

**OPTIMASI LABA PADA PRODUKSI JILBAB HOME
INDUSTRY MENGGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

Laboratorium Komputasi Cerdas dan Visualisasi



Disusun oleh

SAMAHER

NIM. 115060800111111

**PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2015

**OPTIMASI LABA PADA PRODUKSI JILBAB HOME
INDUSTRY MENGGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

Laboratorium Komputasi Cerdas dan Visualisasi

**Untuk memenuhi sebagian persyaratan
untuk meraih gelar Sarjana Komputer**



**Disusun oleh
SAMAHER
NIM. 115060800111111**

**PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG**

2015

LEMBAR PERSETUJUAN

**OPTIMASI LABA PADA PRODUKSI JILBAB HOME INDUSTRY
MENGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun oleh:
SAMAHER
NIM. 115060800111111

Skripsi ini telah disetujui oleh dosen pembimbing

Dosen Pembimbing I

Wayan F. Mahmudy, S.Si.,MT.,Ph.D
NIP. 19720919 199702 1 001

LEMBAR PENGESAHAN

**OPTIMASI LABA PADA PRODUKSI JILBAB HOME INDUSTRY
MENGUNAKAN ALGORITMA GENETIKA**

SKRIPSI

Disusun oleh :

SAMAHER

NIM. 115060800111111

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 23 Juni 2015

Dosen Penguji I

Dosen Penguji II

Dian Eka Ratnawati, S.Si., M.Kom Budi Darma Setiawan, S.Kom., M.Cs.
NIP.19730619 200212 2 001 NIK.841015 06 1 1 0090

Dosen Penguji III

Muhammad Tanzil Furqon, S.Kom., MCompSc
NIK.19820930 200801 1 004

Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Mardji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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

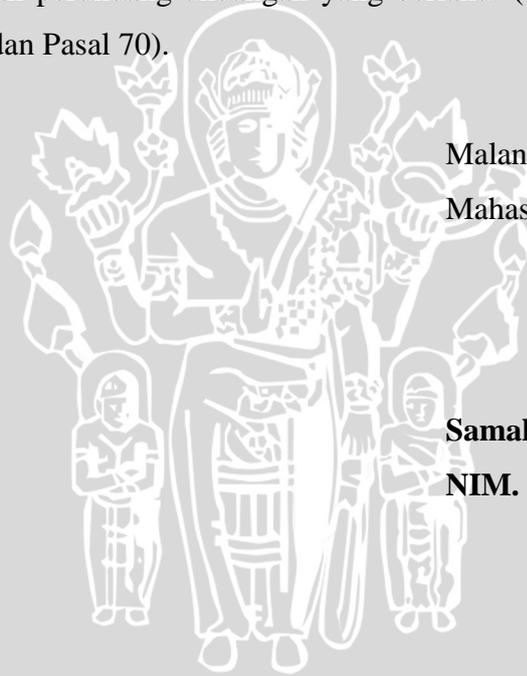
Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat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

Malang, 30 Juni 2015

Mahasiwa

Samaher

NIM. 115060800111111



ABSTRAK

Pada perusahaan yang bergerak pada bidang industri, optimasi laba menjadi tujuan utama dari tiap proses produksi. Untuk pencapaiannya sendiri, dibutuhkan perhitungan dan analisis yang tepat agar mendukung tercapainya laba maksimum. Proses perhitungan optimasi laba secara manual dirasa tidak mampu untuk memberi hasil yang akurat dalam waktu yang cepat. Oleh karena itu dibutuhkan algoritma yang tepat untuk memperoleh hasil optimasi laba maksimum yang akurat dan efisien. Pada penelitian ini akan membahas mengenai proses optimasi laba maksimum menggunakan algoritma genetika. Algoritma genetika memiliki kemampuan dalam menyelesaikan berbagai masalah kompleks dalam menghadapi masalah optimasi. Pada penelitian ini terdapat 10 kromosom sebagai kombinasi produksi jilbab dengan panjang sesuai banyaknya model jilbab. Dari data tersebut akan dibentuk sebuah populasi dengan ukuran yang bervariasi. Ukuran populasi optimum dari hasil uji coba dengan *reciprocal exchange mutation* adalah 180 populasi, dengan generasi optimal sebanyak 100 generasi. Nilai *crossover rate* dan *mutation rate* yang digunakan sebesar masing-masing 0,5 dengan hasil *fitness* pada proses uji coba sebesar 8037773. Sedangkan dengan *random mutation* menghasilkan ukuran populasi 200, kombinasi *cr* dan *mr* masing-masing 0,4 dan 0,6, dan banyak generasi 450 dengan hasil *fitness* 8164814. Hasil akhir dari penelitian adalah kromosom terbaik yang merupakan kombinasi produksi jilbab dengan laba maksimal dan pelanggaran kendala terendah.

Kata kunci: Algoritma genetika, optimasi laba, *home industry* jilbab

ABSTRACT

In a company that runs in industrial area, profit optimization becomes the main goal of each production processes. For the accomplishment itself, it needs accurate computation and analysis to support the achievement of maximum profit. The manual process of profit optimization counting doesn't seem to be able to give accurate result in short time. Therefore, the right algorithm is needed to get maximum profit optimization result that is accurate and efficient. In this study, the process of maximum profit optimization using genetic algorithm will be explained. Genetic algorithm has ability in solving many complex cases of optimization. In this study, there will be 10 chromosomes as the combination of hijab that should be produced with the length determined from the amount of hijab's types that are going to be produced. From those datas, a population will be made with various length. The optimum population size from the trials using reciprocal exchange mutation is 180 population, with 100 generation as the optimum amount of generation. The crossover and mutation rate values that will be used are both 0,5 with the fitness value 8037773. And the trials using random mutation resulting in population size 200, cr and mr value are each 0,4 and 0,6, generation amount 100 with fitness value 8164814. The result of this study is the best chromosome which is the combination of hijab that should be produced with maximum profit and lowest constraints violation.

Keywords: *Genetic algorithm, profit optimization, hijab home industry*

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
DAFTAR PERSAMAAN.....	xii
KATA PENGANTAR	xiii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Kajian Pustaka	5
2.2 Algoritma Genetika	6
2.3 <i>Home Industry</i>	9
2.3.1 Pengertian <i>Home Industry</i>	9
2.3.2 Pelaku <i>Home Industry</i>	10
2.3.3 Pusat Kegiatan <i>Home Industry</i>	10
BAB III METODOLOGI PENELITIAN.....	12
3.1 Tahapan Penelitian	12
3.2 Kebutuhan Sistem.....	13
3.2.1 Spesifikasi Perangkat Keras.....	13
3.2.2 Spesifikasi Perangkat Lunak.....	13



3.2.3	Batasan-Batasan Dalam Implementasi.....	13
3.3	Formulasi Masalah	14
3.4	Siklus Genetika.....	16
3.4.1	Representasi Kromosom	17
3.4.2	Inisialisasi Populasi.....	19
3.4.3	Reproduksi	19
3.4.4	Seleksi	22
BAB IV PERANCANGAN SISTEM.....		23
4.1	Perancangan <i>User Interface</i>	23
4.2	Perancangan Uji Coba dan Evaluasi.....	24
4.2.1	Uji Coba Ukuran Populasi	25
4.2.2	Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>MutationRate</i>	25
4.2.3	Uji Coba Jumlah Generasi	26
BAB V IMPLEMENTASI.....		27
5.1	Implementasi Sistem	27
5.1.2	Perhitungan Nilai <i>Fitness</i>	27
5.1.3	Proses <i>Crossover</i>	29
5.1.4	Proses Mutasi.....	30
5.1.5	Proses Seleksi	32
5.2	Implementasi Antarmuka Pengguna.....	33
5.2.1	Implementasi Halaman Utama.....	33
5.2.2	Implementasi Halaman Hasil Perhitungan	33
BAB VI PENGUJIAN DAN ANALISA		35
6.1	Hasil Pengujian Sistem.....	35
6.1.1	Hasil Pengujian Sistem dengan <i>Reciprocal Exchange Mutation</i>	35
6.1.2	Hasil Pengujian Sistem dengan <i>Random Mutation</i>	36
6.2	Hasil dan Analisis Pengujian Parameter Algoritma Genetika dengan <i>Reciprocal Exchange Mutation</i>	37
6.2.1	Hasil dan Analisis Uji Coba Ukuran Populasi.....	37
6.2.2	Hasil dan Analisis Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	38
6.2.3	Hasil dan Analisa Uji Coba Banyaknya Generasi	40



6.2.4 Hasil Terbaik.....	41
6.3 Hasil dan Analisis Pengujian Parameter Algoritma Genetika dengan <i>Random Mutation</i>	42
6.3.1 Hasil dan Analisis Uji Coba Ukuran Populasi.....	42
6.3.2 Hasil dan Analisis Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	43
6.3.3 Hasil dan Analisa Uji Coba Banyaknya Generasi	44
6.3.4 Hasil Terbaik.....	46
6.4 Perbandingan Hasil <i>Reciprocal Exchange Mutation</i> dan <i>Random Mutation</i>	46
BAB VII PENUTUP	48
7.1 Kesimpulan.....	48
7.2 Saran.....	49
DAFTAR PUSTAKA	50



DAFTAR GAMBAR

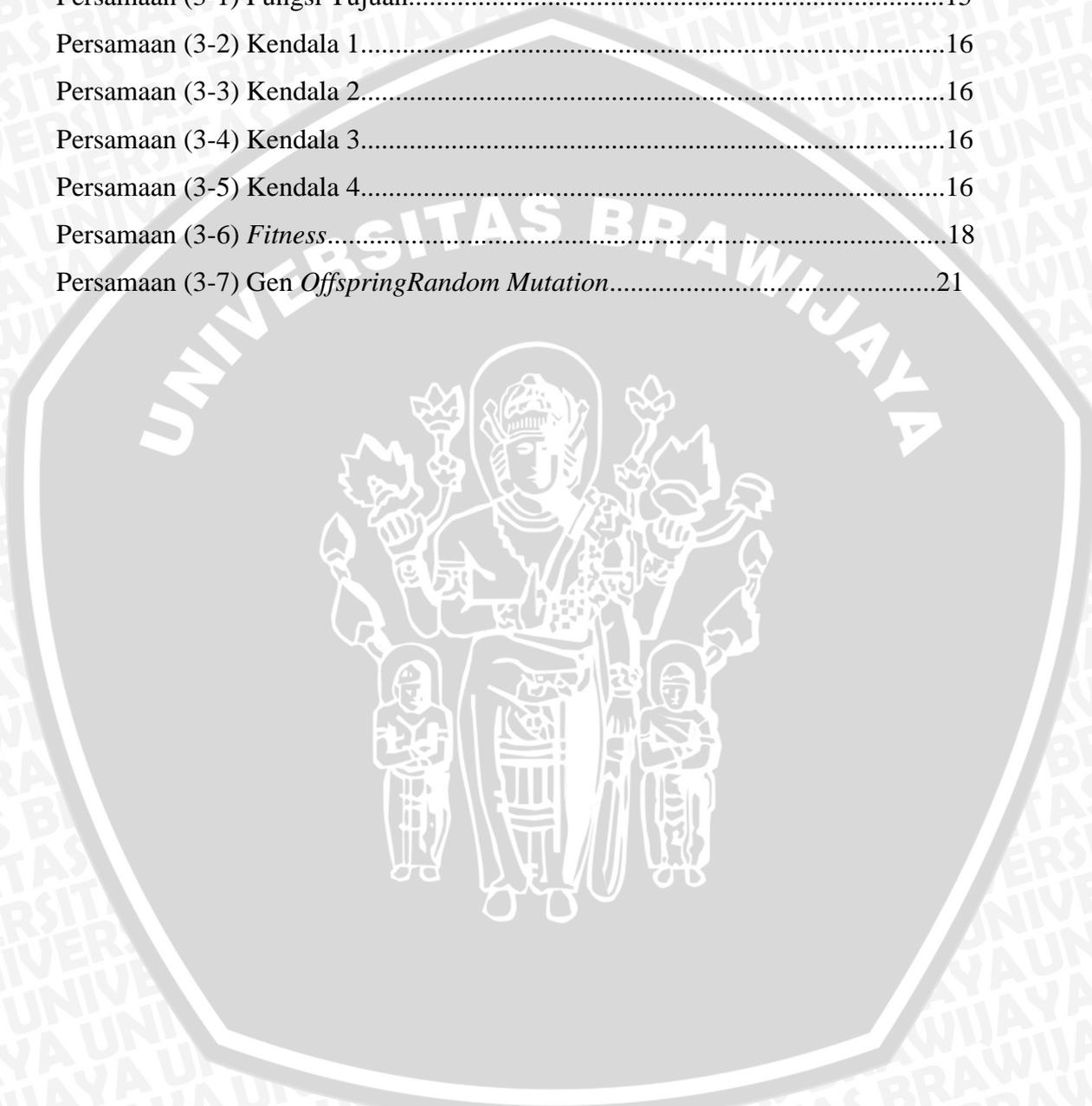
Gambar 2.1 Penyelesaian Masalah Dengan Algoritma Genetika.....	8
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	12
Gambar 3.2 Diagram Alir Algoritma Genetika.....	16
Gambar 3.3 Diagram Alir Proses <i>Crossover</i>	19
Gambar 3.4 Proses <i>Crossover</i>	20
Gambar 3.5 Diagram Alir Proses <i>Reciprocal Exchange Mutation</i>	20
Gambar 3.6 Proses Mutasi <i>Reciprocal Exchange</i>	20
Gambar 3.7 Diagram Alir <i>Random Mutation</i>	21
Gambar 3.8 Proses Mutasi <i>Random Mutation</i>	21
Gambar 3.9 Diagram Alir Proses Seleksi.....	22
Gambar 4.1 Rancangan Interface Sistem: Halaman Utama.....	23
Gambar 4.2 Rancangan Interface Sistem: Hasil Perhitungan.....	24
Gambar 5.1 Implementasi Halaman Utama.....	33
Gambar 5.2 Implementasi Halaman Hasil Perhitungan.....	34
Gambar 5.3 Penjelasan Hasil Akhir.....	34
Gambar 6.1 Grafik Hasil Uji Coba Ukuran Populasi.....	38
Gambar 6.2 Hasil Uji Coba Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i>	39
Gambar 6.3 Grafik Hasil Uji Coba Banyaknya Generasi.....	40
Gambar 6.4 Hasil Terbaik Pengujian <i>Reciprocal Exchange Mutation</i>	41
Gambar 6.5 Grafik Hasil Uji Coba Ukuran Populasi.....	42
Gambar 6.6 Hasil Uji Coba Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i>	44
Gambar 6.7 Grafik Hasil Uji Coba Banyaknya Generasi.....	45
Gambar 6.8 Hasil Terbaik Pengujian <i>Random Mutation</i>	46

DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka.....	6
Tabel 3.1 Spesifikasi Perangkat Keras Komputer.....	13
Tabel 3.2 Spesifikasi Perangkat Lunak Komputer.....	13
Tabel 3.3 Laba Penjualan Jilbab <i>Home Industry</i> dalam Rupiah.....	15
Tabel 3.4 Kebutuhan Bahan Baku dan Biaya Jahit dalam Produksi Jilbab.....	15
Tabel 3.5 Representasi Kromosom.....	17
Tabel 3.6 Contoh Perhitungan <i>Fitness</i>	18
Tabel 3.7 Populasi Inisial.....	19
Tabel 3.8 Individu Baru Hasil Seleksi.....	22
Tabel 4.1 Rancangan Uji Coba Ukuran Populasi.....	25
Tabel 4.2 Rancangan Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i>	26
Tabel 4.3 Rancangan Uji Coba Jumlah Generasi.....	27
Tabel 6.1 Hasil Pengujian Sistem dengan <i>Reciprocal Exchange Mutation</i>	35
Tabel 6.2 Hasil Pengujian Sistem dengan <i>Random Mutation</i>	36
Tabel 6.3 Hasil Uji Coba Populasi.....	37
Tabel 6.4 Hasil Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	39
Tabel 6.5 Hasil Percobaan Banyaknya Generasi.....	40
Tabel 6.6 Hasil Uji Coba Populasi.....	42
Tabel 6.7 Hasil Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	43
Tabel 6.8 Hasil Percobaan Banyaknya Generasi.....	44
Tabel 6.9 Perbandingan Hasil <i>Reciprocal Exchange Mutation</i> dan <i>Random Mutation</i>	46

DAFTAR PERSAMAAN

Persamaan (2-1) Margin Laba Kotor	11
Persamaan (2-2) Margin Laba Operasi Bersih.....	11
Persamaan (3-1) Fungsi Tujuan.....	15
Persamaan (3-2) Kendala 1.....	16
Persamaan (3-3) Kendala 2.....	16
Persamaan (3-4) Kendala 3.....	16
Persamaan (3-5) Kendala 4.....	16
Persamaan (3-6) <i>Fitness</i>	18
Persamaan (3-7) Gen <i>OffspringRandom Mutation</i>	21



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang atas rahmat-Nya maka penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Optimasi Laba Pada Produksi Jilbab Home Industry Menggunakan Algoritma Genetika”. Skripsi ini diajukan sebagai salah satu syarat untuk mendapat gelar sarjana S-1 di Program Teknologi Informatika dan Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala. Namun, berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala-kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terimakasih kepada:

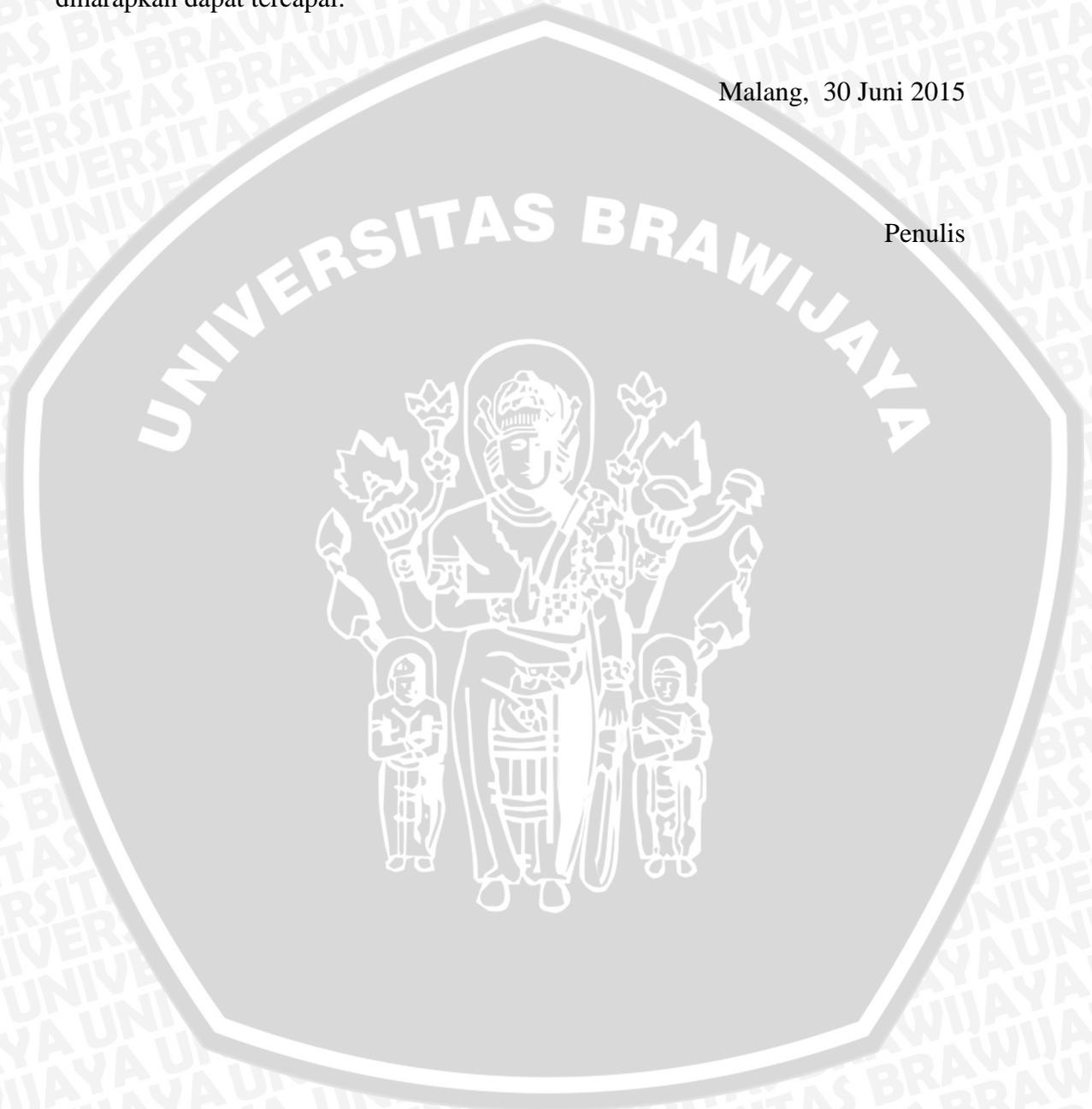
1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. sebagai dosen pembimbing yang dengan sabar memberikan saran dan masukan perbaikan skripsi ini;
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Edy Santoso, S.Si, M.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2, dan Wakil Ketua 3 Fakultas Ilmu Komputer Universitas Brawijaya;
3. Bapak Drs. Mardji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika/Ilmu Komputer serta segenap Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Fakultas Ilmu Komputer Universitas Brawijaya;
4. Kedua orang tua penulis, Ibu Hiba dan Bapak Abdullah Haydar, dan keluarga yang tidak pernah bosan dan lupa untuk memberikan doa dan dorongan semangat hingga terselesaikannya skripsi ini;
5. Ibu Saudah selaku pemilik home industry jilbab “Ayune” yang telah bersedia menjadi narasumber dalam penelitian ini;
6. Seluruh teman-teman mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya khususnya angkatan 2011;

7. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat penulisan sebutkan satu per satu.

Semoga materi ini dapat bermanfaat dan menjadi sumbangan pemikiran bagi pihak yang membutuhkan, khususnya bagi penulis sehingga tujuan yang diharapkan dapat tercapai.

Malang, 30 Juni 2015

Penulis



BAB I PENDAHULUAN

1.1. Latar Belakang

Pada perusahaan yang bergerak pada bidang industri, optimasi laba menjadi tujuan utama dari tiap proses produksi. Untuk pencapaiannya sendiri, dibutuhkan perhitungan dan analisis yang tepat agar mendukung tercapainya laba maksimum.

Subramanyam dan Wild dalam Fitriyani (2014) menyatakan bahwa laba (*income*) atau *earnings* atau *profit* adalah rangkuman hasil bersih kegiatan operasi usaha dalam kurun waktu tertentu yang dinyatakan dalam istilah finansial. Laba yang dihasilkan perusahaan merupakan salah satu ukuran kinerja perusahaan yang sering digunakan sebagai dasar pengambilan keputusan, yang dihitung dengan mengakui pendapatan dan mengaitkan biaya dengan pendapatan yang diakui.

Beberapa penelitian sebelumnya telah membahas mengenai permasalahan optimasi, antara lain penelitian oleh Fardiana (2012) yang membahas mengenai maksimalisasi keuntungan pada toko kue dengan metode simpleks dan penelitian lainnya adalah oleh Pratiwi (2014) yang membahas optimasi biaya pemenuhan kebutuhan gizi dengan algoritma genetika. Produksi jilbab membutuhkan sumber daya manusia dan beberapa bahan, misalnya kain, renda, benang, dan lain-lain. Dari beberapa bahan tersebut diproses dengan model dan kombinasi bahan yang berbeda menghasilkan beberapa jilbab dengan kisaran harga yang berbeda pula. Optimasi diperlukan untuk memaksimalkan laba dengan modal yang terbatas.

Proses perhitungan optimasi laba secara manual dirasa tidak mampu untuk memberi hasil yang akurat dalam waktu yang cepat. Oleh karena itu dibutuhkan algoritma yang tepat untuk memperoleh hasil optimasi laba maksimum yang akurat dan efisien. Pada penelitian ini akan membahas mengenai proses optimasi laba maksimum menggunakan algoritma genetika. Algoritma genetika memiliki kemampuan dalam menyelesaikan berbagai masalah rumit dalam menghadapi masalah optimasi (Mahmudy, 2013). Diharapkan algoritma genetika ini mampu menyelesaikan masalah optimasi laba pada proses produksi jilbab dan

memberikan kualitas solusi yang lebih tinggi dibandingkan penelitian sebelumnya.

1.2. Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana cara menghasilkan laba maksimum dalam produksi jilbab dengan algoritma genetika?
2. Bagaimana cara menentukan parameter algoritma genetika yang tepat untuk permasalahan optimasi laba?
3. Bagaimanamengukur kualitas solusi yang dihasilkan algoritma genetika?

1.3. Batasan Masalah

Batasan masalah yang akan dijadikan sebagai pedoman dalam pelaksanaan penelitian ini adalah sebagai berikut :

1. Data untuk penelitian ini didapatkan dari *home industry* jilbab “Ayune” yang berlokasi di Jl. Ki Ageng Gribig 324 Kedungkandang Malang.
2. Bahan baku dalam pembuatan jilbab yang digunakan dalam perhitungan antara lain kain (tanpa mempertimbangkan warna), benang, busa, dan biaya jahit.
3. Bahasa pemrograman yang digunakan dalam penelitian adalah bahasa Java.
4. Pengujian yang dilakukan adalah pengujian populasi, kombinasi *cr* dan *mr*, serta generasi.

1.4. Tujuan Penelitian

Mengoptimasi perolehan laba pada proses produksi jilbab sesuai kebutuhan dengan manajemen bahan menggunakan algoritma genetika.

1.5. Manfaat Penelitian

Memberikan informasi bagi perusahaan sebagai bahan pertimbangan dalam manajemen bahan jilbab pada proses produksi untuk menghasilkan laba maksimum.

1.6. Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Bab I berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Tinjauan pustaka terbagi menjadi dua subbab yaitu kajian pustaka dan dasar teori. Kajian pustaka membahas tentang perbandingan penelitian yang dilakukan sekarang dengan penelitian yang sudah dilakukan sebelumnya. Dasar teori membahas teori-teori yang mendukung dalam pengembangan dan perancangan sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika antara lain konsep dasar algoritma genetika, laba, dan *home industry*.

BAB III Metodologi Penelitian

Bab III membahas tentang metode yang digunakan dalam penulisan yaitu algoritma genetika.

BAB IV Analisis Kebutuhan dan Perancangan

Bab IV membahas tentang analisis kebutuhan dari sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika dan kemudian merancang hal-hal yang berhubungan dengan analisis tersebut.

BAB V Implementasi dan Pembahasan

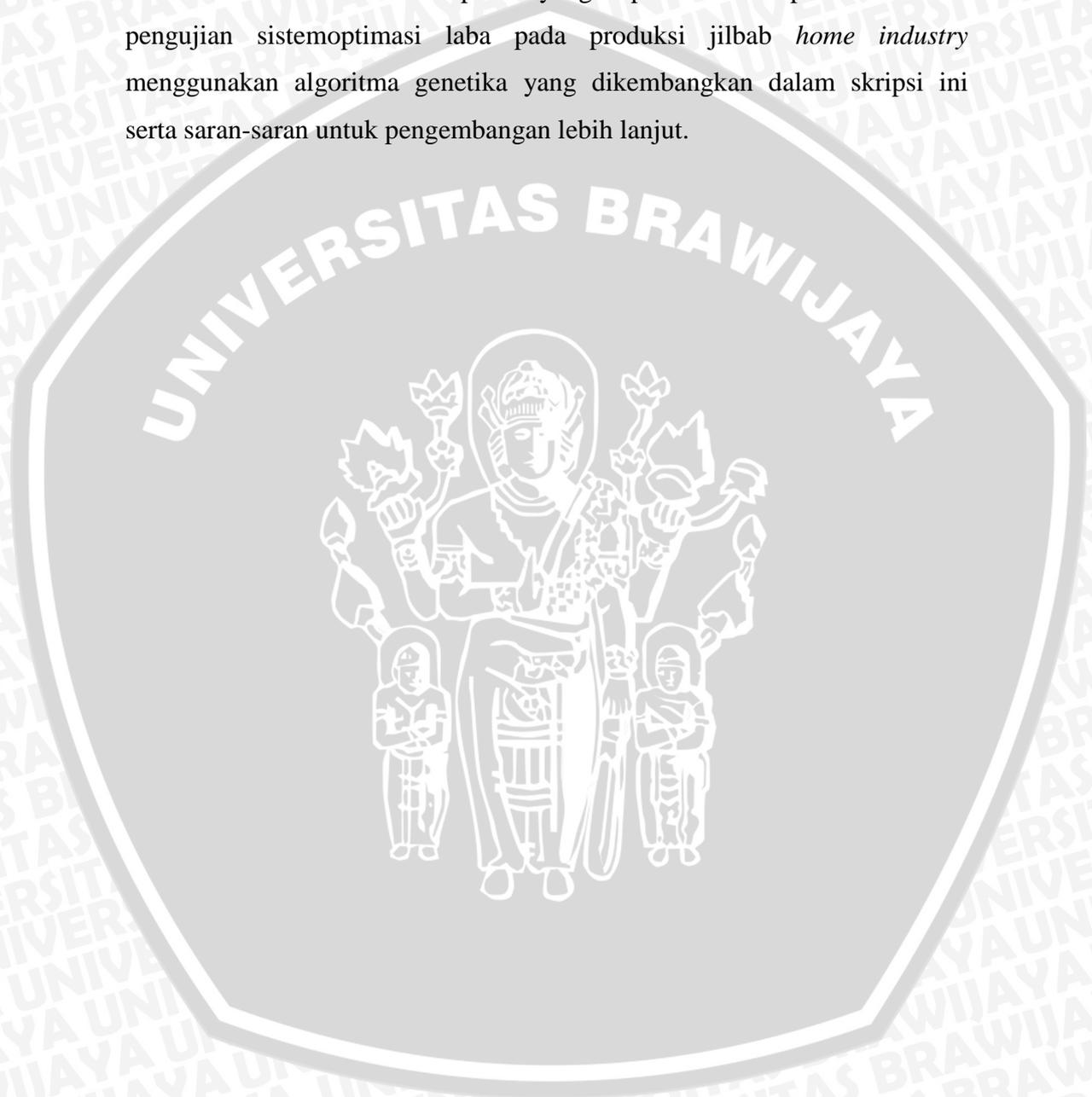
Bab V membahas tentang hasil perancangan dari analisis kebutuhan dan implementasi sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika.

BAB VI Pengujian

Bab VI memuat tentang hasil pengujian dan analisis terhadap sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetik yang telah direalisasikan.

Bab VII Penutup

Bab terakhir memuat kesimpulan yang diperoleh dari pembuatan dan pengujian sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



BAB II

TINJAUAN PUSTAKA

Bab dua terdiri dari kajian pustaka dan dasar teori. Kajian pustaka membahas perbandingan penelitian yang telah ada sebelumnya dan yang sedang dilakukan. Penelitian sebelumnya yang menjadi acuan penelitian ini adalah “Maksimalisasi Keuntungan Pada Toko Kue Martabak Doni dengan Metode Simpleks” oleh Fardiana (2012) dan “Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi” oleh Pratiwi (2014). Dasar teori membahas teori pendukung yang diperlukan untuk menyusun penelitian yang diusulkan. Pada penelitian ini, dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah antara lain adalah konsep dasar algoritma genetika, laba, dan *home industry*.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini adalah membandingkan penelitian ini dengan penelitian sebelumnya oleh Fardiana (2012) yang membahas mengenai maksimalisasi keuntungan pada toko kue dengan metode simpleks dan penelitian lainnya adalah oleh Pratiwi (2014) yang membahas optimasi biaya pemenuhan kebutuhan gizi dengan algoritma genetika.

Penelitian pertama menggunakan metode simpleks. Metode simpleks adalah salah satu teknik penyelesaian dalam program linier yang digunakan sebagai teknik pengambilan keputusan dalam permasalahan yang berhubungan dengan optimasi. Metode simpleks digunakan untuk mendapatkan nilai optimal dari program linier yang melibatkan banyak variabel (lebih dari dua variabel) dan banyak *constraint* atau pembatas (Wirdasari, 2009).

Penelitian kedua menggunakan metode yang sama dengan penelitian ini, yaitu menggunakan algoritma genetika. Algoritma genetika adalah tipe algoritma evolusi yang paling populer. Kelebihan algoritma genetika adalah algoritma genetika menghasilkan himpunan solusi optimal yang sangat berguna pada penyelesaian masalah dengan banyak obyektif (Mahmudy dan Rahman, 2011), sedangkan metode simpleks hanya menghasilkan solusi tunggal. Pengertian algoritma genetika secara lengkap akan dijelaskan pada dasar teori.

Rincian mengenai perbedaan objek dan metode pada masing-masing penelitian acuan yang digunakan dengan penelitian ini diuraikan pada Tabel 2.1 yang memuat tentang parameter input, proses, dan output.

Tabel 2.1 Tabel Kajian Pustaka

No.	Judul Penelitian	Objek	Metode yang digunakan	Hasil dan Pengujian
		Input dan Parameter		
1	Maksimalisasi Keuntungan Pada Toko Kue Martabak Doni Dengan Metode Simpleks (Fardiana, 2012)	<ul style="list-style-type: none"> - Data bahan baku produk - Laba masing-masing produk - Data persediaan bahan baku. 	Metode Simpleks	- Hasil berupa kombinasi produk yang optimum dan laba maksimum yang akan diperoleh.
2	Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi (Pratiwi, 2014)	<ul style="list-style-type: none"> - Data bahan makanan, kandungan gizi, dan harga makanan. 	Algoritma Genetika	- Hasil berupa kombinasi bahan makanan terbaik dengan kandungan gizi mencukupi dan memiliki biaya minimum.
3	Optimasi Laba Pada Produksi Jilbab Home Industry Menggunakan Algoritma Genetika	<ul style="list-style-type: none"> - Data kebutuhan bahan baku, laba masing-masing produk, dan persediaan bahan perminggu - Parameter algoritma genetika (ukuran populasi, generasi, <i>crossover rate</i>, dan <i>mutation rate</i>). 	Algoritma Genetika	- Hasil berupa kombinasi produksi jilbab yang menghasilkan laba paling optimum.

2.2 Algoritma Genetika

Algoritma genetika adalah tipe algoritma evolusi yang paling populer. Algoritma genetika berkembang seiring dengan pesatnya perkembangan teknologi informasi. Algoritma ini banyak digunakan dalam bidang fisika, biologi, ekonomi, sosiologi dan berbagai bidang lainnya yang sering menghadapi masalah optimasi yang model matematikanya kompleks atau bahkan sulit dibangun dikarenakan oleh kemampuannya untuk menyelesaikan berbagai masalah kompleks (Mahmudy, 2013).

Istilah yang digunakan dalam algoritma genetika banyak diadopsi dari ilmu genetika, karena algoritma genetika diilhami oleh ilmu tersebut. Algoritma genetika apabila dibandingkan dengan prosedur pencarian dan optimasi

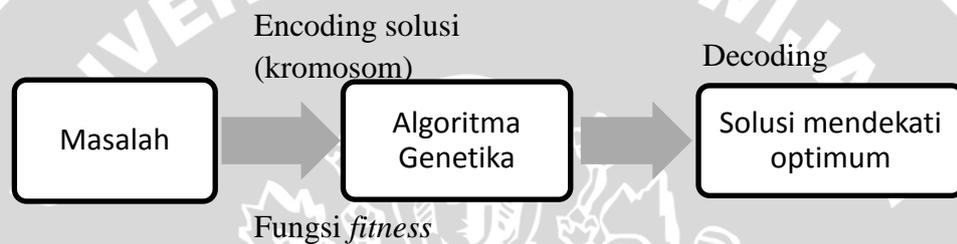
biasamemiliki perbedaan dalam beberapa hal, antara lainsebagai berikut (Michalewicz dalam Mahmudy, 2013):

- Manipulasi tidak secara langsung dilakukan terhadap parameternya sendiri, melainkan terhadap kode dari himpunan parameter (biasa disebut kromosom).
- Proses pencarian tidak hanya dilakukan dari satu titik, melainkan dilakukan dari beberapa titik dalam satu populasi.
- Proses pencarian menggunakan informasi dari fungsi tujuan.
- Pencariannya menggunakan operator stokhastikyang bersifat probabilistik, tidakmenggunakan aturan deterministik.

Kelebihan algoritma genetika sebagai metode optimasi antara lain sebagai berikut(Mahmudy, 2013):

- Algoritma genetika merupakan algoritma yang berbasis populasi yang memungkinkan digunakanpada optimasi masalah dengan ruang pencarian yang sangat luas dan kompleks.
- Individu yang ada pada populasi bisa diletakkan pada beberapa sub-populasi yang diproses pada sejumlah komputer secara paralel.
- Algoritma genetika menghasilkan himpunan solusi optimal yang sangat berguna pada penyelesaian masalah dengan banyak obyektif (Mahmudy dan Rahman, 2011).
- Algoritma genetika dapat digunakan untuk menyelesaikan masalah yang kompleks dengan banyak variabel.
- Algoritma genetika menggunakan kromosomuntuk mengkodekan solusi sehingga bisa melakukanpencarian tanpa memperhatikan informasi derivatif yang spesifik dari masalah yangdiselesaikan (Gen dan Cheng; Haupt dan Haupt dalam Mahmudy, 2013).
- Algoritma genetika bisa diimplementasikan pada berbagai macam data seperti data yangdibangkitkan secara numerik atau menggunakan fungsi analitis (Haupt dan Hauptdalam Mahmudy, 2013).
- Algoritma genetika cukup fleksibel untuk dihibridisasikan dengan algoritma lainnya (Gen dan Chengdalam Mahmudy, 2013).
- Algoritma genetikabersifat *ergodic*, berbagai solusi bisa diperoleh dari solusi yang lain dengannya beberapa langkah.

Penggunaan algoritma genetika untuk memecahkan suatu masalah ditunjukkan pada Gambar 2.1. Penyelesaian suatu masalah harus dipetakan (*encoding*) dalam string kromosom. String kromosom ini disusun dari sejumlah *gen* yang mengilustrasikan variabel-variabel keputusan yang digunakan dalam penyelesaian masalah. Fungsi *fitness* dan representasi string kromosom digunakan untuk menilai seberapa bagus sebuah kromosom dimasukkan ke algoritma genetika. Algoritma genetika akan menghasilkan kromosom ‘terbaik’ setelah melewati sekian generasi dengan menirukan proses genetika dan seleksi alami. Kromosom ‘terbaik’ ini harus diuraikan (*decoding*) menjadi sebuah solusi yang diharapkan mendekati optimum.



Gambar 2.1 Penyelesaian Masalah dengan Algoritma Genetika

Apabila $P(t)$ merupakan populasi (*parents*) dan $C(t)$ merupakan keturunan (*offspring*) pada generasi ke- t , maka struktur umum algoritma genetika dapat dideskripsikan sebagai berikut:

```

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

Proses dalam algoritma genetika dimulai dengan menciptakan individu-individu secara acak dengan susunan gen (kromosom) tertentu yang disebut dengan proses inisialisasi. Solusi dari permasalahan yang akan

dipecahkan diwakili oleh kromosom. Untuk menghasilkan keturunan (*offspring*) dari individu-individu yang ada dalam populasi dilakukan reproduksi. Evaluasi digunakan untuk menghitung kebugaran (*fitness*) dari tiap kromosom. Semakin besar nilai *fitness* maka semakin baik sebuah kromosom untuk dijadikan calon solusi. Individu dari himpunan populasi dan *offspring* yang dipertahankan hidup pada generasi berikutnya dipilih melalui proses seleksi. Individu yang dipertahankan hidup dipilih menggunakan fungsi probabilistik. Individu yang lebih baik dengan nilai kebugaran/*fitness* lebih besar mempunyai peluang lebih besar untuk terpilih (Gen & Cheng dalam Mahmudy, 2013).

Individu terbaik akan didapatkan setelah melewati sekian iterasi (generasi). Individu terbaik ini memiliki susunan kromosom yang dapat dikonversi menjadi solusi terbaik (minimal mendekati optimum). Dengan ini dapat disimpulkan solusi optimum dalam algoritma genetika dihasilkan dengan melakukan pencarian di antara sejumlah alternatif titik optimum berdasarkan fungsi probabilistik (Michalewicz dalam Mahmudy, 2013).

2.3 Home Industry

2.3.1 Pengertian Home Industry

Home berarti rumah, tempat tinggal, ataupun kampung halaman. Sedangkan *industry*, dapat diartikan sebagai kerajinan, usaha produk barang dan ataupun perusahaan. Singkatnya, *home industry* adalah rumah usaha produk barang atau juga perusahaan kecil. Dikatakan sebagai perusahaan kecil karena jenis kegiatan ekonomi ini dipusatkan di rumah. Pengertian usaha kecil secara jelas tercantum dalam UU No. 9 Tahun 1995, yang menyebutkan bahwa usaha kecil adalah usaha dengan kekayaan bersih paling banyak Rp 200 juta (tidak termasuk tanah dan bangunan tempat usaha) dengan hasil penjualan tahunan paling banyak Rp1.000.000.000.

Kriteria lainnya dalam UU No 9 Tahun 1995 adalah: milik WNI, berdiri sendiri, berafiliasi langsung atau tidak langsung dengan usaha menengah atau besar dan berbentuk badan usaha perorangan, baik berbadan hukum maupun

tidak. *Home* industri juga dapat berarti industri rumah tangga, karena termasuk dalam kategori usaha kecil yang dikelola keluarga (Selawati, 2007).

2.3.2 Pelaku *Home Industry*

Pada umumnya, pelaku kegiatan ekonomi yang berbasis di rumah ini adalah keluarga itu sendiri ataupun salah satu dari anggota keluarga yang berdomisili di tempat tinggalnya itu dengan mengajak beberapa orang di sekitarnya sebagai karyawannya. Meskipun dalam skala yang tidak terlalu besar, namun kegiatan ekonomi ini secara tidak langsung membuka lapangan pekerjaan untuk sanak saudara ataupun tetangga di kampung halamannya. Dengan begitu, usaha perusahaan kecil ini otomatis dapat membantu program pemerintah dalam upaya mengurangi angka pengangguran. Lagi, jumlah penduduk miskinpun akan berangsur menurun.

2.3.3 Pusat Kegiatan *Home Industry*

Sebagaimana nama kegiatan ekonomi ini, *home industry* pada umumnya memusatkan kegiatan di sebuah rumah keluarga tertentu dan biasanya para karyawan berdomisili di tempat yang tak jauh dari rumah produksi tersebut. Karena secara geografis dan psikologis hubungan mereka sangat dekat (pemilik usaha dan karyawan), memungkinkan untuk menjalin komunikasi sangat mudah. Dari kemudahan dalam berkomunikasi ini diharapkan dapat memicu etos kerja yang tinggi. Karena masing-masing merasa bahwa kegiatan ekonomi ini adalah milik keluarga, kerabat dan juga warga sekitar. Merupakan tanggung jawab bersama dalam upaya meningkatkan perusahaan mereka.

2.4 Laba

Laba (*income*) atau *earnings* atau *profit* merupakan rangkuman hasil bersih kegiatan operasi usaha dalam kurun waktu tertentu yang dinyatakan dalam istilah finansial. Laba adalah informasi perusahaan yang paling diminati dalam pasar uang. Secara konsep, laba ditugaskan untuk menyediakan sampai sejauh mana perusahaan mampu menutup biaya operasi dan menghasilkan pengembalian kepada pemegang sahamnya, baik pengukuran perubahan kekayaan pemegang saham selama periode ataupun mengestimasi laba usaha sekarang. Sedangkan secara khusus peran lain dari laba adalah sebagai indikator profitabilitas

perusahaan, yang sangat krusial bagi seorang analis karena membantu dalam mengestimasi potensi laba di masa depan yang merupakan tugas terpenting dalam analisis usaha (Subramanyam dan Wild, 2012).

Rumus untuk perhitungan laba dijelaskan dalam Persamaan 2.1 dan 2.2.

Margin laba kotor = $(\text{penjualan} - \text{harga pokok penjualan}) / \text{penjualan}$(2.1)

Margin laba operasi bersih = $\text{laba operasi} / \text{penjualan}$(2.2)



BAB III

METODOLOGI PENELITIAN

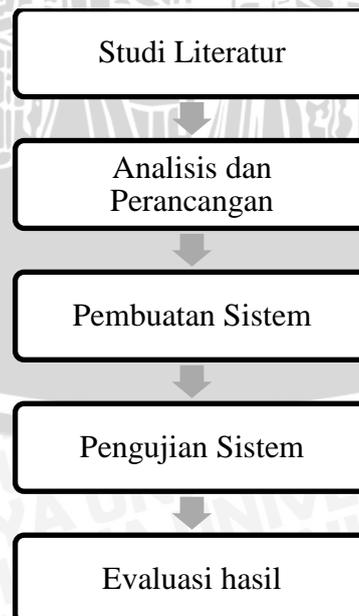
Bab ini menjelaskan langkah-langkah yang digunakan dalam penelitian optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika yang meliputi: tahapan penelitian, kebutuhan sistem, formulasi masalah, dan siklus genetika.

3.1 Tahapan Penelitian

Pada sub bab tahapan penelitian ini, akan dibahas langkah-langkah yang digunakan dalam pembuatan sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika. Tahapan pembuatannya adalah sebagai berikut :

1. Melakukan studi literatur mengenai algoritma genetika dalam permasalahan optimasi laba.
2. Menganalisis dan merancang sistem.
3. Membuat sistem berdasarkan analisa dan perancangan yang dilakukan.
4. Melakukan ujicoba terhadap sistem.
5. Melakukan evaluasi hasil yang diperoleh dari uji coba tersebut dengan cara membandingkan generasi tiap populasi.

Diagram tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.2 Kebutuhan Sistem

Kebutuhan sistem yang dibahas meliputi kebutuhan perangkat keras dan perangkat lunak yang digunakan untuk implementasi. Kebutuhan sistem dibahas secara detail agar implementasi berjalan sesuai tujuan.

3.2.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras pada sistem optimasi laba pada produksi jilbab home industry menggunakan algoritma genetika menggunakan komputer yang dijelaskan pada Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Prosesor	Intel Core i3
Memori(RAM)	2GB
Hardisk	Hardisk Free 50GB
Kartu Grafis	1024MB GetForce GT 520M
Monitor	Screen Monitor 14,0'', Resolusi 1024x600

3.2.2 Spesifikasi Perangkat Lunak

Pembangunan sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika ini menggunakan spesifikasi perangkat lunak yang dirinci pada tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Lunak Komputer

Tools	Nama Software
Program	Java
Dokumentasi	Microsoft Office 2007 (Ms.Word)
Sistem Operasi	Windows 8 64 bit

3.2.3 Batasan-Batasan Dalam Implementasi

Beberapa batasan yang diterapkan pada sistem optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika ini adalah :

- *Input* yang diterima oleh sistem dibaca melalui file dengan parameter model jilbab, jumlah kain, benang, busa, dan biaya jahit.
- Output yang diterima berupa hasil perhitungan dengan algoritma genetika.
- IDE (*Integrated Development Environment*) yang digunakan adalah Netbeans 8.0.1.

- Platform pengembangan yang digunakan adalah Java(TM) SE Runtime Environment 1.8.0-ea-b91.
- Metode yang digunakan yaitu algoritma genetika
- Atribut yang digunakan yaitu terdiri dari 5 variabel:
 1. Model jilbab
 2. Kain (tidak mempertimbangkan aspek warna)
 3. Benang
 4. Busa
 5. Biaya jahit

3.3 Formulasi Masalah

Kasus yang akan dibahas dalam penelitian ini adalah masalah optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika. Peneliti mengambil sampel data yang bersumber dari *home industry* jilbab melalui wawancara kepada narasumber yaitu pemilik *home industry*. Data berupa beberapa model jilbab beserta dengan bahannya, persediaan yang dimiliki perusahaan dan laba yang biasa diperoleh dari produksi masing-masing jilbab. Rincian data adalah sebagai berikut:

1. Sepuluh model jilbab *best selling* dalam perusahaan
2. Bahan pembuatan jilbab, antara lain kain, benang, dan busa.
3. Biaya penjahitan masing-masing jilbab.
4. Persediaan kain, benang, dan busa dalam perusahaan selama satu minggu.
5. Modal biaya penjahitan jilbab selama satu minggu.
6. Laba masing-masing penjualan jilbab secara grosir, eceran, dan distributor.

Dari data tersebut akan disusun model matematisnya lalu dihitung dengan algoritma genetika untuk menentukan bagaimana kombinasi penjualan jilbab yang paling optimum untuk mendapatkan laba maksimum. Dalam kasus ini persediaan bahan baku pembuatan jilbab (kain, benang, busa) selama satu minggu berturut-turut adalah 156,25 kg, 125 roll, dan 550 buah, serta modal biaya jahit sebesar Rp 3.000.000,-. Dalam perhitungan untuk penelitian ini, penggunaan kain tidak mempertimbangkan warna sehingga hanya terdapat satu variabel untuk kain. Keuntungan penjualan penjualan jilbab seperti yang

dijelaskan dalam Tabel 3.3, dengan persentase penjualan eceran, grosir, dan distributor masing-masing 73,52%, 18,72%, dan 7,76%. Persentase penjualan dihitung secara manual berdasarkan data total transaksi penjualan yang telah dilakukan selama beberapa bulan.

Tabel 3.3 Laba Penjualan Jilbab *Home Industry* dalam Rupiah

	Lab a Eceran (LA)	Lab a Grosir (LB)	Lab a Distributor (LC)
x ₁	15960	9960	7960
x ₂	3876	1376	376
x ₃	6195	2195	695
x ₄	12350	7350	5850
x ₅	13130	8130	5630
x ₆	12750	7750	5250
x ₇	14950	8950	6950
x ₈	7195	2195	695
x ₉	13050	8550	6050
x ₁₀	14050	9550	8050

Jika banyaknya jilbab yang akan diproduksi dilambangkan dengan x₁-x₁₀, maka fungsi tujuan dapat dinyatakan sebagai:

$$\begin{aligned} \text{maksimumkan } f(x_1, \dots, x_{10}) = & 73,52\%((L A x_1 * x_1) + \dots + (L A x_{10} * x_{10})) + \\ & 18,72\%((L B x_1 * x_1) + \dots + (L B x_{10} * x_{10})) + \\ & 7,76\%((L C x_1 * x_1) + \dots + (L C x_{10} * x_{10})). \dots \dots \dots (3.1) \end{aligned}$$

Kebutuhan bahan baku dan biaya jahit untuk tiap jenis jilbab dijelaskan pada Tabel 3.4.

Tabel 3.4 Kebutuhan Bahan Baku dan Biaya Jahit dalam Produksi Jilbab

	Kain (kg)	Benang (roll)	Busa	Ongkos Jahit (Rp)
x ₁	0,4	0,2	1	7000
x ₂	0,111	0,091	1	3000
x ₃	0,167	0,125	1	4000
x ₄	0,2	0,125	0	4500
x ₅	0,25	0,1	0	3000
x ₆	0,25	0,167	1	5000
x ₇	0,27	0,167	1	4500
x ₈	0,167	0,125	1	5000
x ₉	0,2	0,125	1	5000
x ₁₀	0,2	0,125	1	4000

Jika dalam produksi jilbab terjadi kekurangan bahan produksi maka akan menjadi kendala dalam proses produksi. Jumlah jilbab yang akan diproduksi dikalikan dengan banyaknya kebutuhan bahan per model tidak boleh melebihi persediaan selama satu minggu. Terdapat empat jenis kendala berdasarkan bahan baku yang dibutuhkan, yaitu kendala kain, benang, busa, dan biaya jahit. Kendala ketersediaan bahan baku dinyatakan sebagai berikut:

$$\text{kendala 1: } 0,4x_1 + 0,111x_2 + 0,167x_3 + 0,2x_4 + 0,25x_5 + 0,25x_6 + 0,27x_7 + 0,167x_8 + 0,2x_9 + 0,2x_{10} \leq 156,25 \dots\dots\dots(3.2)$$

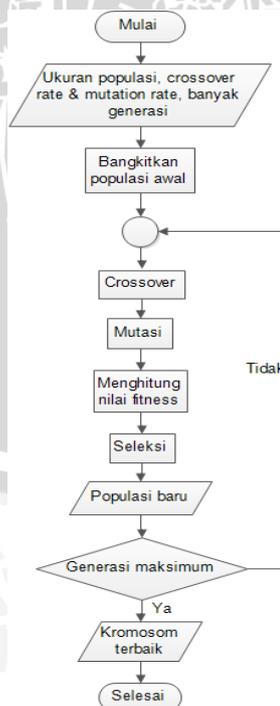
$$\text{kendala 2: } 0,2x_1 + 0,091x_2 + 0,125x_3 + 0,125x_4 + 0,1x_5 + 0,167x_6 + 0,167x_7 + 0,125x_8 + 0,125x_9 + 0,125x_{10} \leq 125 \dots\dots\dots(3.3)$$

$$\text{kendala 3: } 1x_1 + 1x_2 + 1x_3 + 0x_4 + 0x_5 + 1x_6 + 1x_7 + 1x_8 + 1x_9 + 1x_{10} \leq 550 \dots\dots\dots(3.4)$$

$$\text{kendala 4: } 7000x_1 + 3000x_2 + 4000x_3 + 4500x_4 + 3000x_5 + 5000x_6 + 4500x_7 + 5000x_8 + 5000x_9 + 4000x_{10} \leq 3000000 \dots\dots\dots(3.5)$$

3.4 Siklus Genetika

Pada subbab ini akan dibahas proses perhitungan untuk optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika. Diagram alir siklus perhitungan algoritma genetika ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Alir Algoritma Genetika



Langkah pertama yang dilakukan adalah *generate* populasi awal secara random sebanyak n individu. Selanjutnya, lakukan proses *crossover* dengan memilih dua induk (*parent*) secara acak dari populasi, kemudian lakukan proses mutasi pada kromosom induk yang dipilih secara acak. Langkah selanjutnya menghitung nilai *fitness* pada masing-masing kromosom. Langkah terakhir adalah melakukan proses seleksi dengan metode *elitsm selection* untuk menentukan individu yang akan dibangkitkan pada generasi berikutnya. Hasil akhir akan didapatkan kromosom terbaik pada generasi terakhir.

3.4.1 Representasi Kromosom

Pada proses perhitungan optimasi laba ini kromosom akan dikodekan dalam *string* kromosom desimal (*real*). Kromosom memiliki panjang *string* sepuluh, sesuai dengan banyaknya model jilbab yang digunakan sebagai variabel keputusan. Isi kromosom mempunyai rentang 0 hingga 140. Rentang nilai kromosom ditentukan melalui perhitungan manual supaya mendapatkan range yang tidak melebihi maupun kurang dari batas kemampuan produksi. Contoh sebuah kromosom ditampilkan pada Tabel 3.5.

Tabel 3.5 Representasi Kromosom

	Kromosom									
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
P	11,4	96,7	119,5	88,5	121,4	9,1	67,3	127,1	86,5	17,8
1	3	6	2	9	6	6	3	9	7	5

Angka dalam kromosom merupakan jumlah banyaknya jilbab yang akan diproduksi dalam bentuk bilangan pecahan (*real*). Karena permasalahan ini membutuhkan solusi dalam bentuk bilangan bulat maka dalam perhitungan *fitness* nilai kromosom dibulatkan terlebih dahulu.

Setelah menentukan representasi kromosom selanjutnya ditentukan fungsi tujuan (*objective functions*) dan kendala (*constraint*). Fungsi tujuan merepresentasikan tujuan yang ingin dioptimalkan, baik dimaksimalkan maupun diminimumkan (Mahmudy, 2013). Fungsi tujuan dan kendala telah direpresentasikan pada Persamaan (3.1) sampai dengan Persamaan (3.5) pada subbab sebelumnya.

Berdasarkan fungsi tujuan dan kendala yang telah dibuat kemudian ditentukan rumus perhitungan *fitness*. Penentuan rumus perhitungan *fitness* harus

dilakukan secara tepat agar solusi optimum dapat ditemukan secara efisien pada optimasi fungsi berkendala. Beberapa aturan diadopsi dari (Mahmudy dan Rahman, 2011) untuk penentuan individu yang lebih baik dapat dinyatakan sebagai berikut:

- Jika tidak ada pelanggaran kendala maka sebuah individu dikatakan lebih baik dari individu yang lain apabila nilai fungsi tujuannya lebih besar (berlaku untuk kasus maksimasi).
- Jika terjadi pelanggaran terhadap minimal satu kendala oleh individu-individu maka dipilih pelanggaran kendala dengan total yang lebih kecil. Hal ini untuk menjamin sebanyak mungkin solusi yang dipilih memenuhi kendala.

Berdasarkan kedua aturan tersebut dapat disusun fungsi *fitness* sebagai berikut:

$$fitness(x_1, \dots, x_{10}) = f(x_1, \dots, x_{10}) - 10000 (c_1 + c_2 + c_3 + c_4) \dots (3.6)$$

$$c_1 = \begin{cases} 0, & \text{jika } 0,4x_1 + \dots + 0,2x_{10} \leq 156,25 \\ (0,4x_1 + \dots + 0,2x_{10}) - 156,25, & \text{selainnya} \end{cases}$$

$$c_2 = \begin{cases} 0, & \text{jika } 0,2x_1 + \dots + 0,125x_{10} \leq 125 \\ (0,2x_1 + \dots + 0,125x_{10}) - 125, & \text{selainnya} \end{cases}$$

$$c_3 = \begin{cases} 0, & \text{jika } 1x_1 + \dots + 1x_{10} \leq 550 \\ (1x_1 + \dots + 1x_{10}) - 550, & \text{selainnya} \end{cases}$$

$$c_4 = \begin{cases} 0, & \text{jika } 7000x_1 + \dots + 4000x_{10} \leq 3000000 \\ (7000x_1 + \dots + 4000x_{10}) - 3000000, & \text{selainnya} \end{cases}$$

Fungsi tujuan ($f(x_1, \dots, x_{10})$) digunakan untuk menghitung laba dari masing-masing jilbab yang akan diproduksi yang kemudian dikalikan dengan persentase masing-masing laba (laba eceran, grosir, dan distributor). Sedangkan $c_1 + c_2 + c_3 + c_4$ merupakan hasil penjumlahan penalti karena terjadinya pelanggaran kendala. Hasil penjumlahan kendala akan dikalikan dengan pengali sebesar 10000. Pengali berupa bilangan yang cukup besar digunakan untuk menghasilkan solusi yang tidak memiliki penalti. Contoh perhitungan *fitness* dari P1 (Tabel 3.5) diberikan pada Tabel 3.6.

Tabel 3.6 Contoh Perhitungan *Fitness*

	$f(x_1, \dots, x_{10})$	c_1	c_2	c_3	c_4	<i>Fitness</i>
P1	6560633,807	0,00	0,00	0,00	3040219,98	3520414

3.4.2 Inisialisasi Populasi

Populasi awal dibangkitkan secara random. Misal ditentukan $popSize = 5$ maka akan dihasilkan populasi seperti yang ditunjukkan pada Tabel 3.7.

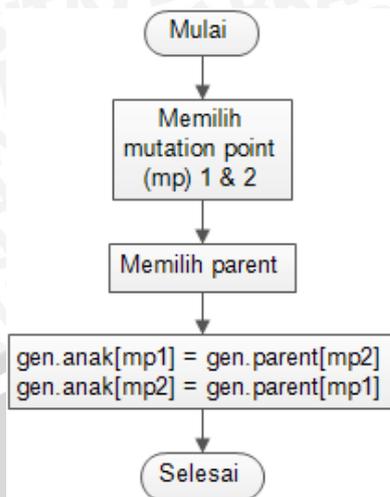
Tabel 3.7 Populasi Inisial

	Kromosom										Fitness
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀	
P1	11,43	96,76	119,52	88,59	121,46	9,16	67,33	127,19	86,57	17,85	3520414
P2	82,33	88,59	86,46	15,76	26,84	60,41	134,80	77,50	136,13	15,42	5061913
P3	90,96	25,31	71,49	27,40	72,18	108,47	33,31	3,51	25,88	17,13	4564116
P4	15,28	1,45	33,46	56,90	53,29	45,26	56,08	3,36	13,97	54,07	3642608
P5	118,48	12,65	61,71	98,23	111,55	126,92	101,49	92,28	78,27	26,10	4574316

3.4.3 Reproduksi

Pada proses reproduksi digunakan proses *crossover* dan mutasi untuk menghasilkan keturunan (*offspring*) dari induk yang telah dibangkitkan sebelumnya. Jika *crossover rate* yang digunakan sebesar 0.4, maka nantinya akan didapatkan $0.4 \times 5 = 2$ anak dari proses *crossover*. Sedangkan *mutation rate* jika ditentukan sebesar 0.6, akan menghasilkan 3 anak dari proses mutasi, sehingga total keturunan yang dihasilkan sebanyak 5 *offsprings*.

Metode *crossover* yang digunakan adalah metode *one-cut-point crossover*. Pada metode ini yang pertama dilakukan adalah memilih dua induk yang akan di-*crossover* lalu menentukan *cut point*. Satu kali proses *crossover* menghasilkan satu anak. Anak hasil *crossover* akan memiliki kromosom dari induk pertama sepanjang *cutpoint*, dan mendapatkan kromosom sisanya dari induk kedua. *Cut point* yang digunakan dalam penelitian sepanjang 5 gen dan bernilai konstan atau tetap. Diagram alir yang menunjukkan proses *crossover* ditunjukkan pada Gambar 3.3.



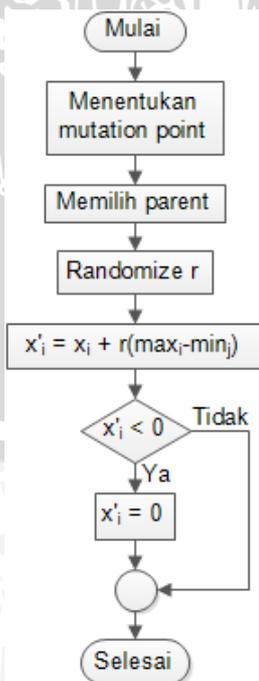
Gambar 3.5 Diagram Alir Proses *Reciprocal Exchange Mutation*

Contoh mutasi *reciprocal exchange* seperti ditunjukkan pada Gambar 3.6.

		↓ XP ₁							↓ XP ₂	
P3	90,96	25,31	71,49	27,40	72,18	108,47	33,31	3,51	25,88	17,13
C2	90,96	25,88	71,49	27,40	72,18	108,47	33,31	3,51	25,31	17,13

Gambar 3.6 Proses Mutasi *Reciprocal Exchange*

Metode *random mutation* dilakukan dengan mengurangi atau menambah nilai gen terpilih dengan bilangan *random* yang kecil (Mahmudy, 2013). Diagram alir yang menunjukkan proses *random mutation* ditunjukkan pada Gambar 3.7.



Gambar 3.7 Diagram Alir Proses *Random Mutation*

Misalkan domain variabel x_j adalah $[min_j, max_j]$ dan keturunan yang dihasilkan adalah $C=[x'_1..x'_n]$, maka nilai *offspring* bisa dibangkitkan sebagai berikut:

$$x'_i = x'_i + r(max_i - min_i) \dots \dots \dots (3.7)$$

Range r yang digunakan adalah $[0, 0,1]$. Misalkan yang terpilih sebagai induk adalah P2, gen yang terpilih nomor 2 (x_2) dan $r= 0,0584$. Maka akan dihasilkan *offspring* seperti pada Gambar 3.8.

P	118,4	12,6	61,7	98,2	111,5	126,9	101,4	92,2	78,2	26,1
5	8	5	1	3	5	2	9	8	7	0

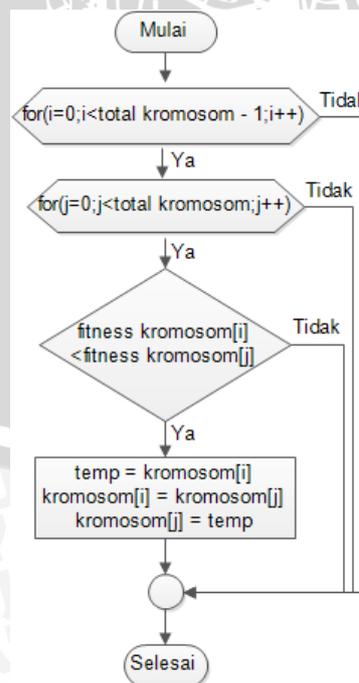
$x_2 = 12,65 - 0,0584 (140-0) = 4,47$

C	118,4	4,4	61,7	98,2	111,5	126,9	101,4	92,2	78,2	26,1
3	8	7	1	3	5	2	9	8	7	0

Gambar 3.8 Proses Mutasi *Random Mutation*

3.4.4 Seleksi

Seleksi menggunakan *elitism selection* melibatkan individu dalam *offspring* dan individu induk dalam populasi. Metode ini memilih *fitness* individu terbaik dari kumpulan individu di populasi (*parent*) dan *offspring* (Mahmudy, 2013). Diagram alir yang menunjukkan proses seleksi ditunjukkan pada Gambar 3.9.



Gambar 3.7 Diagram Alir Proses Seleksi



Dari proses seleksi didapatkan populasi baru seperti yang ditunjukkan pada Tabel 3.8.

Tabel 3.8 Individu Baru Hasil Seleksi

P	asa l	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	<i>Fitness</i>
P 1	p5	118,4 8	12,6 5	61,71	98,2 3	111,5 5	126,9 2	101,4 9	92,28	78,27	26,1 0	489861 1
P 2	c5	118,4 8	78,2 7	61,71	98,2 3	111,5 5	126,9 2	101,4 9	92,28	12,65	26,1 0	449458 0
P 3	p2	82,33	88,5 9	86,46	15,7 6	26,84	60,41	134,8 0	77,50	136,1 3	15,4 2	443527 5
P 4	c1	11,43	96,7 6	119,5 2	88,5 9	121,4 6	82,33	88,59	86,46	15,76	26,8 4	398659 6
P 5	p1	11,43	96,7 6	119,5 2	88,5 9	121,4 6	9,16	67,33	127,1 9	86,57	17,8 5	388398 3



BAB IV PERANCANGAN SISTEM

4.1 Perancangan *User Interface*

Perancangan *user interface* (UI) dari aplikasi ini terdiri dari satu halaman utama dimana halaman tersebut menampilkan data penelitian, setelah menekan tombol hitung akan menampilkan hasil perhitungan. Rancangan tampilan halaman utama dari sistem yang akan dibuat ditampilkan pada Gambar 4.1.

Optimasi Laba Produksi Jilbab

Model	Kain	Benang	Busa	Ong. Jahit	Laba A	Laba B	Laba C

Persediaan kain perminggu:
 Persediaan benang perminggu:
 Persediaan busa perminggu:
 Modal biaya jahit perminggu:
 Persentase penjualan eceran:
 Persentase penjualan grosir:
 Persentase penjualan distributor:

Popsize:

Banyak iterasi:

Crossover rate:

Mutation rate:

Gambar 4.1 Rancangan Interface Sistem: Halaman Utama

Keterangan:

1. Tabel 1 (kiri atas) berisi data model jilbab dan kebutuhan bahan penyusunnya.
2. Tabel 2 (kanan atas) berisi jumlah laba penjualan masing-masing laba A untuk eceran, laba B untuk grosir, dan laba C untuk distributor.
3. Beberapa label berisi data jumlah persediaan dan persentase masing-masing laba.
4. Label Popsize dengan *text field* untuk mengisi ukuran populasi sesuai inputan user.
5. Label Banyak Iterasi dengan *text field* untuk mengisi banyak generasi sesuai inputan user.



6. Label Crossover rate dengan *text field* untuk mengisi nilai *cr* sesuai inputan user.
7. Label Mutation rate dengan *text field* untuk mengisi nilai *mr* sesuai inputan user.
8. Button Hitung untuk mulai menghitung optimasi.

Setelah menginputkan parameter algoritma dan menekan tombol hitung, maka akan muncul tabel hasil perhitungan seperti pada Gambar 4.2.

Gambar 4.2 Rancangan Interface Sistem: Hasil Perhitungan

Keterangan:

1. Tabel 3 (kananatas) berisi individu yang di-generate oleh sistem dengan nilai x_1 - x_{10} bersifat *random* dengan rate 0-140 sebanyak ukuran populasi beserta keturunannya sebanyak *cr* dan *mr* dan nilai *fitness*-nya.
2. Tabel 4 (kanan bawah) berisi hasil seleksi individu beserta *fitness*.

4.2 Perancangan Uji Coba dan Evaluasi

Dikarenakan tidak adanya metode yang pasti untuk menentukan parameter algoritma genetika, maka untuk mengevaluasi program dilakukan beberapa uji coba antara lain:

1. Uji coba untuk menentukan ukuran populasi yang optimal.
2. Uji coba untuk menentukan banyaknya generasi yang optimal.

3. Uji coba untuk mencari kombinasi *crossoverrate* dan *mutation rate* yang terbaik.

4.2.1 Uji Coba Ukuran Populasi

Uji coba ukuran populasi dilakukan untuk mendapatkan ukuran populasi yang tepat agar menghasilkan solusi yang terbaik. Ukuran populasi mempengaruhi kemampuan algoritma genetika dalam menemukan solusi optimum. Banyaknya populasi yang digunakan adalah kelipatan 20. Rancangan uji coba ukuran populasi dapat dilihat pada Tabel 4.1. Parameter yang digunakan pada uji coba populasi antara lain sebagai berikut:

- a. Ukuran populasi : 20-200
- b. Ukuran generasi : 100
- c. *Crossover rate* : 0.5
- d. *Mutation rate* : 0.5

Tabel 4.1 Rancangan Uji Coba Ukuran Populasi

Banyak Populasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan populasi ke-										
	1	2	3	4	5	6	7	8	9	10	
20											
40											
60											
80											
100											
120											
140											
160											
180											
200											

4.2.2 Uji Coba Kombinasi *CrossoverRate* dan *MutationRate*

Uji coba berdasarkan *crossover rate* (*cr*) dan mutasi *rate* (*mr*) dilakukan untuk mengetahui kombinasi *cr* dan *mr* terbaik yang mendapatkan solusi optimum. Nilai *cr* dan *mr* yang digunakan antara 0 dan 1. Rancangan uji coba kombinasi *cr* dan *mr* ditampilkan pada Tabel 4.2. Parameter yang digunakan dalam uji coba antara lain:

- a. Jumlah populasi : Hasil populasi terbaik dari uji coba populasi
- b. Jumlah generasi : 100

Tabel 4.2 Rancangan Uji Coba Kombinasi cr dan mr

Kombinasi		Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
		Percobaan kombinasi cr dan mr ke-										
cr	mr	1	2	3	4	5	6	7	8	9	10	
1	0											
0,9	0,1											
0,8	0,2											
0,7	0,3											
0,6	0,4											
0,5	0,5											
0,4	0,6											
0,3	0,7											
0,2	0,8											
0,1	0,9											
0	1											

4.2.3 Uji Coba Jumlah Generasi

Uji coba jumlah generasi yaitu uji coba untuk mengetahui banyaknya generasi yang tepat untuk menghasilkan laba maksimum. Ukuran banyaknya generasi mempengaruhi kemampuan algoritma genetika dalam menemukan solusi optimum. Banyaknya generasi yang digunakan adalah kelipatan 50. Rancangan uji coba ukuran populasi dapat dilihat pada Tabel 4.3. Parameter yang digunakan pada uji coba populasi adalah :

- Ukuran populasi : Hasil populasi terbaik dari uji coba populasi
- Ukuran generasi : 50 - 550
- Cr dan mr : Hasil cr dan mr terbaik pada uji coba kombinasi cr dan mr

Tabel 4.3 Rancangan Uji Coba Jumlah Generasi

Banyak Generasi	Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
	Percobaan generasi ke-										
	1	2	3	4	5	6	7	8	9	10	
50											
100											
150											
200											
250											
300											
350											
400											
450											
500											
550											

BAB V IMPLEMENTASI

5.1 Implementasi Sistem

Berdasarkan metodologi penelitian dan perancangan sistem yang telah dijelaskan pada Bab 3 dan 4, maka selanjutnya akan dijelaskan implementasi proses penelitian dalam bentuk sistem.

5.1.1 Pembangkitan Populasi Awal

Proses pembangkitan populasi awal dilakukan dengan menginisialisasi kromosom berdasarkan hasil pembangkitan besar populasi yang telah dimasukan oleh pengguna. Suatu individu diperoleh dengan cara membangkitkan bilangan desimal acak (*random*) pada interval 0 sampai 140 yang menyatakan banyaknya jilbab yang akan diproduksi. Proses pembangkitan populasi awal dapat dilihat pada Source Code 5.1.

```
public Individu(int x) {  
    nKrom = x;  
    kromosom = new double[x];  
    Random rnd = new Random();  
    for (int i = 0; i < x; i++) {  
        double tmp;  
        int num;  
        num = rnd.nextInt(14000) + 0; //randomize 0-140  
        tmp = num / 100.0;  
        kromosom[i] = tmp;  
    }  
}
```

Source Code 5.1 Pembangkitan Populasi Awal

5.1.2 Perhitungan Nilai *Fitness*

Perhitungan nilai *fitness* diperoleh dengan menghitung nilai fungsi tujuan lalu dikurangi dengan hasil penambahan nilai empat penalti atau kendala. Terdapat tiga fungsi tujuan berdasarkan masing-masing laba yaitu laba eceran, grosir, dan distributor. Fungsi tujuan merupakan nilai hasil kali persentase laba dengan hasil penjumlahan laba tiap jenis jilba yang dikali dengan banyaknya produksi jilbab (kromosom). Tiga fungsi tujuan yang telah dihitung akan ditotal

menjadi satu fungsi tujuan. Terdapat empat jenis kendala berdasarkan empat kebutuhan pembuatan jilbab yaitu kain, benang, busa dan biaya jahit. Nilai kendala merupakan hasil penjumlahan dari kebutuhan masing-masing jenis jilbab dikalikan dengan kromosom. Nilai kendala akan bernilai 0 jika hasil perhitungan kendala kurang dari persediaan. Perhitungan *fitness* dilakukan pada setiap individu. Proses perhitungan *fitness* dapat dilihat pada Source Code 5.2.

```
double[] k1 = new double[popsize+nCr+nMt];
double[] k2 = new double[popsize+nCr+nMt];
double[] k3 = new double[popsize+nCr+nMt];
double[] k4 = new double[popsize+nCr+nMt];
for(int j=0;j<(popsize+nCr+nMt);j++){
for(int k=0;k<nKrom;k++){
    k1[j]+= keb.kain[k]*(int)pop[j].kromosom[k];
    k2[j]+= keb.benang[k]*(int)pop[j].kromosom[k];
    k3[j]+= keb.busu[k]*(int)pop[j].kromosom[k];
    k4[j]+= keb.ongkos[k]*(int)pop[j].kromosom[k];
}
}
for(int i=0;i<(popsize+nCr+nMt);i++){
    if(k1[i]<pkain){
        k1[i]= 0;
    }
    else{
        k1[i] = k1[i] - pkain;
    }
    if(k2[i]<pbenang){
        k2[i]= 0;
    }
    else{
        k2[i] = k2[i] - pbenang;
    }
    if(k3[i]<pbusa){
        k3[i]= 0;
    }
    else{
        k3[i] = k3[i] - pbusa;
    }
    if(k4[i]<pjahit){
```

```

        k4[i]= 0;
    }
    else{
        k4[i] = k4[i] - pjahit;
    }
}
double[] fxa = new double[popsize+nCr+nMt];
double[] fxb = new double[popsize+nCr+nMt];
double[] fxc = new double[popsize+nCr+nMt];
double[] fx = new double[popsize+nCr+nMt];
for(int i=0;i<(popsize+nCr+nMt);i++){
double tmp1=0;
double tmp2=0;
double tmp3=0;
for(int k=0;k<nKrom;k++){
    tmp1 +=1.laba1[k]*(int)pop[i].kromosom[k];
    tmp2 +=1.laba2[k]*(int)pop[i].kromosom[k];
tmp3 +=1.laba3[k]*(int)pop[i].kromosom[k];
}
fxa[i]= tmp1 * persenA;
fxb[i]= tmp2 * persenB;
fxc[i]= tmp3 * persenC;
fx[i] = fxa[i] + fxb[i] + fxc[i];
}
for(int i=0;i<(popsize+nCr+nMt);i++){
    pop[i].fitness = (long) (fx[i] - (10000 *
(k1[i]+k2[i]+k3[i]+k4[i])));
}

```

Source Code 5.2 Perhitungan Nilai *Fitness*

5.1.3 Proses *Crossover*

Metode *crossover* yang digunakan adalah *one-cut-point crossover*. Hal pertama yang dilakukan adalah menentukan panjang *cut point*. Selanjutnya dilakukan pemilihan dua induk yang akan menjalani proses *crossover* dengan melakukan pembangkitan bilangan secara acak sepanjang kromosom yang telah dibangkitkan pada proses inialisasi populasi awal. Individu yang terpilih yang akan menjalani proses *crossover*. Anak dari proses *crossover* sebanyak hasil

perkalian dari ukuran populasi dan nilai *crossover rate*. Proses *crossover* dapat dilihat pada Source Code 5.3.

```
for (int idCr=1; idCr<=nCr; idCr++){
    int idChd = (popsize+idCr)-1;
    int p1,p2;
    p1 = rnd.nextInt(popsize);
    do {
        p2 = rnd.nextInt(popsize);
    } while (p2==p1);
    int cP = nKrom / 2;
    int idx = 0;
    for (int i=0; i<cP; i++){
        pop[idChd].kromosom[i] = pop[p1].kromosom[i];
    }
    for (int i=cP; i<nKrom; i++){
        pop[idChd].kromosom[i]= pop[p2].kromosom[idx];
        idx++;
    }
}
```

Source Code 5.3 Proses Crossover

5.1.4 Proses Mutasi

Metode mutasi yang digunakan adalah *reciprocal exchange mutation* dan *random mutation*. Pada metode *reciprocal exchange mutation* hal pertama yang dilakukan adalah menentukan titik yang akan ditukar dalam proses mutasi dengan memilih angka acak sepanjang ukuran populasi. Langkah selanjutnya yaitu melakukan pemilihan individu yang akan mengalami proses mutasi dengan cara memilih secara acak dari individu yang telah dibangkitkan pada proses inisialisasi populasi awal. Individu yang terpilih akan menjalani proses mutasi. Hasil anak dari proses mutasi sebanyak perkalian dari jumlah populasi dengan nilai *mutation rate*. Source code metode *reciprocal exchange mutation* dapat dilihat pada Source Code 5.4.

```
int mP1,mP2,idxM;
mP1 = rnd.nextInt(nKrom);
do {
    mP2 = rnd.nextInt(nKrom);
```

```

} while (mP2==mP1);
boolean[] sudahPilih = new boolean[popsize+1];
for (int i=0; i<=popsize; i++){
    sudahPilih[i] = false;
}
for (int idMt=1; idMt<=nMt; idMt++){
    do {
        idxM = rnd.nextInt(popsize);
    } while (sudahPilih[idxM]);
    sudahPilih[idxM] = true;
    int idChd = (popsize+nCr+idMt)-1;
    double isi1 = pop[idxM].kromosom[mP2];
    double isi2 = pop[idxM].kromosom[mP1];
    for(int i=0;i<nKrom;i++){
        pop[idChd].kromosom[i]= pop[idxM].kromosom[i];
    }
    pop[idChd].kromosom[mP1] = isi1;
    pop[idChd].kromosom[mP2] = isi2;
}
}

```

Source code 5.4 Proses Mutasi *Reciprocal Exchange Mutation*

Untuk metode random mutation, hal pertama yang dilakukan adalah memilih gen dan kromosom *parent* yang akan dimutasi. Individu yang terpilih akan menjalani proses mutasi. Gen yang telah dipilih akan dikurangi dengan nilai r yang dibangkitkan secara acak dengan *range* $[-0,1-0,1]$ yang sebelumnya dikalikan dengan batas atas dikurangi dengan batas bawah *range* kromosom. Source code metode *random mutation* dapat dilihat pada Source Code 5.5.

```

int mP1,idxM;
    mP1 = rnd.nextInt(nKrom);
    boolean[] sudahPilih = new boolean[popsize+1];
    for (int i=0; i<=popsize; i++){
        sudahPilih[i] = false;
    }
    for (int idMt=1; idMt<=nMt; idMt++){
        do {
            idxM = rnd.nextInt(popsize);

```

```

} while (sudahPilih[idxM]);
sudahPilih[idxM] = true;
int idChd = (popsize+nCr+idMt)-1;
double alfa = (new Random().nextDouble() - 0.5) * 0.1;
double gen = pop[idxM].kromosom[mP1] + (alfa * 140);
for(int i=0;i<nKrom;i++){
    pop[idChd].kromosom[i]= pop[idxM].kromosom[i];
}
if (gen<0){
    pop[idChd].kromosom[mP1] = 0;
}
else{
    pop[idChd].kromosom[mP1] = gen;
}
}

```

Source code 5.5 Proses Mutasi *Random Mutation*

5.1.5 Proses Seleksi

Proses seleksi pemilihan induk untuk generasi berikutnya dilakukan dengan metode *elitism selection*. Pada seleksi dengan metode *elitism* ini, nilai *fitness* berbanding lurus dengan kemungkinan terpilihnya suatu individu sebagai induk pada generasi berikutnya. Semakin besar nilai *fitness* maka semakin besar pula kemungkinan terpilihnya individu tersebut. Langkah pertama dari proses seleksi ini adalah dengan mengurutkan nilai *fitness* setiap individu dari nilai yang terbesar hingga terkecil, kemudian dipilih individu teratas sebanyak jumlah populasi awal untuk menjadi induk pada generasi berikutnya. *Source code* proses seleksi dapat dilihat pada Source Code 5.6.

```

Individu tmpIdv = new Individu(nKrom);
for (int i=0; i<((popsize+nCr+nMt)-1); i++){
    for (int j=i+1; j<(popsize+nCr+nMt); j++){
        if (pop[i].fitness<pop[j].fitness){
            tmpIdv = pop[i];
            pop[i] = pop[j];
            pop[j] = tmpIdv;
        }
    }
}
}

```

5.2 Implementasi Antarmuka Pengguna

Antarmuka pengguna pada implementasi ini terdiri dari *form* untuk halaman utama, dimana pada halaman utama ini pengguna dapat memasukkan data parameter perhitungan dan mendapatkan hasil perhitungan algoritma genetika dari proses optimasi.

5.2.1 Implementasi Halaman Utama

Halaman utama adalah halaman yang pertama kali muncul saat program dijalankan. Pada halaman utama ini, terdapat dua tabel yang masing-masing berisi data kebutuhan produksi jilbab dan laba penjualan jilbab. Terdapat keterangan persediaan kebutuhan produksi jilbab per minggu dan persentase penjualan untuk eceran, grosir, dan distributor. Kemudian terdapat field untuk pengguna memasukkan besar ukuran populasi, banyak iterasi, *crossover rate*, dan *mutation rate*. Gambar 5.1 merupakan implementasi halaman utama sistem.

Model	Kain	Bena	Busa	Ong...	Laba A	Laba B	Laba C
x1	0.4	0.2	1	7000	1590	990	790
x2	0.111	0.091	1	3000	3876	1376	376
x3	0.167	0.125	1	4000	6195	2195	695
x4	0.2	0.125	0	4500	12350	7350	5850
x5	0.25	0.1	0	3000	13130	8130	5330
x6	0.25	0.167	1	5000	12750	7750	5250
x7	0.27	0.167	1	4500	14950	8950	6950
x8	0.167	0.125	1	5000	7195	2195	695
x9	0.2	0.125	1	5000	13950	8950	6950
x10	0.2	0.125	1	4000	14050	9550	8050

Persediaan Kain per minggu : 156.25
 Persediaan benang per minggu : 125
 Persediaan busa per minggu : 550
 Modal biaya jahit per minggu : 3000000
 Persentase penjualan eceran (Laba A) : 73.52%
 Persentase penjualan grosir (Laba B) : 18.72%
 Persentase penjualan dist. (Laba C) : 7.76%

Popsize :
 Banyak iterasi :
 Crossover rate :
 Mutation rate :

Hitung

Gambar 5.1 Implementasi Halaman Utama

5.2.2 Implementasi Halaman Hasil Perhitungan

Halaman hasil perhitungan berisi hasil perhitungan algoritma genetika sesuai parameter yang dimasukkan oleh pengguna. Hasil akan muncul setelah pengguna mengklik tombol hitung. Hasil yang ditampilkan berupa representasi kromosom dan hasil seleksi masing-masing generasi ke-0 dan generasi terakhir. Gambar 5.2 merupakan implementasi dari halaman hasil perhitungan.

Optimasi Laba Produksi Jilbab

Model	Kain	Bena...	Busa	Ong...	LabA	LabA...	LabA...
x1	0.4	0.2	1	7000	15960	9960	7960
x2	0.111	0.091	1	3000	3876	1376	376
x3	0.167	0.125	1	4000	6195	2195	695
x4	0.2	0.125	0	4500	12350	7350	5850
x5	0.25	0.1	0	3000	13130	8130	5930
x6	0.25	0.167	1	5000	12750	7750	5250
x7	0.27	0.167	1	4500	14950	8950	6950
x8	0.167	0.125	1	5000	7195	2195	695
x9	0.2	0.125	1	5000	13950	8950	6950
x10	0.2	0.125	1	4000	14050	9550	8050

Persediaan Kain perminggu : 156.25
 Persediaan benang perminggu : 125
 Persediaan busa perminggu : 550
 Modal biaya jahit perminggu : 3000000
 Persentase penjualan eceran (Laba A) : 73.52%
 Persentase penjualan grosir (Laba B) : 18.72%
 Persentase penjualan dist. (Laba C) : 7.76%

Popsize : 200
 Banyak Iterasi : 450
 Crossover rate : 0.4
 Mutation rate : 0.6

Hitung

Idv	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Fitness
P1	71.55	34.97	109.89	93.73	61.18	78.15	39.98	30.47	96.27	88.33	-1752834624
P2	115.12	42.84	114.04	87.53	40.33	97.12	37.31	47.35	48.35	0.98	-243839121
P3	71.07	19.92	59.8	109.8	107.07	56.66	66.64	129.43	92.18	77.74	-5907294357
P4	20.29	17.23	96.54	43.31	52.74	5.23	115.83	74.31	91.97	26.25	5265890
P5	58.12	34.73	73.44	108.77	43.56	62.6	19.28	130.6	100.85	85.25	-2998189959

Idv	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Fitness
P1	63.61	11.71	8.21	118.76	129.14	47.18	134.99	59.06	29.71	69.78	7562639
P2	40.52	46.77	25.92	17.65	134.76	11.83	104.82	75.04	132.61	93.84	7233148
P3	40.52	46.77	25.92	17.65	134.76	10.8345	104.82	75.04	132.61	93.84	7221916
P4	3.26	35.69	69.53	133.63	126.37	50.28	24.8	64.7	118.71	87.93	7144641
P5	67.83	122.38	50.82	45.51	99.25	52.81	110.5	7.42	23.23	112.81	6998994

Idv	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Fitness
P1	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760
P2	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760
P3	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3339...	0.0	121.764...	156.569...	8125760
P4	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760
P5	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760

Idv	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Fitness
P1	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760
P2	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760
P3	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3339...	0.0	121.764...	156.569...	8125760
P4	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760
P5	60.9390	1.21828...	0.0	122.223...	153.399...	58.5987...	11.3938...	0.0	121.764...	156.569...	8125760

Gambar 5.2 Implementasi Halaman Hasil Perhitungan

Penjelasan solusi akhir ditampilkan dalam *console* pada Gambar 5.3.

```

Output X Notifications Search Results Variables Breakpoints
skripsi (debug) X Debugger Console X
debug:
Kromosom terbaik didapatkan pada generasi ke 450
P1: 60.93904932690297 1.2182846107773264 0.0 122.22366164121367 153.39900553149403 58.59876521901195 11.393821977
Kendala 1: 159.63099999999997
Kendala 2: 88.789000000000002
Kendala 3: 407.0
Kendala 4: 2999500.0

Nilai constraint (bernilai 0 jika nilai kendala kurang dari persediaan
c1: 3.3809999999999972
c2: 0.0
c3: 0.0
c4: 0.0

Fungsi tujuan untuk laba eceran (Laba A): 6728451.8832
Fungsi tujuan untuk laba grosir (Laba B): 1087981.3152
Fungsi tujuan untuk laba distributor (Laba C): 343136.8016
Fungsi tujuan keseluruhan: 8159570.0

Nilai fitness 8125760
    
```

Gambar 5.3 Penjelasan Hasil Akhir



BAB VI

PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas hasil uji coba yang telah dilakukan dalam optimasi laba pada produksi jilbab *home industry* menggunakan algoritma genetika. Pengujian dilakukan untuk mendapatkan parameter algoritma genetika yang optimum agar dapat menghasilkan nilai *fitness* yang paling optimum. Berdasarkan perancangan yang dijelaskan pada Bab 4, terdapat 3 jenis pengujian yang akan dilakukan antara lain pengujian ukuran populasi, pengujian kombinasi *cr* dan *mr*, serta pengujian ukuran generasi, dimana masing-masing pengujian tersebut akan dibandingkan terhadap nilai *fitness*.

6.1 Hasil Pengujian Sistem

Hasil pengujian terhadap sistem dilakukan untuk mengetahui kromosom terbaik dengan nilai *fitness* tertinggi yang dihasilkan dari proses algoritma genetika. Akan dilakukan dua pengujian untuk masing-masing penelitian menggunakan *random mutation* dan *reciprocal exchange mutation*.

6.1.1 Hasil Pengujian Sistem dengan *Reciprocal Exchange Mutation*

Dalam uji coba ini akan dilakukan pengujian terhadap masukan inisialisasi parameter algoritma genetika dengan ukuran populasi 20 kemudian diproses sejumlah 1 generasi, nilai *cr* dan *mr* masing-masing sebesar 0,5. Hasil pengujian sistem yang menghasilkan 20 kromosom terbaik pada generasi satu dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Sistem dengan *Reciprocal Exchange Mutation*

Idv	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	<i>Fitness</i>
P1	33.96	42.46	6.85	124.55	119.21	82.5	21.33	64.74	37.62	136.04	7207764
P2	8.64	39.34	55.68	39.23	94.69	65.71	101.29	18.24	138.02	92.64	7084386
P3	8.64	39.34	55.68	39.23	94.69	18.48	104.43	123.1	87.51	126.19	7035543
P4	32.25	54.57	83.36	55.04	67.64	60.41	42.77	50.2	132.71	99.22	6835555
P5	69.48	12.85	69.85	101.58	39.99	94.05	59.4	8.14	95.94	63.74	6828125
P6	83.59	99.24	14.78	22.04	66.01	120.16	90.28	77.66	1.13	87.22	6750309
P7	8.64	39.34	126.19	39.23	94.69	18.48	104.43	123.1	87.51	55.68	6485667
P8	20.79	14.4	35.2	97.15	44.96	76.6	123.12	108.28	55.91	34.26	6350671
P9	56.62	85.88	44.96	121.14	16.88	20.79	14.4	35.2	97.15	128.8	6271438
P10	20.79	14.4	35.2	97.15	44.96	76.6	108.28	123.12	55.91	34.26	6239036
P11	33.96	42.46	136.04	124.55	119.21	82.5	21.33	64.74	37.62	6.85	6200270

P12	83.59	99.24	87.22	22.04	66.01	120.16	90.28	77.66	1.13	14.78	6185382
P13	83.59	99.24	14.78	22.04	66.01	40.92	65.27	50.93	119.68	37.22	6117243
P14	29.64	66.85	49.61	24.91	78.86	61.13	46.55	85.03	52.17	125.28	6114511
P15	94.05	59.4	8.14	95.94	63.74	8.64	39.34	55.68	39.23	94.69	6003127
P16	33.96	42.46	6.85	124.55	119.21	41.43	95.73	7.83	30.68	32.37	5989228
P17	40.92	65.27	50.93	119.68	37.22	8.88	41.6	19.07	100.68	91.21	5925685
P18	8.64	39.34	55.68	39.23	94.69	94.05	59.4	8.14	95.94	63.74	5922288
P19	56.62	85.88	128.8	121.14	16.88	8.64	39.34	55.68	39.23	94.69	5888735
P20	56.62	85.88	128.8	121.14	16.88	8.71	90.24	14.85	42.16	49.02	5778163

6.1.2 Hasil Pengujian Sistem dengan *Random Mutation*

Dalam uji coba ini akan dilakukan pengujian terhadap masukan inisialisasi parameter algoritma genetika sama seperti pada percobaan *reciprocal exchange* yaitu dengan ukuran populasi 20 kemudian diproses sejumlah 1 generasi, nilai *cr* dan *mr* masing-masing sebesar 0,5. Hasil pengujian sistem yang menghasilkan 20 kromosom terbaik pada generasi satu dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Sistem dengan *Random Mutation*

Idv	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Fitness
P1	124.9 8	89.79	61.05	77.1 4	123.9 8	65.3	15.43	17.77	16.9 6	49.31	656704 6
P2	124.9 8	89.79	61.05	73.6 6	123.9 8	65.3	15.43	17.77	16.9 6	49.31	652884 6
P3	120.1 4	43.83	113.3 1	28.8 7	73.84	82.38	65.49	12.96	0.74	57.59	624632 4
P4	54.53	13.06	107.3 7	39.6 4	49.47	134.2 9	91.14	17.94	42.0 4	35.66	617386 1
P5	120.1 4	43.83	113.3 1	22.0 1	73.84	82.38	65.49	12.96	0.74	57.59	617087 2
P6	134.2 9	91.14	17.94	42.0 4	35.66	54.53	13.06	107.3 7	39.6 4	49.47	570098 4
P7	42.45	85.46	15.38	74.0 3	28.0	92.58	115.0 2	27.91	21.8 4	41.21	562826 5
P8	132.2	31.26	128.1 5	1.26	2.55	42.45	85.46	15.38	74.0 3	28.0	562549 3
P9	92.58	115.0 2	27.91	21.8 4	41.21	65.47	80.54	49.98	25.0 7	49.2	558662 1
P10	74.31	92.41	42.55	57.5 4	8.72	83.89	63.04	24.07	107. 8	0.17	550800 2
P11	120.1 4	43.83	113.3 1	28.8 7	73.84	65.3	15.43	17.77	16.9 6	49.31	549835 3
P12	74.31	92.41	42.55	51.1 5	8.72	83.89	63.04	24.07	107. 8	0.17	543765 8
P13	54.53	13.06	107.3 7	39.6 4	49.47	18.3	94.18	8.7	49.9 1	61.93	527722 0
P14	92.58	115.0 2	27.91	21.8 4	41.21	82.38	65.49	12.96	0.74	57.59	518471 1
P15	87.78	67.31	80.95	57.3 1	51.67	2.79	69.96	28.66	39.6 7	30.57	510581 3
P16	65.3	15.43	17.77	16.9 6	49.31	42.45	85.46	15.38	74.0 3	28.0	477834 4
P17	65.3	15.43	17.77	16.9 6	49.31	52.07	31.15	28.14	37.0 5	102.9 4	476620 5

P18	65.3	15.43	17.77	17.46	49.31	52.07	25.70	28.14	37.05	102.94	4699353
P19	65.3	15.43	17.77	16.96	49.31	52.07	25.70	28.14	37.05	102.94	4693746
P20	65.3	15.43	17.77	16.96	49.31	132.2	31.26	128.15	1.26	2.55	4543444

6.2 Hasil dan Analisis Pengujian Parameter Algoritma Genetika dengan *Reciprocal Exchange Mutation*

Proses uji coba ini dilakukan untuk mengetahui berapa *fitness* tertinggi dari kromosom terbaik pada setiap percobaan parameter algoritma genetika pada penelitian dengan metode mutasi *reciprocal exchange mutation*. Parameter algoritma genetika yang digunakan terdiri dari banyak populasi, banyak generasi, *crossover rate* dan *mutation rate*. Hasil pengujian ini kemudian dianalisis untuk diambil kesimpulan dari penelitian.

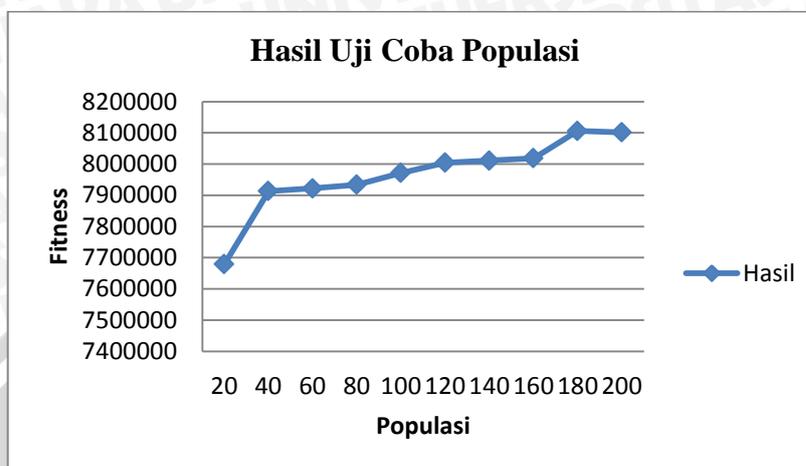
6.2.1 Hasil dan Analisis Uji Coba Ukuran Populasi

Pengujian ini bertujuan untuk mengetahui ukuran populasi yang optimum untuk menghasilkan rata-rata nilai *fitness* terbaik pada penelitian ini. Jumlah generasi yang dipakai adalah 100 dengan ukuran populasi kelipatan 20 dari 20 hingga 200 populasi. Nilai *crossover rate* dan *mutation rate* yang digunakan masing-masing sebesar 0,5. Setiap generasi dilakukan 10 kali percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba tersebut akan diperoleh berapa ukuran populasi optimum untuk menyelesaikan masalah pada penelitian. Hasil dari uji coba populasi bisa dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Uji Coba Populasi

Banyak Populasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan populasi ke-										
	1	2	3	4	5	6	7	8	9	10	
20	7430835	7469006	7646149	7541552	7976822	8116053	7441243	7826139	7781916	7566563	7679628
40	7630235	7840915	8076718	7971628	7883691	7743315	7969465	7992909	7990649	8031595	7913112
60	7958426	7816897	7884018	8033579	7934189	7926031	7868773	7907588	7874557	8015572	7921963
80	7906549	7833295	7869254	7995252	7838721	8021506	7893447	8025792	8060834	7890861	7933551
100	7940625	7789016	8099841	8053600	7980767	8054516	7868350	8013065	8103807	7813330	7971692
120	8034599	8069475	8090158	8000385	8080079	7916463	7976100	7855345	8042857	7976544	8004201
140	8017383	7997041	7972066	8195534	8049190	7893582	7988553	7886078	8038365	8071994	8010979
160	7962709	8027317	8114456	7983888	7966090	8012726	7962868	8057069	7998369	8100488	8018598
180	8107181	8037899	8181986	8135255	8124294	8170572	7951279	7983537	8246693	8124799	8106350
200	8131744	8222871	7960618	8060140	8175062	8082524	8079987	8068089	8108910	8126555	8101650

Dari data pada Tabel 6.3 dibuat sebuah grafik untuk melihat perbedaan nilai rata-rata *fitness* terhadap masing-masing ukuran populasi yang dicobakan. Grafik uji coba populasi ini dapat dilihat pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Uji Coba Ukuran Populasi

Dari grafik Gambar 6.1 dapat dilihat bahwa ukuran populasi berpengaruh terhadap hasil algoritma genetika yang terlihat pada nilai rata-rata *fitness*-nya. Nilai *fitness* terendah terdapat pada populasi 20 dikarenakan ukuran populasi masih terlalu rendah untuk memproses data sehingga algoritma genetika hanya mengevaluasi kemungkinan solusi yang terbatas dan sulit mendapatkan solusi yang mendekati optimum. Meskipun ukuran populasi yang lebih besar cenderung memberikan hasil yang lebih baik tetapi pada batas tertentu kenaikan *fitness* yang didapatkan kurang signifikan (Mahmudy, Marian, dan Luong, 2013). Ukuran populasi terlalu besar dapat membuat waktu proses semakin panjang. Dari hasil pada grafik dapat disimpulkan bahwa ukuran populasi yang optimal sebesar 180 dengan nilai rata-rata *fitness* 8101650, karena setelah jumlah populasi 180 sulit didapatkan nilai *fitness* yang lebih baik.

6.2.2 Hasil dan Analisis Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

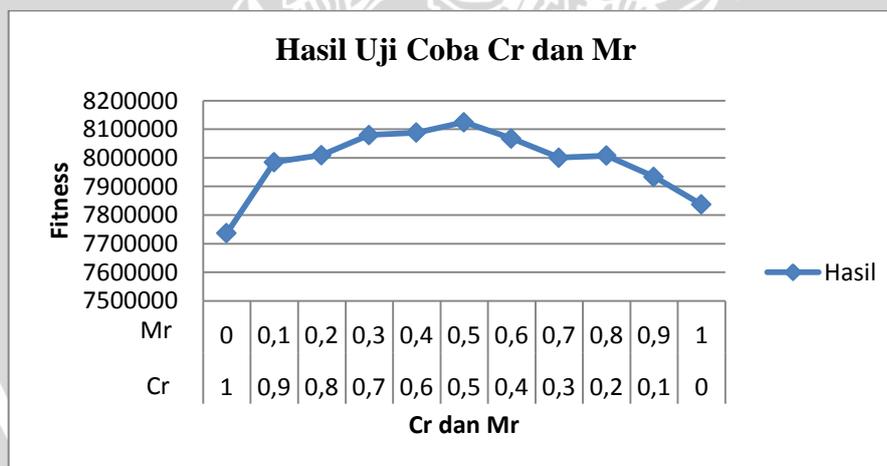
Uji coba kombinasi *crossover rate* (*cr*) dan *mutation rate* (*mr*) bertujuan untuk mengetahui kombinasi *cr* dan *mr* yang optimal pada penelitian. Ukuran populasi yang digunakan adalah yang menghasilkan nilai *fitness* terbaik pada uji coba ukuran populasi yaitu 180. Banyak generasi yang digunakan sebesar 100. Nilai *cr* dan *mr* yang digunakan dalam pengujian berkisar antara 0 sampai dengan

1. Setiap generasi dilakukan percobaan sebanyak 10 kali dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba tersebut akan didapatkan nilai *cr* dan *mr* yang paling optimal untuk menyelesaikan masalah. Hasil dari uji coba kombinasi *cr* dan *mr* dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hasil Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Kombinasi		Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
<i>cr</i>	<i>mr</i>	Percobaan kombinasi <i>cr</i> dan <i>mr</i> ke-										
		1	2	3	4	5	6	7	8	9	10	
1	0	7615800	7668573	7680682	7849379	7654688	7768626	7724849	7829775	7891879	7688256	7737251
0,9	0,1	7922879	7954952	7916133	8086029	7917853	7933304	7975922	7958991	7987478	8198920	7985246
0,8	0,2	8064589	8263268	7897489	7975886	8034484	8043596	7911898	7994140	7955780	7956605	8009774
0,7	0,3	8223612	8073969	8165238	8007993	8117619	8072485	7913569	8063033	8028029	8131288	8079684
0,6	0,4	8146840	7969571	8128486	7995057	8190197	8107922	8073425	8059166	8108058	8104040	8088276
0,5	0,5	8131744	8222871	8073088	8262581	8175062	8082524	8079987	7978718	8108910	8126555	8124204
0,4	0,6	7965585	8025959	8016707	8147585	8139429	7994210	8190183	8239979	7874704	8080614	8067496
0,3	0,7	7909845	8172577	8015105	8032345	8165688	7836165	8113995	8067736	7782027	7910348	8000583
0,2	0,8	7921264	8070848	8019950	7944223	7955710	7929656	8169245	7999884	8039999	8025089	8007587
0,1	0,9	8128641	7930643	7747536	7871127	8154270	7846236	7933836	7849891	8056679	7818915	7933777
0	1	7734002	7828906	7961738	7795881	7740260	7987134	7804955	7719207	8097773	7703016	7837287

Dari data pada Tabel 6.4 dibuat sebuah grafik untuk melihat perbedaan rata-rata nilai *fitness* dari hasil pengujian kombinasi *cr* dan *mr*. Grafik hasil uji coba kombinasi *cr* dan *mr* dapat dilihat pada Gambar 6.2.



Gambar 6.2 Hasil Uji Coba Kombinasi *Crossover rate* dan *Mutation rate*

Kombinasi *cr* dan *mr* yang tepat diperlukan agar algoritma genetika dapat memberikan solusi yang mendekati optimum (Mahmudy, Marian, dan Luong, 2013). Pada Gambar 6.2 dapat dilihat rata-rata *fitness* terbaik adalah 8124204 pada kombinasi *cr* 0,5 dan *mr* 0,5. Dapat disimpulkan bahwa kombinasi *cr* dan *mr* terbaik adalah 0,5 : 0,5.

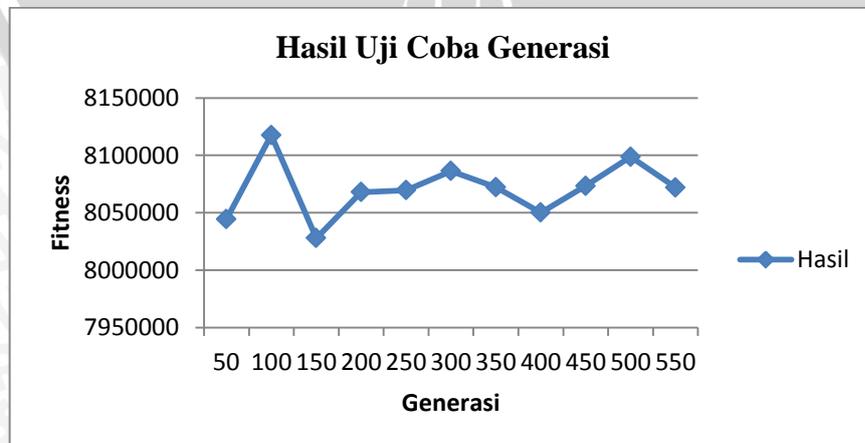
6.2.3 Hasil dan Analisa Uji Coba Banyaknya Generasi

Pengujian ini bertujuan untuk mengetahui jumlah generasi yang optimal untuk menghasilkan rata-rata *fitness* terbaik. Parameter yang akan digunakan antara lain ukuran populasi terbaik sebesar 180 yang didapatkan dari hasil uji coba populasi. Nilai *cr* dan *mr* yang digunakan masing-masing 0,5 yang merupakan hasil terbaik dari uji coba kombinasi *cr* dan *mr*. Banyak generasi yang akan diuji adalah kelipatan 50 mulai dari 50 sampai 550 generasi. Setiap generasi dilakukan pengujian sebanyak 10 kali percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba tersebut akan diperoleh berapa besar generasi yang optimal untuk pemecahan masalah. Hasil uji coba generasi ditampilkan pada Tabel 6.5.

Tabel 6.5 Hasil Percobaan Banyaknya Generasi

Banyak Generasi	Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
	Percobaan generasi ke-										
	1	2	3	4	5	6	7	8	9	10	
50	8049095	7998623	8071369	8054835	8006149	7963060	8070768	8108675	8136293	7986148	8044502
100	8131744	8076788	8063118	8262581	8175062	8082524	8079987	8068089	8108910	8126555	8117536
150	7967838	7958947	8065469	8133771	7982564	8090629	7941508	7988791	8066473	8083159	8027915
200	7982310	8030265	8099496	8047333	8088630	8032437	8064521	8221741	8014429	8098491	8067965
250	8107182	8121120	7981329	8077591	8195572	8095455	7987858	8147494	7999283	7982945	8069583
300	8082475	8160749	8089224	7925748	8213811	8131051	8039920	8035612	8094170	8089424	8086218
350	8055570	8214271	8032453	8092668	8107077	8007006	8080222	7964282	8088628	8080566	8072274
400	8114121	8036212	8087261	7940447	8130644	7947186	7962639	8124228	8112767	8046067	8050157
450	8224909	8006449	8123137	8103678	8026187	8069672	7981107	8100759	8119054	7978843	8073380
500	8077267	8073509	8119628	8088069	8076815	8145874	8221582	8159059	8053828	7971421	8098705
550	8090379	8056384	8049765	8104461	7962245	8070858	8155404	8136033	8217161	7876834	8071952

Berdasarkan hasil uji coba banyaknya generasi pada Tabel 6.5 dapat disimpulkan bahwa pada setiap uji coba dengan jumlah generasi yang berbeda, akan menghasilkan rata-rata nilai *fitness* yang berbeda pula. Dari data tersebut dibuat sebuah grafik untuk melihat perbedaan dari banyaknya generasi terhadap hasil rata-rata nilai *fitness* yang dihasilkan. Grafik dapat dilihat pada Gambar 6.3.



Gambar 6.3 Grafik Hasil Uji Coba Banyaknya Generasi

Dari grafik Gambar 6.3 dapat dilihat bahwa banyaknya generasi yang menghasilkan *fitness* terbaik adalah 100. Nilai *fitness* terendah terdapat pada generasi 150. Secara umum semakin banyak generasi maka akan didapatkan nilai *fitness* yang lebih baik, tetapi terlalu banyak generasi membuat waktu proses semakin panjang. Namun dilihat dari Gambar 6.3, dari hasil pengujian yang dilakukan sulit untuk diambil kesimpulan karena pola grafik cenderung acak. Hal ini biasanya disebabkan algoritma genetika mengalami konvergensi dini, artinya kromosom-kromosom yang ada cenderung bernilai sama sehingga operator reproduksi (*crossover* dan mutasi) tidak mampu menghasilkan kromosom anak yang lebih baik (Mahmudy, 2013).

6.2.4 Hasil Terbaik

Hasil terbaik berdasarkan pengujian yang dilakukan adalah nilai *fitness* dari kombinasi algoritma dengan ukuran populasi 180, kombinasi *cr* dan *mr* masing-masing 0,5, dan besar generasi 100. Hasil terbaik ditunjukkan pada Gambar 6.4.

```

Output x Notifications Search Results Variables Breakpoints
skripsi (debug) x Debugger Console x
debug:
Kromosom terbaik didapatkan pada generasi ke 100
P1: 47.07 1.43 1.43 82.28 117.16 47.07 131.03 2.39 117.16 131.03
Kendala 1: 161.78200000000004
Kendala 2: 92.542
Kendala 3: 477.0
Kendala 4: 2999500.0

Nilai constraint (bernilai 0 jika nilai kendala kurang dari persediaan
c1: 5.532000000000039
c2: 0.0
c3: 0.0
c4: 0.0

Fungsi tujuan untuk laba eceran (laba A): 6799564.1032
Fungsi tujuan untuk laba grosir (laba B): 1089146.6352000001
Fungsi tujuan untuk laba distributor (laba C): 344124.2616
Fungsi tujuan keseluruhan: 8232835.0

Nilai fitness 8177515

```

Gambar 6.4 Hasil terbaik Pengujian *Reciprocal Exchange Mutation*

Dari satu kali hasil pengujian didapatkan nilai *fitness* sebesar 8177515. Laba yang didapatkan ditunjukkan pada fungsi tujuan keseluruhan, yaitu sebesar Rp 8.232.835,-.

6.3 Hasil dan Analisis Pengujian Parameter Algoritma Genetika dengan *Random Mutation*

Proses uji coba ini dilakukan untuk mengetahui berapa *fitness* tertinggi dari kromosom terbaik pada setiap percobaan parameter algoritma genetika pada percobaan dengan metode mutasi *random mutation*. Hasil pengujian ini kemudian dianalisis untuk diambil kesimpulan dari penelitian.

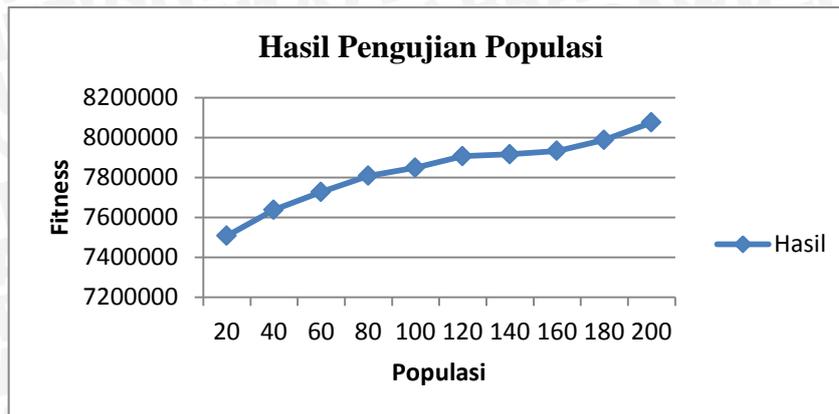
6.3.1 Hasil dan Analisis Uji Coba Ukuran Populasi

Jumlah generasi yang dipakai adalah 100 dengan ukuran populasi kelipatan 20 dari 20 hingga 200 populasi. Nilai *cr* dan *mr* yang digunakan masing-masing sebesar 0,5. Setiap generasi dilakukan 10 kali percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari hasil uji coba akan diperoleh berapa ukuran populasi optimum untuk menyelesaikan masalah pada penelitian. Hasil dari uji coba populasi dapat dilihat pada Tabel 6.6.

Tabel 6.6 Hasil Uji Coba Populasi

Banyak Populasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan populasi ke-										
	1	2	3	4	5	6	7	8	9	10	
20	7450974	7407457	7470667	6812736	7603771	7724148	7695631	7482728	7452759	7972807	7507368
40	7433608	7538208	8169636	7710255	7561259	7727933	7806948	7624984	7274996	7532522	7638035
60	7819582	7682735	7598552	8067498	7987451	7518752	7311079	7650271	7670262	7964418	7727060
80	7903388	7796399	7835173	7688818	8067897	7750805	7895382	7759586	7720721	7664960	7808313
100	8005620	7839149	7582316	8174329	7890509	7793356	7597101	8065850	7768064	7771829	7848812
120	7723683	8259502	7789719	7796504	8185859	7964420	8101575	7940718	7553760	7747299	7906304
140	8071453	7728323	7787750	8181818	7970620	7760140	7986866	7734471	7942592	7995833	7915987
160	8139018	7898071	8089769	7785914	8117538	7868923	7818955	7968227	7700278	7944135	7933083
180	7967629	8051162	7895896	8034544	8076184	7721267	8003281	7891496	8247453	7991230	7988014
200	8008668	8140328	7980704	8060668	8016502	8087749	8068379	8025677	8151962	8210152	8075079

Dari data pada Tabel 6.6 dibuat sebuah grafik untuk melihat perbedaan nilai rata-rata *fitness* terhadap masing-masing ukuran populasi yang dicobakan. Grafik uji coba populasi ini dapat dilihat pada Gambar 6.5.



Gambar 6.5 Grafik Hasil Uji Coba Ukuran Populasi

Dari grafik Gambar 6.5 dapat dilihat bahwa ukuran populasi berpengaruh terhadap hasil algoritma genetika yang terlihat pada nilai rata-rata *fitness*-nya. Nilai *fitness* terendah terdapat pada populasi 20 dan ukuran populasi yang optimal sebesar 200 dengan nilai rata-rata *fitness* 8075079 karena setelah jumlah populasi 200 sulit didapatkan nilai *fitness* yang lebih baik. Pola konvergensi seperti ini juga didapatkan oleh Pitaloka, Mahmudy, dan Sutrisno (2014) yang menerapkan algoritma genetika untuk permasalahan *vehicle routing problem with time windows* (VRPTW) untuk penentuan rute distribusi barang serta Putri, Mahmudy, dan Ratnawati (2014) yang menerapkan algoritma genetika untuk VRPTW pada kasus optimasi distribusi beras bersubsidi.

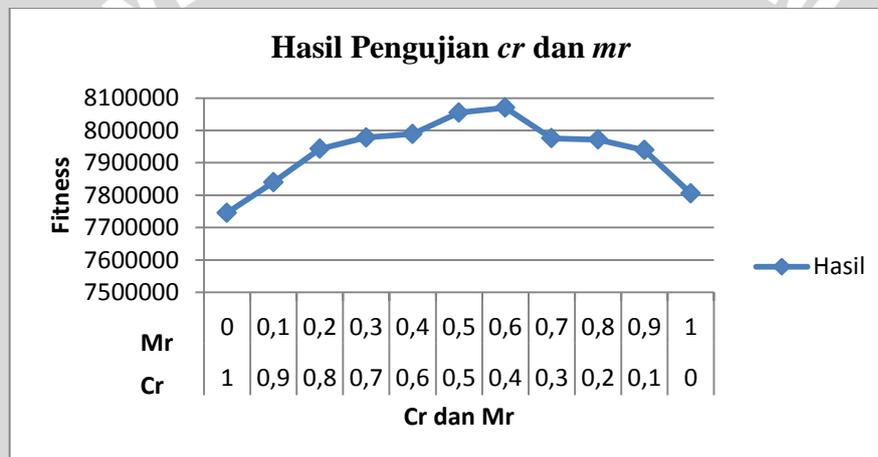
6.3.2 Hasil dan Analisis Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Ukuran populasi yang digunakan adalah yang menghasilkan nilai *fitness* terbaik pada uji coba ukuran populasi yaitu 200. Banyak generasi yang digunakan sebesar 100. Nilai *cr* dan *mr* yang digunakan dalam pengujian berkisar antara 0 sampai dengan 1. Setiap generasi dilakukan percobaan sebanyak 10 kali dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba tersebut akan didapatkan nilai *cr* dan *mr* yang paling optimal untuk menyelesaikan masalah. Hasil dari uji coba kombinasi *cr* dan *mr* dapat dilihat pada Tabel 6.7.

Tabel 6.7 Hasil Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Kombinasi		Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
		Percobaan kombinasi <i>cr</i> dan <i>mr</i> ke-										
<i>cr</i>	<i>mr</i>	1	2	3	4	5	6	7	8	9	10	
1	0	7913568	7698269	7664000	7873587	7764620	7773786	7558113	7806384	7540532	7863197	7745606
0,9	0,1	7881993	7744285	7793988	7927487	7787456	7910094	7866715	7817831	7889763	7780764	7840038
0,8	0,2	7908044	7886883	7794475	7961001	7893419	8138294	7947215	8059391	7867257	7973694	7942967
0,7	0,3	7947311	8135417	8036008	8210010	7776691	7969604	8014595	7875915	7814145	8000746	7978044
0,6	0,4	8280579	8288841	7775084	7993494	7904152	8030166	7901798	7875622	7904409	7935539	7988968
0,5	0,5	7947517	8140328	7940648	7961506	8016502	8087749	8068379	8025677	8151962	8210152	8055042
0,4	0,6	7779858	7692677	7667447	8311027	7977738	8220911	8173173	8295146	8305454	8276520	8069995
0,3	0,7	8220422	7787692	8047410	8048278	8004477	7901490	7883656	8068558	7838427	7957485	7975790
0,2	0,8	8069353	8203325	7677328	8125122	8038445	7976772	7914432	7982717	7865000	7864383	7971688
0,1	0,9	7595912	8011092	7642793	8030142	7992735	7861557	7883137	8206841	8070926	8096197	7939133
0	1	7712295	7666844	7751019	7799316	7905930	8102343	7961085	7776825	7667061	7712687	7805541

Dari data pada Tabel 6.7 dibuat sebuah grafik untuk melihat perbedaan rata-rata nilai *fitness* dari hasil pengujian kombinasi *cr* dan *mr*. Grafik hasil uji coba kombinasi *cr* dan *mr* dapat dilihat pada Gambar 6.6.



Gambar 6.6 Hasil Uji Coba Kombinasi *Crossover rate* dan *Mutation rate*

Pada Gambar 6.6 dapat dilihat rata-rata *fitness* terbaik adalah 8069995 pada kombinasi *cr* 0,4 dan *mr* 0,6. Dapat disimpulkan bahwa kombinasi *cr* dan *mr* terbaik adalah 0,4 : 0,6.

6.3.3 Hasil dan Analisa Uji Coba Banyaknya Generasi

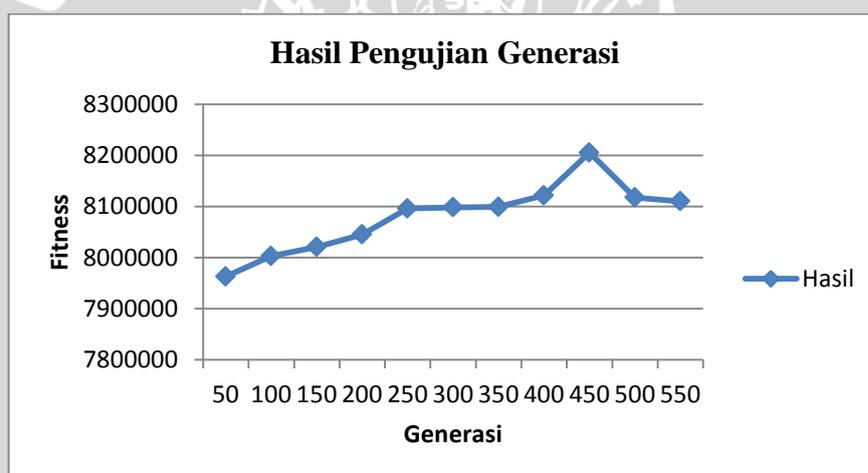
Parameter yang akan digunakan antara lain jumlah populasi terbaik sebesar 200 yang didapatkan dari hasil uji coba populasi. Nilai *cr* dan *mr* yang digunakan masing-masing 0,4 dan 0,6 yang merupakan hasil terbaik dari uji coba kombinasi *cr* dan *mr*. Banyak generasi yang akan diuji adalah kelipatan 50 mulai dari 50 sampai 550 generasi. Setiap generasi dilakukan pengujian sebanyak 10 kali percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba tersebut akan

diperoleh berapa besar generasi yang optimal untuk pemecahan masalah. Hasil uji coba generasi ditampilkan pada Tabel 6.8.

Tabel 6.8 Hasil Percobaan Banyaknya Generasi

Banyak Generasi	Nilai <i>Fitness</i>										Rata-rata <i>fitness</i>
	Percobaan generasi ke-										
	1	2	3	4	5	6	7	8	9	10	
50	7912918	8007894	7891829	7919205	8024869	7930394	8106249	7895887	8099072	7840711	7962903
100	7731719	7692677	7667447	8079924	7935136	8220911	8173173	8295146	8119409	8114839	8003038
150	8032707	7922938	7937059	7911167	8086754	8170951	7851878	8287072	7810601	8196275	8020740
200	7930762	7994694	7890073	8022401	8029229	8067446	8178551	8078351	8435296	7825096	8045190
250	8045285	8561032	8190534	8094275	7805145	8176770	7906211	7997261	8128372	8055027	8095991
300	7957053	8115804	8125638	8192286	8241009	8020438	8257820	8176476	7865393	8029420	8098134
350	8190760	7982633	8130744	8144381	8132687	8151864	8140201	7947641	7866358	8302722	8098999
400	7971366	8135005	8247676	8352450	8224356	7977068	8179544	7898446	8093348	8138056	8121732
450	8057443	8763506	8416380	7958508	8195601	8215183	7978460	8126859	8290878	8049391	8205221
500	8490921	8000167	7999157	8295909	7831237	8076191	8168222	8107650	8177251	8029335	8117604
550	8089827	8135512	8004462	7637169	8281504	8424920	8162563	8055519	8175054	8134478	8110101

Berdasarkan hasil uji coba banyaknya generasi pada Tabel 6.8 dibuat sebuah grafik untuk melihat perbedaan dari banyaknya generasi terhadap hasil rata-rata nilai *fitness* yang dihasilkan. Grafik dapat dilihat pada Gambar 6.7.



Gambar 6.7 Grafik Hasil Uji Coba Banyaknya Generasi

Dari grafik Gambar 6.7 dapat dilihat bahwa banyaknya generasi yang menghasilkan *fitness* terbaik adalah 450. Nilai *fitness* terendah terdapat pada generasi 50. Dari Gambar 6.7 tersebut, dapat disimpulkan bahwa jumlah generasi yang paling optimal adalah dengan jumlah generasi 450, karena setelah generasi 450 sulit didapatkan nilai *fitness* yang lebih baik. Pola seperti ini juga didapatkan oleh Sari, Mahmudy, dan Dewi (2014) yang menerapkan algoritma genetika untuk optimasi asupan gizi pada ibu hamil. Dari hasil penelitian dapat disimpulkan

bahwa metode *random mutation* lebih cocok digunakan untuk menyelesaikan kasus optimasi laba produksi jilbab.

6.3.4 Hasil Terbaik

Hasil terbaik berdasarkan pengujian yang dilakukan adalah nilai *fitness* dari kombinasi algoritma dengan ukuran populasi 200, kombinasi cr dan mr masing-masing 0,4 dan 0,6, dan besar generasi 450. Hasil terbaik ditunjukkan pada Gambar 6.8.

```

debug:
Kromosom terbaik didapatkan pada generasi ke 450
P1: 14.657855449098408 0.03273171632120819 0.0 99.1800378171595 103.61854324278113 139.92888795102033
Kendala 1: 156.24977252014833
Kendala 2: 92.41534372574472
Kendala 3: 466.065806241906
Kendala 4: 2951722.3421688

Nilai constraint (bernilai 0 jika nilai kendala kurang dari persediaan
c1: 0.0
c2: 0.0
c3: 0.0
c4: 0.0

Fungsi tujuan untuk laba eceran (laba A): 6645465.786210575
Fungsi tujuan untuk laba grosir (laba B): 1067655.0784282037
Fungsi tujuan untuk laba distributor (laba C): 337492.5668926961
Fungsi tujuan keseluruhan: 8050613.431531476

Nilai fitness 8050613
  
```

Gambar 6.8 Hasil Terbaik Pengujian *Random Mutation*

Dari satu kali hasil pengujian didapatkan nilai *fitness* sebesar 8050613. Laba yang didapatkan ditunjukkan pada fungsi tujuan keseluruhan, yaitu sebesar Rp 8.050.613,00. *Fitness* dan fungsi tujuan dapat bernilai sama apabila keseluruhan kendala bernilai 0. Solusi terbaik dihasilkan apabila nilai penalti keseluruhan bernilai 0, yang berarti tidak ada kekurangan bahan baku selama proses produksi.

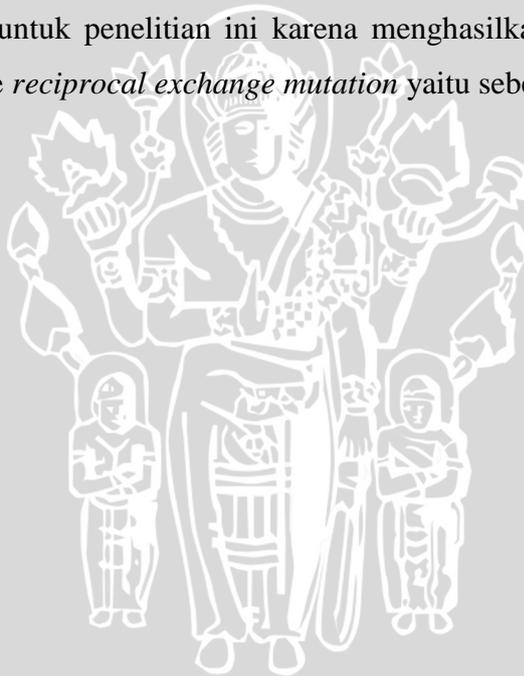
6.4 Perbandingan Hasil *Reciprocal Exchange Mutation* dan *Random Mutation*

Dari pengujian yang telah dilakukan, akan dilakukan sepuluh kali percobaan untuk membandingkan hasil *fitness* dari kedua metode mutasi yang digunakan yaitu *reciprocal exchange* dan *random mutation*. Percobaan dilakukan menggunakan parameter terbaik yang didapatkan dari hasil pengujian parameter algoritma. Hasil perbandingan dapat dilihat pada Tabel 6.9.

Tabel 6.9 Perbandingan Hasil *Reciprocal Exchange Mutation* dan *Random Mutation*

Percobaan ke-	Metode Mutasi	
	<i>Reciprocal Exchange Mutation</i>	<i>Random Mutation</i>
1	8046389	8033616
2	8177515	7983519
3	7977006	8257103
4	8034848	8265348
5	8063982	8329529
6	7977270	8392368
7	7957915	8317883
8	7909157	8078393
9	7984811	7939770
10	8248846	8050613
Rata-rata fitness	8037773	8164814

Dari hasil percobaan dapat disimpulkan bahwa metode *random mutation* lebih baik digunakan untuk penelitian ini karena menghasilkan rata-rata *fitness* lebih besar dari metode *reciprocal exchange mutation* yaitu sebesar 8164814.





BAB VII PENUTUP

7.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian optimasi laba dalam produksi jilbab *home industry* menggunakan algoritma genetika ini adalah sebagai berikut:

1. Representasi kromosom dengan cara *real coded* dengan proses reproduksi *one-cut-point crossover*, metode mutasi *reciprocal exchange mutation* dan *random mutation* yang digunakan dalam penelitian ini mampu menyelesaikan permasalahan optimasi laba dalam produksi jilbab *home industry*. Algoritma genetika mampu menentukan kombinasi jilbab yang akan diproduksi sesuai dengan modal dan persediaan yang mampu menghasilkan laba maksimum.
2. Dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa nilai parameter algoritma genetika (ukuran populasi, generasi maksimum, *crossover rate*, dan *mutation rate*) memiliki pengaruh terhadap hasil optimasi. Nilai parameter yang kecil menyebabkan area pencarian algoritma genetika semakin sempit. Tetapi, ukuran parameter yang terlalu besar menyebabkan waktu komputasi semakin lama dan tidak selalu menghasilkan solusi yang lebih baik.
3. Dari hasil uji coba, untuk metode *reciprocal exchange mutation* didapatkan kombinasi paling optimum yaitu pada ukuran populasi 180, kombinasi *cr* dan *mr* masing-masing 0,5, dan besar generasi 100 yang menghasilkan *fitness* sebesar 8037773.
4. Untuk hasil uji coba dengan metode *random mutation* didapatkan kombinasi paling optimum pada ukuran populasi 200, kombinasi *cr* dan *mr* masing-masing 0,4 dan 0,6, dan besar generasi 450. *Fitness* yang dihasilkan sebesar 8164814.
5. Berdasarkan hasil uji coba didapatkan kesimpulan bahwa metode mutasi *reciprocal exchange mutation* dinilai kurang sesuai untuk penelitian ini karena hasil grafik penelitian pada parameter algoritma genetika berupa generasi cenderung acak. Metode yang lebih sesuai adalah *random mutation*

dikarenakan hasil grafik lebih teratur dan menghasilkan rata-rata fitness yang lebih besar.

7.2 Saran

Saran yang dapat diberikan untuk kemungkinan pengembangan selanjutnya dari penelitian adalah dengan melakukan penggabungan (*hybrid*) algoritma genetika dengan algoritma lainnya untuk pengembangan solusi terbaik yang dihasilkan. Penggabungan (*hybridisation*) dengan algoritma lain mampu meningkatkan efisiensi dan akurasi dalam pencarian solusi optimum (Mahmudy, Marian, dan Luong, 2014). Data yang digunakan juga dapat ditambah atau diubah agar tidak terbatas pada data produksi jilbab *home industry*. Selain itu, dapat dilakukan percobaan metode *crossover*, mutasi, dan seleksi yang lain untuk mendapatkan solusi yang lebih bervariasi.



DAFTAR PUSTAKA

- Fardiana, Elvia. 2012. Maksimalisasi Keuntungan pada Toko Kue Martabak Doni dengan Metode Simpleks. UG Jurnal Vol. 6 No. 09.
- Fitriyani. 2014. Pengaruh Manajemen Laba Terhadap Biaya Modal Ekuitas (Pada Perusahaan Manufaktur yang Terdaftar di Bursa Efek Indonesia). Bandung: Universitas Widyatama.
- Mahmudy, WF. 2013. Algoritma Evolusi. Program Teknologi Informasi dan Ilmu Komputer. Malang: Universitas Brawijaya.
- Mahmudy, WF & Rahman, MA. 2011. Optimasi Fungsi Multi-Obyektif Berkendala Menggunakan Algoritma Genetika Adaptif dengan Pengkodean Real. *Kursor*, vol. 6, no. 1, pp. 19-26.
- Mahmudy, WF, Marian, RM & Luong, LHS. 2013. *Modeling and Optimization of Part Type Selection and Loading Problems in Flexible Manufacturing System Using Real Coded Genetic Algorithms*. *International Journal of Electrical, Electronic Science and Engineering*, vol. 7, no. 4, pp. 181-190.
- Mahmudy, WF, Marian, RM & Luong, LHS. 2014. *Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System*. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 8, no. 1, pp. 80-93.
- Pitaloka, DA, Mahmudy, WF & Sutrisno. 2014. Penerapan *Hybrid* Algoritma Genetika untuk Permasalahan *Vehicle Routing Problem with Time Windows (VRPTW)*. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, vol. 4, no. 11.
- Pratiwi, MI, Mahmudy, WF & Dewi, C. 2014. Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, vol. 4, no. 6.
- Putri, FB, Mahmudy, WF & Ratnawati, DE. 2014. Penerapan Algoritma Genetika untuk *Vehicle Routing Problem with Time Windows (VRPTW)* Pada Kasus Optimasi Distribusi Beras Bersubsidi. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, vol. 5, no. 1.

- Sari, AP, Mahmudy, WF & Dewi, C. 2014. Optimasi Asupan Gizi pada Ibu Hamil dengan Menggunakan Algoritma Genetika. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 4, no. 5.
- Selawati, Suzan Dwi. 2007. Home Industri dan Koperasi: Mutualisme Dua Kegiatan Ekonomi Sebagai Langkah Awal untuk Mengentaskan Kemiskinan. URL: <https://missanetalentist.wordpress.com/2007/08/18/home-industri-dan-koperasi-mutualisme-dua-kegiatan-ekonomi-sebagai-langkah-awal-untuk-mengentaskan-kemiskinan/>. Diakses tanggal 06 April 2014, pukul 18.24.
- Wirdasari, Dian. 2009. Metode Simpleks dalam Program Linier. Jurnal SAINTKOM Vol. 6, No. 1 Januari 2009.
- Wild, John dan Subramanyam, KR. 2012. Analisis Laporan Keuangan. Edisi Sepuluh, Buku Kesatu. Alih Bahasa : Dewi Yanti. Jakarta: Salemba Empat.

