

repository.ub.ac.id

**IMPLEMENTASI ALGORITMA EVOLUTION STRATEGIES
UNTUK OPTIMASI LABA BERDASARKAN JUMLAH PRODUKSI
PADA *HOME INDUSTRY* TEKSTIL**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Mabafasa Al Khuluqiy
NIM: 125150200111109



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

IMPLEMENTASI ALGORITMA EVOLUTION STRATEGIES UNTUK OPTIMASI LABA
BERDASARKAN JUMLAH PRODUKSI PADA *HOME INDUSTRY* TEKSTIL

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Mabafasa Al Khuluqiy
NIM: 125150200111109

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

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

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah proposal 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 disitasi dalam naskah ini dan disebutkan dalam 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, 29 Juli 2016

Mabafasa Al Khulugiy

NIM: 12515020011109



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala, yang atas rahmat-Nya maka penulis dapat menyelesaikan penyusunan skripsi yang berjudul "Implementasi Algoritma *Evolution Strategies* Untuk Optimasi Laba Berdasarkan Jumlah Produksi Pada *Home Industry* Tekstil". Skripsi ini diajukan sebagai salah satu syarat untuk mendapatkan gelar sarjana S-1 di Fakultas 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 Subhanahu Wa Ta'ala sehingga kendala-kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T, Ph.D sebagai dosen pembimbing yang dengan sabar memberikan saran dan masukan perbaikan dalam proses penyusunan skripsi ini;
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, M.T, Bapak Heru Nurwarsito, Ir., M.Kom, Bapak Marji, Drs., M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan I, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer Universitas Brawijaya;
3. Bapak Issa Arwani, S.Kom, M.Sc dan Ibu Fitri Utamingrum, Dr. Eng., S.T, M.T selaku Kepala dan Sekretaris Program Studi Informatika/Illmu Komputer serta segenap Bapak/Ibu Dosen, Staff Administrasi Fakultas Ilmu Komputer Universitas Brawijaya;
4. Kedua orang tua penulis, Ibu Ummy Erliyah dan Bapak Husnul Quluk Maadin Siraj (Alm), dan keluarga yang tidak pernah bosan dan lupa untuk memberikan doa dan dorongan semangat hingga terselesaikannya skripsi ini;
5. Ibu Alfi selaku pemilik home industry tekstil "NAFA" yang telah bersedia menjadi narasumber dalam penelitian ini;
6. Seluruh teman-teman mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya khususnya angkatan 2012;
7. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat penulis sebutkan satu per satu.

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

Akhir kata penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan dan jauh dari kesempurnaan. Karena itu, penulis memohon saran dan kritik yang sifatnya membangun untuk pengembangan penelitian selanjutnya.

Malang, 29 Juli 2016

Penulis
mabafasa@gmail.com



ABSTRAK

Home Industry termasuk salah satu bentuk strategi pemerintah sebagai upaya peningkatan pendapatan masyarakat dan mengurangi jumlah pengangguran. Pada Home Industry, optimasi laba menjadi salah satu tujuan utama dari proses produksi. Untuk pencapaiannya sendiri, dibutuhkan perhitungan dan analisis yang tepat agar mendukung tercapainya laba yang maksimal. Laba yang maksimal bisa dicapai dengan perencanaan produksi yang tepat. Perencanaan produksi menggunakan perhitungan manual bisa diterapkan untuk skala kecil, tetapi jika bahan baku untuk produksi memiliki skala yang cukup besar dibutuhkan metode atau algoritma tertentu agar mendapatkan perencanaan produksi yang akurat dan dalam waktu yang relatif singkat. Alternatif solusi yang diusulkan untuk menyelesaikan permasalahan tersebut adalah dengan menggunakan Algoritma Evolution Strategies (ES). Pada penelitian ini, sampel data yang digunakan berupa 20 macam produk tekstil beserta bahan baku yang digunakan untuk tiap produk, stok bahan baku bulanan, dan laba penjualan tiap produk. Produk tersebut kemudian direpresentasikan kedalam kromosom yang merupakan kombinasi kuantitas produksi setiap produk. Tipe ES yang digunakan adalah $(\mu/r + \lambda)$ sehingga metode rekombinasi dan mutasi digunakan pada proses offspring, dan seleksi melibatkan induk dan offspring dengan menggunakan metode Elitism selection. Berdasarkan pengujian parameter yang telah dilakukan, diperoleh nilai parameter-parameter terbaik yang menghasilkan solusi mendekati optimal meliputi kisaran produksi maksimum sejumlah 30 per produk, lambda sebesar 9, mu sebesar 80, dan jumlah generasi sebesar 700. Dari satu kali hasil percobaan menggunakan parameter dengan nilai terbaik didapatkan nilai fitness sebesar 5543000. Hasil pengujian menunjukkan bahwa optimasi laba menggunakan algoritma *Evolution Strategies* menghasilkan laba yang lebih maksimal.

Kata kunci: *home industry*, laba, optimasi, produksi, *evolution strategies*

ABSTRACT

Home Industry including one of the government's strategy as an effort to increase household incomes and reducing unemployment. At Home Industry, profit optimization became one of the main goals of the production process. For the achievement itself, required precise calculations and analysis in order to support the achievement of the optimum profit. Profit optimum production can be achieved with proper planning. Production planning using manual calculation can be applied to small-scale, but if the raw material for the production have a large enough scale that it takes a certain algorithm method in order to obtain accurate production planning and in a relatively short time. Alternative solutions proposed to solve these problems is to use Algorithm Evolution Strategies (ES). In this study, the sample data used in the form of 20 kinds of textile products and the amount of raw materials used, the monthly stocks of raw materials, and sales profit per product. The products were then represented into the chromosome that is a combination of production quantity of each product. ES mode used is $(\mu / r + \lambda)$ so that recombination and mutation methods used in the offspring, and selection involves the parent and offspring using methods of elitism selection. Based on testing, the best parameter values obtained yield near optimal solution includes a production range to a maximum of 30 lambda at 9, miu is 80, and the number of generations of 700. From the time the results of experiments using the parameter with the best value, obtained fitness value of 5543000. The test result showed that profit optimization using Evolution Strategies algorithm generate maximum profits.

Keywords: home industry, profit, optimation, production, evolution strategies

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
Daftar Persamaan	xiii
DAFTAR SOURCE CODE	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan.....	2
1.4 Manfaat penelitian.....	3
1.5 Batasan masalah.....	3
1.6 Sistematika penulisan.....	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Kajian pustaka.....	5
2.2 Algoritma evolusi.....	9
2.2.1 Tipe algoritma evolusi.....	9
2.3 Evolution strategies	10
2.3.1 Istilah dalam Evolution Strategies.....	10
2.3.2 Struktur dasar evolution strategies	11
2.3.3 Representasi kromosom	13
2.3.4 Inisialisasi	13
2.3.5 Reproduksi	13
2.3.6 Seleksi.....	15
2.3.7 Kondisi berhenti	16

2.4 Home industry	17
2.4.1 Jenis home industry	17
Sumber: Diadaptasi dari Abrianto (2012)	18
2.5 Laba	18
BAB 3 METODOLOGI PENELITIAN	19
3.1 Tahapan penelitian	19
3.2 Studi literatur.....	20
3.3 Analisis kebutuhan	20
3.4 Pengumpulan data	21
3.5 Perancangan	21
3.5.1 Model perancangan sistem.....	21
3.6 Implementasi.....	22
3.6.1 Algoritma yang Digunakan.....	22
3.7 Pengujian dan pembahasan	22
3.7.1 Uji coba kisaran produksi maksimum	22
3.7.2 Uji coba nilai λ (lambda)	23
3.7.3 Uji coba ukuran populasi (miu).....	24
3.7.4 Uji coba jumlah generasi.....	24
BAB 4 PERANCANGAN.....	26
4.1 Formulasi permasalahan	26
4.1.1 Asumsi permasalahan	31
4.2 Siklus algoritma <i>evolution strategies</i>	32
4.3 Penyelesaian masalah menggunakan Algoritma <i>Evolution strategies</i>	33
4.3.1 Representasi kromosom dan perhitungan <i>fitness</i>	33
4.3.2 Inialisasi populasi awal	36
4.3.3 Reproduksi	36
4.3.4 Seleksi.....	41
4.4 Perancangan user interface.....	43
BAB 5 IMPLEMENTASI KELAS	46
5.1 Struktur Class.....	46

5.2 Implementasi algoritma	47
5.2.1 Inisialisasi	47
5.2.2 Fungsi tujuan.....	48
5.2.3 Kendala.....	49
5.2.4 Fitness	50
5.2.5 Rekombinasi.....	50
5.2.6 Mutasi	51
5.2.7 Seleksi.....	52
BAB 6 PENGUJIAN DAN PEMBAHASAN.....	54
6.1 Hasil dan Analisis Pengujian	54
6.1.1 Hasil dan analisis uji coba kisaran produksi maksimum	54
6.1.2 Hasil dan analisis uji coba nilai λ (lambda).....	56
6.1.3 Hasil dan analisis uji coba ukuran populasi (miu).....	58
6.1.4 Hasil dan analisis uji coba jumlah generasi.....	60
6.2 Solusi menggunakan parameter terbaik	61
6.2.1 Perbandingan Solusi.....	63
BAB 7 PENUTUP	65
7.1 Kesimpulan	65
7.2 Saran.....	65
DAFTAR PUSTAKA.....	67

DAFTAR TABEL

Tabel 2.1 Kajian pustaka	7
Tabel 2.2 Istilah-istilah pada algoritma <i>Evolution Strategies</i>	10
Tabel 2.3 Tipe Evolution Strategies.....	12
Tabel 2.4 Contoh populasi awal	13
Tabel 2.5 Contoh hasil rekombinasi.....	14
Tabel 2.6 Contoh hasil mutasi.....	15
Tabel 2.7 Contoh hasil seleksi	16
Tabel 2.8 Jenis Home Industry	18
Tabel 3.1 Uji coba kisaran produksi maksimum	23
Tabel 3.2 Uji coba nilai λ (lambda)	23
Tabel 3.3 Uji coba ukuran populasi (miu)	24
Tabel 3.4 Uji coba jumlah generasi	24
Tabel 4.1 Bahan baku perbulan	26
Tabel 4.2 Laba per produk	26
Tabel 4.3 Bahan baku produk	27
Tabel 4.4 Contoh jumlah dan laba	28
Tabel 4.5 Contoh hasil nilai fungsi tujuan.....	29
Tabel 4.6 Representasi kromosom	33
Tabel 4.7 Contoh hasil perhitungan fitness	35
Tabel 4.8 Populasi awal.....	36
Tabel 4.9 Hasil rekombinasi	37
Tabel 4.10 Nilai $N(0,1)$ pada individu C5.....	39
Tabel 4.11 Hasil mutasi	40
Tabel 4.12 Hasil pengurutan	41
Tabel 4.13 Individu hasil seleksi.....	42
Tabel 5.1 Keterangan Struktur Class	47
Tabel 6.1 Hasil pengujian kisaran produksi maksimum.....	55
Tabel 6.2 Hasil pengujian nilai λ (lambda).....	57
Tabel 6.3 Hasil pengujian ukuran populasi (miu).....	59
Tabel 6.4 Hasil pengujian jumlah generasi	60
Tabel 6.5 Hasil parameter dengan nilai terbaik.....	62
Tabel 6.6 Hasil Percobaan dengan parameter terbaik	63
Tabel 6.7 Kuantitas produksi bulan Juni 2016	63

DAFTAR GAMBAR

Gambar 2.1 Proses-proses dalam Algoritma Evolusi	9
Gambar 2.2 Siklus algoritma <i>Evolution Strategies</i>	12
Gambar 2.3 Pseudo-code elitism selection	16
Gambar 3.1 Diagram Blok Metodologi Penelitian	20
Gambar 3.2 Diagram blok sistem	21
Gambar 4.1 Siklus algoritma <i>Evolution Strategies</i>	32
Gambar 4.2 Struktur kromosom	33
Gambar 4.3 Proses Rekombinasi	37
Gambar 4.4 Proses mutasi	39
Gambar 4.5 Proses seleksi	41
Gambar 4.6 Rancangan halaman bahan baku	43
Gambar 4.7 Rancangan halaman perubahan data	44
Gambar 4.8 Halaman perhitungan <i>evolution strategies</i>	45
Gambar 5.1 Struktur Class	46
Gambar 6.1 Grafik hasil pengujian kisaran produksi maksimum	56
Gambar 6.2 Hasil pengujian nilai λ (lambda)	58
Gambar 6.3 Hasil pengujian ukuran populasi (miu)	59
Gambar 6.4 Hasil pengujian jumlah generasi	61
Gambar 6.5 Hasil solusi menggunakan parameter dengan nilai terbaik	62

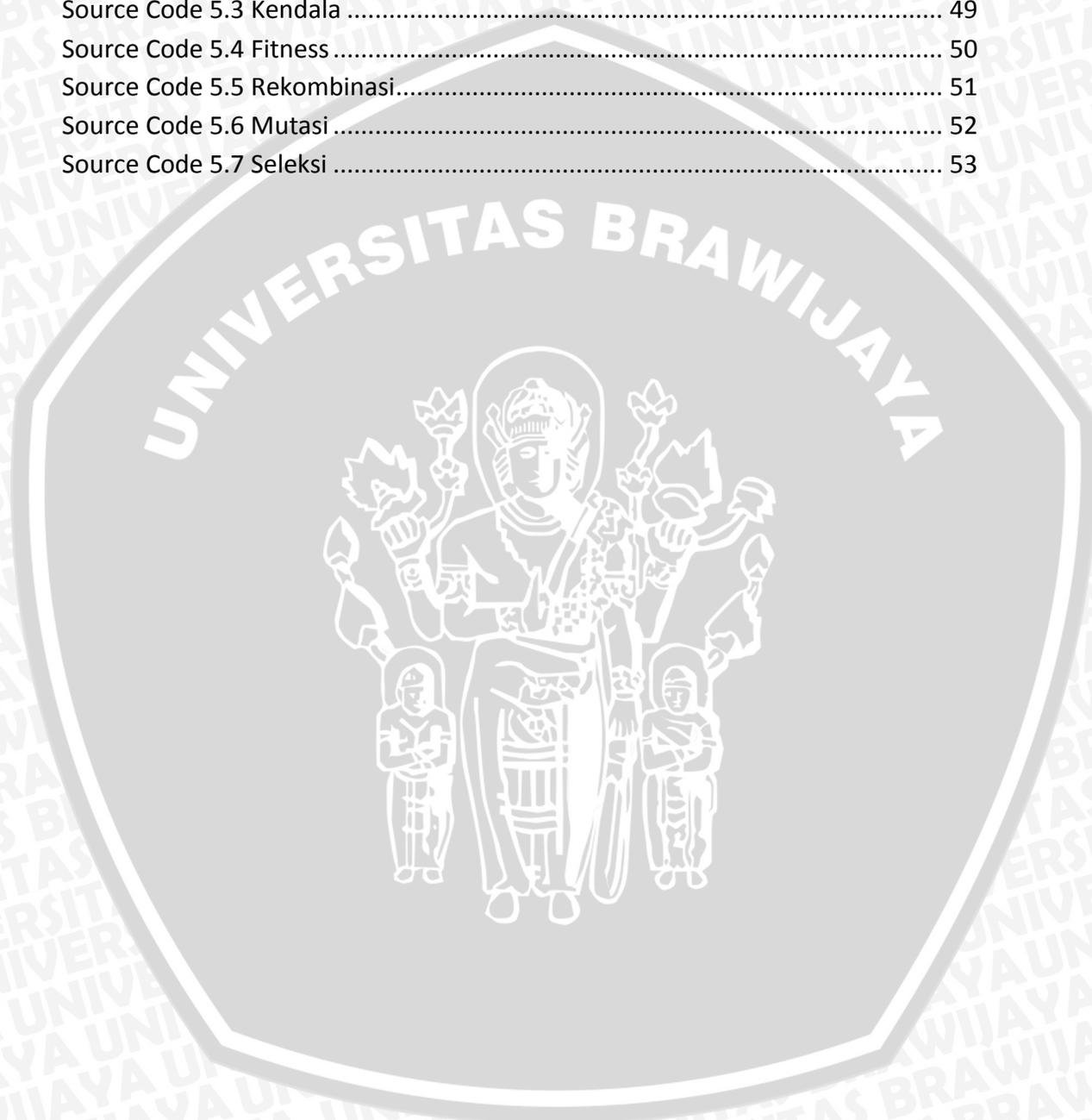
DAFTAR PERSAMAAN

(2.1) Contoh Fungsi Tujuan	13
(2.2) Rumus menentukan x' pada metode Mutasi	14
(2.3) Rumus menentukan $N(1,0)$	14
(4.1) Fungsi tujuan	28
(4.2) Kendala 1	30
(4.3) Kendala 2	30
(4.4) Kendala 3	30
(4.5) Kendala 4	31
(4.6) Kendala 5	31
(4.7) Kendala 6	31
(4.8) Kendala 7	31
(4.9) Rumus Fitness.....	34
(4.10) Rumus menentukan σ	38



DAFTAR SOURCE CODE

Source Code 5.1 Inisialisasi	48
Source Code 5.2 Fungsi Kendala	48
Source Code 5.3 Kendala	49
Source Code 5.4 Fitness	50
Source Code 5.5 Rekombinasi.....	51
Source Code 5.6 Mutasi	52
Source Code 5.7 Seleksi	53



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pertumbuhan ekonomi merupakan salah satu indikator penting dalam menilai kinerja suatu perekonomian negara. Pertumbuhan ekonomi sebuah Negara dapat dilihat dari beberapa indikator perekonomian. Salah satu diantaranya adalah tingkat pengangguran. Berdasarkan tingkat pengangguran dapat dilihat kondisi suatu Negara, apakah perekonomiannya berkembang atau lambat dan/atau bahkan mengalami kemunduran (Amri Amir dalam Alghofari & Arif, 2010). Menurut Berita Resmi Statistik BPS (2015), jumlah penganggur di Negara Indonesia pada bulan Agustus 2015 mencapai 7,6 juta orang dengan Tingkat Pengangguran Terbuka (TPT) mengalami peningkatan dari 5.81 persen pada Februari 2015 menjadi 6.18 persen pada Agustus 2015. Berdasarkan data tersebut, maka kondisi perekonomian Indonesia belum mengalami perkembangan. Sehingga dibutuhkan solusi berupa lapangan pekerjaan baru maupun pembinaan wirausaha kepada masyarakat agar dapat meminimalkan jumlah pengangguran sehingga kondisi perekonomian di Indonesia bisa berkembang.

Home Industry merupakan suatu industri/perusahaan kecil yang memusatkan kegiatan di rumah. *Home Industry* termasuk salah satu bentuk strategi pemerintah sebagai upaya peningkatan pendapatan masyarakat dan mengurangi jumlah pengangguran. *Home Industry* telah diakui sangat strategis dan penting terhadap peningkatan pendapatan masyarakat (Sika, *et al.*, 2012).

Pada perusahaan yang bergerak pada bidang industry seperti *Home Industry*, optimasi laba menjadi tujuan utama dari tiap proses produksi. Untuk pencapaiannya sendiri, dibutuhkan perhitungan dan analisis yang tepat agar mendukung tercapainya laba maksimum (Samaher & Mahmudy, 2015). Laba maksimum bisa dicapai dengan perencanaan produksi yang tepat. Perencanaan produksi menggunakan perhitungan manual bisa diterapkan untuk skala kecil, tetapi jika bahan baku untuk produksi memiliki skala yang cukup besar dibutuhkan metode atau algoritma tertentu agar mendapatkan perencanaan produksi yang akurat dan dalam waktu yang relatif singkat. Alternatif solusi yang bisa digunakan untuk menyelesaikan permasalahan tersebut adalah dengan menggunakan Algoritma Evolusi.

Algoritma Evolusi (*Evolutionary Algorithms*, EAs) merupakan sub-set dari komputasi evolusi (*evolutionary computation*, EC) yang merupakan bentuk generik dari algoritma optimasi meta-heuristic berbasis populasi (Mahmudy, 2013). Algoritma Evolusi mampu menyelesaikan beberapa masalah optimasi seperti pembuatan jadwal kuliah, pemilihan rute terpendek, penentuan komposisi makanan, dan minimalisasi ongkos transportasi. Beberapa tipe EAs yang telah berkembang antara lain Algoritma Genetika (*Genetics Algorithm*),

Evolution strategies (ES), Genetic Programming (GP), dan Evolutionary Programming (EP).

Penggunaan algoritma evolusi untuk penyelesaian optimasi produksi telah dibahas dalam beberapa penelitian sebelumnya, antara lain Palupi *et al.* (2011) yang menerapkan Algoritma Genetika untuk menentukan jumlah persediaan tiap jenis barang untuk memaksimalkan pemakaian gudang dan meminimumkan biaya total persediaan. Penelitian yang sejenis dilakukan oleh Samaher & Mahmudy (2015) pada obyek yang berbeda, yaitu permasalahan optimasi laba pada produksi jilbab. Penelitian lainnya oleh Milah & Mahmudy (2015) yang menerapkan algoritma *Evolution Strategies* pada permasalahan optimasi komposisi produksi pakan ternak sapi potong.

Berdasarkan pemaparan informasi yang telah dijelaskan, maka penulis mengusulkan mengenai penerapan algoritma *Evolution strategies* untuk optimasi laba berdasarkan jumlah produksi pada *Home Industry* tekstil. Algoritma *Evolution strategies* dipilih karena algoritma tersebut mampu menyelesaikan permasalahan optimasi sebagaimana algoritma evolusi yang lain seperti algoritma genetika.

1.2 Rumusan masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah yang bisa dikaji adalah sebagai berikut:

1. Bagaimana menerapkan algoritma *Evolution Strategy* untuk optimasi laba pada *Home Industry* tekstil?
2. Bagaimana kinerja yang dihasilkan dari optimasi laba berdasarkan jumlah produksi menggunakan algoritma *Evolution strategies*?
3. Bagaimana menentukan parameter algoritma *Evolution Strategy* yang tepat untuk optimasi laba pada *Home Industry* tekstil?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Membuat dan merancang algoritma *Evolution strategies* untuk optimasi laba untuk membantu pihak *Home Industry* mendapatkan jumlah produksi yang menghasilkan laba maksimal.
2. Menguji dan mengevaluasi kinerja yang dihasilkan dari optimasi laba berdasarkan jumlah produksi menggunakan algoritma *Evolution strategies*.
3. Mendapatkan nilai dari tiap parameter pada algoritma *Evolution Strategies* yang menghasilkan solusi terbaik.

1.4 Manfaat penelitian

Manfaat dari penelitian ini adalah

1. Mempermudah pengguna dalam menentukan jumlah produksi yang menghasilkan laba maksimal.
2. Menentukan jumlah tiap produk dengan waktu yang relatif singkat

1.5 Batasan masalah

Batasan masalah dalam penelitian ini adalah:

1. Algoritma yang digunakan adalah algoritma Evolution Strategy.
2. Data yang digunakan untuk penelitian adalah data yang berasal dari *Home Industry* tekstil "NAFA".
3. Optimasi laba yang dimaksud dalam penelitian ini adalah menentukan jumlah produksi pada setiap jenis produk untuk mendapatkan laba yang maksimal berdasarkan modal bahan baku yang tersedia, kebutuhan bahan per produk, dan laba tiap produk. Laba tersebut adalah laba penjualan eceran.
4. Bahan baku yang digunakan antara lain kain (tanpa mempertimbangkan corak), benang, karet, resleting, tali, silicon.
5. Jumlah produk yang dihasilkan memiliki batas minimum sejumlah produk yang dipesan (kondisional jika terdapat pesanan pada suatu produk) dan kisaran batas maksimum sejumlah bahan baku yang tersedia.
6. Parameter penentuan optimasi adalah jumlah kisaran produksi maksimum, lambda, ukuran populasi, dan jumlah generasi.
7. Jumlah produksi untuk seluruh produk yang melebihi 600 maka akan dikenakan biaya upah lembur. Biaya upah lembur memiliki nilai yang sama untuk setiap produk yaitu sebesar Rp 500 per produk.

1.6 Sistematika penulisan

Sistematika penulisan yang disusun dalam tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan.

BAB II Tinjauan Pustaka

Bab ini terbagi menjadi kajian pustaka dan dasar teori. Kajian pustaka membahas tentang perbandingan penelitian yang telah dilakukan dengan penelitian sekarang. Dasar teori membahas teori-teori yang mendukung dalam perancangan aplikasi optimasi laba pada produksi tekstil *Home Industry*

menggunakan algoritma *Evolution strategies* antara lain konsep dasar algoritma evolusi, *Evolution strategies*, laba, dan *Home Industry*.

BAB III Metodologi Penelitian

Membahas tentang metode yang digunakan untuk perancangan algoritma *Evolution strategies* untuk optimasi laba dan langkah kerja yang dilakukan dalam penulisan tugas akhir.

BAB IV Perancangan Sistem

Membahas mengenai rancangan aplikasi agar bisa memberikan solusi berupa jumlah produksi tiap produk agar mendapatkan laba maksimal.

BAB V Implementasi Program

Membahas tentang hasil perancangan berupa implementasi aplikasi dalam menentukan jumlah produk yang menghasilkan laba maksimal.

BAB VI Pengujian dan Pembahasan

Memuat hasil pengujian terhadap aplikasi yang telah direalisasikan dan telah memenuhi kriteria dan mampu memberikan solusi berupa jumlah produksi tiap produk.

BAB VII Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian algoritma *Evolution strategies* untuk optimasi serta saran untuk pengembangan lebih lanjut.



BAB 2 TINJAUAN PUSTAKA

Tinjauan pustaka terdiri atas kajian pustaka dan dasar teori. Kajian Pustaka adalah pembahasan tentang penelitian yang sudah dilakukan dan yang akan diusulkan untuk penelitian berikutnya. Dasar teori adalah pembahasan tentang teori yang diperlukan untuk perencanaan aplikasi.

2.1 Kajian pustaka

Penelitian pertama dengan judul “Algoritma Genetika Untuk Optimasi Persediaan Multi Barang Dalam Proses Produksi” (Palupi, Irhamah, & Mumpuni, 2011). Penelitian ini mengembangkan model persediaan EOQ (Economic Order Quantity) multi-item di PT UWBM Sidoarjo pada persediaan bahan baku untuk keperluan produksi. Tujuan yang ingin dicapai adalah menentukan jumlah persediaan tiap jenis barang yang sebaiknya disimpan sehingga memaksimalkan pemakaian gudang, yang berarti juga dapat meminimumkan biaya total persediaan. Secara umum biaya total persediaan dipengaruhi oleh 3 komponen biaya yaitu biaya pemesanan, biaya penyimpanan, dan biaya pembelian. Data yang digunakan adalah data persediaan dan permintaan bahan baku tepung terigu, gula, dan tepung tapioca mulai Januari 2010 sampai April 2011. Dari data tersebut maka akan dilakukan peramalan permintaan untuk tiga bulan mulai bulan Februari 2011 hingga April 2011 sebagai perbandingan. Terdapat 3 gen tiap kromosom yang masing-masing merupakan representasi dari data persediaan tepung terigu, gula, dan tepung tapioca. Dari model persediaan yang sesuai, dilakukan optimasi dengan metode Algoritma Genetika.

Penelitian kedua berjudul “Optimasi laba pada produksi jilbab *Home Industry* menggunakan algoritma genetika” (Samaher & Mahmudy, 2015). Pada penelitian ini 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. Hasil akhir dari penelitian adalah kromosom terbaik yang merupakan kombinasi produksi jilbab dengan laba maksimal dan pelanggaran kendala terendah. Namun, dalam penelitian tersebut belum ada mekanisme untuk mengatasi masalah konvergensi dini, sehingga dikhawatirkan mendapatkan nilai yang sama sebelum mencapai titik optimum.

Penelitian ketiga berjudul “Implementasi Algoritma *Evolution Strategies* Untuk Optimasi Komposisi Pakan Ternak Sapi Potong” (Milah & Mahmudy, 2105). Penelitian ini berusaha untuk mengatasi masalah komposisi pakan ternak sapi potong menggunakan algoritma *evolution strategies* (ES). Tipe proses ES yang digunakan adalah $(\mu/r + \lambda)$ sehingga proses reproduksi melibatkan rekombinasi dan hasil mutasi nantinya akan diikuti bersama parent dalam proses seleksi dan seleksi menggunakan elitism selection. Solusi optimal

diperoleh dari ukuran populasi sebanyak 50, ukuran offspring 4, range kromosom 0-20 (real code) dan jumlah generasi 500 memperoleh rata-rata nilai *fitness* tertinggi yaitu 0.998. Hasil akhir berupa rekomendasi komposisi pakan ternak sapi potong.

Analisis tentang perbandingan penelitian sebelumnya yang telah dilakukan dan penelitian yang diusulkan dengan menggunakan metode *Evolution Strategies* diuraikan pada Tabel 2.1.



Tabel 2.1 Kajian pustaka

No.	Judul	Object (Input)	Metode (Proses)	Hasil
1	Algoritma Genetika Untuk Optimasi Persediaan Multi Barang Dalam Proses Produksi	Object: persediaan tepung trigu, gula, dan tapioka Input: <ul style="list-style-type: none"> - Data permintaan - persediaan bahan baku - Ukuran populasi - Ukuran offspring - Jumlah generasi 	Metode: Algoritma Genetika Proses: <ol style="list-style-type: none"> 1. Inialisasi 2. Evaluasi nilai <i>fitness</i> 3. Seleksi orang tua 4. Crossover 5. Mutasi 6. Etilisme 7. Seleksi 	Frekuensi dalam sekali pemesanan, jumlah persediaan tiap bahan baku selama tiga bulan yang memaksimumkan pemakaian gudang dan meminimumkan biaya total persediaan
2	Optimasi laba pada produksi jilbab <i>Home Industry</i> menggunakan algoritma genetika	Object: Jilbab. Input: <ul style="list-style-type: none"> - Data kebutuhan bahan baku, - Laba setiap produk - Persediaan bahan perminggu - Ukuran populasi - generasi - <i>crossover rate</i> - <i>mutation rate</i> 	Metode: Algoritma Genetika Proses: <ol style="list-style-type: none"> 1. Inialisasi 2. Reproduksi <ol style="list-style-type: none"> a. Crossover b. Mutasi (Reciprocal exchange & Random) 3. Evaluasi 4. Seleksi (<i>Elitism Selection</i>) 	Kombinasi produksi jilbab dengan laba maksimal, nilai <i>fitness</i>

Tabel 2.1 Kajian pustaka (lanjutan)

No.	Judul	Object (Input)	Metode (Proses)	Hasil
3	Implementasi Algoritma <i>Evolution Strategies</i> Untuk Optimasi Komposisi Pakan Ternak Sapi Potong	Object: Pakan Ternak Input: <ul style="list-style-type: none"> - <i>Miu</i> - λ - Panjang String kromosom minimal - Panjang String kromosom maksimal - Jumlah Perulangan 	Metode: Evolution Strategy Proses: <ol style="list-style-type: none"> 1. Inisialisasi 2. Reproduksi <ol style="list-style-type: none"> a. Rekombinasi b. Mutasi 3. Evaluasi 4. Seleksi 	Rekomendasi komposisi pakan ternak sapi potong
4	Implementasi Algoritma <i>Evolution strategies</i> Untuk Optimasi Laba Berdasarkan Jumlah Produksi Pada Home Industry Tekstil	Object: Tekstil. Input: <ul style="list-style-type: none"> - Data kebutuhan bahan baku - Data keuntungan per produk - Jumlah pesanan per produk (jika ada) - Persediaan perbulan - <i>Miu</i> (μ) - λ - Jumlah Generasi 	Metode: Evolution Strategy Proses: <ol style="list-style-type: none"> 1. Inisialisasi 2. Reproduksi <ol style="list-style-type: none"> a. Rekombinasi b. Mutasi 3. Evaluasi 4. Seleksi (<i>Elitism Selection</i>) 	Jumlah produksi per produk yang menghasilkan laba maksimum, nilai <i>fitness</i>

2.2 Algoritma evolusi

Evolutionary Algorithms (EAs) merupakan sebuah algoritma optimasi berbasis populasi metaheuristic generik. EAs menggunakan beberapa mekanisme yang terinspirasi oleh evolusi biologi: reproduksi, mutasi, rekombinasi, dan seleksi (Rifqi, Maharani, & Shaufiah, 2011). Menurut teori evolusi terdapat sejumlah individu dalam populasi. Dari generasi ke generasi, individu-individu ini berperan sebagai induk (parent) yang melakukan reproduksi menghasilkan keturunan (offspring). Individu-individu ini (beserta offspring) berevolusi dan individu-individu yang lebih baik (mampu beradaptasi dengan lingkungannya) mempunyai peluang lebih besar untuk melewati seleksi alam (natural selection) dan bertahan hidup. Individu yang lebih baik juga cenderung (tidak selalu tapi mempunyai kemungkinan lebih besar) menghasilkan keturunan yang lebih baik sehingga dari generasi ke generasi akan terbentuk populasi yang lebih baik. Keseluruhan proses dalam EAs ditunjukkan pada Gambar 2.1



Gambar 2.1 Proses-proses dalam Algoritma Evolusi

Sumber: (Mahmudy, 2013)

Individu-individu dalam populasi di EAs merepresentasikan solusi dari masalah yang akan diselesaikan. Sebuah fungsi *fitness* digunakan untuk mengukur seberapa baik suatu individu. Individu terbaik di akhir generasi bisa didekodekan sebagai solusi terbaik yang bisa diperoleh. Dari penjelasan di atas, EAs bisa dikelompokkan dalam algoritma 'generate and test' yang berbasis populasi (population based). EA juga bersifat stochastic, setiap kali dijalankan untuk masalah yang sama ada kemungkinan menghasilkan solusi yang berbeda (Smith & Eiben dalam Mahmudy, 2013).

2.2.1 Tipe algoritma evolusi

Berbagai tipe EAs telah dikembangkan sebagai berikut (Mahmudy, 2013):

- Algoritma genetika (Genetic Algorithms, GAs), merupakan tipe EAs yang paling populer dan banyak diterapkan pada masalah-masalah kompleks. Pada awalnya banyak menggunakan representasi string biner tapi kemudian berkembang dengan menggunakan vektor bilangan integer dan pecahan (real). Pembangkitkan solusi baru banyak mengandalkan proses tukar silang (crossover). Mutasi biasanya dipakai sebagai operator tambahan untuk menjaga keragaman populasi.
- Evolution strategies* (ES), representasi solusi biasanya menggunakan vektor bilangan pecahan. Mutasi merupakan operator reproduksi utama. Mekanisme self-adaptation digunakan untuk mengontrol perubahan nilai parameter pencarian.

- c. Genetic Programming (GP), digunakan untuk mengoptimasi rangkaian program komputer yang direpresentasikan dalam bentuk struktur data pohon (tree).
- d. Evolutionary Programming (EP), mempunyai tujuan seperti GP tapi prinsip kerjanya seperti ES. Finite State Machines (FSM) digunakan untuk merepresentasikan program komputer.

2.3 Evolution strategies

Teknik optimasi *evolution strategies* (ES) dicetuskan sejak awal tahun 1960-an dan kemudian dikembangkan lebih lanjut pada tahun 1970-an oleh Ingo Rechenberg, Hans-Paul Schwefel, dan rekan-rekannya di Technical University of Berlin (TUB) (Beyer & Schwefel dalam Mahmudy, 2013). Seperti halnya GAs, ES telah diaplikasikan dalam berbagai bidang, misalnya penjadwalan pemrosesan sinyal digital pada system multiprocessor (Greenwood *et al.* dalam Mahmudy, 2013), pemrosesan citra dan computer vision (Louchet dalam Mahmudy, 2013), optimasi pelepasan airbag secara otomatis pada mobil (Ostertag *et al.* dalam Mahmudy, 2013), optimasi produksi (Ramuna & Mahmudy, 2015), penjadwalan tugas pada real-time distributed computing systems (Greenwood *et al.* dalam Mahmudy, 2013), optimasi pengelompokan (Kashan, Akbari, & Ostadi, 2015), dan transkripsi music polyphonic (Mbiyaki, 2013).

Ciri utama *Evolution Strategies* (ES) adalah penggunaan vektor bilangan pecahan (real-vector) sebagai representasi solusi. Berbeda dengan GAs yang menggunakan crossover sebagai operator reproduksi utama dan mutasi sebagai operator penunjang, ES lebih bertumpu pada operator mutasi. Mekanisme self-adaptation digunakan untuk mengontrol perubahan nilai parameter pencarian. GAs dan ES bisa digunakan untuk menyelesaikan permasalahan yang sama. Tetapi mana yang terbaik di antara kedua metode tersebut sangat tergantung pada permasalahan yang dihadapi (Mahmudy, 2013). Keuntungan menggunakan algoritma *evolution strategies* adalah memungkinkan untuk menangani secara bersamaan beberapa tujuan dan kendala untuk mencapai pendekatan yang lebih optimum (Binh & Korn).

2.3.1 Istilah dalam Evolution Strategies

Pada algoritma Evolution Strategies terdapat beberapa istilah yang digunakan untuk merepresentasikan suatu proses dan hal-hal lain yang berkaitan dengan algoritma tersebut. Beberapa istilah yang digunakan tersebut disajikan pada Tabel 2.2.

Tabel 2.2 Istilah-istilah pada algoritma *Evolution Strategies*

Istilah	Definisi
Genotype (Gen)	Sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom

Tabel 2.2 Istilah-istilah pada algoritma *Evolution Strategies* (lanjutan)

Istilah	Definisi
Kromosom	Gabungan gen-gen yang membentuk nilai tertentu
Individu	Satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
Populasi	Merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi
Generasi	Menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma Evolusi
Offspring	Suatu proses reproduksi untuk menghasilkan keturunan dari hasil mutasi maupun rekombinasi
Rekombinasi	Proses pindah silang untuk membangkitkan individu baru. Bisa menggunakan lebih dari dua induk
Mutasi	Proses untuk membangkitkan individu baru dengan mengganti gen dengan nilai tertentu
μ (miu)	Menyatakan ukuran populasi
λ (lambda)	Menyatakan banyaknya offspring yang dihasilkan pada proses reproduksi (sama seperti crossover rate dan mutation rate pada GAs)
r	Notasi pada tipe algoritma <i>Evolution Strategies</i> yang menggunakan metode Rekombinasi pada tahap reproduksi

Sumber: Diadaptasi dari Satriyanto (2009)

2.3.2 Struktur dasar evolution strategies

Tiga mekanisme utama yang digunakan dalam *Evolution Strategies* adalah rekombinasi, mutasi, dan seleksi (Kramer, 2014). Beberapa notasi digunakan oleh ES adalah μ (miu) menyatakan ukuran populasi (sama seperti μ pada GAs), λ (lambda) menyatakan banyaknya offspring yang dihasilkan pada proses reproduksi (sama seperti crossover rate dan mutation rate pada GAs). Beberapa penelitian menyarankan besarnya nilai λ sebesar 7μ (Mahmudy, 2013).

Apabila $P(t)$ dan $C(t)$ merupakan populasi (parents) dan offspring pada generasi ke- t , maka siklus ES dapat dideskripsikan pada Gambar 2.2.

```

Procedure EvolutionStrategies
Begin
  t = 0
  inialisasi P(t): generate  $\mu$  individu
  while (bukan kondisi berhenti) do
    rekombinasi: produksi C(t) sebanyak  $\lambda * \mu/2$  dari P(t)
    mutasi C(t)
    seleksi P(t+1) dari P(t) dan C(t)
    t = t+1
  end while
end
  
```

Gambar 2.2 Siklus algoritma *Evolution Strategies*

Sumber: diadopsi dari Mahmudy (2013)

Perbedaan nyata ES dan GAs adalah pada operator yang digunakan. Perbedaan yang lain adalah mutasi pada GAs digunakan untuk menghasilkan individu baru (offspring) sebagai tambahan dari offspring yang diproduksi oleh operator crossover. Pada ES, mutasi diterapkan pada offspring yang dihasilkan proses rekombinasi. Rekombinasi pada ES mirip dengan operator crossover pada GAs tapi bisa menggunakan lebih dari dua induk. Karena ES lebih mengandalkan mutasi, maka proses rekombinasi tidak selalu digunakan (Mahmudy, 2013). Secara umum terdapat empat tipe proses dari ES, yaitu:

Tabel 2.3 Tipe *Evolution Strategies*

Tipe ES	Rekombinasi	Seleksi melibatkan
(μ, λ)	Tidak	Offspring
$(\mu/r, \lambda)$	Ya	Offspring
$(\mu + \lambda)$	Tidak	Induk dan Offspring
$(\mu/r + \lambda)$	Ya	Induk dan Offspring

Sumber: Diadaptasi Mahmudy (2013)

$ES(\mu, \lambda)$ tidak menggunakan rekombinasi dalam proses reproduksi. Seleksi menggunakan elitism selection hanya melibatkan individu dalam offspring, individu induk dalam populasi tidak dilibatkan. $ES(\mu/r, \lambda)$ serupa dengan $ES(\mu, \lambda)$ dengan tambahan melibatkan proses rekombinasi. $ES(\mu+\lambda)$ tidak menggunakan rekombinasi dan proses seleksi menggunakan elitism selection melibatkan individu offspring dan induk.

2.3.3 Representasi kromosom

Seperti halnya untuk real-coded GA, variabel keputusan (x_1 dan x_2) langsung menjadi gen string chromosome. Selain gen yang menyatakan variabel keputusan, parameter tambahan yang melekat pada setiap chromosome adalah σ (sigma). Nilai ini menyatakan level mutasi untuk chromosome tersebut. Nilai ini akan ikut berubah secara adaptif sepanjang generasi. Jika P adalah satu chromosome maka $P=(x_1, x_2, \sigma_1, \sigma_2)$ dengan panjang string sebesar 4 (Mahmudy, 2013).

2.3.4 Inisialisasi

Inisialisasi dilakukan untuk membangkitkan himpunan solusi baru secara acak/random yang terdiri atas sejumlah string chromosome dan ditempatkan pada penampungan yang disebut populasi. Dalam tahap ini harus ditentukan (μ) mu yang merupakan ukuran populasi. Nilai ini menyatakan banyaknya individu/chromosome yang ditampung dalam populasi. Panjang setiap string chromosome (*stringLen*) dihitung berdasarkan presisi variabel solusi yang kita cari (Mahmudy, 2013).

Populasi inisial dibangkitkan secara random. Misalkan ditentukan masalah maksimasi (mencari nilai maksimum) dari sebuah fungsi sebagai berikut:

$$\max f(x_1, x_2) = 19 + x_1 \sin(x_1\pi) + (10 - x_2) \sin(x_2\pi) \quad (2.1)$$
$$-5,0 \leq x_1 \leq 9,8, \quad 0,0 \leq x_2 \leq 7,3$$

Nilai σ_1 dan σ_2 dibangkitkan dalam rentang $[0,1]$. Misalkan ditentukan $\mu = 4$ maka akan dihasilkan populasi seperti pada Tabel 2.4.

Tabel 2.4 Contoh populasi awal

P(t)	x_1	x_2	σ_1	σ_2	$f(x_1, x_2)$
1	1.4898	2.0944	0.14197	0.9109	19.82128
2	8.4917	2.5754	0.53801	0.86904	34.70609
3	-1.8461	1.7097	0.99835	0.49351	11.58639
4	5.8114	5.0779	0.40521	0.98911	14.56208

Sumber: Mahmudy (2013)

2.3.5 Reproduksi

Pada proses reproduksi akan dilakukan proses untuk menghasilkan keturunan dari individu-individu yang telah dibangkitkan pada tahap inisialisasi. Pada algoritma Evolution Strategies terdapat dua proses utama dalam reproduksi yaitu rekombinasi dan mutasi.

2.3.5.1 Rekombinasi

Rekombinasi dilakukan untuk menghasilkan offspring sebanyak $\lambda * \mu / 2$ dari sejumlah μ individu dalam populasi. Setiap satu individu offspring dihasilkan

dari beberapa induk. Induk dipilih secara acak dari populasi. Metode rekombinasi paling sederhana adalah dengan menghitung rata-rata nilai elemen induk. Contoh proses rekombinasi diberikan sebagai berikut:

- Misalkan offspring didapatkan dari 2 induk. Jika P1 dan P3 terpilih maka akan didapatkan offspring $C=(-0,17815, 1,90205, 0,57016, 0,70221)$.
 - Misalkan offspring didapatkan dari 3 induk. Jika P1, P2 dan P3 terpilih maka akan didapatkan offspring $C=(2,71180, 2,12650, 0,55944, 0,75782)$.
- Pada studi kasus ini, misalkan $\lambda=6$ dan offspring didapatkan dari 2 induk.

Contoh hasil rekombinasi ditampilkan pada Tabel 2.5.

Tabel 2.5 Contoh hasil rekombinasi

C(t)	Induk	x_1	x_2	σ_1	σ_2	$f(x_1, x_2)$
C1	P1 dan P3	-0.17815	1.90205	0.57016	0.70221	16.64183
C2	P2 dan P3	3.3228	2.14255	0.76818	0.68128	19.5813
C3	P1 dan P4	3.6506	3.58615	0.27359	0.95001	9.57005
C4	P2 dan P4	7.15155	3.82665	0.47161	0.92907	12.52404
C5	P1 dan P3	-0.17815	1.90205	0.57016	0.70221	16.64183
C6	P3 dan P4	1.98265	3.3938	0.70178	0.74131	12.65007

Sumber: Mahmudy (2013)

2.3.5.2 Mutasi

Pada algoritma Evolution Strategies, mutasi merupakan operator reproduksi utama. Mutasi digunakan untuk menjaga keragaman populasi. Mekanisme self-adaptation digunakan untuk mengontrol perubahan nilai parameter pencarian. Misalkan $P = (x_1, x_2, \sigma_1, \sigma_2)$ adalah individu yang terpilih untuk melakukan mutasi, maka dihasilkan offspring $P' = (x'_1, x'_2, \sigma'_1, \sigma'_2)$ sebagai berikut (Mahmudy, 2013):

$$x' = x + \sigma N(0,1) \quad (2.2)$$

Rumusan ini bisa didetailkan sebagai berikut:

$$x'_1 = x_1 + \sigma_1 N_1(0,1)$$

$$x'_2 = x_2 + \sigma_2 N_2(0,1)$$

$N(0,1)$ merupakan bilangan acak yang mengikuti sebaran normal dengan rata-rata sebesar 0 dan standard deviasi sebesar 1. Pada program komputer, nilai $N(0,1)$ bisa didapatkan dengan membangkitkan dua bilangan random r_1 dan r_2 pada interval $[0,1]$. Rumus yang digunakan adalah (Schwefel dalam Mahmudy, 2013):

$$N(0,1) = \sqrt{-2 \cdot \ln r_1} \sin 2\pi r_2 \quad (2.3)$$

Misalkan $r_1 = 0,4749$ dan $r_2 = 0,3296$ maka didapatkan $N(0,1) = 1,0709$.

Nilai σ dinaikkan jika hasil mutasi menghasilkan individu yang lebih baik dari induknya. Jika tidak maka nilai σ diturunkan. Nilai σ dinaikkan dengan rumusan $\sigma' = \sigma \times 1.1$ dan diturunkan dengan rumusan $\sigma' = \sigma \times 0.9$ (Mahmudy, 2013). Berikut contoh hasil mutasi ditampilkan pada Tabel 2.6.

Tabel 2.6 Contoh hasil mutasi

C'(t)	N1(0,1)	N2(0,1)	x ₁	x ₂	σ_1	σ_2	f(x ₁ ,x ₂)
C7	0.1885	0.2747	-0.07068	2.094946	0.62717	0.77243	21.3387
C8	-1.7947	-0.1359	1.944154	2.049965	0.84499	0.7494	19.90344
C9	-0.5603	-1.8657	3.497309	1.813724	0.30095	1.04501	10.98098
C10	0.6189	-0.4613	7.443427	3.398068	0.42445	0.83617	5.407509
C11	-0.1371	-0.4201	-0.25632	1.607053	0.51314	0.63199	11.26206
C12	1.1125	-0.2153	2.763377	3.234196	0.77195	0.81544	16.32937

Sumber: Mahmudy (2013)

Pada Tabel 2.6 nilai gen x₁ pada individu C7 didapatkan dari perhitungan berikut:

$$\begin{aligned} x_1' &= x_1 + \sigma_1 \times N_1(0,1) \\ &= -0.17815 + 0.57016 \times 0.1885 \\ &= -0.07068 \end{aligned}$$

Nilai fitness yang dihasilkan pada individu C7 sebesar 21.3387 lebih besar dibandingkan dengan nilai fitness pada individu induknya (C1) sebesar 16.64183. Sehingga nilai σ_1' pada individu C7 didapatkan dengan perhitungan berikut:

$$\begin{aligned} \sigma' &= \sigma \times 1.1 \\ &= 0.57016 \times 1.1 \\ &= 0.62717 \end{aligned}$$

2.3.6 Seleksi

Seleksi merupakan proses untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan hidup pada generasi berikutnya (Mahmudy, 2013). Pada penelitian ini, metode seleksi yang digunakan adalah metode seleksi elitism. Metode seleksi *elitism* bekerja dengan mengumpulkan semua individu dalam populasi (*parent*) dan *offspring* dalam satu penampungan. *miu* individu yang memiliki fitness terbaik dalam penampungan ini akan lolos untuk masuk dalam generasi selanjutnya. Metode seleksi ini menjamin individu yang terbaik akan selalu lolos. Pseudo-code *elitism selection* disajikan pada Gambar 2.3 sebagai berikut:

```

PROCEDURE ElitismSelection
Input:
    POP:      himpunan individu pada populasi
    pop_size: ukuran populasi
    OS:       himpunan individu anak (offspring) hasil reproduksi
              menggunakan crossover and mutasi

Output:
    POP:      himpunan individu pada populasi setelah proses
              seleksi selesai

/* gabungkan individu pada POP dan OS ke dalam TEMP */
TEMP ← Merge (POP, OS)
/* urutkan individu berdasarkan fitness secara ascending */
OrderAscending (Temp)
/* copy pop_size individu terbaik ke POP */
POP ← CopyBest (Temp, pop_size)

END PROCEDURE
    
```

Gambar 2.3 Pseudo-code elitism selection

Sumber: Mahmudy (2013)

Salah satu kelemahan dari *elitism selection* adalah tidak memberikan kesempatan kepada individu dengan nilai fitness rendah untuk bereproduksi. Dalam beberapa kasus, solusi optimum justru bisa dicapai dari hasil reproduksi individu dengan nilai fitness rendah (Mahmudy, 2013). Seleksi menggunakan *elitism selection* melibatkan individu dalam *offspring* dan individu induk dalam populasi. Dari proses ini didapatkan populasi baru yang dapat dilihat pada Tabel 2.7.

Tabel 2.7 Contoh hasil seleksi

P(t+1)	asal	x ₁	x ₂	σ ₁	σ ₂	f(x ₁ ,x ₂)
P1	P2	8.4917	2.5754	0.53801	0.86904	34.70609
P2	C1	-0.07068	2.09495	0.62717	0.77243	21.3387
P3	C2	1.94415	2.04996	0.84499	0.7494	19.90344
P4	P1	1.4898	2.0944	0.14197	0.9109	19.82128

Sumber: Mahmudy (2013)

2.3.7 Kondisi berhenti

Setelah proses seleksi dijalankan dan menghasilkan parent baru di dalam populasi, maka akan dilakukan pengecekan kondisi berhenti. Kondisi berhenti standar yang biasa digunakan adalah (Beyer & Schwefel dalam Munawarah & Mahmudy, 2015) :

1. Batas maksimum iterasi / generasi telah tercapai.

2. Telah melewati batas waktu pemrosesan.
3. Terjadi konvergensi pada nilai fitness yaitu nilai fitness tidak mengalami perubahan yang signifikan atau bahkan tidak mengalami perubahan sama sekali.

2.4 Home industry

Secara harfiah, Home berarti rumah, tempat tinggal, ataupun kampung halaman, sedang Industri, dalam Kamus Besar Bahasa Indonesia dapat diartikan sebagai kerajinan, usaha produk barang dan ataupun perusahaan. Jadi, Home Industry adalah rumah usaha produk barang atau bisa juga disebut perusahaan kecil. Dikatakan sebagai perusahaan kecil karena jenis kegiatan ekonomi ini dipusatkan di rumah (Abrianto, 2012).

Pengertian usaha kecil secara jelas tercantum dalam UU No. 20 Tahun 2008 tentang Usaha Mikro Kecil Menengah, yang menyebutkan bahwa usaha kecil adalah usaha dengan kekayaan bersih paling banyak Rp200 juta (tidak termasuk tanah dan bangunan tempat usaha) dengan hasil penjualan tahunan paling banyak Rp1.000.000.000. Kriteria lainnya dalam UU No. 20 Tahun 2008 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 (Abrianto, 2012).

Jika terdaftar pada Dinas Perdagangan Kabupaten/kota permohonan izin ke pemerintah untuk menjalankan usaha, Home Industry termasuk dalam kategori peraturan Surat Izin Usaha Perdagangan (SIUP) Putih, yaitu perusahaan kecil yang dengan kekayaan kurang dari 200 juta. Home Industry juga dapat berarti Industri Rumah Tangga, karena termasuk dalam kategori usaha kecil yang dikelola keluarga. Kriteria-kriteria suatu usaha dikatakan Industri Rumah Tangga (Home Industry) yaitu (Abrianto, 2012):

1. Kegiatan Industri dilakukan di rumah tangga
2. Tenaga kerja yang dipekerjakan tidak lebih dari 3 orang
3. Peralatan pengolahan yang digunakan mulai dari manual hingga alat semi otomatis.

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.

2.4.1 Jenis home industry

Berikut adalah beberapa bentuk dan jenis Home Industry ditunjukkan pada Tabel 2.8.

Tabel 2.8 Jenis Home Industry

Jenis Home Industry	Contoh
Home Industry bidang kosmetik (alat-alat kecantikan)	Face Lotion (Lotion Muka), Skin Tonic Lotion, Cleansing Cream, Bedak Powder, Minyak Rambut Kental, Minyak Rambut Hair Cream.
Home Industry bidang kebutuhan sehari-hari	sabun mandi, sabun cuci batangan, sabun cuci deterjen, pasta gigi.
Home Industry bidang obat-obatan ringan	minyak angin, obat gosok, obat kutu busuk, obat nyamuk.
Home Industry bidang makanan	keripik ubi, keripik pisang, emping.
Home Industry bidang sandang	batik, jilbab, sandal.
Home Industry bidang minuman	soda, jus buah.

Sumber: Diadaptasi dari Abrianto (2012)

2.5 Laba

Laba merupakan main goals atau tolak ukur keberhasilan manajemen perusahaan berbasis bisnis atau profit seeking terlebih pada bagian manajemen keuangan. Salvatore dalam Dahlia (2011) membedakan antara laba bisnis dan laba ekonomi. Laba bisnis (business profit) mengacu pada penerimaan perusahaan dikurangi biaya eksplisit atau biaya akuntansi perusahaan. Biaya eksplisit merupakan biaya yang benar-benar dikeluarkan dari kantong perusahaan untuk membeli atau menyewa input yang dibutuhkan dalam produksi. Sedangkan laba ekonomi merupakan penerimaan perusahaan dikurangi oleh biaya eksplisit dan biaya implisit. Biaya implisit mengacu pada nilai input yang dimiliki perusahaan dan dipergunakan untuk proses produksinya sendiri.

Menurut Tunggal dalam Dahlia (2011) terdapat dua definisi dari pendapatan atau laba yaitu:

- Pendapatan operasi didefinisikan sebagai pendapatan operasi dikurangi semua biaya selain bunga (interest) dan pajak penghasilan.
- Laba bersih didefinisikan sebagai total pendapatan dikurangi semua biaya dan pengeluaran.

Dari kedua pendapat tersebut, maka pengertian laba adalah ganjaran atas selisih pendapatan dan pengeluaran. Dalam analisis cost-volume-profit yaitu total penjualan (price per unit times quantities of sales) dikurangi total cost (fixed cost plus variable cost)

BAB 3 METODOLOGI PENELITIAN

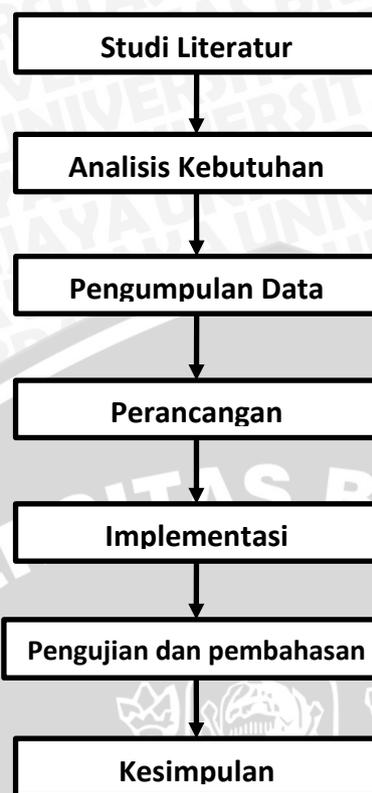
Skripsi dengan judul “Implementasi Algoritma Evolution Strategies Untuk Optimasi Laba Berdasarkan Jumlah Produksi Pada Home Industry Tekstil” ini bisa digolongkan sebagai penelitian implementatif dengan pendekatan perancangan (*design*). Penelitian ini akan menghasilkan sebuah purwarupa (*prototype*) berupa perangkat lunak (*software*) yang bisa digunakan untuk optimasi laba berdasarkan jumlah produksi.

3.1 Tahapan penelitian

Sebagai penelitian implementatif dengan pendekatan perancangan (*design*) maka terdapat proses yang utuh mulai dari analisis, perancangan hingga pengujian. Tahapan penelitian adalah sebagai berikut:

1. Studi literatur: pada tahap ini dilakukan proses mempelajari literature dari beberapa bidang ilmu yang berhubungan dengan implementasi algoritma *evolution strategies* untuk optimasi laba berdasarkan jumlah produksi pada home industry tekstil
2. Analisis kebutuhan: menganalisis kebutuhan yang diperlukan untuk implementasi algoritma *evolution strategies*.
3. Pengumpulan data: mengumpulkan data yang dibutuhkan untuk penelitian.
4. Perancangan: siklus penyelesaian masalah menggunakan algoritma *Evolution Strategies*, perancangan operator dalam algoritma *Evolution Strategies* (Mutasi, Rekombinasi, seleksi).
5. Implementasi: membangun aplikasi yang mengacu pada perancangan *Evaluation Strategies* dan menerapkan hal yang telah didapatkan dalam proses studi literatur.
6. Pengujian: melakukan uji parameter *Evolution Strategies* yang menghasilkan solusi terbaik.
7. Analisis hasil: Dari uji coba yang telah dilakukan, maka hasil yang telah didapatkan akan dianalisis. Analisis hasil dilakukan dengan cara menganalisis hasil nilai *fitness* yang telah didapatkan dalam tahap pengujian.
8. Kesimpulan: Kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian metode yang diterapkan sudah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode.

Diagram tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Metodologi Penelitian

3.2 Studi literatur

Pada tahap ini dilakukan proses mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan implementasi algoritma evolution strategies untuk optimasi laba berdasarkan jumlah produksi pada home industry tekstil. Literatur tersebut didapatkan dari buku, jurnal, penelitian sebelumnya dan dokumentasi proyek. Beberapa literatur yang dibutuhkan diantaranya:

- a. Algoritma Evolution Strategies
- b. Pemrograman dengan menggunakan bahasa JAVA
- c. Home Industry

3.3 Analisis kebutuhan

Agar implementasi berjalan dengan baik, maka dibutuhkan spesifikasi tertentu pada perangkat lunak, perangkat keras, dan data yang digunakan. Berikut ini adalah kebutuhan yang digunakan dalam implementasi algoritma *Evolution Strategies* untuk optimasi laba berdasarkan jumlah produksi pada *Home Industry* tekstil:

1. Kebutuhan perangkat keras meliputi:
 - Laptop Toshiba AMD A4 RAM 4GB HardDisk 500GB

2. Kebutuhan perangkat lunak meliputi:

- Sistem operasi Windows 10
- Bahasa pemrograman JAVA
- Aplikasi Netbeans IDE 8.0

3. Data yang dibutuhkan

- Data kebutuhan bahan baku tiap produk
- Data keuntungan penjualan tiap produk
- Data persediaan bahan baku per bulan

3.4 Pengumpulan data

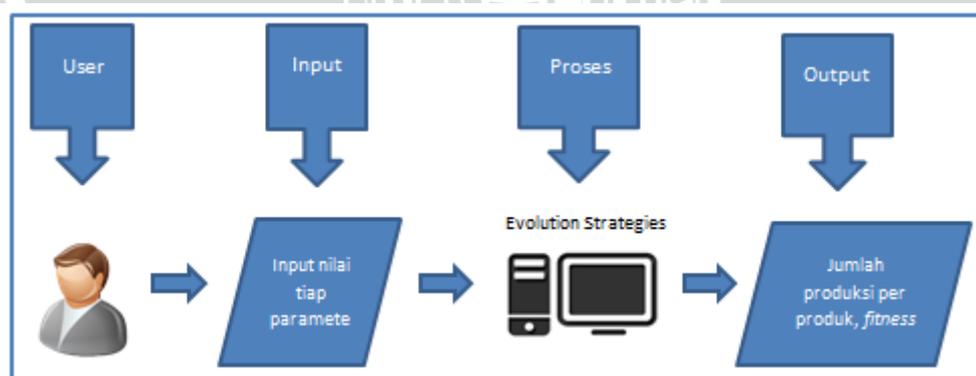
Lokasi penelitian skripsi ini adalah Jalan Yos Sudarso RT 9 RW 3 No. 80 Kelurahan Medaeng Kecamatan Waru Kabupaten Sidoarjo. Data yang digunakan untuk penelitian ini adalah data Primer. Data primer adalah data yang didapatkan langsung dari responden penelitian. Metode pengumpulan data dilakukan dengan wawancara terhadap responden. Pada penelitian ini menggunakan data kebutuhan bahan baku per produk, laba penjualan per produk, dan persediaan bahan baku per bulan.

3.5 Perancangan

Perancangan dibuat untuk rancangan langkah kerja dari aplikasi yang akan dibangun secara menyeluruh. Aplikasi yang akan dibuat untuk optimasi laba menggunakan metode Evolution Strategies. Pada system ini akan dihasilkan jumlah produksi optimum per produk yang menghasilkan laba maksimum.

3.5.1 Model perancangan sistem

Model perancangan sistem menjelaskan mengenai cara kerja sistem secara terstruktur mulai dari *input* yang dimasukkan hingga mendapatkan hasil. Diagram model perancangan sistem dapat dilihat pada gambar 3.2.



Gambar 3.2 Diagram blok sistem

Pada tahap awal User meginputkan nilai tiap parameter berupa λ , μ , jumlah generasi, dan jumlah minimal perproduk (jika ada pesanan). Kemudian data masukan akan diproses dengan algoritma *Evolution Strategies*

sehingga menghasilkan data keluaran berupa jumlah produksi dan nilai fitness yang dihasilkan. Tipe ES yang digunakan adalah $(\mu/r + \lambda)$ sehingga metode rekombinasi dan mutasi digunakan pada proses *offspring*, dan seleksi melibatkan induk dan *offspring*.

3.6 Implementasi

Implementasi adalah fase membangun aplikasi yang mengacu pada perancangan Evaluation Strategies dan menerapkan hal yang telah didapatkan dalam proses studi literatur. Fase – fase yang ada dalam implementasi antara lain implementasi interface menggunakan software Netbeans dan implementasi algoritma kedalam bahasa pemrograman JAVA.

3.6.1 Algoritma yang Digunakan

Penelitian ini menggunakan algoritma *Evolution Strategies* yang seperti telah diuraikan pada Bab 2 telah terbukti efektif digunakan untuk berbagai masalah optimasi. Implementasi algoritma menggunakan bahasa pemrograman JAVA karena merupakan bahasa pemrograman mudah dipelajari dan mempunyai kinerja yang tinggi.

3.7 Pengujian dan pembahasan

Uji coba dilakukan untuk mengetahui apakah aplikasi berjalan dengan baik dan sesuai dengan spesifikasi kebutuhan yang telah. Uji coba aplikasi dilakukan dengan melakukan beberapa perubahan pada nilai parameter maupun memodifikasi metode yang ada pada algoritma *Evolution Strategies* agar mendapatkan nilai *fitness* yang maksimum. Beberapa uji coba yang diajukan antara lain:

1. Uji coba untuk menentukan kisaran produksi maksimum (*Range* kromosom).
2. Uji coba untuk menentukan nilai λ (lambda) yang menghasilkan nilai *fitness* optimum.
3. Uji coba untuk menentukan ukuran populasi(μ) yang optimum.
4. Uji coba untuk menentukan jumlah generasi yang menghasilkan nilai *fitness* optimum.

3.7.1 Uji coba kisaran produksi maksimum

Uji coba menentukan kisaran produksi maksimum dilakukan untuk mengetahui kisaran batas maksimum kuantitas produksi suatu produk agar mampu menghasilkan fitness yang optimum berdasarkan jumlah bahan yang tersedia. Nilai kisaran produksi maksimum yang akan diujikan adalah rentang 10 sampai 100. Jumlah generasi sebanyak 100 generasi. Nilai λ (lambda) yang digunakan adalah 4. Nilai μ yang digunakan dalam setiap percobaan berbeda-beda agar didapatkan hasil yang lebih bervariasi. Nilai μ yang digunakan berturut-turut adalah 15, 20, 25, 30, 35, 40, 45, 50. Rancangan uji coba menentukan nilai kisaran produksi maksimum yang menghasilkan nilai *fitness* yang optimum disajikan dalam Tabel 3.1.

Tabel 3.1 Uji coba kisaran produksi maksimum

Nilai kisaran produksi maksimum	Nilai <i>Fitness</i> pada <i>miu</i> -								Rata-rata <i>fitness</i>
	15	20	25	30	35	40	45	50	
10									
20									
30									
40									
50									
60									
70									
80									
90									
100									

3.7.2 Uji coba nilai λ (lambda)

Uji coba menentukan nilai λ (lambda) dilakukan untuk mengetahui pengaruh nilai λ (lambda) yang terhadap nilai *fitness*. Nilai λ (lambda) yang digunakan adalah rentang 2 sampai 10. Nilai kisaran produksi maksimum yang digunakan adalah nilai terbaik pada uji coba kisaran produksi maksimum. Jumlah generasi sebanyak 100 generasi. Nilai *miu* yang digunakan dalam tiap percobaan berbeda-beda agar hasil yang didapatkan bisa lebih bervariasi. Nilai *miu* yang digunakan berturut-turut adalah 15, 20, 25, 30, 35, 40, 45, 50. Rancangan uji coba menentukan nilai λ (lambda) yang menghasilkan nilai *fitness* yang optimum disajikan dalam Tabel 3.2.

Tabel 3.2 Uji coba nilai λ (lambda)

Nilai λ (lambda)	Nilai <i>Fitness</i> pada <i>miu</i> -								Rata-rata <i>fitness</i>
	15	20	25	30	35	40	45	50	
2									
3									
4									
5									
6									
7									
8									
9									
10									



3.7.3 Uji coba ukuran populasi (miu)

Uji coba menentukan ukuran populasi dilakukan untuk mengetahui pengaruh ukuran populasi terhadap nilai *fitness* yang dihasilkan. Ukuran populasi yang akan diujikan adalah rentang 10-100. Nilai kisaran produksi maksimum yang digunakan adalah nilai terbaik pada uji coba kisaran produksi maksimum. Jumlah generasi sebanyak 100 generasi. Nilai λ (lambda) yang digunakan adalah nilai λ (lambda) terbaik pada uji coba nilai λ (lambda). Rancangan uji coba menentukan ukuran populasi untuk mendapatkan nilai *fitness* yang optimum disajikan pada Tabel 3.3.

Tabel 3.3 Uji coba ukuran populasi (miu)

ukuran populasi	Nilai <i>Fitness</i> pada percobaan ke-								Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	
10									
20									
30									
40									
50									
60									
70									
80									
90									
100									

3.7.4 Uji coba jumlah generasi

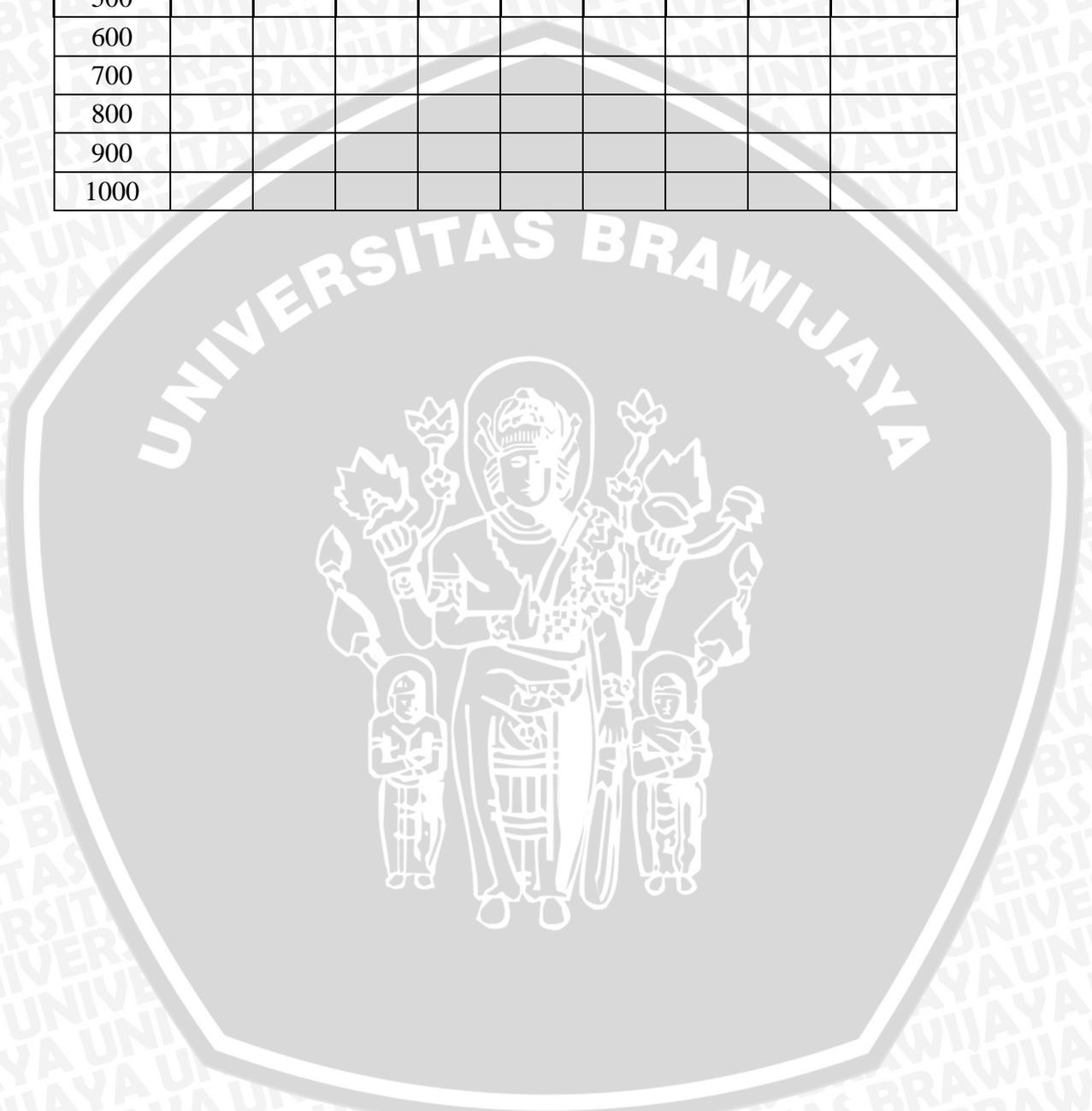
Uji coba menentukan jumlah generasi dilakukan untuk mengetahui jumlah generasi yang efektif dan efisien dalam menghasilkan nilai *fitness* yang optimum. Jumlah generasi yang digunakan dalam uji coba ini adalah rentang 100 sampai 1000. Nilai kisaran produksi maksimum yang digunakan adalah nilai terbaik pada uji coba kisaran produksi maksimum. Nilai λ (lambda) yang digunakan adalah nilai λ (lambda) terbaik pada uji coba nilai λ (lambda). Ukuran populasi yang digunakan adalah ukuran populasi terbaik pada uji coba ukuran populasi. Rancangan uji coba menentukan jumlah generasi untuk mendapatkan nilai *fitness* yang optimum disajikan pada Tabel 3.4.

Tabel 3.4 Uji coba jumlah generasi

Jumlah generasi	Nilai <i>Fitness</i> pada percobaan ke-								Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	
100									
200									
300									

Tabel 3.4 Uji coba jumlah generasi (lanjutan)

Jumlah generasi	Nilai <i>Fitness</i> pada percobaan ke-								Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	
400									
500									
600									
700									
800									
900									
1000									



BAB 4 PERANCANGAN

4.1 Formulasi permasalahan

Studi kasus yang dibahas pada penelitian ini adalah optimasi laba berdasarkan jumlah produksi pada Home Industry tekstil menggunakan algoritma *evolution strategies*. Pada penelitian ini, sampel data yang digunakan bersumber dari *home industry* tekstil. Data berupa macam-macam produk yang diproduksi, bahan baku yang digunakan, stok bahan baku, dan laba penjualan tiap produk. Rincian data sebagai berikut:

1. 20 macam produk tekstil.
2. Bahan baku tiap produk.
3. Stok bahan baku tiap bulan.
4. Laba penjualan tiap produk.

Dari data tersebut kemudian perlu disusun model matematis agar bisa dilakukan perhitungan untuk mencari nilai laba maksimal dengan menggunakan algoritma *Evolution Strategies*. Data bahan baku yang tersedia selama satu bulan disajikan pada Tabel 4.1.

Tabel 4.1 Bahan baku perbulan

Bahan baku	jumlah persediaan
luas kain(m ²)	2000
panjang Benang (m)	9282
karet(cm)	8000
resleting(cm)	9000
tali(cm)	6500
Silikon (kg)	200

Laba yang didapatkan dari penjualan per produk tekstil disajikan pada Tabel 4.2.

Tabel 4.2 Laba per produk

Produk	Laba (Rp)
Sarung bantal normal	7000
Sarung bantal kursi	15000
Sarung bantal cinta	8000
Sarung bantal santai	8000
Sarung bantal bayi	10000
Sarung guling bayi	10000
Sarung guling dewasa	7000

Tabel 4.2 Laba per produk (lanjutan)

Produk	Laba (Rp)
sprei 180	25000
sprei 160	20000
sprei 150	15000
sprei 120	10000
sprei 100	7000
sprei 90	5000
bantal normal	21000
bantal kursi	14000
bantal cinta	24000
bantal santai	24000
bantal bayi	11000
guling bayi	8000
guling dewasa	26000

Bahan baku yang dibutuhkan untuk pembuatan produk tekstil disajikan pada Tabel 4.3.

Tabel 4.3 Bahan baku produk

Produk	luas kain (m ²)	Benang (m)	Karet (cm)	Resleting (cm)	Tali (cm)	Silikon (kg)
Sarung bantal normal	0.629	3.867	0	50	0	0
Sarung bantal kursi	0.441	3.437	0	50	0	0
Sarung bantal cinta	0.911	4.297	0	50	0	0
Sarung bantal santai	0.897	4.297	0	70	0	0
Sarung bantal bayi	0.629	3.291	0	40	0	0
Sarung guling bayi	0.288	3.916	0	0	96	0
Sarung guling dewasa	0.75	4.990	0	0	150	0
sprei 180	3.96	8	200	0	0	0
sprei 160	3.52	7.6	180	0	0	0
sprei 150	3.3	7.4	160	0	0	0
sprei 120	2.64	6.8	140	0	0	0
sprei 100	2.2	6.4	120	0	0	0
sprei 90	1.98	6.2	120	0	0	0

Tabel 4.3 Bahan baku produk (lanjutan)

Produk	luas kain (m2)	Benang (m)	Karet (cm)	Resleting (cm)	Tali (cm)	Silikon (kg)
bantal normal	0.48	3.683	0	20	0	1
bantal kursi	0.32	3.291	0	25	0	0.65
bantal cinta	0.72	4.071	0	30	0	1.5
bantal santai	0.72	4.071	0	25	0	1.5
bantal bayi	0.24	3.157	0	15	0	0.25
guling bayi	0.24	3.727	0	10	0	0.6
guling dewasa	0.665	4.687	0	15	0	1.5

Dari data yang telah dipaparkan tersebut maka dapat disusun model matematis yang tersusun atas sejumlah fungsi tujuan dan sejumlah kendala. Fungsi tujuan merepresentasikan tujuan yang ingin dioptimalkan (maksimumkan atau minimumkan) (Mahmudy, 2013). Pada permasalahan ini, jika produksi tidak melebihi 600 produk maka fungsi tujuan diperoleh berdasarkan jumlah perkalian laba dengan kuantitas produk. Namun, jika jumlah produksi melebihi 600 produk maka fungsi tujuan diperoleh berdasarkan jumlah perkalian laba dengan kuantitas produk dikurangi upah lembur. Jika kuantitas tiap jenis produk dilambangkan dengan x_1 - x_{20} , maka fungsi tujuan dinyatakan sebagai berikut:

Maksimumkan

$$f(x_1, \dots, x_{20}) = \begin{cases} \sum_{i=1}^{20} \text{labax}_i \times x_i, & \text{jika produksi} \leq 600 \\ \sum_{i=1}^{20} \text{labax}_i \times x_i - 500 \times ((\sum_{i=1}^{20} x_i) - 600), & \text{jika produksi} > 600 \end{cases} \quad (4.1)$$

Misalkan terdapat dua puluh macam produk dengan jumlah produksi dan laba tiap produk yang ditampilkan pada Tabel 4.4.

Tabel 4.4 Contoh jumlah dan laba

Produk	Jumlah Produksi	Laba
x1	26	7000
x2	49	15000
x3	11	8000
x4	43	8000
x5	53	10000
x6	49	10000
x7	12	7000
x8	47	25000
x9	16	20000
x10	30	15000
x11	15	10000
x12	58	7000

Tabel 4.4 Contoh jumlah dan laba (lanjutan)

Produk	Jumlah Produksi	Laba
x13	24	5000
x14	27	21000
x15	26	14000
x16	24	24000
x17	50	24000
x18	58	11000
x19	33	8000
x20	26	26000
Total produksi	677	

Dari data tersebut maka akan dilakukan perkalian antara jumlah produk dengan laba yang didapatkan. Karena jumlah produksi adalah 677 (>600) maka laba total perlu dikurangi dengan biaya lembur. Hasil ditampilkan pada Tabel 4.5.

Tabel 4.5 Contoh hasil nilai fungsi tujuan

Produk	Labax _i × x _i
x1	182000
x2	735000
x3	88000
x4	344000
x5	530000
x6	490000
x7	84000
x8	1175000
x9	320000
x10	450000
x11	150000
x12	406000
x13	120000
x14	567000
x15	364000
x16	576000
x17	1200000
x18	638000
x19	264000
x20	676000
Total	9359000

Sehingga berdasarkan rumus fungsi tujuan yang telah dipaparkan sebelumnya, maka nilai fungsi tujuan yang didapatkan adalah

$$9359000 - (500 \cdot (677 - 600)) = 9320500.$$

Dalam proses produksi, jumlah bahan yang digunakan untuk memproduksi seluruh produk tidak boleh melebihi jumlah bahan baku yang tersedia. Apabila jumlah bahan baku yang digunakan melebihi bahan baku yang tersedia maka akan menjadi kendala. Kendala tersebut disesuaikan dengan macam-macam bahan baku yang tersedia. Sehingga terdapat kendala pada bahan baku yang dibutuhkan yaitu, kain (kendala 1), benang (kendala 2), karet (kendala 3), resleting (kendala 4), tali (kendala 5), dan silikon (kendala 6).

Selain kendala dalam hal bahan, terdapat kendala lain mengenai jumlah minimal untuk suatu produk yang harus diproduksi, karena selain untuk dijual barang kali terdapat produk pesanan atau pun untuk keperluan pribadi. Sehingga terdapat kendala untuk jumlah minimum (kendala 7). Beberapa kendala yang telah dipaparkan sebelumnya dinyatakan sebagai berikut:

Kendala 1:

$$0.629x_1 + 0.441x_2 + 0.911x_3 + 0.897x_4 + 0.629x_5 + 0.288x_6 + 0.75x_7 + 3.96x_8 + 3.52x_{10} + 3.3x_{11} + 2.64x_{12} + 2.2x_{12} + 1.98x_{13} + 0.48x_{14} + 0.32x_{15} + 0.72x_{16} + 0.72x_{17} + 0.24x_{18} + 0.24x_{19} + 0.665x_{20} \leq 2000 \quad (4.2)$$

Pada kendala satu, angka 0.629 merupakan luas kain dalam satuan m² (meter persegi) yang dibutuhkan untuk membuat produk x₁. Luas kain untuk produk x₁ tersebut akan dijumlahkan dengan luas kain yang digunakan untuk produk x₂-x₂₀. Jumlah kain yang digunakan tidak boleh melebihi stok kain yang tersedia yakni 2000 m².

Kendala 2:

$$3.867x_1 + 3.437x_2 + 4.297x_3 + 4.297x_4 + 3.291x_5 + 3.916x_6 + 4.99x_7 + 8x_8 + 7.6x_9 + 7.4x_{10} + 6.8x_{11} + 6.4x_{12} + 6.2x_{13} + 3.683x_{14} + 3.291x_{15} + 4.071x_{16} + 4.071x_{17} + 3.157x_{18} + 3.727x_{19} + 4.687x_{20} \leq 9282 \quad (4.3)$$

Pada kendala dua, angka 3.867 merupakan panjang benang dalam satuan m (meter) yang dibutuhkan untuk membuat produk x₁. Panjang benang untuk produk x₁ tersebut akan dijumlahkan dengan panjang benang yang digunakan untuk produk x₂-x₂₀. Jumlah benang yang digunakan tidak boleh melebihi stok benang yang tersedia yakni 9282 m.

Kendala 3:

$$200x_8 + 180x_9 + 160x_{10} + 140x_{11} + 120x_{12} + 120x_{13} \leq 8000 \quad (4.4)$$

Pada kendala tiga, angka 200 merupakan panjang karet dalam satuan cm (centimeter) yang dibutuhkan untuk membuat produk x₈. Panjang karet untuk produk x₈ tersebut akan dijumlahkan dengan panjang karet yang digunakan untuk produk x₉-x₁₃. Jumlah karet yang digunakan tidak boleh melebihi stok karet yang tersedia yakni 8000 cm.

Kendala 4:

$$250x_1 + 50x_2 + 50x_3 + 70x_4 + 40x_5 + 20x_{14} + 25x_{15} + 30x_{16} + 25x_{17} + 15x_{18} + 10x_{19} + 15x_{20} \leq 9000 \quad (4.5)$$

Pada kendala empat, angka 250 merupakan panjang resleting dalam satuan cm (centimeter) yang dibutuhkan untuk membuat produk x_1 . Luas kain untuk produk x_1 tersebut akan dijumlahkan dengan panjang resleting yang digunakan untuk produk x_2 - x_5 dan x_{14} - x_{20} . Jumlah resleting yang digunakan tidak boleh melebihi stok resleting yang tersedia yakni 9000 cm.

Kendala 5:

$$96x_6 + 150x_7 \leq 6500 \quad (4.6)$$

Pada kendala lima, angka 96 merupakan panjang tali dalam satuan cm (centimeter) yang dibutuhkan untuk membuat produk x_6 . Luas kain untuk produk x_6 tersebut akan dijumlahkan dengan panjang tali yang digunakan untuk produk x_7 . Jumlah tali yang digunakan tidak boleh melebihi stok tali yang tersedia yakni 6500 cm.

Kendala 6:

$$x_{14} + 0.65x_{15} + 1.5x_{16} + 1.5x_{17} + 0.25x_{18} + 0.6x_{19} + 1.5x_{20} \leq 200 \quad (4.7)$$

Pada kendala enam, bahan yang digunakan adalah silikon (satuan kilogram). Produk yang menggunakan silikon adalah produk x_{14} - x_{20} . Pada x_{14} berat silikon yang dibutuhkan untuk membuat produk x_{14} adalah sejumlah 1 kg. berat silikon untuk produk x_{14} tersebut akan dijumlahkan dengan berat silikon yang digunakan untuk produk x_{15} - x_{20} . Jumlah silikon yang digunakan tidak boleh melebihi stok silikon yang tersedia yakni 200 kg.

Kendala 7:

$$x_i \geq \text{minimal}(x_i) \quad (4.8)$$

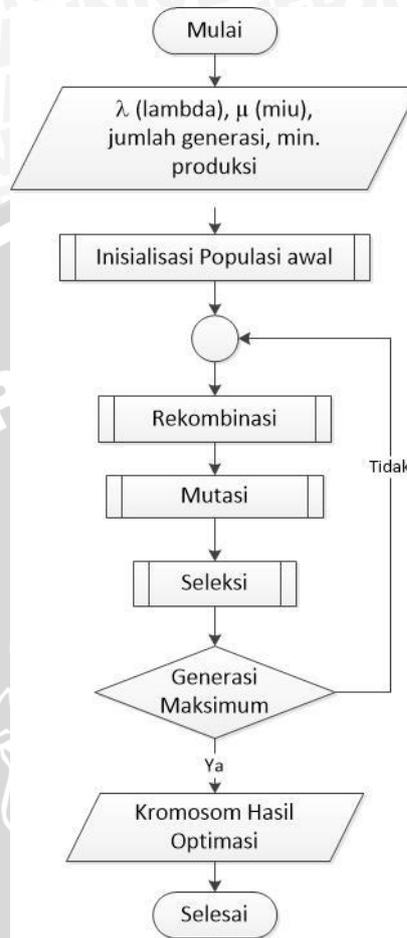
Pada kendala tujuh, jumlah produksi untuk suatu produk tidak boleh kurang dari jumlah minimal. Jumlah minimal diperoleh apabila ada pesanan untuk produk tertentu.

4.1.1 Asumsi permasalahan

Cara untuk mendapatkan laba yang maksimal selain berdasarkan jumlah produksi juga bisa dilakukan dengan mengkombinasikan perkiraan minat pasar. dalam melakukan perkiraan/peramalan minat pasar bisa menggunakan beberapa metode seperti peramalan *time series*. Namun, tidak tersedianya data dari narasumber yang dibutuhkan untuk peralaman minat pasar, maka dalam penelitian ini diasumsikan bahwa laba yang maksimal diperoleh dari penjualan produk berdasarkan jumlah produksi yang mampu menghasilkan laba maksimal.

4.2 Siklus algoritma *evolution strategies*

Siklus penyelesaian optimasi laba dengan menggunakan algoritma *evolution strategies* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Siklus algoritma *Evolution Strategies*

Pada algoritma *evolution strategies* yang digunakan dalam penelitian ini, langkah pertama adalah membangkitkan populasi awal secara random sesuai dengan ukuran populasi yang telah diinputkan. Selanjutnya melakukan reproduksi. Tipe ES yang digunakan adalah $(\mu/r + \lambda)$, maka pada tahap reproduksi dilakukan melalui dua proses yaitu rekombinasi dan mutasi. Kemudian melakukan seleksi menggunakan metode *elitism selection*. Pada tahap seleksi, dipilih individu terbaik sejumlah 75% dari μ (miu). Lalu pada tahap populasi baru dilakukan random injection sejumlah 25% dari μ (miu). Sehingga didapatkan n individu baru sejumlah μ dari hasil seleksi dan random injection. Individu baru yang telah terbentuk tersebut digunakan sebagai parent bagi generasi selanjutnya. Iterasi dilakukan hingga mencapai generasi maksimum. Hasil seleksi yang diperoleh pada generasi terakhir merupakan individu dengan kromosom terbaik.

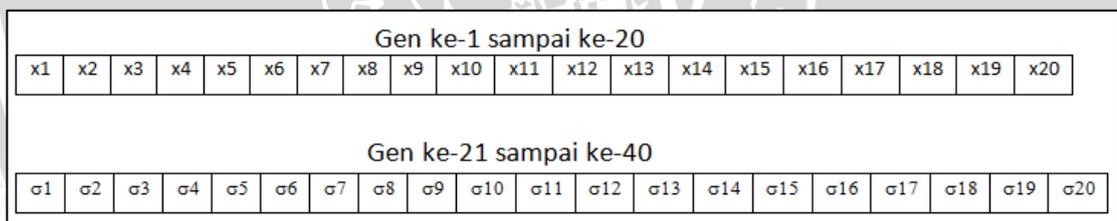
4.3 Penyelesaian masalah menggunakan Algoritma *Evolution strategies*

Berdasarkan formulasi permasalahan yang sudah dijelaskan pada subbab 4.1, maka pada subbab ini akan dibahas mengenai penyelesaian masalah optimasi laba berdasarkan jumlah produksi menggunakan algoritma *evolution strategies*. Parameter yang digunakan pada contoh studi kasus tersebut adalah sebagai berikut:

- a. λ (lambda) : 6
- b. μ (poSize) : 4
- c. Jumlah minimal : 5 untuk sarung bantal cinta, 0 untuk selainnya
- d. Rentang kromosom : 0-50

4.3.1 Representasi kromosom dan perhitungan *fitness*

Dalam merepresentasikan kromosom pada kasus optimasi laba ini, sesuai jumlah produk yang diproduksi maka variabel keputusan (x_1 - x_{20}) langsung menjadi gen string kromosom. Selain gen yang menyatakan variabel keputusan, parameter tambahan yang melekat pada setiap kromosom adalah σ (sigma). Nilai ini menyatakan level mutasi untuk kromosom tersebut. Nilai ini akan ikut berubah secara adaptif sepanjang generasi (Mahmudy, 2013). Kromosom akan dikodekan dalam bilangan *Real*. Panjang kromosom adalah empat puluh, yakni dua puluh gen pertama adalah x_1 - x_{20} dan dua puluh gen selanjutnya adalah σ_1 - σ_{20} . Berikut adalah struktur kromosom pada setiap individu yang ditampilkan pada Gambar 4.2.



Gambar 4.2 Struktur kromosom

Contoh representasi kromosom dapat dilihat pada Tabel 4.6.

Tabel 4.6 Representasi kromosom

P(t)	Kromosom
P1	[19.99, 41.38, 21.04, 19.86, 26.47, 17.95, 22.58, 4.23, 2.72, 17.71, 23.35, 4.78, 10.82, 32.87, 14.77, 26.84, 10.71, 13.68, 3.17, 27.67, 0.18, 0.55, 0.66, 0.29, 0.44, 0.94, 0.33, 0.48, 0.32, 0.40, 0.88, 0.20, 0.53, 0.46, 0.37, 0.48, 0.38, 0.52, 0.63, 0.27]

Pada panjang dua puluh gen pertama, nilai real yang ditampilkan merupakan jumlah produk yang akan diproduksi. Nilai real tersebut kemudian akan dikonversi kedalam bilangan bulat agar sesuai dengan solusi permasalahan.

Lalu pada pada panjang dua puluh gen selanjutnya merupakan nilai dari masing-masing σ (sigma) yang memiliki rentang nilai [0,1].

Langkah selanjutnya adalah menentukan rumus perhitungan *fitness*. Karena studi kasus pada penelitian ini merupakan permasalahan optimasi fungsi berkendala, maka terdapat aturan tertentu agar didapatkan rumus perhitungan *fitness* yang tepat. Terdapat beberapa aturan yang diadopsi dari (Mahmudy & Rahman 2011 dalam Mahmudy, 2013) untuk menentukan individu yang lebih baik sebagai berikut:

- a. Jika tidak ada kendala yang dilanggar maka sebuah individu dikatakan lebih baik dari individu yang lain jika nilai fungsi obyektifnya lebih besar (berlaku untuk masalah maksimasi).
- b. Jika kedua individu melanggar minimal satu kendala maka dipilih yang total pelanggaran terhadap kendala lebih kecil. Hal ini untuk menjamin solusi yang dipilih memenuhi kendala sebanyak mungkin.

Berdasarkan fungsi tujuan dan kendala yang telah dijelaskan pada subbab 4.1 dan mengacu pada dua aturan yang telah dipaparkan tersebut maka dapat ditentukan rumus perhitungan *fitness* sebagai berikut:

$$fitness(x_1, \dots, x_{20}) = f(x_1, \dots, x_{20}) - 2000 (c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7) \quad (4.9)$$

Dimana c_i adalah:

$$c_1 = \begin{cases} 0, & \text{jika } 0.629x_1 + \dots + 0.665x_{20} \leq 2000 \\ (0.629x_1 + \dots + 0.665x_{20}) - 2000, & \text{selainnya} \end{cases}$$

$$c_2 = \begin{cases} 0, & \text{jika } 3.867x_1 + \dots + 4.687x_{20} \leq 9282 \\ (3.867x_1 + \dots + 4.687x_{20}) - 9282, & \text{selainnya} \end{cases}$$

$$c_3 = \begin{cases} 0, & \text{jika } 200x_8 + \dots + 120x_{13} \leq 8000 \\ (200x_8 + \dots + 120x_{13}) - 8000, & \text{selainnya} \end{cases}$$

$$c_4 = \begin{cases} 0, & \text{jika } 250x_1 + \dots + 15x_{20} \leq 9000 \\ (250x_1 + \dots + 15x_{20}) - 6000, & \text{selainnya} \end{cases}$$

$$c_5 = \begin{cases} 0, & \text{jika } 96x_6 + 150x_7 \leq 6500 \\ (96x_6 + 150x_7) - 6500, & \text{selainnya} \end{cases}$$

$$c_6 = \begin{cases} 0, & \text{jika } x_{14} + \dots + 1.5x_{20} \leq 200 \\ (x_{14} + \dots + 1.5x_{20}) - 50, & \text{selainnya} \end{cases}$$

$$c_7 = 100 * \left(\sum_{i=0}^{20} x_i \mid x_i \begin{cases} 0, & \text{jika } x_i \geq \text{mimum}(x_i) \\ x_i - \text{minimum}(x_i), & \text{selainnya} \end{cases} \right)$$

Pada rumus perhitungan *fitness*, $f(x_1, \dots, x_{20})$ adalah fungsi tujuan yang merupakan hasil penjumlahan dari perkalian laba dan jumlah untuk masing-masing produk tekstil. Sedangkan $c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7$ adalah penalti yang merupakan penjumlahan dari masing-masing kendala. Lalu penalti tersebut akan dikalikan dengan angka sebesar 2000. Angka 2000 tersebut

merupakan angka acak yang memiliki nilai cukup besar agar mampu meminimalisasi kendala. Berikut adalah contoh hasil perhitungan fitness:

Tabel 4.7 Contoh hasil perhitungan fitness

Keterangan	Nilai
Kromosom	[19.99, 41.38, 21.04, 19.86, 26.47, 17.95, 22.58, 4.23, 2.72, 17.71, 23.35, 4.78, 10.82, 32.87, 14.77, 26.84, 10.71, 13.68, 3.17, 27.67, 0.18, 0.55, 0.66, 0.29, 0.44, 0.94, 0.33, 0.48, 0.32, 0.40, 0.88, 0.20, 0.53, 0.46, 0.37, 0.48, 0.38, 0.52, 0.63, 0.27]
Total laba semua produk	Rp 5155000
Jumlah total Produksi	364
Fungsi tujuan	Rp 5155000
Kendala 1	0
Kendala 2	0
Kendala 3	1360
Kendala 4	320
Kendala 5	0
Kendala 6	0
Kendala 7	0
Total kendala*2000	3360000
fitness	1795000

Pada contoh perhitungan fitness yang ditampilkan pada Tabel 4.7 didapatkan total laba semua produk adalah Rp 5155000. Kemudian akan dihitung jumlah total produk yang diproduksi. Didapatkan jumlah total produksi adalah 364 (tidak melebihi 600). Sehingga nilai fungsi tujuan adalah sama dengan total laba dari semua produk karena tidak perlu dikurangi biaya lembur. Terdapat kendala pada kendala 3 (karet) dan kendala 4 (Resleting) dengan nilai masing-masing sebesar 1360 dan 320. Nilai kendala tersebut dijumlahkan lalu dikali dengan angka 2000. Didapatkan nilai total kendala adalah 3360000. Sehingga nilai fitness yang didapatkan adalah nilai fungsi tujuan dikurangi dengan nilai total kendala yaitu $5155000 - 3360000 = 1795000$.

4.3.2 Inisialisasi populasi awal

Pada inisialisasi awal akan dibangkitkan individu secara random sebanyak μ . Contoh inisialisasi populasi awal ditunjukkan pada Tabel 4.8.

Tabel 4.8 Populasi awal

P(t)	Kromosom(x_1-x_{20})	fitness
P1	[39.95, 30.11, 17.86, 9.86, 19.10, 3.82, 15.59, 17.17, 13.35, 2.38, 0.35, 4.07, 5.58, 7.83, 25.86, 40.10, 5.10, 32.83, 11.64, 37.36, 0.33, 0.33, 0.51, 0.60, 0.57, 0.32, 0.34, 0.71, 0.24, 0.35, 0.07, 0.16, 0.18, 0.73, 0.50, 0.03, 0.59, 1.00, 0.12, 0.01, 0.01]	4772000
P2	[31.41, 29.44, 17.02, 11.82, 26.61, 4.37, 17.92, 18.81, 10.40, 6.36, 9.74, 3.89, 12.47, 13.39, 25.18, 38.83, 9.50, 34.51, 17.51, 38.95, 0.41, 0.50, 0.58, 0.62, 0.48, 0.41, 0.74, 0.22, 0.32, 0.12, 0.36, 0.38, 0.71, 0.38, 0.22, 0.47, 0.96, 0.22, 0.04, 0.02]	1151000
P3	[31.41, 29.70, 17.19, 11.92, 26.76, 4.99, 17.93, 18.95, 10.48, 6.33, 9.80, 3.99, 12.88, 13.86, 25.10, 38.91, 9.27, 34.46, 17.52, 38.95, 0.37, 0.45, 0.52, 0.56, 0.43, 0.37, 0.66, 0.20, 0.29, 0.11, 0.33, 0.34, 0.64, 0.34, 0.20, 0.42, 0.86, 0.20, 0.03, 0.02]	841000
P4	[47.21, 14.11, 38.03, 42.19, 23.50, 34.44, 16.74, 12.25, 42.69, 8.01, 48.19, 44.58, 26.90, 45.10, 6.11, 34.95, 34.30, 42.30, 16.12, 6.18, 0.79, 0.43, 0.64, 0.02, 0.50, 0.51, 0.17, 0.26, 0.47, 0.07, 0.85, 0.18, 0.91, 0.87, 0.74, 0.50, 0.92, 0.57, 0.54, 0.36]	-37401000

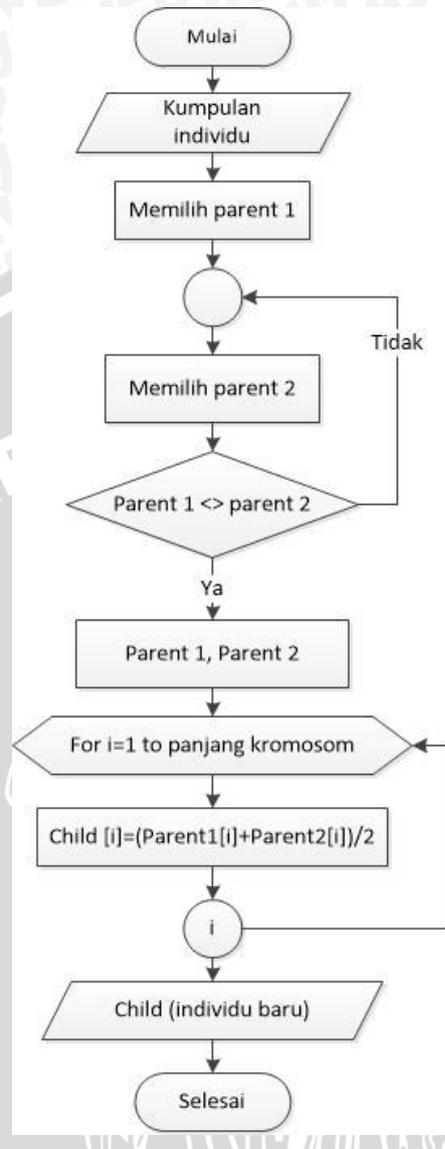
4.3.3 Reproduksi

Karena Tipe ES yang digunakan adalah $(\mu/r + \lambda)$, maka pada proses reproduksi ini digunakan metode rekombinasi dan mutasi. Rekombinasi dilakukan untuk menghasilkan offspring sebanyak $\lambda * \mu/2$ individu dalam populasi. Proses rekombinasi memiliki kemiripan dengan metode *crossover*, namun dalam metode rekombinasi hasil dari offspring akan digunakan sebagai induk (*parent*) untuk metode mutasi.

4.3.3.1 Rekombinasi

Pada proses rekombinasi, tiap individu dihasilkan dari beberapa induk (dua atau lebih). Semakin banyak jumlah induk yang digunakan pada setiap tahap rekombinasi tidak menjamin menghasilkan individu anak dengan nilai fitness yang lebih baik dan waktu yang dibutuhkan untuk komputasi semakin bertambah. Sehingga pada studi kasus ini hanya akan memilih sejumlah dua individu induk saja. Induk dipilih secara acak dari beberapa individu yang telah dibangkitkan/terseleksi sebelumnya. Metode rekombinasi yang digunakan pada penelitian ini adalah dengan menghitung nilai rata-rata dari kromosom induk.

Berikut ini adalah *Flowchart* proses rekombinasi untuk menghasilkan satu individu child yang ditunjukkan pada Gambar 4.3.



Gambar 4.3 Proses Rekombinasi

Misalkan induk yang terpilih adalah P1 dan P3 maka contoh hasil rekombinasi ditunjukkan pada Tabel 4.9

Tabel 4.9 Hasil rekombinasi

P(t)	Kromosom	<i>fitness</i>
P1	[39.95, 30.11, 17.86, 9.86, 19.10, 3.82, 15.59, 17.17, 13.35, 2.38, 0.35, 4.07, 5.58, 7.83, 25.86, 40.10, 5.10, 32.83, 11.64, 37.36, 0.33, 0.33, 0.51, 0.60, 0.57, 0.32, 0.34, 0.71, 0.24, 0.35, 0.07, 0.16, 0.18, 0.73, 0.50, 0.03, 0.59, 1.00, 0.12, 0.01, 0.01]	4772000
P3	[31.41, 29.70, 17.19, 11.92, 26.76, 4.99, 17.93, 18.95, 10.48, 6.33, 9.80, 3.99, 12.88, 13.86, 25.10, 38.91, 9.27, 34.46, 17.52, 38.95, 0.37, 0.45, 0.52, 0.56, 0.43, 0.37, 0.66, 0.20, 0.29, 0.11, 0.33, 0.34, 0.64, 0.34, 0.20, 0.42, 0.86, 0.20, 0.03, 0.02]	841000

Tabel 4.9 Hasil rekombinasi (lanjutan)

P(t)	Kromosom	fitness
C1	[35.68, 29.90, 17.52, 10.89, 22.93, 4.40, 16.76, 18.06, 11.92, 4.35, 5.07, 4.03, 9.23, 10.85, 25.48, 39.51, 7.18, 33.64, 14.58, 38.15, 0.35, 0.48, 0.56, 0.56, 0.38, 0.35, 0.69, 0.22, 0.32, 0.09, 0.24, 0.26, 0.68, 0.42, 0.11, 0.51, 0.93, 0.16, 0.02, 0.01]	3387000

Contoh hasil rekombinasi pada Tabel 4.9 bisa kita amati bahwa pada x_1 kromosom C1 dengan nilai 35.68 merupakan hasil dari rata-rata antara nilai x_1 pada kromosom P1 dan P2 yaitu 39.95 dan 31.41. Perhitungan rata-rata dilakukan sampai pada gen σ_{20} .

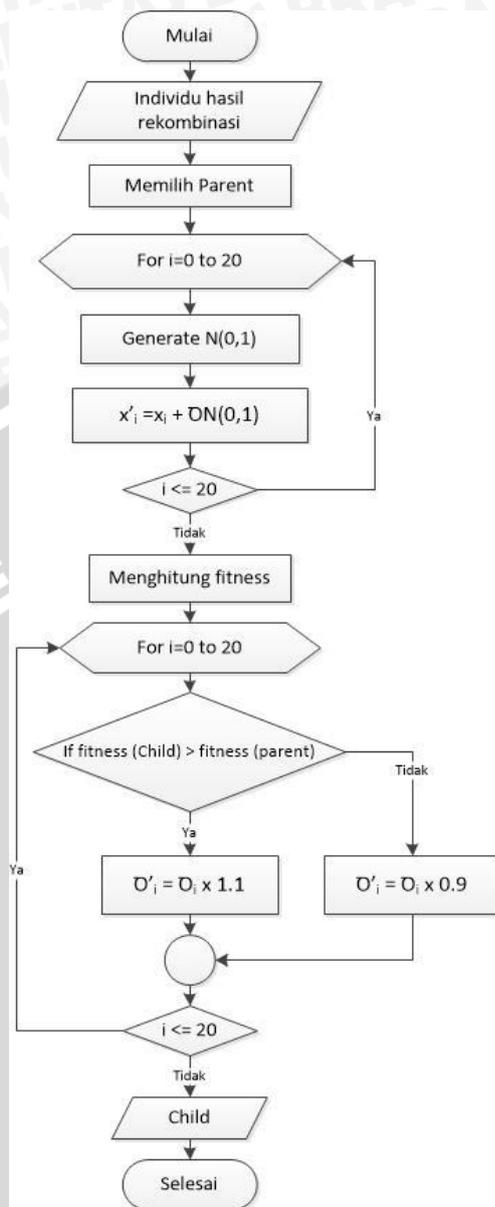
4.3.3.2 Mutasi

Metode mutasi pada algoritma *evolution strategies* berbeda dengan metode mutasi pada algoritma genetika karena nilai σ (Sigma) mempengaruhi penentuan nilai x . Dalam Mahmudy (2013) dijelaskan bahwa misalkan $P=(x_1, x_2, \sigma_1, \sigma_2)$ adalah individu yang terpilih untuk melakukan mutasi, maka dihasilkan offspring $P'=(x'_1, x'_2, \sigma'_1, \sigma'_2)$. x' didapatkan dengan cara menggunakan rumus pada persamaan (2-1).

Dalam tipe ES ($\mu/r + \lambda$) nilai σ untuk *Child* didapatkan melalui perbandingan nilai *fitness* Child dengan Parent. Rumus menentukan σ' untuk child dapat dilihat pada persamaan 4.10.

$$\sigma' = \begin{cases} \sigma \times 1.1, & \text{jika } fitness(Child) > fitness(Parent) \\ \sigma \times 0.9, & \text{jika } fitness(Child) \leq fitness(Parent) \end{cases} \quad (4.10)$$

Apabila nilai *fitness* child lebih tinggi daripada nilai *fitness* parent, maka nilai sigma dikalikan dengan 1.1. Namun jika nilai *fitness* child lebih rendah daripada nilai *fitness* parent, maka sigma dikali dengan 0.9. Jumlah individu yang dihasilkan dalam proses mutasi adalah sama dengan jumlah individu yang terdapat pada rekombinasi. Sehingga misalkan pada proses rekombinasi terdapat individu C1, C2, C3, dan C4 maka pada proses mutasi akan dihasilkan individu C5, C6, C7, dan C8. C5 dibangkitkan dari proses mutasi individu C1, C6 dibangkitkan dari proses mutasi individu C2 dan hal ini berlaku untuk semua individu pada proses mutasi sesuai urutan individu yang akan dihasilkan. Berikut adalah flowchart proses mutasi untuk menghasilkan satu individu child ditampilkan pada Gambar 4.4.



Gambar 4.4 Proses mutasi

Misalkan pada induk C1, maka akan dihasilkan individu C5. Pada tahap awal mutasi, akan dibangkitkan nilai $N(0,1)$ secara random untuk masing-masing gen kromosom. Berikut adalah nilai $N(0,1)$ untuk individu C5 yang ditunjukkan pada Tabel 4.10.

Tabel 4.10 Nilai $N(0,1)$ pada individu C5

P(t)	Nilai $N(0,1)$
C5	[0.5608, 0.3878, -0.4889, -0.0125, 0.5646, 0.6069, 0.2906, 0.0530, 0.6261, 0.3210, 0.4187, 0.5599, 0.5623, 0.0565, 0.0731, 1.0850, 0.3587, 0.5000, 0.3294, 0.2555]

Dari nilai gen pada individu C1 dan nilai $N(0,1)$ yang telah dihasilkan, maka akan dibangkitkan nilai gen-gen untuk individu C5. Misalkan nilai pada gen kesatu, maka nilainya adalah sebagai berikut:

$$\begin{aligned} x_1' &= x_1 + \sigma_1 N_1(0,1) \\ &= 35.68 + 0.35 * 0.5608 \\ &= 35.87628 \end{aligned}$$

Angka 35.68 merupakan nilai gen kesatu pada individu C1, 0.35 merupakan nilai sigma kesatu pada individu C1, dan 0.5608 merupakan nilai $N(0,1)$ pada individu C5. Lalu dengan cara perhitungan menggunakan persamaan (2-1) didapatkan nilai gen kesatu individu C5 adalah 35.87628. Kemudian setelah didapatkan semua nilai gen pada individu C5 dan dihitung nilai fitnessnya, selanjutnya adalah menghitung nilai sigma. Jika nilai fitness child (C5) lebih besar daripada nilai fitness parent (C1) maka sigma child didapatkan dengan cara mengalikan sigma parent dengan 1.1, jika nilai fitness child (C5) lebih kecil daripada nilai fitness parent (C1) maka sigma child didapatkan dengan cara mengalikan sigma parent dengan 0.9. Berikut adalah contoh hasil proses mutasi ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil mutasi

P(t)	Kromosom	fitness
C1	[35.68, 29.9, 17.52, 10.89, 22.93, 4.4, 16.76, 18.06, 11.92, 4.35, 5.07, 4.03, 9.23, 10.85, 25.48, 39.51, 7.18, 33.64, 14.58, 38.15, 0.35, 0.48, 0.56, 0.56, 0.38, 0.35, 0.69, 0.22, 0.32, 0.09, 0.24, 0.26, 0.68, 0.42, 0.11, 0.51, 0.93, 0.16, 0.02, 0.01]	3387000
C5	[35.87, 30.09, 17.25, 10.88, 23.14, 4.62, 16.96, 18.07, 12.12, 4.38, 5.18, 4.18, 9.61, 10.87, 25.49, 40.06, 7.52, 33.72, 14.59, 38.15, 0.31, 0.43, 0.50, 0.51, 0.34, 0.32, 0.62, 0.20, 0.29, 0.08, 0.22, 0.23, 0.61, 0.38, 0.10, 0.46, 0.84, 0.14, 0.02, 0.01]	3228000

Pada hasil mutasi tersebut dapat lihat bahwa nilai *fitness* pada Child lebih kecil dari pada induk, maka sigma child didapatkan dengan cara $\sigma' = \sigma \times 0.9$. Sehingga pada sigma kesatu didapatkan dengan cara berikut:

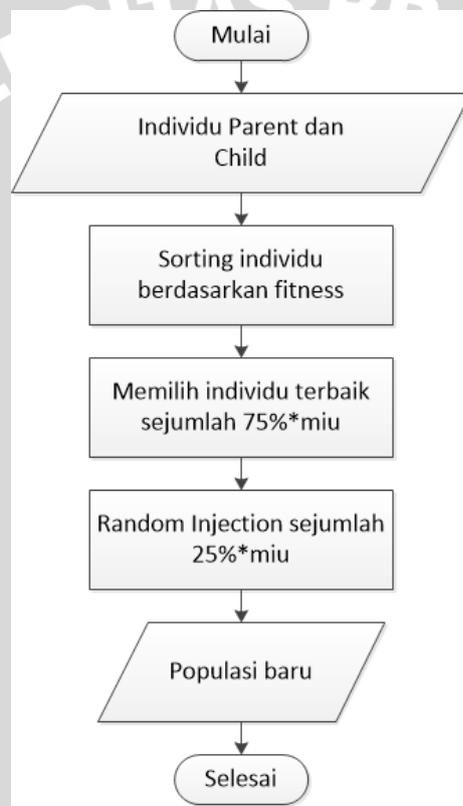
$$\begin{aligned} \sigma_1' &= \sigma_1 * 0.9 \\ &= 0.35 * 0.9 \\ &= 0.315 \end{aligned}$$

Angka 0.35 merupakan nilai sigma kesatu pada individu C1, sehingga setelah dikali dengan 0.9 maka didapatkan nilai sigma kesatu pada individu C5 adalah 0.315.



4.3.4 Seleksi

Seleksi yang digunakan adalah dengan metode *elitism selection*. Seleksi melibatkan individu induk (Parent) dan anak (Child). Pertama, individu induk yang telah dibangkitkan pada tahap inialisasi dan individu anak yang telah dihasilkan pada proses rekombinasi dan proses mutasi akan diurutkan secara *descending* berdasarkan nilai *fitness* masing-masing. Kemudian akan dipilih individu dengan nilai *fitness* terbaik sejumlah $75\% \cdot \mu$. Kemudian akan dilakukan random injection sejumlah $25\% \cdot \mu$ individu agar memenuhi individu sejumlah μ yang akan digunakan untuk generasi berikutnya. Metode random injection digunakan agar terjadi keragaman populasi sehingga mengurangi resiko terjadinya konvergensi dini. Flowchart untuk proses elitism selection dan random injection ditunjukkan pada Gambar 4.5.



Gambar 4.5 Proses seleksi

Berikut adalah semua individu parent dan child yang telah diurutkan secara *descending* ditunjukkan pada Tabel 4.12.

Tabel 4.12 Hasil pengurutan

P(t)	Fitness
P1	4772000
C2	3479000
C3	3387000
C4	3387000

Tabel 4.12 Hasil pengurutan (lanjutan)

P(t)	Fitness
C6	3299000
C7	3228000
C8	3109000
P2	1151000
P3	841000
C5	-17943000
C1	-18554000
P4	-37401000

Dari hasil seleksi pada Tabel 4.12 maka akan diambil individu sejumlah $75\% \cdot \mu$ yaitu $75\% \cdot 4 = 3$ individu. Lalu akan dilakukan random injection sejumlah $25\% \cdot \mu$ yaitu $25\% \cdot 4 = 1$ individu. Sehingga didapatkan populasi baru dari proses seleksi sebagai berikut:

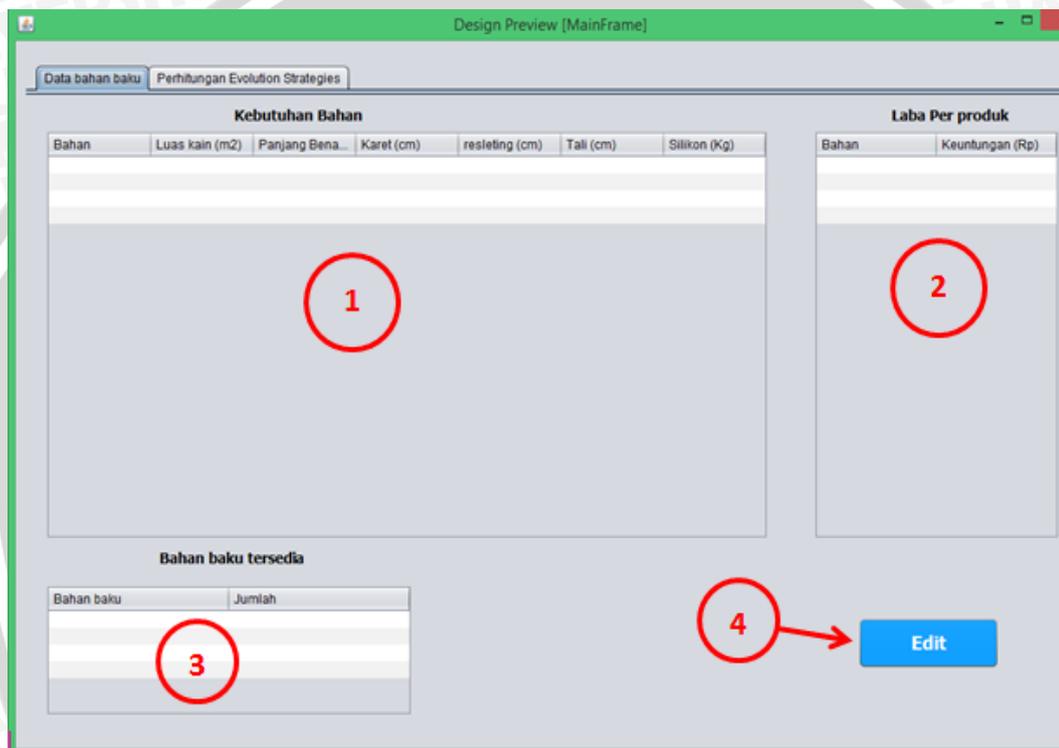
Tabel 4.13 Individu hasil seleksi

P(t+1)	P(t)	Kromosom	fitness
P1	P1	[39.95, 30.11, 17.86, 9.86, 19.10, 3.82, 15.59, 17.17, 13.35, 2.38, 0.35, 4.07, 5.58, 7.83, 25.86, 40.10, 5.10, 32.83, 11.64, 37.36, 0.33, 0.51, 0.60, 0.57, 0.32, 0.34, 0.71, 0.24, 0.35, 0.07, 0.16, 0.18, 0.73, 0.50, 0.03, 0.59, 1.00, 0.12, 0.01, 0.01]	4772000
P2	C2	[35.68, 29.77, 17.44, 10.84, 22.85, 4.09, 16.75, 17.99, 11.88, 4.37, 5.04, 3.98, 9.02, 10.61, 25.52, 39.47, 7.30, 33.67, 14.58, 38.15, 0.37, 0.51, 0.59, 0.59, 0.40, 0.37, 0.73, 0.23, 0.34, 0.09, 0.26, 0.28, 0.72, 0.44, 0.12, 0.53, 0.98, 0.17, 0.02, 0.01]	3479000
P3	C3	[35.68, 29.90, 17.52, 10.89, 22.93, 4.40, 16.76, 18.06, 11.92, 4.35, 5.07, 4.03, 9.23, 10.85, 25.48, 39.51, 7.18, 33.64, 14.58, 38.15, 0.35, 0.48, 0.56, 0.56, 0.38, 0.35, 0.69, 0.22, 0.32, 0.09, 0.24, 0.26, 0.68, 0.42, 0.11, 0.51, 0.93, 0.16, 0.02, 0.01]	3387000
P4	Random Injection	[31.51, 3.20, 40.16, 43.34, 24.31, 44.95, 30.81, 37.24, 48.70, 46.17, 3.24, 44.82, 35.67, 0.71, 47.29, 2.48, 10.09, 40.00, 46.82, 15.11, 0.01, 0.08, 1.00, 0.25, 0.64, 0.52, 0.20, 0.32, 0.59, 0.20, 0.52, 0.07, 0.72, 0.59, 0.96, 0.66, 0.42, 0.87, 0.14, 0.07]	-52287000

Pada Tabel 4.13 individu keempat (baris ke-4) merupakan hasil dari random injection. Dapat kita amati bahwa dari individu tersebut memiliki perbedaan nilai *fitness* yang cukup jauh. Maka dengan adanya individu hasil dari random injection mampu menambah keragaman populasi.

4.4 Perancangan user interface

User interface pada aplikasi ini terdiri dari dua halaman. Halaman pertama adalah halaman data bahan baku berfungsi untuk menginput jumlah bahan yang digunakan, keuntungan per produk, dan jumlah bahan yang tersedia. Halaman kedua adalah halaman untuk melakukan perubahan data yang muncul jika tombol “Edit” ditekan. Perubahan data bisa dilakukan pada kebutuhan bahan untuk produksi suatu produk, laba per produk, dan juga pada stoc bahan baku yang tersedia. Halaman ketiga yakni halaman perhitungan *Evolution strategies* berfungsi untuk pengujian *evolution strategies*. Perancangan tampilan untuk halaman data bahan baku pada aplikasi ditampilkan pada Gambar 4.6.

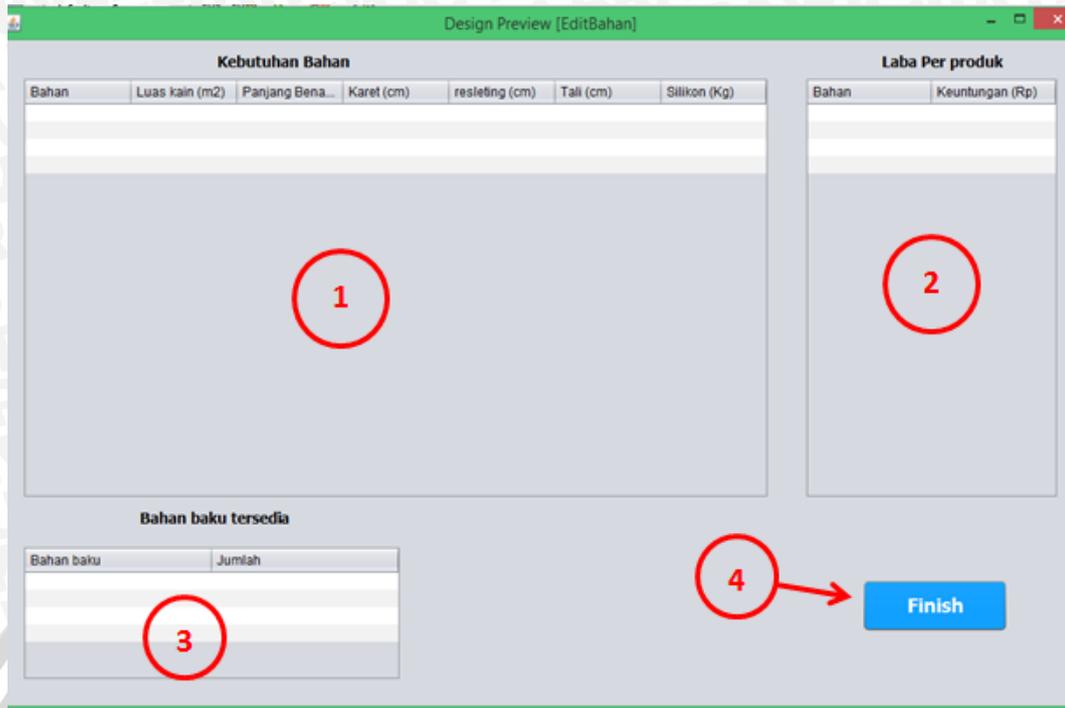


Gambar 4.6 Rancangan halaman bahan baku

Keterangan:

1. Tabel data jumlah bahan baku yang diperlukan untuk pembuatan produk.
2. Tabel data laba yang didapatkan untuk penjualan perproduk.
3. Tabel data jumlah bahan baku yang tersedia.
4. Tombol update apabila terjadi perubahan pada data tabel.

Perancangan tampilan untuk halaman perubahan data ditampilkan pada Gambar 4.7.

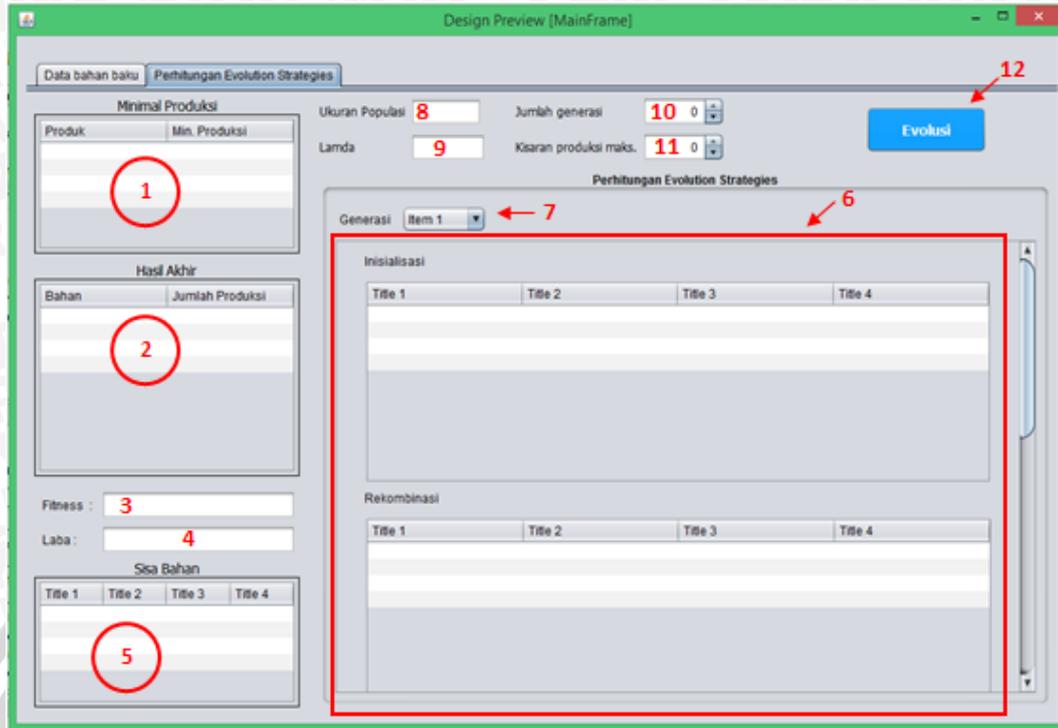


Gambar 4.7 Rancangan halaman perubahan data

Keterangan:

1. Tabel edit data jumlah bahan baku untuk pembuatan produk.
2. Tabel edit data laba yang didapatkan untuk penjualan perproduk.
3. Tabel edit data jumlah bahan baku yang tersedia.
4. Tombol finish apabila telah selesai melakukan perubahan pada data.

Perancangan tampilan untuk halaman perhitungan *evolution strategies* ditampilkan pada Gambar 4.8.



Gambar 4.8 Halaman perhitungan *evolution strategies*

Keterangan:

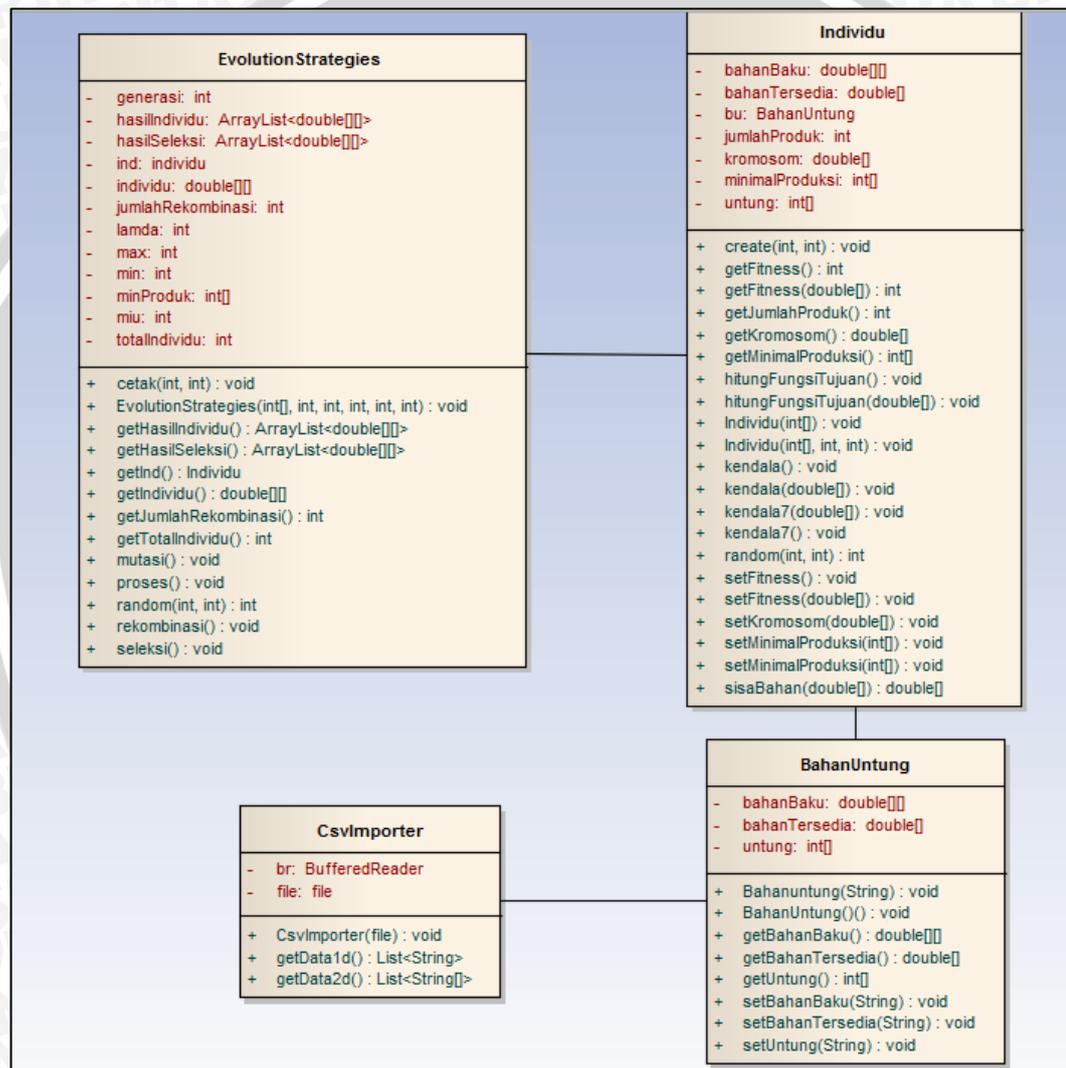
1. Tabel jumlah minimal produk yang harus diproduksi sesuai yang masukan user.
2. Tabel Hasil akhir berupa jumlah produk yang disarankan berdasarkan perhitungan algoritma *evolution strategies*.
3. Nilai fitness yang didapatkan berdasarkan perhitungan algoritma *evolution strategies*.
4. Laba yang didapatkan berdasarkan perhitungan algoritma *evolution strategies*.
5. Jumlah bahan yang tersisa setelah digunakan untuk memproduksi sejumlah produk yang ditampilkan pada produk akhir.
6. Panel untuk memberikan informasi mengenai perhitungan *evolution strategies* berdasarkan urutan generasi yang dipilih.
7. Spinner untuk menentukan generasi yang akan ditampilkan pada panel.
8. Textfield untuk menginput ukuran populasi.
9. Textfield untuk menginput nilai lamda.
10. Textfield untuk menginput jumlah generasi.
11. Nilai kisaran maksimal jumlah produk yang diproduksi.
12. Tombol untuk menjalankan proses komputasi Algoritma *Evolution Strategies*.

BAB 5 IMPLEMENTASI KELAS

Bab implementasi membahas tentang implementasi perangkat lunak berdasarkan metodologi penelitian dan perancangan. Implementasi terdiri dari Struktur Class dan Implementasi algoritma.

5.1 Struktur Class

Struktur Class utama dari perangkat lunak yang telah diimplementasikan dapat dilihat pada Gambar 5.1.



Gambar 5.1 Struktur Class

Penjelasan Struktur Class pada Gambar 5.1 disajikan pada Tabel 5.1

Tabel 5.1 Keterangan Struktur Class

Class	Keterangan
CsvImporter	Class CsvImporter berfungsi untuk mengimport file yang memiliki format “.csv” kedalam List. Method getData1d berfungsi untuk mengimport data “.csv” yang berdimensi satu. Method getData2d berfungsi untuk mengimport data “.csv” yang berdimensi dua. Class ini digunakan untuk mengimport data bahan baku yang tersedia, bahan baku untuk produksi produk dan laba produk.
BahanUntung	Class BahanUntung berfungsi sebagai berikut: <ol style="list-style-type: none"> Menetapkan jumlah bahan yang dibutuhkan untuk pembuatan per produk berdasarkan database bahan baku. Menetapkan jumlah stok bahan baku yang tersedia untuk masing-masing bahan berdasarkan database bahanTersedia. Menetapkan nilai laba yang didapatkan untuk penjualan per produk berdasarkan database keuntungan.
Individu	Class Individu merupakan Class yang memiliki method-method untuk menghitung variabel-variabel yang dibutuhkan setiap individu antara lain method untuk menghitung nilai fungsi tujuan, menghitung kendala, menghitung nilai fitness, dan menghitung sisa bahan habis pakai.
EvolutionStrategies	Class EvolutionStrategies merupakan kelas utama dalam algoritma <i>Evolution Strategies</i> . Pada Class tersebut terdapat method untuk melakukan proses inialisasi, rekombinasi, mutasi dan seleksi.

5.2 Implementasi algoritma

Pada proses implementasi algoritma menjelaskan tentang implementasi algoritma Evolution Strategies yang terdiri atas algoritma inialisasi, fungsi tujuan, kendala, fitness, rekombinasi, mutasi, dan seleksi.

5.2.1 Inialisasi

Pada proses inialisasi dilakukan pembangkitan individu sejumlah μ . Nilai kromosom pada individu diperoleh dengan cara membangkitkan bilangan random pada interval min sampai max yang menyatakan rentang jumlah

produksi sesuai dengan nilai yang diinputkan. Algoritma proses inialisasi dapat dilihat pada Source Code 5.1.

```

1 public void create(int min, int max) {
2     for (int i = 0; i < popsize; i++) {
3         for (int i = 0; i < 20; i++) {
4             kromosom[i]=random(min,max)+
5             Math.round(random(1,999))/
6             Math.pow(10,3);
7
8             kromosom[i+20]=Math.round
9             (random(1,999)) / Math.pow(10, 3);
10        }
11        individu[i] = kromosom;
12    }
13 }

```

Source Code 5.1 Inialisasi

Penjelasan algoritma inialisasi pada Source Code 5.1. adalah sebagai berikut:

1. Pada baris ke-4 hingga ke-9 merupakan pembangkitan bilangan random ke dalam kromosom.
2. Baris ke-11 menginputkan kromosom yang telah dibuat kedalam individu.

5.2.2 Fungsi tujuan

Pada proses perhitungan fungsi tujuan jika jumlah kumulatif produksi semua produk tidak melebihi 600 maka fungsi tujuan didapatkan melalui penjumlahan dari kuantitas produksi dikali dengan laba per produk. Namun jika jumlah kumulatif produksi semua produk melebihi 600 maka fungsi tujuan didapatkan melalui penjumlahan dari kuantitas produksi dikali dengan laba per produk dikurangi dengan jumlah produksi dikurangi 600 yang dikalikan dengan 500. Angka 500 diasumsikan sebagai biaya lembur untuk pembuatan per produk. Algoritma fungsi tujuan dapat dilihat pada Source Code 5.2.

```

1 public void hitungFungsiTujuan() {
2     int temp = 0;
3     if (getJumlahProduk() <= 600) {
4         for (int i = 0; i < 20; i++) {
5             temp+=Math.round(kromosom[i])*untung[i];
6         }
7     } else {
8         for (int i = 0; i < 20; i++) {
9             temp+= Math.round(kromosom[i])*untung[i];
10        }
11        temp=temp - (500 *(getJumlahProduk() - 600));
12    }
13    kromosom[40] = temp;
14 }

```

Source Code 5.2 Fungsi Kendala

Penjelasan algoritma fungsi tujuan pada Source Code 5.2 adalah sebagai berikut:

1. Baris ke-3 merupakan pengecekan apabila jumlah produk melebihi 600.

2. Baris ke-4 sampai ke-5 melakukan perhitungan nilai fungsi tujuan jika jumlah produksi tidak melebihi 600.
3. Baris ke-8 sampai ke-11 menghitung nilai fungsi tujuan jika jumlah produksi melebihi 600.

5.2.3 Kendala

Pada proses kendala terdapat 7 macam kendala. Kendala 1 sampai 6 merupakan kendala dalam pemakaian bahan. Kendala 7 merupakan kendala dalam memproduksi produk sesuai dengan jumlah minimal produk yang harus diproduksi. Algoritma kendala dapat dilihat pada Source Code 5.3.

```

1 public void kendala() {
2 // Perhitungan kendala 1-6
3   for (int i = 0; i < 6; i++) {
4     double temp = 0;
5     for (int j = 0; j < 20; j++) {
6       temp+= Math.round(kromosom[j])*bahanBaku[j][i];
7     }
8     if (temp <= bahanTersedia[i]) {
9       kromosom[41 + i] = 0;
10    } else {
11      kromosom[41 + i] = temp - bahanTersedia[i];
12    }
13  }
14
15 // Perhitungan kendala 7
16   int temp = 0;
17   for (int i = 0; i < 20; i++) {
18     int tamp;
19     if(Math.round(kromosom[i])>= minimalProduksi[i]) {
20       tamp = 0;
21     }else{
22       tamp=(int) (minimalProduksi[i]-
23         Math.round(kromosom[i]));
24     }
25     temp += tamp;
26   }
27   kromosom[47] = 100 * temp;
28 }

```

Source Code 5.3 Kendala

Penjelasan algoritma kendala pada Source Code 5.3 adalah sebagai berikut:

1. Baris 5-6 menghitung jumlah bahan habis pakai untuk memproduksi produk.
2. Baris 8-11 melakukan pengecekan apakah bahan habis pakai melebihi stok bahan baku yang tersedia. Jika bahan habis pakai tidak melebihi stok maka kendala bernilai 0. Jika bahan habis pakai melebihi stok maka kendala bernilai jumlah bahan habis pakai dikurangi stok bahan yang tersedia.
3. Baris 16-26 melakukan proses perhitungan kendala 7.
4. Baris 19 melakukan pengecekan apakah kuantitas produk yang diproduksi telah memenuhi syarat minimal produksi. Jika kuantitas produksi memenuhi

syarat minimal produksi, maka nilai kendala 7 pada produk tersebut bernilai 0. Jika kuantitas produksi tidak memenuhi syarat minimal produksi, maka nilai kendala 7 pada produk tersebut bernilai syarat jumlah minimal dikurangi kuantitas produk.

- Baris 27 menjumlahkan nilai kendala 7 tiap produk kemudian dikalikan 100.

5.2.4 Fitness

Proses perhitungan fitness diperoleh dengan cara nilai fungsi tujuan dikurangi kendala. Algoritma fitness dapat dilihat pada Source Code 5.4.

```

1 public void setFitness() {
2     double temp = 0;
3     for (int i = 0; i < 7; i++) {
4         temp += kromosom[41 + i];
5     }
6     kromosom[48]=(int) (kromosom[40]-(2000 * temp));
7 }

```

Source Code 5.4 Fitness

Penjelasan algoritma kendala pada Source Code 5.4 adalah sebagai berikut:

- Baris ke-4 melakukan penjumlahan kendala mulai dari kendala 1 hingga kendala 7.
- Baris ke-6 menghitung nilai fitness dengan cara nilai fungsi tujuan dikurangkan dengan jumlah kendala yang dikali dengan 2000.

5.2.5 Rekombinasi

Pada proses Rekombinasi dihasilkan offspring sebanyak λ dikali μ dibagi 2 individu dalam populasi. Tiap individu dihasilkan dari dua induk. Induk dipilih secara acak dari beberapa individu yang telah dibangkitkan/terseleksi sebelumnya. Nilai kromosom rekombinasi diperoleh dengan cara menghitung nilai rata-rata dari kromosom induk. Algoritma Rekombinasi dapat dilihat pada Source Code 5.5.

```

1 public void rekombinasi() throws IOException {
2     for (int i =0; i < jumlahRekombinasi; i++) {
3         double[] temp = new double[49];
4         int[] p = new int[2];
5         p[0] = random(0, popsize);
6         do {
7             p[1] = random(0, popsize);
8         } while (p[1] == p[0]);
9         for (int a = 0; a<40; a++) {
10            temp[a]=(individu[p[0]][a]+individu[p[1]][a])/2;
11        }
12        ind.setKromosom(temp);
13        individu[popsize + i] = temp;
14    }
15 }
16 }

```

Source Code 5.5 Rekombinasi

Penjelasan algoritma rekombinasi pada Source Code 5.5 adalah sebagai berikut:

1. Baris ke-5 sampai baris ke-8 digunakan untuk memilih induk P1 dan P2.
2. Baris ke-10 melakukan perhitungan rata-rata kromosom setiap gen dari individu P1 dan P2.
3. Baris ke-12 melakukan perhitungan fitness dengan cara mengakses method setKromosom pada class ind.
4. Baris ke-13 menginputkan individu yang telah dibuat kedalam himpunan individu rekombinasi.

5.2.6 Mutasi

Pada proses mutasi dihasilkan individu sejumlah individu yang terdapat pada proses rekombinasi. Algoritma Mutasi dapat dilihat pada Source Code 5.6.

```

1 public void mutasi() throws IOException {
2     double r1, r2, N;
3     double[] temp;
4     for(int i=popsize;i<(popsize+jumlahRekombinasi);i++){
5         temp = new double[49];
6         for (int j = 0; j < 20; j++) {
7             r1=Math.round(random(1,1000))/Math.pow(10, 3);
8             r2=Math.round(random(1,1000))/Math.pow(10,3);
9             N=Math.sqrt(-2*(Math.log10(r1)))*(Math.sin(2
10                *Math.PI * r2));
11             if (N < -0.5) {
12                 N *= -0.5;
13             }
14             temp[j]=(individu[i][j]+(individu[i][j+20]*N));
15         }
16         ind.setKromosom(temp);
17
18         if(ind.getFitness(temp)>ind.getFitness(individu[i])){
19             for (int j = 0; j < 20; j++) {
20                 temp[j + 20] = individu[i][j + 20] * 1.1;
21             }
22         }
23         else {
24             for (int j = 0; j < 20; j++) {
25                 temp[j + 20] = individu[i][j + 20] * 0.9;
26             }
27         }
28
29         double tt = Math.random() * 1;
30
31         for (int j = 0; j < 20; j++) {
32             if (temp[j + 20] > 1) {
33                 temp[j + 20] = tt;
34             }
35         }
36     }

```

```

37     individu[i + jumlahRekombinasi] = temp;
38     }
39 }

```

Source Code 5.6 Mutasi

Penjelasan algoritma mutasi pada Source Code 5.6 adalah sebagai berikut:

1. Baris ke-7 dan ke-8 membangkitkan nilai r1 dan r2.
2. Baris ke-9 membangkitkan nilai N.
3. Baris ke-12 akan mengubah nilai N jika nilai N dibawah -0.5. Hal ini dilakukan agar nilai sigma kromosom tidak bernilai dibawah -0.5. Sehingga bisa memperkecil kemungkinan nilai gen dibawah 0.
4. Baris ke-14 melakukan perhitungan nilai gen kromosom untuk individu child.
5. Baris ke-16 melakukan perhitungan fitness dengan cara mengakses method setKromosom pada class ind.
6. Baris ke-18 melakukan pengecekan apakah nilai fitness individu mutasi ke-i lebih besar dari nilai fitness individu rekombinasi ke-i.
7. Baris ke-19 sampai baris ke-26 membangkitkan nilai sigma mulai dari sigma ke-1 hingga sigma ke-20 untuk individu yang dihasilkan pada proses mutasi. Nilai sigma didapatkan dari perkalian nilai sigma induk dikali dengan 1.1 jika fitness anak lebih besar dibanding nilai fitness induk, dan dikali 0.9 jika fitness anak lebih kecil dibanding nilai fitness induk.
8. Baris ke-31 melakukan perbaikan nilai pada sigma apabila nilai sigma melebihi 1. Jika melebihi 1, maka nilai sigma tersebut akan diganti dengan suatu nilai random dengan rentang 0-1. Hal ini karena nilai sigma adalah rentang 0-1.
9. Baris ke-36 menginputkan individu yang telah dibuat kedalam himpunan individu mutasi.

5.2.7 Seleksi

Pada proses seleksi melibatkan individu parent dan child. Pertama, individu induk yang telah diinisialisasi dan individu child yang telah dihasilkan pada proses rekombinasi dan proses mutasi akan diurutkan secara *descending* berdasarkan nilai *fitness*. Kemudian akan dipilih individu dengan nilai *fitness* terbaik sejumlah $75\% \cdot \mu$. Kemudian akan dilakukan random injection sejumlah $25\% \cdot \mu$ individu agar memenuhi individu sejumlah μ yang akan digunakan untuk generasi berikutnya. Algoritma seleksi dapat dilihat pada Source Code 5.7.

```

1     public void Seleksi() throws IOException {
2         Individu baru;
3         double[] temp = new double[49];
4         for (int i = 0; i < totalIndividu; i++) {
5             int maks = i;
6             for (int j = i + 1; j < totalIndividu; j++) {
7                 if (individu[maks][48] < individu[j][48]) {
8                     maks = j;
9                 }
10            }
11            temp = individu[i];

```

```
12     individu[i] = individu[maks];
13     individu[maks] = temp;
14 }
15
16     int jum = (int) Math.round(0.25 * popsize);
17     int jumlah = popsize - jum;
18
19     for (int i = totalIndividu; i > jumlah; i--) {
20         individu[jumlah] = null;
21     }
22
23     for (int i = 0; i < jum; i++) {
24         baru = new Individu(min, max, minProduk);
25         individu[i + jumlah] = baru.getKromosom();
26     }
27 }
```

Source Code 5.7 Seleksi

Penjelasan algoritma seleksi pada Source Code 5.7 adalah sebagai berikut:

1. Baris ke-4 hingga ke -14 merupakan algoritma untuk mengurutkan individu yang telah dihasilkan pada proses inialisasi, rekombinasi, dan mutasi berdasarkan nilai fitness.
2. Baris ke-19 hingga ke-21 mengambil individu teratas sejumlah 75% dari jumlah miu.
3. Baris ke-23 hingga ke-25 merupakan implementasi random injection yang membangkitkan individu baru sejumlah 25% dari jumlah miu.

BAB 6 PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil pengujian pada algoritma Evolution Strategies untuk optimasi laba berdasarkan jumlah produksi pada home Industry tekstil. Berdasarkan perancangan yang telah dipaparkan pada Bab 4, dilakukan empat macam pengujian. Pengujian pertama adalah pengujian kisaran produksi maksimum yang bertujuan untuk mengetahui kisaran produksi maksimum suatu produk yang mampu menghasilkan fitness yang optimum berdasarkan jumlah bahan baku yang tersedia. Pengujian kedua adalah pengujian nilai λ (lambda) yang bertujuan untuk mengetahui pengaruh nilai λ (lambda) terhadap fitness yang dihasilkan. Pengujian ketiga adalah pengujian ukuran populasi yang bertujuan untuk mendapatkan ukuran populasi yang tepat dalam menghasilkan nilai fitness yang terbaik. Pengujian keempat adalah pengujian jumlah generasi yang bertujuan untuk mengetahui jumlah generasi yang efektif dan efisien dalam menghasilkan nilai fitness yang optimum. Pengujian tersebut dilakukan dengan asumsi tidak ada jumlah minimal produksi.

6.1 Hasil dan Analisis Pengujian

6.1.1 Hasil dan analisis uji coba kisaran produksi maksimum

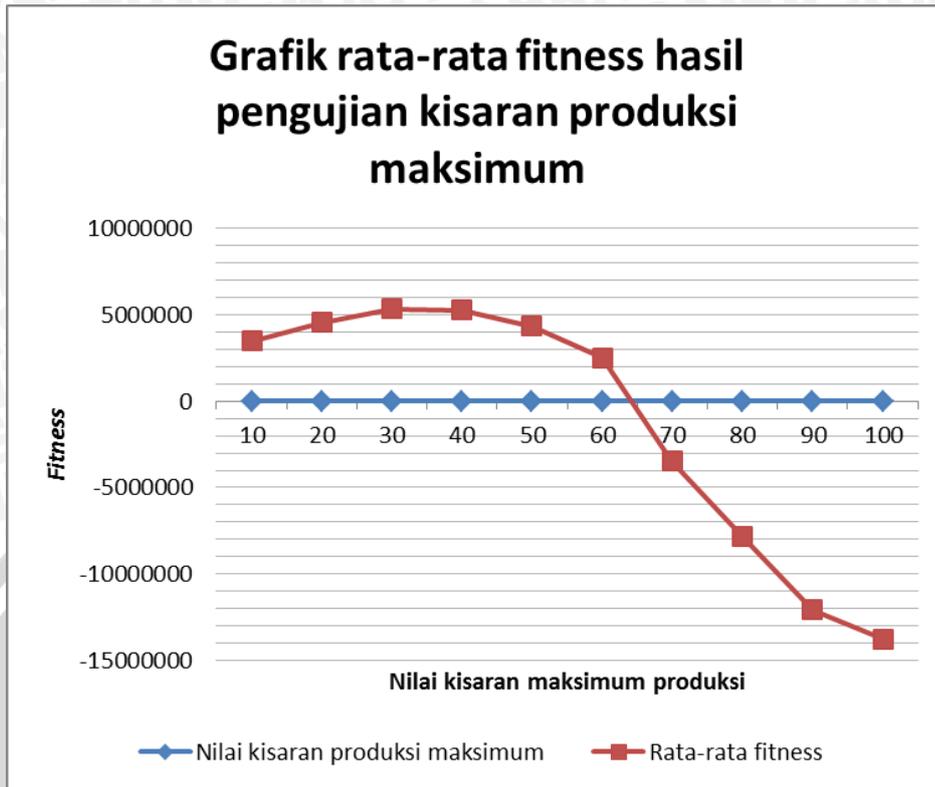
Pada pengujian ini, nilai λ (lambda) yang digunakan adalah 4, jumlah generasi sebanyak 100 generasi dan nilai ukuran populasi (μ) yang digunakan bervariasi yakni 15, 20, 25, 30, 35, 40, 45, dan 50. Nilai uji kisaran produksi maksimum yang digunakan adalah rentang 10 hingga 100 dengan nilai kelipatan 10. Setelah didapatkan nilai fitness pada masing-masing nilai kisaran produksi maksimum dengan parameter yang telah diinputkan, maka akan dihitung rata-rata nilai fitness. Lalu akan didapatkan nilai rata-rata fitness yang terbaik. Sehingga dapat dilakukan analisis terhadap nilai kisaran produksi maksimum yang digunakan. Hasil pengujian kisaran produksi maksimum disajikan pada Tabel 6.1.

Tabel 6.1 Hasil pengujian kisaran produksi maksimum

Nilai kisaran produksi maksimum	Nilai Fitness pada miu-								Rata-rata fitness
	15	20	25	30	35	40	45	50	
10	3237000	3301000	2743000	3489000	3575000	3728000	3535000	4067000	3459375
20	3871000	4594000	4370000	4470000	4557000	4781000	4650000	4788000	4510125
30	5185000	5430000	5425000	5123000	5324000	5382000	5327000	5436000	5329000
40	5222000	5338000	5433000	5520650	4911000	4967000	5263000	5414000	5258581.25
50	3576500	5453125	5115000	5254000	2593500	3900000	3486000	4999000	4297140.625
60	4026350	-893500	2420000	3484500	3461160	3288000	565000	4004000	2480876.25
70	-1892650	-4331525	-3743750	-4037500	-5869000	-5793699	-1566500	-2127750	-3494065.5
80	-11824525	-6252000	-8026000	-9649750	-6109525	-7480875	-8839000	-4644574	-7853281.125
90	-16855800	-15073600	-6533275	-10222550	-15154625	-13044550	-12747400	-6755025	-12048353.13
100	-14009700	-16790375	-9851125	-11202750	-16760025	-12199375	-15222950	-13865825	-13737765.63

Pada Tabel 6.1 dapat diketahui bahwa nilai fitness terbesar adalah 5520650 yang terjadi pada nilai kisaran produksi maksimum 40 dan miu berukuran 30. Sedangkan nilai fitness terkecil adalah -16855800 yang dihasilkan pada nilai kisaran produksi maksimum 90 dan miu berukuran 15. Akan tetapi, jika diambil nilai rata-rata fitness yang dihasilkan dari masing-masing kisaran produksi maksimum, didapatkan nilai rata-rata fitness terbesar adalah 5329000 yang terjadi pada kisaran produksi maksimum sejumlah 30. Sedangkan nilai rata-rata fitness terendah adalah -1373765.63 yang terjadi pada nilai kisaran produksi maksimum 100. Grafik hasil pengujian kisaran produksi maksimum ditunjukkan pada Gambar 6.1.





Gambar 6.1 Grafik hasil pengujian kisaran produksi maksimum

Pada Gambar 6.1 dapat diketahui bahwa rata-rata fitness mengalami kenaikan ketika nilai kisaran produksi maksimum sejumlah 10 hingga sejumlah 30. Kemudian nilai rata-rata fitness mengalami penurunan terus menerus hingga nilai kisaran produksi maksimum sejumlah 100. Berdasarkan hasil pengujian tersebut dapat diketahui bahwa nilai kisaran produksi maksimum yang menghasilkan rata-rata fitness terbaik adalah 30. Nilai fitness yang didapatkan tersebut sangat dipengaruhi oleh jumlah bahan baku yang tersedia. Sehingga, jika jumlah bahan baku yang tersedia semakin banyak, maka nilai kisaran produksi maksimum yang mampu menghasilkan nilai fitness terbaik juga akan meningkat.

6.1.2 Hasil dan analisis uji coba nilai λ (lambda)

Pada pengujian ini, nilai kisaran produksi maksimum yang digunakan adalah nilai terbaik pada pengujian kisaran produksi maksimum yaitu 30. Jumlah generasi dilakukan sebanyak 100 generasi. Nilai ukuran populasi (μ) yang digunakan bervariasi yakni 15, 20, 25, 30, 35, 40, 45, dan 50. Nilai uji λ (lambda) yang digunakan adalah 2 sampai 10. Setelah didapatkan nilai fitness pada masing-masing nilai λ (lambda) berdasarkan parameter yang telah diinputkan maka akan dihitung rata-rata nilai fitness. Kemudian akan didapatkan nilai rata-rata fitness yang terbaik. Sehingga dapat dilakukan analisis terhadap nilai λ (lambda) yang diuji. Hasil pengujian nilai λ (lambda) disajikan pada Tabel 6.2.

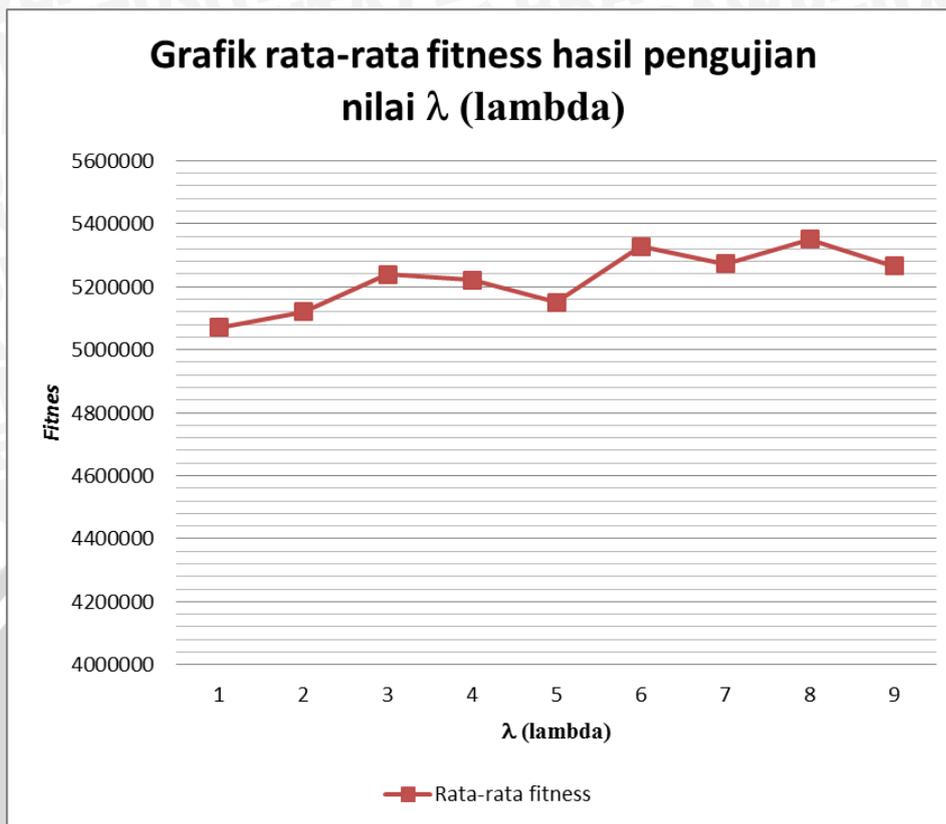


Tabel 6.2 Hasil pengujian nilai λ (lambda)

Nilai λ (lambda)	Nilai Fitness pada μ -								Rata-rata fitness
	15	20	25	30	35	40	45	50	
2	5219000	4590000	4894000	5017000	5233000	5088000	5341000	5185000	5070875
3	4655000	5311000	5171000	4943000	5377000	5442500	4867000	5193000	5119938
4	5234000	4808000	5700000	5212000	5305000	5326000	5118500	5210000	5239188
5	4807000	5337000	5442500	4971000	5017000	5325500	5673000	5203000	5222000
6	4773000	5072000	5545000	5382000	5271000	5229000	4712000	5214500	5149813
7	5061500	5205000	5512000	5281000	5244000	5279000	5704500	5340000	5328375
8	5036000	4985000	5612725	5287000	5253000	5201000	5379500	5421000	5271903
9	5348000	5562000	5143000	5375500	5259000	5380000	5375000	5357000	5349938
10	5154500	5355000	5308000	4986000	5306000	5390000	5187000	5437500	5265500

Dari Tabel 6.2 dapat diketahui bahwa nilai fitness terbesar adalah 5704500 yang terjadi pada saat λ (lambda) bernilai 7 dan populasi (μ) berukuran 45. Sedangkan nilai fitness terkecil adalah 4590000 yang dihasilkan pada saat λ (lambda) bernilai 2 dan populasi (μ) berukuran 20. Pada rata-rata fitness yang dihasilkan dari masing-masing nilai λ (lambda) yang diujikan didapatkan nilai rata-rata fitness terbesar adalah 5349938 yang terjadi pada λ (lambda) yang bernilai 9. Sedangkan nilai rata-rata fitness terkecil adalah 5070875 yang terjadi pada λ (lambda) yang bernilai 2. Grafik hasil pengujian nilai λ (lambda) ditunjukkan pada Gambar 6.2.





Gambar 6.2 Hasil pengujian nilai λ (lambda)

Dari Gambar 6.2 dapat diketahui bahwa rata-rata nilai fitness mengalami peningkatan yang fluktuatif. Namun, secara umum jika nilai λ (lambda) semakin bertambah, maka rata-rata nilai fitness cenderung mengalami kenaikan. Berdasarkan hasil tersebut, dapat diambil kesimpulan bahwa perbedaan nilai fitness yang didapatkan pada pengujian masing-masing nilai λ (lambda) memiliki kenaikan yang tidak terlalu signifikan.

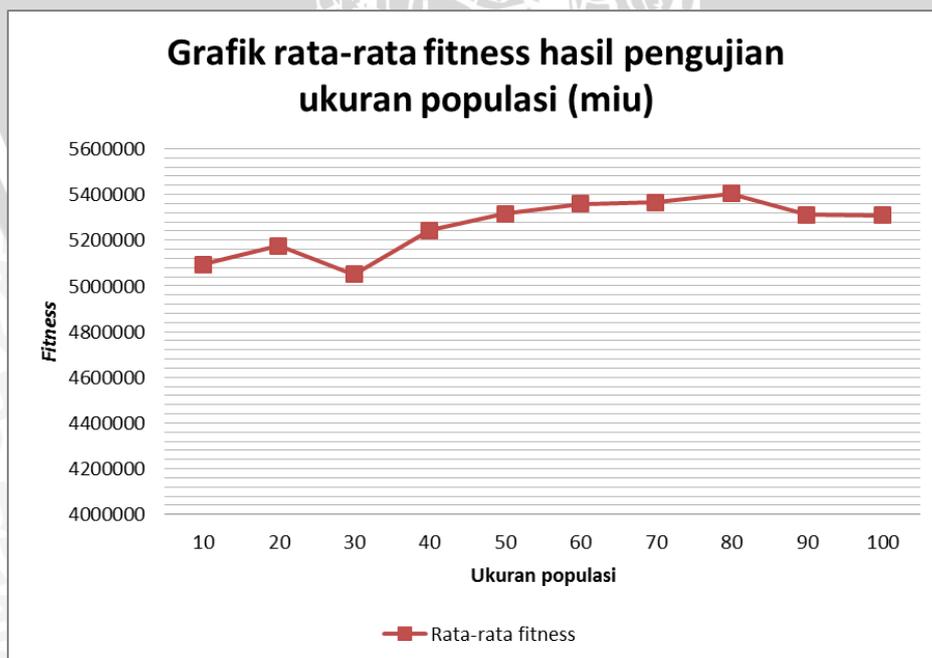
6.1.3 Hasil dan analisis uji coba ukuran populasi (miu)

Pada pengujian ini, nilai kisaran produksi maksimum yang digunakan adalah nilai terbaik pada pengujian kisaran produksi maksimum yaitu 30. Nilai λ (lambda) yaitu 9 yang merupakan nilai λ (lambda) terbaik yang didapatkan pada pengujian sebelumnya. Jumlah generasi dilakukan sebanyak 100 generasi. Nilai ukuran populasi (miu) yang diujikan adalah rentang 10 hingga 100 dengan kelipatan 10. Percobaan dilakukan sebanyak delapan kali untuk setiap ukuran populasi yang diujikan. Kemudian dari hasil fitness setiap ukuran populasi akan dihitung nilai rata-ratanya agar bisa dilakukan analisis terhadap ukuran populasi yang diuji. Hasil pengujian ukuran populasi (miu) disajikan pada Tabel 6.3.

Tabel 6.3 Hasil pengujian ukuran populasi (miu)

ukuran populasi	Nilai <i>Fitness</i> pada percobaan ke-								Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	
10	5180000	4942000	5713025	4906000	5328000	4728000	5357000	4586000	5092503.125
20	5475000	5156000	5052000	5194000	4856000	5135000	5290000	5249000	5175875
30	5172000	4860500	4972000	4775000	5020500	5342000	5055000	5203000	5050000
40	5332000	5346000	5310500	5412000	5068000	5099000	5157500	5214000	5242375
50	5343000	5062000	5363000	5555000	5297000	5348500	4893000	5663675	5315646.875
60	5369000	5162500	5187000	5180500	5680000	5722000	5232000	5328500	5357687.5
70	5796375	5461500	5136500	5276000	5146000	5671500	5057500	5379500	5365609.375
80	5280000	5303500	5414000	5150000	5399000	5420500	5757475	5505000	5403684.375
90	4997500	5320000	5306000	5296500	5524500	5329000	5118000	5589000	5310062.5
100	5214000	5283500	5211000	5555000	4988000	5213500	5677900	5329000	5308987.5

Berdasarkan Tabel 6.3 dapat diketahui bahwa nilai fitness terbesar yang didapatkan adalah 5796375 pada ukuran populasi sejumlah 70 pada percobaan pertama. Sedangkan nilai fitness terkecil adalah 4586000 yang terjadi pada ukuran populasi sejumlah 10 pada percobaan kedelapan. Pada rata-rata fitness yang dihasilkan dari masing-masing ukuran populasi yang diuji didapatkan nilai rata-rata fitness terbesar adalah 5403684.375 pada ukuran populasi sejumlah 80. Sedangkan nilai rata-rata fitness terkecil adalah 5050000 yang terjadi pada saat jumlah ukuran populasi sebanyak 30. Grafik hasil pengujian ukuran populasi ditunjukkan pada Gambar 6.3.



Gambar 6.3 Hasil pengujian ukuran populasi (miu)



Dari Gambar 6.3 dapat diketahui bahwa ukuran populasi berpengaruh terhadap hasil algoritma Evolution Strategies yang ditunjukkan pada rata-rata fitness yang dihasilkan. Semakin besar ukuran populasi maka hasil rata-rata fitness yang didapatkan juga cenderung akan semakin meningkat. Jika ukuran populasi terlalu kecil maka algoritma 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. Begitu juga dengan waktu komputasi yang digunakan, apabila ukuran populasi semakin besar maka akan membutuhkan waktu komputasi yang lebih lama.

6.1.4 Hasil dan analisis uji coba jumlah generasi

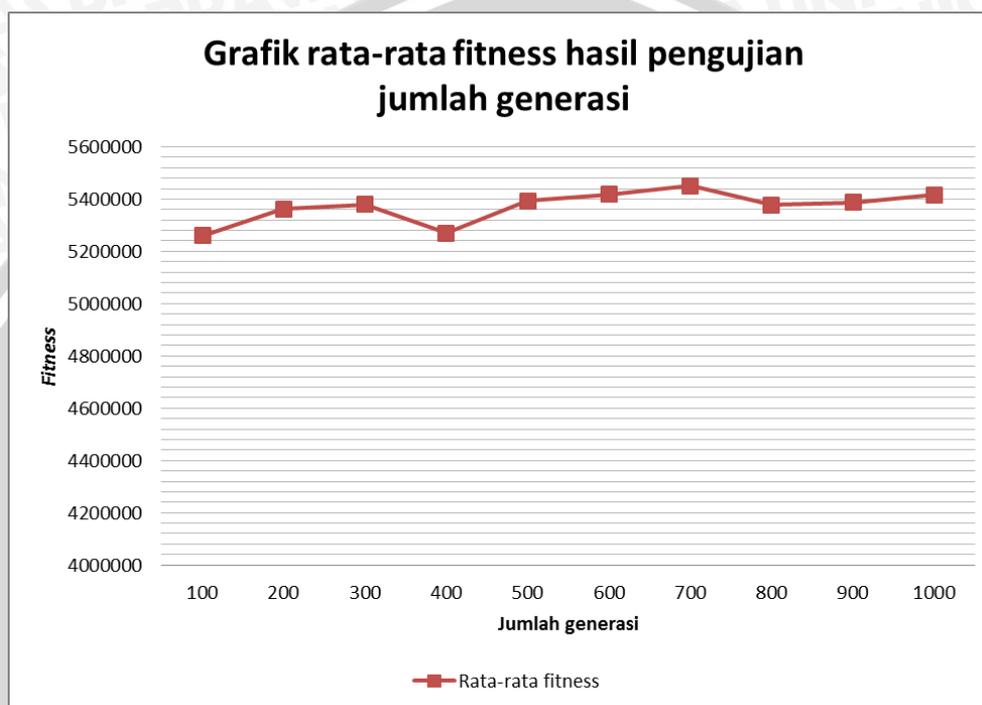
Pada pengujian ini, nilai kisaran produksi maksimum yang digunakan adalah nilai terbaik pada pengujian kisaran produksi maksimum yaitu sebanyak 30. Nilai λ (lambda) yaitu 9 yang merupakan nilai λ (lambda) terbaik yang didapatkan pada pengujian sebelumnya. Nilai ukuran populasi yang digunakan adalah sejumlah 80 yang merupakan ukuran populasi yang memiliki rata-rata fitness terbaik pada pengujian ukuran populasi. Jumlah generasi yang diujikan adalah rentang 100 hingga 1000 dengan kelipatan nilai sejumlah 100. Percobaan dilakukan sebanyak delapan kali untuk setiap jumlah generasi yang diuji. Kemudian akan dihitung nilai rata-rata dari hasil fitness yang didapatkan pada setiap jumlah generasi agar bisa dianalisis. Hasil pengujian jumlah generasi disajikan pada Tabel 6.4.

Tabel 6.4 Hasil pengujian jumlah generasi

Jumlah generasi	Nilai <i>Fitness</i> pada percobaan ke-								Rata-rata <i>fitness</i>
	1	2	3	4	5	6	7	8	
100	5295500	4997000	5394000	5032000	5563000	5018000	5395000	5379000	5259187.5
200	5436000	5398000	5218500	5329000	5226000	5302000	5451000	5540000	5362562.5
300	5641000	5249000	5320000	5375000	5312500	5414500	5409000	5312000	5379125
400	5529500	5254000	5670000	5105500	5152000	5218000	5004500	5224000	5269687.5
500	5395500	5444500	5763000	5202000	5703475	4915500	5332000	5384000	5392496.9
600	5260500	5243500	5486500	5490500	5615500	5191500	5399000	5662000	5418625
700	5534725	5564500	5409500	5330000	5470500	5463500	5264000	5573000	5451215.6
800	5517500	5053000	5133500	5585000	5602000	5448000	5437000	5244500	5377562.5
900	5253500	5512000	5346000	5444000	5108500	5353500	5327500	5745000	5386250
1000	5528000	5412500	5412500	5257000	5328000	5490000	5475000	5430500	5416687.5

Berdasarkan Tabel 6.4 didapatkan nilai fitness terbesar adalah 5763000 yang terjadi pada saat uji generasi sejumlah 500 dan pada percobaan ketiga. Nilai fitness terkecil juga terjadi pada saat jumlah generasi sebanyak 500 pada percobaan keenam dengan nilai 491550. Hal ini bisa terjadi karena inisialisasi

yang dihasilkan pada populasi awal didapatkan dengan cara membangkitkan angka random. Sehingga tidak menutup kemungkinan dalam pengujian jumlah generasi yang sama didapatkan nilai fitness yang paling tinggi maupun paling rendah seperti hasil tersebut. Pada rata-rata fitness yang dihasilkan dari masing-masing jumlah generasi yang diuji, didapatkan nilai rata-rata fitness terbesar adalah 5451215.6 pada jumlah generasi sebanyak 700. Sedangkan nilai rata-rata fitness terkecil adalah 5259188 yang terjadi pada saat jumlah generasi sebanyak 100. Grafik hasil pengujian jumlah generasi ditunjukkan pada Gambar 6.4.



Gambar 6.4 Hasil pengujian jumlah generasi

Pada Gambar 6.4 dapat dilihat jika jumlah generasi bertambah maka nilai rata-rata fitness yang dihasilkan juga cenderung bertambah. Namun, pada hasil tersebut terjadi dua kali penurunan yaitu pada pengujian jumlah generasi sebanyak 400 dan jumlah generasi sebanyak 800. Pada setiap kenaikan jumlah generasi pengujian terjadi perubahan nilai rata-rata fitness yang kurang signifikan dan waktu yang dibutuhkan untuk melakukan komputasi menjadi lebih lama. Sehingga dapat disimpulkan bahwa ketika jumlah generasi yang digunakan semakin banyak maka akan didapatkan nilai fitness yang lebih baik namun tidak signifikan serta membutuhkan waktu komputasi yang lebih lama.

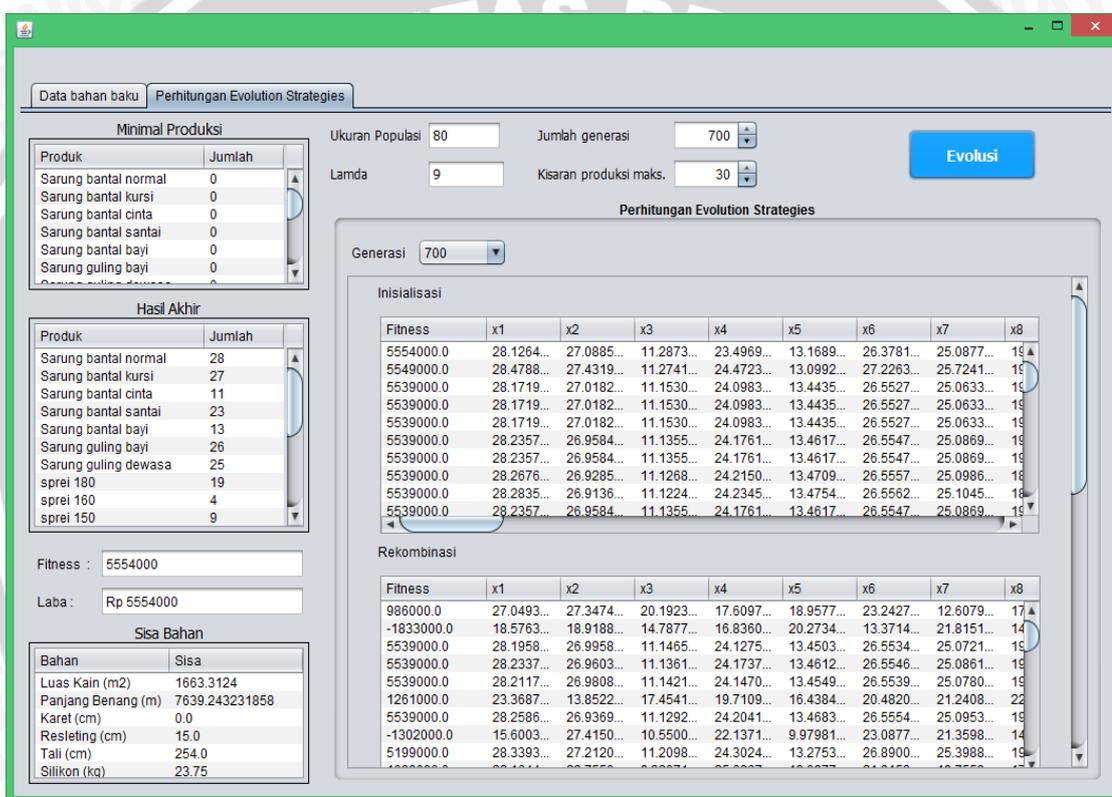
6.2 Solusi menggunakan parameter terbaik

Hasil terbaik berdasarkan pengujian yang telah dilakukan sebelumnya akan dijadikan nilai input masing-masing parameter untuk mendapatkan solusi yang paling mendekati optimum. Hasil parameter terbaik berdasarkan pengujian sebelumnya disajikan pada Tabel 6.5.

Tabel 6.5 Hasil parameter dengan nilai terbaik

Parameter	Nilai
Kisaran produksi maksimum	30
λ (lambda)	9
ukuran populasi (miu)	80
Jumlah generasi	700

Solusi yang didapatkan berdasarkan parameter dengan nilai terbaik ditampilkan pada Gambar 6.5



Gambar 6.5 Hasil solusi menggunakan parameter dengan nilai terbaik

Dari satu kali hasil percobaan menggunakan parameter dengan nilai terbaik didapatkan nilai fitness sebesar 5554000. Fungsi tujuan bernilai sama yaitu sebesar Rp 5,554,000 karena kendala bahan yang digunakan untuk produksi tidak melebihi stok bahan yang tersedia. Pada Gambar 6.5 terlihat bahwa sisa bahan karet habis digunakan, resleting tersisa 15 cm, tali tersisa 254 cm, silikon 23.75 kg, luas kain 1663.3124 m², dan benang 7639.243 m². Dari sisa bahan tersebut terjadi perbedaan jumlah sisa bahan yang cukup jauh. Padahal, jumlah bahan yang tersedia sangat berpengaruh terhadap fitness yang dihasilkan karena merupakan kendala dalam proses produksi. Sehingga, agar fitness yang dihasilkan mampu memberikan hasil yang lebih optimal selain dengan parameter terbaik yang telah didapatkan, maka jumlah bahan yang tersedia khususnya pada

karet, resleting, tali, dan silicon perlu ditambah agar kain dan benang yang tersisa masih bisa digunakan dan jumlah produksi menjadi bertambah.

6.2.1 Perbandingan Solusi

Solusi yang dihasilkan dengan menggunakan algoritma Evolution Strategies akan dibandingkan dengan solusi real berdasarkan data produksi dari narasumber pada bulan Juni 2016. Parameter terbaik yang ditampilkan pada Tabel 6.5 akan digunakan untuk percobaan mendapatkan nilai fitness sebanyak sepuluh kali dengan asumsi minimal produksi terdapat pada sarung bantal normal sejumlah 20 (sesuai data dari narasumber) untuk tiap percobaan. Hasil dari percobaan tersebut disajikan pada Tabel 6.6.

Tabel 6.6 Hasil Percobaan dengan parameter terbaik

Percobaan ke-	fitness	Labar
1	5241000	5241000
2	5265000	5275000
3	5272000	5302000
4	5309000	5309000
5	5520000	5530000
6	5413000	5413000
7	5139000	5159000
8	5435000	5435000
9	5420000	5430000
10	5146000	5146000
Rata-rata	5316000	5324000

Dari percobaan tersebut didapatkan nilai rata-rata fitness sebesar 5316000 dan rata-rata laba sebesar 5324000 yang akan digunakan sebagai pembandingan terhadap nilai fitness real dan laba real yang didapatkan dari data produksi pada bulan Juni 2016 melalui narasumber dengan asumsi tanpa menyertakan sisa bahan baku pada bulan sebelumnya ke dalam bulan Juni. Berikut adalah data produksi pada bulan Juni yang disajikan pada Tabel 6.7.

Tabel 6.7 Kuantitas produksi bulan Juni 2016

Produk	Jumlah Produksi
Sarung bantal normal	50
Sarung bantal kursi	8
Sarung bantal cinta	10

Tabel 6.7 Kuantitas produksi bulan Juni 2016 (lanjutan)

Produk	Jumlah Produksi
Sarung bantal santai	0
Sarung bantal bayi	10
Sarung guling bayi	20
Sarung guling dewasa	15
sprei 180	10
sprei 160	5
sprei 150	0
sprei 120	5
sprei 100	0
sprei 90	10
bantal normal	25
bantal kursi	3
bantal cinta	5
bantal santai	0
bantal bayi	5
guling bayi	10
guling dewasa	20
Fitness	2747000
Laba	Rp 2,747,000

Dari data produksi pada Tabel 6.7 tersebut didapatkan nilai fitness sebesar 2747000 dan laba sebesar Rp 2,747,000. Jika dibandingkan antara nilai fitness dan laba yang didapatkan dari kuantitas produksi melalui Algoritma *Evolution Strategies* dengan nilai fitness dan laba yang didapatkan berdasarkan data kuantitas produksi dari narasumber maka dapat diketahui bahwa nilai fitness yang dihasilkan dengan menggunakan algoritma *Evolution Strategies* menghasilkan fitness yang lebih besar yaitu 5316000 dibanding 2747000 dengan selisih sebesar 2569000. Laba yang dihasilkan dengan menggunakan algoritma *Evolution Strategies* juga menghasilkan nilai yang lebih besar yaitu Rp 5,324,000 dibanding Rp 2,747,000 dengan selisih sebesar Rp 2,577,000. Dari hasil tersebut maka algoritma *Evolution Strategies* menghasilkan solusi yang lebih baik.

BAB 7 PENUTUP

Bab ini membahas kesimpulan terkait penelitian optimasi laba berdasarkan jumlah produksi pada home industry tekstil menggunakan algoritma Evolution Strategies yang telah dilakukan dan juga saran untuk penelitian terkait dan juga pengembangan penelitian selanjutnya agar sistem lebih baik lagi.

7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis dari implementasi algoritma *Evolution Strategies* Untuk Optimasi Laba Berdasarkan Jumlah Produksi Pada *Home Industry* Tekstil, maka diperoleh kesimpulan sebagai berikut:

1. Algoritma Evolution Strategies dapat diterapkan untuk mengoptimasi laba berdasarkan jumlah produksi pada home industry tekstil dan menghasilkan solusi yang lebih baik dibandingkan dengan proses perkiraan kuantitas produksi secara manual.
2. Berdasarkan pengujian parameter yang telah dilakukan, diperoleh nilai parameter-parameter terbaik yang menghasilkan solusi mendekati optimal. Parameter tersebut meliputi kisaran produksi maksimum sebesar 30, lambda sebesar 9, mu sebesar 80, dan jumlah generasi sebesar 700.
3. Nilai parameter kisaran produksi maksimum sangat bergantung pada jumlah bahan baku yang tersedia. Jika jumlah bahan baku yang tersedia semakin banyak maka nilai parameter jumlah kisaran produksi yang menghasilkan fitness optimum juga akan bertambah. Begitu juga sebaliknya jika jumlah bahan baku yang tersedia semakin sedikit maka nilai parameter jumlah kisaran produksi yang menghasilkan fitness optimum juga semakin kecil.
4. Nilai parameter lambda, mu, dan jumlah generasi yang semakin besar akan mampu memberikan kesempatan algoritma untuk meningkatkan keragaman individu. Sehingga fitness yang didapatkan juga akan semakin optimal. Namun terkadang terjadi peningkatan yang tidak terlalu signifikan dan waktu yang dibutuhkan untuk melakukan komputasi juga semakin bertambah.
5. Hasil terbaik berdasarkan uji coba dengan menggunakan parameter terbaik pernah didapatkan laba sebesar Rp 5,882,000.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka saran yang dapat diberikan untuk penelitian lebih lanjut adalah sebagai berikut:

1. Agar laba yang didapatkan semakin maksimal, maka tidak hanya menggunakan data keuntungan penjualan, namun bisa ditambahkan data peramalan minat pasar (animo konsumen). Hal ini karena bisa jadi suatu produk tertentu yang secara matematis mampu menghasilkan laba paling

maksimal belum tentu merupakan produk yang paling diminati pasar. Sehingga dengan adanya kombinasi data laba penjualan dan peramalan minat pasar diharapkan lebih mampu memaksimalkan laba yang didapatkan. Data peramalan minat pasar tersebut dapat berupa data time series.

2. Perhitungan atau algoritma tertentu untuk menentukan rentang kisaran produksi secara otomatis sesuai dengan jumlah bahan baku yang tersedia dapat ditambahkan pada program. Sehingga apabila pada bulan tertentu terjadi perubahan jumlah bahan baku maka program dapat langsung digunakan tanpa harus melakukan pengujian ulang pada kisaran produksi maksimum.
3. Algoritma Evolution Strategies dapat dimodifikasi atau dilakukan penggabungan (*hybridization*) dengan algoritma yang lain agar mampu menghasilkan solusi yang lebih baik.



DAFTAR PUSTAKA

- Abrianto. (2012). Pertanggungjawaban Terhadap Produk Industri Rumah Tangga (Home Industry) Tanpa Izin Dinas Kesehatan. *universitas Hasanuddin*.
- Alghofari, F., & Pujiyono, A. (2010). ANALISIS TINGKAT PENGANGGURAN DI INDONESIA Tahun 1980-2007. [pdf] Tersedia di: <http://eprints.undip.ac.id/26483/2/Jurnal_Skripsi.pdf> [Diakses 1 Februari 2016].
- Badan Pusat Statistik. (2015). *KEADAAN KETENAGAKERJAAN AGUSTUS 2015*. [pdf] Badan Pusat Statistik. Tersedia di: <https://www.bps.go.id/website/brs_ind/brsInd-20151105121046.pdf> [Diakses 1 Februari 2016]. Jakarta: Badan Pusat Statistik.
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F. J., Cano, J. C., Calafate, C. T., et al. (2014). Reducing emergency services arrival time by using vehicular communications and Evolution Strategies. *Elsevier*, 1206–1217.
- Binh, T., & Korn, U. (n.d.). An Evolution Strategy For The Multiobjective Optimization. *Institute for Automation, Otto von Guericke University*.
- Dahlia. (2011). Analisis Biaya-Volume-Laba Sebagai Alat Bantu Dalam Perencanaan Laba Pt Pabrik Gula Takalar. *Universitas Hasanuddin*.
- Kashan, A. H., Akbari, A. A., & Ostadi, B. (2015). Grouping evolution strategies: An effective approach for grouping problems. *Elsevier*, 2703–2720.
- Kramer, O. (2014). A Brief Introduction to Continous Evolutionary Optimization. *Springer*, 29.
- Mahmudy, W. F. (2013). *Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer.
- Mbiyaki, H. K. (2013). Toward Evolution Strategies Application in Automatic Polyphonic Music Transcription using Electronic Synthesis. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 3.
- Milah, H., & Mahmudy, W. F. (2105). Implementasi Algoritma Evolution Strategies Untuk Optimasi Komposisi Pakan Ternak Sapi Potong. *Doro Jurnal*, vol. 5, no. 11.
- Munawarah, F., & Mahmudy, W. F. (2015). Optimasi Distribusi Pupuk Menggunakan Evolution Strategies. *Journal of Environmental Engineering & Sustainable Technology (JEEST)*, Vol. 02, No. 02, Pages 89-96.

- Palupi, L. D., Irhamah, & Mumpuni, S. (2011). Algoritma Genetika Untuk Optimasi Persediaan Multi Barang Dalam Proses Produksi, [e-journal]. Tersedia melalui : Perpustakaan Institut Teknologi Sepuluh November <<http://digilib.its.ac.id/public/ITS-Undergraduate-17334-1307100073-Presentation.pdf>> [Diakses 2 Februari 2016].
- Ramuna, M. D., & Mahmudy, W. F. (2015). Optimasi Persediaan Barang Dalam Produksi Jilbab . *Doro Journal*, vol. 5, no.14.
- Rifqi, N., Maharani, W., & Shaufiah. (2011). ANALISIS DAN IMPLEMENTASI KLASIFIKASI DATA MINING. *Konferensi Nasional Sistem dan Informatika 2011 (KNS)*, 183-191.
- Ringgi Sika, A. B., Sasongko, T., & Setyawan, D. (2012). Strategi Pemerintahan Desa Dalam Meningkatkan Pendapatan Masyarakat Melalui Sektor Home Industry. *JISIP: Jurnal Ilmu Sosial dan Ilmu Politik*, vol. 1, no. 1.
- Samaher, & Mahmudy, W. F. (2015). Penerapan algoritma genetika untuk memaksimalkan laba. *Journal of Environmental Engineering & Sustainable Technology*, vol. 2, no. 11, pp. 6-11.
- Satriyanto. (2009). *Algoritma Genetika*. [pdf] Tersedia di: <<http://entin.lecturer.pens.ac.id/Kecerdasan%20Buatan/Buku/Bab%207%20Algoritma%20Genetika.pdf>> [Diakses 27 Mei 2016].

