

**OPTIMASI PERENCANAAN PRODUKSI AGREGAT MULTI-
PRODUCT MENGGUNAKAN HIBRIDISASI METODE
SIMULATED ANNEALING DAN ALGORITME GENETIKA
ADAPTIF**

TESIS

Untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh :
GUSTI EKA YULIASTUTI
166150100111027



**PROGRAM STUDI MAGISTER ILMU KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**



PENGESAHAN

**OPTIMASI PERENCANAAN PRODUKSI AGREGAT MULTI-
PRODUCT MENGGUNAKAN HIBRIDISASI METODE
SIMULATED ANNEALING DAN ALGORITME GENETIKA
ADAPTIF**

TESIS

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh :
GUSTI EKA YULIASTUTI
166150100111027

Tesis ini telah diuji dan dinyatakan lulus pada :
3 Agustus 2018

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

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

Ishardita Pambudi Tama, S.T, M.T, Ph.D
NIP. 19730819 199903 1 002

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 Tesis 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 serata disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah Tesis ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia Tesis ini digugurkan dan gelar akademik yang telah saya peroleh (Magister) dibatalkan serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 Ayat 2 dan Pasal 70).

Malang, 8 Agustus 2018

Gusti Eka Yulianti
166150100111027

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan tesis ini dengan judul “Optimasi Perencanaan Produksi Agregat *Multi-Product* Menggunakan Hibridisasi Metode *Simulated Annealing* dan Algoritme Genetika Adaptif” yang diajukan untuk menempuh ujian akhir Program Studi Magister Ilmu Komputer.

Dalam menyelesaikan penulisan tesis ini tidak terlepas dari peran serta dukungan berbagai pihak. Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D dan Bapak Ishardita Pambudi Tama, S.T, M.T, Ph.D; selaku dosen pembimbing penulis yang selalu memberikan perbaikan serta arahan dengan sangat baik selama proses pengerjaan tesis ini.
2. Ibu Dr. Eng. Fitri Utamingrum, S.T, M.T; Bapak Dr. Eng. Herman Tolle, S.T, M.T dan Bapak Dr. Eng. Fitra Abdurrachman Bachtiar, S.T, M.T; selaku dosen penguji pada saat seminar proposal tesis dan ujian akhir tesis yang selalu memberikan kritik dan saran agar kualitas tesis ini meningkat.
3. Papa, Mama, Ayah, Ibu, Om Widodo dan seluruh keluarga besar penulis; yang selalu mendoakan serta memberikan dukungan baik secara moril maupun materil.
4. Agung Mustika Rizki; yang selalu membantu penulis kapanpun dan dimanapun mulai dari awal pengerjaan tesis hingga akhirnya dinyatakan lulus ujian.
5. Teman-teman Magister Ilmu Komputer FILKOM UB, MCS [4.0], Grup Riset Sistem Cerdas, GP Family, TIF-M(enangan), Kotelawala dan teman-teman lainnya yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan tesis ini masih banyak terdapat kekurangan. Untuk itu, penulis menyampaikan permohonan maaf sebelumnya, serta diharapkan kritik dan saran yang bersifat membangun dalam penyempurnaan di masa mendatang.

Malang, 8 Agustus 2018

Penulis

gustiekay@gmail.com

ABSTRAK

Perencanaan produksi agregat termasuk jenis perencanaan jangka menengah bagi perusahaan. Dalam merencanakan produksi agregat, *stakeholder* perusahaan memerlukan waktu lama dikarenakan banyaknya variabel produksi yang harus dipertimbangkan agar nilai produksi yang dihasilkan dapat memenuhi permintaan konsumen namun dengan biaya produksi seminimal mungkin. Jika perusahaan memproduksi lebih dari satu jenis produk, maka variabel yang harus dipertimbangkan akan lebih banyak dan membutuhkan waktu yang lebih lama lagi. Permasalahan tersebut sangatlah penting untuk diselesaikan karena jika terjadi sedikit kesalahan dalam perencanaan akan berdampak langsung pada biaya produksi yang harus dikeluarkan serta keuntungan yang didapat perusahaan. Permasalahan tersebut termasuk ke dalam permasalahan optimasi. Penulis menerapkan Algoritme Genetika untuk menyelesaikan permasalahan perencanaan produksi agregat karena Algoritme Genetika memiliki kelebihan yakni dapat mengeksplorasi dan mengeksploitasi ruang pencarian agar didapat solusi yang optimal. Namun, Algoritma Genetika sering terjebak pada solusi optimal lokal dan juga mengalami konvergensi dini. Untuk mengatasi permasalahan yang dihadapi oleh Algoritme Genetika ini, penulis akan memperbaikinya dengan menggabungkan metode lain yakni *Simulated Annealing*. Fungsi dari *Simulated Annealing* adalah untuk memperbaiki setiap solusi hasil dari Algoritme Genetika. Metode usulan penulis selanjutnya disebut menjadi metode HSAAGA mampu memberikan solusi yang optimal berupa nilai produksi dengan biaya produksi minimal serta telah mempertimbangkan berbagai variabel produksinya. Hal tersebut dibuktikan dengan adanya selisih yang cukup besar apabila perencanaan produksi agregat dilakukan dengan menerapkan metode HSAAGA biaya produksi yang dikeluarkan oleh perusahaan sebesar Rp. 612.731.500,- sedangkan biaya produksi aktualnya adalah sebesar Rp. 645.703.000,-.

Kata Kunci: Agregat, Algoritme Genetika, *Multi-Product*, Perencanaan, Produksi, *Simulated Annealing*

DAFTAR ISI

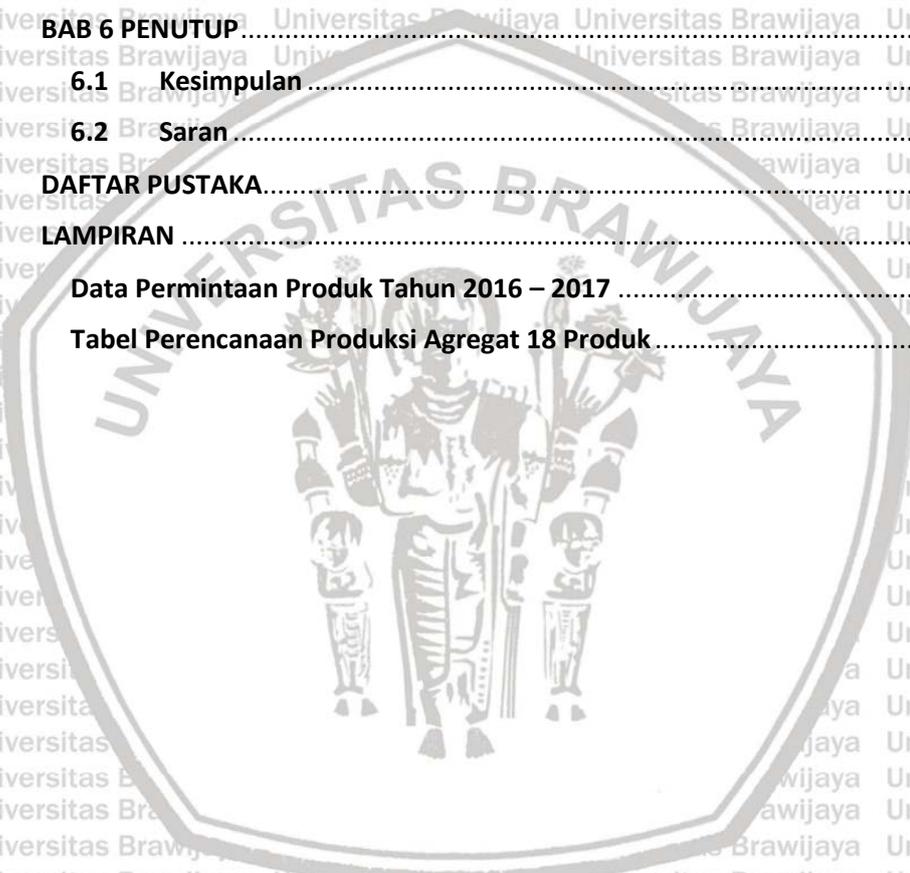
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR PERSAMAAN	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Perencanaan Produksi Agregat	6
2.2 Studi Kasus	8
2.3 Penelitian Terkait	10
2.4 Algoritme Genetika	12
2.4.1 Representasi Kromosom	14
2.4.2 Proses Evaluasi	17
2.4.3 Reproduksi Crossover	17
2.4.4 Reproduksi Mutasi	19
2.4.5 Proses Seleksi	20
2.5 Modifikasi Algoritme Genetika	22
2.6 Simulated Annealing	23
2.7 Mekanisme Hibridisasi	26
BAB 3 METODOLOGI	28
3.1 Metode Penelitian	28



3.2	Studi Literatur.....	29
3.3	Data.....	29
3.4	Skenario Pengujian.....	29
3.5	Skenario Implementasi.....	30
3.6	Skenario Perbandingan.....	30
BAB 4 PERANCANGAN.....		31
4.1	Perancangan Metode.....	31
4.1.1	Pembentukan Representasi Solusi.....	31
4.1.2	Pemilihan Jenis Operator Reproduksi.....	32
4.1.3	Pemilihan Jenis Operator Seleksi.....	34
4.1.4	Pemilihan Jenis Modifikasi.....	34
4.1.5	Perancangan Hibridisasi Metode.....	35
4.1.6	Perhitungan Nilai <i>Fitness</i>	37
4.1.7	Contoh Solusi dari Perhitungan Manual.....	37
4.1.8	Contoh Solusi menggunakan Algoritme Genetika.....	42
4.1.9	Contoh Solusi menggunakan Algoritme Genetika Adaptif.....	44
4.1.10	Contoh Solusi menggunakan Simulated Annealing.....	44
4.2	Perancangan Uji Coba dan Evaluasi.....	45
4.2.1	Uji Coba Ukuran Populasi.....	45
4.2.2	Uji Coba Jumlah Generasi.....	46
4.2.3	Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	47
4.2.4	Uji Coba Jumlah Solusi Baru.....	47
4.2.5	Uji Coba Nilai Temperatur.....	48
4.2.6	Uji Coba Nilai Faktor Reduksi.....	49
4.2.7	Uji Coba Jumlah Iterasi Maksimal.....	49
4.2.8	Uji Coba Perbandingan Metode.....	50
BAB 5 HASIL DAN PEMBAHASAN.....		51
5.1	Hasil Uji Coba Metode Algoritme Genetika.....	51
5.1.1	Hasil Uji Coba Ukuran Populasi.....	51
5.1.2	Hasil Uji Coba Jumlah Generasi.....	52
5.1.3	Hasil Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	53
5.2	Hasil Uji Coba Metode Algoritme Genetika Adaptif.....	54
5.2.1	Hasil Uji Coba Ukuran Populasi.....	54



5.2.2	Hasil Uji Coba Jumlah Generasi.....	56
5.3	Hasil Uji Coba Metode <i>Simulated Annealing</i>	57
5.3.1	Hasil Uji Coba Jumlah Solusi Baru.....	57
5.3.2	Hasil Uji Coba Nilai Temperatur.....	58
5.3.3	Hasil Uji Coba Nilai Faktor Reduksi.....	59
5.3.4	Hasil Uji Coba Jumlah Iterasi Maksimal.....	60
5.4	Hasil Uji Coba Metode HSAAGA.....	61
5.5	Hasil Uji Coba Perbandingan Metode.....	62
5.6	Perbandingan Hasil Penerapan Metode dengan Data Asli.....	63
BAB 6 PENUTUP.....		66
6.1	Kesimpulan.....	66
6.2	Saran.....	67
DAFTAR PUSTAKA.....		68
LAMPIRAN.....		71
	Data Permintaan Produk Tahun 2016 – 2017	71
	Tabel Perencanaan Produksi Agregat 18 Produk.....	77



DAFTAR GAMBAR

Gambar 2.1 Skema Algoritme Genetika	14
Gambar 2.2 Ilustrasi Metode N-Point Crossover.....	17
Gambar 2.3 Ilustrasi Metode Uniform Crossover	18
Gambar 2.4 Ilustrasi Metode Position Based Crossover.....	18
Gambar 2.5 Ilustrasi Metode Order Based Crossover.....	19
Gambar 2.6 Ilustrasi Metode Binary Code Mutation.....	19
Gambar 2.7 Ilustrasi Metode Position Based Mutation.....	20
Gambar 2.8 Ilustrasi Metode Order Based Mutation.....	20
Gambar 2.9 Ilustrasi Metode Scramble Mutation.....	20
Gambar 2.10 Skema Algoritme Simulated Annealing	26
Gambar 3.1 Alur Metode Penelitian.....	28
Gambar 4.1 Gambaran Umum Penyelesaian Permasalahan	31
Gambar 4.2 Kromosom Perencanaan Produksi Agregat Multi-Product	31
Gambar 4.3 Ilustrasi Modifikasi Crossover	32
Gambar 4.4 Ilustrasi Crossover pada Segmen 1.....	32
Gambar 4.5 Ilustrasi Crossover pada Segmen 2.....	32
Gambar 4.6 Ilustrasi Crossover pada Segmen 3.....	33
Gambar 4.7 Kromosom Child Hasil Crossover.....	33
Gambar 4.8 Ilustrasi Modifikasi Mutasi.....	33
Gambar 4.9 Ilustrasi Mutasi pada Segmen 1.....	33
Gambar 4.10 Ilustrasi Mutasi pada Segmen 2.....	34
Gambar 4.11 Ilustrasi Mutasi pada Segmen 3.....	34
Gambar 4.12 Kromosom Child Hasil Mutasi.....	34
Gambar 4.13 Pseudocode Algoritme Genetika Adaptif	35
Gambar 4.14 Skema HSAAGA.....	36
Gambar 4.15 Contoh Solusi Simulated Annealing.....	45
Gambar 5.1 Grafik Hasil Uji Coba Ukuran Populasi.....	52
Gambar 5.2 Grafik Hasil Uji Coba Jumlah Generasi.....	53
Gambar 5.3 Grafik Hasil Uji Coba Kombinasi CR dan MR.....	54
Gambar 5.4 Grafik Hasil Uji Coba Ukuran Populasi.....	55
Gambar 5.5 Grafik Hasil Uji Coba Jumlah Generasi.....	56
Gambar 5.6 Grafik Hasil Uji Coba Jumlah Solusi Baru.....	58
Gambar 5.7 Grafik Hasil Uji Coba Nilai temperatur.....	59
Gambar 5.8 Grafik Hasil Uji Coba Nilai Faktor Reduksi.....	60
Gambar 5.9 Grafik Hasil Uji Coba Jumlah Iterasi Maksimal.....	61
Gambar 5.10 Hasil Uji Coba Perbandingan Metode.....	63
Gambar 5.11 Perbandingan Permintaan dan Produksi.....	65

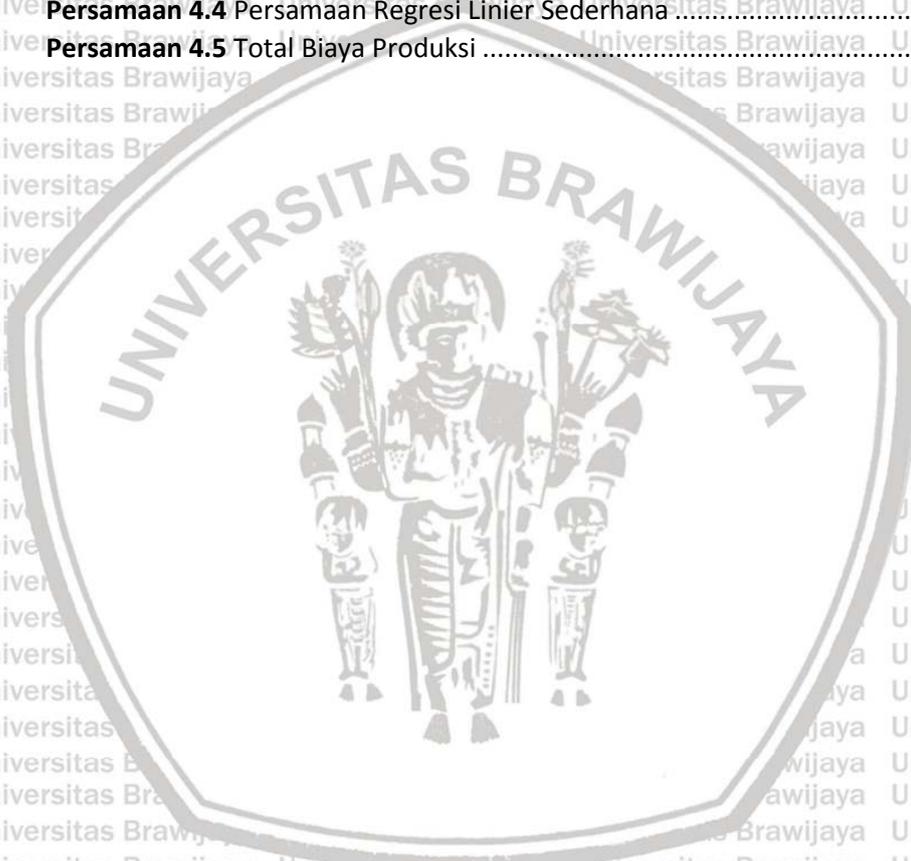
DAFTAR TABEL

Tabel 2.1 Konversi Proses Alamiah Menjadi Proses Komputasi.....	13
Tabel 2.2 Pengkodean Biner untuk Representasi Bilangan Bulat.....	15
Tabel 2.3 Pengkodean Biner untuk Representasi Bilangan Riil.....	15
Tabel 2.4 Pengkodean Riil.....	16
Tabel 2.5 Pengkodean Integer.....	16
Tabel 2.6 Pengkodean Permutasi.....	16
Tabel 4.1 Contoh Data 1 Jenis Produk.....	37
Tabel 4.2 Parameter dalam Proses Produksi.....	38
Tabel 4.3 Nilai Parameter Proses Produksi.....	38
Tabel 4.4 Perhitungan Nilai a dan b untuk Peramalan.....	39
Tabel 4.5 Hasil Peramalan Regresi Linier Sederhana.....	40
Tabel 4.6 Perencanaan Produksi Agregat.....	41
Tabel 4.7 Contoh Solusi Menggunakan Algoritme Genetika.....	44
Tabel 4.8 Contoh Solusi Menggunakan Algoritme Genetika Adaptif.....	44
Tabel 4.9 Skenario Uji Coba Ukuran Populasi.....	46
Tabel 4.10 Skenario Uji Coba Jumlah Generasi.....	46
Tabel 4.11 Skenario Uji Coba Kombinasi cr dan mr.....	47
Tabel 4.12 Skenario Uji Coba Jumlah Solusi Baru.....	48
Tabel 4.13 Skenario Uji Coba Nilai Temperatur.....	48
Tabel 4.14 Skenario Uji Coba Nilai Faktor Reduksi.....	49
Tabel 4.15 Skenario Uji Coba Jumlah Iterasi Maksimal.....	50
Tabel 4.16 Skenario Uji Coba Perbandingan Metode.....	50
Tabel 5.1 Hasil Uji Coba Ukuran Populasi.....	51
Tabel 5.2 Hasil Uji Coba Jumlah Generasi.....	52
Tabel 5.3 Hasil Uji Coba Kombinasi CR dan MR.....	53
Tabel 5.4 Hasil Uji Coba Ukuran Populasi.....	55
Tabel 5.5 Hasil Uji Coba Jumlah Generasi.....	56
Tabel 5.6 Hasil Uji Coba Jumlah Solusi Baru.....	57
Tabel 5.7 Hasil Uji Coba Nilai Temperatur.....	58
Tabel 5.8 Hasil Uji Coba Nilai Faktor Reduksi.....	59
Tabel 5.9 Hasil Uji Coba Jumlah Iterasi Maksimal.....	60
Tabel 5.10 Hasil Uji Coba Metode HSAAGA.....	62
Tabel 5.11 Hasil Uji Coba Perbandingan Metode.....	63
Tabel 5.12 Tabel Agregat Aktual.....	64
Tabel 5.13 Tabel Agregat HSAAGA.....	64



DAFTAR PERSAMAAN

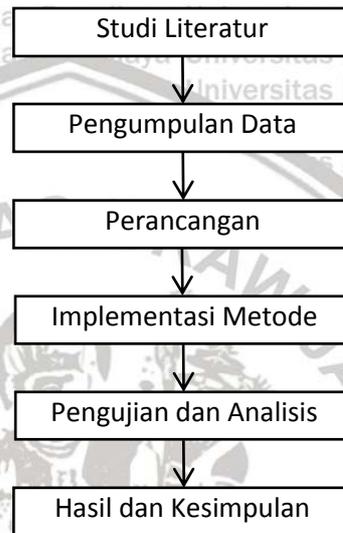
Persamaan 2.1 Nilai Fitness untuk Seleksi Sebanding.....	21
Persamaan 2.2 Probabilitas Kromosom untuk Seleksi Sebanding.....	21
Persamaan 2.3 Nilai Fitness untuk Seleksi Peringkat.....	21
Persamaan 2.4 Laju Pendinginan pada Simulated Annealing.....	23
Persamaan 2.5 Evaluasi Nilai Objektif pada Simulated Annealing.....	24
Persamaan 2.6 Evaluasi Probabilitas pada Simulated Annealing.....	25
Persamaan 4.1 Nilai Fitness.....	37
Persamaan 4.2 Rumus Nilai b pada Regresi Linier.....	39
Persamaan 4.3 Rumus Nilai a pada Regresi Linier.....	40
Persamaan 4.4 Persamaan Regresi Linier Sederhana.....	40
Persamaan 4.5 Total Biaya Produksi.....	42



BAB 3 METODOLOGI

3.1 Metode Penelitian

Penelitian ini dilakukan dengan cara mengimplementasi metode usulan guna menyelesaikan permasalahan yang ada pada perusahaan sebagai studi kasus. Tujuannya adalah agar dapat mengetahui apakah metode usulan tersebut cocok diterapkan pada perusahaan dan apakah memberikan pengaruh yang signifikan terhadap jumlah produksi serta keuntungan perusahaan. Alur metode penelitian yang diajukan oleh penulis seperti ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Metode Penelitian

Adapun tahapan yang dilakukan penulis dalam melakukan penelitian ini yakni sebagai berikut:

1. Melakukan studi literatur terkait permasalahan mengenai perencanaan produksi agregat dan juga terkait dengan metode-metode yang dapat digunakan dalam menyelesaikan permasalahan tersebut.
2. Mengumpulkan data pada perusahaan yang dapat dijadikan sebagai studi kasus penelitian.
3. Merancang mekanisme pengujian untuk setiap parameter pada metode usulan penulis.
4. Mengimplementasikan metode usulan penulis untuk menyelesaikan permasalahan perencanaan produksi agregat pada perusahaan.
5. Menganalisis dan membandingkan hasil implementasi antara metode usulan dengan metode pembanding lainnya.

6. Menganalisis dan membandingkan hasil implementasi metode usulan dengan hasil produksi nyata pada perusahaan yang selanjutnya digunakan untuk menghitung selisih keuntungan perusahaan.
7. Membuat kesimpulan berdasarkan penerapan metode usulan dalam menyelesaikan permasalahan perusahaan.

3.2 Studi Literatur

Studi literatur yang dilakukan penulis yakni dengan mengumpulkan dasar teori mengenai permasalahan perencanaan produksi agregat, menganalisis penelitian-penelitian sebelumnya yang sejenis dan juga menganalisis metode-metode yang dapat diterapkan dalam menyelesaikan permasalahan perencanaan produksi agregat.

3.3 Data

Pada penelitian ini penulis berfokus menyelesaikan permasalahan perencanaan produksi agregat dengan jumlah produk lebih dari satu yang selanjutnya dapat disebut *multi-product*. Data diperoleh dari salah satu perusahaan selaku industri tekstil rumahan di Kabupaten Bojonegoro. Data yang didapatkan berupa jumlah permintaan produk seragam sekolah sebanyak 18 jenis produk mencakup atasan dan bawahan, setiap jenis produk tersebut memiliki 8 ukuran yang berbeda. Data tersebut nantinya akan dibandingkan dengan data hasil penerapan metode usulan, untuk mengetahui apakah metode usulan tersebut optimal dalam menyelesaikan permasalahan di perusahaan tersebut. Untuk lebih lengkapnya data permintaan akan dilampirkan.

3.4 Skenario Pengujian

Pengujian diperlukan dalam mendapatkan solusi yang optimal. Pengujian dilakukan untuk mengetahui parameter terbaik yang terdapat pada kedua metode yakni Algoritme Genetika dan *Simulated Annealing*.

Adapun skenario yang perlu dilakukan yakni dengan cara menguji parameter-parameter yang ada dalam hibridisasi metode. Dalam menguji kehandalan Algoritme Genetika yang telah dimodifikasi dengan menggunakan adaptif parameter, tentu saja parameter yang perlu diperhatikan adalah nilai probabilitas *crossover* dan mutasi. Disamping itu, perlu juga diperhatikan jumlah populasi dan banyak iterasi yang dilakukan.

3.5 Skenario Implementasi

Skenario implementasi metode yang dilakukan oleh penulis sesuai dengan skenario pengujiannya, yakni dengan menerapkan masing-masing metode tunggal hingga metode hibridisasi usulan penulis dengan menggunakan parameter yang telah ditentukan di awal.

3.6 Skenario Pembandingan

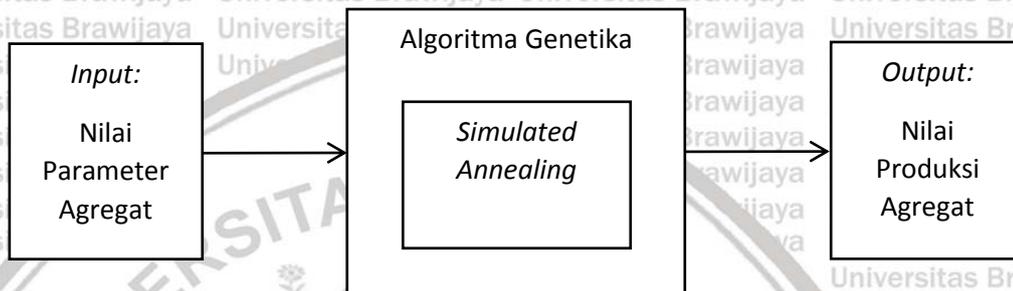
Untuk mengetahui hasil implementasi dengan menerapkan metode usulan penulis tersebut optimal, tentu saja perlu dibandingkan dengan metode lainnya. Metode usulan penulis adalah hibridisasi *Simulated Annealing* dan Algoritme Genetika Adaptif. Hasil dari metode usulan tersebut nantinya akan dibandingkan dengan hasil dari penerapan metode pembandingan yakni Algoritme Genetika, Algoritme Genetika Adaptif dan *Simulated Annealing* saja.



BAB 4 PERANCANGAN

4.1 Perancangan Metode

Pada bagian ini akan dijelaskan perancangan terkait dengan metode yang akan digunakan untuk penelitian ini. Perancangan metode ini terdiri dari pembentukan representasi kromosom, perancangan alur proses setiap metode, perancangan hibridisasi dua metode dan juga contoh solusi manual. Secara umum, alur proses penerapan metode untuk penyelesaian permasalahan pada penelitian ini ditunjukkan pada Gambar 4.1.

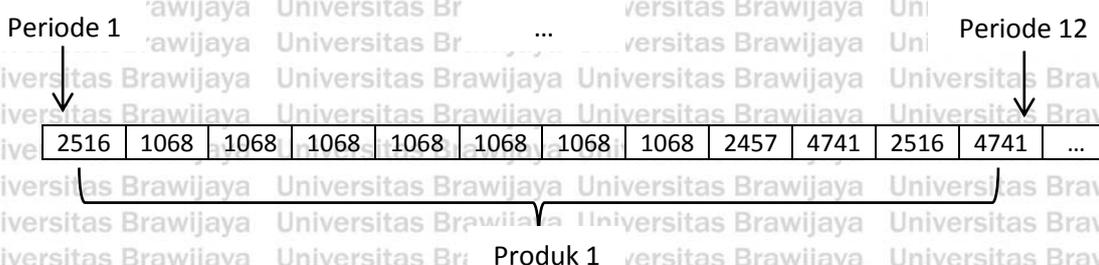


Gambar 4.1 Gambaran Umum Penyelesaian Permasalahan

4.1.1 Pembentukan Representasi Solusi

Pada penelitian ini, jenis pengkodean kromosom yang digunakan untuk merepresentasikan solusi adalah pengkodean riil (*real-coded*). Bilangan pecahan pada kromosom akan tetap seperti itu di kromosomnya guna menunjukkan jumlah produksi. Namun, bilangan tersebut akan dibulatkan pada saat perhitungan nilai *fitness*-nya. Untuk bilangan pecahan kurang dari setengah maka akan dibulatkan kebawah, sedangkan untuk bilangan pecahan sama dengan dan lebih dari setengah akan dibulatkan keatas.

Jenis permasalahan perencanaan produksi agregat yang akan diselesaikan oleh penulis adalah dalam 1 periode yakni 12 bulan terdapat lebih dari 1 produk dan setiap produk memiliki beberapa ukuran atau selanjutnya disebut *multi-product*. Sehingga kromosom yang akan terbentuk seperti pada Gambar 4.2.



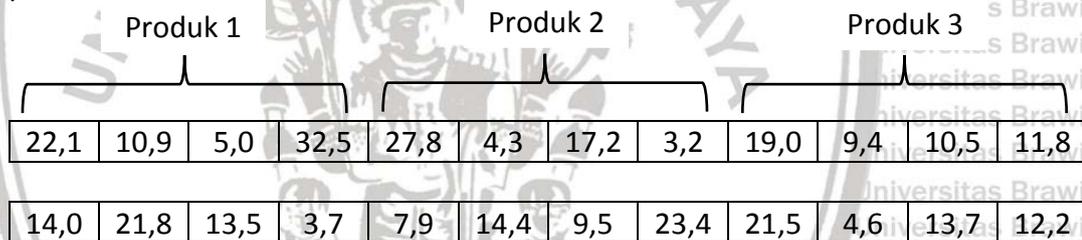
Gambar 4.2 Kromosom Perencanaan Produksi Agregat Multi-Product

Setiap gen pada kromosom menunjukkan jumlah produksi pada produk setiap periodenya. Dalam studi kasus pada penelitian ini, perusahaan melakukan perencanaan produksi untuk 1 tahun berikutnya dimana selanjutnya akan disebut 12 periode yang artinya 12 bulan. Pada gen 1 diisi untuk mewakili jumlah produksi pada periode 1 dan begitu seterusnya. Setiap 12 gen pada kromosom tersebut mewakili 1 produk yang sama dan begitu seterusnya hingga produk 18. Sehingga total panjang kromosom pada studi kasus penelitian ini adalah sebanyak 216 gen.

4.1.2 Pemilihan Jenis Operator Reproduksi

Jenis operator yang digunakan ada dua yakni *crossover* dan mutasi. Jenis *crossover* yang dipilih adalah *one-cut-point crossover* pada tiap segmennya. Hal pertama yang dilakukan adalah membangkitkan angka *random integer* dalam setiap segmen dengan *range* 1-11 (karena terdapat 12 gen pada tiap segmen yang menunjukkan 12 bulan). Angka tersebut nantinya digunakan sebagai acuan untuk gen yang akan menjadi titik perpotongan. Agar lebih mudah dipahami, penulis memberikan ilustrasi seperti ditunjukkan pada Gambar 4.3.

Dimisalkan pada kromosom *parent* terdapat 12 gen dengan 3 segmen produk.



Gambar 4.3 Ilustrasi Modifikasi Crossover

Pada segmen 1, misalkan nilai *random* = 2 maka dilakukan pemotongan pada gen kedua.



Gambar 4.4 Ilustrasi Crossover pada Segmen 1

Pada segmen 2, misalkan nilai *random* = 3 maka dilakukan pemotongan pada gen ketiga.



Gambar 4.5 Ilustrasi Crossover pada Segmen 2

Pada segmen 3, misalkan nilai *random* = 1 maka dilakukan pemotongan pada gen pertama.



Gambar 4.6 Ilustrasi Crossover pada Segmen 3

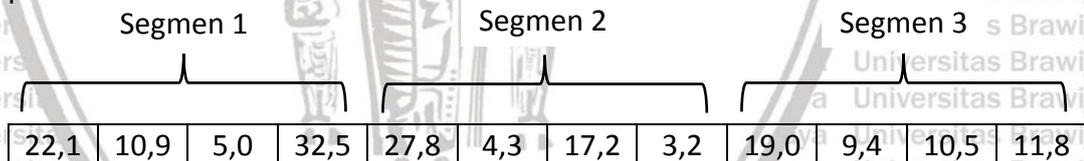
Sehingga kromosom *child* yang dihasilkan dari *crossover* menjadi seperti yang ditunjukkan Gambar 4.7.



Gambar 4.7 Kromosom Child Hasil Crossover

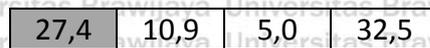
Sedangkan untuk jenis mutasinya, penulis memilih mutasi dasar berbasis posisi dengan sedikit modifikasi. Hal pertama yang dilakukan adalah membangkitkan angka *random integer* dalam setiap segmen dengan *range* 1-12 (karena terdapat 12 gen pada tiap segmen yang menunjukkan 12 bulan). Angka *random* tersebut menunjukkan posisi gen yang akan diubah nilainya. Perubahan nilai juga akan dilakukan secara *random* dengan *range* -10 hingga +10 dari nilai awalnya. Agar lebih mudah dipahami, penulis memberikan ilustrasi seperti ditunjukkan pada Gambar 4.8.

Dimisalkan pada kromosom *parent* terdapat 12 gen dengan 3 segmen produk..



Gambar 4.8 Ilustrasi Modifikasi Mutasi

Pada segmen 1, misalkan nilai *random* = 1 maka dilakukan perubahan nilai gen pertama. Misalkan nilai baru didapatkan dari pembangkitan nilai *random* = 27,4.



Gambar 4.9 Ilustrasi Mutasi pada Segmen 1

Pada segmen 2, misalkan nilai *random* = 4 maka dilakukan perubahan nilai gen keempat. Misalkan nilai baru didapatkan dari pembangkitan nilai *random* = 12,5.

27,8	4,3	17,2	12,5
------	-----	------	------

Gambar 4.10 Ilustrasi Mutasi pada Segmen 2

Pada segmen 3, misalkan nilai *random* = 3 maka dilakukan perubahan nilai gen ketiga. Misalkan nilai baru didapatkan dari pembangkitan nilai *random* = 17,3.

19	9,4	17,3	11,8
----	-----	------	------

Gambar 4.11 Ilustrasi Mutasi pada Segmen 3

Sehingga kromosom *child* yang dihasilkan dari mutasi menjadi seperti yang ditunjukkan Gambar 4.12.

27,4	10,9	5,0	32,5	27,8	4,3	17,2	12,5	19,0	9,4	17,3	11,8
------	------	-----	------	------	-----	------	------	------	-----	------	------

Gambar 4.12 Kromosom Child Hasil Mutasi

Kedua jenis operator tersebut dipilih dan dimodifikasi penulis dengan harapan adanya variasi dalam setiap individu baru yang dihasilkan, sehingga kemungkinan untuk mendapatkan solusi yang lebih optimum sangat besar.

4.1.3 Pemilihan Jenis Operator Seleksi

Jenis operator seleksi yang digunakan adalah *elitism*. Dengan menggunakan seleksi *elitism* akan tidak membuang banyak waktu dalam melakukan proses seleksi. Disamping itu, hasil seleksi *elitism* ini akan mendapatkan individu terbaik dari setiap generasi.

4.1.4 Pemilihan Jenis Modifikasi

Jenis modifikasi Algoritme Genetika yang dipilih oleh penulis adalah parameter adaptif. Pemilihan parameter adaptif bertujuan agar dapat menghasilkan individu yang lebih bervariasi guna mencapai solusi yang lebih optimal. Untuk lebih jelasnya pseudocode Algoritme Genetika dengan parameter adaptif seperti ditunjukkan pada Gambar 4.13.

```

procedure adaptive genetic algorithm
begin
  t <- 0;
  inialisasi P(t);
  evaluasi P(t);
  while (belum kondisi berhenti) do
  begin
    pilih acak parent P(t) untuk membentuk C(t);
    evaluasi C(t);
    pilih P(t+1) dari offspring P(t) dan C(t);
    P(t) = elitism (P(t), C(t));
    if (C(t) tidak lebih baik selama 50 iterasi)
      set cr acak;
      mr = 1 - cr;
    t <- t+1;
  end
end
  
```

Gambar 4.13 Pseudocode Algoritme Genetika Adaptif

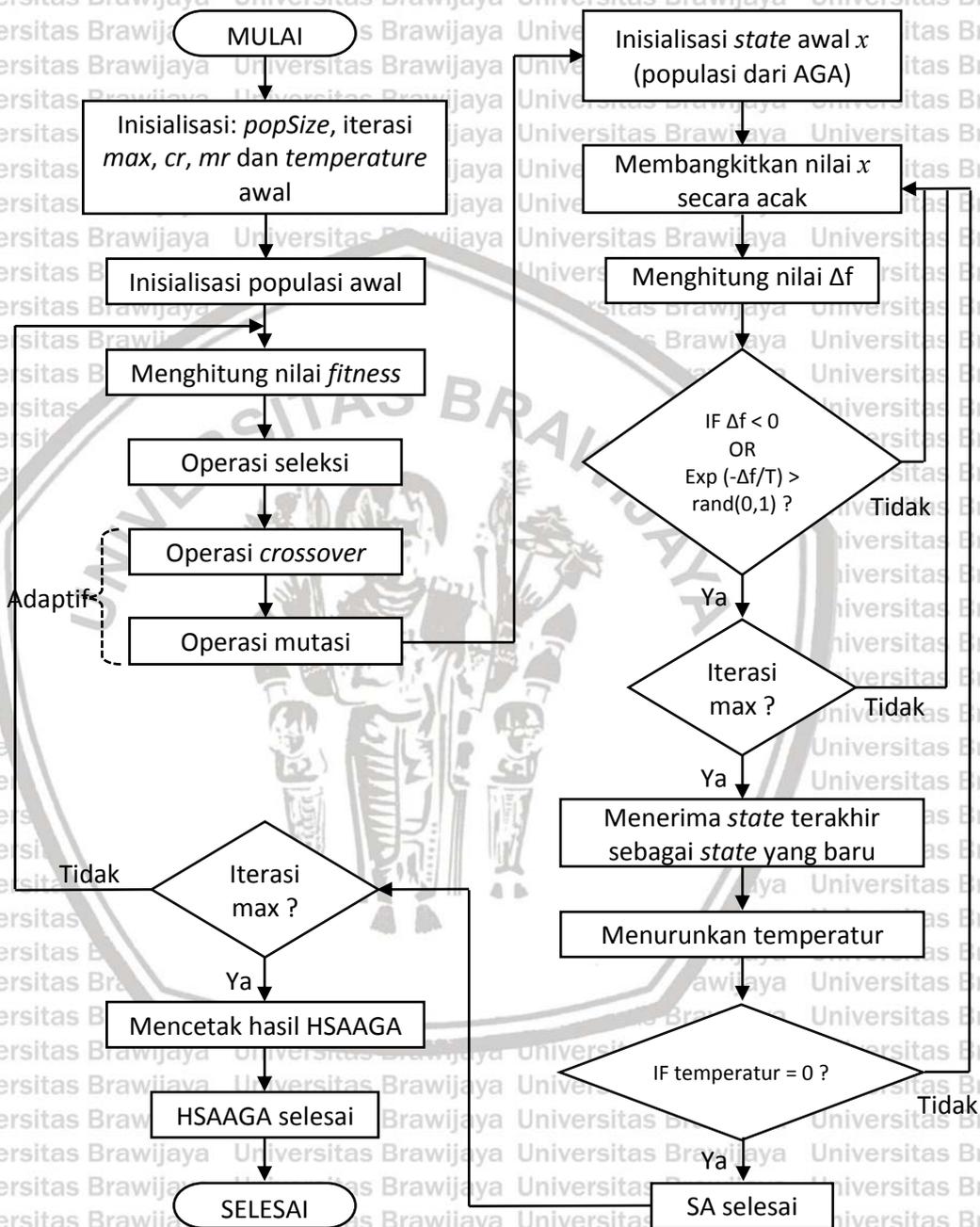
4.1.5 Perancangan Hibridisasi Metode

Algoritme *Simulated Annealing* (SA) dan Algoritme Genetika (AG) tentu memiliki kelemahannya masing-masing. Penelitian ini akan menggabungkan kedua algoritme tersebut dengan harapan akan mendapatkan hasil yang lebih optimal. Proses hibridisasi Algoritme *Simulated Annealing* dan Algoritme Genetika (HSAAG) ini terdiri dari penggabungan Algoritme Genetika dengan Algoritme *Simulated Annealing* secara iteratif (Zhang, Huang, Gao, & Sun, 2010). Perbedaan penelitian ini dengan penelitian sebelumnya adalah modifikasi pada AG menggunakan parameter adaptif yang selanjutnya diajukan menjadi metode Hibridisasi *Simulated Annealing* dan Algoritme Genetika Adaptif (HSAAGA). Pada HSAAGA, SA dianggap sebagai operator dalam AGA yang berfungsi untuk mengatur populasi secara optimal. Selama proses awal AGA, operator genetika digunakan seperti biasa untuk menghasilkan individu baru dan kemudian operator SA digunakan untuk mengidentifikasi individu tersebut.

Proses evolusi dasar dari awal yakni membangkitkan sejumlah individu dalam populasi secara acak tetap dilakukan. Dilanjutkan dengan proses reproduksi dengan menggunakan operator *crossover* dan mutasi guna menghasilkan populasi baru. Setelah proses AGA, semua individu dalam populasi terbaru tersebut dikirim ke SA untuk dilakukan perbaikan lebih lanjut. Populasi pembangkit baru akan dianggap sebagai populasi generasi berikutnya. Dapat disimpulkan bahwa pada HSAAGA terdapat dua aspek utama yakni populasi terbaik pertama dihasilkan oleh operator genetika pada AGA yang menekankan pencarian solusi secara global dan gen individu terbaik dihasilkan oleh operator SA yang menekankan pencarian solusi secara lokal. Dengan HSAAGA ini,

pencarian solusi untuk suatu permasalahan seperti pada studi kasus penelitian ini akan lebih optimal.

Skema penyelesaian permasalahan dengan HSAAGA dapat dilihat pada Gambar 4.14.



Gambar 4.14 Skema HSAAGA

4.1.6 Perhitungan Nilai *Fitness*

Untuk mengetahui kualitas solusi dari setiap kromosom tersebut tentu diperlukan adanya nilai *fitness*. Nilai *fitness* tersebut memperhitungkan seluruh biaya dengan berbanding terbalik, sehingga solusi terbaik tentu memiliki nilai *fitness* yang tinggi dengan total biaya minimum. Perhitungan nilai *fitness* ditunjukkan pada Persamaan (4-1). Adapun beberapa parameter biaya yang diperhitungkan dalam nilai *fitness* antara lain: biaya produksi pada *regular time* (*rt*), biaya produksi pada *overtime* (*ot*), biaya produksi pada *subcontract* (*sc*), biaya merekrut pekerja (*brp*), biaya merumahkan pekerja (*bmp*) dan juga biaya penyimpanan produk (*bi*). Disamping perhitungan biaya tersebut, tentu ada batasan lain yang dicapai guna menentukan kondisi selanjutnya, yakni: batas produksi maksimal pada *regular time*, batas produksi maksimal pada *overtime*, batas produksi maksimal pada *subcontract*, batas produksi maksimal setiap pekerjaanya dan juga *penalty* saat kondisi tidak memenuhi. *Penalty* yang dimaksud adalah adanya nilai minus pada *inventory*, hal tersebut memiliki arti bahwa perusahaan memiliki nilai produksi kecil dan tidak cukup persediaan untuk memenuhi permintaan konsumen.

Persamaan 4.1 Nilai *Fitness*

$$\text{Nilai Fitness} = \frac{100000000}{\text{Total Biaya Produksi} + \text{Penalty}} \quad (4-1)$$

4.1.7 Contoh Solusi dari Perhitungan Manual

Untuk gambaran awal, penulis menjelaskan tahapan perencanaan produksi agregat secara manual dengan cara coba dan uji (*trial and error*). Adapun contoh data yang digunakan yakni data produksi jenis hem putih dengan 8 macam ukuran yang berbeda selama 12 bulan mulai bulan September 2015 hingga Agustus 2016 seperti ditunjukkan pada Tabel 4.1.

Tabel 4.1 Contoh Data 1 Jenis Produk

Periode	Ukuran							
	II	III	IV	V	S	M	L	XL
Agt-15	5	81	181	192	127	124	65	65
Sep-15	0	60	137	136	163	144	46	75
Okt-15	2	30	159	237	196	163	121	39
Nov-15	10	10	115	188	168	135	75	25
Des-15	5	43	118	208	145	408	110	34
Jan-16	10	30	91	100	101	95	85	50
Feb-16	5	77	177	257	179	144	86	27
Mar-16	20	190	299	830	244	224	138	71
Apr-16	128	485	483	446	417	364	109	128
Mei-16	211	548	611	575	236	181	176	122
Jun-16	177	1024	1802	1142	978	821	831	333
Jul-16	134	789	1243	824	690	571	455	166

Disamping data tersebut, diperlukan juga nilai beberapa parameter terkait proses produksi yang dilakukan pada perusahaan tersebut seperti ditunjukkan pada Tabel 4.2.

Tabel 4.2 Parameter dalam Proses Produksi

Parameter	Keterangan
<i>n</i>	Jumlah pekerja
<i>rt</i>	Jumlah produksi rata-rata tiap pekerja dalam 1 hari (<i>regular time</i>)
<i>ot</i>	Jumlah produksi rata-rata tiap pekerja dalam 1 hari (<i>overtime</i>)
<i>sc</i>	Jumlah produksi rata-rata tiap pekerja dalam 1 hari (<i>subcontract</i>)
<i>jrt</i>	Jam kerja dalam 1 hari (<i>regular time</i>)
<i>jot</i>	Jam lembur maksimal (<i>overtime</i>)
<i>msc</i>	Jumlah maksimal pekerja <i>subcontract</i>
<i>brt</i>	Biaya produksi <i>regular time</i>
<i>bot</i>	Biaya produksi <i>overtime</i>
<i>bsc</i>	Biaya produksi <i>subcontract</i>
<i>rp</i>	Jumlah pekerja baru yang direkrut (<i>hiring</i>)
<i>prp</i>	Jumlah produksi rata-rata tiap pekerja baru dalam 1 hari
<i>brp</i>	Biaya merekrut pekerja
<i>mp</i>	Jumlah pekerja yang dirumahkan
<i>bmp</i>	Biaya merumahkan pekerja (<i>firing</i>)
<i>kp</i>	Jumlah kekurangan produksi
<i>i</i>	Jumlah produk pada penyimpanan (<i>inventory</i>)
<i>bi</i>	Biaya penyimpanan tiap produk

Adapun nilai parameter-parameter tersebut akan berbeda pada tiap perusahaan. Pada perusahaan PT. X ini nilai awal parameter-parameter tersebut ditunjukkan pada Tabel 4.3.

Tabel 4.3 Nilai Parameter Proses Produksi

Parameter	Nilai Parameter
<i>n</i>	10
<i>rt</i>	17
<i>ot</i>	5
<i>sc</i>	15
<i>jrt</i>	8
<i>jot</i>	3
<i>msc</i>	3
<i>brt</i>	Rp. 23.000,-
<i>bot</i>	Rp. 28.000,-
<i>bsc</i>	Rp. 30.000,-
<i>rp</i>	12
<i>brp</i>	Rp. 2.700.000,-

$$\frac{b_{mp}}{b_i} = \frac{\text{Rp. 500.000,-}}{\text{Rp. 500,-}}$$

Langkah awal yang perlu dilakukan dalam melakukan perencanaan produksi agregat adalah melakukan peramalan permintaan periode berikutnya terlebih dahulu berdasarkan data periode saat ini. Penulis menggunakan regresi linier sederhana sebagai contoh perhitungan peramalan. Tahapan pada proses peramalan permintaan menggunakan regresi linier sederhana adalah sebagai berikut:

1. Menghitung nilai a dan b berdasarkan data pada periode awal.

Tabel 4.4 Perhitungan Nilai a dan b untuk Peramalan

Periode	t	y	t.y	t ²
Agt-15	1	872	872	1
Sep-15	2	811	1622	4
Okt-15	3	980	2940	9
Nov-15	4	741	2964	16
Des-15	5	1091	5455	25
Jan-16	6	572	3432	36
Feb-16	7	977	6839	49
Mar-16	8	2026	16208	64
Apr-16	9	2597	23373	81
Mei-16	10	2738	27380	100
Jun-16	11	7228	79508	121
Jul-16	12	4945	59340	144
Jumlah	78	25578	229933	650
	Σt	Σy	Σ t.y	Σt²

Data pada 12 bulan awal diasumsikan menjadi periode (t) 1 hingga 12. Kemudian dilakukan proses perhitungan sehingga didapatkan Σt, Σy, Σ t.y, Σt². Setelah didapatkan keempat nilai tersebut selanjutnya menggunakan rumus ditunjukkan pada Persamaan (4-2) dan Persamaan (4-3) untuk mendapatkan nilai a dan b. Variabel n pada Persamaan tersebut merupakan jumlah t yang diperhitungkan sebelumnya.

Persamaan 4.2 Rumus Nilai b pada Regresi Linier

$$b = \frac{n \sum(t.y) - \sum t \sum y}{n \sum t^2 - (\sum t)^2} \quad (4-2)$$

$$b = \frac{(12 \cdot 229933) - (78 \cdot 25578)}{(12 \cdot 650) - (78)^2} = 445,28671$$

Persamaan 4.3 Rumus Nilai a pada Regresi Linier

$$a = \frac{\sum y - b \sum t}{n} \quad (4-3)$$

$$a = \frac{25578 - 445,28671 \cdot 78}{12} = -762,8636$$

2. Setelah mendapatkan nilai *a* dan *b*, kemudian kedua nilai tersebut dimasukkan ke dalam persamaan linier sederhana ditunjukkan pada Persamaan (4-4). Variabel *y'* merupakan hasil peramalan yang dihasilkan oleh regresi linier sederhana. Hasil peramalan tersebut ditunjukkan pada Tabel 4.5.

Persamaan 4.4 Persamaan Regresi Linier Sederhana

$$F(t) = y' = a + bt \quad (4-4)$$

Tabel 4.5 Hasil Peramalan Regresi Linier Sederhana

Periode	t	y	y'	y'
Agt-16	13	872	5025.864	5026
Sep-16	14	811	5471.15	5471
Okt-16	15	980	5916.437	5916
Nov-16	16	741	6361.724	6362
Des-16	17	1091	6807.01	6807
Jan-17	18	572	7252.297	7252
Feb-17	19	977	7697.584	7698
Mar-17	20	2026	8142.871	8143
Apr-17	21	2597	8588.157	8588
Mei-17	22	2738	9033.444	9033
Jun-17	23	7228	9478.731	9479
Jul-17	24	4945	9924.017	9924

3. Hasil peramalan tersebut dapat dianalisis untuk didapatkan akurasinya. Untuk mengetahuinya dapat menggunakan Mean Absolute Deviation (MAD), Mean Square Error (MSE) maupun Mean Absolute Percentage Error (MAPE). Namun, pada penelitian tidak dilakukan analisis hasil peramalan karena fokus penulis hanya sampai mendapatkan hasil peramalan permintaan produk periode berikutnya.

Setelah mengetahui hasil peramalan permintaan produk pada periode berikutnya, maka selanjutnya dapat dilanjutkan dengan melakukan perencanaan produksi agregat. Agar dapat lebih mudah dipahami, penulis melakukan perencanaan produksi seperti ditunjukkan pada Tabel 4.6.

Tabel 4.6 Perencanaan Produksi Agregat

Periode	Permin-taan	Hari Kerja	Produksi	<i>rt</i>	<i>ot</i>	<i>sc</i>	<i>kp</i>	<i>rp</i>	<i>mp</i>	<i>i</i>
Agt-16	5026	21	8690	3570	1050	3150	920	4		3664
Sep-16	5471	20	7856	7856						2385
Okt-16	5916	21	5771	5771						-145
Nov-16	6362	21	5801	4998	803					-561
Des-16	6807	21	7858	4998	1470	1390				1051
Jan-17	7252	20	8189	5811	1400	978				937
Feb-17	7698	22	9292	6173	1540	1579				1594
Mar-17	8143	20	6416	6354	62					-1727
Apr-17	8588	20	6860	4760	1400	700				-1728
Mei-17	9033	17	9951	4046	1190	3570	1145	6		918
Jun-17	9479	21	5410	5410						-4069
Jul-17	9924	22	6148	6148						-3776

Pada kolom produksi berisi nilai perkiraan produksi yang didapatkan secara *random*. Nilai ini yang nantinya akan ditentukan jumlahnya menggunakan metode optimasi agar didapatkan hasil yang optimal dengan biaya minimum. Nilai produksi tersebut dimasukkan hingga tak tersisa ke dalam beberapa parameter produksi yakni *regular time*, *overtime*, *subcontract*, merekrut pekerja, merumahkan pekerja dan juga *inventory*. Sebelumnya perlu diketahui bahwa setiap parameter memiliki nilai maksimal berdasarkan nilai awal yang telah ditentukan sebelumnya. Sebagai contoh, untuk nilai maksimal pada *regular time* (*rt*) tersebut didapatkan dari perkalian antara jumlah pekerja, jumlah produksi rata-rata tiap pekerja dalam 1 hari pada *regular time* dan juga jumlah hari kerja pada bulan terkait. Sehingga didapatkan nilai maksimal untuk *rt* awal adalah 3570. Sedangkan nilai maksimal produksi *overtime* (*ot*) dan *subcontract* (*sc*) adalah 1050 dan 3150. Jika nilai acak produksi tersebut masih tersisa setelah masuk ke dalam 3 fase sebelumnya, maka langkah selanjutnya adalah melakukan perekrutan pekerja baru. Setiap merekrut pekerja baru juga diperhitungkan nilai maksimal yang berasal dari perkalian jumlah produksi rata-rata pekerja baru dalam 1 hari dengan hari kerja. Untuk jumlah pekerja yang akan direkrut dihitung berdasarkan sisa dari nilai acak produksi yang dibagi dengan nilai maksimal produksi pekerja baru. Nilai *inventory* (*i*) merupakan sisa dari jumlah produksi yang berasal dari nilai acak produksi dikurangi dengan hasil peramalan permintaan. Nilai *i* yang minus dianggap merupakan suatu kerugian dan diasumsikan setiap minus tersebut dikalikan dengan Rp. 30.000,-.

Nilai produksi pada baris pertama adalah 8690. Nilai tersebut dimasukkan ke dalam rt terlebih dahulu sebesar nilai maksimalnya yakni 3570 hingga menyisakan 5120. Nilai 5120 ini dimasukkan ke fase berikutnya ot sebesar nilai maksimal yakni 1050 dan masih menyisakan 4070. Kemudian nilai dilanjutkan masuk ke fase sc sebesar nilai maksimalnya yakni 3150. Dikarenakan masih ada sisa nilai sebesar 920, maka proses dilanjutkan ke fase perekrutan pekerja baru. Nilai produksi maksimal tiap pekerja baru pada baris pertama adalah sebesar 252 yang didapatkan dari perkalian antara jumlah produksi rata-rata pekerja baru dalam 1 hari sebesar 12 dengan hari kerja pada bulan tersebut yakni Bulan September 2016. Untuk menentukan jumlah pekerja baru yang akan direkrut dapat dihitung berdasarkan sisa nilai 920 tadi dibagi dengan nilai maksimal 252 sehingga jika dibulatkan akan didapatkan hasil 4 yang berarti merekrut pekerja baru sebanyak 4 orang. Nilai i pada baris pertama tersebut didapatkan dari nilai acak produksi yakni 8690 dikurangi dengan nilai hasil peramalan permintaan sebesar 5026, yang nantinya nilai i ini akan berpengaruh terhadap proses selanjutnya. Sehingga proses selanjutnya tidak perlu nilai maksimal karena sudah ditambahkan dengan nilai i pada proses sebelumnya. Hal tersebut dilakukan terus menerus secara berurutan hingga baris akhir yakni perencanaan produksi pada Bulan Agustus 2017.

Untuk menentukan apakah nilai acak tersebut optimal atau tidak berdasarkan total biaya produksi (TBP) yang dikeluarkan ditunjukkan pada Persamaan (4-5) dan juga beberapa *penalty* yang dihasilkan yakni nilai i menjadi minus atau dianggap sebuah kerugian dan PT.X menentukan nilai kerugian tersebut sebesar Rp. 30.000,- per produk. Maka diperlukan suatu pengukuran kualitas nilai tersebut yang selanjutnya disebut sebagai nilai *fitness* (NF) seperti ditunjukkan pada Persamaan (4-1).

Persamaan 4.5 Total Biaya Produksi

$$TBP = (\sum rt . brt) + (\sum ot . bot) + (\sum sc . bsc) + (\sum rp . brp) + (\sum mp . bmp) + (\sum i . bi) \quad (4-5)$$

$$TBP = 65895.23000 + 8915.28000 + 11367.30000 + 10.2700000 + 0 + 10549.500 = 2.138.489.500$$

$$NF = \frac{1000000000}{TBP + \sum Penalty}$$

$$NF = \frac{1000000000}{2138489500 + 360180000} = \frac{1000000000}{2498669500} = 0,4002129$$

4.1.8 Contoh Solusi menggunakan Algoritme Genetika

Penerapan Algoritme Genetika untuk menyelesaikan permasalahan adalah dengan mengetahui nilai prediksinya terlebih dahulu selama 12 bulan atau satu periode berikutnya menggunakan regresi linier sederhana. Pada cara

sebelumnya terdapat suatu tabel dengan nama kolom "Produksi" dan pada Algoritme Genetika kolom tersebut ditransformasikan menjadi kromosom. Langkah awal yang dilakukan yakni membangkitkan beberapa individu untuk populasi awal pada generasi pertama secara *random*. Pada contoh ini, penulis akan mengasumsikan ukuran populasi awal sejumlah 5 individu dengan panjang kromosom sebanyak 12 sesuai dengan banyaknya bulan dalam 1 periode.

Setiap individu tersebut dihitung nilai *fitness*nya guna mengetahui kualitas setiap individunya. Langkah selanjutnya adalah melakukan proses reproduksi untuk menghasilkan individu baru. Pada Algoritme Genetika terdapat 2 cara reproduksi yakni pindah silang (*crossover*) dan mutasi (*mutation*). Terdapat parameter rasio perbandingan bagian antara banyaknya jumlah untuk *crossover* (*Crossover Rate*) dan banyaknya jumlah untuk mutasi (*Mutation Rate*) yang awalnya ditentukan secara *random*. Saat *crossover* dibutuhkan 2 individu sebagai induk untuk menghasilkan individu baru. Pada penelitian ini penulis menggunakan *one-cut-point crossover* pada tiap segmennya.

Kemudian selanjutnya dilakukan proses reproduksi mutasi berbasis posisi dengan sedikit modifikasi pada tiap segmennya. Saat mutasi hanya diperlukan satu individu sebagai induk untuk menghasilkan individu baru.

Setelah menghasilkan individu-individu baru tersebut individu sebagai induk dan juga individu baru sebagai anak digabung untuk ditampung dalam *offspring*, kemudian dilanjutkan dengan proses evaluasi. Pada saat evaluasi inilah dinilai kualitas setiap individu dengan menggunakan perhitungan nilai *fitness*. Semakin besar nilai *fitness*nya maka individu tersebut semakin baik pula. Individu dinilai dengan kualitas baik yang dimaksud adalah minimnya biaya produksi, minimnya selisih produksi dengan prediksi dan juga minimnya *constraint* yang terjadi, seperti jam kerja para pekerjanya, jumlah produksi setiap pekerjanya dan juga lain sebagainya sesuai peraturan yang berlaku di perusahaan.

Seluruh nilai *fitness* individu yang terdapat pada *offspring* akan diseleksi untuk dilanjutkan pada generasi berikutnya. Seleksi yang digunakan oleh penulis yakni *elistism* guna mempersingkat waktu komputasi. Semua individu akan diurutkan berdasarkan nilai *fitness* terbesar hingga terkecil. Setelah diurutkan tersebut kemudian diambil beberapa individu sejumlah ukuran populasi di awal. Individu-individu terpilih tersebut akan menjadi populasi awal pada generasi berikutnya.

Dengan menerapkan Algoritme Genetika dihasilkan individu terbaik seperti ditunjukkan pada Tabel 4.7.

Tabel 4.7 Contoh Solusi Menggunakan Algoritme Genetika

I	Kromosom												NF
P1	5262	6318	5352	5464	6547	7564	6464	7472	7868	9765	7475	4367	0,4931
P2	6839	7543	7759	8651	7535	8783	3674	7575	4657	6473	7348	9660	0,4628
P3	7538	7937	9352	8735	9279	8472	7292	9811	7481	8420	8290	7025	0,3712
P4	8204	9275	8905	9053	8730	8392	8174	8912	8793	8937	9385	9829	0,4563
P5	9821	9820	7577	8685	5785	4734	7543	5737	4757	5474	3637	8453	0,4045

4.1.9 Contoh Solusi menggunakan Algoritme Genetika Adaptif

Penerapan Algoritme Genetika Adaptif pada dasarnya sama seperti penerapan Algoritme Genetika biasa, hanya terdapat perbedaan pada saat proses reproduksinya. Nilai *crossover rate* dan *mutation rate* pada Algoritme Genetika Adaptif ini akan terus berubah menyesuaikan kondisi yang terjadi. Jika dalam beberapa generasi tidak didapatkan solusi yang lebih optimal, maka kedua nilai parameter tersebut akan berubah secara *random*.

Dengan menerapkan Algoritme Genetika Adaptif dihasilkan individu terbaik seperti ditunjukkan pada Tabel 4.8

Tabel 4.8 Contoh Solusi Menggunakan Algoritme Genetika Adaptif

I	Kromosom												NF
P1	5367	6013	6446	4757	7546	4361	4637	4907	7895	6436	8906	6787	0,4893
P2	7463	8477	9057	8543	7152	7463	6352	6895	6024	7390	6903	9660	0,4902
P3	6245	8701	8527	6789	8971	8721	8424	6703	6899	8094	7432	8790	0,4465
P4	9754	8667	8467	8924	8909	9097	8345	7895	8745	9084	5673	6987	0,4719
P5	7870	6789	7603	7945	8924	8723	7543	8462	8905	9024	8793	8326	0,4826

4.1.10 Contoh Solusi menggunakan Simulated Annealing

Dalam penerapan metode *Simulated Annealing*, diperlukan solusi awal sepanjang 12 yang menunjukkan jumlah produksi selama 12 bulan dalam 1 periode. Serta dihitung pula nilai objektif atau selanjutnya akan disamakan menjadi nilai *fitness* dari solusi awal tersebut. Pada *Simulated Annealing*, terdapat beberapa parameter yang perlu dipenuhi pertama kali diantaranya: jumlah solusi baru setiap kali iterasi, nilai parameter kontrol atau temperatur (T) awal, nilai faktor reduksi (α) dan juga jumlah iterasi maksimal yang akan dilakukan. Pada contoh ini, penulis menentukan parameter awal yakni sebagai berikut:

- Jumlah solusi baru = 3
- Temperatur (T) awal = 125
- Faktor reduksi (α) = 0,45
- Iterasi maksimal = 2

Untuk lebih memudahkan, penulis memberikan alur contoh perhitungannya. Adapun contoh solusi awal yang ditentukan oleh penulis seperti ditunjukkan Gambar 4.15.

7890	7935	8167	8013	8467	8909	9276	9238	9012	8975	9024	8721
------	------	------	------	------	------	------	------	------	------	------	------

Gambar 4.15 Contoh Solusi Simulated Annealing

Setelah menentukan solusi awal, kemudian dihitung nilai *fitness*nya berdasarkan rumus yang telah dijelaskan sebelumnya. Kemudian dilanjutkan dengan membangkitkan calon-calon solusi baru secara acak.

Pada iterasi pertama dengan nilai parameter awal yang telah ditentukan diatas, membangkitkan 12 nilai secara acak sebagai 1 calon solusi. Dikarenakan di awal telah ditentukan bahwa jumlah solusi baru sebanyak 3, maka membangkitkan nilai tersebut sebanyak 3 kali serta dihitung pula nilai *fitness*nya. Dari ketiga nilai *fitness* calon solusi baru, dipilih 1 nilai *fitness* terbaik. Nilai *fitness* terbaik nantinya akan dibandingkan dengan nilai *fitness* pada solusi awal. Terdapat aturan baku jika nilai *fitness* calon solusi baru lebih besar dari pada nilai *fitness* solusi awal, maka solusi tersebut akan diterima. Jika solusi telah diterima, maka nilai temperatur akan berubah dikalikan dengan faktor reduksi. Jika nilai *fitness* calon solusi baru tidak lebih besar dari pada nilai *fitness* solusi awal, maka solusi tersebut akan dipertimbangkan. Solusi yang masih dipertimbangkan tersebut dihitung selisih nilai *fitness*nya. Jika selisih nilai *fitness* lebih besar dari 0, maka solusi tersebut diterima. Jika selisih nilai *fitness* kurang dari sama dengan 0, maka solusi tersebut akan dipertimbangkan kembali. Solusi yang masih dipertimbangkan tersebut dilanjutkan untuk dihitung probabilitasnya.

4.2 Perancangan Uji Coba dan Evaluasi

Pada bagian ini akan dijelaskan skenario uji coba untuk setiap metode. Uji coba penting dilakukan untuk mengetahui nilai parameter metode terbaik dalam menghasilkan solusi yang optimal. Perancangan ini mencakup skenario uji coba semua parameter pada Algoritme Genetika, Algoritme Genetika Adaptif dan juga Algoritme Genetika dengan *Simulated Annealing*. Selain itu juga dilakukan perbandingan hasil dengan metode pembandingan lainnya, antara lain: *Simulated Annealing*, Algoritme Genetika dan Algoritme Genetika Adaptif.

4.2.1 Uji Coba Ukuran Populasi

Ukuran populasi ini cukup berpengaruh dalam menghasilkan solusi, karena semakin besar ukuran populasi maka kromosom yang dibangkitkan pada populasi akan semakin beragam. Namun jika ukuran populasi terlampaui besar,

maka akan mengakibatkan waktu eksekusi yang cukup lama. Oleh sebab itu diperlukannya skenario uji coba ini untuk mengetahui ukuran populasi yang tepat dalam menghasilkan solusi optimal seperti ditunjukkan pada Tabel 4.9.

Pada uji coba ukuran populasi ini digunakan nilai kelipatan 10. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Ukuran Populasi : 10-100
- b. Jumlah Generasi : 1000
- c. Crossover Rate : 0,5
- d. Mutation Rate : 0,5

Tabel 4.9 Skenario Uji Coba Ukuran Populasi

Ukuran Populasi	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											

4.2.2 Uji Coba Jumlah Generasi

Uji coba jumlah generasi ini bertujuan untuk mengetahui jumlah generasi yang tepat dalam menghasilkan solusi optimal. Pada uji coba jumlah generasi ini akan menggunakan jumlah dengan kelipatan 200 seperti ditunjukkan pada Tabel 4.10. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Ukuran Populasi : Hasil uji coba ukuran populasi terbaik
- b. Jumlah Generasi : 200-2000
- c. Crossover Rate : 0,5
- d. Mutation Rate : 0,5

Tabel 4.10 Skenario Uji Coba Jumlah Generasi

Jumlah Generasi	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
200											

400												
600												
800												
1000												
1200												
1400												
1600												
1800												
2000												

4.2.3 Uji Coba Kombinasi Crossover Rate dan Mutation Rate

Uji coba kombinasi *crossover rate* (*cr*) dan *mutation rate* (*mr*) bertujuan untuk mengetahui kombinasi yang tepat dalam menghasilkan solusi optimal. Uji coba yang dilakukan yaitu dengan menggunakan kombinasi 2 nilai *cr* dan *mr* dimana jumlah dari kedua nilai tersebut sama dengan 1 seperti ditunjukkan pada Tabel 4.11. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Ukuran Populasi : Hasil uji coba ukuran populasi terbaik
- b. Jumlah Generasi : Hasil uji coba jumlah generasi terbaik
- c. Crossover Rate : 0,1 – 0,9
- d. Mutation Rate : 0,1 – 0,9

Tabel 4.11 Skenario Uji Coba Kombinasi *cr* dan *mr*

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

4.2.4 Uji Coba Jumlah Solusi Baru

Uji coba jumlah solusi baru dilakukan untuk mengetahui jumlah terbaik dan juga tepat solusi baru yang dibangkitkan. Pada uji coba ini menggunakan nilai untuk mewakili jumlah solusi baru diatas 1 dan kurang dari sama dengan 10 karena penulis mempertimbangkan waktu komputasi yang akan berjalan seperti

ditunjukkan pada Tabel 4.12. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Jumlah solusi baru : 2 – 15
- b. Temperatur (T) : 100
- c. Faktor reduksi (α) : 0,5
- d. Iterasi Maksimal : 100

Tabel 4.12 Skenario Uji Coba Jumlah Solusi Baru

Jumlah Solusi Baru	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											

4.2.5 Uji Coba Nilai Temperatur

Uji coba nilai temperatur dilakukan untuk mengetahui temperatur awal yang optimal dalam mempengaruhi solusi yang dihasilkan. Penulis menggunakan nilai dengan kelipatan 25 seperti ditunjukkan pada Tabel 4.13. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Jumlah solusi baru : Hasil uji coba jumlah solusi baru terbaik
- b. Temperatur (T) : 25 – 200
- c. Faktor reduksi (α) : 0,5
- d. Iterasi Maksimal : 100

Tabel 4.13 Skenario Uji Coba Nilai Temperatur

Nilai Temperatur	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
25											
50											

75											
100											
125											
150											
175											
200											

4.2.6 Uji Coba Nilai Faktor Reduksi

Uji coba nilai faktor reduksi dilakukan untuk mengetahui nilai α yang optimal dalam mempengaruhi solusi yang dihasilkan. Penulis menggunakan nilai mulai dari 0,1 hingga 0,9 seperti ditunjukkan pada Tabel 4.14. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Jumlah solusi baru : Hasil uji coba jumlah solusi baru terbaik
- b. Temperatur (T) : Hasil uji coba nilai temperatur terbaik
- c. Faktor reduksi (α) : 0,1 – 0,9
- d. Iterasi Maksimal : 100

Tabel 4.14 Skenario Uji Coba Nilai Faktor Reduksi

Faktor Reduksi	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
0,1											
0,2											
0,3											
0,4											
0,5											
0,6											
0,7											
0,8											
0,9											

4.2.7 Uji Coba Jumlah Iterasi Maksimal

Uji coba jumlah iterasi dilakukan untuk mengetahui seberapa banyak diperlukan iterasi dalam menghasilkan solusi optimal. Penulis memulai dari iterasi sebesar 200 hingga 2000 seperti ditunjukkan pada Tabel 4.15. Adapun nilai parameter awal yang digunakan untuk uji coba adalah sebagai berikut:

- a. Jumlah solusi baru : Hasil uji coba jumlah solusi baru terbaik
- b. Temperatur (T) : Hasil uji coba nilai temperatur terbaik
- c. Faktor reduksi (α) : Hasil uji coba nilai faktor reduksi terbaik
- d. Iterasi Maksimal : 200 – 2000

Tabel 4.15 Skenario Uji Coba Jumlah Iterasi Maksimal

Iterasi Maksimal	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
200											
400											
600											
800											
1000											
1200											
1400											
1600											
1800											
2000											

4.2.8 Uji Coba Perbandingan Metode

Uji coba ini dilakukan dengan cara membandingkan hasil dari metode usulan yakni HSAAGA dengan metode pembanding lainnya yakni Algoritme Genetika, Algoritme Genetika Adaptif dan juga *Simulated Annealing*. Setiap metode akan dieksekusi sebanyak 10 kali kemudian diambil nilai *fitness* rata-rata seperti ditunjukkan pada Tabel 4.16.

Tabel 4.16 Skenario Uji Coba Perbandingan Metode

Metode	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
Algoritme Genetika											
Algoritme Genetika Adaptif											
<i>Simulated Annealing</i>											
Hibridisasi <i>Simulated Annealing</i> – Algoritme Genetika Adaptif (HSAAGA)											

BAB 5 HASIL DAN PEMBAHASAN

5.1 Hasil Uji Coba Metode Algoritme Genetika

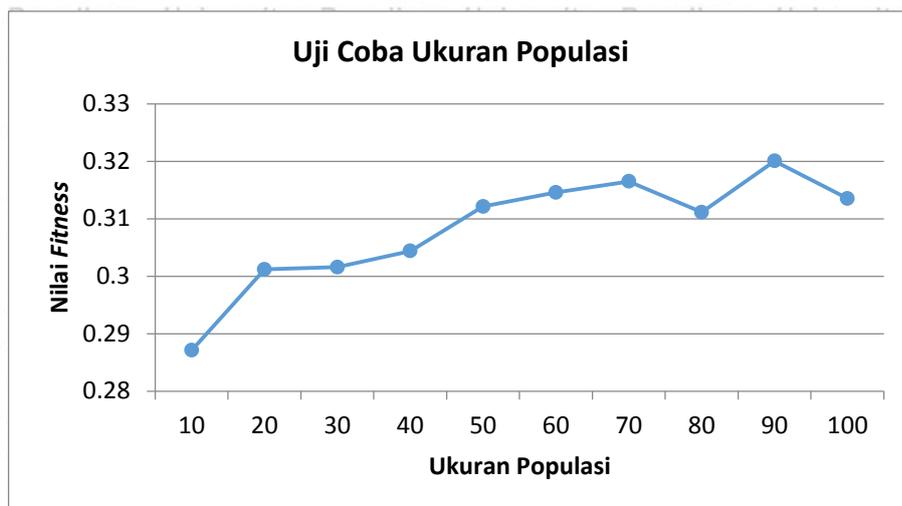
Hasil uji coba metode pertama yakni Algoritme Genetika akan dijelaskan pada sub-bab ini.

5.1.1 Hasil Uji Coba Ukuran Populasi

Hasil uji coba ukuran populasi pada Algoritme Genetika dapat dilihat pada Tabel 5.1. Untuk melakukan uji coba ukuran populasi, penulis menggunakan beberapa parameter yang telah ditentukan di awal yakni jumlah generasi = 1000, $cr = 0,5$ dan $mr = 0,5$.

Tabel 5.1 Hasil Uji Coba Ukuran Populasi

Ukuran Populasi	Nilai Fitness										
	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
10	0,2993	0,2885	0,2994	0,2715	0,2903	0,2687	0,2830	0,2941	0,2843	0,2913	0,2871
20	0,3030	0,2985	0,2924	0,2959	0,2991	0,3067	0,2989	0,2943	0,3055	0,3171	0,3011
30	0,2976	0,3047	0,2887	0,3004	0,3040	0,2985	0,3114	0,3095	0,3073	0,2933	0,3015
40	0,3175	0,3104	0,3200	0,2953	0,2886	0,2909	0,3091	0,3053	0,2907	0,3155	0,3043
50	0,3081	0,3154	0,3148	0,3258	0,3125	0,3081	0,3069	0,3083	0,3266	0,2943	0,3121
60	0,2951	0,3152	0,3137	0,3211	0,3178	0,3117	0,3185	0,3077	0,3227	0,3219	0,3145
70	0,3156	0,3159	0,3094	0,3219	0,3162	0,3164	0,3187	0,3110	0,3111	0,3282	0,3164
80	0,3049	0,3104	0,3053	0,3048	0,3181	0,3056	0,3133	0,3219	0,3124	0,3141	0,3111
90	0,3118	0,3179	0,3219	0,3198	0,3183	0,3215	0,3321	0,3297	0,3048	0,3222	0,3200
100	0,3242	0,3102	0,3232	0,2860	0,3287	0,3300	0,3316	0,2918	0,3026	0,3063	0,3135



Gambar 5.1 Grafik Hasil Uji Coba Ukuran Populasi

Jika dilihat pada Gambar 5.1, titik tertinggi terdapat pada ukuran populasi sebesar 90 dengan rata-rata nilai *fitness*nya 0,3200. Namun, hal ini tidak dapat disimpulkan bahwa semakin besar ukuran populasi maka semakin meningkat pula nilai *fitness*nya. Hal tersebut dapat dibuktikan pada saat ukuran populasi sebesar 80 dan 100 terjadi sedikit penurunan. Pola tersebut akan kemungkinan berulang pada generasi berikutnya. Penulis tidak melakukan uji coba untuk ukuran populasi diatas 100 karena hanya akan menambah waktu komputasi sehingga penulis memutuskan untuk melakukan uji coba ukuran populasi maksimal sebesar 100.

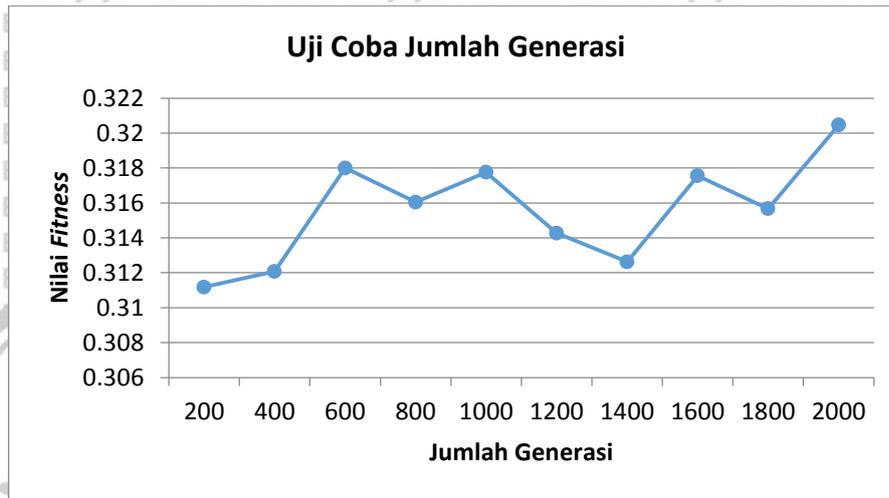
5.1.2 Hasil Uji Coba Jumlah Generasi

Hasil uji coba jumlah generasi pada Algoritme Genetika dapat dilihat pada Tabel 5.2. Uji coba jumlah generasi dilakukan untuk mengetahui generasi terbaik pada saat mencapai titik optimal atau terjadinya konvergensi dini. Konvergensi dini terjadi pada saat generasi setelahnya tidak terdapat kenaikan nilai yang signifikan. Untuk melakukan uji coba jumlah generasi, penulis menggunakan beberapa parameter yang telah ditentukan di awal yakni $cr = 0,5$ dan $mr = 0,5$. Sedangkan untuk ukuran populasinya menggunakan hasil terbaik yang didapat pada uji coba sebelumnya yakni sebesar 90.

Tabel 5.2 Hasil Uji Coba Jumlah Generasi

Jumlah Generasi	Nilai Fitness										
	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
200	0,2992	0,3040	0,2999	0,3075	0,3240	0,3171	0,3183	0,3066	0,3200	0,3148	0,3111
400	0,3074	0,3083	0,3116	0,3161	0,3090	0,3087	0,3181	0,3176	0,3136	0,3100	0,3120

600	0,3163	0,3124	0,3244	0,3173	0,3132	0,3189	0,3072	0,3269	0,3245	0,3183	0,3180
800	0,3268	0,3290	0,3096	0,3217	0,3140	0,3106	0,3190	0,2982	0,3165	0,3146	0,3160
1000	0,3142	0,3173	0,3073	0,3143	0,3154	0,3240	0,3261	0,3339	0,3122	0,3126	0,3177
1200	0,3020	0,3128	0,3178	0,3113	0,3085	0,3174	0,3115	0,3341	0,3228	0,3041	0,3142
1400	0,3109	0,3045	0,3182	0,3039	0,3145	0,3140	0,3924	0,3267	0,3297	0,3110	0,3126
1600	0,3280	0,3228	0,3252	0,3067	0,3141	0,3128	0,3253	0,3169	0,3096	0,3137	0,3175
1800	0,3130	0,3060	0,3254	0,3112	0,3074	0,3329	0,3288	0,3051	0,3039	0,3225	0,3156
2000	0,3185	0,3317	0,3112	0,3069	0,3049	0,3262	0,3269	0,3307	0,3154	0,3318	0,3204



Gambar 5.2 Grafik Hasil Uji Coba Jumlah Generasi

Jika dilihat pada Gambar 5.2, titik tertinggi terjadi pada saat generasi 2000. Pola yang terlihat pada generasi sebelumnya fluktuatif. Hal tersebut menunjukkan bahwa semakin banyak generasi tidak menjamin akan menemukan solusi yang lebih optimal.

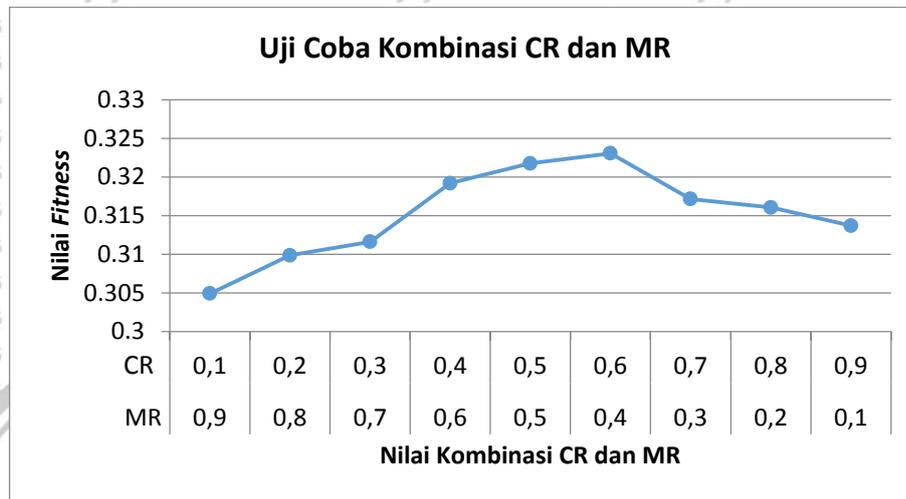
5.1.3 Hasil Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Setelah mengetahui ukuran populasi terbaik sebesar 90 dan jumlah generasi terbaik sebesar 2000, maka dilanjutkan dengan melakukan uji coba kombinasi *cr* dan *mr*. Hasil uji coba kombinasi tersebut dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil Uji Coba Kombinasi CR dan MR

Kombinasi	Nilai Fitness											
	Uji Coba ke-											Rata-rata
<i>cr</i>	<i>mr</i>	1	2	3	4	5	6	7	8	9	10	
0,1	0,9	0,2998	0,3085	0,3057	0,3100	0,3070	0,3016	0,3065	0,3093	0,2966	0,3037	0,3049
0,2	0,8	0,3051	0,3051	0,3214	0,2960	0,3059	0,3253	0,3322	0,3111	0,2969	0,2990	0,3098
0,3	0,7	0,3031	0,3130	0,3078	0,3200	0,3137	0,3165	0,3215	0,3128	0,2956	0,3116	0,3116

0,4	0,6	0,3164	0,3117	0,3346	0,3101	0,3292	0,3265	0,3210	0,3066	0,3113	0,3242	0,3191
0,5	0,5	0,3170	0,3192	0,3296	0,3154	0,3052	0,3211	0,3292	0,3241	0,3204	0,3361	0,3217
0,6	0,4	0,3286	0,3253	0,3213	0,3202	0,3241	0,3216	0,3168	0,3185	0,3294	0,3242	0,3230
0,7	0,3	0,3131	0,3306	0,3218	0,3191	0,3268	0,3025	0,3336	0,3301	0,2848	0,3087	0,3171
0,8	0,2	0,3065	0,3229	0,3288	0,3038	0,3159	0,3194	0,3093	0,3116	0,3137	0,3282	0,3160
0,9	0,1	0,3163	0,3157	0,3158	0,3225	0,3066	0,3019	0,3149	0,3208	0,3112	0,3110	0,3137



Gambar 5.3 Grafik Hasil Uji Coba Kombinasi CR dan MR

Seperti yang ditunjukkan Gambar 5.3, titik tertinggi dihasilkan pada saat kombinasi $cr = 0,6$ dan $mr = 0,4$. Dengan kombinasi tersebut, dapat disimpulkan bahwa proses *crossover* lebih berperan pada saat melakukan reproduksi untuk menghasilkan calon solusi baru. Hal tersebut juga menunjukkan bahwa adanya modifikasi yang dilakukan oleh penulis pada proses *crossover* cukup efektif karena dapat mencapai seluruh kemungkinan solusi yang lebih optimal sehingga menghasilkan nilai *fitness* yang tinggi.

5.2 Hasil Uji Coba Metode Algoritme Genetika Adaptif

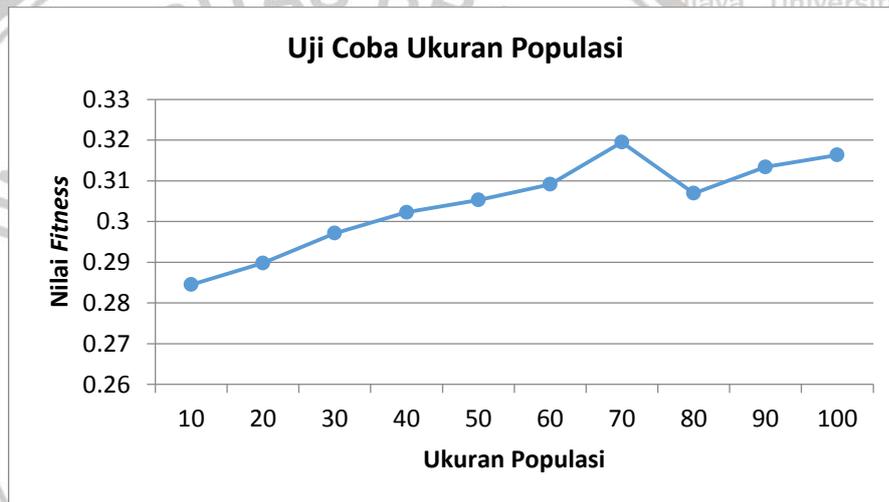
Hasil uji coba metode kedua yakni Algoritme Genetika Adaptif akan dijelaskan pada sub-bab ini.

5.2.1 Hasil Uji Coba Ukuran Populasi

Hasil uji coba ukuran populasi pada Algoritme Genetika dapat dilihat pada Tabel 5.4. Untuk melakukan uji coba ukuran populasi, penulis menggunakan beberapa parameter yang telah ditentukan di awal yakni jumlah generasi = 1000, dan kombinasi awal $cr = 0,5$ dan $mr = 0,5$.

Tabel 5.4 Hasil Uji Coba Ukuran Populasi

Ukuran Populasi	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
10	0,2826	0,2821	0,2914	0,2839	0,2660	0,2976	0,2910	0,2807	0,2816	0,2878	0,2845
20	0,2982	0,2960	0,3011	0,2967	0,3007	0,2698	0,2554	0,2866	0,3060	0,2870	0,2898
30	0,2884	0,2972	0,2859	0,3106	0,2799	0,2911	0,3012	0,3112	0,3025	0,3029	0,2971
40	0,3026	0,3182	0,3150	0,2908	0,3006	0,3011	0,2870	0,3175	0,2755	0,3142	0,3022
50	0,3041	0,2971	0,3024	0,3169	0,2918	0,308	0,3130	0,3056	0,2987	0,3141	0,3053
60	0,3058	0,3150	0,3141	0,3006	0,3097	0,3011	0,3120	0,3111	0,3145	0,3069	0,3091
70	0,3261	0,3047	0,3202	0,3246	0,3079	0,3364	0,3058	0,3249	0,3375	0,3064	0,3194
80	0,3064	0,3165	0,3092	0,3165	0,3048	0,3121	0,3037	0,3011	0,2913	0,3074	0,3069
90	0,3043	0,3059	0,3317	0,3291	0,3023	0,2972	0,3011	0,3304	0,3028	0,3287	0,3134
100	0,3147	0,3250	0,3232	0,3184	0,3122	0,3133	0,3150	0,3089	0,3339	0,2982	0,3163



Gambar 5.4 Grafik Hasil Uji Coba Ukuran Populasi

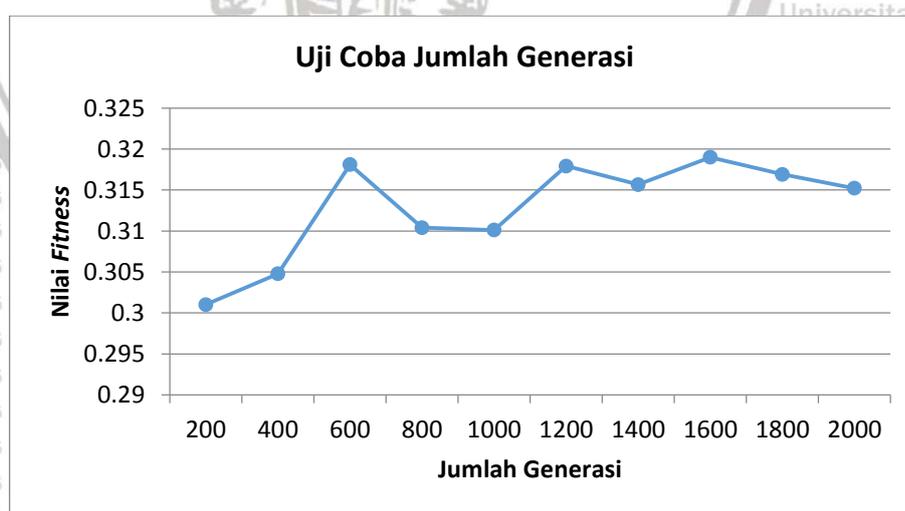
Jika dilihat pada Gambar 5.4, titik tertinggi terletak pada ukuran populasi sebesar 70. Terjadi penurunan yang cukup signifikan pada satu titik setelah populasi sebesar 70, namun kemudian terjadi peningkatan kembali. Pola tersebut sangat memungkinkan apabila ukuran populasi diatas 100, nilai *fitness* akan lebih optimal. Namun Penulis mengambil titik tertinggi pada uji coba tersebut yakni sebesar 70 sebagai ukuran populasi terbaik dengan memperhatikan waktu komputasi karena nantinya metode ini akan digabung dengan metode lainnya dan ukuran populasi tentu akan sangat berpengaruh terhadap lamanya suatu metode untuk dieksekusi. Untuk kombinasi *cr* dan *mr* pada saat mencapai solusi optimal tersebut adalah *cr* = 0,7 dan *mr* = 0,3.

5.2.2 Hasil Uji Coba Jumlah Generasi

Hasil uji coba jumlah generasi pada Algoritme Genetika dapat dilihat pada Tabel 5.5. Uji coba jumlah generasi dilakukan untuk mengetahui generasi terbaik pada saat mencapai titik optimal atau terjadinya konvergensi dini. Konvergensi dini terjadi pada saat generasi setelahnya tidak terdapat kenaikan nilai yang signifikan. Untuk melakukan uji coba jumlah generasi, penulis menggunakan beberapa parameter yang telah ditentukan di awal yakni $cr = 0,5$ dan $mr = 0,5$. Sedangkan untuk ukuran populasinya menggunakan hasil terbaik yang didapat pada uji coba sebelumnya yakni ukuran populasi sebesar 70.

Tabel 5.5 Hasil Uji Coba Jumlah Generasi

Jumlah Generasi	Nilai Fitness										
	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
200	0,3064	0,3073	0,3019	0,2998	0,3017	0,3027	0,2999	0,2874	0,3030	0,2996	0,3010
400	0,3117	0,3078	0,3000	0,3099	0,2977	0,3101	0,3067	0,3012	0,3006	0,3018	0,3047
600	0,2893	0,3190	0,3210	0,2888	0,3209	0,3195	0,3330	0,3279	0,3313	0,3300	0,3181
800	0,3183	0,3081	0,3251	0,3229	0,3068	0,3001	0,3174	0,2890	0,2907	0,3151	0,3104
1000	0,3107	0,3185	0,3046	0,3081	0,3096	0,3095	0,3179	0,3080	0,2994	0,3142	0,3101
1200	0,3324	0,3335	0,3276	0,3096	0,3025	0,3229	0,3085	0,3106	0,3077	0,3237	0,3179
1400	0,3297	0,3992	0,3088	0,3128	0,3266	0,3147	0,3179	0,3135	0,3042	0,3288	0,3156
1600	0,3208	0,3250	0,3306	0,3028	0,3156	0,3227	0,3075	0,3277	0,3982	0,3386	0,3190
1800	0,3049	0,3299	0,3091	0,3071	0,3092	0,3230	0,3166	0,3272	0,3184	0,3235	0,3169
2000	0,3200	0,3111	0,3210	0,3239	0,3182	0,3053	0,3291	0,3075	0,3037	0,3119	0,3152



Gambar 5.5 Grafik Hasil Uji Coba Jumlah Generasi

Berdasarkan grafik pada Gambar 5.5, titik tertinggi yang dicapai terdapat pada generasi 1600. Pada generasi selanjutnya terjadi penurunan cukup signifikan. Untuk kombinasi cr dan mr pada saat mencapai solusi optimal tersebut adalah $cr = 0,7$ dan $mr = 0,3$.

5.3 Hasil Uji Coba Metode *Simulated Annealing*

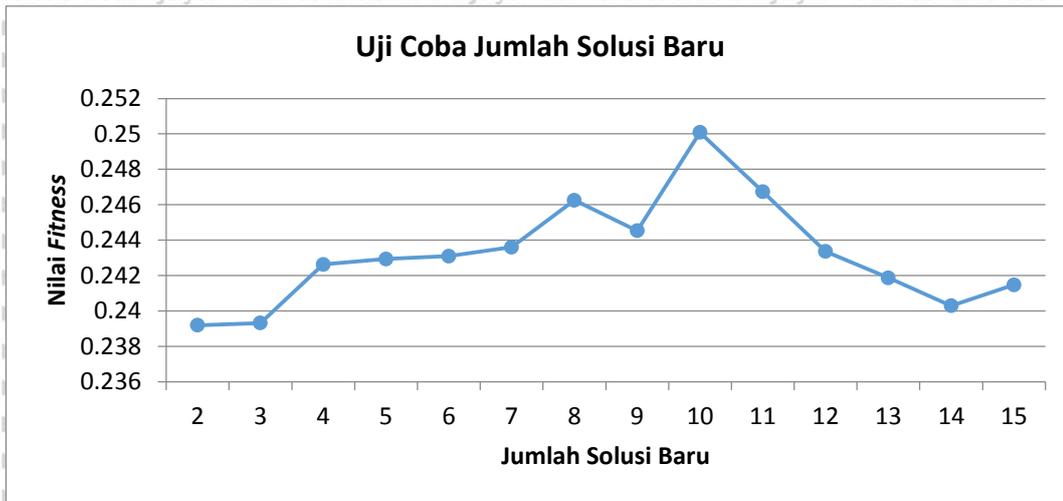
Hasil uji coba metode perbandingan lainnya adalah *Simulated Annealing* yang akan dibahas pada sub-bab ini.

5.3.1 Hasil Uji Coba Jumlah Solusi Baru

Hasil uji coba jumlah solusi baru ditunjukkan pada Tabel 5.6. Penulis melakukan uji coba membangkitkan solusi baru pada tiap iterasi sebanyak 2 hingga 15 solusi. Untuk melakukan uji coba jumlah solusi baru, penulis menggunakan beberapa parameter awal yakni temperatur = 100, faktor reduksi = 0,5 dan iterasi maksimal sebanyak 100.

Tabel 5.6 Hasil Uji Coba Jumlah Solusi Baru

Jumlah Solusi Baru	Nilai Fitness										
	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
2	0,2389	0,2353	0,2380	0,2355	0,2389	0,2456	0,2371	0,2340	0,2484	0,2397	0,2391
3	0,2394	0,2397	0,2376	0,2397	0,2394	0,2451	0,2375	0,2387	0,2348	0,2410	0,2393
4	0,2431	0,2398	0,2403	0,2383	0,2458	0,2475	0,2441	0,2393	0,2466	0,2410	0,2426
5	0,2438	0,2403	0,2440	0,2404	0,2395	0,2463	0,2402	0,2414	0,2424	0,2505	0,2429
6	0,2463	0,2407	0,2430	0,2381	0,2422	0,2441	0,2484	0,2410	0,2419	0,2449	0,2430
7	0,2445	0,2430	0,2417	0,2378	0,2423	0,2416	0,2465	0,2432	0,2421	0,2529	0,2436
8	0,2452	0,2491	0,2497	0,2489	0,2456	0,2413	0,2537	0,2455	0,2419	0,2410	0,2462
9	0,2443	0,2436	0,2477	0,2425	0,2451	0,2468	0,2437	0,2450	0,2419	0,2442	0,2445
10	0,2535	0,2483	0,2432	0,2528	0,2470	0,2510	0,2445	0,2472	0,2652	0,2476	0,2500
11	0,2440	0,2494	0,2509	0,2425	0,2487	0,2499	0,2436	0,2446	0,2437	0,2497	0,2467
12	0,2488	0,2350	0,2482	0,2487	0,2589	0,2334	0,2347	0,2434	0,2353	0,2467	0,2433
13	0,2350	0,2453	0,2489	0,2317	0,2312	0,2446	0,2354	0,2592	0,2463	0,2405	0,2418
14	0,2307	0,2301	0,2480	0,2446	0,2406	0,2488	0,2430	0,2333	0,2423	0,2411	0,2402
15	0,2459	0,2329	0,2450	0,2595	0,2313	0,2375	0,2403	0,2345	0,2459	0,2415	0,2414



Gambar 5.6 Grafik Hasil Uji Coba Jumlah Solusi Baru

Jika melihat grafik pada Gambar 5.6, titik tertinggi yang dicapai yakni pada saat jumlah solusi baru dibangkitkan sebanyak 10. Jumlah tersebut dapat menghasilkan nilai *fitness* yang paling tinggi dikarenakan ruang solusi yang tersedia cukup luas sehingga seluruh kemungkinan solusi optimal akan tercapai. Namun semakin banyaknya jumlah calon solusi baru, tidak menjamin bahwa akan didapatkan solusi yang lebih optimal. Hal tersebut terbukti pada titik setelahnya terjadi penurunan yang cukup signifikan. Kemudian setelah didapatkan jumlah optimal dalam membangkitkan solusi yang baru, penulis akan menggunakan jumlah tersebut sebagai acuan untuk uji coba parameter selanjutnya.

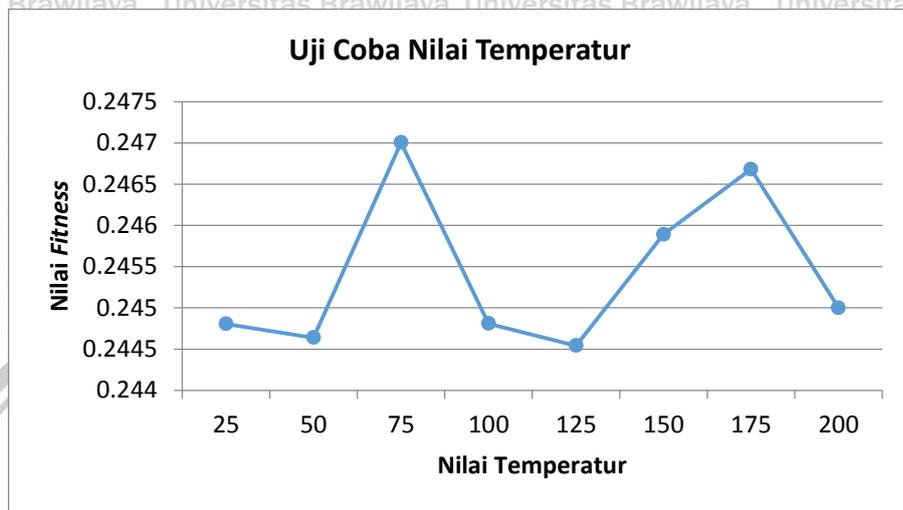
5.3.2 Hasil Uji Coba Nilai Temperatur

Hasil uji coba nilai temperatur ditunjukkan pada Tabel 5.7. Penulis melakukan uji coba nilai temperatur menggunakan nilai dengan kelipatan 25 mulai dari 25 hingga 200. Untuk melakukan uji coba nilai temperatur, penulis menggunakan beberapa parameter awal yakni faktor reduksi = 0,5 dan iterasi maksimal sebanyak 100. Sedangkan untuk jumlah solusi baru yang dibangkitkan adalah sebanyak 10 berdasarkan hasil uji coba parameter terbaik sebelumnya.

Tabel 5.7 Hasil Uji Coba Nilai Temperatur

Temperatur	Nilai Fitness										
	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
25	0,2391	0,2431	0,2414	0,2505	0,2472	0,2480	0,2437	0,2497	0,2420	0,2430	0,2448

50	0,2471	0,2444	0,2405	0,2443	0,2440	0,2423	0,2525	0,2429	0,2432	0,2448	0,2446
75	0,2461	0,2505	0,2425	0,2432	0,2494	0,2453	0,2471	0,2446	0,2555	0,2453	0,2470
100	0,2459	0,2401	0,2461	0,2435	0,2455	0,2446	0,2466	0,2494	0,2438	0,2420	0,2448
125	0,2417	0,2435	0,2461	0,2421	0,2421	0,2457	0,2453	0,2493	0,2483	0,2407	0,2445
150	0,2427	0,2443	0,2472	0,2457	0,2442	0,2423	0,2401	0,2491	0,2524	0,2503	0,2458
175	0,2424	0,2497	0,2470	0,2505	0,2506	0,2474	0,2504	0,2398	0,2464	0,2422	0,2466
200	0,2449	0,2422	0,2472	0,2432	0,2508	0,2454	0,2445	0,2425	0,2468	0,2421	0,2450



Gambar 5.7 Grafik Hasil Uji Coba Nilai temperatur

Jika dilihat pada Gambar 5.7, titik tertinggi yang dicapai adalah pada saat temperatur = 75. Nilai temperatur akan mempengaruhi pertimbangan dalam memilih solusi yang baru. Nilai temperatur ini akan digunakan untuk uji coba parameter selanjutnya.

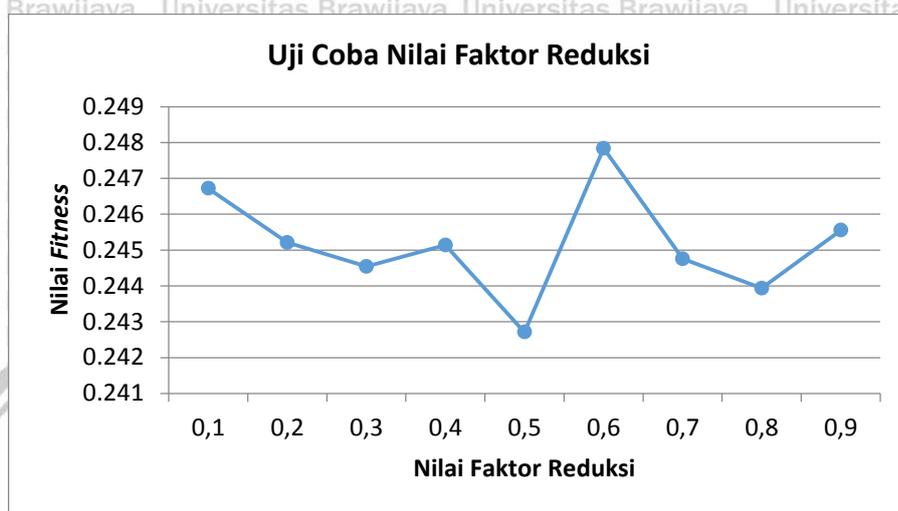
5.3.3 Hasil Uji Coba Nilai Faktor Reduksi

Hasil uji coba nilai faktor reduksi ditunjukkan pada Tabel 5.8. Penulis melakukan uji coba nilai faktor reduksi menggunakan nilai mulai dari 0,1 hingga 0,9. Untuk melakukan uji coba nilai temperatur, penulis menggunakan beberapa parameter awal yakni iterasi maksimal sebanyak 100. Sedangkan untuk jumlah solusi baru yang dibangkitkan adalah sebanyak 10 dan juga nilai temperatur yang digunakan sebesar 75 berdasarkan hasil uji coba terbaik sebelumnya.

Tabel 5.8 Hasil Uji Coba Nilai Faktor Reduksi

Faktor Reduksi	Nilai Fitness										
	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
0,1	0,2534	0,2443	0,2450	0,2451	0,2463	0,2479	0,2464	0,2440	0,2460	0,2484	0,2467
0,2	0,2442	0,2355	0,2414	0,2429	0,2474	0,2423	0,2416	0,2479	0,2451	0,2533	0,2452

0,3	0,2429	0,2499	0,2422	0,2469	0,2538	0,2455	0,2436	0,2413	0,2436	0,2452	0,2445
0,4	0,2414	0,2414	0,2455	0,2423	0,2455	0,2423	0,2493	0,2485	0,2459	0,2489	0,2451
0,5	0,2436	0,2445	0,2427	0,2425	0,2486	0,2451	0,2424	0,2409	0,2393	0,2429	0,2427
0,6	0,2495	0,2469	0,2444	0,2487	0,2483	0,2497	0,2493	0,2492	0,2472	0,2447	0,2478
0,7	0,2432	0,2468	0,2420	0,2456	0,2489	0,2437	0,2413	0,2504	0,2447	0,2404	0,2447
0,8	0,2493	0,2421	0,2425	0,2459	0,2409	0,2437	0,2432	0,2435	0,2436	0,2444	0,2439
0,9	0,2468	0,2451	0,2418	0,2483	0,2463	0,2474	0,2381	0,2459	0,2483	0,2471	0,2455



Gambar 5.8 Grafik Hasil Uji Coba Nilai Faktor Reduksi

Berdasarkan grafik yang dihasilkan pada Gambar 5.8, nilai faktor reduksi terbaik sebesar 0,6. Nilai faktor reduksi akan mempengaruhi pertimbangan dalam memilih solusi yang baru. Nilai faktor reduksi ini akan digunakan untuk uji coba parameter selanjutnya.

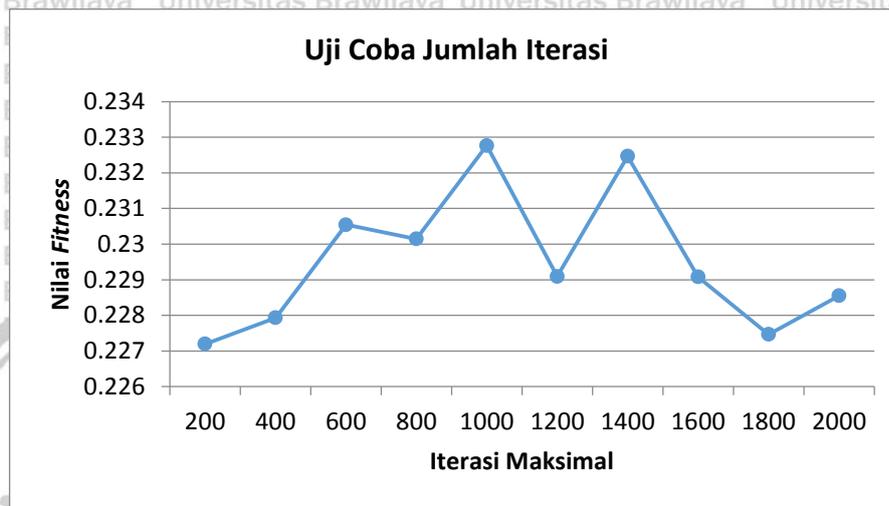
5.3.4 Hasil Uji Coba Jumlah Iterasi Maksimal

Hasil uji coba jumlah iterasi maksimal ditunjukkan pada Tabel 5.9. Penulis melakukan uji coba jumlah iterasi maksimal menggunakan nilai mulai dari 200 hingga 2000. Untuk melakukan uji coba jumlah iterasi maksimal, penulis menggunakan beberapa parameter awal yang telah didapatkan dari hasil uji coba terbaik sebelumnya yakni jumlah solusi baru yang dibangkitkan sebanyak 10, nilai temperatur = 75 dan faktor reduksi = 0,6.

Tabel 5.9 Hasil Uji Coba Jumlah Iterasi Maksimal

Jumlah Iterasi Maksimal	Nilai Fitness										Rata-rata
	Uji Coba ke-										
	1	2	3	4	5	6	7	8	9	10	
200	0,2239	0,2210	0,2251	0,2335	0,2228	0,2250	0,2225	0,2308	0,2271	0,2298	0,2272
400	0,2276	0,2260	0,2365	0,2221	0,2256	0,2246	0,2305	0,2252	0,2317	0,2292	0,2279

600	0,2286	0,2259	0,2259	0,2354	0,2302	0,2278	0,2245	0,2323	0,2379	0,2273	0,2305
800	0,2270	0,2311	0,2208	0,2261	0,2299	0,2326	0,2284	0,2299	0,2289	0,2365	0,2327
1000	0,2395	0,2275	0,2284	0,2328	0,2338	0,2312	0,2226	0,2344	0,2263	0,2308	0,2290
1200	0,2251	0,2318	0,2262	0,2357	0,2314	0,2302	0,2249	0,2280	0,2259	0,2312	0,2290
1400	0,2309	0,2307	0,2248	0,2393	0,2394	0,2358	0,2214	0,2261	0,2303	0,2453	0,2324
1600	0,2338	0,2236	0,2246	0,2256	0,2326	0,2310	0,2150	0,2290	0,2321	0,2429	0,2290
1800	0,2290	0,2299	0,2242	0,2266	0,2330	0,2296	0,2258	0,2226	0,2242	0,2293	0,2274
2000	0,2263	0,2332	0,2357	0,2306	0,2295	0,2194	0,2241	0,2285	0,2290	0,2288	0,2285



Gambar 5.9 Grafik Hasil Uji Coba Jumlah Iterasi Maksimal

Berdasarkan Gambar 5.9 didapatkan titik tertinggi jumlah iterasi maksimal terbaik adalah sebanyak 1000 iterasi. Jumlah iterasi maksimal ini sangat menentukan solusi akhir yang dihasilkan oleh *Simulated Annealing* karena pada setiap iterasi terdapat kemungkinan solusi baru terpilih dan semakin banyak iterasi maka semakin besar pula kemungkinan adanya pergantian solusi akhir.

5.4 Hasil Uji Coba Metode HSAAGA

Hasil uji coba metode usulan yakni Hibridisasi *Simulated Annealing* – Algoritme Genetika Adaptif (HSAAGA) seperti ditunjukkan pada Tabel 5.10. Pada uji coba sebelumnya didapatkan semua parameter terbaik dari masing-masing metodenya. Semua parameter terbaik tersebut akan digunakan penulis untuk melakukan uji coba metode HSAAGA. Adapun parameter terbaik pada Algoritme Genetika Adaptif yakni ukuran populasi = 70, jumlah generasi = 1600, kombinasi $cr = 0,7$ dan $mr = 0,3$. Sedangkan untuk parameter terbaik pada *Simulated Annealing* yakni jumlah solusi baru yang dibangkitkan sebanyak 10, temperatur = 75, faktor reduksi = 0,6 dan iterasi maksimal = 1000.

Tabel 5.10 Hasil Uji Coba Metode HSAAGA

	Uji Coba ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
Nilai <i>Fitness</i>	0,4713	0,4490	0,4439	0,4320	0,4414	0,4483	0,4468	0,4486	0,4400	0,4477	0,4480
Waktu Eksekusi (ms)	19701866	19965587	19712745	19820371	19854968	20131036	19846575	19827304	19826300	19795550	19848230

Dari 10 kali percobaan, didapatkan rata-rata nilai *fitness* sebesar 0,4480.

Solusi terbaik dihasilkan dengan nilai *fitness* tertinggi yakni sebesar 0,4713 dengan waktu eksekusi selama 19701866 ms dan nilai produksinya adalah sebagai berikut [2516, 1068, 1068, 1068, 1068, 1068, 1068, 1068, 1068, 2457, 4741, 2516, 4741, 831, 531, 531, 831, 831, 1603, 831, 831, 1603, 831, 2541, 3607, 615, 267, 615, 714, 615, 267, 743, 267, 714, 615, 615, 615, 344, 344, 255, 255, 344, 255, 344, 1405, 344, 1405, 1405, 1405, 367, 107, 173, 173, 107, 107, 173, 173, 107, 107, 159, 197, 159, 159, 197, 159, 159, 159, 159, 197, 159, 467, 307, 307, 307, 307, 307, 307, 307, 307, 307, 315, 467, 454, 466, 224, 224, 224, 224, 224, 224, 224, 466, 231, 231, 111, 111, 111, 111, 111, 111, 111, 111, 111, 231, 231, 231, 236, 112, 112, 207, 112, 112, 112, 112, 112, 112, 112, 112, 426, 170, 170, 170, 150, 170, 170, 170, 170, 170, 170, 150, 283, 111, 111, 111, 111, 111, 111, 111, 111, 527, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 480, 381, 134, 134, 134, 134, 134, 134, 134, 134, 381, 134, 539, 539, 119, 119, 119, 119, 119, 119, 119, 119, 119, 119, 334, 131, 131, 102, 102, 102, 131, 102, 131, 102, 415, 216, 152, 129, 152, 129, 152, 129, 152, 129, 129, 152, 362, 130, 130, 130, 130, 130, 130, 130, 130, 130, 130].

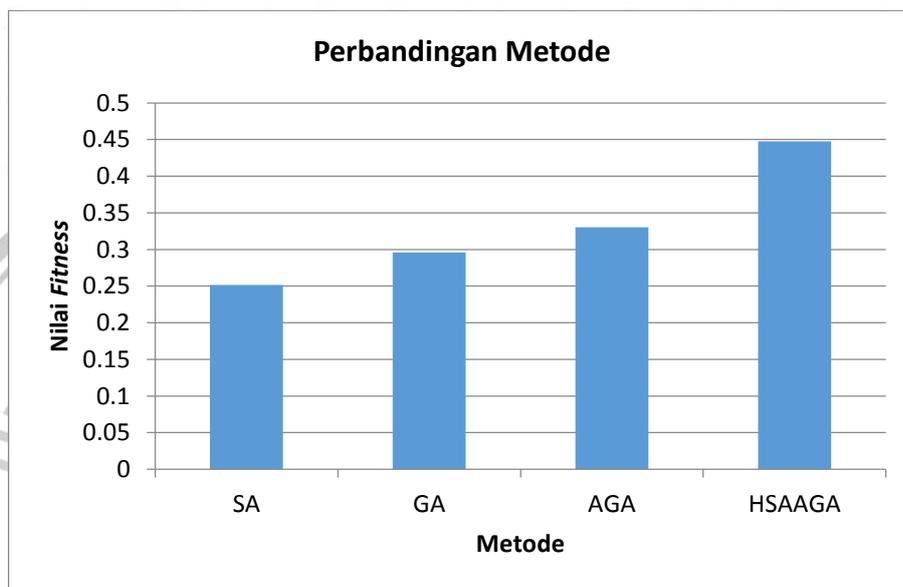
Solusi tersebut diambil oleh penulis untuk dijadikan contoh perbandingan dalam melakukan perencanaan produksi agregat pada sub bab berikutnya.

5.5 Hasil Uji Coba Perbandingan Metode

Untuk mengetahui bahwa metode usulan ini merupakan metode terbaik dalam menyelesaikan permasalahan perencanaan produksi agregat dengan hasil optimal, maka perlu dibuktikan dengan cara membandingkannya dengan metode lain seperti ditunjukkan pada Tabel 5.11. Masing-masing metode akan dijalankan sebanyak 5 kali dengan batasan waktu yang sama yakni selama 120 menit, agar solusi yang dihasilkan lebih adil dalam uji perbandingan metode ini.

Tabel 5.11 Hasil Uji Coba Perbandingan Metode

Metode	Nilai <i>Fitness</i>					
	Uji Coba ke-					Rata-rata
	1	2	3	4	5	
SA	0,248457	0,279402	0,253831	0,231252	0,243663	0,251321
GA	0,299849	0,308543	0,285767	0,310063	0,275091	0,295862
AGA	0,321411	0,328813	0,329429	0,341064	0,331064	0,330356
HSAAGA	0,471322	0,449077	0,443953	0,432027	0,441454	0,447567



Gambar 5.10 Hasil Uji Coba Perbandingan Metode

Berdasarkan hasil uji coba perbandingan metode yang ditunjukkan pada Gambar 5.10, dapat dilihat bahwa metode usulan penulis menghasilkan nilai *fitness* rata-rata paling tinggi. Hibridisasi Algoritme Genetika Adaptif yang diperbaiki menggunakan *Simulated Annealing* cukup efektif terbukti dengan adanya peningkatan nilai *fitness*. Adanya parameter adaptif yakni *cr* dan *mr* pada Algoritme Genetika sangat berperan penting dalam meningkatkan hasil. Selain itu, adanya modifikasi pada saat proses reproduksi juga sangat berperan penting dalam memperluas ruang pencarian solusi sehingga besar kemungkinan untuk mencapai solusi optimal.

5.6 Perbandingan Hasil Penerapan Metode dengan Data Asli

Untuk mengetahui apakah dengan menerapkan metode usulan ini akan dapat berdampak pada perusahaan, maka diperlukan perbandingan secara jelas. Pada penelitian ini, penulis akan membandingkan dengan menggunakan

perhitungan tabel agregat. Penulis memberikan contoh salah satu produk yakni atasan putih panjang. Produk tersebut akan dibuat tabel agregatnya berdasarkan data produksi aktual seperti ditunjukkan pada Tabel 5.12.

Tabel 5.12 Tabel Agregat Aktual

Periode	Permin-taan	Hari Kerja	Produksi	rt	ot	sc	kp	rp	mp	i
Agt-16	1843	21	1764	263						184
Sep-16	866	20	953							271
Okt-16	1202	21	1322							391
Nov-16	1327	21	1460							524
Des-16	744	21	818							598
Jan-17	1304	20	1434							728
Feb-17	1393	22	1532							867
Mar-17	1650	20	1680	135						1032
Apr-17	1606	20	1680	87						1193
Mei-17	2367	17	1428	510	666					1430
Jun-17	4374	21	1764	630	882	1535				1867
Jul-17	6339	22	1848	660	924	3541				2501

Berdasarkan Tabel 5.12, perencanaan produksi agregat aktual untuk produk pertama yang telah dilakukan oleh perusahaan memerlukan biaya produksi sebesar Rp. 645.703.000 ,-. Hasil yang berbeda didapatkan dengan menerapkan metode HSAAGA untuk melakukan perencanaan produksi agregat untuk produk pertama seperti ditunjukkan Tabel 5.13.

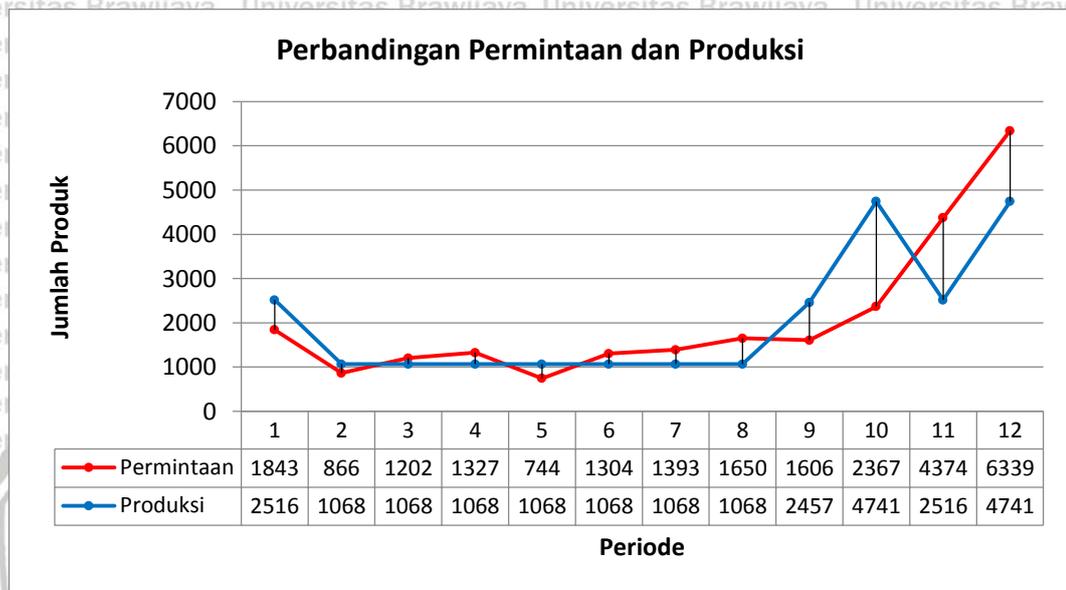
Tabel 5.13 Tabel Agregat HSAAGA

Periode	Permin-taan	Hari Kerja	Produksi	rt	ot	sc	kp	rp	mp	i
Agt-16	1843	21	2516	1764	630	122				673
Sep-16	866	20	1068	1068						875
Okt-16	1202	21	1068	1068						741
Nov-16	1327	21	1068	1068						482
Des-16	744	21	1068	1068						806
Jan-17	1304	20	1068	1068						570
Feb-17	1393	22	1068	1068						245
Mar-17	1650	20	1068	1068						-337
Apr-17	1606	20	2457	1680	600	177				851
Mei-17	2367	17	4741	1428	510	714	2089	10		3225
Jun-17	4374	21	2516	2516						1367
Jul-17	6339	22	4741	4741						-231

Total biaya produksi yang akan dikeluarkan oleh perusahaan apabila menerapkan metode HSAAGA yakni sebesar Rp. 612.731.500 ,-. Terdapat selisih biaya produksi sebesar Rp. 32.971.500 ,-, pada produksi produk pertama yang cukup berpengaruh bagi perusahaan karena produk yang diproduksi oleh

perusahaan cukup banyak yakni 18 produk. Dengan menerapkan metode HSAAGA ini, perusahaan dapat melakukan perencanaan produksi agregat secara efektif dan efisien serta mendapatkan hasil yang optimal.

Setelah didapatkan solusi akhir berupa jumlah produksi dari penerapan metode tersebut kemudian penulis membandingkan dengan permintaan konsumen untuk dapat dilihat selisihnya seperti ditunjukkan pada Gambar 5.11.



Gambar 5.11 Perbandingan Permintaan dan Produksi

Jika dilihat pada Gambar 5.11, nilai produksi yang dihasilkan dari penerapan metode HSAAGA terlihat cukup stabil pada bulan kedua hingga bulan kedelapan. Setelah itu, terjadi lonjakan cukup tinggi dan juga terdapat titik dimana nilai produksi jauh dibawah permintaan. Namun, hal tersebut tidak berarti bahwa produksi perusahaan tidak dapat memenuhi permintaan konsumen. Persediaan perusahaan sudah cukup banyak untuk memenuhi permintaan konsumen karena pada bulan sebelumnya nilai produksi jauh melebihi permintaan, sehingga persediaan bertambah dan perusahaan dapat memenuhi permintaan konsumen.