan *positions*

Letak gen di dalam kromosom disebut *locus*. Setiap *feature* memiliki urutan posisi di dalam *string*.

7. *Fitness function* dengan fungsi tujuan

Setiap kromosom memiliki fungsi sesuaian (*fitness*) berdasarkan nilai dan letak gen. Fungsi *fitness* merupakan fungsi tujuan yang digunakan dalam proses evaluasi permasalahan yang dihadapi.

2.2.4 Mekanisme Kerja Algoritma Genetika

Proses algoritma genetika dalam mencari suatu nilai optimalisasi (maksimum atau minimum) adalah sebagai berikut.

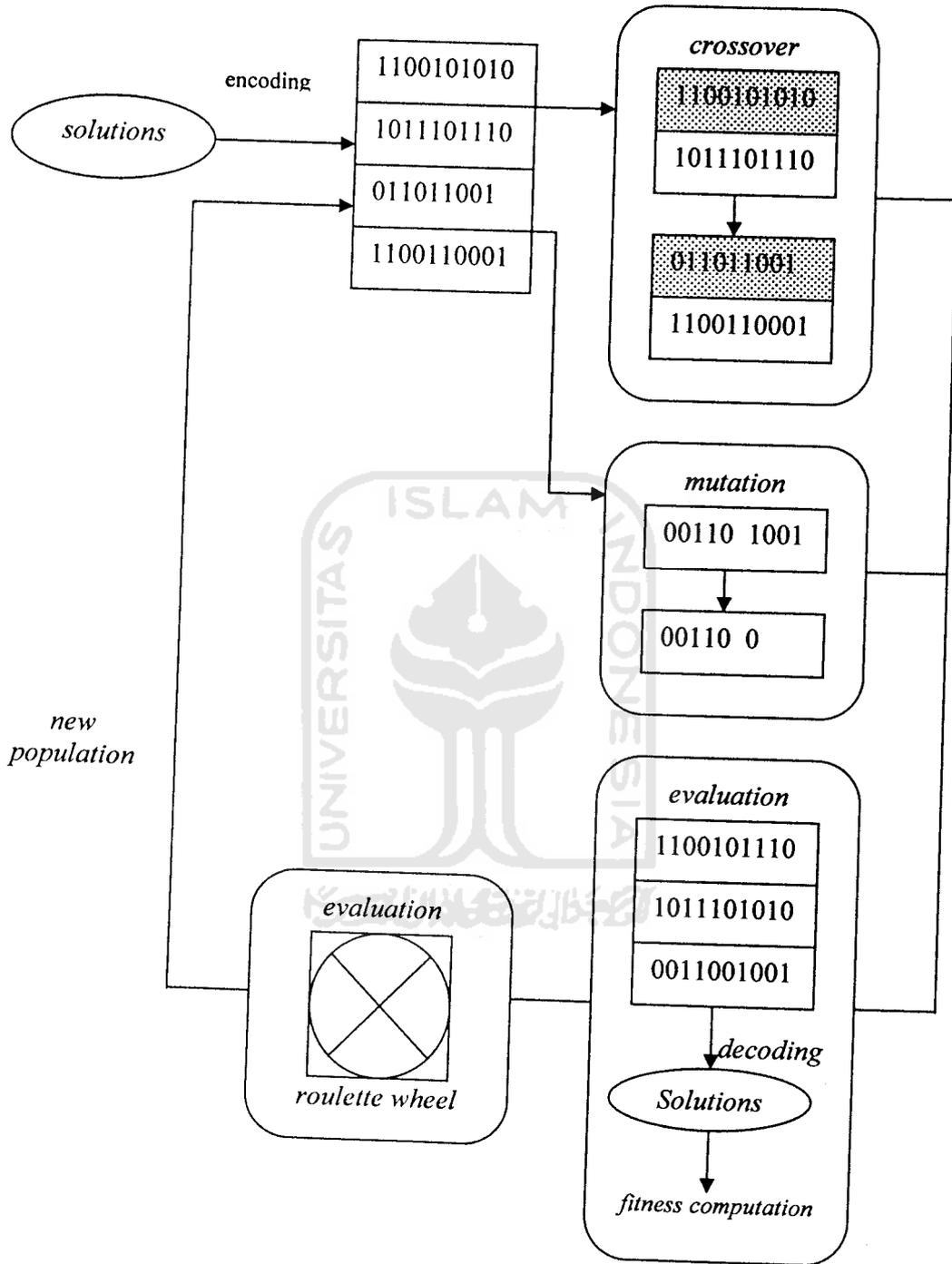
1. Penentuan model dari sistem buatan dengan mendefinisikan spesies-spesies dengan struktur gen dan kromosom yang ditentukan berdasarkan sifat-sifatnya.
2. Perhitungan nilai *fitness* dari setiap spesies berdasarkan struktur gennya. Nilai ini digunakan sebagai patokan optimal atau tidaknya suatu nilai.
3. Pemilihan induk yaitu individu-individu dengan nilai *fitness* terbaik untuk dijadikan induk dalam menghasilkan individu-individu baru.

4. Proses reproduksi terdiri dari *crossover* (perkawinan silang) dan mutasi. Dari reproduksi ini akan dihasilkan individu-individu terbaru.

Proses di atas secara dilakukan secara berulang-ulang, sehingga diharapkan induk yang baik akan diperoleh generasi anak dengan spesies yang lebih baik. Dengan demikian, algoritma genetika melakukan simulasi evolusi pada suatu populasi kromosom. Seperti kromosom, algoritma ini menyelesaikan masalah pencarian kromosom yang baik, dengan memanipulasi bagian yang terdapat pada kromosom secara acak (*random*).

Algoritma genetika tidak mengetahui tipe permasalahan yang dipecahkan. Informasi yang digunakan hanyalah evaluasi dari nilai *fitness* setiap kromosom yang dihasilkan. Satu-satunya fungsi dari informasi ini adalah untuk mempengaruhi pemilihan kromosom sehingga kromosom yang mempunyai evaluasi tinggi akan diproduksi lebih banyak dibandingkan dengan kromosom yang mempunyai evaluasi rendah. Algoritma ini secara efektif menggunakan informasi masa lalu untuk menuju titik pencarian baru dengan kinerja yang diharapkan akan meningkat.

Struktur algoritma genetika digambarkan oleh Mitsuo Gen [GEN97] sebagai berikut.



Gambar 2.7 Struktur Algoritma Genetika Menurut Misuo Gen

2.2.5 Parameter Algoritma Genetika

Yang dimaksud dengan parameter di sini adalah parameter kontrol algoritma genetika, yaitu ukuran populasi (*popsi*), peluang *crossover* (p_c), dan peluang mutasi (p_m). Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan. Ada beberapa rekomendasi yang bisa digunakan antara lain [KUS03]

- untuk permasalahan yang memiliki kawasan solusi yang cukup besar, De Jong merekomendasikan untuk nilai parameter kontrol yaitu

$$(popsi; p_c; p_m) = (50; 0,6; 0,001)$$

- bila rata-rata *fitness* setiap generasi digunakan sebagai indikator, maka Grefenstette merekomendasikan

$$(popsi; p_c; p_m) = (30; 0,95; 0,01)$$

- bila *fitness* dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah

$$(popsi; p_c; p_m) = (80; 0,45; 0,01)$$

- ukuran populasi sebaiknya tidak lebih kecil dari 30 untuk sembarang jenis permasalahan.

2.2.6 Prosedur Inisialisasi

Kodifikasi atau pengkodean kromosom merupakan masalah yang penting dalam algoritma genetika. Pada awal algoritma genetika diperkenalkan oleh Holland, kromosom ditampilkan dalam bentuk biner. Namun pada sebagian besar aplikasi

algoritma genetika, permasalahan yang akan diselesaikan akan sulit diubah ke dalam bentuk tersebut, sebab kode biner bukan merupakan kode alami. Maka beberapa macam kode *non string* dikembangkan seperti

- penggunaan kode riil (*real number coding*) untuk permasalahan optimalisasi;
- penggunaan kode bilangan bulat (*integer number coding*) untuk permasalahan optimalisasi kombinatorial.

2.2.7 Representasi Kromosom

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, selanjutnya dilakukan representasi kromosom yang terdapat pada populasi tersebut. Representasi kromosom ini dapat disebut juga dengan inisialisasi kromosom. Inisialisasi dilakukan secara acak, namun harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

Pada kasus ini, hal pertama yang dilakukan adalah *encoding* variabel keputusan ke dalam bentuk deretan bilangan biner. Domain dari variabel sudut x_j adalah $[a_j, b_j]$ dan tingkat ketelitian yang dikehendaki adalah 4 angka di belakang desimal, maka range domain dari masing-masing variabel sudut harus dibagi paling sedikit berukuran $(b_j - a_j) \times 10^4$ range. Bits yang dikehendaki dari variabel sudut adalah sebagai berikut.

$$2^{m_{ji}-1} < (b_j - a_j) \times 10^4 \leq 2^{m_{ji}} \dots\dots\dots(2.7)$$

$$m = m_{j_1} + m_{j_{(i+1)}} \dots\dots\dots(2.8)$$

dengan:

m_{j_i} = jumlah bits dalam kromosom pada masing-masing batasan (jika terdapat lebih dari satu batasan)

m = jumlah total bits dalam kromosom yang dikehendaki

b_j = batas atas variabel sudut

a_j = batas bawah variabel sudut

2.2.8 Fungsi Evaluasi

Fungsi evaluasi pada masalah optimasi pada umumnya merupakan fungsi tujuan. Jadi fungsi evaluasi berperan sebagai alat ukur apakah *string* tersebut merupakan solusi yang baik atau bukan. Begitu juga dengan lingkungan yang menentukan apakah suatu makhluk hidup akan bertahan hidup atau mati. Nilai fungsi evaluasi biasanya disebut dengan nilai kesesuaian (*fitness value*).

Kasus ini menggunakan metode seleksi roda roulette dan dapat ditentukan berdasarkan faktor-faktor di bawah ini.

1. Dilakukan perhitungan nilai *fitness* evaluasi pada masing-masing kromosom

$$eval(v_k) = f(x) \dots\dots\dots(2.9)$$

dengan:

$eval(v_k)$ = nilai *fitness* pada tiap kromosom yang merepresentasikan
besarnya sudut pengangkatan

$k = 1, 2, \dots$, ukuran populasi

2. Dilakukan perhitungan nilai total *fitness* pada populasi

$$F = \sum_{k=1}^{pop\ size} eval(v_k) \dots\dots\dots(2.10)$$

dengan:

F = nilai total *fitness*

$k = 1, 2, \dots$, ukuran populasi

3. Dilakukan perhitungan probabilitas relatif (p_k) untuk setiap kromosom v_k

$$p_k = \frac{eval(v_k)}{F}, \dots\dots\dots(2.11)$$

dengan:

$k = 1, 2, \dots$, ukuran populasi

4. Dilakukan perhitungan probabilitas komulatif (q_k) untuk setiap kromosom v_k

$$q_k = \sum_{j=1}^k p_j, \dots\dots\dots(2.12)$$

dengan:

$k = 1, 2, \dots$, ukuran populasi

Berikut ini merupakan 3 langkah proses evaluasi tingkat kemampuan atau performansi dari kromosom.

1. *Genotype* kromosom diubah menjadi *phenotype* kromosom, dengan kata lain deretan bilangan biner diubah menjadi deretan bilangan real.

$$x^k = (x^k_1, x^k_2) \dots \dots \dots (2.12)$$

dengan:

$$k = 1, 2, \dots, \text{ukuran populasi}$$

Perubahan dari deretan bilangan biner menjadi angka real untuk variabel sudut x_j berdasarkan

$$x_j = a_j + \text{decimal}(\text{substring}_j) \times \frac{b_j - a_j}{2^{m_j} - 1} \dots \dots \dots (2.13)$$

dengan: *decimal (substring_j)* merepresentasikan nilai desimal dari *substring_j* untuk variabel sudut x_j

2. Fungsi obyektif $f(x^k)$ dievaluasi.
3. Nilai fungsi obyektif diubah ke dalam *fitness*. Untuk kasus minimasi, nilai *fitness* dapat ditentukan berdasarkan

$$\text{eval}(v_k) = f(x^k) \dots \dots \dots (2.14)$$

dengan:

$$k = 1, 2, \dots, \text{ukuran populasi}$$

Selain itu, diperlukan suatu penskalaan fungsi *fitness* pada setiap kromosom untuk menjaga jangan sampai super kromosom mendominasi populasi yang dapat mengakibatkan algoritma genetika tergiring dalam konvergensi dini sehingga tidak dapat menghasilkan solusi yang optimal. Persamaan penskalaan yang digunakan adalah

$$eval(v_k) = C - \sum_{i=1}^8 Fm_i \dots\dots\dots(2.15)$$

Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Apabila ternyata fungsi objektif memiliki nilai negatif, maka ditambahkan konstanta *C* agar nilai *fitness* yang terbentuk tidak menjadi negatif [KUS03]. Sehingga persamaannya menjadi

$$eval(v_k) = C + \sum_{i=1}^8 Fm_i \dots\dots\dots(2.16)$$

dengan:

C = konstanta = 10^6

Fm_i = nilai total gaya otot *spinal erector* yang dihasilkan

2.2.9 Seleksi

Pemilihan induk atau seleksi induk dilakukan agar kromosom yang mempunyai nilai *fitness* paling tinggi mempunyai peluang reproduksi yang lebih tinggi pula. Proses seleksi akan menentukan individu-individu yang akan dipilih dalam proses rekombinasi. Selain itu, proses seleksi juga akan menentukan bagaimana *offspring* terbentuk dari individu-individu terpilih tersebut. Maka, langkah pertama yang dilakukan dalam proses seleksi ini adalah pencarian nilai *fitness*. Nilai *fitness* dapat ditentukan dengan suatu fungsi tujuan atau dengan suatu penilaian subyektif. Nilai *fitness* inilah yang akan digunakan pada tahap-tahap seleksi berikutnya.

2.2.9.1 Seleksi Roda Roulette (*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.

2.2.10 Crossover

Crossover atau perkawinan silang merupakan suatu proses penukaran gen dari dua individu induk untuk memperoleh dua individu anak. *Crossover* berfungsi menggabungkan dua *string* induk yang berbeda menjadi dua *string* anak yang berbeda dengan *string* induknya. Hal ini dilakukan dengan harapan agar diperoleh individu-individu anak dengan struktur gen yang lebih baik.

Dalam kasus ini, metode *crossover* yang digunakan adalah perkawinan silang satu titik (*one-cut-point crossover*) dengan ciri-ciri sebagai berikut.

- Memilih *site* secara random dari *parent* pertama.
- Isi di sebelah kanan *site* pada *parent* pertama ditukar dengan *parent* kedua untuk menghasilkan *offspring* [GEN97].

Berikut ini merupakan ilustrasi *one-cut-point crossover*.

Parent 1	0	1	1	0	1	0	1	1
Parent 2	1	1	0	0	0	1	1	1
Offspring 1	0	1	1	0	0	1	1	1
Offspring 2	1	1	0	0	1	0	1	1

Gambar 2.8 Ilustrasi *one-cut-point crossover*

2.2.11 Mutasi (*Mutation*)

Operasi mutasi melakukan perubahan nilai gen-gen pada kromosom dengan cara menyisipkan kode informasi ke dalam suatu individu. Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Selain itu, mutasi juga berfungsi untuk menyediakan gen yang tidak ada dalam populasi awal, sehingga mutasi akan meningkatkan variasi populasi.

Proses mutasi yang terlalu sering akan menghasilkan individu yang buruk. Hal tersebut dikarenakan untaian kromosom individu superior akan rusak oleh proses mutasi, sehingga kemungkinan munculnya individu superior akan berjalan lambat atau bahkan tidak muncul sama sekali. Maka probabilitas populasi yang digunakan harus dipertimbangkan. Mengenai laju mutasi, ada yang berpendapat bahwa laju mutasi sebesar $1/n$ akan memberikan hasil yang cukup baik. Namun ada pula yang berpendapat bahwa laju mutasi tidak bergantung pada ukuran populasinya. Kromosom hasil mutasi harus diperiksa, apakah masih berada pada domain solusi, dan bila perlu bisa dilakukan perbaikan.

Mutasi dibedakan berdasarkan bilangan yang digunakan, antara lain sebagai berikut [KUS03].

1. Mutasi bilangan real

Pada mutasi bilangan real, ukuran mutasi biasanya sangat sulit ditentukan.

Ukuran yang kecil biasanya sering mengalami kesuksesan, namun ada kalanya ukuran yang lebih besar akan berjalan lebih cepat.

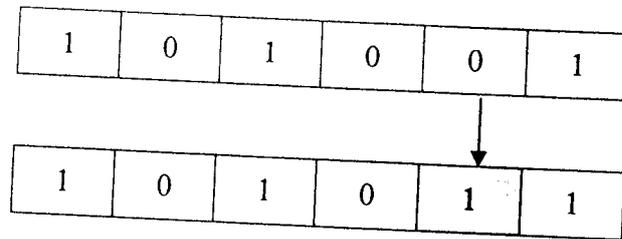
Operasi mutasi untuk bilangan real dapat ditetapkan adalah

- variabel yang dimutasi = variabel \pm range*delta; (+ atau - memiliki probabilitas yang sama);
- range = 0.5 * dominan variabel; (interval pencarian);
- delta = $\sum(a_i * 2^{-i})$; $a_i = 1$ dengan probabilitas $1/m$, selain itu $a_i = 0$, dengan $m = 20$.

2. Mutasi biner

Kasus ini menggunakan mutasi biner. Cara yang sederhana untuk mendapatkan mutasi biner adalah dengan mengganti satu atau beberapa nilai gen dari kromosom. Langkah-langkah mutasi biner adalah

- dilakukan perhitungan jumlah gen pada populasi (panjang kromosom dikalikan dengan ukuran populasi);
- gen yang akan dimutasi dipilih secara acak atau random;
- kromosom dari gen yang terpilih ditentukan untuk dimutasi;
- nilai gen (0 ke 1, atau 1 ke 0) diganti dari kromosom yang akan dimutasi tersebut.



Gambar 2.9 *Binary Mutation*

2.2.12 Pembaharuan Generasi

Crossover dan mutasi merupakan proses pembaharuan generasi yang dilakukan untuk mengganti populasi lama dengan populasi baru hasil seleksi. *Crossover* dan mutasi dilakukan dengan harapan bahwa populasi baru tersebut akan mempunyai nilai *fitness* yang lebih baik daripada populasi lama. Terdapat tiga jenis pembaharuan generasi, antara lain sebagai berikut.

1. *Generational update*, apabila pembaharuan dilakukan dengan menggantikan secara keseluruhan populasi lama dengan populasi baru.
2. *Continous update*, apabila sebagian kromosom pada populasi lama diganti dengan kromosom-kromosom baru.
3. *Steady state update*, apabila pendekatan selektif dilakukan cenderung mempertahankan kromosom yang baik dari generasi lama ke generasi baru dan hanya menggantikan kromosom yang nilai *fitness*-nya kurang baik.

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

3.1 Metode Analisis

Sistem yang akan digunakan dalam menganalisis kebutuhan perangkat lunak pada Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika ini adalah menggunakan metode analisis terstruktur, pada metode transformasi input, proses dan output dinyatakan dengan *flowchart*. Pada tahap ini, penggunaan notasi-notasi digunakan untuk menggambarkan arus data sistem, dimana akan sangat membantu dalam proses komunikasi dengan *user*.

3.2 Hasil Analisis

Berdasarkan analisis yang telah dilakukan, maka dapat diketahui apa saja yang menjadi masukan sistem (*input*), keluaran sistem fungsi (*output*) atau metode yang digunakan oleh sistem, kebutuhan perangkat keras maupun perangkat lunak serta antarmuka sistem (*interface*) yang akan dibuat, sehingga sistem yang dibuat akan sesuai dengan yang diharapkan.

3.2.1 Input

Input untuk permasalahan optimasi pada Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika ini berupa

variabel-variabel yang digunakan dalam suatu masalah optimasi serta penggunaan algoritma genetika.

1. Kebutuhan input data awal

- a. Panjang telapak tangan (SL1)
- b. Panjang lengan bawah (SL2)
- c. Panjang lengan atas (SL3)
- d. Panjang tulang belakang (SL4)
- e. Sudut pergelangan tangan dengan bidang horizontal (α)
- f. Sudut kemiringan beban terhadap bidang vertikal (θ_1)
- g. Sudut kemiringan beban terhadap bidang horizontal (β)
- h. Sudut antara siku dengan bidang horizontal (θ_2)
- i. Sudut antara bahu dengan bidang horizontal (θ_3)
- j. Massa beban angkat
- k. Massa badan pekerja

2. Kebutuhan input bagi proses algoritma genetika adalah sebagai berikut:

- a. Jumlah Populasi

Variabel ini digunakan untuk menentukan seberapa banyak jumlah populasi yang akan di-*generate* dalam suatu proses algoritma genetika.

b. Jumlah Generasi

Variabel ini digunakan untuk menentukan seberapa banyak operasi genetika, baik seleksi, *crossover* maupun mutasi yang akan dilakukan dari suatu representasi permasalahan dalam kromosom yang akan dicari solusinya.

c. Probabilitas *Crossover*

Variabel ini digunakan untuk menentukan seberapa banyak kromosom dari populasi yang akan diikutsertakan dalam proses *crossover*.

d. Probabilitas Mutasi

Variabel ini digunakan untuk menentukan seberapa banyak kromosom dari populasi yang telah mengalami proses *crossover* yang akan diikutsertakan dalam proses mutasi.

3.2.2 Output

Output atau hasil keluaran yang diharapkan dari sistem yang akan dibuat adalah sebagai berikut.

1. Sudut optimal antara punggung dengan bidang horisontal (θ_4).
2. Sudut inklinasi badan relatif terhadap bidang horisontal (θ_H).
3. Sudut inklinasi kaki relatif terhadap bidang horisontal (θ_T).
4. Gaya kompresi (F_c).

3.2.3 Kebutuhan Proses

Proses-proses yang dibutuhkan dalam menyelesaikan permasalahan optimasi sudut adalah sebagai berikut.

1. Proses perhitungan gaya dan momen gaya

Setelah dilakukan masukan data awal, maka dilakukan perhitungan besarnya gaya dan momen gaya yang bekerja pada telapak tangan, lengan bawah, lengan atas, dan tulang belakang untuk output akhir pada proses selanjutnya.

2. Proses algoritma genetika

a. Proses seleksi

Proses seleksi yang menggunakan metode seleksi roda rolet merupakan suatu proses pencarian nilai *fitness* yang didapatkan dengan meminimumkan nilai gaya otot *spinal erector* (F_m).

b. Proses *crossover*

c. Proses mutasi

3.2.4 Antarmuka Yang Diinginkan

Suatu sistem dapat dengan mudah digunakan oleh *user* tanpa mengalami kesulitan. Selain itu *user* tidak memerlukan keahlian khusus atau mempelajarinya terlebih dahulu untuk dapat menggunakan sistem ini. Oleh karena itu pada perancangan antarmuka dibuat semudah mungkin untuk dapat dipahami oleh *user*.

3.2.4 Kinerja yang Diharapkan

Kinerja yang diharapkan dari hasil analisis di atas adalah perangkat lunak yang dibuat mampu menangani berbagai kemungkinan masukan data (input) dan parameter operasi genetika unuk menghasilkan output yang optimum.

3.2.5 Kebutuhan Perangkat Lunak dan Perangkat Keras

3.2.5.1 Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk pengembangan dan implementasi sistem ini antara lain,

- sistem operasi Windows 9x atau Windows 2000 atau yang lebih tinggi;
- bahasa pemrograman Borland Delphi 7.0 untuk pembuatan perangkat lunak.

3.2.5.2 Kebutuhan Perangkat Keras

Sistem optimasi ini akan dapat berjalan dengan baik jika memenuhi standar minimum dari perangkat keras (*hardware*), antara lain

- prosesor: Intel Pentium III 600 Mhz;
- *hardisk*: 5GB;
- RAM: 256 MB;
- monitor: VGA dengan resolusi minimal 800 x 600;
- *keyboard*: standar;
- *mouse*: PS2.

BAB IV PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

Metode perancangan yang digunakan pada pengembangan perangkat lunak Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika adalah perancangan sistem berupa metode perancangan berarah aliran data berupa *flowchart*.

Flowchart merupakan teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output. *Flowchart* juga digunakan untuk menyajikan sebuah sistem atau perangkat lunak pada setiap tingkat abstraksi. Sedangkan diagram arus data digunakan untuk menggambarkan sistem data baru yang akan dikembangkan secara logika tanpa terlebih dahulu mempertimbangkan lingkungan fisik di mana data tersebut diproses atau di mana data tersebut akan disimpan.

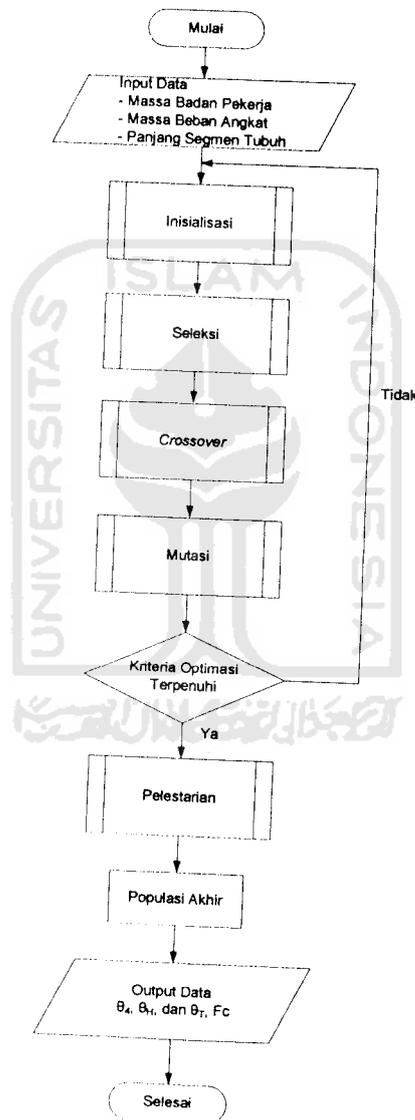
4.2 Hasil Perancangan

4.2.1 Perancangan *Flowchart*

Untuk memudahkan memahami mengenai proses suatu sistem maka disajikan dalam bentuk *flowchart* (aliran data). Perancangan *flowchart* untuk proses penyelesaian masalah Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika adalah sebagai berikut.

4.2.1.1 Flowchart Aplikasi Algoritma Genetika

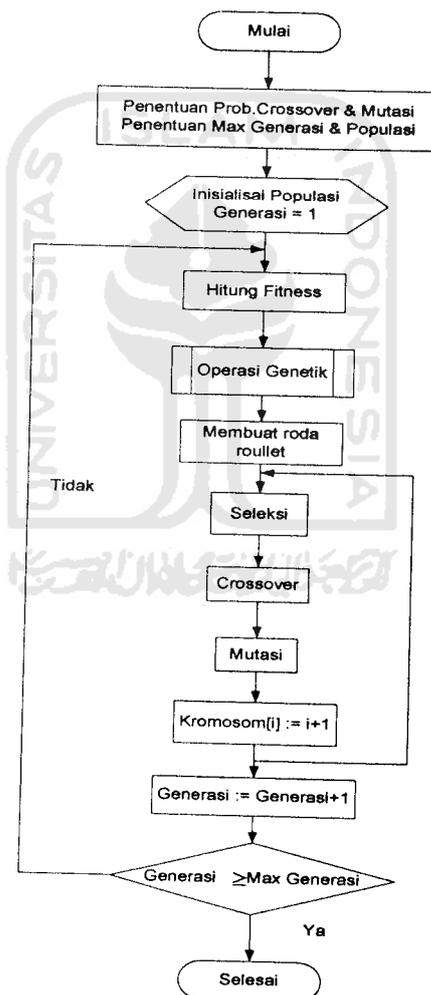
Flowchart ini menjelaskan jalannya aplikasi algoritma genetika dalam menyelesaikan permasalahan optimasi pada Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika seperti pada Gambar 4.1 berikut ini.



Gambar 4.1 Flowchart Aplikasi Algoritma Genetika

4.2.1.2 Flowchart Proses Algoritma Genetika

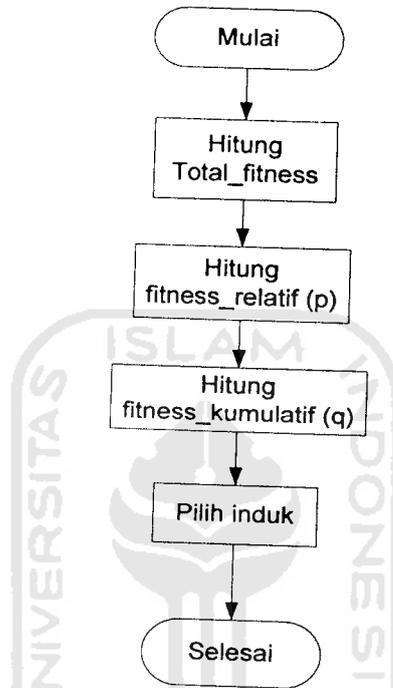
Flowchart ini menjelaskan jalannya proses algoritma genetika menggunakan metode seleksi roda rolet (*roulette wheels*), proses *crossover* dengan metode *one-cut-point crossover*, serta proses mutasi dengan metode *binary mutation*. Proses dimulai dari inialisasi populasi awal, generasi secara acak (*random*), evaluasi nilai *fitness* pada setiap individu dalam generasi, dilanjutkan dengan proses kondisional hingga generasi mencapai maksimum generasi.



Gambar 4.2 Flowchart Proses Algoritma Genetika

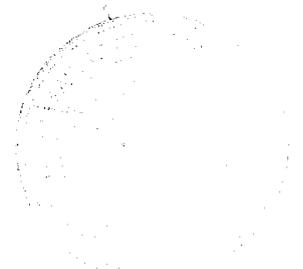
4.2.1.3 Flowchart Proses Seleksi Roda Roulette

Flowchart ini menjelaskan jalannya proses seleksi induk dengan menggunakan metode roda roulette (*roulette wheels*), dimulai dari menghitung nilai *fitness* sampai dengan selesai, seperti yang terdapat pada Gambar 4.3.



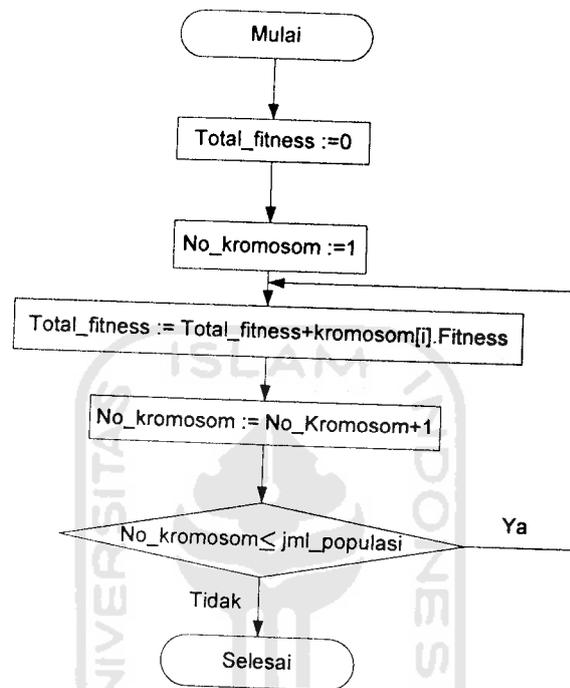
Gambar 4.3 *Flowchart* Seleksi dengan Roda Roulette

Pada *flowchart* di atas terlihat bahwa di dalam proses seleksi roda rolet masih terdapat tiga sub-proses. Masing-masing proses tersebut dijelaskan sebagai berikut.



4.2.1.3.1 Flowchart Proses Perhitungan Total Fitness

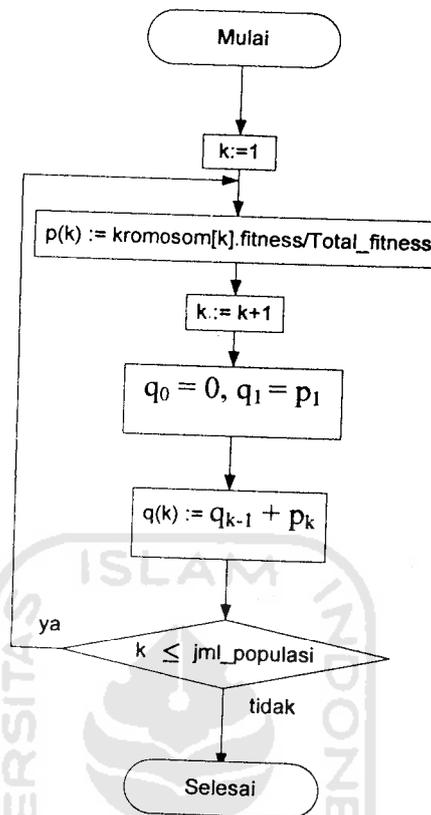
Flowchart ini menjelaskan jalannya proses perhitungan total *fitness*, dimulai dari inisialisasi nilai total *fitness* = 0 dan nomor kromosom = 1, sampai dengan akhir proses, seperti yang tampak pada Gambar 4.4 berikut ini.



Gambar 4.4 Flowchart Total Fitness

4.2.1.3.2 Flowchart Proses Perhitungan Fitness Relatif dan Fitness Kumulatif

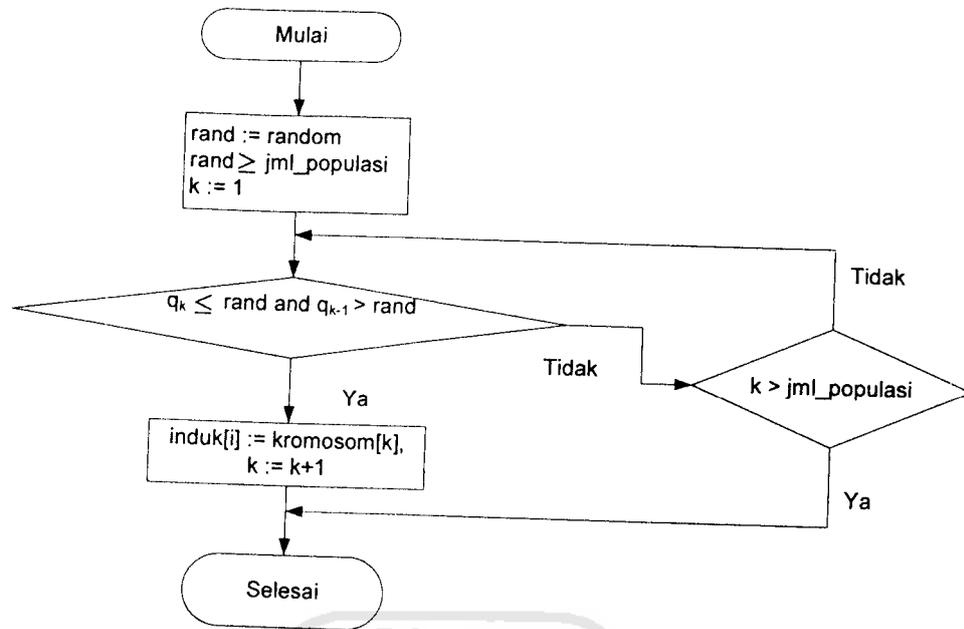
Flowchart ini menjelaskan jalannya proses perhitungan *fitness* relatif dan *fitness* kumulatif. Proses diawali dengan inisialisasi nilai k , dilanjutkan dengan proses kondisional, inisialisasi nilai q_0 dan q_1 . Proses kondisional ini dilakukan hingga sebanyak jumlah populasi dalam generasi, seperti yang tampak pada Gambar 4.5 berikut ini.



Gambar 4.5 Flowchart Hitung Fitness Relatif dan Fitness Kumulatif

4.2.1.3.3 Flowchart Proses Pilih Induk

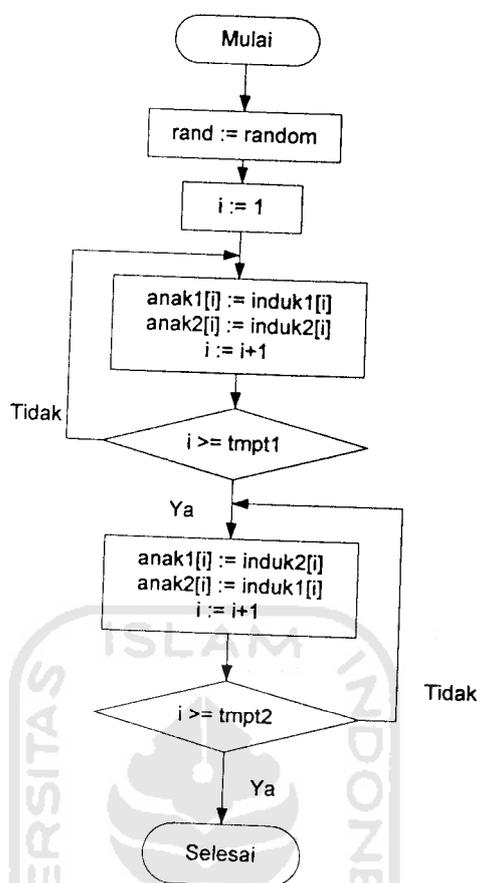
Flowchart ini menjelaskan jalannya proses pemilihan induk. Proses diawali dengan inialisasi nilai k , membangkitkan bilangan *random* biner sebanyak jumlah populasi dalam generasi, dilanjutkan dengan proses kondisional hingga sebanyak jumlah populasi dalam generasi, seperti yang tampak pada Gambar 4.6 berikut ini.



Gambar 4.6 Flowchart Pilih Induk

4.2.1.4 Flowchart Proses Crossover

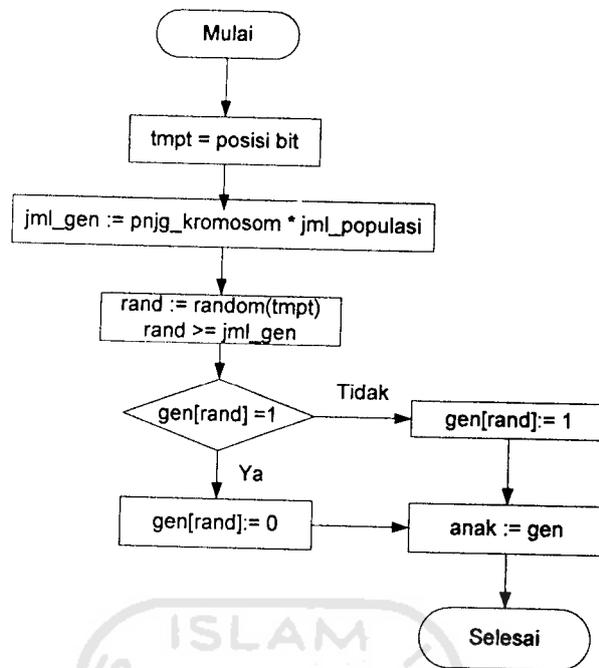
Flowchart ini menjelaskan jalannya proses *crossover* dengan menggunakan metode *one-cut-point-crossover*. Proses diawali dengan inialisasi nilai i , membangkitkan satu bilangan random, dilanjutkan dengan proses kondisional hingga sejumlah peluang *crossover* yang diharapkan dalam generasi, seperti yang terdapat pada Gambar 4.7.



Gambar 4.7 Flowchart Crossover dengan One-Cut-Point -Crossover

4.2.1.5 Flowchart Proses Mutasi

Flowchart ini menjelaskan jalannya proses mutasi dengan menggunakan metode *flipping mutation*. Proses diawali dengan inialisasi nilai *temp*, dilanjutkan dengan kondisi kondisional sejumlah peluang mutasi yang diharapkan pada sejumlah bit dalam generasi, seperti yang terdapat pada Gambar 4.8.



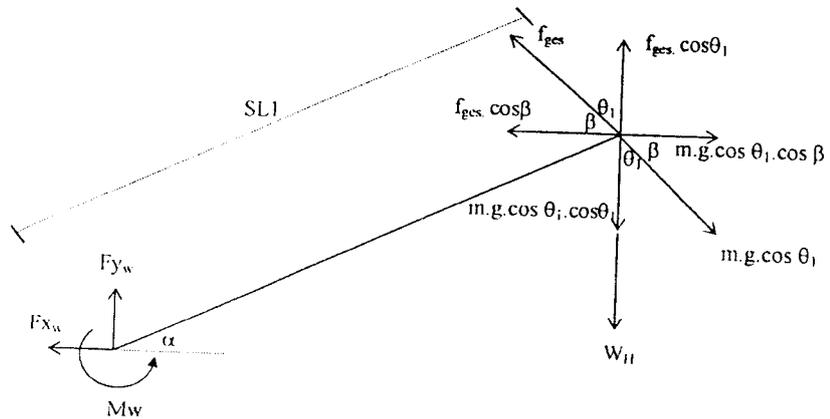
Gambar 4.8 Flowchart Mutasi Biner

4.2.2 Perancangan Variabel Input

4.2.2.1 Input Resultan Momen

Dengan menggunakan teknik perhitungan keseimbangan gaya pada segmen tubuh manusia, maka didapat momen resultan pada tulang belakang.

a. Telapak tangan



$$\Sigma F_x = 0, \Sigma F_y = 0, \Sigma M = 0$$

$$W_H = 0,6\% \times W_{\text{badan}}$$

$$F_{xw} = (m.g.\cos\theta_1.\cos\beta - f_{\text{ges}}.\cos\beta) : 2$$

$$F_{yw} = W_H + [(m.g.\cos\theta_1.\cos\theta_1 - f_{\text{ges}}.\cos\theta_1) : 2]$$

$$M_w = [W_H + (m.g.\cos\theta_1.\cos\theta_1 - f_{\text{ges}}.\cos\theta_1) : 2] SL_1.\cos\alpha - (m.g.\cos\theta_1.\cos\beta - f_{\text{ges}}.\cos\beta) SL_1.\sin\alpha$$

Keterangan:

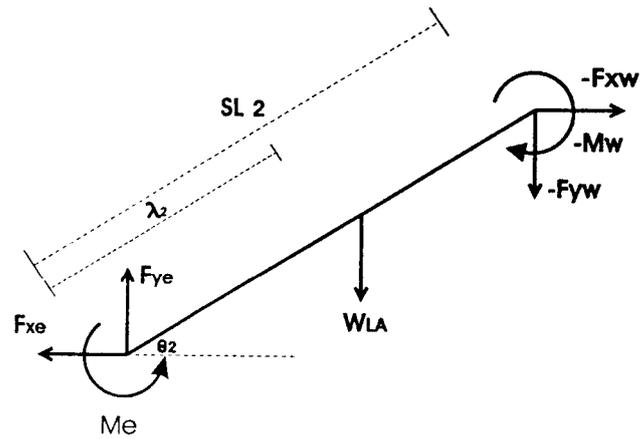
W_H = massa telapak tangan

F_{xw} = gaya pada gengaman telapak tangan sejajar sumbu x

F_{yw} = gaya pada gengaman telapak tangan sejajar sumbu y

M_w = momen gaya pada telapak tangan

b. Lengan bawah



$$\Sigma F_x = 0, \Sigma F_y = 0, \Sigma M = 0$$

$$\lambda_2 = 43\%$$

$$W_{LA} = 1,7\% \cdot W_{\text{badan}}$$

$$F_{xe} = F_{xw}$$

$$F_{ye} = F_{yw} + W_{LA}$$

$$M_e = M_w + (W_{LA} \cdot \lambda_2 \cdot SL_2 \cdot \cos \theta_2) + (F_{yw} \cdot SL_2 \cdot \cos \theta_2)$$

Keterangan:

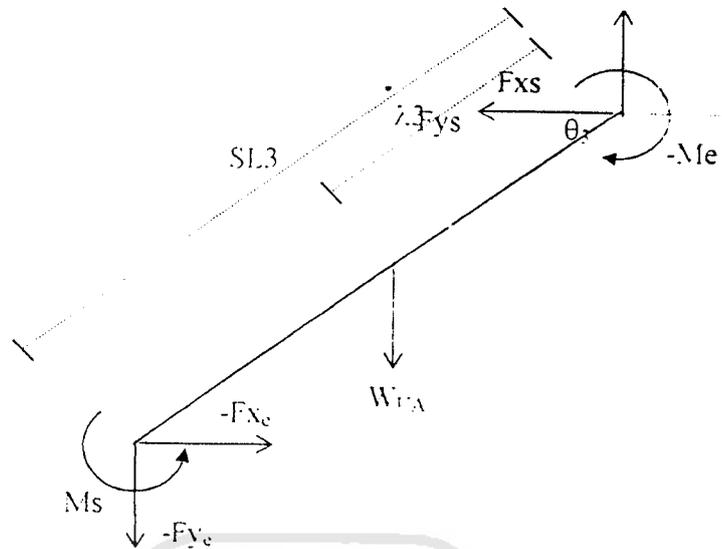
W_{LA} = berat lengan bawah

F_{xe} = gaya pada lengan bawah tangan sejajar sumbu x

F_{ye} = gaya pada lengan bawah sejajar sumbu y

M_w = momen gaya pada lengan bawah

c. Lengan atas



$$\Sigma F_x = 0, \Sigma F_y = 0, \Sigma M = 0$$

$$\lambda_3 = 43,6\%$$

$$W_{UA} = 2,8\% \times W_{\text{badan}}$$

$$F_{xs} = F_{xe}$$

$$F_{ys} = F_{ye} + W_{UA}$$

$$M_s = M_e + (W_{UA} \cdot \lambda_3 \cdot SL_3 \cdot \cos \theta_3) + (F_{ye} \cdot SL_3 \cdot \cos \theta_3)$$

Keterangan:

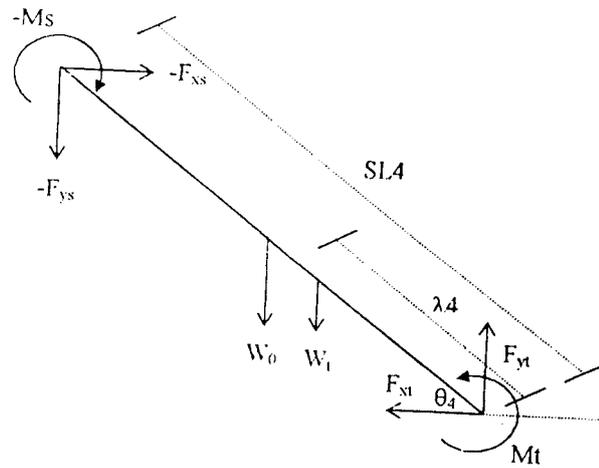
W_{UA} = berat lengan bawah

F_{xs} = gaya pada lengan atas sejajar sumbu x

F_{ys} = gaya pada lengan atas sejajar sumbu y

M_s = momen gaya pada lengan atas

d. Link punggung



$$\Sigma F_y = 0, \Sigma F_x = 0, \Sigma M = 0$$

$$\lambda_4 = 67\%$$

$$W_t = 45\% \times W_{\text{badan}}$$

$$F_{xt} = 2F_{xs}$$

$$F_{yt} = 2F_{ys} + W_0 + W_t$$

$$M_t = 2M_s + (W_0 \cdot 0,5 \cdot SL_4 \cdot \cos \theta_4) + (W_t \cdot \lambda_4 \cdot SL_4 \cdot \cos \theta_4) + (2F_{ys} \cdot SL_4 \cdot \cos \theta_4)$$

Keterangan:

W_t = berat segmen punggung

F_{xt} = gaya pada punggung sejajar sumbu x

F_{yt} = gaya pada punggung sejajar sumbu y

M_t = momen gaya pada punggung

Dengan menggunakan parameter gaya kompresi di atas, kemudian diolah dengan menggunakan algoritma genetika untuk mendapatkan hasil posisi pengangkatan yang optimal. Informasi yang digunakan adalah evaluasi persamaan total gaya otot *spinal erector* (F_m total) dari nilai kesesuaian (*fitness*) setiap kromosom yang dihasilkan. Fungsi tujuan yang akan dicapai yaitu meminimasi gaya total otot *spinal erector* (F_m total) yang di dalamnya dipengaruhi oleh beberapa sudut posisi pengangkatan, sehingga dihasilkan rekomendasi terhadap sudut posisi pengangkatan dinamis yang aman bagi pekerja untuk melakukan pengangkatan material secara manual.

4.2.2.2 Input Algoritma Genetika

1. Representasi kromosom

Representasi yang digunakan adalah dengan melakukan *encoding* variabel keputusan ke dalam bentuk deretan bilangan biner. Domain dari variabel sudut x_j adalah $[a_j, b_j]$ dan tingkat ketelitian yang dikehendaki adalah 4 angka di belakang desimal, maka range domain dari masing-masing variabel sudut harus dibagi paling sedikit berukuran $(b_j - a_j) \times 10^4$ range. Bits yang dikehendaki dari variabel sudut adalah sebagai berikut.

$$2^{m_{ji}-1} < (b_j - a_j) \times 10^4 \leq 2^{m_{ji}}$$

$$m = m_{ji} + m_{j(i+1)}$$

dengan:

m_{ji} = jumlah bits dalam kromosom pada masing-masing batasan (jika terdapat lebih dari satu batasan)

m = jumlah total bits dalam kromosom yang dikehendaki

b_j = batas atas variabel sudut

a_j = batas bawah variabel sudut.

Maka pembangkitan kromosom melalui representasi nilai-nilai sudut yang akan dicari, yaitu θ_H dan θ_T , sebagai berikut.

Tahap pertama

Encoding variabel keputusan ke dalam deretan bilangan biner dengan tingkat ketelitian 4 angka di belakang angka desimal. Maka syarat ketelitian dan *bits* yang dikehendaki dari setiap variabel sudut ditentukan berdasarkan

- sudut inklinasi badan relatif terhadap bidang horisontal (θ_H)

$$(39 - 29) \times 10^4 = 100000$$

$$2^{16} < 100000 \leq 2^{17}, \text{ sehingga } m_1 = 17$$

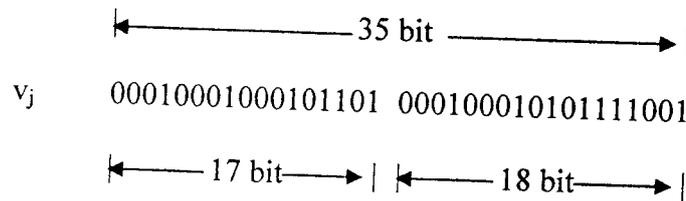
- sudut inklinasi kaki relatif terhadap bidang horisontal (θ_T)

$$(57 - 41) \times 10^4 = 160000$$

$$2^{17} < 160000 \leq 2^{18}, \text{ sehingga } m_2 = 18$$

Sehingga panjang total kromosom adalah $m = m_1 + m_2 = 17 + 18 = 35$ bit.

Dengan pembangkitan bilangan biner secara *random*, maka panjang kromosom pada satu populasi dapat direpresentasikan sebagai berikut.



Tahap kedua

Bilangan biner diubah menjadi bilangan real untuk variabel sudut θ_H dan θ_T .

- sudut $\theta_H = 000100010001011101$

maka didapat bilangan real

$$2^0 + 2^2 + 2^3 + 2^5 + 2^9 + 2^{13} = 8749$$

$$\theta_H = 29 + 8749 \times \frac{39 - 29}{2^{17} - 1} = 29,66750082^\circ$$

- sudut $\theta_T = 000100010101111001$

maka didapat bilangan real

$$2^3 + 2^4 + 2^5 + 2^6 + 2^8 + 2^{10} + 2^{14} = 17785$$

$$\theta_T = 41 + 17785 \times \frac{57 - 41}{2^{18} - 1} = 42,08551439^\circ$$

2. Evaluasi

Nilai *fitness* didapatkan dari minimalisasi gaya otot *spinal erector* (F_m). Agar algoritma genetika tidak terjaring konvergensi dini yang menyebabkan solusi menjadi tidak optimal maka diperlukan suatu konstanta C . Persamaannya

adalah $eval(v_k) = C - \sum_{i=1}^8 Fm_i$. Apabila ternyata fungsi objektif memiliki

nilai negatif, maka ditambahkan konstanta C agar nilai *fitness* yang terbentuk tidak menjadi negatif, sehingga persamaannya menjadi

$$eval(v_k) = C + \sum_{i=1}^8 Fm_i$$

dengan:

$$C = \text{konstanta} = 10^6$$

Fm_i = nilai total gaya otot *spinal erector* yang dihasilkan

3. Seleksi

Dalam kasus ini, metode yang digunakan adalah seleksi roda rolet (*roulette wheel parents selection*). Dalam seleksi roda rolet, meskipun pemilihan induk kromosom dilakukan secara random, namun pemilihannya masih proporsional dengan nilai evaluasi setiap kromosom.

Fungsi evaluasi setiap kromosom yang digunakan juga memperhatikan penskalaan setiap kromosom serta menjaga agar jangan sampai super kromosom mendominasi populasi yang dapat mengakibatkan proses algoritma genetika tergiring pada konvergensi dini, sehingga algoritma genetika tidak dapat menghasilkan solusi yang optimal.

4. Crossover

Dalam kasus ini, digunakan persilangan dengan satu titik potong (*one-cut point crossover*), di mana pemilihan titik potong dilakukan secara random dan menukar bagian yang tepat dari dua induk kromosom untuk menghasilkan anak kromosom.

5. Mutasi

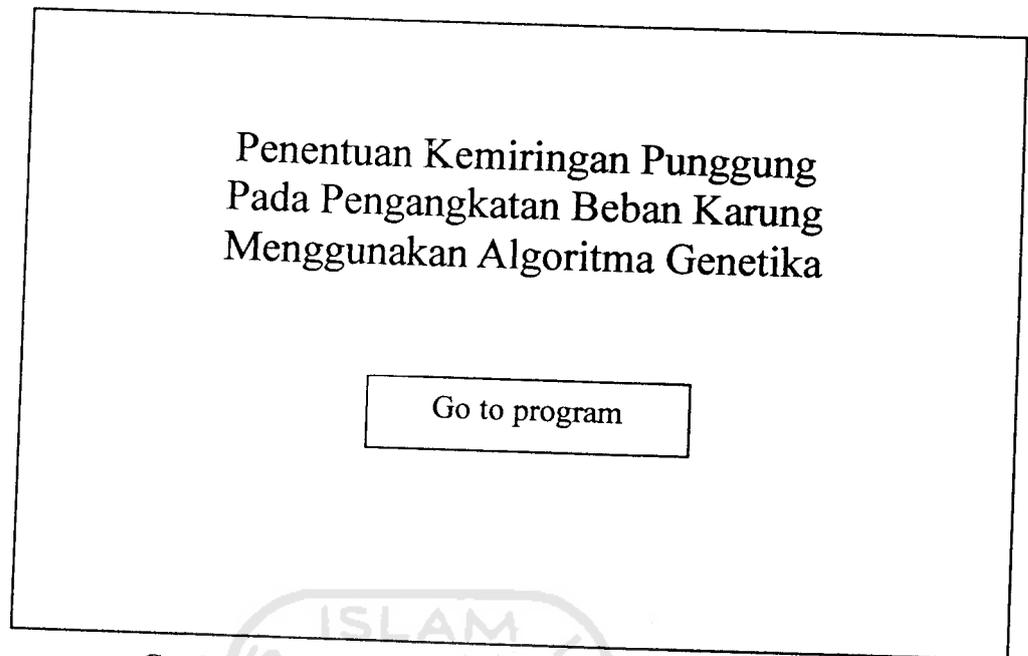
Mutasi merupakan *unary operator*, yang artinya dibutuhkan satu induk untuk dapat menghasilkan satu anak. Dalam kasus ini digunakan mutasi biner, dengan cara memilih gen dalam kromosom secara acak (berdasarkan nilai random yang dibangkitkan) dan menggantinya dengan bilangan selain bilangan gen tersebut. Jika angka gen yang terpilih adalah 0, maka diganti dengan 1, dan sebaliknya.

4.3 Perancangan Antarmuka

Rancangan antarmuka (*interface*) dari perangkat lunak ini ditujukan agar mempermudah penggunaan dalam mengoperasikan sistem. Rancangan antarmuka dari sistem yang akan dibuat ini menggunakan model grafis atau visual. Perancangan tersebut dibagi menjadi beberapa bagian yang akan dijelaskan di bawah ini.

4.3.1 Rancangan Antarmuka Halaman Utama

Antarmuka ini berisi judul dan menu utama dari sistem yang akan dibuat, seperti yang ditunjukkan pada Gambar 4.9.



Gambar 4.9 Rancangan Antarmuka Halaman Utama

4.3.2 Rancangan Antarmuka Menu

Antarmuka ini berisi dua sub-bagian menu antara lain Input dan Proses sebagai berikut.

4.3.2.1 Input

Antarmuka ini berfungsi untuk mendefinisikan input untuk proses sistem. Input yang dimasukkan adalah panjang segmen-segmen tubuh, berat beban yang diangkat, dan berat badan pekerja. Rancangannya seperti pada Gambar 4.10 berikut ini.

Panjang Telapak Tangan (SL_1)
 Berat Beban

Panjang Lengan Bawah (SL_2)

Panjang Lengan Atas (SL_3)
 Berat Badan

Panjang Tulang Belakang (SL_4)

Sudut Pergelangan Tangan
 Dengan Bidang Horizontal (α)

Sudut Kemiringan Beban
 Terhadap Vertikal (θ_1)

Sudut Kemiringan Beban
 Terhadap Bidang Horizontal (β)

Sudut Antara Siku dengan
 Bidang Horizontal (θ_2)

Sudut Antara Bahu dengan
 Bidang Horizontal (θ_3)

Memo

Gambar 4.10 Rancangan Antarmuka Input

4.3.2.3 Proses

Antarmuka ini digunakan untuk melakukan dua proses. Proses-proses tersebut adalah sebagai berikut.

1. Proses perhitungan gaya dan momen gaya

Setelah dilakukan masukan data awal, maka dilakukan perhitungan besarnya gaya dan momen gaya yang bekerja pada telapak tangan, lengan bawah, lengan atas, dan tulang belakang untuk output akhir pada proses selanjutnya.

Proses ini ditampilkan pada bagian memo pada Gambar 4.10 di atas.

2. Proses algoritma genetika

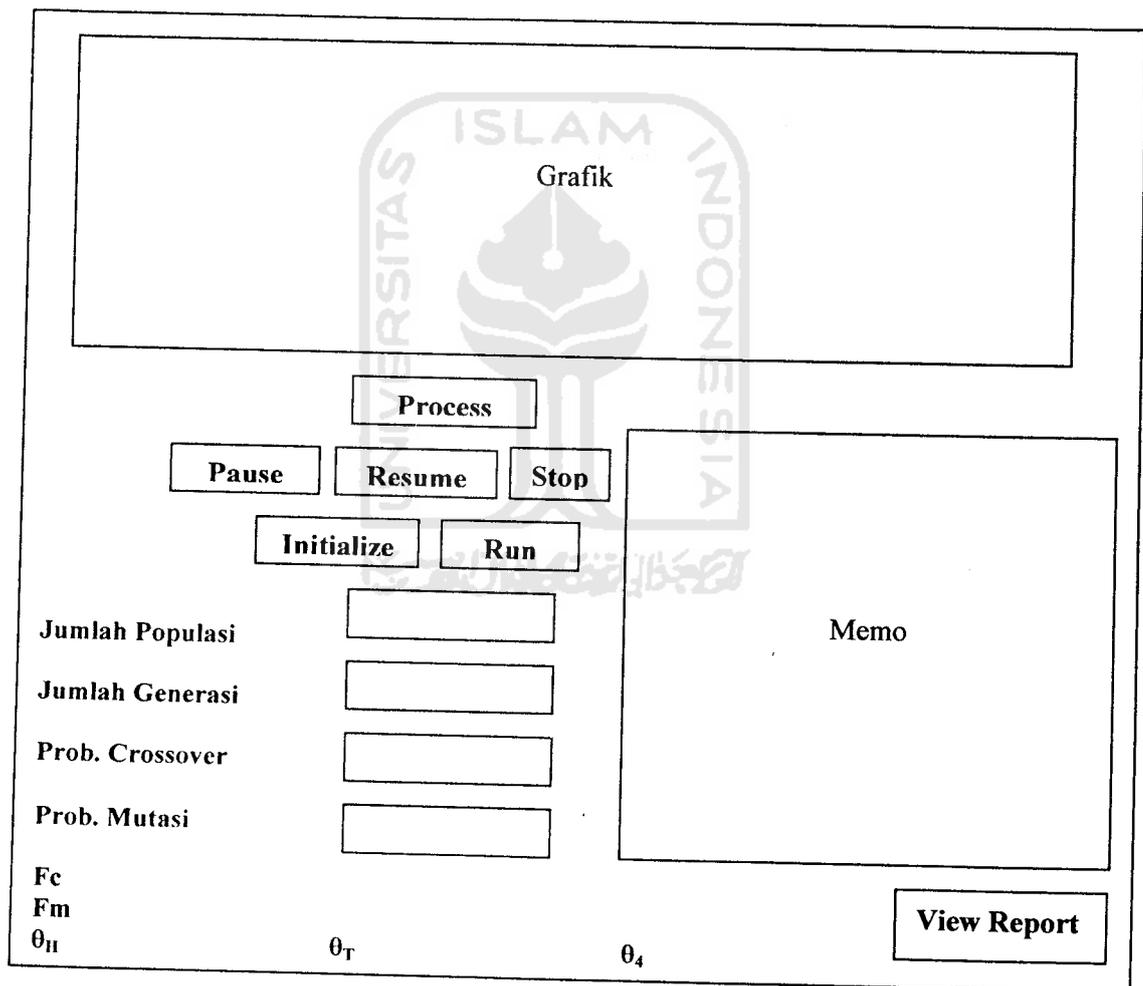
a. Proses seleksi

Seleksi roda rolet yang digunakan berfungsi untuk mencari nilai *fitness* yang didapatkan dengan meminimumkan nilai gaya otot (F_m).

b. Proses *crossover*

c. Proses mutasi.

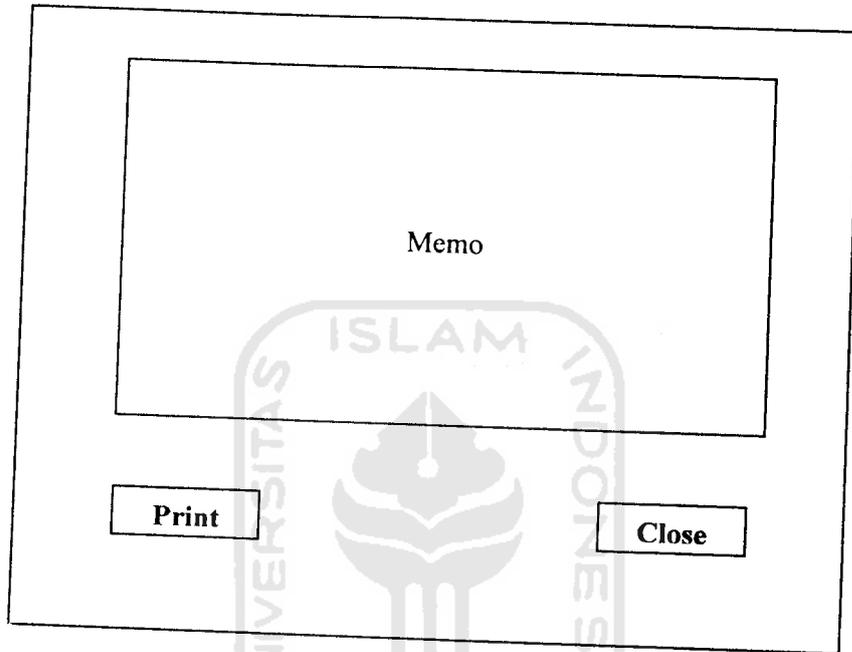
Proses algoritma genetika ini ditunjukkan pada Gambar 4.11.



Gambar 4.11 Rancangan Antarmuka Proses

4.3.2.4 Report

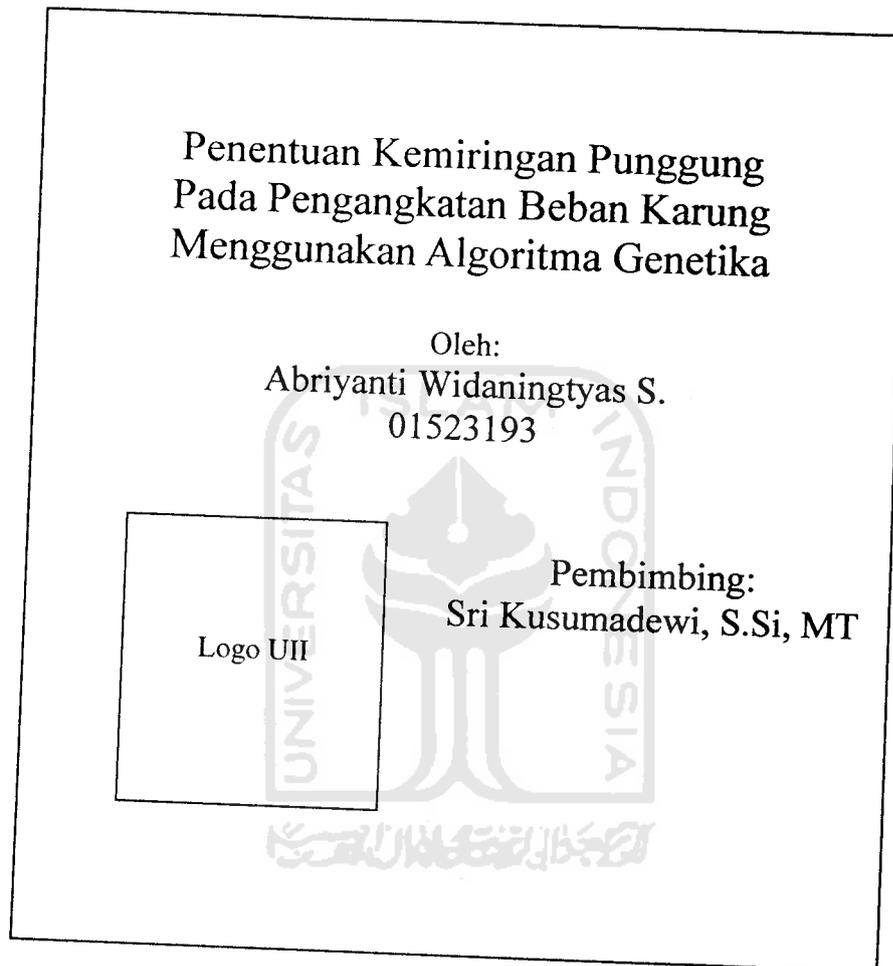
Form Report ini berisi penjelasan tentang hasil dari proses yang telah dilakukan. *Form* ini berisi parameter algoritma genetika yang digunakan serta hasil pemrosesan.



Gambar 4.12 Rancangan Antarmuka Report

4.3.2.5 About

Form About ini berisi tentang data diri pembuat perangkat lunak. Rancangannya seperti tampak pada Gambar 4.13 di bawah ini.



Gambar 4.13 Rancangan Antarmuka Tentang Perangkat Lunak

BAB V IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi

Implementasi dari Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika ini menggunakan bahasa pemrograman Borland Delphi 7.0. Implementasi merupakan salah satu tahap di mana sistem dioperasikan pada keadaan yang sebenarnya, sehingga dapat diketahui dan dipastikan bahwa sistem yang dibangun benar-benar sesuai dengan yang diharapkan.

5.1.1 Perangkat Keras yang Dibutuhkan

Perangkat keras minimal yang dianjurkan untuk dapat menggunakan sistem adalah :

- a. Prosesor Intel Pentium IV 2,4 Ghz;
- b. RAM minimal 256 MB;
- c. *Harddisk* dengan kapasitas 2GB atau lebih;
- d. VGA dan monitor dengan resolusi minimal 800 x 600 piksel;
- e. CD ROM *drive*;
- f. *Mouse*;
- g. *Keyboard*.

5.1.2 Perangkat Lunak yang Dibutuhkan

Perangkat lunak yang digunakan dalam pengembangan sistem adalah :

- a. Sistem Operasi : Microsoft Windows XP SP1
- b. Bahasa Pemrograman : Borland Delphi 7.0

5.1.3 Bahasa dan *Compiler* yang Digunakan

Bahasa yang digunakan adalah bahasa Pascal untuk pemrograman di lingkungan Borland Delphi. Meskipun pada borland Delphi, bahasa Pascal yang digunakan bukan bahasa Pascal murni, namun merupakan bahasa yang umumnya digunakan untuk menyelesaikan permasalahan matematis, numeris dan pemrograman basis data. Sedangkan *compiler* yang dipilih adalah Borland Delphi 7.0, karena memiliki kelebihan dalam mengimplementasikan sistem, antara lain pendekatan pemrograman yang bersifat visual, ketersediaan sarana untuk mengolah basis data dengan mudah, dan kecepatan serta keandalannya ketika dioperasikan.

5.2 Implementasi Perangkat Lunak

Implementasi sistem merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan yang sesungguhnya. Dari implementasi ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak dan menghasilkan output yang sesuai dengan perancangan yang ada.

5.3 Implementasi Prosedural

5.3.1 Modul-modul dalam Perangkat Lunak

Pemrograman pada Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika ini menggunakan modul-modul yang berisi prosedur dan fungsi-fungsi sehingga program dapat berjalan dengan baik. Modul-modul yang digunakan pada pengembangan sistem ini adalah sebagai berikut.

5.3.1.1 Math

Modul ini berisi prosedur dan fungsi-fungsi matematika untuk proses perhitungan dengan menggunakan rumus seperti perpangkatan, penambahan, pengurangan, perkalian, pembagian, sinus, cosinus dan lain sebagainya.

5.3.1.2 Chart

Modul ini berisi *chart* atau grafik yang diperlukan untuk menunjukkan *progress* besarnya sudut θ_H dan θ_T yang optimum sehingga diperoleh gaya kompresi (F_C) yang optimum pada proses genetika.

5.3.2 Prosedur-prosedur dalam Perangkat Lunak

Pemrograman pada aplikasi Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika ini terdiri atas

prosedur-prosedur yang dapat memudahkan dalam pembuatan program. Berikut ini merupakan prosedur-prosedur yang digunakan.

5.3.2.1 Kode Program untuk Algoritma Genetika

1. *Source code* untuk mendeklarasikan tipe data

```

pop : array of array of integer;
tpop : array of array of integer;
res : array of array of real;
tp : array of integer;
ccrom : integer;
cpop : integer;
cgen : integer;
offsp : array of array of integer;
bestfit : real;
bestres : real;
bestH : real;
bestT : real;
bestgen : integer;
curgen : integer;
AA, P, Q, R, S, T, U, D, E, Te4, TeH, TeT, F : real;
Pa, Fa, Wto, Wh, Wla, Wua, Fc, Fm : real;

```

2. *Source code* untuk memasukkan data populasi

```

SetLength(pop, cpop, ccrom);
for i := 0 to (cpop - 1) do
begin
  ss := '';
  for j := 0 to (ccrom - 1) do
  begin
    pop[i][j] := Floor(Random(2));
    ss := ss + ' ' + IntToStr(pop[i][j]);
  end;
end;

```

3. Representasi kromosom

Source code untuk mendeklarasikan representasi kromosom adalah

```
tFit := 0;
for i := 0 to (cpop - 1) do
begin
  //representasi sudut tetha H
  bin := '';
  for j := 0 to 16 do
  begin
    bin := bin + IntToStr(pop[i][j]);
  end;
  sudut1 := bin2dec(bin);
  //representasi sudut tetha T
  bin := '';
  for j := 17 to 35 do
  begin
    bin := bin + IntToStr(pop[i][j]);
  end;
  sudut2 := bin2dec(bin);
  //res[i][0] := sudut1;
  //res[i][1] := sudut2;

  res[i][0] := 29 + (sudut1 * (39 - 29) / (Power(2, 17) - 1));
  res[i][1] := 41 + (sudut2 * (57 - 41) / (Power(2, 18) - 1));

  TeH := 29 + (sudut1 * (39 - 29) / (Power(2, 17) - 1));
  TeT := 41 + (sudut2 * (57 - 41) / (Power(2, 18) - 1));
  Te4 := TeH - 5;
```

4. Evaluasi

Source code di bawah ini berfungsi untuk menentukan nilai *fitness*

```
if F >= 0 then
begin
  res[i][2] := 1000000 - Fm;
end
else
begin
  res[i][2] := 1000000 + Fm;
end;
```

Source code di bawah ini berfungsi untuk menentukan nilai *fitness* kumulatif

```
kFit := 0;
for i := 0 to (cpop - 1) do
begin
  kFit := kFit + res[i][2] / tFit;
  res[i][3] := kFit;
  if (i = (cpop - 1)) then res[i][3] := 1;
end;
```

5. Seleksi

Source code untuk proses seleksi adalah

```
for i := 0 to (cpop - 1) do
begin
  lSelect := 0;
  Randomize;
  rr := Random;
  for j := 0 to ((cpop - 1) - 1) do
begin
  if ((rr >= res[j][3]) and (rr < res[j + 1][3])) then
begin
  lSelect := j + 1;
end;
end;
for j := 0 to (ccrom - 1) do
begin
  tpop[i][j] := pop[lSelect][j];
end;
end;
for i := 0 to (cpop - 1) do
begin
  for j := 0 to (ccrom - 1) do
begin
  pop[i][j] := tpop[i][j];
end;
end;
end;
```

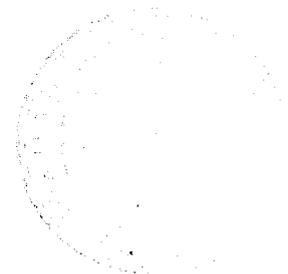
6. Crossover

Source code ini berfungsi untuk memilih populasi yang akan melalui proses *crossover*.

```
//memilih populasi yg akan dicrossover
for i := 0 to (cpop -1) do
begin
  Randomize;
  rr := Random;
  if (rr < StrToFloat(eCross.Text)) then
  begin
    for j := 0 to (ccrom -1) do
    begin
      tpop[i][j] := pop[i][j];
    end;
  end
  else
  begin
    for j := 0 to (ccrom -1) do
    begin
      tpop[i][j] := -1;
    end;
  end;
end;
```

Source code untuk proses *crossover* adalah

```
//proses CROSSOVER
i1 := -1;
i2 := -1;
for i := 0 to (cpop - 1) do
begin
  if (tpop[i][0] <> -1) then
  begin
    if (i1 = -1) then
    begin
      i1 := i;
    end
    else if (i2 = -1) then
    begin
      i2 := i;
    end
    rr := Random(ceil((ccrom / 2) - 1));
    ixR1 := Ceil(rr);
    ixR2 := Ceil(rr + (ccrom / 2));
    SetLength(offsp, 0, 0);
    SetLength(offsp, 2, ccrom);
    for j := ixR1 to ixR2 do
```



```

begin
  offsp[0][j] := tpop[i1][j];
  offsp[1][j] := tpop[i2][j];
end;

for j := 0 to (ixR1 - 1) do
begin
  offsp[0][j] := tpop[i2][j];
  offsp[1][j] := tpop[i1][j];
end;

for j := (ixR2 + 1) to (ccrom - 1) do
begin
  offsp[0][j] := tpop[i2][j];
  offsp[1][j] := tpop[i1][j];
end;

for j := 0 to (ccrom - 1) do
begin
  tpop[i1][j] := offsp[0][j];
  tpop[i2][j] := offsp[1][j];
end;
i1 := -1;
i2 := -1;
end;
end;
end;

for i := 0 to (cpop - 1) do
begin
  if (tpop[i][0] <> -1) then
  begin
    for j := 0 to (ccrom - 1) do
    begin
      pop[i][j] := tpop[i][j];
    end;
  end;
end;
end;

mmShow.Lines.Add('');
mmShow.Lines.Add('');

for i := 0 to (cpop - 1) do
begin
  ss := '';
  for j := 0 to (ccrom - 1) do
  begin
    ss := ss + ' ' + IntToStr(pop[i][j]);
  end;
  mmShow.Lines.Add(ss)
end;
end;

```

7. Mutasi

Source code untuk proses mutasi adalah

```

for i := 0 to (cpop - 1) do
begin
  rr := Random;
  if (rr < StrToFloat(eMutasi.Text)) then
  begin
    ixR1 := Ceil(Random(ccrom));
    ixR2 := Ceil(Random(ccrom));
    nR1 := pop[i][ixR1];
    nR2 := pop[i][ixR2];
    pop[i][ixR1] := nR2;
    pop[i][ixR2] := nR1;
  end;
end;

mmShow.Lines.Add('');
mmShow.Lines.Add('');

for i := 0 to (cpop - 1) do
begin
  ss := '';
  for j := 0 to (ccrom - 1) do
  begin
    ss := ss + ' ' + IntToStr(pop[i][j]);
  end;
  mmShow.Lines.Add(ss)
end;

```

8. Pembaharuan Generasi

Source code berikut ini berfungsi untuk memindahkan data populasi untuk di-
sort

```

for i := 0 to (cpop - 1) do
begin
  for j := 0 to (ccrom - 1) do
  begin
    tpop[i][j] := pop[i][j];
  end;
end;

sudah := false;
while (sudah = false) do
begin
  sudah := true;

```

```

for i := 0 to ((cpop - 1) - 1) do
begin
  if (res[i][2] < res[i + 1][2]) then
  begin
    sudah := false;
    for j := 0 to (ccrom - 1) do

      begin
        t1 := tpop[i][j];
        t2 := tpop[i + 1][j];
        tpop[i][j] := t2;
        tpop[i + 1][j] := t1;
      end;
    for j := 0 to 4 do
    begin
      r1 := res[i][j];
      r2 := res[i + 1][j];
      res[i][j] := r2;
      res[i + 1][j] := r1;
    end;
  end;
end;
end;
end;

```

Source code ini berfungsi untuk mencari nilai fitness terbaik.

```

if ((curgen = 1) or (bestfit < res[0][2])) then
begin
  bestfit := res[0][2];
  bestres := res[0][4];
  bestH := res[0][0];
  bestT := res[0][1];
  bestgen := curgen;

```

Source code untuk proses pelestarian adalah

```

cc := 0;
for i := 0 to (cpop - 1) do
begin
  rr := Random;
  if (rr < 0.2) then // <- besarnya prob. pelestarian
  begin
    for j := 0 to (ccrom - 1) do
    begin
      pop[i][j] := tpop[cc][j];
    end;
    cc := cc + 1;
  end;
end;
end;

```

5.3.3 Proses yang Digunakan

Proses-proses yang digunakan dalam aplikasi Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika ini antara lain

1. Proses perhitungan gaya dan momen gaya

Setelah dilakukan masukan data awal, maka dilakukan perhitungan besarnya gaya dan momen gaya yang bekerja pada telapak tangan, lengan bawah, lengan atas, dan tulang belakang untuk output akhir pada proses selanjutnya.

2. Proses algoritma genetika

- a. Proses seleksi

Proses seleksi yang menggunakan metode seleksi roda rolet merupakan suatu proses pencarian nilai *fitness* yang didapatkan dengan meminimumkan nilai gaya otot *spinal erector* (F_m).

- b. Proses *crossover*

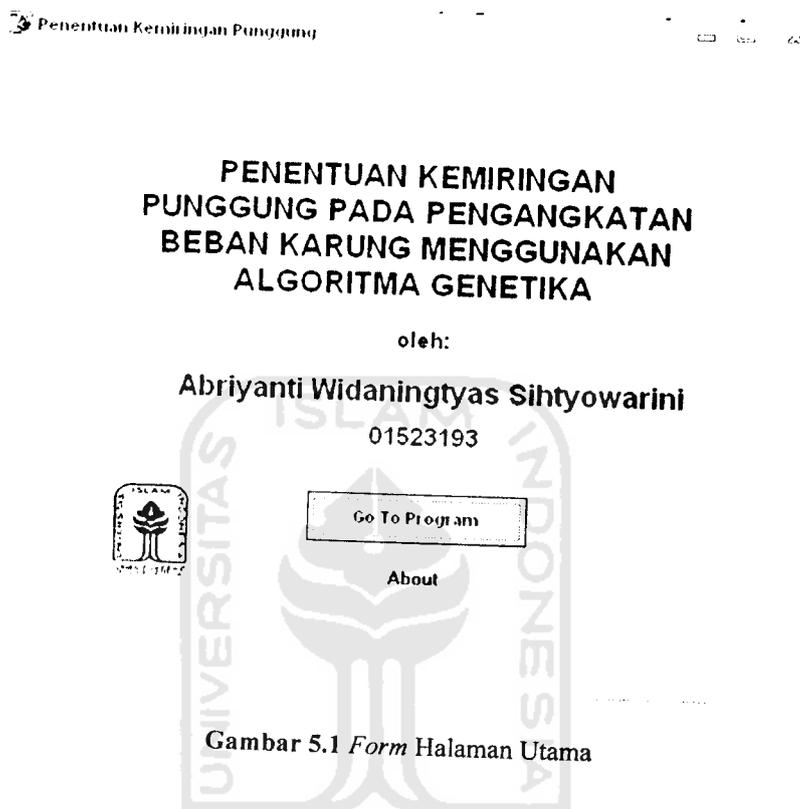
- c. Proses mutasi.

5.3.4 Implementasi Antarmuka

Implementasi antarmuka dari Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika adalah sebagai berikut.

5.3.4.1 Tampilan *Form* Halaman Utama

Form halaman utama dari sistem yang telah dibuat ini ditunjukkan pada Gambar 5.1.



Gambar 5.1 *Form* Halaman Utama

5.3.4.2 Tampilan *Form* Input

Form input berisi *input* data yang terdiri dari panjang segmen-segmen tubuh yang terlibat dalam proses pengangkatan beban, berat beban dan berat badan pekerja.

Input

Panjang Telapak Tangan (SL1)

Berat Beban

Panjang Lengan Bawah (SL2)

Berat Badan

Panjang Lengan Atas (SL3)

Panjang Tulang Belakang (SL4)

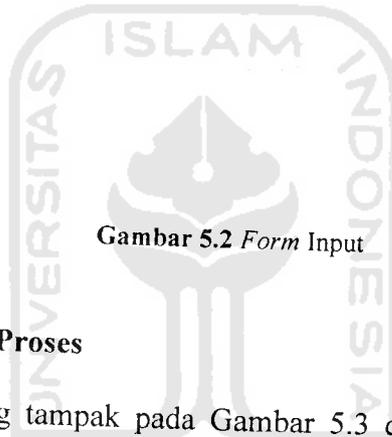
Sudut Pergelangan Tangan
dengan Bidang Horizontal (Alpha)

Sudut Kemiringan Beban
Terhadap Vertikal (Tetha 1)

Sudut Kemiringan Beban
Terhadap Horizontal (Beta)

Sudut Antara Siku dengan
Bidang Horizontal (Tetha2)

Sudut Antara Bahu dengan
Bidang Horizontal (Tetha3)



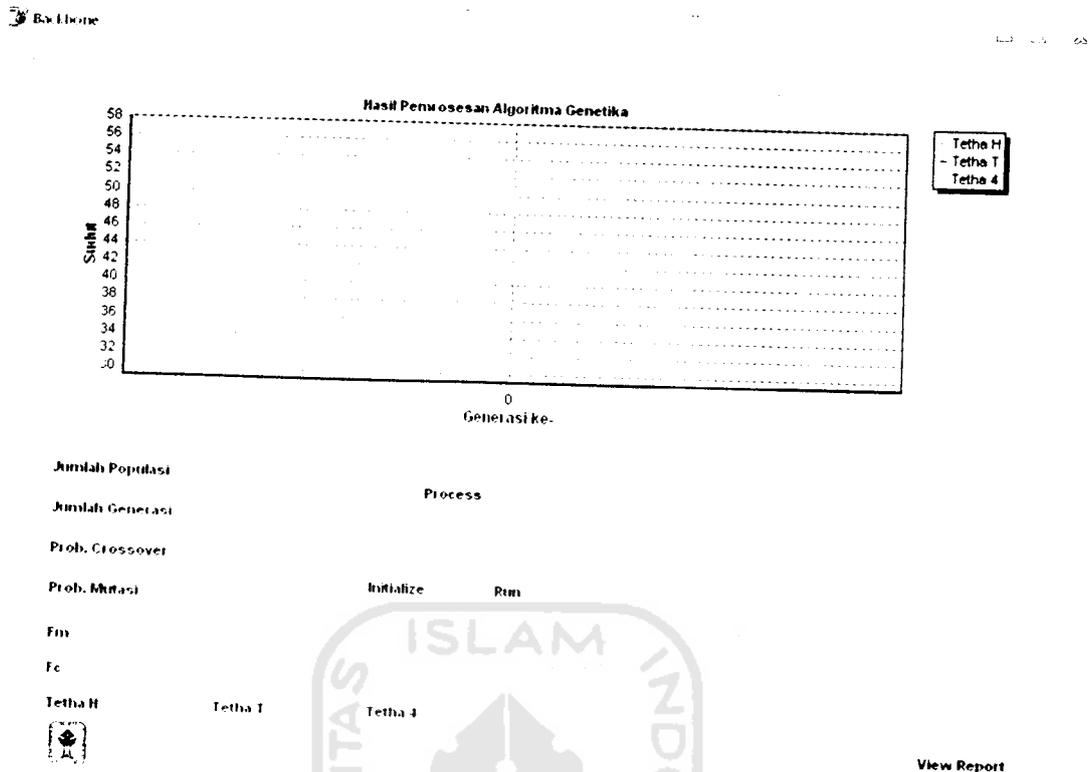
Hitung

Next

Gambar 5.2 *Form* Input

5.3.4.3 Tampilan *Form* Proses

Form proses yang tampak pada Gambar 5.3 di bawah ini berfungsi untuk memvisualisasikan jalannya proses pencarian nilai minimum dari fungsi tujuan, atau dengan kata lain meminimumkan nilai F_m , serta proses pencarian sudut optimum dengan algoritma genetika.



Gambar 5.3 Form Proses

5.3.4.4 Tampilan Form Report

Gambar 5.4 di bawah ini menampilkan sebuah *form* yang berisi penjelasan tentang parameter-parameter algoritma genetika yang digunakan serta hasil proses yang telah dilakukan antara lain nilai *fitness* terbaik, generasi terbaik, θ_H terbaik, θ_T terbaik, θ_4 terbaik, nilai F_m , nilai F_c serta kromosom dari generasi terbaik. Selain itu juga terdapat fungsi untuk mencetak *report* tersebut.

Hasil

HASIL PENCARIAN SUDUT OPTIMAL MENGUNAKAN ALGORITMA GENETIKA

PARAMETER ALGORITMA GENETIKA

JUMLAH POPULASI = 80
 JUMLAH MAKSIMUM GENERASI = 200
 PROBABILITAS CROSSOVER = 0.45
 PROBABILITAS MUTASI = 0.01

HASIL PEMROSESAN

FITNES TERBAIK = 0
 GENERASI TERBAIK = 0
 θ_{11} TERBAIK = 0
 θ_T TERBAIK = 0



Print

Close

Gambar 5.4 Form Report

5.3.4.5 Tampilan Form About

Pada gambar 5.5 di bawah ini tampak sebuah *form* yang pada sistem dinamakan About. *Form* ini digunakan untuk menerangkan data diri pembuat *software* Penentuan Kemiringan Punggung Pada Pengangkatan Beban Karung Menggunakan Algoritma Genetika.

About

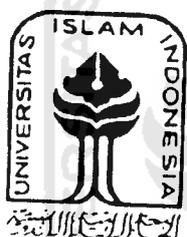


**PENENTUAN KEMIRINGAN
PUNGGUNG PADA PENGANGKATAN
BEBAN KARUNG MENGGUNAKAN
ALGORITMA GENETIKA**

oleh:

Abriyanti Widaningtyas Sihtyowarini

01523193



Pembimbing:

Sri Kusumadewi S.Si., MT

Close

Gambar 5.5 Form About

BAB VI ANALISIS KINERJA PERANGKAT LUNAK

6.1 Pengujian Perangkat Lunak

Dalam membangun sebuah perangkat lunak, diperlukan sebuah pengujian (*testing*) untuk memastikan jalannya fungsi serta untuk memastikan kemampuan suatu perangkat lunak. Pengujian ini dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi, baik yang disebabkan oleh kesalahan logika maupun kesalahan keluaran (*output*), sehingga dapat segera diantisipasi dengan melakukan perbaikan-perbaikan dan meminimumkan kesalahan-kesalahan yang terjadi.

Pengujian perangkat lunak dibagi menjadi dua tahap, antara lain pengujian tidak normal dan pengujian normal.

6.2 Pengujian Terhadap Sistem

6.2.1 Pengujian Tidak Normal

Yang dimaksud dengan pengujian tidak normal adalah apabila *user* melakukan kesalahan dalam memasukkan input, maka sistem akan menampilkan suatu pesan kesalahan. Selain itu, sistem juga tidak dapat menjalankan proses selanjutnya. Pengujian tidak normal ini memiliki aturan-aturan sebagai berikut.

1. Apabila *user* tidak memasukkan input, maka sistem akan menampilkan pesan kesalahan seperti Gambar 6.1 dan Gambar 6.2.

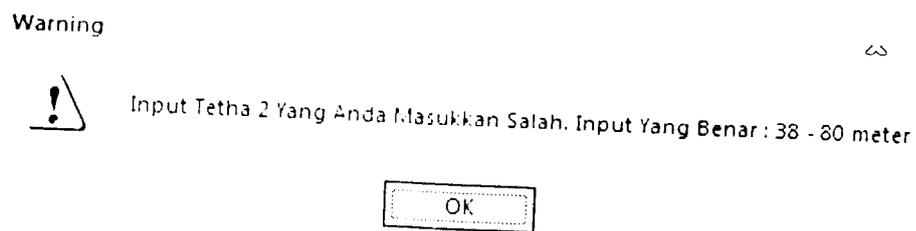


Gambar 6.1 Pesan Kesalahan Input SL_4 belum terisi



Gambar 6.2 Pesan Kesalahan Input Jumlah Generasi belum terisi

2. Apabila *user* salah memasukkan input, misal memasukkan input di luar *range* yang telah ditentukan atau memasukkan input 0, maka tampilan pesan kesalahan akan tampak seperti Gambar 6.3 dan Gambar 6.4.



Gambar 6.3 Pesan Kesalahan untuk Input θ_2

Warning



Probabilitas Crossover Yang Anda Masukkan Salah. Input Yang Benar : 0 - 1

Gambar 6.4 Pesan Kesalahan untuk Input *Crossover*

3. Apabila *user* memasukkan sembarang input, seperti tanda negatif atau tanda baca lainnya, atau menginputkan huruf, maka sistem tidak akan menampilkan masukan tersebut pada *form*. *Source code*-nya adalah sebagai berikut.

```

if not (Key in ['0'..'9', ',', '.', #8]) then
begin
  Key := #0;
end;
if Key in [',', '.', #8] then
begin
  Key := DecimalSeparator;
end;

```

6.2.2 Pengujian Normal

Pengujian dan analisis perangkat lunak adalah sesuatu yang harus dilakukan untuk melihat kinerja sistem yang dihasilkan dengan kesesuaian yang diharapkan, dengan demikian dapat diketahui kinerja dari aplikasi tersebut. Pada pengujian ini, dilakukan analisis terhadap jumlah generasi yang digunakan pada proses algoritma genetika. Jumlah generasi pada proses algoritma genetika yang akan diujikan adalah 100 generasi, 200 generasi, 300 generasi, dan 400 generasi.

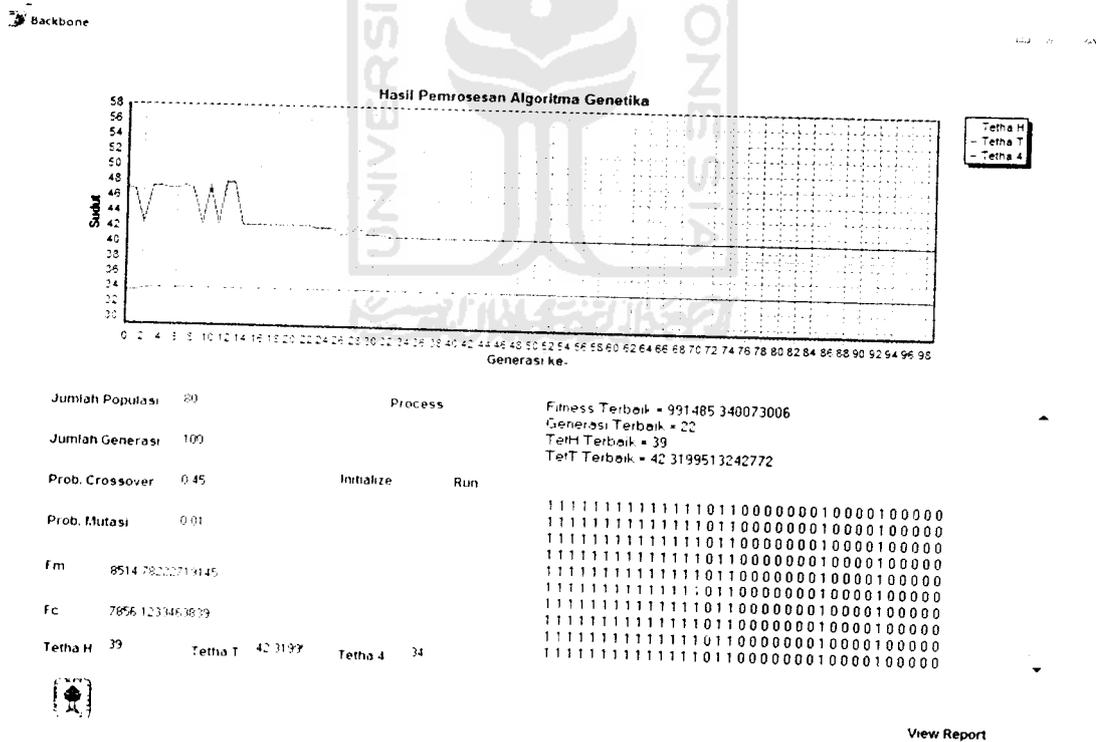
Dalam pengujian, parameter algoritma genetika yang digunakan merupakan parameter yang telah dibahas pada bab II. Sistem ini menggunakan parameter nilai *fitness* dari individu terbaik pada setiap generasi, maka digunakan jumlah populasi sebesar 80, probabilitas *crossover* (p_c) sebesar 0,45 dan probabilitas mutasi (p_m) sebesar 0,01 [KUS03]. Namun untuk penggunaan selanjutnya, *user* tidak terikat pada parameter tersebut. Selain itu, untuk memudahkan pengujian perangkat lunak, maka untuk masing-masing pengujian, digunakan input yang sama antara lain

- panjang telapak tangan (SL_1) = 0.114 m;
- panjang lengan bawah (SL_2) = 0.256 m;
- panjang lengan atas (SL_3) = 0.268 m;
- panjang tulang belakang (SL_4) = 0.449 m;
- sudut pergelangan tangan dengan bidang horisontal (α) = 13.69 m;
- sudut kemiringan beban terhadap bidang vertikal (θ_1) = 38.33 m;
- sudut kemiringan beban terhadap bidang horisontal (β) = 51.675 m;
- sudut antara siku dengan bidang horisontal (θ_2) = 54.75 m;
- sudut antara bahu dengan bidang horisontal (θ_3) = 35.96 m;
- massa badan pekerja = 50 kg;
- massa beban angkat = 50 kg;
- jumlah populasi = 80 populasi.
- probabilitas *crossover* (p_c) = 0,45;
- probabilitas mutasi (p_m) = 0,01.

Berikut ini merupakan pengujian-pengujian yang dilakukan terhadap sistem yang telah dibuat.

a. Pengujian 1

- Pada pengujian ini, digunakan jumlah generasi 100.
- Hasil pengujian 1, sudut-sudut yang diperoleh antara lain $\theta_H = 39^\circ$, $\theta_T = 42,3199513242772^\circ$, dan $\theta_4 = 34^\circ$.
- Gaya kompresi (F_c) yang dihasilkan adalah 7858,1233463839 Newton.
- Generasi terbaik = 22, nilai *fitness* terbaik = 991485,340073006.
- Proyeksi sudut dan besarnya gaya kompresi (F_c) pada pengujian 1 ditunjukkan pada Gambar 6.5.



Gambar 6.5 Hasil Pengujian 1

- Laporan hasil pemrosesan dapat dilihat pada Gambar 6.6 di bawah ini.

Hasil

PARAMETER ALGORITMA GENETIKA

JUMLAH POPULASI	= 80
JUMLAH MAKSIMUM GENERASI	= 100
PROBABILITAS CROSSOVER	= 0.45
PROBABILITAS MUTASI	= 0.01

HASIL PEMROSESAN

FITNES TERBAIK	= 991485.340073006
GENERASI TERBAIK	= 22
θ_1 TERBAIK	= 39°
θ_2 TERBAIK	= 42.3199513242772°
θ_4 TERBAIK	= 34°
F_M	= 8514.78222719145 NEWTON
F_c	= 7856.1233463839 NEWTON
KROMOSOM DARI GENERASI TERBAIK	= 1111111111111111111111000101010001111010

Cetak

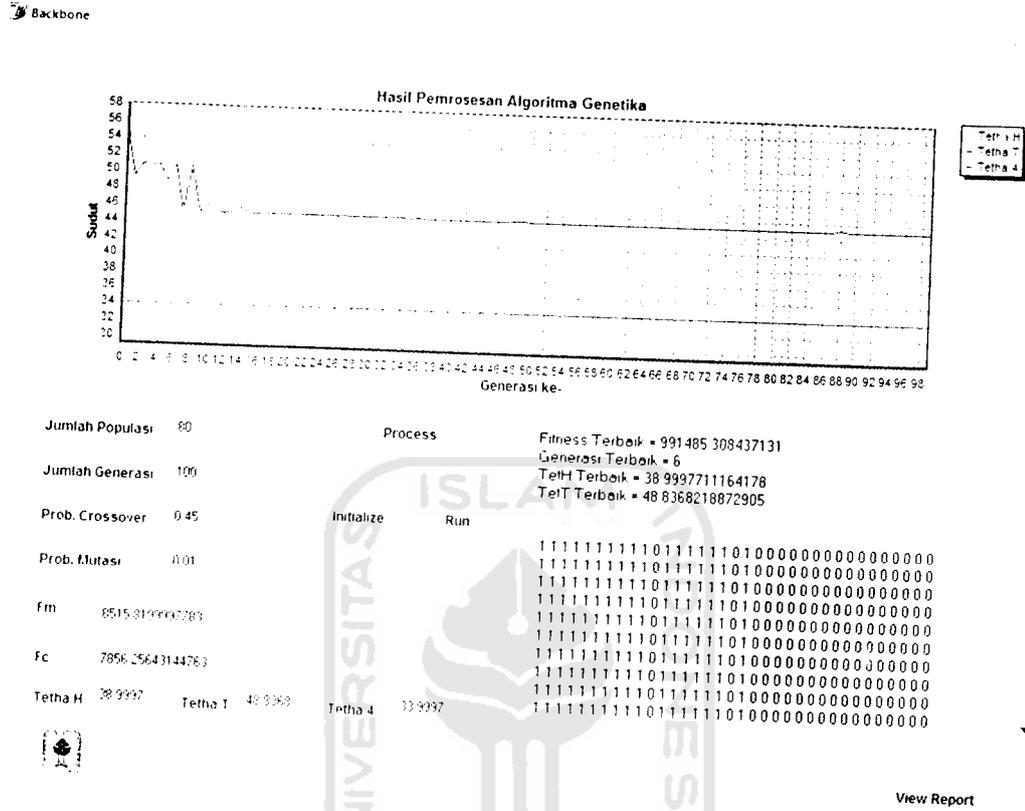
Close

Gambar 6.6 Laporan Hasil Pengujian 1

b. Pengujian 2

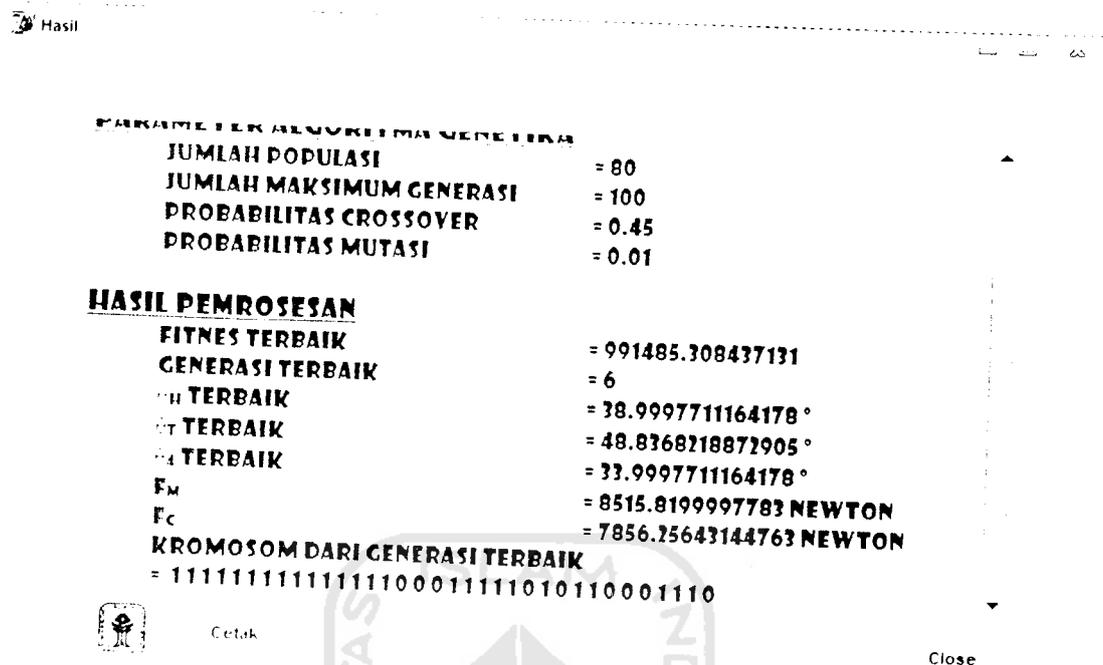
- Pada pengujian ini, masih digunakan jumlah generasi 100.
- Hasil pengujian 2, sudut-sudut yang diperoleh antara lain $\theta_{11} = 38,9997711164178^\circ$, $\theta_T = 48,8368218872905^\circ$, dan $\theta_4 = 33,9997711164178^\circ$.
- Gaya kompresi (F_c) yang dihasilkan adalah 7856,256431447636 Newton.
- Generasi terbaik = 6, nilai *fitness* terbaik = 991485,308437131.

- Proyeksi sudut dan besarnya gaya kompresi (F_c) pada pengujian 2 ditunjukkan pada Gambar 6.7.



Gambar 6.7 Hasil Pengujian 2

- Laporan hasil pemrosesan dapat dilihat pada Gambar 6.8 di bawah ini.

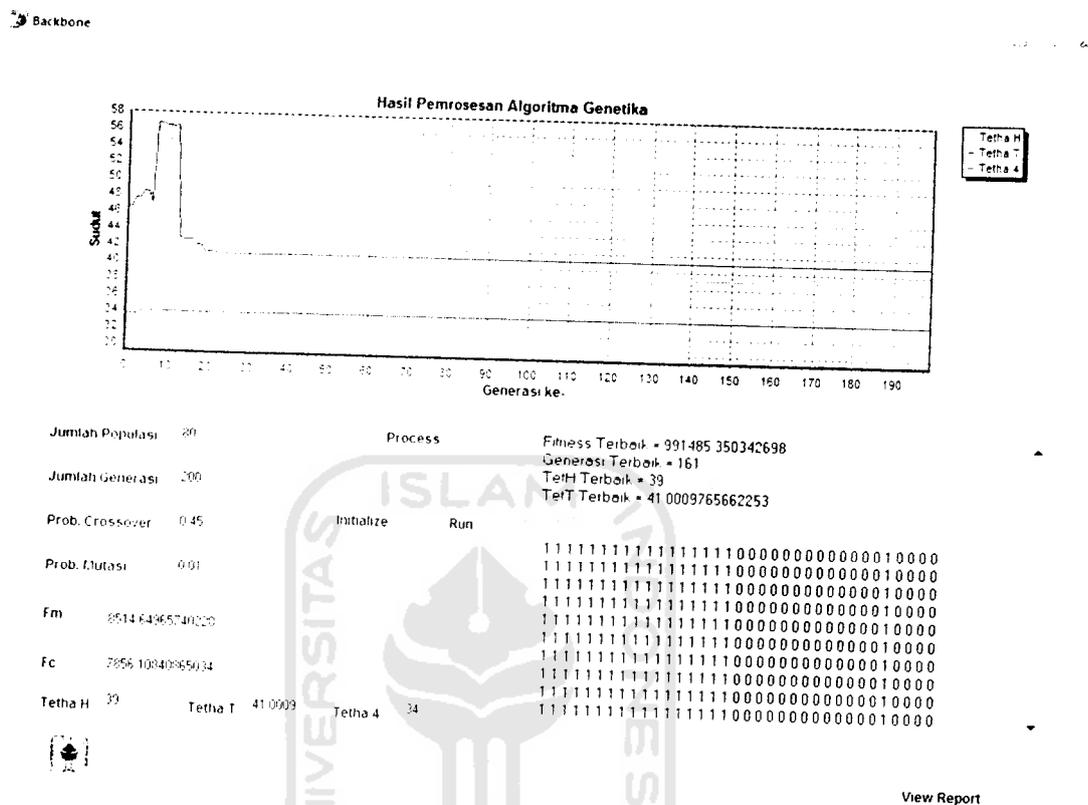


Gambar 6.8 Laporan Hasil Pengujian 2

c. Pengujian 3

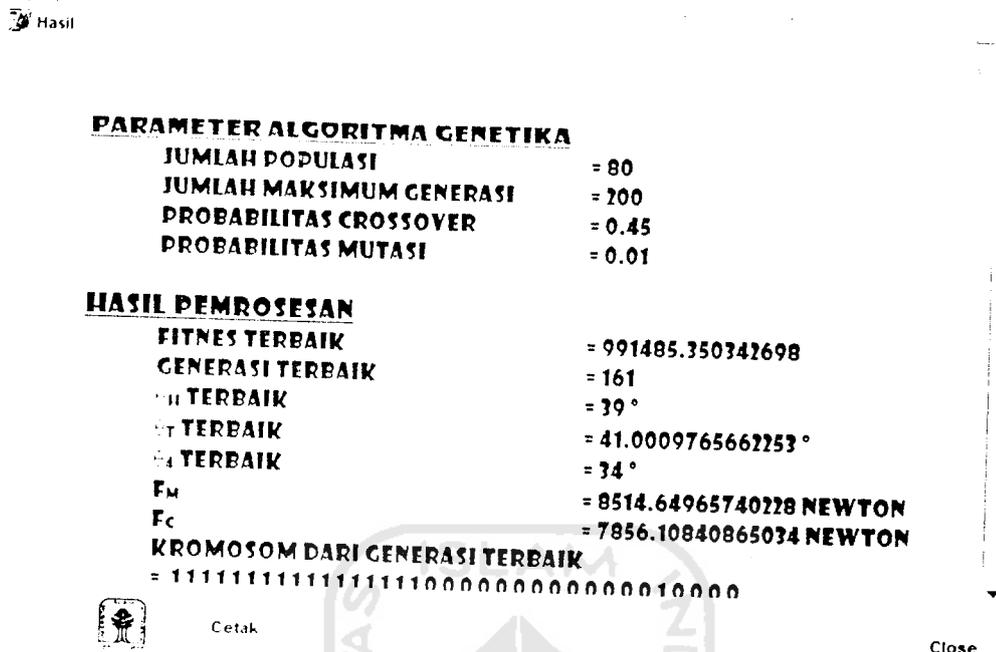
- Pada pengujian ini, digunakan jumlah generasi 200.
- Hasil pengujian 3, sudut-sudut yang diperoleh antara lain $\theta_H = 39^\circ$, $\theta_T = 41,0009765662253^\circ$, dan $\theta_4 = 34^\circ$.
- Gaya kompresi (F_c) yang dihasilkan adalah 7856,10840865034 Newton.
- Generasi terbaik = 161, nilai *fitness* terbaik = 991485,350342968.

- Proyeksi sudut dan besarnya gaya kompresi (F_c) pada pengujian 3 ditunjukkan pada Gambar 6.9.



Gambar 6.8 Hasil Pengujian 3

- Laporan hasil pemrosesan dapat dilihat pada Gambar 6.9 di bawah ini.

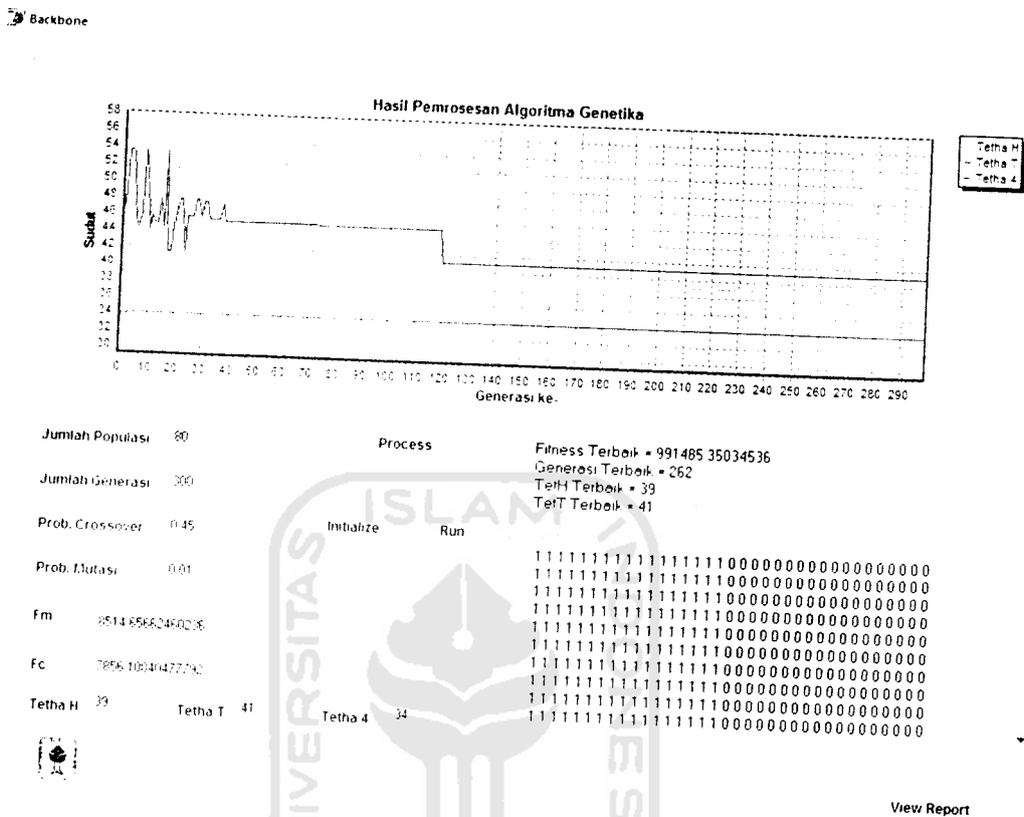


Gambar 6.9 Laporan Hasil Pengujian 3

d. Pengujian 4

- Pada pengujian ini, masih digunakan jumlah generasi 200.
- Hasil pengujian 4, sudut-sudut yang diperoleh antara lain $\theta_H = 39^\circ$, $\theta_T = 41,0002441415563$, dan $\theta_4 = 34^\circ$.
- Gaya kompresi (F_c) yang dihasilkan adalah 7856,10840578243 Newton.
- Generasi terbaik = 138, nilai *fitness* terbaik = 991485,35034467.

- Proyeksi sudut dan besarnya gaya kompresi (F_c) pada pengujian 5 ditunjukkan pada Gambar 6.12.



Gambar 6.12 Hasil Pengujian 5

- Laporan hasil pemrosesan dapat dilihat pada Gambar 6.13 di bawah ini.

Hasil

PARAMETER ALGORITMA GENETIKA

JUMLAH POPULASI	= 80
JUMLAH MAKSIMUM GENERASI	= 300
PROBABILITAS CROSSOVER	= 0.45
PROBABILITAS MUTASI	= 0.01

HASIL PEMROSESAN

FITNES TERBAIK	= 991485.35034536
GENERASI TERBAIK	= 162
θ_H TERBAIK	= 39°
θ_T TERBAIK	= 41°
θ_4 TERBAIK	= 34°
F_m	= 8514.65662460236 NEWTON
F_c	= 7856.10840477792 NEWTON

KROMOSOM DARI GENERASI TERBAIK
 = 111111111111111111111100000000000000000000

Cetak

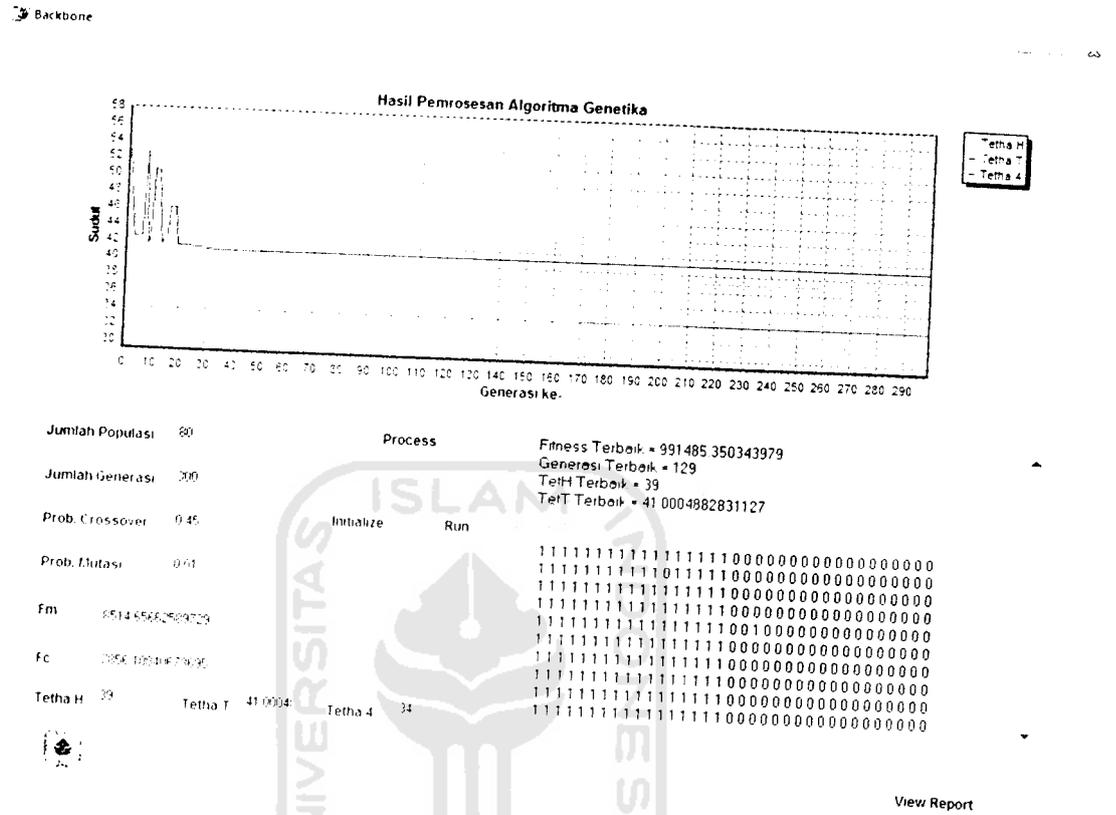
Close

Gambar 6.13 Laporan Hasil Pengujian 5

f. Pengujian 6

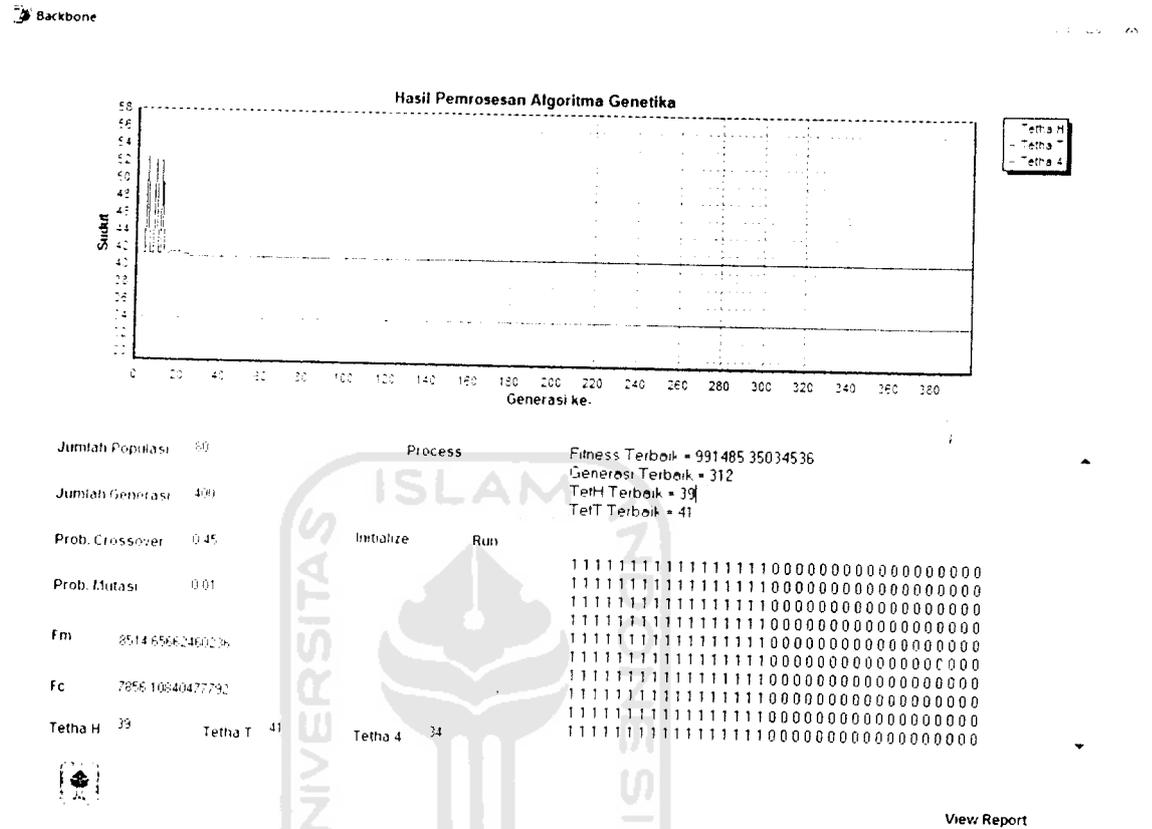
- Pada pengujian ini, masih digunakan jumlah generasi 300.
- Hasil pengujian 6, sudut-sudut yang diperoleh antara lain $\theta_H = 39^\circ$, $\theta_T = 41,0004882831127^\circ$, dan $\theta_4 = 34^\circ$.
- Gaya kompresi (F_c) yang dihasilkan adalah 7856,10840678695 Newton.
- Generasi terbaik = 129, nilai *fitness* terbaik = 991485,35033979.

- Proyeksi sudut dan besarnya gaya kompresi (F_c) pada pengujian 6 ditunjukkan pada Gambar 6.14.



Gambar 6.14 Hasil Pengujian 6

- Proyeksi sudut dan besarnya gaya kompresi (F_c) pada pengujian 7 ditunjukkan pada Gambar 6.16.



Gambar 6.16 Hasil Pengujian 7

- Laporan hasil pemrosesan dapat dilihat pada Gambar 6.17 di bawah ini.

PARAMETER ALGORITMA GENETIKA

JUMLAH POPULASI	= 80
JUMLAH MAKSIMUM GENERASI	= 400
PROBABILITAS CROSSOVER	= 0.45
PROBABILITAS MUTASI	= 0.01

HASIL PEMROSESAN

FITNES TERBAIK	= 991485.35034536
GENERASI TERBAIK	= 312
θ_H TERBAIK	= 39°
θ_T TERBAIK	= 41°
θ_4 TERBAIK	= 34°
F_M	= 8514.65662460236 NEWTON
F_c	= 7856.10840477792 NEWTON

KROMOSOM DARI GENERASI TERBAIK
 = 1111111111111111111100000000000000000000

Cetak Close

Gambar 6.17 Laporan Hasil Pengujian 7

h. Pengujian 8

- Pada pengujian ini, masih digunakan jumlah generasi 400.
- Hasil pengujian 8, sudut-sudut yang diperoleh antara lain $\theta_H = 39^\circ$, $\theta_T = 41,0009765662253^\circ$, dan $\theta_4 = 34^\circ$.
- Gaya kompresi (F_c) yang dihasilkan adalah 7856,10840879598 Newton.
- Generasi terbaik = 214, nilai *fitness* terbaik = 991485,350342598.

- Laporan hasil pemrosesan dapat dilihat pada Gambar 6.19 di bawah ini.

Hasil

PARAMETER ALGORITMA GENETIKA

JUMLAH POPULASI	= 80
JUMLAH MAKSIMUM GENERASI	= 400
PROBABILITAS CROSSOVER	= 0.45
PROBABILITAS MUTASI	= 0.01

HASIL PEMROSESAN

FITNES TERBAIK	= 991485.350342598
GENERASI TERBAIK	= 214
θ_{11} TERBAIK	= 39°
θ_{12} TERBAIK	= 41.0009765662253°
θ_{13} TERBAIK	= 34°
F_M	= 8514.65662719221 NEWTON
F_c	= 7856.10840879598 NEWTON

KROMOSOM DARI GENERASI TERBAIK
 = 111111111111111111111111000000000000010000

Cetak

Close



Gambar 6.19 Laporan Hasil Pengujian 8

6.3 Analisis Pengujian Perangkat Lunak

Tabel 6.1 di bawah ini merupakan tabel yang menunjukkan hasil pengujian normal dengan membandingkan jumlah generasi.

Tabel 6.1 Tabel Hasil Pengujian Perangkat Lunak Tahap I

No.	Max. Gen	θ_H (derajat)	θ_T (derajat)	θ_4 (derajat)	Fc (Newton)	Generasi Terbaik	Fitness Terbaik
1	100	38,9949	42,1583	33,9949	7856,1233	29	991485,0054
2	100	38,9997	48,4549	33,9997	7856,2564	6	991485,3084
3	200	39	41,0009	34	7856,1084	161	991485,3503
4	200	39	41,0002	34	7856,1084	138	991485,3503
5	300	39	41	34	7856,1084	262	991485,3503
6	300	39	41,0004	34	7856,1084	129	991485,3503
7	400	39	41	34	7856,1084	312	991485,3503
8	400	39	41,0009	34	7856,1084	214	991485,3503

Analisis pengujian

1. Pada pengujian 1 dan 2 dengan jumlah generasi sebesar 100 generasi, terdapat perbedaan cukup signifikan pada besarnya sudut θ_T yaitu $42,1583^\circ$ dan $48,4549^\circ$, serta besarnya nilai *fitness* dari individu terbaik yaitu 991485,0054 dan 991485,3804.
2. Hasil pengujian pada pengujian 3 sampai pengujian 8, besar sudut θ_H , θ_4 , dan nilai *fitness* terbaik memiliki besaran nilai yang sama. Sedangkan untuk nilai θ_T hanya terdapat perbedaan yang sangat kecil.

3. Dapat disimpulkan bahwa kinerja perangkat lunak akan lebih optimum apabila jumlah generasi yang digunakan minimum sebesar 200 generasi, sebab pada penggunaan jumlah generasi sebesar 100 generasi, sistem belum dapat menemukan solusi yang optimum.



BAB VII PENUTUP

7.1 Kesimpulan

Setelah dilakukan serangkaian pengujian terhadap sistem yang telah dibuat, maka dapat diambil kesimpulan sebagai berikut.

1. Algoritma genetika dapat diterapkan pada permasalahan optimasi sudut sebagai alternatif pencarian solusi.
2. Hasil dari pemrosesan dengan algoritma genetika untuk setiap kali proses belum tentu menghasilkan solusi yang optimum, walaupun untuk keseluruhan generasi dalam satu kali proses merupakan solusi yang optimum.
3. Kinerja perangkat lunak akan lebih optimum apabila jumlah generasi yang digunakan minimum sebesar 200 generasi. Pada penggunaan jumlah generasi sebesar 100 generasi, sistem belum dapat menemukan solusi yang optimum sebab ruang solusi pencarian kecil.
4. Untuk menghasilkan solusi yang benar-benar optimum dibutuhkan proses pencarian yang berulang-ulang.

7.2 Saran

Berikut ini merupakan beberapa saran untuk pengembangan dan penelitian selanjutnya.

1. Aplikasi ini dapat dikembangkan dengan metode seleksi, *crossover*, dan mutasi yang lain, misal *local selection*, *multi-point crossover*, mutasi bilangan real, dan lain-lain.
2. Aplikasi ini dapat dikembangkan dengan metode pencarian yang lain, misal jaringan saraf tiruan, *tabu search*, dan lain-lain.
3. Antarmuka diharapkan dapat dikembangkan lagi menjadi suatu perantara komunikasi yang lebih mudah dipahami dan digunakan tanpa pelatihan dan pembelajaran khusus.



DAFTAR PUSTAKA

- [DEL03] Delphi Indonesia, <http://www.delphi-id.org/dpr/PNphpBB2-viewtopic-t-1142.pas>, diakses tanggal 21 September 2006.
- [GOL89] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, USA: Addition Wesley Publishing Company, Inc., 1989.
- [HAR02] Haryanto, Totok, *Optimasi Posisi Kemiringan Punggung Pada Aktivitas Pengangkatan Gula Dalam Karung Menggunakan Algoritma Genetik (Studi Kasus Dolog DIY)*, Yogyakarta: Tugas Akhir Jurusan Teknik Industri Universitas Islam Indonesia, 2002.
- [KUR04] Kurniawan, Didik, *Perbaikan Postur Pengangkatan Awal Pada Aktivitas Pengangkatan Batako dengan Algoritma Genetik*, Yogyakarta: Tugas Akhir Jurusan Teknik Industri Universitas Islam Indonesia, 2004.
- [KUS05] Kusumadewi, S., dan Purnomo, H., *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*, Yogyakarta: Graha Ilmu, 2005.
- [MAD03] Madcoms, *Pemrograman Delphi 7*, Yogyakarta: Andi, 2003.
- [MIC92] Michalewicz, Zbigniew, *Genetic Algorithms + Data Structures = Evolution Programs (Third, Revised and Extended Edition)*, Charlotte: Springer, 1992.

- [POH03] Pohlheim, Hartmut, *GEA Toolbox: Genetic and Evolutionary Algorithms: Principal, Methods, and Algorithms*. <http://www.geatbx.com/docu/algeindex-03.html>, diakses tanggal 14 Juni 2006.
- [SAN00] Sandikci, B., *Genetics Algorithms*, <http://www.bilkent.edu.tr/~burhan>, diakses tanggal 17 April 2006.
- [SYA06] Syafutra, Eko, *Penentuan Persamaan Membawa Beban dengan Menggunakan Tas Punggung Berdasarkan Kriteria Fisiologis Pada Aktivitas Naik Gunung*, Bandung: Tugas Akhir Program Studi Teknik Industri ITB, 2006.
- [RAY95] Ray, T., E., *Mutation*, <http://www.nis.atr.jp/~ray/pubs/tierra/node10.html>, diakses tanggal 2 Desember 2006.

