

**OPTIMASI BIAYA PRODUKSI
MENGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

Oleh :
FAHRON DAKKA
0610962002-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2009**

LEMBAR PENGESAHAN SKRIPSI
OPTIMASI BIAYA PRODUKSI MENGGUNAKAN
ALGORITMA GENETIKA

oleh:
FAHRON DAKKA
0610962002-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 13 Februari 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,

Pembimbing II,

Wayan Firdaus M., SSi., MT
NIP. 132 158 724

Agus Wahyu Widodo, ST
NIP. 132 295 994

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc
NIP. 132 126 049

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Fahron Dakka
NIM : 0610962002-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis tugas akhir berjudul : Optimasi Biaya Produksi
Menggunakan Algoritma
Genetika

Dengan ini menyatakan bahwa :

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

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 13 Februari 2009

Yang menyatakan,

Fahron Dakka
NIM. 0610962002

UNIVERSITAS BRAWIJAYA



OPTIMASI BIAYA PRODUKSI MENGGUNAKAN ALGORITMA GENETIKA

ABSTRAK

Tujuan utama perusahaan adalah untuk memaksimalkan laba. Salah satunya cara dalam memaksimalkan laba adalah dengan melakukan optimasi biaya produksi. Bukanlah masalah yang mudah melakukan optimasi biaya produksi pada perusahaan manufaktur mengingat banyaknya produk yang diproduksi dimana tiap produk memerlukan alokasi biaya yang berbeda. Pada algoritma genetika teknik pencarian solusi menggunakan prinsip seleksi alam, dimana individu yang lebih kuat akan memiliki tingkat reproduksi yang lebih tinggi. Pada tugas akhir ini dibahas mengenai optimasi biaya produksi menggunakan algoritma genetika. Penelitian ini bertujuan membuat model genetika untuk masalah optimasi biaya produksi dan menerapkan algoritma genetika sebagai alternatif solusi masalah optimasi biaya produksi dengan menggunakan perangkat lunak. Hasil akhir dari optimasi biaya produksi diperoleh dari kromosom yang memiliki nilai *fitness* terbaik. Berdasarkan uji coba yang telah dilakukan, nilai *fitness* terbaik diperoleh dengan menggunakan nilai probabilitas perkawinan silang 0.2 dan nilai probabilitas mutasi 0.7.

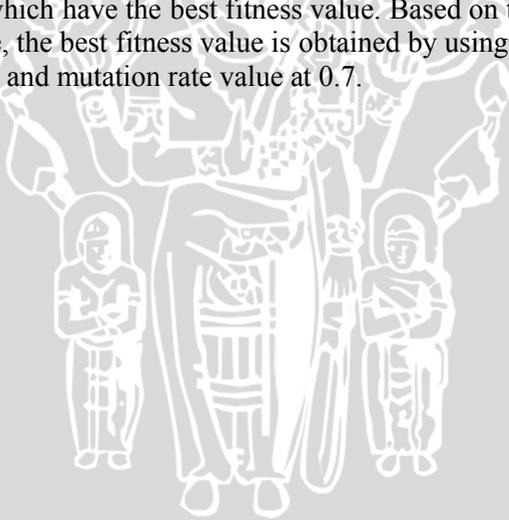
UNIVERSITAS BRAWIJAYA



COST PRODUCTION OPTIMIZATION USING GENETIC ALGORITHM

ABSTRACT

The main purpose of a company is to maximize profit. One of the ways to maximize profit is done by cost production optimization. It is not a simple problem at manufacturing company since there are a lot of products where each of them need different cost allocation. In genetic algorithms, the technique of searching solution uses the principles of natural selection, in which stronger individuals will have higher level of reproduction. This final project discuss about cost production optimization that uses genetic algorithms. This study aims at making a genetic model for problem of cost production optimization and applying genetic algorithms as a solution alternative to cost production optimization problem using software. The final result of this cost production optimization is obtained from chromosomes which have the best fitness value. Based on the testing which was done, the best fitness value is obtained by using crossover rate value at 0.2 and mutation rate value at 0.7.



UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

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

Proposal tugas akhir ini bertujuan untuk menerapkan algoritma genetika sebagai alternatif solusi pemecahan masalah optimasi biaya produksi.

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

1. Bapak Wayan Firdaus Mahmudy, S.Si, MT, selaku pembimbing utama penulisan tugas akhir dan selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
2. Bapak Agus Wahyu Widodo, ST, selaku pembimbing pendamping dalam penulisan tugas akhir.
3. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
4. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan proposal tugas akhir ini.
5. Orang tua Penulis atas dukungan materi dan doa restunya kepada Penulis.
6. Rekan-rekan di Program Studi Ilmu Komputer FMIPA Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan proposal tugas akhir ini.
7. Dan semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak dapat Penulis sebutkan satu per satu.

Penulis sadari bahwa masih banyak kekurangan dalam laporan ini, oleh karena itu Penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi proposal tugas akhir ini untuk kelanjutan penelitian serupa di masa mendatang.

Semoga tugas akhir ini dapat bermanfaat. Amin.

Malang, Februari 2009

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
LEMBAR PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR PSEUDOCODE	xxi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi Pemecahan Masalah	3
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	
2.1 Biaya	5
2.1.1 Pengertian Biaya	5
2.1.2 Klasifikasi Biaya	5
2.1.3 Biaya Produksi	6
A. Biaya Bahan Baku	6
B. Biaya Tenaga Kerja Langsung	6
C. Biaya Overhead Pabrik	6
2.1.4 Optimasi Biaya Produksi	7
2.2 Algoritma Genetika	8
2.2.1 Struktur Umum Algoritma Genetika	8
2.2.2 Pengkodean	9
2.2.3 Fungsi Evaluasi	9
2.2.4 Seleksi	10
A. Seleksi Roda <i>Roulette</i>	10
B. Seleksi Rangkang	11
C. Seleksi Turnamen	12

2.2.5 Operator Genetika	12
A. Perkawinan Silang (<i>Crossover</i>)	12
a. Perkawinan silang satu titik	12
b. Perkawinan silang dua titik	13
c. Perkawinan silang seragam	13
d. Perkawinan silang aritmatik	14
B. Mutasi	14
2.2.5 Parameter Genetika	15
A. Ukuran Populasi	15
B. Probabilitas <i>Crossover</i>	15
C. Probabilitas Mutasi	15

BAB III METODOLOGI DAN PERANCANGAN SISTEM

3.1 Faktor-Faktor yang Mempengaruhi Penentuan Formula Biaya Produksi	17
3.2 Aturan-aturan Penentuan Formula Biaya Produksi	17
3.3 Model Genetika	17
3.4 Inisialisasi Kromosom	20
3.5 Fungsi Evaluasi	21
3.6 Seleksi	22
3.7 Perkawinan Silang	22
3.8 Mutasi	24
3.9 Contoh Perhitungan Manual	24
3.9.1 Inisialisasi Kromosom	25
3.9.2 Evaluasi	25
3.9.3 Seleksi	26
3.9.4 Perkawinan Silang	27
3.9.5 Mutasi	29
3.9.6 Pembentukan Generasi Baru	29
3.10 Data Uji	30

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Program	31
4.1.1 Inisialisasi	32
4.1.2 Optimasi	32
4.2 Deskripsi Program	33
4.2.1 Inisialisasi Kromosom	33
4.2.2 Seleksi	33
4.2.3 Perkawinan Silang	34

4.2.4 Mutasi	35
4.2.5 Perbaikan Kromosom	35
4.2.6 Proses Perhitungan <i>Fitness</i>	36
4.3 Penerapan Aplikasi	36
4.4 Analisa Hasil	38

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	43
5.2 Saran.....	43

DAFTAR PUSTAKA

LAMPIRAN-LAMPIRAN



UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

Gambar 2.1	Pengkodean Nilai	9
Gambar 2.2	Contoh populasi dengan 5 kromosom	10
Gambar 2.3	Probabilitas suatu kromosom dalam roda <i>roulette</i>	10
Gambar 2.4	Keadaan sebelum dirangking	11
Gambar 2.5	Keadaan setelah dirangking.....	11
Gambar 3.1	<i>Flowchart</i> Algoritma Genetika	18
Gambar 3.2	Inisialisai Kromosom	21
Gambar 3.3	<i>Pseudocode repair</i> kromosom biaya produksi.....	23
Gambar 3.4	<i>Pseudocode</i> mutasi kromosom biaya produksi ...	24
Gambar 3.5	Ilustrasi mutasi	24
Gambar 3.2	Model tahap 2.....	21
Gambar 3.3	Skema basis data	23
Gambar 3.4	Ilustrasi perkawinan silang.....	25
Gambar 3.5	Ilustrasi mutasi cara penyisipan	26
Gambar 3.6	Ilustrasi mutasi cara <i>swap</i>	26
Gambar 4.1	Tampilan utama aplikasi	31
Gambar 4.2	<i>Panel</i> inisialisasi.....	32
Gambar 4.3	<i>Panel</i> optimasi	32
Gambar 4.4	Tampilan aplikasi setelah inisialisasi kromosom .	37
Gambar 4.5	Tampilan aplikasi setelah proses optimasi	37
Gambar 4.6	Grafik perbandingan rata-rata <i>fitness</i> dengan <i>crossover rate</i> yang berbeda	39
Gambar 4.7	Grafik perbandingan <i>fitness</i> maksimal dengan <i>crossover rate</i> sebesar 0.2.....	39
Gambar 4.8	Grafik perbandingan rata-rata <i>fitness</i> dengan <i>mutation rate</i> yang berbeda.....	41
Gambar 4.9	Grafik perbandingan <i>fitness</i> maksimal dengan <i>mutation rate</i> sebesar 0.7	41

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Tabel 3.1	Contoh matrik biaya produksi.....	20
Tabel 3.2	Matrik biaya produksi.....	24
Tabel 3.3	Nilai <i>fitness</i> kromosom.....	26
Tabel 3.4	Nilai <i>fitness</i> kromosom setelah terurut.....	26
Tabel 3.5	Nilai <i>Fitness</i> kromosom <i>child</i>	29
Tabel 3.6	Hasil Nilai Fitness Kromosom <i>Parent</i> dan <i>Child</i>	30
Tabel 3.7	Contoh matrik biaya produksi.....	30
Tabel 4.1	Hasil uji <i>crossover rate</i> berbeda.....	38
Tabel 4.2	Hasil uji <i>mutation rate</i> berbeda.....	40
Tabel 4.3	Jumlah produksi tiap produk.....	42



UNIVERSITAS BRAWIJAYA



DAFTAR PSEUDOCODE

<i>Pseudocode</i> 4.1 Prosedur inialisasi kromosom.....	33
<i>Pseudocode</i> 4.2 Prosedur pilih <i>parent</i>	33
<i>Pseudocode</i> 4.3 Prosedur penentuan mask.....	34
<i>Pseudocode</i> 4.4 Prosedur perkawinan silang	34
<i>Pseudocode</i> 4.5 Prosedur mutasi.....	35
<i>Pseudocode</i> 4.6 Prosedur <i>repair</i>	35
<i>Pseudocode</i> 4.7 Prosedur perhitungan <i>fitness</i>	36



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Persaingan yang semakin ketat mengharuskan perusahaan untuk mengambil tindakan yang tepat agar dapat tetap eksis. Oleh karena itu, untuk menjamin kelangsungan hidupnya, perusahaan melaksanakan berbagai kebijaksanaan untuk mencapai tujuan utamanya. Tujuan utama perusahaan secara umum, yaitu untuk memaksimalkan laba yang dapat dicapai, salah satunya melalui optimasi biaya produksi.

Optimasi biaya produksi dapat dicapai dengan cara menentukan formula optimasi biaya produksi. Optimasi yang dimaksud di sini adalah pencarian nilai-nilai variabel yang dianggap optimal, efektif dan efisien untuk mencapai hasil yang diinginkan. Dalam hal ini, formula optimasi diharapkan dapat memaksimalkan laba melalui pengalokasian biaya produksi yang tepat dengan tetap menjaga kualitas barang yang diproduksi.

Pada perusahaan manufaktur, perhitungan atas biaya-biaya yang timbul dengan tepat sangat berguna, sehingga perusahaan diharapkan dapat bersaing dengan perusahaan lain yang menghasilkan produk yang sejenis. Dimana perusahaan dapat menekan biaya produksinya, sehingga menetapkan harga jual yang lebih rendah tanpa mengurangi kualitas produknya.

Dalam melakukan proses produksi, diperlukan adanya beberapa input faktor produksi untuk sistem produksi dalam perusahaan yang bersangkutan. Beberapa input yang diperlukan untuk sistem produksi dalam perusahaan antara lain adalah bahan baku yang dipergunakan dalam proses produksi, tenaga kerja langsung yang diperlukan serta dana yang tersedia untuk modal kerja.

PT. UVW yang bergerak di bidang manufaktur, dimana perusahaan memproduksi berbagai jenis barang dengan alokasi biaya yang berbeda untuk tiap jenis barang yang diproduksi, mengharapkan adanya optimasi dalam pembiayaan produksi. Salah satu metode optimasi adalah algoritma genetika.

Algoritma genetika adalah suatu teknik pencarian solusi dengan menggunakan prinsip seleksi alami (Gen dan Cheng, 1997). Ide awal algoritma ini adalah teori evolusi dalam konsep biologi yang

dikemukakan oleh Charles Darwin. Algoritma genetika dimulai dengan memilih himpunan penyelesaian, yang direpresentasikan dengan kromosom, yang disebut dengan populasi. Solusi dari suatu populasi diambil untuk membentuk populasi baru, dimana pemilihannya tergantung dari nilai *fitness*. Hal ini diharapkan agar populasi baru yang terbentuk akan lebih baik dari populasi terdahulu. Proses ini dilakukan berulang-ulang sampai terpenuhi kondisi tertentu.

Optimasi biaya produksi adalah salah satu contoh dari permasalahan kombinatorial dengan beberapa batasan atau biasa disebut dengan *multiconstraint knapsack problem*. Karena dalam perhitungan biaya produksi, tiap barang yang diproduksi memerlukan beberapa faktor biaya produksi. Setiap faktor biaya produksi tersebut memiliki batasan (*constraint*) dalam proses pemenuhannya. Tujuan dari *multiconstraint knapsack problem* adalah untuk menentukan kombinasi jumlah optimal tiap barang (dalam hal ini adalah barang yang diproduksi) sehingga diperoleh keuntungan yang maksimal.

Dari beberapa penelitian yang telah dilakukan, secara umum dapat disimpulkan bahwa algoritma genetika merupakan metode yang cukup handal dan efektif dalam memecahkan *multiconstraint knapsack problem*. Dalam penelitiannya, Gunther R. Raidl menyebutkan bahwa algoritma genetik mampu menghasilkan solusi yang lebih baik daripada metode heuristik. Oleh karena itu, maka dalam tugas akhir ini dirancang dan dibuat aplikasi optimasi biaya produksi dengan menggunakan algoritma genetika.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka perumusan masalah dalam penelitian ini adalah "Bagaimanakah penerapan algoritma genetika untuk menghasilkan formula biaya produksi yang optimal pada PT. UVW ?".

1.3 Batasan Masalah

Ruang lingkup yang akan membatasi permasalahan yang akan dibahas dalam tugas akhir ini antara lain :

1. Perusahaan yang menjadi obyek penelitian adalah perusahaan manufaktur.
2. Biaya produksi yang menjadi obyek penelitian hanya terbatas pada biaya produksi langsung.

1.4 Tujuan

Sesuai dengan perumusan masalah yang telah disebutkan maka tujuan penelitian adalah untuk menentukan model genetika serta membandingkan nilai fitness yang dihasilkan dari kombinasi *mutation rate* dan *crossover rate* yang berbeda dalam menghasilkan alokasi biaya produksi yang optimal pada PT. UVW.

1.5 Manfaat

Sebagai sumber informasi bagi manajemen perusahaan yang diharapkan dapat memberikan sumbangan pemikiran dan bahan pertimbangan di dalam penentuan biaya produksi seoptimal mungkin sehingga dapat meningkatkan laba perusahaan.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan tugas akhir ini adalah:

1. Studi Literatur
Mempelajari teori-teori yang berhubungan dengan konsep biaya produksi dan algoritma genetik dari berbagai referensi.
2. Pendefinisian dan Analisa Masalah
Mendefinisikan dan menganalisa masalah untuk mencari solusi yang tepat.
3. Perancangan dan Implementasi Sistem
Membuat perancangan perangkat lunak dengan analisa terstruktur dan mengimplementasikan hasil rancangan tersebut.
4. Uji Coba dan Analisa Hasil Implementasi
Menguji perangkat lunak dengan data yang sebenarnya, dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, manfaat dan metode pemecahan masalah serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan algoritma genetik dan optimasi biaya produksi dalam proses produksi.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam menyelesaikan masalah optimasi biaya produksi dengan menggunakan algoritma genetik.

4. BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini berisi tentang penjelasan implementasi sistem dan hasil pengujian yang dilakukan.

5. BAB V PENUTUP

Berisi tentang kesimpulan dari seluruh rangkaian penulisan tugas akhir ini serta kemungkinan saran pengembangannya.

BAB II TINJAUAN PUSTAKA

2.1 Biaya

2.1.1 Pengertian Biaya

Dalam arti luas, biaya adalah pengorbanan sumber ekonomi, yang diukur dalam satuan uang, yang telah terjadi atau kemungkinan akan terjadi untuk tujuan tertentu.

Biaya (*cost*) digunakan untuk mengukur pengorbanan ekonomis yang dilakukan untuk mencapai tujuan organisasi. Untuk suatu produk, biaya menunjukkan ukuran moneter sumber daya yang digunakan, seperti bahan, tenaga kerja dan overhead. Untuk suatu jasa, biaya merupakan pengorbanan moneter yang dilakukan untuk menyediakan jasa (Mulyadi, 1999).

2.1.2 Klasifikasi Biaya

Menurut Hammer, penggolongan biaya dari aktifitas yang terjadi dalam perusahaan, adalah (Mulyadi, 1999):

- a. Biaya Produksi
Semua biaya yang berhubungan dengan fungsi produksi atau kegiatan pengolahan bahan baku menjadi produk selesai.
- b. Biaya pemasaran
Biaya dalam rangka penjualan produk selesai sampai dengan pengumpulan piutang menjadi kas.
- c. Biaya Administrasi Umum
Semua biaya yang berhubungan dengan fungsi administrasi dan umum. Termasuk dalam biaya ini gaji pimpinan tertinggi perusahaan, personalia, sekretariat, akuntansi, hubungan masyarakat dan keamanan.
- d. Biaya Keuangan
Semua biaya yang terjadi dalam melaksanakan fungsi keuangan, misalnya: biaya bunga pinjaman, biaya administrasi bank, kerugian selisih kurs.

2.1.3 Biaya Produksi

Seperti dijelaskan diatas, biaya produksi (*manufacturing cost*) adalah Semua biaya yang berhubungan dengan fungsi produksi atau kegiatan pengolahan bahan baku menjadi produk selesai. Biaya produksi digolongkan ke dalam (Riwayadi, 2006):

A. Biaya Bahan Baku

Biaya bahan baku adalah harga perolehan dari bahan baku yang dipakai di dalam pengolahan produk (Riwayadi, 2006). Biaya bahan baku dapat diklasifikasi menjadi dua, yaitu: biaya bahan baku langsung (*direct raw material cost*) dan biaya bahan baku tidak langsung (*indirect raw material cost*). Bahan baku langsung (*direct raw material*) adalah bahan yang dapat secara mudah dan akurat ditelusuri ke produk jadi. Sedangkan bahan baku tidak langsung (*indirect raw material*) adalah bahan yang tidak secara mudah dan akurat ditelusuri ke produk jadi. Oleh karena tidak mudah menelusurinya ke produk jadi, maka biaya bahan baku tidak langsung (*indirect raw material cost*) diklasifikasikan sebagai biaya overhead pabrik (*factory overhead cost*).

B. Biaya Tenaga Kerja Langsung

Biaya tenaga kerja langsung (*direct labor cost*) adalah balas jasa yang diberikan kepada karyawan pabrik yang manfaatnya dapat diidentifikasi atau diikuti jejaknya pada produk tertentu yang dihasilkan perusahaan. Sama halnya dengan bahan baku langsung, biaya tenaga langsung merupakan biaya produksi langsung.

Sebagai biaya produksi langsung, biaya tenaga kerja langsung dibebankan secara langsung ke produk. Misalnya, bila upah untuk membuat meja sebesar Rp 50.000 per unit, maka upah sebesar Rp 50.000 tersebut dibebankan sepenuhnya ke meja yang dihasilkan.

C. Biaya Overhead Pabrik

Biaya overhead pabrik (*factory overhead cost*) adalah biaya produksi selain biaya bahan baku langsung dan biaya tenaga kerja langsung. Biaya overhead pabrik merupakan biaya produksi tidak langsung, oleh karena itu biaya overhead pabrik tidak dapat langsung dibebankan ke produk.

2.1.4 Optimasi Biaya Produksi

Optimasi adalah ukuran dari hubungan antara masukan dan keluaran. Dengan optimasi akan menggambarkan berapa banyak masukan (*input*) yang diberikan untuk menghasilkan suatu unit keluaran (*output*). Disini unit yang paling optimal adalah unit yang dapat menghasilkan jumlah keluaran terbanyak dengan menggunakan masukan yang tersedia.

Dengan melakukan optimasi terhadap biaya produksi. Diharapkan didapatkan laba perusahaan yang lebih besar.

Misalkan ditentukan matrik biaya produksi sebagai berikut:

Biaya Produksi	Produk			Alokasi Maks
	A	B	C	
1	5	4	2	600
2	1	4	2	500
3	2	2	3	500
Profit	20	35	50	

Dengan menggunakan metode simpleks, sebagai salah satu metode pemecahan masalah optimasi, matrik biaya produksi diatas dapat dimodelkan menjadi berikut:

$$\begin{aligned} 5A + 4B + 2C &\leq 600 \\ 1A + 4B + 2C &\leq 500 \\ 2A + 2B + 3C &\leq 500 \\ Z(\text{profit}) &= 20A + 35B + 50C \rightarrow \text{maks} \end{aligned}$$

Setelah proses perhitungan akan didapatkan hasil akhir seperti berikut:

$$B = 62,5; C = 125; Z = 8437,5$$

Hasil diatas berarti bahwa, keuntungan maksimal (sejumlah 8437,5) dapat diraih dengan cara memproduksi produk B sebanyak 62,5 unit dan produk C sebanyak 125 unit.

Metode Simpleks adalah metode dalam pemrograman linear, dengan kata lain metode Simpleks hanya bisa digunakan dalam permasalahan optimasi yang linear.

Berbeda dengan metode Simpleks, Algoritma genetika dapat digunakan untuk menyelesaikan permasalahan optimasi, baik yang bersifat linear maupun non linear.

2.2 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme seleksi alami dan genetika alami (Gen dan Cheng, 1997).

Konsep dasar yang mengilhami timbulnya algoritma genetika adalah teori evolusi alam yang dikemukakan oleh Charles Darwin. Dalam teori tersebut dijelaskan bahwa pada proses evolusi alami, setiap individu harus melakukan adaptasi terhadap lingkungan disekitarnya agar dapat bertahan hidup.

Individu yang lebih kuat akan memiliki tingkat ketahanan yang lebih tinggi jika dibandingkan dengan individu yang kurang fit. Sehingga populasi secara keseluruhannya akan lebih banyak memuat organisme yang fit (Kusumadewi dan Purnomo, 2005).

2.2.1 Struktur Umum Algoritma Genetika

Algoritma genetika merupakan teknik pencarian yang didasarkan pada mekanisme seleksi dan genetika alami. Berbeda dengan teknik pencarian konvensional, algoritma genetika dimulai dengan membentuk kumpulan solusi (kromosom) yang dinamakan populasi. Solusi dari suatu populasi akan diambil untuk membentuk populasi baru. Solusi ini diambil berdasarkan nilai *fitness*. Proses pembentukan populasi baru ini dilakukan dengan harapan bahwa populasi baru yang terbentuk merupakan populasi yang lebih baik dari sebelumnya.

Sehingga secara umum, sebuah algoritma genetika (Gen dan Cheng, 1997) mempunyai lima komponen dasar, yaitu:

1. Representasi kromosom berdasarkan permasalahan.
2. Bagaimana cara untuk membentuk initial populasi.
3. Mengevaluasi kromosom berdasarkan nilai *fitness*.
4. Operator genetika yang merubah komposisi kromosom anak selama proses reproduksi.
5. Nilai parameter dari algoritma genetika.

2.2.2 Pengkodean

Langkah pertama yang dilakukan pada algoritma genetika adalah menerjemahkan atau merepresentasikan masalah riil menjadi terminologi biologi. Cara untuk merepresentasikan masalah ke dalam bentuk kromosom disebut pengkodean. Dalam permasalahan ini akan dipergunakan pengkodean Nilai.

Pengkodean ini dapat digunakan untuk masalah yang kompleks, dimana nilai yang dikodekan langsung merupakan representasi dari masalah. Pada pengkodean nilai, setiap kromosom adalah barisan dari beberapa nilai. Nilai dapat berupa apa saja, seperti bilangan biasa, bilangan riil, karakter sampai dengan obyek-obyek yang rumit (Michalewicz, 1996). Contoh dari pengkodean nilai dapat dilihat pada Gambar 2.1.

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

Gambar 2.1 Pengkodean nilai

2.2.3 Fungsi Evaluasi

Suatu individu dievaluasi berdasarkan fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi h (maksimasi), maka nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yakni $f = h$. Tetapi jika masalahnya adalah meminimalkan fungsi h (minimasi), maka fungsi h tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu nilai *fitness* yang bisa digunakan adalah $f = 1/h$. Tetapi hal ini akan menjadi masalah jika h bernilai 0, yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasinya, h perlu ditambah sebuah bilangan yang dianggap sangat kecil sehingga nilai *fitness* menjadi:

$$f = \frac{1}{(h + a)} \quad (2.1)$$

Dimana a adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan (Suyanto, 2005).

2.2.4 Seleksi

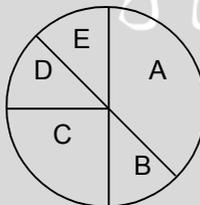
Proses seleksi bertanggung jawab untuk melakukan pemilihan terhadap individu yang hendak diikuti dalam proses reproduksi. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Seleksi mempunyai tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai nilai *fitness* terbaik. Ada beberapa metode seleksi dari induk, antara lain:

A. Seleksi Roda Roulette

Metode seleksi roda *roulette* (Suyanto, 2005) merupakan suatu metode yang menirukan permainan *roulette-wheel* dimana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*-nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah. Gambar 2.2 dan Gambar 2.3 mengilustrasikan contoh penggunaan metode roda *roulette*.

Kromosom	<i>Fitness</i>
A	15
B	5
C	10
D	5
E	5

Gambar 2.2 Contoh populasi dengan 5 kromosom



Gambar 2.3 Probabilitas suatu kromosom dalam roda *roulette*

Gambar 2.2 merupakan contoh dimana dalam satu populasi terdiri dari lima kromosom. Pada tiap kromosom memiliki nilai *fitness* yang berbeda-beda. Pada Gambar 2.3 dapat diketahui probabilitas terpilihnya masing-masing kromosom untuk menjadi induk. Kromosom A memiliki nilai *fitness* 15 dan nilai tersebut adalah nilai *fitness* tertinggi pada populasi. Sehingga kromosom A memiliki probabilitas terbesar untuk terpilih menjadi induk.

Metode seleksi roda *roulette* akan bermasalah saat terdapat perbedaan *fitness* yang besar. Sebagai contoh, jika *fitness* kromosom yang terbaik adalah 90% dari semua roda *roulette* dapat menyebabkan kromosom lain memiliki kesempatan yang kecil untuk dapat terpilih.

B. Seleksi Rangking

Seleksi rangking pertama-tama dimulai dengan merangking kromosom berdasarkan nilai *fitness*nya dan kemudian setiap kromosom yang telah terurut tersebut akan menerima nilai *fitness* yang baru sesuai dengan urutannya. Kromosom yang mempunyai nilai *fitness* paling jelek akan mendapatkan nilai 1 untuk *fitness* yang baru, begitu seterusnya. Dan kromosom dengan nilai *fitness* terbaik akan mempunyai nilai *fitness* N (sesuai dengan jumlah kromosom dalam populasi). Contoh seleksi rangking dapat dilihat pada Gambar 2.4 dan Gambar 2.5

Kromosom	<i>Fitness</i>
A	15
B	5
C	10
D	5

Gambar 2.4 Keadaan sebelum dirangking

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

Gambar 2.5 Keadaan setelah dirangking

Setelah adanya proses seleksi, maka saat ini seluruh kromosom memiliki kesempatan untuk dipilih. Akan tetapi, metode ini dapat menyebabkan konvergensi menjadi lambat, karena kromosom terbaik tidak terlalu berbeda dengan yang lainnya.

C. Seleksi Turnamen

Metode seleksi turnamen (Suyanto, 2005) diadaptasi dari seleksi alami yang terjadi di dunia nyata. Metode ini akan mengambil dua kromosom secara acak dan kemudian akan menyeleksi, kromosom yang terpilih menjadi orang tua pertama adalah kromosom yang mempunyai nilai *fitness* lebih besar. Cara yang sama juga digunakan untuk mendapatkan orang tua yang kedua.

2.2.5 Operator Genetika

Setelah proses seleksi dilakukan, proses selanjutnya adalah operasi genetika. Terdapat 2 operator genetika, yaitu perkawinan silang dan mutasi.

A. Perkawinan Silang (*Crossover*)

Setelah menentukan metode pengkodean apa yang dipakai, maka langkah selanjutnya adalah perkawinan silang atau *crossover*. Proses perkawinan silang (*crossover*) berfungsi untuk menghasilkan keturunan dari dua buah kromosom induk yang terpilih. Kromosom anak yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom induk.

a. Perkawinan silang satu titik

Proses perkawinan satu titik dimulai dengan memilih satu titik pada barisan bit kromosom secara acak sebagai titik perkawinan silang. Kromosom baru akan dibentuk dengan cara menyalin barisan bit induk pertama dari bit pertama sampai titik perkawinan silang, sedangkan sisanya disalin dari induk kedua.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 01110 | 0101110

Induk 2: 11010 | 0001101

Posisi penyilangan yang terpilih: misalkan 5
Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 01110 | 0001101

Anak 2: 11010 | 0101110

b. Perkawinan silang dua titik

Perkawinan silang dua titik dimulai dengan menentukan secara acak dua titik yang akan disilangkan. Kromosom baru akan dibentuk dengan cara menyalin barisan bit kromosom induk pertama dari bit pertama sampai dengan dengan titik perkawinan silang pertama dari titik perkawinan silang kedua sampai dengan bit terakhir, sedangkan sisanya, yaitu titik perkawinan silang pertama sampai titik perkawinan silang kedua disalin dari induk kedua.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 011100101110

Induk 2: 110100001101

Posisi penyilangan yang terpilih: misalkan 2 dan 6

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 01 | 0100 | 101110

Anak 2: 11 | 1100 | 001101

c. Perkawinan silang seragam

Sebuah *mask* penyilangan dibuat sepanjang kromosom secara random yang menunjukkan bit-bit dalam *mask* yang mana induk akan mensuplai anak dengan bit-bit yang ada. Disini, *anak_1* akan dihasilkan dari *induk_1* jika bit *mask* bernilai 1, dan *anak_1* akan dihasilkan dari *induk_2* jika bit *mask* bernilai 0. Sedangkan *anak_2* dihasilkan dari kebalikan *mask*.

Contoh:

Misalkan ada 2 kromosom dengan panjang 12:

Induk 1: 011100101110

Induk 2: 110100001101

Mask bit: 1001110011010

Setelah penyilangan, diperoleh kromosom baru:

Anak 1: 010100001100

Anak 2: 111100101111

d. Perkawinan silang aritmatik

Pada perkawinan silang aritmatik, digunakan beberapa operator aritmatika untuk mendapatkan *offspring* yang baru.

Contoh:

Induk 1 : 11001011

Induk 2 : 11011111

Anak (AND) : 11001011

B. Mutasi

Setelah melalui proses perkawinan silang, pada *offspring* dapat dilakukan proses mutasi. Mutasi dilakukan dengan cara melakukan perubahan pada sebuah gen atau lebih dari sebuah individu.

Tujuan dari mutasi adalah untuk membentuk individu-individu dengan fitur superior atau memiliki kualitas di atas rata-rata. Selain itu mutasi dipergunakan untuk mengembalikan kerusakan materi genetika akibat proses *crossover*. Mutasi sangat berguna dalam mempertahankan keanekaragaman individu dalam populasi meskipun dengan mutasi tidak dapat diketahui apa yang terjadi pada individu baru.

Cara paling sederhana untuk melakukan mutasi adalah dengan mengubah satu atau lebih gen (bagian dari kromosom) dengan probabilitasnya tergantung pada *mutation rate*. Operasi ini dimulai dengan memilih individu dari populasi berdasarkan nilai *fitness*nya. Akan dipilih sebuah titik pada kromosom secara *random* dan gen kromosom pada titik tersebut akan dirubah nilainya secara *random*. Perubahan individu yang dihasilkan tersebut kemudian akan dimasukkan ke dalam populasi.

2.2.6 Parameter Genetika

Terdapat dua parameter dasar dari algoritma genetika, yaitu probabilitas *crossover* dan probabilitas mutasi. Selain itu juga terdapat parameter lain yang juga tidak kalah penting, yaitu ukuran populasi.

A. Ukuran Populasi

Ukuran populasi merupakan parameter yang berfungsi untuk menentukan jumlah kromosom yang berada pada populasi setiap generasi. Jika kromosom yang ada terlalu sedikit, algoritma genetika hanya memiliki sedikit kemungkinan untuk melakukan *crossover*. Juga sebaliknya, jika terdapat terlalu banyak kromosom, maka *performance* dari algoritma genetika juga akan menjadi lambat. Penelitian menunjukkan bahwa pada batasan tertentu, meningkatkan ukuran populasi akan menjadi sangat tidak berguna, karena pemecahan masalah tidak akan berjalan lebih cepat (Obitko, 1998).

B. Probabilitas Crossover

Probabilitas *crossover* atau juga biasa disebut dengan *crossover rate* (P_c), berfungsi untuk menentukan seberapa sering proses *crossover* dilakukan (Obitko, 1998). Semakin besar probabilitas *crossover* akan memungkinkan pencapaian alternatif solusi yang lebih bervariasi dan mengurangi kemungkinan menghasilkan nilai optimum yang tidak dikehendaki. Tetapi bila nilai ini terlalu tinggi akan mengakibatkan pemborosan waktu untuk melakukan perhitungan di daerah solusi yang tidak menjanjikan hasil yang optimal.

C. Probabilitas Mutasi

Parameter ini menunjukkan seberapa sering sebuah kromosom akan dimutasi. Jika tidak terdapat mutasi, maka semua anak akan berasal dari proses *crossover* tanpa adanya perubahan. Jika terdapat proses mutasi, maka sebagian dari kromosom tersebut akan berubah. Jika probabilitas mutasi adalah 100%, maka perubahan akan terjadi pada seluruh kromosom, jika sebaliknya, maka tidak akan ada yang berubah pada kromosom (Obitko, 1998).

UNIVERSITAS BRAWIJAYA



BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Faktor-Faktor yang Mempengaruhi Penentuan Formula Biaya Produksi

Beberapa komponen yang mempengaruhi dalam penentuan formula biaya produksi optimal, antara lain:

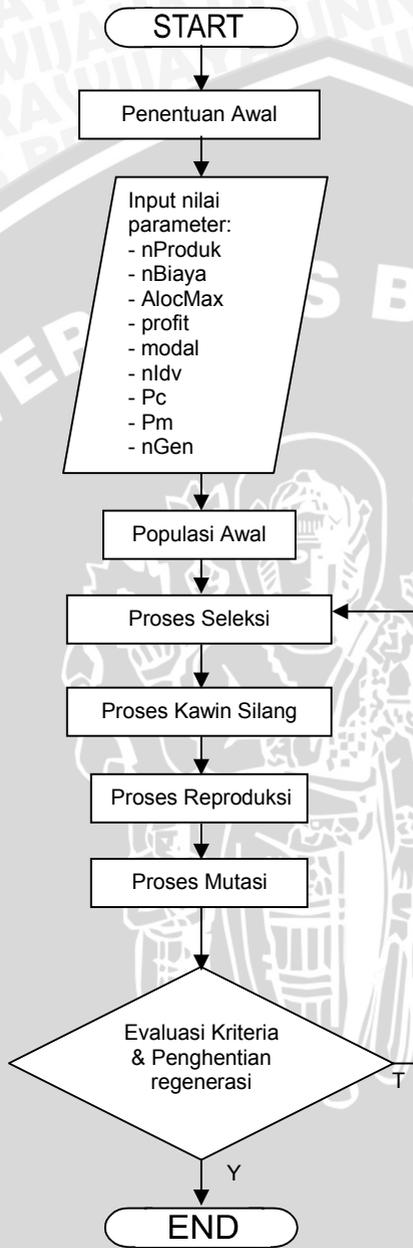
1. Produk
Banyaknya jumlah produk yang diproduksi.
2. Biaya Produksi
Banyaknya jenis biaya produksi yang dibutuhkan dari setiap produk yang diproduksi.
3. Alokasi Biaya Produksi
Jumlah alikasi maksimal untuk setiap biaya produksi.
4. Modal Produksi
Jumlah modal yang tersedia dalam melakukan proses produksi.
5. Keuntungan
Jumlah keuntungan untuk setiap produk yang diproduksi.

3.2 Aturan-aturan Penentuan Formula Biaya Produksi

1. Jumlah total pembiayaan produksi tidak boleh melebihi jumlah modal produksi.
2. Setiap produk yang diproduksi memerlukan beberapa biaya produksi.
3. Jumlah setiap biaya produksi tidak boleh melebihi alokasi maksimal biaya produksi.

3.3 Model Genetika

Pada dasarnya algoritma genetika adalah algoritma pencarian solusi yang terbaik atau yang mendekati terbaik dari begitu banyak solusi yang ada. Agar algoritma genetika menjadi lebih jelas dalam susunan atau urutan prosesnya, maka perlu dibuat *flowchart* (diagram alir). *Flowchart* dari cara kerja algoritma genetika dapat dilihat pada Gambar 3.1



Gambar 3.1 Flowchart Algoritma Genetika

Flowchart tersebut akan dijelaskan secara umum sebagai berikut, yaitu:

1. Penentuan nilai awal
Penentuan nilai awal ini dilakukan pada beberapa bagian yaitu penentuan nilai awal dari panjang kromosom dan penentuan cara teknik dekode.
2. Input nilai parameter
Pada bagian input ini merupakan keinginan dari *user* dalam menentukan hal-hal yang akan dicari optimasinya. Input yang akan dimasukkan dalam *software* ini meliputi bermacam-macam hal, yaitu:
 - Memasukkan banyaknya produk yang akan diproduksi
 - Memasukkan banyaknya biaya produksi
 - Memasukkan jumlah biaya produksi untuk setiap produk yang diproduksi
 - Memasukkan jumlah alokasi maksimal untuk setiap produk yang diproduksi
 - Memasukkan jumlah keuntungan untuk setiap produk yang diproduksi
 - Memasukkan modal produksi
 - Memasukkan jumlah individu dalam satu populasi
 - Memasukkan *crossover rate* (nilai kawin silang)
 - Memasukkan *mutation rate* (nilai mutasi)
 - Memasukkan jumlah generasi
3. Populasi awal
Populasi awal yang dipakai dapat bermacam-macam dan besarnya juga dapat diatur sesuai kebutuhan. Jumlah kromosom tiap populasi mengikuti nilai populasi awal.
4. Proses seleksi
Setelah terjadi populasi awal, maka selanjutnya hasil populasi awal itu akan diseleksi. Metode seleksi yang digunakan di dalam algoritma genetika ini adalah metode seleksi *roulette-wheel*.
5. Proses kawin silang
Proses kawin silang pada tugas akhir ini nilainya berkisar antara nol sampai satu.

6. Proses reproduksi
Proses reproduksi merupakan suatu proses dimana kromosom *parent* sama dengan kromosom *child* sehingga proses reproduksi ini tidak mengubah sifat dari kromosom *parent*.
7. Proses mutasi
Proses mutasi pada tugas akhir ini nilainya berkisar antara nol sampai satu.
8. Kriteria penghentian regenerasi
Kriteria penghentian regenerasi ini dilakukan jika tidak terdapat nilai optimal yang baru setelah terjadi sebanyak beberapa generasi.

3.4 Inisialisasi Kromosom

Penentuan formula biaya produksi pada algoritma genetika dilakukan sekaligus atas sejumlah solusi yang mungkin terjadi yang dikenal sebagai kromosom. Individu-individu yang terdapat dalam suatu populasi disebut kromosom.

Inisialisasi kromosom bergantung pada masalah yang akan diselesaikan. Pengkodean yang digunakan untuk menentukan formula biaya produksi yaitu pengkodean nilai. Masing-masing kromosom berisi *gen-gen* yang merepresentasikan produk yang akan diproduksi. Nilai pada setiap *gen* merepresentasikan jumlah dari masing-masing produk yang akan diproduksi. Sebagai contoh, misalkan ada 4 produk yang diproduksi dan 3 biaya produksi dengan jumlah yang dibutuhkan untuk setiap produk yang diproduksi seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Contoh Matrik Biaya Produksi

Biaya Produksi	Produk				Alokasi Maks
	A	B	C	D	
1	5	2	3	4	200
2	5	3	2	1	100
3	2	4	5	7	150
Profit	10	5	15	15	

Pada tabel diatas, alokasi maksimal adalah jumlah biaya maksimal yang dapat dialokasikan untuk setiap produk yang diproduksi, yang diambil dari modal produksi. Dengan demikian,

modal produksi adalah jumlah total alokasi maksimal setiap produk, dalam hal ini adalah 450. Bentuk tabel di atas dapat direpresentasikan dalam bentuk kromosom sehingga menjadi seperti pada Gambar 3.2.

1	4	2	3
---	---	---	---

Gambar 3.2 Inisialisai Kromosom

Pada gambar 3.2, sel 1 berisi nilai 1 berarti jumlah yang akan diproduksi untuk produk A sejumlah 1 buah, bila dipetakan ke matrik biaya produksi, maka biaya produksi yang dibutuhkan untuk produk A sebesar 12 (berasal dari: $1 \times (5+5+2)$), dengan *profit* sebesar 10 (berasal dari: 1×10). Sel 2 berisi nilai 4 berarti biaya produksi yang dibutuhkan untuk produk B sebesar 36 (berasal dari: $4 \times (2+3+4)$) dengan *profit* sebesar 20 (berasal dari: 4×5), dan seterusnya.

3.5 Fungsi Evaluasi

Setelah individu-individu dalam populasi telah terbentuk, maka langkah selanjutnya adalah menghitung nilai *fitness* setiap individu. Fungsi *fitness* sendiri bertujuan untuk mengetahui baik tidaknya solusi yang ada pada suatu individu.

Setelah kromosom terbentuk, maka selanjutnya dilakukan proses perhitungan *fitness*. Aturan perhitungan fungsi *fitness* yang digunakan adalah sebagai berikut:

$$fitness = totalProfit \quad (3.1)$$

Dimana *totalProfit* adalah jumlah keseluruhan *profit* (keuntungan) dari setiap produk yang diproduksi.

Kromosom dengan nilai *fitness* terbesar akan terpilih sebagai kromosom terbaik. Metode seleksi yang digunakan adalah seleksi roda *roulette*, dimana nilai *fitness* terbesar mempunyai kemungkinan lebih besar untuk menjadi induk pada generasi selanjutnya. Sedangkan untuk nilai *fitness* terburuk akan digantikan oleh individu baru dengan nilai *fitness* yang lebih besar.

3.6 Seleksi

Setelah inialisasi awal, dilakukan proses pencarian *fitness* dengan nilai terbesar, hanya individu yang memiliki nilai *fitness* terbaik yang dapat bertahan dan lolos seleksi untuk proses lebih lanjut. Setelah dilakukan proses rekombinasi dan mutasi, sistem akan kembali melakukan pencarian nilai *fitness* dan individu yang mempunyai nilai *fitness* terbaik adalah individu yang lolos seleksi, sampai tercapai nilai yang terbaik atau telah mencapai generasi tertentu.

3.7 Perkawinan Silang

Apabila proses seleksi telah dilaksanakan dan sudah terpilih induk baru, maka operator berikutnya adalah perkawinan silang. Perkawinan silang adalah cara mengkombinasikan *gen-gen* induk untuk menghasilkan keturunan baru. Perkawinan silang yang digunakan adalah *uniform crossover* untuk tiap *stage* dari kromosom.

Hasil kromosom perkawinan silang adalah perpaduan *gen* dari dua *parent* yang terlibat berdasarkan *mask bit* yang telah ditentukan.

Langkah-langkah pada perkawinan silang ini adalah sebagai berikut:

Parent 1

3	4	1	6
---	---	---	---

Parent 2

2	5	3	2
---	---	---	---

Langkah 1:

Penentuan *mask bit*.

1	0	0	1
---	---	---	---

Langkah 2:

Menyalin *gen* dari induk 1 dan Induk 2 ke Anak 1 dan Anak 2 sesuai dengan *mask bit* yang telah ditentukan. Disini, apabila *mask bit* bernilai 1 maka *gen* Anak 1 akan disalinkan dari *gen* Parent 1 dan *gen* Anak 2 akan disalinkan dari *gen* Parent 2, dan apabila *mask bit* bernilai 0 maka *gen* Anak 1 akan disalinkan dari *gen* Parent 2 dan *gen* Anak 2 akan disalinkan dari *gen* Parent 1

Anak 1

3	5	3	6
---	---	---	---

Anak 2

2	4	1	2
---	---	---	---

Langkah 3:

Hasil dari perkawinan silang memungkinkan munculnya *child* yang *invalid*, sehingga harus dilakukan *repair*. Dikatakan *invalid* dengan ketentuan sebagai berikut:

- Jumlah total biaya dari kromosom yang dihasilkan lebih besar dari jumlah modal produksi.
- Jumlah biaya produksi lebih besar dari alokasi maksimal biaya produksi.

Pseudocode repair kromosom dapat dilihat pada Gambar 3.3

procedure Repair (Pop)

```
repeat  
  chromosome[Idx] := RandomRange(1, (chromosome[gIdx]-1))  
until (valid(chromosome))
```

function valid(factor):boolean

```
result := true  
for i from 1 to nFactor do  
  sum := 0  
  for j from 1 to nGen do  
    sum := sum+CostMatrix[i,j]  
    if (sum>cost[i])  
      return false  
    exit  
  end if  
end for
```

```
sum ← 0  
for i ← 1 to nCost do  
  for j ← 1 to nGen do  
    sum ← sum+CostMatrix[i,j]  
  end for  
  if (sum>capital)  
    result ← false  
    exit  
  end if
```

Gambar 3.3 *Pseudocode repair* kromosom biaya produksi

Pada Gambar 3.3 dapat dilihat bahwa proses *repair* akan terus diulang sampai dipenuhi syarat *valid* sebuah kromosom.

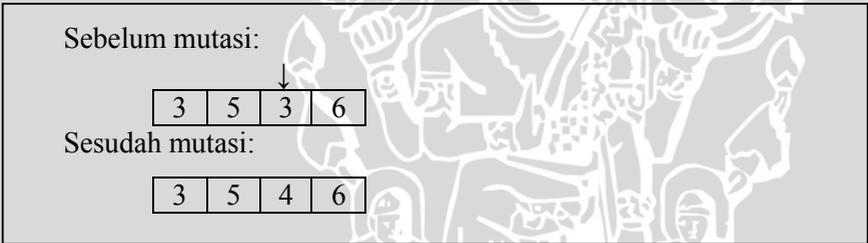
Setelah proses perkawinan silang maka dilakukan perhitungan *cost* untuk memperoleh nilai *fitness*.

3.8 Mutasi

Mutasi dilakukan dengan cara mengganti nilai dari gen yang sudah terpilih dengan nilai acak. Pemilihan *gen* yang akan diganti dilakukan secara random. *Pseudocode* mutasi kromosom dapat dilihat pada Gambar 3.4. Sedangkan untuk ilustrasi mutasi dapat dilihat pada Gambar 3.5.

```
procedure Mutation (Pop)
  genIdx := RandomRange(1, (nGen*nIdv))
  repeat
    chromosome[Idx] := random()
  until (valid(chromosome))
```

Gambar 3.4 *Pseudocode* mutasi kromosom biaya produksi



Gambar 3.5 Ilustrasi mutasi

3.9 Contoh Perhitungan Manual

Misal dalam sebuah perusahaan ditentukan matrik biaya produksi seperti ditunjukkan pada Tabel 3.2 berikut:

Tabel 3.2 Matrik biaya produksi

Biaya Produksi	Produk				Alokasi Maks
	A	B	C	D	
1	5	2	3	4	100
2	5	3	2	1	100
3	2	4	5	7	150
Profit	5	5	15	10	

3.9.1 Inisialisasi Kromosom

Kromosom 1

5	8	10	6
---	---	----	---

Kromosom 2

2	1	6	9
---	---	---	---

Kromosom 3

5	7	8	10
---	---	---	----

3.9.2 Evaluasi

Setelah melakukan proses inisialisasi kromosom, selanjutnya adalah melakukan pengecekan, untuk mengetahui apakah kromosom-kromosom tersebut telah memenuhi persyaratan atau tidak.

Proses evaluasi juga menghitung besarnya nilai *fitness* untuk masing-masing kromosom atau individu. Perhitungan nilai *fitness* adalah sebagai berikut:

Kromosom 1

5	8	10	6
---	---	----	---

Biaya untuk setiap biaya produksi:

$$\text{Biaya1} = (5*5) + (8*2) + (10*3) + (6*4) = 25 + 16 + 30 + 24 = 95$$

$$\text{Biaya2} = (5*5) + (8*3) + (10*2) + (6*1) = 25 + 24 + 20 + 6 = 75$$

$$\text{Biaya3} = (5*2) + (8*4) + (10*5) + (6*7) = 10 + 32 + 50 + 42 = 134$$

$$\begin{aligned} \text{Total biaya produksi} &= \text{Biaya1} + \text{Biaya1} + \text{Biaya3} \\ &= 95 + 75 + 134 = 304 \end{aligned}$$

$$\begin{aligned} \text{fitness} = \text{profit} &= (5*5) + (8*5) + (10*15) + (6*10) \\ &= 25 + 40 + 150 + 60 = 275 \end{aligned}$$

Kromosom 2

2	1	6	9
---	---	---	---

Biaya untuk setiap biaya produksi:

$$\text{Biaya1} = (2*5) + (1*2) + (6*3) + (9*4) = 10 + 2 + 18 + 36 = 66$$

$$\text{Biaya2} = (2*5) + (1*3) + (6*2) + (9*1) = 10 + 3 + 12 + 9 = 34$$

$$\text{Biaya3} = (2*2) + (1*4) + (6*5) + (9*7) = 4 + 4 + 30 + 63 = 101$$

$$\begin{aligned} \text{Total biaya produksi} &= \text{Biaya1} + \text{Biaya1} + \text{Biaya3} \\ &= 66 + 34 + 101 = 201 \end{aligned}$$

$$fitness = profit = (2*5) + (1*5) + (6*15) + (9*10) \\ = 10 + 5 + 90 + 90 = 195$$

Kromosom 3

5	7	8	10
---	---	---	----

Biaya untuk setiap biaya produksi:

$$Biaya1 = (5*5) + (7*2) + (8*3) + (10*4) = 25 + 14 + 24 + 40 = 103$$

$$Biaya2 = (5*5) + (7*3) + (8*2) + (10*1) = 25 + 21 + 16 + 10 = 72$$

$$Biaya3 = (5*2) + (7*4) + (8*5) + (10*7) = 10 + 28 + 40 + 70 = 148$$

$$\text{Total biaya produksi} = \text{Biaya1} + \text{Biaya2} + \text{Biaya3} \\ = 103 + 72 + 148 = 323$$

$$fitness = profit = (5*5) + (7*5) + (8*15) + (10*10) \\ = 25 + 35 + 120 + 100 = 280$$

3.9.3 Seleksi

Seleksi yang digunakan adalah seleksi *roulette wheel*. Pada seleksi ini, hanya individu yang memiliki *fitness* terbaik yang akan bertahan dan lolos seleksi untuk diproses lebih lanjut, sedangkan nilai *fitness* terburuk akan digantikan oleh nilai *fitness* yang lebih baik. Hasil dari perhitungan nilai *fitness* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Nilai *Fitness* kromosom

<i>Index</i>	<i>Fitness</i>
Kromosom 1	275
Kromosom 2	195
Kromosom 3	280

Setelah mendapat nilai *fitness*, maka hasilnya diurutkan dari nilai *fitness* terbaik sampai terburuk. Hasil pengurutan nilai *fitness* dapat dilihat pada Tabel 3.4.

Tabel 3.4 Nilai *Fitness* kromosom setelah terurut

<i>Index</i>	<i>Fitness</i>
Kromosom 3	280
Kromosom 1	275
Kromosom 2	195

3.9.4 Perkawinan Silang

Apabila proses seleksi telah dilakukan dan sudah terpilih induk baru, maka operator berikutnya adalah perkawinan silang. Dalam contoh ini diambil dua kromosom untuk dijadikan *parent* dan kemudian di *crossover*, sehingga menghasilkan 2 *child*.

Perkawinan silang yang digunakan adalah *uniform crossover* yaitu perkawinan silang dimana kromosom dari *child* akan diambil 50% dari *parent* 1 dan 50% dari *parent* 2 berdasarkan mask bit yang telah ditentukan.

Parent 1 :

10	7	8	10
----	---	---	----

Parent 2

4	5	8	9
---	---	---	---

mask bit

0	1	1	0
---	---	---	---

Setelah ditentukan *mask bit* maka langsung dilakukan penukaran gen antara *parent* 1 dan *parent* 2 untuk menghasilkan *child* 1 dan *child* 2.

Child 1

4	7	8	9
---	---	---	---

Child 2

10	5	8	10
----	---	---	----

Setelah proses penukaran gen tersebut, dilakukan pengecekan apakah individu baru yang terbentuk ilegal. Apabila terdapat kromosom yang ilegal maka dilakukan *repair*.

Setelah dilakukan *repair* pada kromosom maka dilakukan lagi proses penghitungan nilai *fitness* untuk masing-masing kromosom *child*.

Child 1

4	7	8	9
---	---	---	---

Biaya untuk setiap biaya produksi:

$$\text{Biaya1} = (4*5) + (7*2) + (8*3) + (9*4) = 20 + 14 + 24 + 36 = 94$$

$$\text{Biaya2} = (4*5) + (7*3) + (8*2) + (9*1) = 20 + 21 + 16 + 9 = 66$$

$$\text{Biaya3} = (4*2) + (7*4) + (8*5) + (9*7) = 10 + 28 + 40 + 63 = 141$$

$$\begin{aligned} \text{Total biaya produksi} &= \text{Biaya1} + \text{Biaya1} + \text{Biaya3} \\ &= 94 + 66 + 141 = 301 \end{aligned}$$

$$\begin{aligned} \text{fitness} = \text{profit} &= (4*5) + (7*5) + (8*15) + (9*10) \\ &= 20 + 35 + 120 + 90 = 265 \end{aligned}$$

Child 2

10	5	8	10
----	---	---	----

Biaya untuk setiap biaya produksi:

$$\text{Biaya1} = (10*5) + (5*2) + (8*3) + (10*4) = 50 + 10 + 24 + 40 = 124$$

$$\text{Biaya2} = (10*5) + (5*3) + (8*2) + (10*1) = 50 + 15 + 16 + 10 = 91$$

$$\text{Biaya3} = (10*2) + (5*4) + (8*5) + (10*7) = 20 + 20 + 40 + 70 = 150$$

$$\begin{aligned} \text{Total biaya produksi} &= \text{Biaya1} + \text{Biaya1} + \text{Biaya3} \\ &= 124 + 91 + 150 = 365 \end{aligned}$$

Karena jumlah biaya1 dan total biaya dari *Child 2* lebih besar dari alokasi maksimal biaya1 dan modal yang tersedia, maka perlu dilakukan *repair*. Misal *gen* yang akan diganti pada *child 2* adalah *gen* pada posisi 1.

Child 2

10	5	8	10
----	---	---	----

↓

5	5	8	10
---	---	---	----

Biaya untuk setiap biaya produksi:

$$\text{Biaya1} = (5*5) + (5*2) + (8*3) + (10*4) = 25 + 10 + 24 + 40 = 99$$

$$\text{Biaya2} = (5*5) + (5*3) + (8*2) + (10*1) = 25 + 15 + 16 + 10 = 66$$

$$\text{Biaya3} = (5*2) + (5*4) + (8*5) + (10*7) = 10 + 20 + 40 + 63 = 133$$

$$\begin{aligned} \text{Total biaya produksi} &= \text{Biaya1} + \text{Biaya1} + \text{Biaya3} \\ &= 99 + 66 + 133 = 298 \end{aligned}$$

$$\begin{aligned} \text{fitness} = \text{profit} &= (5*5) + (5*5) + (8*15) + (10*10) \\ &= 25 + 25 + 120 + 100 = 270 \end{aligned}$$

Setelah dilakukan perhitungan pada *child 1* dan *child 2* maka dilakukan penghitungan *fitness* masing-masing kromosom sehingga didapatkan data seperti pada Tabel 3.5.

Tabel 3.5 Nilai *Fitness* kromosom *child*

<i>Index</i>	<i>Fitness</i>
<i>Child 1</i>	265
<i>Child 2</i>	270

3.9.5 Mutasi

Diasumsikan *child 1* mengalami mutasi maka proses mutasi terjadi seperti ilustrasi di bawah ini. Misal *gen* yang akan diganti pada *child 1* adalah *gen* pada posisi 3.



$$\begin{aligned}
 \text{fitness} = \text{profit} &= (4*5) + (7*5) + (10*15) + (9*10) \\
 &= 20 + 35 + 150 + 90 = 295
 \end{aligned}$$

Setelah dilakukan mutasi pada *child 1* didapat hasil nilai *fitness* yang lebih baik. Sebelum dilakukan mutasi nilai *fitness* dari *child 1* adalah 265. Setelah dilakukan mutasi, nilai *fitness* dari *child 1* menjadi 295.

3.9.6 Pembentukan Generasi Baru

Setelah proses seleksi, perkawinan silang, dan mutasi terhadap kromosom dilakukan, proses pembentukan generasi baru dilakukan dengan cara membandingkan *fitness* kromosom anak dengan *fitness* kromosom induk. Kromosom anak akan menggantikan kromosom induk jika kromosom anak mempunyai nilai *fitness* yang lebih baik. Hasil nilai *fitness* kromosom *parents*, *child* dan mutasi dapat dilihat pada Tabel 3.6.

Tabel 3.6 Hasil Nilai Fitness Kromosom *Parent* dan *Child*

<i>Index</i>	<i>Fitness</i>
Kromosom 1	275
Kromosom 2	195
<i>Child 1</i>	295
<i>Child 2</i>	270

Selanjutnya 2 kromosom dengan nilai *fitness* terbaik diambil untuk dijadikan *parent* pada generasi selanjutnya. Dari kasus ini kromosom yang diambil sebagai *parent* adalah *child 1* karena nilai *fitness* yang dihasilkan lebih baik.

3.10 Data Uji

Dalam tugas akhir ini akan dibuat sebuah aplikasi untuk mengetahui efektifitas dari nilai parameter algoritma genetika (*crossover rate* dan *mutation rate*) yang digunakan dalam memecahkan masalah optimasi biaya produksi. Sebagai contoh, ditentukan matrik biaya produksi seperti ditunjukkan pada tabel 3.7.

Tabel 3.7 Contoh Matrik Biaya Produksi

Biaya Produksi	Produk										Alokasi Maks
	A	B	C	D	E	F	G	H	I	J	
1	5	2	3	4	2	1	2	4	2	4	200
2	5	3	2	5	6	0	3	7	1	1	100
3	2	1	2	4	2	2	3	2	5	0	350
4	6	0	3	7	1	3	2	4	2	4	250
5	2	4	5	4	2	2	3	7	1	1	100
Profit	10	5	15	20	25	10	15	25	30	15	

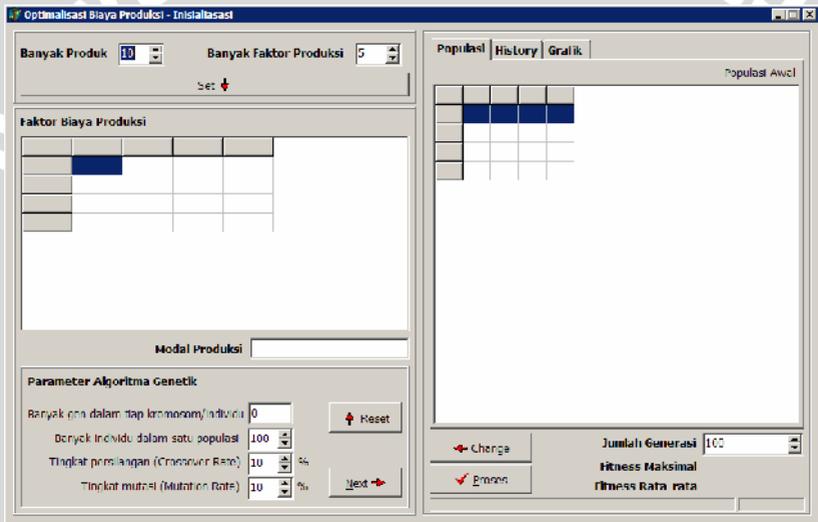
Dengan jumlah modal produksi sebesar 1000 (berasal dari penjumlahan alokasi maksimal tiap produk yaitu: $200 + 100 + 350 + 250 + 100$), akan dihitung nilai *fitness* yang dihasilkan dengan menggunakan nilai *crossover rate* dan nilai *mutation rate* yang berbeda. Nilai *crossover rate* dan *mutation rate*, yang akan diuji adalah: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dan 1.0.

Nilai *fitness* digunakan untuk mengetahui tingkat efektifitas dari parameter algoritma genetika yang digunakan. Semakin tinggi nilai *fitness* yang dihasilkan, maka semakin baik tingkat efektifitas dari nilai parameter yang diuji.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Program

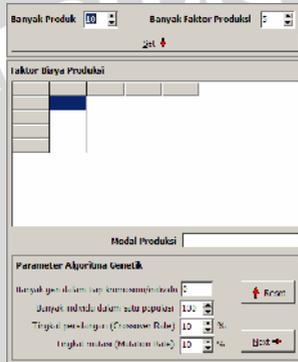
Tampilan utama dari aplikasi optimasi modal produksi menggunakan algoritma genetika dapat dilihat pada Gambar 4.1. Dalam tampilan aplikasi ini terdiri dari dua panel utama yaitu panel inisialisasi dan panel optimasi. Dalam panel inisialisasi terdapat dua panel yaitu biaya produksi dan algoritma genetika. Dalam panel optimasi terdapat tiga *tab* yaitu Populasi, *History*, Grafik..



Gambar 4.1 Tampilan utama aplikasi

4.1.1 Inisialisasi

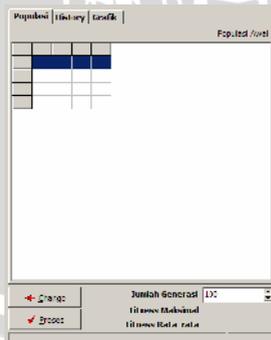
Inisialisasi merupakan proses penentuan nilai awal dari parameter yang akan dijadikan acuan, yang meliputi banyak barang yang diproduksi, banyak faktor produksi, matrik biaya produksi, modal produksi serta parameter algoritma genetika (banyak individu dalam populasi, jumlah generasi, *crossover rate*, *mutation rate*). Tampilan dari *panel* inisialisasi dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Panel* inisialisasi

4.1.2 Optimasi

Optimasi merupakan proses algoritma genetika pada aplikasi optimasi modal produksi. Proses yang ada adalah inisialisasi kromosom, hasil inisialisasi, optimasi dan hasil optimasi. Proses ini dapat berjalan dengan mengacu pada nilai-nilai parameter yang telah diinisialisasi. Tampilan dari *panel* optimasi dapat dilihat pada Gambar 4.3.



Gambar 4.3 *Panel* optimasi

4.2 Deskripsi Program

4.2.1 Inisialisasi Kromosom

Tahap awal dalam algoritma genetika adalah pembentukan individu yang dapat disebut dengan inisialisasi kromosom. Prosedur untuk inisialisasi kromosom diawali dengan membuat kromosom sejumlah individu. Setiap kromosom yang *valid* akan dimasukkan ke dalam populasi. Prosedur membuat kromosom dapat dilihat pada *Pseudocode* 4.4.

```
randomize;
i:=0; finish:=false;
repeat
  for j:=0 to nG-1 do
    Pop[i].Chrom[j]:=random(51);
    if (valid(Pop[i])) then
      begin
        calcFitness(pop[i]);
        if (i<(nIndv-1)) then inc(i)
          else finish:=true;
      end;
until (finish);
```

Pseudocode 4.1 Prosedur inisialisasi kromosom

4.2.2 Seleksi

Seleksi dimulai dengan menghitung nilai semua *fitness* kromosom. Kemudian memilih kromosom *parent1* dan *parent2* pada populasi secara *random* berdasarkan persentase dari *fitness* yang dimiliki. Prosedur pilih *parent* dapat dilihat pada *Pseudocode* 4.2.

```
randomize;
finish:=false; i:=1;
repeat
  rdVal:=random; j:=-1;
  repeat inc(j);
  until (rdVal<=Pop[j].Range);
  if (i=1) then
    begin P1:=j; inc(i); end
  else
    begin
      P2:=j; if P1<>P2 then finish:=true;
    end;
until finish;
```

Pseudocode 4.2 Prosedur pilih *parent*

4.2.3 Perkawinan Silang

Proses perkawinan silang dimulai dengan menentukan mask yang akan digunakan dalam proses perkawinan silang. Prosedur penentuan mask dapat dilihat pada *Pseudocode* 4.3.

```
for i:=0 to nG-1 do
  ary[i]:=0;
i:=0; n:=0;
randomize;
repeat
  if ary[i]=0 then ary[i]:=random(2);
  if (ary[i]=1) then inc(n);
  if (i<nG) then inc(i)
  else if (i=nG) then i:=0;
until (n/nG)>=0.5;
```

Pseudocode 4.3 Prosedur penentuan mask

Setelah menentukan mask, maka dilakukan proses perkawinan silang. Pertukaran nilai gen pada kromosom induk dilakukan berdasarkan mask yang telah ditentukan. Dari hasil perkawinan silang akan menghasilkan dua *child*. Prosedur perkawinan silang dapat dilihat pada *Pseudocode* 4.4.

```
for i:=0 to nG-1 do
begin
  if (ary[i]=1) then
  begin
    C1.Chrom[i]:=P2.Chrom[i];
    C2.Chrom[i]:=P1.Chrom[i];
  end
  else
  begin
    C1.Chrom[i]:=P1.Chrom[i];
    C2.Chrom[i]:=P2.Chrom[i];
  end;
end;

if (valid(C1)) then calcFitness(C1)
else repair(C1,s1);
if (valid(C2)) then calcFitness(C2)
else repair(C2,s2);
```

Pseudocode 4.4 Prosedur perkawinan silang

4.2.4 Mutasi

Mutasi dimulai dengan membangkitkan angka *random* untuk menentukan kromosom dan posisi gen yang akan terkena mutasi. Nilai gen yang baru didapatkan dengan cara membangkitkan angka secara *random*. Prosedur mutasi dapat dilihat pada *Pseudocode* 4.5.

```
n:=0; randomize;
check=[];
while (n<Round(Mr*nChild)) do
begin
  repeat
    id:=random(round(nChild*nG));
  until (not (id in check));
  rnd:=random(101);
  if (rnd<>Child[id div nG].Chrom[id mod nG]) then
  begin
    tmp:=Child[id div nG].Chrom[id mod nG];
    Child[id div nG].Chrom[id mod nG]:=rnd;
    if (valid(Child[id div nG])) then
    begin
      calcFitness(Child[id div nG]);
      Include(check,id); //add a value to 'set'
      inc(n);
    end else Child[id div nG].Chrom[id mod nG]:=tmp;
  end;
end;
```

Pseudocode 4.5 Prosedur mutasi

4.2.5 Perbaikan Kromosom

Perbaikan kromosom dilakukan untuk memperbaiki kromosom yang *invalid*. Perbaikan dimulai dengan membangkitkan angka *random* untuk menentukan posisi *gen* yang akan diperbaiki. Setelah itu, nilai gen yang didapatkan dengan cara membangkitkan angka secara *random*. Prosedur *repair* dapat dilihat pada *Pseudocode* 4.6.

```
randomize;
repeat
  repeat
    idx:=random(nG);
  until Idv.Chrom[idx]>2;
  idv.Chrom[idx]:=Random(Idv.Chrom[idx]-2)+1;
until (valid(Idv));
calcFitness(Idv);
```

Pseudocode 4.6 Prosedur *repair*

4.2.6 Proses Perhitungan Fitness

Proses perhitungan *fitness* merupakan penjumlahan dari perkalian antara jumlah barang yang diproduksi dengan keuntungan tiap barang yang diproduksi, yang mana merupakan *gen* dalam satu kromosom. Prosedur perhitungan *fitness* dapat dilihat pada *Pseudocode* 4.7.

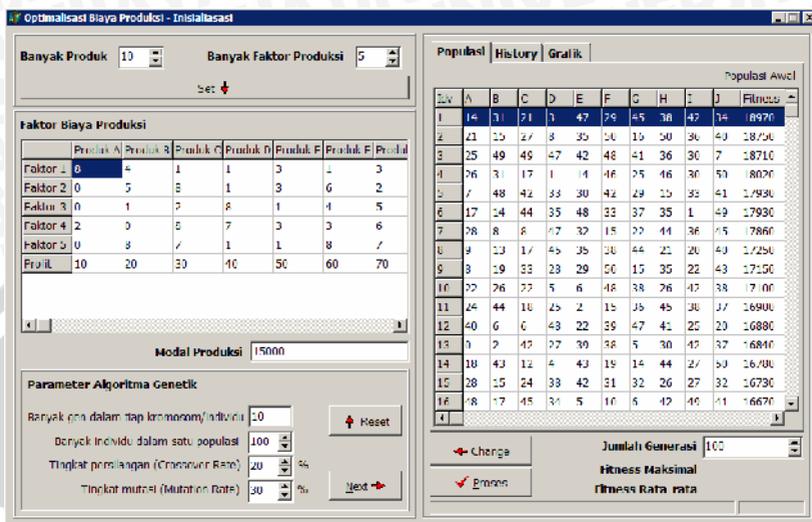
```
Idv.Fitness:=0;
for i:=0 to nG-1 do
begin
  if (i<3) then
  begin
    lim:=i*25;
    if (Idv.Chrom[i]>lim) then
      Idv.Fitness:=Idv.Fitness+
        (Idv.Chrom[i]*(CM[Length(CM)-1,i])+
          ((Idv.Chrom[i]-lim)*round((1.05)*(CM[Length(CM)-1,i])));
    end
  else Idv.Fitness:=Idv.Fitness+
    (Idv.Chrom[i]*(CM[Length(CM)-1,i]));
end;
```

Pseudocode 4.7 Prosedur perhitungan *fitness*

4.3 Penerapan Aplikasi

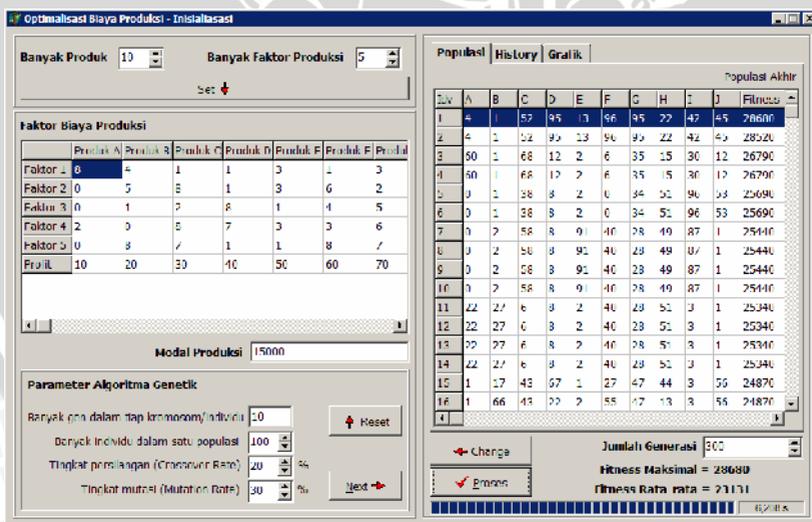
Aplikasi diterapkan dengan memasukkan data sesuai dengan keinginan *user*. Dalam kasus ini, data yang ingin dicari adalah keuntungan maksimal untuk modal produksi sebesar 15000.

Dalam kasus ini parameter genetika yang digunakan adalah jumlah populasi 100, iterasi/generasi yang dilakukan sebanyak 300 kali, dengan seleksi *roulette wheel* serta *crossover rate* sebesar 0,2 dan *mutation rate* sebesar 0,3. Untuk lebih jelasnya seperti pada Gambar 4.4.



Gambar 4.4 Tampilan aplikasi setelah inisialisasi kromosom

Apabila proses inisialisasi kromosom telah selesai, maka proses selanjutnya adalah proses optimasi. Proses optimasi disini adalah proses genetika yang dimulai dengan *crossover*. Setelah melalui proses genetika maka akan diperoleh keuntungan total (*fitness*) maksimal, seperti pada Gambar 4.5.



Gambar 4.5 Tampilan aplikasi setelah proses optimasi

4.4 Analisa Hasil

Perubahan nilai *fitness* dari inisialisasi awal sampai menjadi *fitness* terbaik dipengaruhi oleh model kromosom, *crossover rate*, *mutation rate*, banyaknya iterasi atau generasi yang digunakan, serta besarnya ukuran individu dalam satu populasi.

Pada penelitian ini mencoba melakukan analisa terhadap nilai *fitness* yang dihasilkan dengan mengganti nilai parameter *crossover rate* dan *mutation rate*. Uji coba dilakukan untuk mengetahui adanya perubahan nilai *fitness*. Nilai *crossover rate* dan nilai *mutation rate* yang diujikan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dan 1.0. Uji coba dilakukan masing-masing 5 kali percobaan untuk setiap nilai probabilitas.

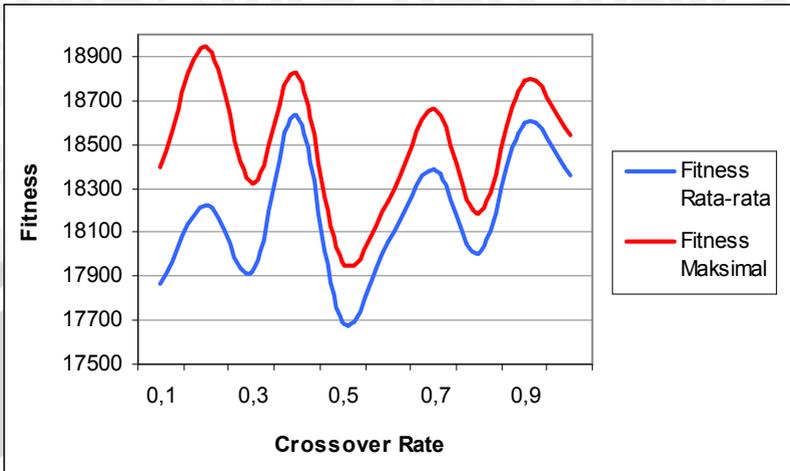
Rata-rata hasil dari uji coba perubahan nilai *crossover rate* (dengan nilai *mutation rate* sebesar 0.0) ditunjukkan pada Tabel 4.1.

Tabel 4.1 Hasil uji *crossover rate* berbeda

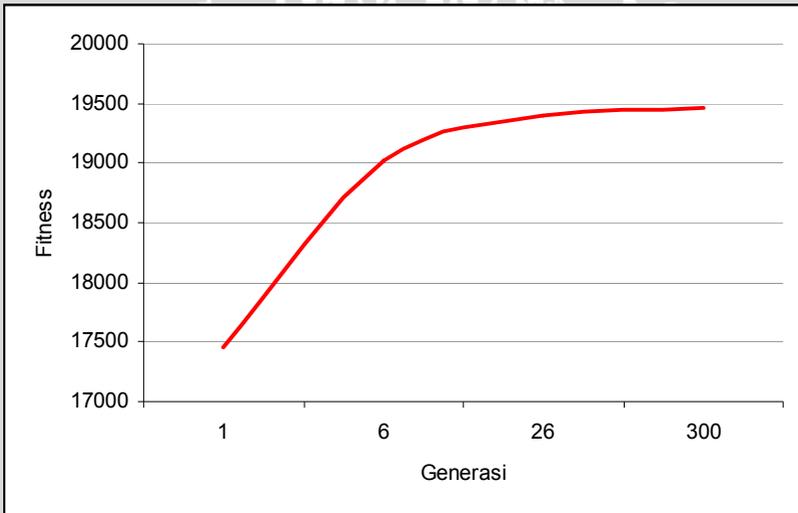
Crossover Rate	Fitness Rata-rata	Fitness Maksimal
0.1	17862	18394
0.2	18225	18942
0.3	17922	18324
0.4	18637	18825
0.5	17691	17969
0.6	18061	18244
0.7	18385	18657
0.8	18006	18188
0.9	18600	18788
1.0	18361	18546

Pada Tabel 4.3 dapat diketahui *fitness* maksimal dan *fitness* rata-rata pada setiap kali dilakukan percobaan. Dari hasil uji coba dapat dilihat bahwa nilai *fitness* maksimal tertinggi diperoleh dengan nilai *crossover rate* sebesar 0.2 dan nilai *fitness* rata-rata tertinggi diperoleh dengan nilai *crossover rate* sebesar 0.4. Sedangkan Nilai *fitness* rata-rata dan nilai *fitness* maksimal terendah diperoleh dengan nilai *crossover rate* sebesar 0.5.

Perbandingan nilai rata-rata *fitness* dapat dilihat pada Gambar 4.6. Sedangkan perbandingan nilai *fitness* maksimal tiap generasi dengan *crossover rate* 0.2 dapat dilihat pada Gambar 4.7.



Gambar 4.6 Grafik perbandingan rata-rata *fitness* dengan *crossover rate* yang berbeda



Gambar 4.7 Grafik perbandingan *fitness* maksimal dengan *crossover rate* sebesar 0.2

Dengan menggunakan *crossover rate* sebesar 0.2, akan dilakukan uji coba dengan menggunakan nilai *mutation rate* 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dan 1.0. Uji coba dilakukan masing-masing 5 kali percobaan untuk setiap nilai probabilitas.

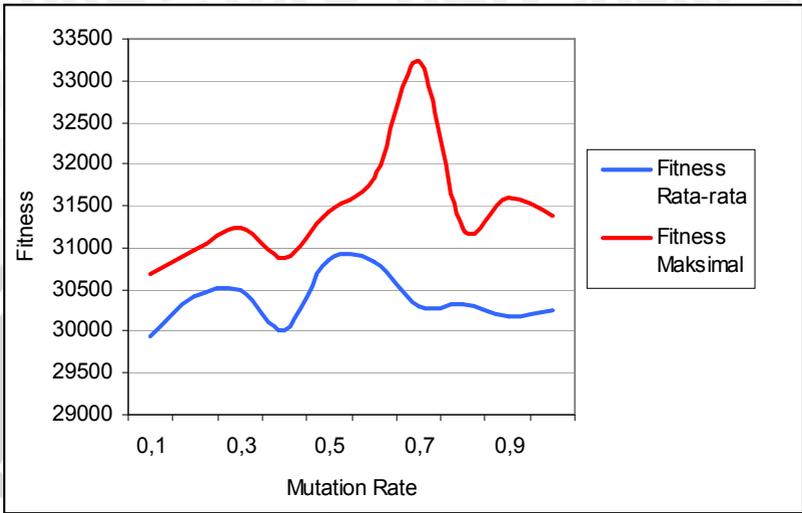
Rata-rata hasil dari uji coba perubahan nilai *mutation rate* (dengan nilai *crossover rate* sebesar 0.2) ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil uji *mutation rate* berbeda

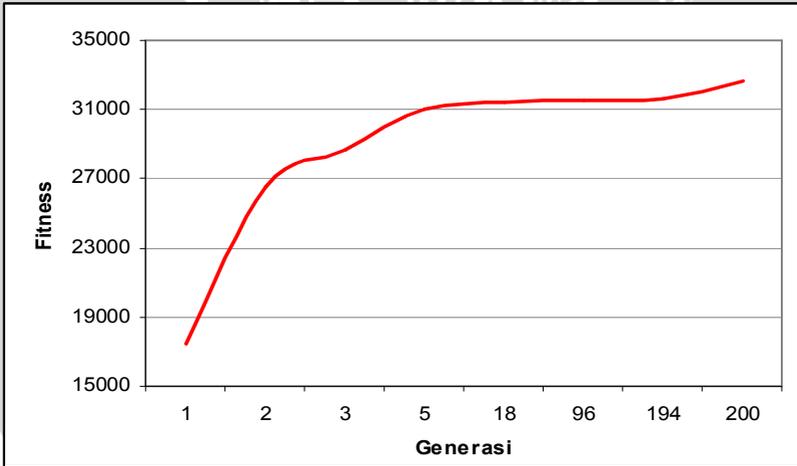
Mutation Rate	Fitness Rata-rata	Fitness Maksimal
0,1	29949	30686
0,2	30422	30982
0,3	30498	31239
0,4	30001	30881
0,5	30855	31422
0,6	30824	31839
0,7	30292	33243
0,8	30320	31246
0,9	30180	31599
1,0	30251	31385

Pada Tabel 4.4 dapat diketahui *fitness* maksimal dan *fitness* rata-rata pada setiap kali dilakukan percobaan. Dari hasil uji coba dapat dilihat bahwa nilai *fitness* maksimal tertinggi diperoleh dengan nilai *mutation rate* sebesar 0.7 dan nilai *fitness* rata-rata tertinggi diperoleh dengan nilai *mutation rate* sebesar 0.5. Sedangkan nilai *fitness* rata-rata terendah dan nilai *fitness* maksimal terendah diperoleh dengan nilai *mutation rate* sebesar 0.1.

Perbandingan nilai rata-rata *fitness* dapat dilihat pada Gambar 4.8. Sedangkan perbandingan nilai *fitness* maksimal tiap generasi dengan *mutation rate* 0.7 dapat dilihat pada Gambar 4.9.



Gambar 4.8 Grafik perbandingan rata-rata *fitness* dengan *mutation rate* yang berbeda



Gambar 4.9 Grafik perbandingan *fitness* maksimal dengan *mutation rate* sebesar 0.7

Dengan menggunakan nilai *crossover rate* sebesar 0.2 dan nilai *mutation rate* sebesar 0.7 diperoleh nilai *fitness* sebesar 32690 dengan kromosom sebagai berikut:

99	22	17	26	6	77	4	16	27	14
----	----	----	----	---	----	---	----	----	----

Hasil diatas dapat diartikan bahwa untuk mendapatkan keuntungan maksimal (yaitu sebesar 32690), perusahaan harus memproduksi masing-masing produk dengan jumlah produksi sesuai dengan yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Jumlah produksi tiap produk

Produk	Jumlah produksi
A	99
B	22
C	17
D	26
E	6
F	77
G	4
H	16
I	27
J	14

Berdasarkan hasil percobaan diatas, dapat dinyatakan bahwa dengan menggunakan nilai *crossover rate* sebesar 0.2 dan *mutation rate* sebesar 0.7 akan didapatkan nilai *fitness* yang lebih baik. Hal ini dikarenakan dengan menggunakan nilai parameter tersebut ($P_c=0.2$ dan $P_m=0.7$) didapatkan nilai *fitness* maksimal terbaik dan relatif lebih stabil jika dibandingkan dengan kombinasi parameter yang lain (dapat dilihat pada Lampiran 2).

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari Tugas Akhir ini adalah:

1. Pada penelitian ini, telah dibuat model genetika untuk masalah optimasi biaya produksi dengan menggunakan inisialisasi kromosom secara random, metode seleksi yang digunakan adalah *roulette wheel* dengan metode *uniform crossover*.
2. Optimasi biaya produksi pada PT. UVW merupakan contoh *multiconstraint knapsack problem*, dimana terdapat beberapa biaya produksi yang dibatasi oleh beberapa batasan (modal dan alokasi maksimal). Optimasi pada *multiconstraint knapsack problem* dapat dilakukan dengan menggunakan algoritma genetika. Algoritma genetika memerlukan adanya parameter untuk dapat bekerja dan menghasilkan nilai *fitness*. Parameter yang digunakan, diantaranya adalah *crossover rate* dan *mutation rate*. Penggunaan *crossover rate* dan *mutation rate* yang berbeda, menimbulkan perubahan terhadap nilai *fitness* yang dihasilkan. Dari hasil percobaan yang telah dilakukan, algoritma genetika dapat menghasilkan nilai *fitness* (keuntungan) terbaik dengan menggunakan nilai *crossover rate* sebesar 0.2 dan *mutation rate* sebesar 0.7.

5.2 Saran

Aplikasi yang dibangun masih belum sempurna. Hal yang dapat bermanfaat untuk mengembangkan aplikasi ini adalah:

1. Aplikasi optimasi biaya produksi ini dapat dikembangkan sehingga dapat menentukan alokasi biaya optimal bukan hanya pada biaya produksi langsung tetapi juga pada biaya produksi tidak langsung.
2. Pencarian alokasi biaya optimal masih bersifat linier, dapat dikembangkan sehingga dapat menentukan formula biaya produksi optimal yang bersifat non linear.

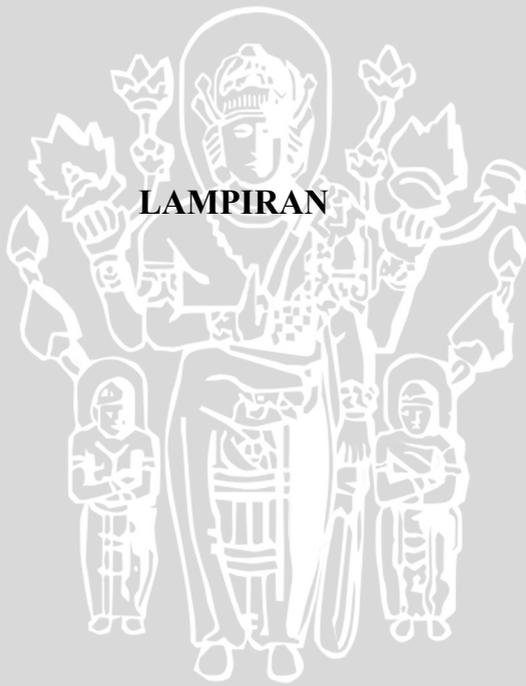
UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Gen, Mitsuo dan Runwei Cheng. 1997. *Genetic Algorithms and Engineering Design*. John Wiley and Sons. New York.
- Kusumadewi, Sri dan Hari Purnomo. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Graha Ilmu. Yogyakarta.
- Michalewicz, Zbigniew. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.
- Mulyadi. *Akuntansi Biaya*. 1999. Aditya Media. Yogyakarta.
- Obitko, M. 1998. *Introduction to Genetic Algorithm*, <http://cs.felk.cvut.cz/~xobitko/ga/>. Tanggal akses: 20 September 2008.
- Riwayadi. *Akuntansi Biaya*. 2006. Andalas University Press. Padang.
- Suyanto. 2005. *Algoritma Genetika dalam Matlab*. Penerbit Andi. Yogyakarta.
- Yento, Nico Saputro. *Pemakaian Algoritma Genetik untuk Penjadwalan Job Shop Dinamis Non Deterministik*. Jurusan Ilmu Komputer – FMIPA, Universitas Katolik Parahyangan. Tanggal akses: 15 September 2008.

UNIVERSITAS BRAWIJAYA



LAMPIRAN

UNIVERSITAS BRAWIJAYA



LAMPIRAN 1 HASIL PERCOBAAN DENGAN CROSSOVER RATE BERBEDA

Pc=0.1

Perc. Ke	AvgFit	MaxFit
1	17262	18740
2	17627	18090
3	18157	18340
4	17980	18330
5	18285	18470
Rata-rata	17862	18394

Pc=0.2

Perc. Ke	AvgFit	MaxFit
1	18617	19500
2	18347	18870
3	17296	18540
4	19523	19720
5	17342	18080
Rata-rata	18225	18942

Pc=0.3

Perc. Ke	AvgFit	MaxFit
1	17434	17610
2	18414	18600
3	17637	18370
4	18394	18580
5	17731	18460
Rata-rata	17922	18324

Pc=0.4

Perc. Ke	AvgFit	MaxFit
1	18655	18843
2	17474	17650
3	20236	20440
4	18127	18310
5	18691	18880
Rata-rata	18637	18825

Pc=0.5

Perc. Ke	AvgFit	MaxFit
1	18077	18260
2	17434	18106
3	17464	17640
4	17493	17670
5	17988	18170
Rata-rata	17691	17969

Pc=0.6

Perc. Ke	AvgFit	MaxFit
1	18869	19060
2	18592	18780
3	17285	17460
4	17285	17460
5	18275	18460
Rata-rata	18061	18244

Pc=0.7

Perc. Ke	AvgFit	MaxFit
1	17682	17681
2	18317	18502
3	17462	18070
4	19636	19834
5	18830	19020
Rata-rata	18385	18657

Pc=0.8

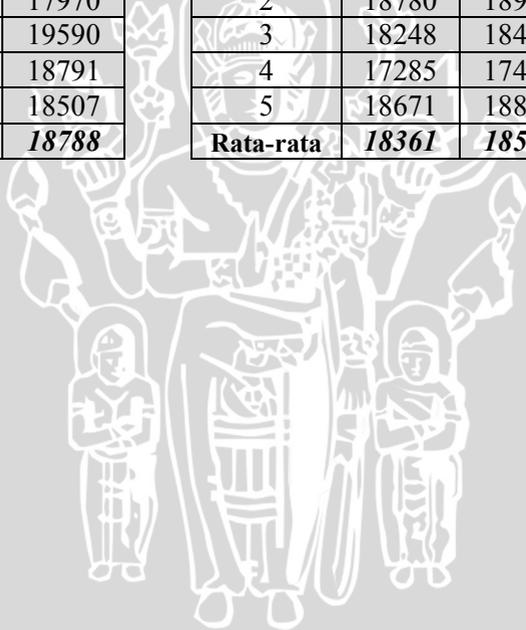
Perc. Ke	AvgFit	MaxFit
1	18582	18770
2	17355	17530
3	18493	18680
4	18305	18490
5	17295	17470
Rata-rata	18006	18188

Pc=0.9

Perc. Ke	AvgFit	MaxFit
1	18889	19080
2	17790	17970
3	19394	19590
4	18603	18791
5	18322	18507
Rata-rata	18600	18788

Pc=1.0

Perc. Ke	AvgFit	MaxFit
1	18820	19010
2	18780	18970
3	18248	18432
4	17285	17460
5	18671	18860
Rata-rata	18361	18546



LAMPIRAN 2 HASIL PERCOBAAN DENGAN MUTATION RATE BERBEDA

Pc=0.2 Pm=0.1

Perc. Ke	AvgFit	MaxFit
1	32591	32920
2	29052	29520
3	30511	31630
4	27410	28140
5	30183	31220
Rata-rata	29949	30686

Pc=0.2 Pm=0.2

Perc. Ke	AvgFit	MaxFit
1	31175	31820
2	30968	31281
3	30568	31460
4	29890	30384
5	29508	29967
Rata-rata	30422	30982

Pc=0.2 Pm=0.3

Perc. Ke	AvgFit	MaxFit
1	30494	31310
2	30325	31250
3	31224	31790
4	30560	31010
5	29885	30836
Rata-rata	30498	31239

Pc=0.2 Pm=0.4

Perc. Ke	AvgFit	MaxFit
1	28706	29300
2	30017	31660
3	30302	31094
4	31028	31470
5	29952	30880
Rata-rata	30001	30881

Pc=0.2 Pm=0.5

Perc. Ke	AvgFit	MaxFit
1	32523	33350
2	30441	31040
3	30529	30880
4	30241	30950
5	30541	30890
Rata-rata	30855	31422

Pc=0.2 Pm=0.6

Perc. Ke	AvgFit	MaxFit
1	30484	31540
2	30428	31483
3	31619	32670
4	31272	31790
5	30319	31710
Rata-rata	30824	31839

UNIVERSITAS BRAWIJAYA



Pc=0.2 Pm=0.7

Perc. Ke	AvgFit	MaxFit
1	30114	33286
2	30061	33253
3	30776	33324
4	30623	33230
5	29888	33121
Rata-rata	30292	33243

Pc=0.2 Pm=0.8

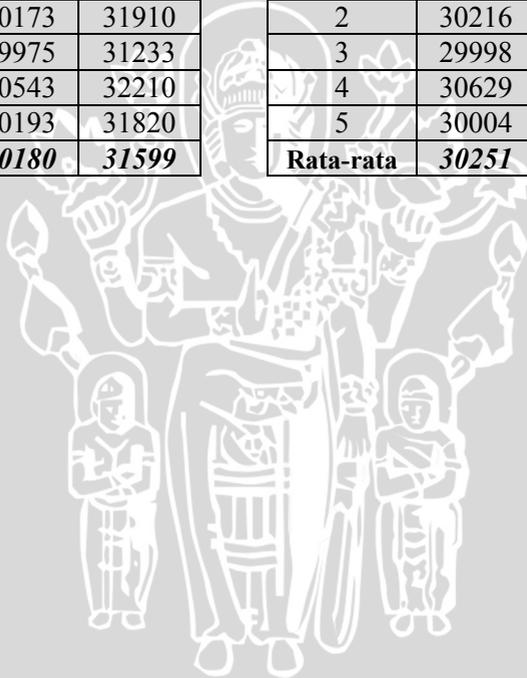
Perc. Ke	AvgFit	MaxFit
1	29857	30550
2	30344	32131
3	30189	30900
4	30586	31500
5	30626	31150
Rata-rata	30320	31246

Pc=0.2 Pm=0.9

Perc. Ke	AvgFit	MaxFit
1	30018	30821
2	30173	31910
3	29975	31233
4	30543	32210
5	30193	31820
Rata-rata	30180	31599

Pc=0.2 Pm=1.0

Perc. Ke	AvgFit	MaxFit
1	30409	31520
2	30216	31610
3	29998	30826
4	30629	31520
5	30004	31447
Rata-rata	30251	31385



LAMPIRAN 3 CONTOH PERHITUNGAN PERMASALAHAN OPTIMASI BIAYA PRODUKSI

Diketahui Matrik biaya produksi sebagai berikut:

Biaya	Produk			Alokasi Maks
	A	B	C	
1	5	4	2	600
2	1	4	6	800
3	2	2	4	500
Profit	20	25	50	

Penyelesaian dengan Menggunakan Metode Simpleks

Dengan menggunakan metode Simpleks, dimodelkan menjadi:

$$5A + 4B + 2C \leq 600$$

$$1A + 4B + 6C \leq 800$$

$$2A + 2B + 4C \leq 500$$

$$z = 20A + 30B + 50C$$

Dengan tujuan untuk memaksimalkan z, dirubah menjadi:

Fungsi Tujuan : Maksimumkan Z - 20A - 30B - 50C

$$\text{Batasan-batasan: } 5A + 4B + 2C + S1 = 600$$

$$1A + 4B + 6C + S2 = 800$$

$$2A + 2B + 4C + S3 = 500$$

Tabel Simpleks dari persamaan diatas adalah sebagai berikut:

Basis	A	B	C	S1	S2	S3	z	b
S1	5	4	2	1	0	0	0	600
S2	1	4	6	0	1	0	0	800
S3	2	2	4	0	0	1	0	500
z	-20	-25	-50	0	0	0	1	0

Adapun tabel penyelesaian dari permasalahan di atas adalah:

Basis	A	B	C	S1	S2	S3	z	b	rasio
S1	5	4	2	1	0	0	0	600	300
S2	1	4	6	0	1	0	0	800	800/6
S3	2	2	4	0	0	1	0	500	125
z	-20	-25	-50	0	0	0	1	0	
S1	4	3	0	1	0	-0,5	0	350	
S2	-2	1	0	0	1	-1,5	0	50	
C	0,5	0,5	1	0	0	0,25	0	125	
z	5	0	0	0	0	12,5	1	6250	

Penyelesaian dengan Menggunakan Algoritma Genetika

Dengan menggunakan algoritma genetika akan dilakukan percobaan menggunakan parameter:

$P_c = 0.2$ $p_m = 0.7$ Generasi = 300 Populasi = 100

Hasil dari percobaan tanpa faktor non-linear adalah sebagai berikut:

8	31	39
---	----	----

Dengan nilai *fitness* sebesar 6250.

Dengan menggunakan aturan non-linear,

- Apabila Total dari Biaya1 > 50 maka Biaya1 akan menjadi $0.5 \times \text{Biaya1}$
- Apabila jumlah Produk3 > 25 maka *profit* dari Produk3 menjadi $1.05 \times \text{profit Produk2}$

didapatkan hasil sebagai berikut:

39	90	40
----	----	----

Dengan nilai *fitness* sebesar 6370.

