

PERBANDINGAN *PARTICLE SWARM OPTIMIZATION* DAN *GENETIC ALGORITHM* UNTUK SISTEM KONTROL *VENDOR MANAGED INVENTORY* MULTI-PRODUK MULTI-KENDALA

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Pada Program Studi Teknik Industri Fakultas Teknologi Industri**



Nama : Bayu Pebrian Prakoso

NIM : 10522020

**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2014

PERBANDINGAN *PARTICLE SWARM OPTIMIZATION* DAN *GENETIC ALGORITHM* UNTUK SISTEM KONTROL *VENDOR MANAGED INVENTORY* MULTI-PRODUK MULTI-KENDALA

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Pada Program Studi Teknik Industri Fakultas Teknologi Industri**



Nama : Bayu Pebrian Prakoso

NIM : 10522020

**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2014

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Demi Allah, saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 24 November 2014



Bayu Pebrian Prakoso

10522020

LEMBAR PENGESAHAN DOSEN PEMBIMBING

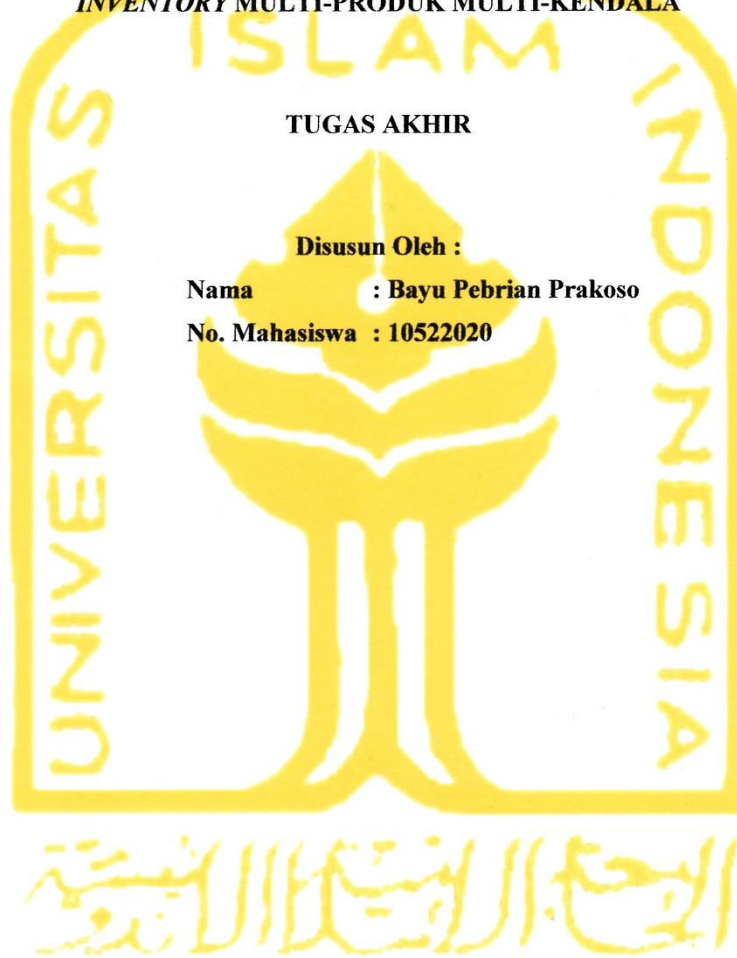
PERBANDINGAN *PARTICLE SWARM OPTIMIZATION* DAN *GENETIC ALGORITHM* UNTUK SISTEM KONTROL *VENDOR MANAGED INVENTORY* MULTI-PRODUK MULTI-KENDALA

TUGAS AKHIR

Disusun Oleh :

Nama : Bayu Pebrian Prakoso

No. Mahasiswa : 10522020



Yogyakarta, 24 November 2014

Dosen Pembimbing

(Prof. Dr. Ir. R. Chairul Saleh, M.Sc)

LEMBAR PENGESAHAN DOSEN PENGUJI

PERBANDINGAN *PARTICLE SWARM OPTIMIZATION* DAN *GENETIC ALGORITHM* UNTUK SISTEM KONTROL *VENDOR MANAGED INVENTORY* MULTI-PRODUK MULTI-KENDALA

TUGAS AKHIR

Oleh :

Nama : Bayu Pebrian Prakoso
No. Mahasiswa : 10522020

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk
Memperoleh Gelar Sarjana Teknik Industri Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, 24 November 2014

Tim Penguji

Prof. Dr. Ir. R. Chairul Saleh, M.Sc.

Ketua Penguji

Dwi Ana Ratna Wati, ST., M.Eng

Anggota I

Amarria Dila Sari, ST., M.Eng

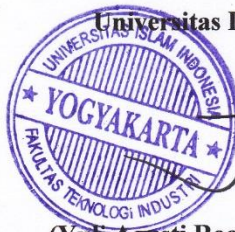
Anggota II

Mengetahui,

Ketua Program Studi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Yuli Agusti Rochman, ST., M.Eng)

HALAMAN PERSEMBAHAN

Alhamdulillahil'alamin...

Kupersembahkan hasil karyaku ini untuk...

Kedua orang tuaku (Alm. R. Trasmono Bambang Sutopo dan Asiah)

Kakak-kakakku (Ratna Iriawati, Dwi Irnanto, dan Catur Prastyo)

Yang tiada henti memberikan doa, dukungan, bimbingan serta motivasi

Untukku agar selalu melangkah maju....

Semua sahabatku yang tidak bisa aku sebutkan satu persatu

Terimakasih atas doa dan dukungannya....

Semua guru-guruku yang telah memberikan ilmu yang sangat berharga

Dan seluruh motivator mengagumkan yang telah dikirim Allah untukku

Amalmu akan menjadi pahala jariyahmu kelak....

HALAMAN MOTTO

بَلِّغُوا عَنِّي وَلَوْ آيَةً

“Sampaikanlah dariku walau hanya satu ayat” (HR. Bukhari)

إِنَّ اللَّهَ يُدْخِلُ الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ جَنَّاتٍ تَجْرِي مِنْ تَحْتِهَا الْأَنْهَارُ ۚ إِنَّ اللَّهَ يَفْعَلُ مَا يُرِيدُ

“Sesungguhnya Allah memasukkan orang-orang yg beriman & mengerjakan amal yg saleh ke dalam surga-surga yg di bawahnya mengalir sungai-sungai. Sesungguhnya Allah berbuat apa yg Dia kehendaki” (QS. Al Hajj : 14)

فَاذْكُرُونِي أَذْكُرْكُمْ وَاشْكُرُوا لِي وَلَا تَكْفُرُونِ

“Ingatlah kamu sekalian kepada-Ku niscaya Aku ingat kepadamu, serta Bersyukurlah kepada-Ku, dan janganlah kamu mengingkari nikmat-Ku.” (QS. Al-Baqarah : 152)

يَرْفَعُ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۚ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

“Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.” (Q.s. Al-Mujadalah : 11)

KATA PENGANTAR

Assalamu'alaikum Warahmatullah Wabarakatuh

Alhamdulillah, segala puji syukur kepada Allah SWT yang Maha Pengasih dan Penyayang, sholawat serta salam selalu dihaturkan pada Nabi junjungan kita Muhammad SAW sebaik-baik ciptaan-Nya yang telah membawa kita ke jalan yang diridhai-Nya.

Dengan Rahmat dan Hidayah Allah SWT akhirnya tugas akhir yang berjudul “Perbandingan *Particle Swarm Optimization* dan *Genetic Algorithm* Untuk Sistem Kontrol *Vendor Managed Inventory* Multi-Produk Multi-Kendala” dapat terselesaikan dengan baik.

Tugas Akhir merupakan salah satu syarat untuk memperoleh gelar sarjana strata-1 pada Program Studi Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia. Kemudahan dan kelancaran dalam penyusunan laporan tugas akhir ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Ketua Prodi Teknik Industri dan seluruh staf Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu tercinta dan Almarhum Ayah atas segala do'a, dukungan dan motivasi yang tiada henti-hentinya mengalir untukku.
4. Bapak Prof. Dr. Ir. R. Chairul Saleh M.Sc. Selaku Dosen Pembimbing yang telah memberikan bimbingan, bantuan dan arahnya dalam penyusunan Tugas Akhir ini.
5. Ibu Dwi Ana Ratna Wati, ST., M.Eng yang telah memberikan ilmu dan bantuannya dalam penyusunan Tugas Akhir ini.
6. Kakak-kakakku tersayang yang telah memberikan fasilitas, dukungan dan motivasi.
7. Keluarga Bapak Chairul Saleh yang telah memberikan dukungan secara moril maupun materil.
8. Teman-teman serta semua pihak yang tidak dapat disebutkan satu persatu.

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat khususnya di dunia ilmu pengetahuan bagi semua pihak. Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih banyak terdapat kekurangan sehingga dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun demi penyempurnaan pada masa mendatang.

Wassalamu'alaikum Warahmatullah Wabarakatuh

Yogyakarta, 27 Oktober 2014

Bayu Pebrian Prakoso

DAFTAR ISI

COVER	i
LEMBAR PERNYATAAN	ii
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan Laporan	6
BAB II KAJIAN LITERATUR	8
2.1 Kajian Induktif	8
2.2 Kajian Deduktif	15
2.2.1 <i>Supply Chain Management</i>	15
2.2.2 Sistem Inventori	18
2.2.3 Fungsi Inventori	19
2.2.4 Biaya-biaya Inventori	20
2.2.5 Perkembangan <i>Vendor Managed Inventory</i>	21
2.2.6 <i>Genetic Algorithm</i>	23
2.2.7 <i>Particle Swarm Optimization</i>	28

BAB III METODOLOGI PENELITIAN	34
3.1 Fokus Kajian	34
3.2 Model Kajian	34
3.3 Jenis dan Sumber Data	36
3.4 Optimasi Model dengan <i>Particle Swarm Optimization</i>	37
3.5 Alat Penelitian	40
3.6 Kerangka Penelitian	41
3.7 Penjelasan Kerangka Penelitian	42
 BAB IV PENGUMPULAN DAN PENGOLAHAN DATA	 44
4.1 Pemakaian Data	44
4.2 Data yang Digunakan	44
4.2.1 Data Inventori <i>Supplier</i>	45
4.2.2 Data Inventori <i>Retailer</i>	46
4.2.3 Data Hasil Optimasi GA	47
4.3 Verifikasi Data Hasil Optimasi GA	47
4.3.1 Verifikasi Kendala	48
4.3.2 Verifikasi Total Biaya	49
4.4 Optimasi menggunakan <i>Particle Swarm Optimization</i>	52
4.4.1 Optimasi pada $n = 3$	52
4.4.2 Optimasi pada $n = 5$	56
4.4.3 Optimasi pada $n = 8$	57
4.4.4 Optimasi pada $n = 10$	59
4.5 Hasil Optimasi PSO	60
4.6 Verifikasi Hasil Optimasi PSO	61
4.6.1 Verifikasi Kendala	61
4.6.2 Verifikasi Total Biaya	62
4.7 Perbandingan Hasil Optimasi GA dan PSO	65
 BAB V PEMBAHASAN	 67
5.1 Optimasi Total Biaya Inventori	67
5.2 Analisis Perbandingan GA dan PSO	69

BAB VI PENUTUP	71
6.1 Kesimpulan	71
6.2 Saran	71

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 2.1 <i>State Of The Art</i>	9
Tabel 3.1 Data c_1 dan c_2 untuk Percobaan	38
Tabel 3.2 Alat Penelitian	40
Tabel 4.1 Data Produk	45
Tabel 4.2 Data Inventori <i>Supplier</i>	45
Tabel 4.3 Data A_{jS} dan f_j	45
Tabel 4.4 Data D_j , A_{jR} dan h_{jR}	46
Tabel 4.5 Data Hasil Optimasi GA	47
Tabel 4.6 Perhitungan Verifikasi Kendala untuk Hasil Optimasi GA	48
Tabel 4.7 Hasil Percobaan Optimasi PSO ($n = 3$)	52
Tabel 4.8 Hasil Percobaan Optimasi PSO ($n = 5$)	56
Tabel 4.9 Hasil Percobaan Optimasi PSO ($n = 8$)	57
Tabel 4.10 Hasil Percobaan Optimasi PSO ($n = 10$)	59
Tabel 4.11 Hasil Optimasi PSO	61
Tabel 4.12 Perhitungan Verifikasi Kendala untuk Hasil Optimasi PSO	62
Tabel 4.13 Perbandingan Hasil Optimasi GA dan PSO	66

DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Algoritma PSO	37
Gambar 3.2 Diagram Alir Penelitian	41
Gambar 4.1 Grafik Konvergen untuk $n = 3$	55
Gambar 4.2 Grafik Konvergen untuk $n = 5$	57
Gambar 4.3 Grafik Konvergen untuk $n = 8$	58
Gambar 4.4 Grafik Konvergen untuk $n = 10$	60
Gambar 4.5 Grafik Perbandingan Hasil Optimasi GA dan PSO	66

ABSTRAK

Perkembangan kajian tentang *Vendor Managed Inventory* (VMI) hingga saat ini menjadi kecendrungan untuk diteliti oleh para peneliti di seluruh dunia. Berbagai model telah dibangun dan diteliti. Disamping itu masalah yang dihadapi adalah berbagai jenis produk dan kendala serta jumlah *supplier* dan *retailer*. Hampir semuanya telah dilakukan termasuk model penyelesaian menggunakan teknik *Artificial Intelligence* (AI) seperti *Genetic Algorithm* (GA), *Simulated Annealing* (SA), *Tabu Search* (TB), *Ant Colony Optimization* (ACO) dan *Particle Swarm Optimization* (PSO). Metode GA sudah sering digunakan dalam menyelesaikan masalah optimasi. Metode PSO dibandingkan dengan metode lain memiliki kelebihan utama yaitu mempunyai konsep sederhana, mudah diimplementasikan, dan efisien dalam perhitungan. Tujuan dari penelitian ini adalah mengoptimalkan total biaya inventori untuk sistem kontrol *Vendor Managed Inventory* multi-produk multi-kendala pada *single supplier single retailer* dan kemudian membandingkan hasil optimasi *Particle Swarm Optimization* dan *Genetic Algorithm*. Pesanan menggunakan model *Economic Order Quantity*. Pada penelitian ini digunakan tiga, lima, delapan dan sepuluh jenis produk dengan kendala kapasitas gudang dan jumlah pesanan untuk semua jenis produk. Hasil yang dicapai penelitian ini adalah bahwa PSO memberikan optimasi yang lebih optimal dibandingkan GA. PSO memberikan hasil untuk 3 jenis produk \$ 384,59 atau meningkatkan efisiensi 58,55%, untuk 5 jenis produk \$ 908,52 atau meningkatkan efisiensi 23,72%, untuk 8 jenis produk \$ 2362,19 atau meningkatkan efisiensi 31,41% dan untuk 10 jenis produk \$ 4042,18 atau meningkatkan efisiensi 39,54%.

Kata Kunci : *Particle Swarm Optimization, Genetic Algorithm, Vendor Managed Inventory*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Isu tentang inventori selalu menarik perhatian para peneliti, hal ini karena inventori dapat merepresentasikan 20 hingga 40 persen dari biaya pokok produksi (Ganeshan & Harrison, 1995). Jika dapat meminimalkan inventori maka perusahaan akan dapat meraih keuntungan yang lebih besar. Permasalahan inventori dapat ditemui diberbagai tempat seperti di rumah tangga, kantor, perusahaan jasa maupun perusahaan manufaktur. Dalam kehidupan sehari-hari yang melibatkan aktivitas mulai dari pasokan barang mentah atau setengah jadi kedalam perusahaan untuk diolah menjadi produk akhir, dan kemudian didistribusikan kepada retailer-retailer, dan destinasi terakhir adalah konsumen juga tidak terlepas dari persoalan inventori.

Para pengusaha selalu berusaha untuk meminimumkan inventori sehingga banyak dilakukan penelitian untuk mengatasi kelemahan-kelemahan sistem inventori yang digunakan. Dalam aktivitas bisnis yang melibatkan pemasok, pamanufaktur dan distributor disebut dengan *Supply Chain Management* (SCM) (Vanany et al., 2009). Dalam pengaplikasian SCM banyak dihadapi kesulitan untuk mengatur sistem inventori baik oleh pemasok, pamanufaktur ataupun distributor. Untuk dapat meningkatkan *service level* salah satu faktor utama adalah dengan mengelola inventori secara efisien dari masing-masing pihak baik pemasok, pamanufaktur serta distributor (Kwak et al., 2009). Khususnya pihak manufaktur sebagai penerima pasokan dan pembuat produk

akan menanggung biaya inventori yang tinggi. Sehingga perlu manajemen inventori yang dapat meminimalkan biaya inventori dimana pihak manufaktur juga memiliki *service level* yang memuaskan (Tyan & Wee, 2003).

Dalam aplikasi pengaturan inventori ditemukan permasalahan yang cukup kompleks yaitu *item* barang yang disimpan memiliki jenis berbeda, umur barang, *lead time* pasokan, cara pengemasan, cara penyimpanan, dan cara pengiriman yang berbeda. Selanjutnya pesanan yang diterima pihak manufaktur juga berbagai jenis *item* dan volumenya tidak sama. Sedangkan kapasitas gudang yang dimiliki pihak manufaktur dan jumlah pesanan untuk semua *item* adalah terbatas. Adanya permasalahan-permasalahan yang kompleks ini sehingga muncul sebuah konsep yang terkenal yaitu *Vendor Managed Inventory* (VMI) (Pasandideh et al., 2011).

Pihak manufaktur yang memanfaatkan pemasok untuk menyediakan bahan baku, komponen, maupun produk jadi memiliki satu masalah yang penting yaitu dalam menentukan kuantitas pesanan dan titik-titik untuk melakukan pesanan. Berbagai model dalam sistem inventori telah diusulkan dan ditujukan untuk memecahkan masalah ini dalam skenario yang berbeda (Pasandideh et al., 2011). *Economic Order Quantity* (EOQ) adalah salah satu model optimasi yang paling populer dan sukses dalam manajemen *supply chain*, karena kesederhanaan penggunaan, kesederhanaan konsep, dan ketahanan (Teng, 2009). Namun, model EOQ tersebut dibangun berdasarkan beberapa asumsi dan kondisi yang terikat penerapannya dalam situasi dunia nyata.

Berbagai penelitian telah dilakukan oleh beberapa pakar tentang VMI ini seperti Yao et al. (2007) yang menyatakan bahwa keuntungan penerapan VMI adalah

pengurangan biaya simpan baik pada pemasok maupun *retailer*, peningkatan *customer service level*, seperti dengan pengurangan waktu siklus pemesanan barang dan peningkatan frekuensi penggantian atau pengisian persediaan. Dalam sistem *supply chain single-vendor multi-retailer* model VMI digunakan dengan tujuan untuk mengetahui total biaya inventori terendah dalam kerjasama antara vendor dan *retailer* yang akan berdampak pada keuntungan yang optimal. Namun masih dibatasi asumsi-asumsi yang ada sehingga beberapa variabel keputusan ditambahkan sebagai tolok ukur dalam penentuan biaya inventori (Darwish & Odah, 2010)

Pada penelitian Yu et al. (2012) mengembangkan model VMI dengan mempertimbangkan faktor keusangan yang cepat pada bahan baku dan keusangan yang lambat pada produk jadi. Dari hasil model yang telah dikembangkan hasilnya memperlihatkan bahwa tingkat keusangan produk dapat meningkatkan total biaya hingga 40% dibandingkan dengan produk yang tidak memiliki tingkat keusangan, sedangkan tingkat keusangan pada bahan baku memiliki dampak yang lebih kecil pada total biaya. Selain itu Yu et al. (2013) pada penelitian lainnya merumuskan tentang masalah dalam seleksi *retailer* menggunakan model *Stackelberg Game* untuk membantu pihak manufaktur memilih *retailer* yang tepat dalam membentuk sistem VMI. Dalam pemecahan model digunakan *Hybrid Algorithm* yaitu menggabungkan metode *Dinamic Programming* (DP) dan metode analisis. Kemudian Saleh et al. (2014) melanjutkan penelitian Yu et al. (2013) dengan mengoptimasi kasus *single vendor multi retailer* menggunakan metode *Genetic Algorithm* (GA) agar lebih efektif dan efisien.

Selanjutnya pada penelitian Pasandideh et al. (2011) metode GA digunakan untuk menemukan jumlah pesanan optimum dan tingkat *backorder* yang maksimum

dalam sebuah model VMI yang telah dikembangkannya sehingga biaya total persediaan dapat diminimalkan. Namun demikian penelitian ini masih belum mempertimbangkan metode-metode optimisasi lainnya yang akhir dekade ini sering digunakan yaitu *Simulated Annealing* (SA), *Tabu Search* (TB), *Ant Colony Optimization* (ACO) dan *Particle Swarm Optimization* (PSO). Metode PSO dalam mengoptimasi cenderung terjebak dalam lokal optimum (Purba et al., 2014). Namun metode PSO dibandingkan dengan metode SA, TB, dan ACO memiliki kelebihan utama yaitu mempunyai konsep sederhana, mudah diimplementasikan, dan efisien dalam perhitungan (Tuegeh et al., 2009).

Berdasarkan kajian literatur diatas memunculkan ide untuk membandingkan total biaya persediaan yang dioptimasi menggunakan metode *Genetic Algorithm* pada penelitian Pasandideh et al. (2011) dengan metode *Particle Swarm Optimization*.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan maka disusun rumusan masalah yang menghasilkan pertanyaan penelitian sebagai berikut :

1. Bagaimana total biaya inventori yang optimal jika dioptimasi dengan PSO?
2. Bagaimana perbandingan antara metode GA pada penelitian Pasandideh et al. (2011) dengan metode PSO dalam mengoptimalkan total biaya inventori?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan penelitian maka batasan yang digunakan agar fokus penelitian ini tetap terarah adalah :

1. Data yang digunakan merupakan data-data pada penelitian Pasandideh et al. (2011).
2. Penelitian di fokuskan pada sistem VMI multi-produk multi-kendala pada *single-supplier* dan *single-retailer*.
3. Kajian optimasi menggunakan metode *Particle Swarm Optimization*.
4. Pengolahan data menggunakan bantuan *software* Microsoft Excel 2013 dan Matlab R2013b.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan pertanyaan penelitian maka dapat disusun tujuan penelitian sebagai berikut :

1. Mengoptimalkan total biaya inventori pada model VMI multi-produk multi-kendala pada level *single-supplier* dan *single-retailer* dengan metode PSO.
2. Membandingkan metode GA dengan PSO dalam mengoptimasi total biaya inventori pada model VMI multi-produk multi-kendala.

1.5 Manfaat Penelitian

Berdasarkan kajian literatur yang telah dilakukan maka penelitian ini dapat memberikan manfaat sebagai berikut :

1. Dapat mengisi data base ilmu pengetahuan khususnya pada bidang *Supply Chain Management*, *Vendor Managed Inventory*, dan metode optimasi.
2. Dapat meningkatkan pengembangan khasanah ilmu pengetahuan dalam ruang lingkup kajian teknik industri.

1.6 Sistematika Penulisan Laporan

Penyusunan laporan tugas akhir ini akan disusun berdasarkan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini akan berisi latar belakang permasalahan penelitian, perumusan masalah, pertanyaan penelitian, tujuan penelitian, batasan penelitian, manfaat penelitian dan sistematika penulisan laporan.

BAB II KAJIAN LITERATUR

Pada bab ini akan dipresentasikan hasil kajian literatur yang akan mendukung penelitian yang terdiri dari kajian induktif dan kajian deduktif. Kajian induktif menjelaskan tentang penelitian-penelitian terdahulu yang berkaitan dengan penelitian ini. Sedangkan kajian deduktif menjelaskan tentang landasan teori atau metode yang digunakan pada penelitian ini.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan mengenai metode penelitian yang disusun dari beberapa sub bab seperti fokus kajian, model kajian yang digunakan

dalam penelitian, jenis dan sumber data, algoritma PSO dalam mengoptimasi model VMI, alat-alat penelitian, dan kerangka penelitian.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini menjelaskan tentang pengumpulan dan pengolahan data, analisis serta hasil yang dicapai.

BAB V PEMBAHASAN

Pada bab ini akan didiskusikan hasil penelitian yang dilakukan sesuai dengan tujuan penelitian.

BAB VI PENUTUP

Pada bab ini akan dipaparkan kesimpulan hasil penelitian dan saran yang perlu diberikan untuk penelitian di masa yang akan datang.

DAFTAR PUSTAKA

LAMPIRAN

Tabel

Gambar

BAB II

KAJIAN LITERATUR

Pada bab ini akan dipresentasikan hasil kajian literatur yang akan mendukung penelitian yang terdiri dari kajian induktif dan kajian deduktif. Kajian induktif menjelaskan tentang penelitian-penelitian terdahulu yang berkaitan dengan penelitian ini. Sedangkan kajian deduktif menjelaskan tentang landasan teori atau metode yang digunakan pada penelitian ini.

2.1 Kajian Induktif

Pada tahun 1990-an beberapa perusahaan telah melihat bahwa dengan menerapkan VMI sebagai manajemen persediaan dalam *supply chain*, perusahaan dapat menghasilkan keunggulan yang kompetitif (Pasandideh, Niaki, & Nia, 2011). Disney & Towill (2002) menyebutkan tujuan utama diterapkannya VMI adalah dapat meminimalkan biaya inventori sekaligus memuaskan beberapa indikator *service level* pelanggan yang dapat dicapai jika saling bertukar informasi permintaan dan persediaan. Kemudian Disney & Towill (2003) menyatakan konsep VMI di sektor grosir telah menjadi strategi yang sangat populer dan telah diterapkan juga di sektor yang beragam seperti baja, buku dan petrokimia. Selain itu, VMI menjadi konsep yang efektif dalam melaksanakan koordinasi antar pihak dalam *supply chain* dan juga menjadi konsep yang kritis dan penting dalam meningkatkan kinerja keuangan seluruh pihak dalam *supply chain* (Lee & Chu, 2005). Berbagai penelitian tentang model VMI telah dilakukan oleh para peneliti yang dapat ditunjukkan pada Tabel 2.1 sebagai berikut:

Tabel 2.1 State Of The Art

Peneliti	Judul	Model	Metode	Hasil
Zavanella & Zaroni (2009)	<i>A One-Vendor Multi-Buyer Integrated Production-Inventor Model: The "Consignment Stock" Case</i>	VMI Single-Vendor Multi-Buyer	Analisis Model dan Analisis Sensitivitas	Peningkatan pada manfaat ekonomi dan pengaruh parameter yang relevan untuk kinerja <i>supply chain</i>
Darwish & Odah (2010)	<i>Vendor Managed Inventory Model For Single-Vendor Multi-Retailer Supply Chains</i>	VMI Single-Vendor Multi-Buyer	Pengembangan Model, Analisis Sensitivitas, Percobaan Numerik	Model yang lebih efisien dalam menentukan solusi optimal secara global
Yu, Hong, Zhang, Liang, & Chu (2013)	<i>Optimal Selection Of Retailers For A Manufacturing Vendor In A Vendor Managed Inventory System</i>	VMI Single-Vendor Multi-Retailer	Pengembangan Model, Analisis Sensitivitas, <i>Stackelberg Game</i>	Pemilihan jumlah <i>retailer</i> yang optimal
Nachiappan & Jawahar (2007)	<i>A Genetic Algorithm For Optimal Operating Parameters Of VMI system In A Two-Echelon Supply Chain</i>	VMI Single-Vendor Multi-Buyer	Pengembangan Model dan <i>Genetic Algorithm</i>	Jumlah penjualan yang optimal untuk setiap pembeli
Pasandideh & Niaki (2008)	<i>A Genetic Algorithm Approach To Optimize A Multi-products EPQ Model With Discrete Delivery Orders And Constrained Space</i>	EPQ Multi-Produk	Pengembangan Model dan <i>Genetic Algorithm</i>	Jumlah order yang optimal

Peneliti	Judul	Model	Metode	Hasil
Saleh, Hassan, Deros, Rahman, Leuveano, Adiyoga (2014)	<i>Parameter Optimization Of VMI System In A Manufacturer and Multi Retailer Using Genetic Algorithm</i>	VMI Single-Vendor Multi-Retailer	Pengembangan Model, <i>Stackelberg Game</i> , dan <i>Genetic Algorithm</i>	Peningkatan keuntungan kepada manufaktur sebesar 37,14%
Mousavi, Hajipour, Niaki, & Aalifar (2013)	<i>A Multi-Product Multi-Period Inventory Control Problem Under Inflation And Discount: A Parameter-Tuned Particle Swarm Optimization</i>	EOQ Multi-Produk Multi-Periode	Pengembangan Model, <i>Particle Swarm Optimization</i> dan <i>Genetic Algorithm</i>	Jumlah produksi yang optimal pada periode yang berbeda sehingga didapatkan total biaya inventori yang minimal
Sadeghi, Sadeghi, & Niaki (2014)	<i>Optimizing A Hybrid Vendor Managed Inventory And Transportation Problem With Fuzzy Demand: An Improved Particle Swarm Optimization Algorithm</i>	VMI Single-Vendor Single-Retailer dengan permintaan berupa fuzzy	Pengembangan Model, <i>Particle Swarm Optimization</i> , dan <i>Genetic Algorithm</i>	Total biaya inventori dan biaya transportasi dapat diminimalkan
Pasandideh, Niaki, & Nia (2011)	<i>A Genetic Algorithm For Vendor Managed Inventory Control System Of Multi-Product Multi-Constraint Economic Order Quantity Model</i>	VMI Multi-Produk Multi-Kendala pada Single-Supplier Single-Retailer	<i>Genetic Algorithm</i>	Jumlah kuantitas pesanan dan maksimum <i>backorder</i> yang optimal sehingga total biaya inventori menjadi
Bayu Pebrian Prakoso & Chairul Saleh (2014)	<i>Perbandingan Particle Swarm Optimization dan Genetic Algorithm Untuk Sistem Kontrol Vendor Managed Inventory Multi-Produk Multi-Kendala</i>	VMI Multi-Produk Multi-Kendala Single-Supplier Single-Retailer	<i>Particle Swarm Optimization</i> dan <i>Genetic Algorithm</i>	-

Dari Tabel 2.1 dapat dijelaskan bahwa Zavanella & Zanoni (2009) meneliti sistem VMI yang diterapkan pada level *single-vendor multi-buyer* yaitu *supply chain* pada satu pemasok yang memiliki banyak pembeli atau *retailer*. Pada penelitian ini hasilnya menunjukkan bahwa kerjasama pengelolaan inventori memberikan peningkatan pada manfaat ekonomi. Selain itu penelitian ini menghasilkan pengembangan analisis sensitivitas yang memungkinkan menggambarkan beberapa hal menarik tentang pengaruh parameter yang relevan untuk kinerja ekonomi pada *supply chain* tersebut.

Penelitian terhadap model VMI yang diterapkan pada level *single-vendor multi-buyer* lainnya dilakukan oleh Darwish & Odah (2010) yang bertujuan untuk mengembangkan model untuk *supply chain* dengan vendor tunggal dan beberapa pengecer dengan model VMI. Model yang dikembangkan dapat dengan mudah mendeskripsikan *supply chain* dengan keterbatasan kapasitas dengan memilih biaya penalti yang tertinggi. Kemudian teorema yang didirikan adalah untuk mengurangi kompleksitas model dan membuat matematika yang mudah dikerjakan. Selain itu, algoritma yang efisien bertujuan untuk mencari solusi optimal global. Algoritma ini untuk mengurangi upaya komputasi secara signifikan. Di samping itu, eksperimen numerik dilakukan untuk menunjukkan utilitas dari tujuan model tersebut.

Selain itu Yu et al. (2013) pada penelitiannya merumuskan tentang masalah dalam seleksi *retailer* menggunakan model *Stackelberg Game* untuk membantu pihak manufaktur memilih *retailer* yang tepat dalam membentuk sistem VMI. Dalam pemecahan model digunakan *Hybrid Algorithm* yaitu menggabungkan metode *Dinamic Programming* (DP) dan metode analisis.

Dalam beberapa penelitian tentang model VMI masih terdapat kekurangan karena asumsi-asumsi yang masih melekat sehingga total biaya inventori masih dapat dioptimalkan dengan metode-metode optimasi saat ini. Beberapa penelitian telah dilakukan diantaranya, Nachiappan & Jawahar (2007) merumuskan masalah *Non-linear Integer-Programming* (NIP) dan mengusulkan metode GA berbasis heuristik untuk menemukan jumlah penjualan yang optimal untuk setiap pembeli. Kemudian Pasandideh & Niaki (2008) mengembangkan model EPQ multi-produk, dimana kapasitas gudang terbatas dan pesanan dikirim dalam bentuk beberapa palet. Sehingga mereka merumuskan masalah tersebut sebagai model *Non-linear Integer-Programming* (NIP) dan mengusulkan GA sebagai metode untuk menyelesaikannya. Kemudian Saleh et al. (2014) melanjutkan penelitian Yu et al. (2013) dengan mengoptimasi kasus *single vendor multi retailer* menggunakan metode *Genetic Algorithm* (GA) agar lebih efektif dan efisien.

Selain metode GA terdapat metode optimasi lainnya yaitu metode PSO. Seperti yang dilakukan Mousavi et al. (2013) menggunakan metode PSO untuk menemukan jumlah produksi yang optimal pada periode yang berbeda untuk meminimalkan total biaya persediaan (biaya pemesanan, penyimpanan, *stockout* dan pembelian). Karena tidak ada acuan dalam literatur untuk memverifikasi dan memvalidasi hasil yang didapatkan, Mousavi et al. (2013) menggunakan metode GA sebagai perbandingan. Dari hasil perbandingan contoh numerik kedua metode memiliki kinerja statistik yang serupa, namun PSO cenderung menjadi algoritma yang lebih baik hampir di semua masalah. Selanjutnya penelitian lainnya dilakukan oleh Sadeghi et al. (2014) yang menggunakan PSO dalam mengoptimasi model VMI dengan permintaan yang berupa *fuzzy* untuk menemukan kuantitas pesanan *retailer* yang optimal sehingga biaya total

inventori dan biaya transportasi dapat diminimalkan. Namun karena tidak ada acuan yang tersedia pada kajian literatur Sadeghi et al. (2014) menggunakan GA untuk memverifikasi solusi yang diperoleh PSO.

Kemudian Pasandideh et al. (2011) melanjutkan hasil penelitian sebelumnya Pasandideh & Niaki (2008) dengan menggunakan metode optimasi GA terhadap model VMI multi-produk multi-kendala pada level *single-supplier single-retailer*. Pada penelitian ini yang menjadi batasan-batasan model adalah kapasitas gudang dan jumlah pesanan untuk semua *item*. Metode GA yang diusulkan digunakan untuk mencari jumlah pesanan (Q) dan level maksimum *backorder* (b) yang optimal sehingga didapatkan total biaya persediaan yang minimum. Namun, dari hasil total biaya persediaan optimal berdasarkan metode GA yang diusulkan ternyata jumlah Q dan b tidak memenuhi batasan-batasan model yaitu ruang gudang yang tersedia (F) dan total jumlah order untuk semua produk (M). Model VMI yang dikembangkan oleh Pasandideh et al. (2011) sebagai berikut :

Notasi:

Untuk produk $j = 1, 2, \dots, n$

TC_{VMI}	Total biaya sistem VMI
A_S	Biaya pemesanan tetap <i>supplier</i> per unit untuk semua produk
A_R	Biaya pemesanan tetap <i>retailer</i> per unit untuk semua produk
Q_j	Kuantitas pesanan produk j
D_j	Jumlah permintaan <i>retailer</i> untuk produk j
b_j	Tingkat maksimum <i>backorder</i> untuk produk j pada siklus sistem VMI
π	Biaya tetap <i>backorder</i> per unit

$\hat{\pi}$	Biaya tetap <i>backorder</i> per unit per waktu
h_{jR}	Biaya simpan per unit produk j pada toko <i>retailer</i> dalam satu periode
KR_{VMI}	Biaya inventori <i>retailer</i> dalam sistem VMI
KS_{VMI}	Biaya inventori <i>supplier</i> dalam sistem VMI
f_j	Ruang yang dipakai per unit produk j
F	Total ruang penyimpanan yang tersedia untuk semua produk
M	Total jumlah pesanan untuk semua produk
n	Jumlah produk

Berdasarkan notasi diatas Pasandideh et al. (2011) mengembangkan model VMI multi-produk multi-kendala pada level *single-supplier single-retailer* dengan model matematik sebagai berikut :

$$KR_{VMI} = 0, \quad (2.1)$$

$$KS_{VMI} = \sum_{j=1}^n \left(\frac{A_{jS}D_j}{Q_j} + \frac{A_{jR}D_j}{Q_j} + \frac{h_{jR}}{2Q_j} (Q_j - b_j)^2 + \frac{\hat{\pi} b_j^2}{2Q_j} + \frac{\pi b_j D_j}{Q_j} \right), \quad (2.2)$$

$$TC_{VMI} = KR_{VMI} + KS_{VMI}$$

$$= \sum_{j=1}^n \left(\frac{D_j}{Q_j} (A_{jS} + A_{jR}) + \frac{h_{jR}}{2Q_j} (Q_j - b_j)^2 + \frac{\hat{\pi} b_j^2}{2Q_j} + \frac{\pi b_j D_j}{Q_j} \right). \quad (2.3)$$

Berdasarkan kajian literatur diatas memunculkan ide untuk membandingkan total biaya persediaan yang dioptimasi dengan metode GA pada penelitian Pasandideh et al. (2011) dengan metode PSO.

2.2 Kajian Deduktif

2.2.1 *Supply Chain Management*

Menurut Mentzer et al. (2001) *Supply Chain Management* merupakan koordinasi sistem strategis fungsi bisnis tradisional dan taktik seluruh fungsi-fungsi bisnis suatu perusahaan untuk tujuan meningkatkan kinerja jangka panjang bagi keseluruhan pihak yang terlibat dalam *supply chain* tersebut. Sedangkan menurut Simchi-Levi (2000) *Supply Chain Management* diartikan sebagai rangkaian pendekatan yang digunakan untuk mengintegrasikan *supplier*, pemanufaktur, gudang, dan *retailer* secara efektif agar persediaan barang dapat diproduksi dan didistribusikan pada jumlah, lokasi, dan waktu yang tepat sehingga dapat menurunkan biaya keseluruhan sistem dan memenuhi kebutuhan pelanggan.

SCM pada hakekatnya merupakan pengembangan dari konsep manajemen logistik, dimana urutan operasional dalam SCM lebih panjang daripada manajemen logistik. Sebagai konsep baru SCM memandang permasalahan logistik sebagai masalah yang lebih luas yang terbentang sangat panjang sejak dari bahan mentah hingga barang jadi yang dipakai konsumen akhir. Sedangkan konsep manajemen logistik sebagai konsep lama memandang permasalahan logistik sebagai masalah *intern* masing-masing perusahaan sehingga dalam pemecahannya dititikberatkan pada masing-masing perusahaan secara individu (Indrajit & Djokopranoto, 2002).

Memaksimalkan nilai keseluruhan yang dihasilkan untuk memenuhi kebutuhan dan permintaan pelanggan merupakan tujuan dari SCM. Selain itu SCM bertujuan untuk meminimalkan biaya-biaya yang ada dalam sistem yaitu biaya

pemesanan, biaya inventori, biaya bahan baku, dan biaya transportasi (Chopra & Meindl, 2004). Ada tiga keuntungan *supply chain* menurut Indrajit & Djokopranoto (2002) antara lain :

1. Mengurangi inventori barang

Inventori merupakan bagian paling besar dari asset perusahaan yang berkisar antara 30%-40%. Sedangkan biaya penyimpanan barang (*inventory carrying cost*) berkisar antara 20%-40% dari nilai barang yang disimpan. Oleh karena itu usaha dan cara harus dikembangkan untuk menekan penimbunan barang di dalam gudang agar biaya dapat ditekan menjadi seminimal mungkin.

2. Menjamin kelancaran penyediaan barang

Kelancaran barang yang perlu dijamin adalah mulai dari barang asal (pabrik pembuat), *supplier*, perusahaan sendiri, *wholesaler*, *retailer*, sampai kepada konsumen akhir. Jadi, rangkai perjalanan dari bahan baku sampai menjadi barang jadi dan diterima oleh konsumen akhir merupakan suatu mata rantai yang panjang yang perlu dikelola dengan baik.

3. Menjamin mutu

Mutu barang jadi (*finished goods*) ditentukan tidak hanya oleh proses produksi barang tersebut, tetapi juga oleh mutu bahan mentahnya dan mutu keamanan dalam pengiriman. Jaminan mutu ini juga merupakan serangkaian mata rantai panjang yang harus dikelola dengan baik.

4. Mengurangi jumlah *supplier*

Mengurangi umlah *supplier* bertujuan untuk mengurangi ketidakseragaman, biaya-biaya negosiasi, dan palacakan (*tracking*).

5. Mengembangkan *supplier partnership* atau *strategic alliance*

Kelancaran pergerakan barang dalam *supply chain* dapat terjamin jika adanya kerjasama dengan *supplier* (*supplier partnership*) dan juga dengan mengembangkan *strategic alliance*.

Dalam konsep SCM memiliki beberapa ruang lingkup pembahasan yang sangat berpengaruh terhadap performa *supply chain* itu sendiri. Menurut Chopra & Meindl (2004) ruang lingkup pembahasan tersebut adalah sebagai berikut :

1. Inventori

Inventori merupakan salah satu penggerak *supply chain* yang penting karena perubahan kebijakan inventori dapat mengubah secara drastis tingkat responsivitas dan efisiensi *supply chain*.

2. Transportasi

Transportasi adalah memindahkan inventori dari satu titik ke titik lain dalam *supply chain*. Transportasi terdiri atas banyak kombinasi dari model dan bentuk yang memiliki keunggulan masing-masing. Pemilihan transportasi juga mempunyai dampak yang besar dalam tingkat responsifitas dan efisiensi *supply chain*.

3. Fasilitas

Fasilitas adalah tempat-tempat dalam jaringan *supply chain* dimana inventori disimpan, dirakit atau diproduksi. Dua jenis umum dari fasilitas adalah tempat produksi dan tempat penyimpanan. Bila perusahaan memilih tingkat efisiensi tinggi, maka memiliki lebih sedikit gudang. Jadi penentuan fasilitas mempunyai dampak besar dalam tingkat responsifitas dan efisiensi *supply chain*.

4. Informasi

Informasi terdiri dari data dan analisis berkaitan dengan inventori, transportasi, fasilitas, dan pelanggan di seluruh *supply chain*. Informasi menyajikan pihak manajemen kesempatan untuk membuat *supply chain* lebih responsif dan efisien.

Adanya koordinasi yang baik antara tiap pihak di dalam *supply chain* menjadi sangat penting melihat ruang lingkup pembahasan mengenai *supply chain* sangat luas. Pada beberapa penelitian banyak dijumpai kajian mengenai optimalisasi inventori pada pihak-pihak yang terlibat dalam suatu *supply chain* baik itu *supplier*, pemanufaktur, distributor maupun *retailer*.

2.2.2 Sistem Inventori

Inventori adalah sumber daya penting perusahaan yang akan digunakan untuk mengantisipasi fluktuasi permintaan dimasa yang akan datang. Dalam masalah inventori, hal yang perlu mendapat perhatian adalah bahan-bahan dan komponen-komponen yang diperlukan dalam pembuatan produk, seberapa banyak dan kapan digunakan dalam proses produksi suatu produk (Tersine, 1994). Arti inventori dapat ditujukan pada barang-barang yang tersedia untuk dijual dalam kegiatan bisnis normal, namun dalam perusahaan manufaktur inventori dapat ditujukan untuk barang-barang yang berkaitan dalam kegiatan proses produksi (Skousen et al., 2000). Dalam perusahaan manufaktur inventori dijumpai dalam berbagai bentuk diantaranya *raw material* sebagai input untuk proses produksi, bahan penolong untuk membantu terlaksananya proses produksi, suku cadang untuk menggantikan komponen yang

mengalami kerusakan, serta dalam bentuk barang setengah jadi (*work in process*) maupun barang jadi (*finished good*) yang siap dipasarkan kepada konsumen (Bahagia, 2006). Pada dasarnya adanya inventori menimbulkan biaya-biaya yang harus dikeluarkan oleh suatu perusahaan sehingga menurut Elsayed & Boucher (1994) tujuan utama dari analisis sistem inventori adalah untuk menemukan jawaban atas dua pertanyaan berikut :

1. Berapa banyak yang harus dipesan atau diproduksi?
2. Kapan seharusnya pesanan ditempatkan sedemikian rupa sehingga total biaya persediaan dapat diminimalkan?

2.2.3 Fungsi Inventori

Menurut Rangkti (2004) terdapat tiga fungsi inventori yaitu :

1. Fungsi *Decoupling*

Merupakan inventori yang memungkinkan perusahaan dapat memenuhi permintaan pelanggan yang tidak pasti tanpa tergantung pada *supplier*.

2. Fungsi *Economic Lot Sizing*

Merupakan inventori yang pengadaannya dalam jumlah besar daripada kebutuhan pada saat itu. Hal ini dilakukan agar perusahaan mendapatkan keuntungan berupa potongan harga karena pembelian dalam jumlah besar serta penghematan biaya pengangkutan per unit yang lebih rendah.

3. Fungsi Antisipasi

Merupakan inventori yang berfungsi sebagai pengaman bagi perusahaan yang sering menghadapi ketidakpastian jangka waktu pengiriman maupun

ketidakpastian permintaan dari pelanggan. Inventori ini penting agar kelancaran proses produksi tidak terganggu.

2.2.4 Biaya-biaya Inventori

Secara umum biaya-biaya inventori merupakan semua pengeluaran dan kerugian yang timbul sebagai akibat dari adanya inventori tersebut. Biaya-biaya inventori dapat digolongkan menjadi beberapa jenis yaitu (Chase et al., 2001) :

1. Biaya Penyimpanan (*Carrying/Holding Costs*)

Merupakan biaya yang timbul karena adanya inventori yang disimpan sehingga perlu adanya usaha untuk mengamankan inventori tersebut dari kerusakan, keusangan, dan kehilangan. Yang termasuk biaya penyimpanan diantaranya adalah biaya sewa gedung, biaya administrasi pergudangan, biaya listrik, biaya kerusakan, kehilangan maupun penyusutan barang selama dalam penyimpanan.

2. Biaya Penyiapan (*Set up Costs*)

Merupakan biaya yang timbul dari aktivitas proses produksi serta apabila bahan-bahan tidak dibeli melainkan dibuat sendiri. Yang termasuk biaya penyiapan diantaranya biaya mesin menganggur, biaya persiapan tenaga kerja langsung, serta biaya penjadwalan.

3. Biaya Pemesanan (*Ordering Costs*)

Merupakan biaya-biaya yang dikeluarkan sehubungan dengan kegiatan pemesanan bahan maupun barang, sejak dari penempatan pemesanan sampai tersedianya barang di gudang. Biaya pemesanan tidak tergantung dari jumlah yang dipesan namun tergantung dari berapa kali pesanan dilakukan yang dinyatakan dalam rupiah per pesanan. Yang termasuk biaya pemesanan

diantaranya biaya pengiriman ke gudang, biaya surat menyurat, serta biaya telepon.

4. Biaya Kekurangan (*Shortage Costs*)

Merupakan biaya-biaya yang timbul karena bahan-bahan maupun barang-barang tidak tersedia saat ada permintaan sehingga terjadi *stockout*. Saat terjadi *stockout* perusahaan akan mengalami kerugian berupa biaya kehilangan kesempatan (*lost sales*) maupun biaya kehilangan pelanggan yang kecewa.

2.2.5 Perkembangan *Vendor Managed Inventory*

Sebuah sistem VMI telah banyak diadopsi oleh banyak industri selama bertahun-tahun. Kesuksesan VMI dimulai oleh perusahaan Amerika yaitu Campbell Soup dan Johnson & Johnson, dan juga oleh perusahaan Eropa yaitu Barilla yang merupakan perusahaan pasta terkemuka di Eropa. Kisah sukses klasik untuk sistem VMI lainnya ditemukan dalam hubungan kemitraan antara Wal-Mart dan Procter & Gamble (P&G) pada tahun 1985. Kemitraan keduanya secara dramatis meningkatkan pengiriman oleh P&G menjadi tepat waktu dan meningkatkan penjualan Wal-Mart (Buzzell & Ortmeyer, 1995). Kemudian Kmart juga mengikuti sistem tersebut dan pada tahun 1992 Kmart telah mengembangkan lebih dari 200 mitra VMI (Schonberger, 1996). Selain ritel industri, VMI juga diadopsi oleh perusahaan-perusahaan kimia terkemuka untuk meningkatkan efisiensi *supply chain* dan untuk meningkatkan hubungan pelanggan dan *supplier* (Challener, 2000).

VMI adalah salah satu dari bentuk kerjasama yang bertujuan untuk meningkatkan efisiensi inventori dalam *supply chain*. Selain itu, VMI merupakan

tingkat tertinggi kemitraan di mana *vendor* adalah pembuat keputusan utama dalam penempatan order dan manajemen inventori. Dibawah sistem VMI, *supplier* memutuskan tingkat persediaan dari masing-masing produk dan kebijakan-kebijakan untuk mengatur inventori yang tepat pada level ini (Simchi-Levi, 2000). Pada sistem VMI informasi penjualan dan tingkat persediaan pihak *retailer* sangat penting karena berdasarkan informasi tersebut *supplier* dapat menentukan jumlah dan waktu pengisian yang tepat pada setiap periode (Dong & Xu, 2002).

Mengelola hubungan kemitraan dan transparansi manajemen dalam hubungan antara *retailer* dan *supplier* merupakan faktor penting untuk mencapai keberhasilan dan mendapatkan keuntungan dari sistem VMI. Transparansi dalam hubungan yang mencakup pertukaran informasi penjualan, persediaan, biaya, serta pengetahuan dapat dijaga dengan melakukan sebuah tinjauan reguler dan mekanisme negosiasi antara *retailer* dan *supplier*. Selain itu, sistem pendukung keputusan taktis dapat dikembangkan untuk menjamin transparansi data dan untuk membantu dalam mengkoordinasikan kebijakan manajemen inventori dan transportasi (Lamming et al., 2001). Dalam menerapkan sistem VMI terdapat beberapa kelemahan yang harus diperhitungkan. Aichlmayr (2000) menyelidiki pelaksanaan VMI dan melaporkan bahwa dari sepuluh implementasi VMI, hanya tiga atau empat mencapai sukses besar, kemudian tiga atau empat mendapatkan beberapa manfaat tetapi tidak sebanyak seperti yang diharapkan, serta dua atau tiga tidak memberi manfaat apapun. Kelemahan utama dari VMI antara lain kerahasiaan informasi antara *retailer* dan *supplier*, resiko hilangnya kontrol oleh *retailer*, kenaikan biaya administrasi *vendor* serta minimnya manfaat yang didapatkan *supplier*.

Penerapan sistem VMI memiliki motivasi yaitu bahwa *retailer* dan *supplier* keduanya bekerja sama untuk memaksimalkan daya saing *supply chain*. Sehingga produktivitas dan perbaikan *service level* dapat menyebabkan penjualan meningkat dan mendapatkan keuntungan yang lebih besar. Namun manfaat yang paling utama dari VMI adalah pengurangan biaya inventori bagi *retailer* dan pengurangan biaya total bagi *supplier* (Tyan & Wee, 2003). Dalam pengaplikasian VMI terhadap sistem inventori masih ditemukan beberapa permasalahan. Seperti masih adanya asumsi-asumsi yang melekat sehingga perlu dilakukan optimasi. Kemudian model-model VMI yang dikembangkan sebagian besar merupakan model *non-linear* yang tidak dapat diselesaikan secara eksak. Sehingga perlu adanya metode-metode yang lebih baik untuk menyelesaikannya. Dalam belakang ini digunakan metode-metode heuristik yang digunakan dalam mengoptimasi model-model VMI diantaranya *Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO).

2.2.6 Genetic Algorithm

Genetic Algorithm atau algoritma genetika pertama kali diperkenalkan oleh John Holland dari Universitas Michigan pada awal 1970-an di New York, Amerika Serikat. Kemudian Jhon Holland bersama murid-murid serta rekan kerjanya menghasilkan buku yang berjudul “*Adaption in Natural and Artificial System*” pada tahun 1975. Kemudian David Goldberg (1989) yaitu salah satu murid Jhon Holland mempopulerkan algoritma genetika ini dan mendefinisikan algoritma genetika sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetik alam.

Algoritma genetika sebagai cabang dari algoritma evolusi merupakan metode *adaptive* yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau “siapa yang kuat, dia yang bertahan (*survive*)”. Dengan meniru teori evolusi ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata. Algoritma genetika menggunakan analogi secara langsung dari kebiasaan yang alami yaitu seleksi alam. Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-individu, yang masing-masing individu mempresentasikan sebuah solusi yang mungkin bagi persoalan yang ada. Dalam hal ini, individu dilambangkan dengan sebuah nilai *fitness* yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada.

Pertahanan yang tinggi dari individu memberikan kesempatan untuk melakukan reproduksi melalui perkawinan silang dengan individu yang lain dalam populasi tersebut. Individu baru yang dihasilkan dalam hal ini dinamakan keturunan, yang membawa beberapa sifat dari induknya. Sedangkan individu dalam populasi yang tidak terseleksi dalam reproduksi akan mati dengan sendirinya. Dengan jalan ini, beberapa generasi dengan karakteristik yang bagus akan bermunculan dalam populasi tersebut, untuk kemudian dicampur dan ditukar dengan karakter yang lain. Dengan mengawinkan semakin banyak individu, maka akan semakin banyak kemungkinan terbaik yang dapat diperoleh.

Sebelum algoritma genetika dapat dijalankan, maka sebuah kode harus dirancang untuk merepresentasikan persoalan. Kemudian titik solusi dalam ruang

permasalahan dikodekan dalam bentuk kromosom yang terdiri atas komponen genetik terkecil yaitu gen. dengan teori evolusi dan teori genetika, didalam penerapan algoritma genetika akan melibatkan beberapa operator, yaitu :

1. Operasi evolusi yang melibatkan proses seleksi (*selection*) di dalamnya.
2. Operasi genetika yang melibatkan operator pindah silang (*crossover*) dan mutasi (*mutation*).

Untuk memeriksa hasil optimasi, maka dibutuhkan fungsi *fitness* yang menandakan gambaran hasil atau solusi yang sudah dikodekan. Selama berjalan, induk harus digunakan untuk reproduksi, pindah silang dan mutasi untuk menciptakan keturunan. Jika algoritma genetika didesain secara baik, populasi akan mengalami konvergensi dan akan didapatkan sebuah solusi yang optimal.

Dalam algoritma genetika terdapat beberapa komponen utama antara lain :

1. Teknik Pengkodean

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

2. Membangkitkan populasi awal

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal. Syarat-syarat yang harus dipenuhi untuk

menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap individunya.

3. Fungsi obyektif

Suatu individu dievaluasi berdasarkan suatu fungsi obyektif sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang dicari adalah maksimasi fungsi maka nilai fungsi *fitness* sama dengan nilai fungsi obyektif. Sedangkan untuk mencari solusi minimasi, fungsi obyektif perlu dimodifikasi dengan rumus :

$$\text{Fungsi } fitness = \frac{1}{1 + \text{Fungsi obyektif}} \quad (2.4)$$

4. Seleksi

Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses *crossover* dan mutasi. Seleksi digunakan untuk mendapatkan calon induk yang baik. “Induk yang baik akan menghasilkan keturunan yang baik”. Semakin tinggi nilai *fitness* suatu individu semakin besar kemungkinannya untuk terpilih. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Nilai *fitness* ini nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Masing-masing individu dalam wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai obyektif dirinya sendiri terhadap nilai obyektif dari semua individu dalam wadah seleksi tersebut. Terdapat beberapa metode seleksi diantaranya *roulette wheel*, *rank-based fitness assignment*, *local selection*, *truncation selection*, *tournament selection*.

5. *Crossover*

Setelah individu-individu terpilih dalam proses seleksi, selanjutnya pada individu-individu tersebut akan dilakukan perkawinan silang. Kawin silang atau *crossover* adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. *Crossover* menghasilkan titik baru dalam ruang pencarian yang siap untuk diuji. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan penyilangan dengan P_c antara 0,6 sampai dengan 0,95. Jika *crossover* tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan. Terdapat beberapa model *crossover* yang sesuai untuk representasi kromosom yaitu *crossover* satu titik, *crossover* banyak titik, *crossover aritmatika*, serta *crossover* untuk representasi kromosom permutasi.

6. Mutasi

Mutasi merupakan operator pada algoritma genetika yang berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Kromosom anak dimutasi dengan menambahkan nilai random yang sangat kecil, dengan probabilitas yang rendah. Probabilitas mutasi (P_m) didefinisikan sebagai persentase dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikannya banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika P_m terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila P_m terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma kehilangan kemampuan untuk belajar dari histori pencarian. Kromosom hasil mutasi harus diperiksa, apakah masih berada

pada domain solusi, dan bila perlu bias dilakukan perbaikan. Terdapat beberapa jenis mutasi diantaranya mutasi biner, mutasi bilangan real, dan mutasi kromosom permutasi.

2.2.7 Particle Swarm Optimization

Particle Swarm Optimization atau disingkat PSO adalah suatu teknik optimasi stokastik berdasarkan suatu populasi yang pertama kali dikemukakan oleh Kennedy & Eberhart (1995) kemudian dikembangkan juga oleh Clerc (1999). Lahirnya PSO terinspirasi dari perilaku social kawanan burung atau sekumpulan ikan (Wati, 2011). Algoritma PSO meniru perilaku sosial organisme ini yang terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Misalnya seekor burung dalam kawanan burung dapat disebut sebagai individu atau partikel. Setiap partikel berperilaku secara terdistribusi dengan menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya. Dengan demikian, jika satu partikel atau seekor burung menemukan jalan yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan tersebut meskipun mereka jauh di kelompok tersebut.

Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik yaitu posisi dan kecepatan. Dalam ruang atau *space* tertentu setiap partikel bergerak dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi obyektif. Setiap partikel menyampaikan informasi

atau posisi bagus kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut. Jadi algoritma PSO dikembangkan berdasarkan pada model berikut ini (Santosa, 2006):

1. Ketika seekor burung mendekati target atau makanan (maksimasi atau minimasi suatu fungsi obyektif) secara cepat mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu
2. Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung
3. Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya.

Berdasarkan ketiga model diatas maka dalam PSO dapat disimulasikan yaitu sebanyak p partikel disebar secara acak pada ruang solusi multidimensi yang ada. Posisi partikel i saat waktu t , yaitu $x_i(t)$ akan diperbaiki menurut persamaan posisi sebagai berikut :

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2.5)$$

Dimana $v_i(t+1)$ adalah kecepatan partikel yang dapat dihitung dengan persamaan berikut :

$$v_i(t + 1) = wv_i(t) + c_1r_1(pbest_i(t) - x_i(t)) + c_2r_2(gbest(t) - x_i(t)) \quad (2.6)$$

Titik $pbest_i(t)$ adalah solusi lokal terbaik yang dicapai oleh partikel i sampai saat t , dan merepresentasikan kontribusi kognitif terhadap vektor $v_i(t+1)$. Sedangkan titik $gbest(t)$ adalah solusi global terbaik yang telah dicapai diantara partikel-partikel

sampai saat t dan merepresentasikan kontribusi sosial terhadap vector kecepatan. Kemudian r_1, r_2 adalah bilangan random yang terdistribusi seragam antara 0 dan 1. Sedangkan w, c_1 , dan c_2 termasuk sebagai parameter-parameter dalam algoritma PSO. Selain itu yang termasuk parameter dalam PSO lainnya adalah jumlah populasi ($Npop$), iterasi (i), dan *constriction factor* (C) yang dapat dijelaskan sebagai berikut :

1. Bobot inersia (w)

Bobot inersia yang besar memfasilitasi eksplorasi global atau pencarian dalam daerah yang luas. Sedangkan bobot inersia yang kecil menghasilkan eksplorasi local atau pencarian dalam daerah yang sempit. Oleh karena itu, nilai w merupakan faktor kritis yang menentukan perilaku konvergen algoritma PSO. Untuk itu, direkomendasikan untuk memilih nilai w yang besar pada awal iterasi agar menghasilkan eksplorasi global pada ruang solusi, selanjutnya nilai w diturunkan secara bertahap untuk mendapatkan solusi yang lebih baik. Dalam penerapannya bobot inersia dapat menggunakan rumus berikut (Haupt & Haupt, 2004) :

$$w_i = \frac{maxit - i}{maxit} \quad (2.7)$$

Dimana :

w_i = bobot inersia pada iterasi ke- i

$maxit$ = jumlah maksimum iterasi

i = iterasi

2. Parameter kognitif (c_1)

Parameter kognitif merupakan memori partikel jangka pendek. Parameter ini menunjukkan inklinasi partikel untuk mengulang perilaku sebelumnya yang

terbukti sukses pada partikel tersebut. Hal ini juga menunjukkan adanya pengaruh memori partikel tersebut. Pada umumnya besarnya $c_1 = 2$.

3. Parameter sosial (c_2)

Parameter sosial merupakan perilaku memaksa pada kawanan lain bagi sebuah partikel atau disebut *peer pressure*. Parameter sosial ini menunjukkan inklinasi partikel meniru atau mengemulasi perilaku partikel lain yang telah sukses. Hal ini menunjukkan adanya pengaruh tetangga partikel. Pada umumnya besarnya c_2 sama dengan c_1 yaitu sebesar 2.

4. Jumlah populasi (N_{pop})

Jumlah populasi merupakan jumlah dari kumpulan individu atau partikel. Untuk mengurangi jumlah evaluasi fungsi yang diperlukan dalam menemukan solusi, sebaiknya N_{pop} tidak terlalu besar, tetapi juga tidak terlalu kecil agar ada banyak kemungkinan posisi menuju solusi terbaik.

5. Iterasi (i)

Iterasi merupakan jumlah proses perulangan. Terdapat banyak cara untuk menentukan kapan iterasi tersebut berhenti diantaranya ketika PSO telah menemukan nilai optimum atau kesalahan minimum yang diinginkan, dan ketika PSO telah mencapai iterasi maksimum yang jumlahnya telah ditentukan sebelumnya.

6. Constriction Factor (C)

Parameter lainnya dalam PSO adalah *constriction factor*, parameter ini diperkenalkan pertama kali oleh Clerc (1999) dengan tujuan dapat menjamin suatu penelusuran dalam algoritma PSO untuk konvergen lebih cepat. Dalam menerapkan parameter *Constriction Factor* ke dalam algoritma PSO digunakan rumus sebagai berikut (Eberhart & Shi, 2001) :

$$C = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \text{ dengan } \varphi = c_1 + c_2, \varphi > 4 \quad (2.8)$$

Pada umumnya ketika peneliti menggunakan parameter *constriction factor*, nilai φ ditetapkan sebesar 4,1 maka untuk nilai c_1 dan c_2 diset sebesar 2,05 sehingga nilai C menjadi 0,729. Kemudian untuk persamaan kecepatan berubah menjadi sebagai berikut :

$$v_i(t+1) = C \left(wv_i(t) + c_1r_1(pbest_i(t) - x_i(t)) + c_2r_2(gbest(t) - x_i(t)) \right) \quad (2.9)$$

Dalam beberapa tahun terakhir, PSO telah berhasil diterapkan di banyak bidang penelitian dan aplikasi. Misalnya seperti yang dilakukan Sadeghi et al., (2014) yang menggunakan PSO dalam mengoptimasi masalah VMI dan transportasi. Hal ini disebabkan karena PSO dapat memberikan hasil yang lebih baik dengan cara yang cepat dan sederhana dibandingkan dengan metode lain.

Langkah-langkah PSO dalam menyelesaikan masalah optimasi dapat dijelaskan sebagai berikut :

1. Inisialisasi

Dilakukan dengan membangkitkan populasi partikel sebanyak N_{pop} yang masing-masing berisi variabel sebanyak n_{par} dengan posisi dan kecepatan secara random pada ruang solusi. Kemudian tentukan nilai dari parameter PSO yaitu c_1 dan c_2 yang kemudian didapatkan besarnya *constriction factor*.

2. Perhitungan Nilai *Fitness*

Perhitungan nilai *fitness* digunakan untuk mengetahui baik buruknya posisi partikel dengan dievaluasi menggunakan fungsi objektif.

3. Tentukan *pbest*

Dari hasil evaluasi fungsi objektif diperoleh nilai *fitness* sebanyak N_{pop} . Selanjutnya *pbest* ditentukan jika nilai *fitness* dari sebuah partikel lebih baik (lebih kecil) dari nilai *fitness* sebelumnya maka *pbest* akan diset ke nilai *fitness* yang baru.

4. Tentukan *gbest*

Jika nilai minimum *fitness* dari seluruh partikel lebih baik (lebih kecil) dari nilai minimum *fitness* dari seluruh partikel pada iterasi sebelumnya maka nilai *gbest* akan diset sebagai nilai minimum *fitness* yang baru tersebut.

5. Cek kondisi berhenti

Jika kondisi berhenti tercapai maka keluar dari proses perulangan atau iterasi, jika belum tercapai lanjutkan ke iterasi berikutnya.

6. Perhitungan kecepatan

Perhitungan kecepatan untuk masing-masing partikel sesuai dengan persamaan 2.9

7. Perhitungan posisi

Perhitungan posisi yang baru untuk masing-masing partikel sesuai dengan persamaan 2.5

BAB III

METODOLOGI PENELITIAN

Pada bab ini menjelaskan mengenai metode penelitian yang disusun dari beberapa sub bab seperti fokus kajian, model kajian yang digunakan dalam penelitian, jenis dan sumber data, algoritma PSO dalam mengoptimasi model VMI, alat-alat penelitian, dan kerangka penelitian.

3.1 Fokus Kajian

Fokus kajian pada penelitian ini adalah optimasi menggunakan metode *Particle Swarm Optimization* terhadap model *Vendor Managed Inventory* multi-produk multi-kendala pada *single-supplier single-retailer* yang telah dioptimasi oleh Pasandideh et al. (2011) menggunakan metode *Genetic Algorithm*. Sehingga kemudian dapat dibandingkan hasil optimasi antara metode GA dan metode PSO.

3.2 Model Kajian

Model VMI yang telah dikembangkan dan dioptimasi oleh Pasandideh et al. (2011) bertujuan untuk meminimasi total biaya inventori (*TC*) *supplier* dan *retailer*. Dengan menentukan kuantitas pemesanan (Q) dan tingkat *backorder* (b) yang optimum maka total biaya inventori dapat diminimasi. Dalam model VMI ini terdapat berbagai jenis produk sehingga dalam penentuan Q dan b yang optimum terdapat dua hal yang menjadi kendala yaitu kapasitas gudang (F) dan jumlah pesanan untuk semua produk (M). Model

yang akan dioptimasi telah ditunjukkan pada persamaan 2.1 pada Bab 2. Namun untuk mempermudah dalam presentasi maka persamaan tersebut ditulis lebih rinci beserta batasan-batasannya yang dapat ditunjukkan sebagai berikut :

Notasi:

Untuk produk $j = 1, 2, \dots, n$

TC_{VMI}	Total biaya sistem VMI
A_S	Biaya pemesanan tetap <i>supplier</i> per unit untuk semua produk
A_R	Biaya pemesanan tetap <i>retailer</i> per unit untuk semua produk
Q_j	Kuantitas pesanan produk j
D_j	Jumlah permintaan <i>retailer</i> untuk produk j
b_j	Tingkat maksimum <i>backorder</i> untuk produk j pada siklus sistem VMI
π	Biaya tetap <i>backorder</i> per unit
$\hat{\pi}$	Biaya tetap <i>backorder</i> per unit per waktu
h_{jR}	Biaya simpan per unit produk j pada toko <i>retailer</i> dalam satu periode
KR_{VMI}	Biaya inventori <i>retailer</i> dalam sistem VMI
KS_{VMI}	Biaya inventori <i>supplier</i> dalam sistem VMI
f_j	Ruang yang dipakai per unit produk j
F	Total ruang penyimpanan yang tersedia untuk semua produk
M	Total jumlah pesanan untuk semua produk
n	Jumlah produk

$$\text{Min } TC_{VMI} = \sum_{j=1}^n \left(\frac{D_j}{Q_j(A_{jS} + A_{jR})} + \frac{h_{jR}}{2Q_j} (Q_j - b_j)^2 + \frac{\hat{\pi} b_j^2}{2Q_j} + \frac{\pi b_j D_j}{Q_j} \right) \quad (3.1)$$

Dengan syarat :

$$\sum_{j=1}^n f_j Q_j \leq F \quad (3.2)$$

$$\sum_{j=1}^n \frac{D_j}{Q_j} \leq M \quad (3.3)$$

$$Q_j, b_j \geq 0 : \text{bilangan bulat} \quad (3.4)$$

Berdasarkan model diatas pencarian besarnya Q dan b yang optimal dilakukan dengan menggunakan metode *Particle Swarm Optimization* yang kemudian hasilnya akan dibandingkan dengan hasil metode *Genetic Algorithm* yang telah dilakukan Pasandideh et al. (2011). Adapun parameter dalam mengkomparasi kedua metode ini adalah besarnya total biaya inventori (TC_{VMI}).

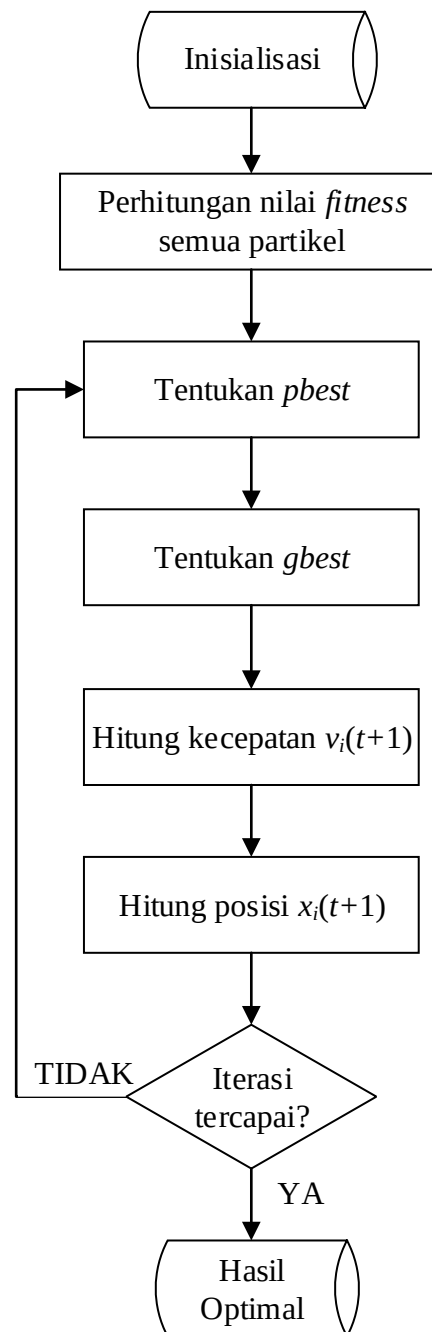
3.3 Jenis dan Sumber Data

Data yang digunakan adalah data numerik yang telah dipaparkan oleh Pasandideh et al. (2011). Adapun data-data adalah sebagai berikut (seperti yang tersirat dalam Notasi model diatas) :

$$A_S, A_R, D_j, \pi, \hat{\pi}, h_{jR}, f_j, F, M$$

3.4 Optimasi Model dengan *Particle Swarm Optimization*

Secara umum proses algoritma PSO dalam mengoptimasi model kajian dapat ditunjukkan pada Gambar 3.1 sebagai berikut :



Gambar 3.1 Diagram Alir Algoritma PSO

Berdasarkan Gambar 3.1 proses algoritma PSO dalam mengoptimasi model kajian dapat dijelaskan sebagai berikut:

1. Inisialisasi

Inisialisasi merupakan langkah pertama dalam algoritma PSO yaitu dengan membangkitkan populasi partikel (N_{pop}) sebanyak 80, 100, 120, 140, 160, dan 180 sesuai dengan yang dipaparkan Pasandideh et al. (2011). Kemudian dilakukan percobaan untuk menentukan nilai parameter *constriction factor* sesuai rumus dibawah ini :

$$C = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \text{ dengan } \varphi = c_1 + c_2, \varphi > 4 \quad (3.5)$$

Pada umumnya ketika peneliti menggunakan parameter *constriction factor*, nilai φ ditetapkan sebesar 4,1 maka untuk nilai c_1 dan c_2 diset sebesar 2,05 sehingga nilai C menjadi 0,729 (Clerc, 1999). Namun menurut James & Russell (1995) nilai c_1 dan c_2 ditetapkan sebesar 2. Berdasarkan kajian ini memunculkan ide untuk melakukan percobaan terhadap nilai c_1 dan c_2 yang jika dijumlahkan nilai φ sebesar 4 atau 4,1. Adapun nilai-nilai c_1 dan c_2 yang akan dilakukan percobaan dapat ditunjukkan pada Tabel 3.1 sebagai berikut :

Tabel 3.1 Data c_1 dan c_2 untuk Percobaan			
Percobaan	c_1	c_2	$c_1 + c_2 (\varphi)$
1	2	2	4
2	2,05	2,05	4,1
3	2	2,1	4,1
4	2,1	2	4,1
5	1,9	2,1	4
6	1,9	2,2	4,1
7	1,8	2,2	4

Percobaan	c_1	c_2	$c_1 + c_2 (\varphi)$
8	2,1	1,9	4
9	2,2	1,9	4,1
10	2,2	1,8	4

2. Perhitungan Nilai *Fitness*

Perhitungan nilai *fitness* digunakan untuk mengetahui baik buruknya posisi partikel dengan dievaluasi menggunakan fungsi objektif TC_{VMI} sesuai persamaan 3.1.

3. Tentukan *pbest*

Dari hasil evaluasi fungsi objektif diperoleh nilai *fitness* sebanyak $Npop$. Selanjutnya *pbest* ditentukan jika nilai *fitness* dari sebuah partikel lebih baik (lebih kecil) dari nilai *fitness* sebelumnya maka *pbest* akan diset ke nilai *fitness* yang baru.

4. Tentukan *gbest*

Jika nilai minimum *fitness* dari seluruh partikel lebih baik (lebih kecil) dari nilai minimum *fitness* dari seluruh partikel pada iterasi sebelumnya maka nilai *gbest* akan diset sebagai nilai minimum *fitness* yang baru tersebut.

5. Perhitungan kecepatan

Perhitungan kecepatan untuk masing-masing partikel sesuai dengan persamaan berikut ini :

$$v_i(t+1) = C \left(wv_i(t) + c_1r_1(pbest_i(t) - x_i(t)) + c_2r_2(gbest(t) - x_i(t)) \right) \quad (3.6)$$

6. Perhitungan posisi

Perhitungan posisi yang baru untuk masing-masing partikel sesuai dengan persamaan berikut ini :

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (3.7)$$

7. Iterasi

Iterasi merupakan jumlah proses perulangan. Terdapat banyak cara untuk menentukan kapan iterasi tersebut berhenti diantaranya ketika PSO telah menemukan nilai optimum atau kesalahan minimum yang diinginkan, dan ketika PSO telah mencapai iterasi maksimum yang jumlahnya telah ditentukan sebelumnya. Pada penelitian ini jumlah iterasi ditentukan sesuai dengan iterasi yang dipaparkan oleh Pasandideh et al. (2011) yaitu sebanyak 400 iterasi.

8. Definisi Masalah PSO

Model dan data kajian dirumuskan kedalam *software* Matlab R2013b (Lampiran 2) yang selanjutnya dioptimasi menggunakan PSO. Pada penelitian ini *programming* untuk PSO dalam *software* Matlab R2013b mengacu pada *programming* yang telah dibuat Haupt & Haupt (2004) (Lampiran 1).

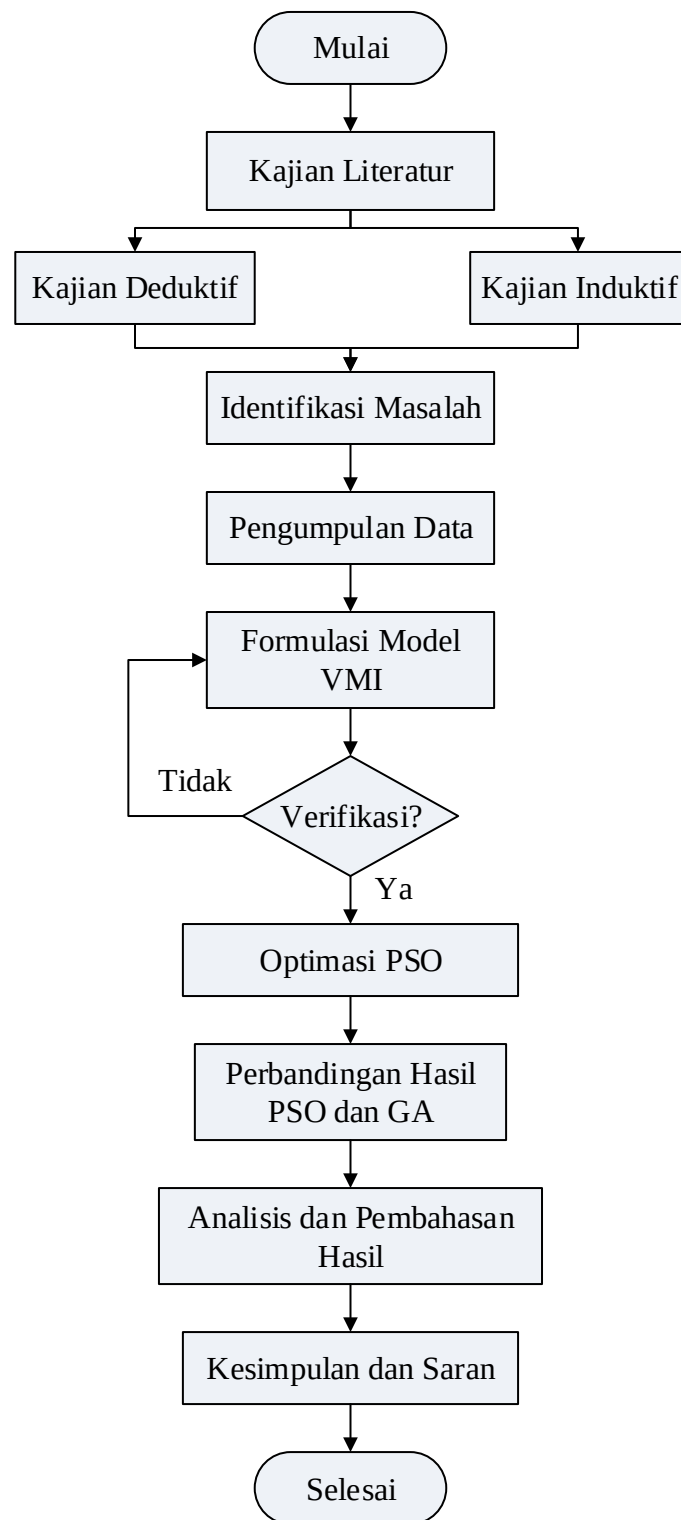
3.5 Alat Penelitian

Dalam melakukan penelitian ini digunakan alat-alat yang dapat ditunjukkan pada Tabel 3.2 sebagai berikut :

Tabel 3.2 Alat Penelitian

No	Alat	Spesifikasi
1	Laptop Dell Inspiron 1464	CPU Intel® Core™ i3, Memory RAM 2gb, OS Windows 7 Home Basic
2	Microsoft Excel	OS 2013
3	Matlab	Versi R2013b

3.6 Kerangka Penelitian



Gambar 3.2 Diagram Alir Penelitian

3.7 Penjelasan Kerangka Penelitian

Adapun penjelasan mengenai kerangka penelitian pada Gambar 3.2 diatas adalah sebagai berikut :

1. Kajian Literatur

Kajian literatur terdiri dari kajian induktif dan kajian deduktif. Kajian induktif menjelaskan tentang penelitian-penelitian terdahulu yang berkaitan dan mendukung penelitian ini. Sedangkan kajian deduktif menjelaskan tentang latar belakang teori atau metode yang digunakan pada penelitian ini.

2. Identifikasi Masalah

Masalah yang menjadi latar belakang dirumuskan menjadi pertanyaan penelitian dan menetapkan ruang lingkup kajian serta menentukan tujuan penelitian.

3. Pengumpulan Data

Data-data yang digunakan merupakan data inventori *retailer* dan *supplier*, serta data hasil optimasi GA yang telah dipaparkan pada penelitian Pasandideh et al. (2011).

4. Formulasi Model VMI

Model kajian merupakan model VMI yang dikembangkan oleh Pasandideh et al. (2011) yang selanjutnya pada penelitian ini dipotimasi menggunakan metode PSO.

5. Verifikasi

Verifikasi dilakukan dengan menghitung ulang secara manual hasil optimasi GA oleh Pasandideh et al. (2011) sehingga dapat diketahui hasil telah memenuhi batasan-batasan model atau tidak.

6. Optimasi PSO

Model VMI dioptimasi dengan metode PSO dengan tujuan meminimumkan total biaya inventori.

7. Perbandingan Hasil PSO dan GA

Total biaya inventori dari hasil optimasi PSO kemudian dibandingkan dengan hasil optimasi GA yang dilakukan Pasandideh et al. (2011) sehingga dapat diketahui metode yang lebih baik.

8. Analisis dan Pembahasan Hasil

Analisis dan pembahasan dilakukan terhadap hasil total biaya inventori yang dioptimasi dengan metode PSO serta terhadap hasil perbandingan PSO dan GA.

9. Kesimpulan dan Saran

Merupakan tahap dimana seluruh rangkaian penelitian telah selesai sehingga dapat ditarik kesimpulan berdasarkan hasil yang didapatkan. Pemberian saran juga dilakukan untuk mengembangkan hasil penelitian ini sehingga dapat menyempurnakan penelitian-penelitian selanjutnya dalam kajian permasalahan yang sama.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini menjelaskan tentang pengumpulan dan pengolahan data, analisis serta hasil yang dicapai.

4.1 Pemakaian Data

Model dan data yang digunakan pada penelitian ini menggunakan model dan data dari penelitian sebelumnya yaitu Pasandideh et al. (2011). Kemudian dilakukan verifikasi terhadap model dengan melakukan perhitungan ulang terhadap hasil yang telah dipaparkan dalam artikel tersebut. Verifikasi dilakukan dengan bantuan *software* Microsoft Excel 2013. Selanjutnya model dioptimasi dengan *Particle Swarm Optimazation* (PSO) menggunakan bantuan *software* Matlab R2013b sehingga dapat tercapai hasil dan tujuan pada penelitian ini.

4.2 Data yang Digunakan

Data-data yang digunakan pada penelitian ini merupakan data dari *supplier* dan *retailer* terhadap 10 jenis produk. Selain itu data-data hasil optimasi GA yang dilakukan Pasandideh et al. (2011) juga akan dipaparkan pada sub bab ini. Adapun data-data tersebut dapat ditunjukkan pada Tabel 4.1 sebagai berikut :

Tabel 4.1 Data Produk

No	Parameter	Simbol	Angka	Satuan
1	Jumlah produk	n	10	Bilangan bulat positif
2	Indeks produk	j	$1,2,\dots,n$	Bilangan bulat positif

4.2.1 Data Inventori *Supplier*

Adapun data inventori dari *supplier* dapat ditunjukkan pada Tabel 4.2 sebagai berikut :

Tabel 4.2 Data Inventori *Supplier*

No	Parameter	Simbol	Angka	Satuan
1	Total ruang penyimpanan yang tersedia untuk semua produk	F	18000	Meter
2	Total jumlah pesanan untuk semua produk	M	12	Kali
3	Biaya tetap <i>backorder</i> per unit	II	0	\$/unit
4	Biaya tetap <i>backorder</i> per unit per waktu	$\hat{\pi}$	3	\$/unit

Selain data-data diatas terdapat data biaya tetap pemesanan *supplier* (A_{js}) dan data ukuran ruangan yang dapat ditempati oleh masing-masing produk (f_j). Adapun data-data tersebut dapat ditunjukkan pada Tabel 4.3 sebagai berikut :

Tabel 4.3 Data A_{js} dan f_j

Produk (j)	A_{js} (\$/unit)	f_j (Meter)
1	1	3
2	2	2
3	3	3
4	5	1
5	2	4
6	4	3
7	1	2

Produk (j)	A_{js} (\$/unit)	f_j (Meter)
8	2	1
9	3	3
10	4	4

4.2.2 Data Inventori *Retailer*

Adapun yang termasuk dalam data inventori *retailer* adalah data jumlah permintaan semua produk dari pelanggan (D_j) yang bersifat deterministik, data biaya tetap pemesanan yang dilakukan *retailer* (A_{jR}) untuk masing-masing produk serta data biaya penyimpanan (h_{jR}) pada *retailer* untuk masing-masing produk. Adapun data-data tersebut dapat ditunjukkan pada Tabel 4.4 sebagai berikut :

Tabel 4.4 Data D_j , A_{jR} dan h_{jR}			
Produk (j)	D_j (Unit)	A_{jR} (\$/unit)	h_{jR} (\$/unit)
1	420	3	4
2	360	2	9
3	540	1	7
4	390	4	2
5	480	2	4
6	510	2	6
7	530	3	5
8	380	1	3
9	430	4	2
10	580	2	8

4.2.3 Data Hasil Optimasi GA

Adapun jumlah Q_j , b_j dan TC_{VMI} yang optimum berdasarkan hasil optimasi menggunakan metode GA yang dilakukan oleh Pasandideh et al. (2011) dapat ditunjukkan pada Tabel 4.5 dibawah ini :

Tabel 4.5 Data Hasil Optimasi GA			
n	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
3	469, 158, 231	182, 99, 185	925.73
5	147, 236, 206, 356, 147	76, 170, 76, 76, 40	1191.06
8	368, 82, 444, 368, 368, 473, 784, 363	378, 69, 285, 100, 171, 218, 506, 72	3443.9
10	774, 737, 400, 1257, 177, 583, 626, 199, 1005, 510	300, 473, 399, 300, 300, 286, 292, 85, 430, 399	6686.32

4.3 Verifikasi Data Hasil Optimasi GA

Pada sub bab ini dilakukan verifikasi data berdasarkan hasil optimasi GA. Verifikasi data dilakukan dengan menghitung secara manual yang akan dibagi menjadi 2 yaitu pertama adalah menghitung jumlah Q terhadap kendala-kendala (F dan M) agar dapat diketahui apakah jumlah tersebut memenuhi batasan model atau tidak. Kemudian yang kedua adalah melakukan perhitungan secara manual terhadap hasil total biaya inventori (TC_{VMI}). Verifikasi jumlah Q dan TC_{VMI} dilakukan pada 3, 5, 8 dan 10 jenis produk. Adapun proses perhitungannya dapat dilihat sebagai berikut :

4.3.1 Verifikasi Kendala

Verifikasi kendala dilakukan terhadap 2 kendala yaitu kapasitas penyimpanan (F) sebesar 18000 dan jumlah pesanan (M) sebanyak 12 kali. Verifikasi kendala ini dengan melakukan perhitungan secara manual untuk 3, 5, 8 dan 10 jenis produk (n). Hasil perhitungan dapat ditunjukkan pada Tabel 4.6 sebagai berikut :

Tabel 4.6 Perhitungan Verifikasi Kendala untuk Hasil Optimasi GA			
n	$\sum_{j=1}^n f_j q_j \leq 18000$	$\sum_{j=1}^n \frac{D_j}{Q_j} \leq 12$	Keterangan
3	$(3 \times 469) + (2 \times 158) +$ $(3 \times 231) = \mathbf{2416}$	$(420/469) + (360/158) +$ $(540/231) = \mathbf{5,51}$	Memenuhi batasan
5	$(3 \times 147) + (2 \times 236) +$ $(3 \times 206) + (1 \times 356) +$ $(4 \times 147) = \mathbf{2475}$	$(420/147) + (360/236) +$ $(540/206) + (390/356) +$ $(480/147) = \mathbf{11,36}$	Memenuhi batasan
8	$(3 \times 368) + (2 \times 82) +$ $(3 \times 444) + (1 \times 368) +$ $(4 \times 368) + (3 \times 473) +$ $(2 \times 784) + (1 \times 363) = \mathbf{7790}$	$(420/368) + (360/82) +$ $(540/444) + (390/368) +$ $(480/368) + (510/473) +$ $(530/784) + (380/363) = \mathbf{11,91}$	Memenuhi batasan
10	$(3 \times 774) + (2 \times 737) +$ $(3 \times 400) + (1 \times 1257) +$ $(4 \times 177) + (3 \times 583) +$ $(2 \times 626) + (1 \times 199) +$ $(3 \times 1005) + (4 \times 510) =$ $\mathbf{15216}$	$(420/774) + (360/737) +$ $(540/400) + (390/1257) +$ $(480/177) + (510/583) +$ $(530/626) + (380/199) +$ $(430/1005) + (580/510) =$ $\mathbf{10,59}$	Memenuhi batasan

Dari Tabel 4.6 diatas menunjukkan bahwa untuk 3, 5, 8, dan 10 jenis produk telah memenuhi batasan. Sehingga total biaya hasil optimasi GA selanjutnya dapat dibandingkan dengan hasil optimasi PSO.

4.3.2 Verifikasi Total Biaya

Verifikasi total biaya dilakukan secara perhitungan manual yang kemudian hasilnya dibandingkan dengan hasil total biaya yang telah dipaparkan Pasandideh et al. (2011). Verifikasi dilakukan untuk 3, 5, 8 dan 10 jenis produk (n). Adapun hasil perhitungan dapat ditunjukkan sebagai berikut :

$$TC_{VMI} = \sum_{j=1}^n \left(\frac{D_j}{Q_j(A_{jS} + A_{jR})} + \frac{h_{jR}}{2Q_j} (Q_j - b_j)^2 + \frac{\hat{\pi} b_j^2}{2Q_j} + \frac{\pi b_j D_j}{Q_j} \right) \quad (4.1)$$

1. Untuk $n = 3$

$$\begin{aligned} TC_{VMI} &= \left(\frac{420}{469(1+3)} + \frac{4}{2 \times 469} (469 - 182)^2 + \frac{3 \times 182^2}{2 \times 469} + \frac{0 \times 182 \times 420}{469} \right) \\ &\quad + \left(\frac{360}{158(2+2)} + \frac{9}{2 \times 158} (158 - 99)^2 + \frac{3 \times 99^2}{2 \times 158} + \frac{0 \times 99 \times 360}{158} \right) \\ &\quad + \left(\frac{540}{231(3+1)} + \frac{7}{2 \times 231} (231 - 185)^2 + \frac{3 \times 185^2}{2 \times 231} + \frac{0 \times 185 \times 540}{231} \right) \\ &= 925,7314 \text{ (Sesuai)} \end{aligned}$$

2. Untuk $n = 5$

$$\begin{aligned} TC_{VMI} &= \left(\frac{420}{147(1+3)} + \frac{4}{2 \times 147} (147 - 76)^2 + \frac{3 \times 76^2}{2 \times 147} + \frac{0 \times 76 \times 420}{147} \right) \\ &\quad + \left(\frac{360}{236(2+2)} + \frac{9}{2 \times 236} (236 - 170)^2 + \frac{3 \times 170^2}{2 \times 236} + \frac{0 \times 170 \times 360}{236} \right) \\ &\quad + \left(\frac{540}{206(3+1)} + \frac{7}{2 \times 206} (206 - 76)^2 + \frac{3 \times 76^2}{2 \times 206} + \frac{0 \times 76 \times 540}{206} \right) \end{aligned}$$

$$\begin{aligned}
& + \left(\frac{390}{356(5+4)} + \frac{2}{2 \times 356} (356 - 76)^2 + \frac{3 \times 76^2}{2 \times 356} + \frac{0 \times 76 \times 390}{356} \right) \\
& + \left(\frac{480}{147(2+2)} + \frac{4}{2 \times 147} (147 - 40)^2 + \frac{3 \times 40^2}{2 \times 147} + \frac{0 \times 40 \times 480}{147} \right) \\
& = 1191,057 \text{ (Sesuai)}
\end{aligned}$$

3. Untuk $n = 8$

$$\begin{aligned}
TC_{VMI} &= \left(\frac{420}{368(1+3)} + \frac{4}{2 \times 368} (368 - 378)^2 + \frac{3 \times 378^2}{2 \times 368} + \frac{0 \times 378 \times 420}{368} \right) \\
&+ \left(\frac{360}{82(2+2)} + \frac{9}{2 \times 82} (82 - 69)^2 + \frac{3 \times 69^2}{2 \times 82} + \frac{0 \times 69 \times 360}{82} \right) \\
&+ \left(\frac{540}{444(3+1)} + \frac{7}{2 \times 444} (444 - 285)^2 + \frac{3 \times 285^2}{2 \times 444} + \frac{0 \times 285 \times 540}{444} \right) \\
&+ \left(\frac{390}{368(5+4)} + \frac{2}{2 \times 368} (368 - 100)^2 + \frac{3 \times 100^2}{2 \times 368} + \frac{0 \times 100 \times 390}{368} \right) \\
&+ \left(\frac{480}{368(2+2)} + \frac{4}{2 \times 368} (368 - 171)^2 + \frac{3 \times 171^2}{2 \times 368} + \frac{0 \times 171 \times 480}{368} \right) \\
&+ \left(\frac{510}{473(4+2)} + \frac{6}{2 \times 473} (473 - 218)^2 + \frac{3 \times 218^2}{2 \times 473} + \frac{0 \times 218 \times 510}{473} \right) \\
&+ \left(\frac{530}{784(1+3)} + \frac{5}{2 \times 784} (784 - 506)^2 + \frac{3 \times 506^2}{2 \times 784} + \frac{0 \times 506 \times 530}{784} \right) \\
&+ \left(\frac{380}{363(2+1)} + \frac{3}{2 \times 363} (363 - 72)^2 + \frac{3 \times 72^2}{2 \times 363} + \frac{0 \times 72 \times 380}{363} \right) \\
&= 3443,89 \text{ (Sesuai)}
\end{aligned}$$

4. Untuk $n = 10$

$$\begin{aligned}
TC_{VMI} = & \left(\frac{420}{774(1+3)} + \frac{4}{2 \times 774} (774 - 300)^2 + \frac{3 \times 300^2}{2 \times 774} + \frac{0 \times 300 \times 420}{774} \right) \\
& + \left(\frac{360}{737(2+2)} + \frac{9}{2 \times 737} (737 - 473)^2 + \frac{3 \times 473^2}{2 \times 737} + \frac{0 \times 473 \times 360}{737} \right) \\
& + \left(\frac{540}{400(3+1)} + \frac{7}{2 \times 400} (400 - 399)^2 + \frac{3 \times 399^2}{2 \times 400} + \frac{0 \times 399 \times 540}{400} \right) \\
& + \left(\frac{390}{1257(5+4)} + \frac{2}{2 \times 1257} (1257 - 300)^2 + \frac{3 \times 300^2}{2 \times 1257} + \frac{0 \times 300 \times 390}{1257} \right) \\
& + \left(\frac{480}{177(2+2)} + \frac{4}{2 \times 177} (177 - 300)^2 + \frac{3 \times 300^2}{2 \times 177} + \frac{0 \times 300 \times 480}{177} \right) \\
& + \left(\frac{510}{583(4+2)} + \frac{6}{2 \times 583} (583 - 286)^2 + \frac{3 \times 286^2}{2 \times 583} + \frac{0 \times 286 \times 510}{583} \right) \\
& + \left(\frac{530}{626(1+3)} + \frac{5}{2 \times 626} (626 - 292)^2 + \frac{3 \times 292^2}{2 \times 626} + \frac{0 \times 292 \times 530}{626} \right) \\
& + \left(\frac{380}{199(2+1)} + \frac{3}{2 \times 199} (199 - 85)^2 + \frac{3 \times 85^2}{2 \times 199} + \frac{0 \times 85 \times 380}{199} \right) \\
& + \left(\frac{430}{1005(3+4)} + \frac{2}{2 \times 1005} (1005 - 430)^2 + \frac{3 \times 430^2}{2 \times 1005} + \frac{0 \times 430 \times 430}{1005} \right) \\
& + \left(\frac{580}{510(4+2)} + \frac{8}{2 \times 510} (510 - 399)^2 + \frac{3 \times 399^2}{2 \times 510} + \frac{0 \times 399 \times 580}{510} \right) \\
= & 6686,32 \text{ (Sesuai)}
\end{aligned}$$

Berdasarkan hasil perhitungan manual dengan menggunakan persamaan 4.1 untuk verifikasi kendala dan total biaya, hasilnya menunjukkan bahwa data-data hasil optimasi dengan GA yang dilakukan Pasandideh et al. (2011) sudah memenuhi batasan

model dan total biaya sudah sesuai dengan hasil perhitungan yang telah dilakukan. Sehingga untuk Q_j , b_j , dan TC_{VMI} pada 3, 5, 8, dan 10 jenis produk dapat dibandingkan dengan hasil optimasi menggunakan metode PSO.

4.4 Optimasi menggunakan *Particle Swarm Optimization*

Optimasi menggunakan PSO dilakukan untuk mencari Q_j dan b_j yang optimum pada 3, 5, 8, dan 10 jenis produk (n) sehingga didapatkan hasil TC_{VMI} yang minimum. Dalam percobaan yang dilakukan parameter yang digunakan adalah $Npop$, iterasi, c_1 dan c_2 . Untuk $Npop$ dan iterasi ditentukan sesuai dengan yang dipaparkan oleh Pasandideh et al. (2011). Sedangkan untuk nilai c_1 dan c_2 yang terbaik didapatkan berdasarkan percobaan pada $n = 3$. Sehingga pada percobaan dengan n sebesar 5, 8 dan 10 menggunakan nilai c_1 dan c_2 yang sudah optimal. Keseluruhan percobaan dipilih menjadi yang terbaik berdasarkan nilai TC_{VMI} yang paling minimum. Berikut adalah percobaan-percobaan yang telah dilakukan :

4.4.1 Optimasi pada $n = 3$

Pada percobaan $n = 3$ ini dilakukan pencarian c_1 dan c_2 yang optimum pada $Npop$ sebesar 80, 100, 120, 140, 160, dan 180 serta jumlah iterasi sebanyak 400 kali. Adapun hasil percobaan yang telah dilakukan dapat ditunjukkan pada Tabel 4.7 sebagai berikut:

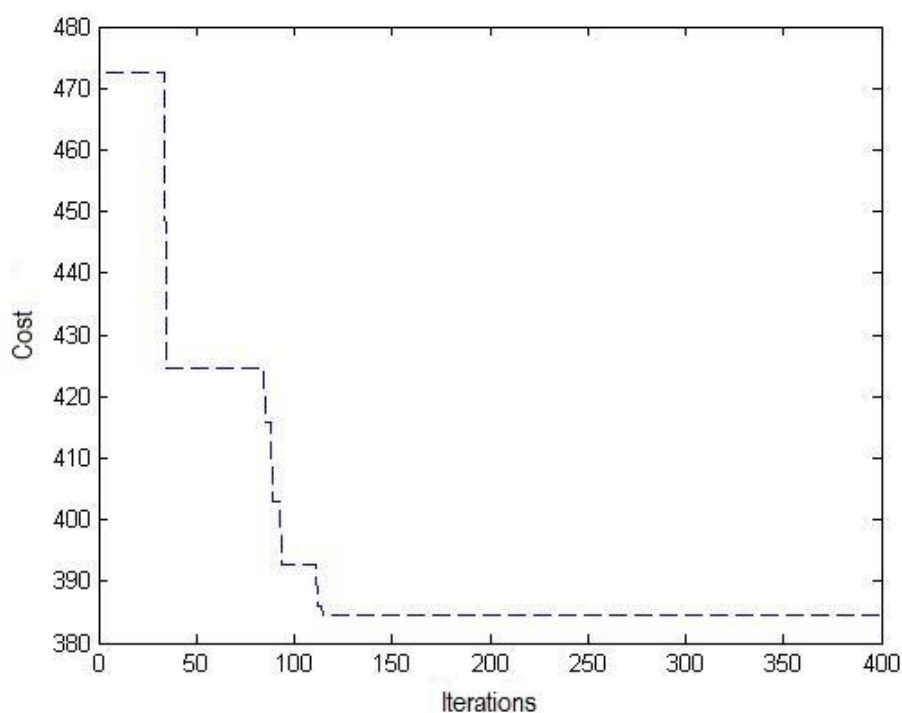
Tabel 4.7 Hasil Percobaan Optimasi PSO ($n = 3$)					
$Npop$	c_1	c_2	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
80	2	2	95, 105, 131	30, 64, 81	423.52
	2.05	2.05	110, 91, 128	54, 38, 62	471.62

<i>Npop</i>	<i>c</i> ₁	<i>c</i> ₂	<i>Q_j</i> (Unit)	<i>b_j</i> (Unit)	<i>TC_{VMI}</i> (\$)
	2	2.1	109, 169, 90	51, 100, 75	463.48
	2.1	2	172, 81, 106	89, 67, 77	402.79
	1.9	2.1	156, 117, 113	100, 100, 92	443.88
	1.9	2.2	176, 189, 150	100, 96, 66	669.7
	1.8	2.2	169, 194, 71	100, 100, 50	549.72
	2.1	1.9	200, 75, 106	100, 57, 86	425.27
	2.2	1.9	193, 61, 138	70, 16, 83	550.11
	2.2	1.8	67, 119, 200	34, 100, 100	496.07
100	2	2	63, 140, 200	15, 100, 100	534.82
	2.05	2.05	66, 153, 165	10, 83, 48	668.24
	2	2.1	103, 166, 113	75, 91, 89	491.54
	2.1	2	172, 89, 112	39, 85, 78	504.37
	1.9	2.1	135, 144, 162	100, 100, 100	505.35
	1.9	2.2	176, 98, 142	100, 70, 98	450.48
	1.8	2.2	108, 200, 86	62, 100, 62	530.93
	2.1	1.9	160, 54, 200	100, 17, 100	558.83
	2.2	1.9	112, 104, 200	68, 88, 60	628.92
	2.2	1.8	200, 65, 124	100, 53, 100	434.99
120	2	2	200, 68, 118	100, 57, 42	496.3
	2.05	2.05	95, 128, 114	50, 65, 76	439.37
	2	2.1	152, 113, 111	52, 94, 91	457.7
	2.1	2	181, 64, 200	100, 53, 96	530.7
	1.9	2.1	200, 117, 105	100, 100, 74	465.86
	1.9	2.2	200, 117, 173	59, 100, 100	591.95
	1.8	2.2	200, 50, 200	100, 38, 100	529.28
	2.1	1.9	127, 88, 118	75, 72, 90	384.59
	2.2	1.9	200, 117, 136	96, 100, 95	495.99
	2.2	1.8	89, 129, 122	10, 100, 100	472.13
140	2	2	183, 120, 81	100, 100, 38	451.74
	2.05	2.05	178, 110, 85	97, 75, 37	446.8
	2	2.1	157, 68, 190	91, 67, 72	574.36

<i>Npop</i>	<i>c</i> ₁	<i>c</i> ₂	<i>Q_j</i> (Unit)	<i>b_j</i> (Unit)	<i>TC_{VMI}</i> (\$)
	2.1	2	114, 86, 200	55, 83, 73	586.01
	1.9	2.1	151, 133, 83	61, 100, 61	429.51
	1.9	2.2	180, 81, 161	45, 50, 55	632.03
	1.8	2.2	136, 82, 131	72, 57, 60	433.45
	2.1	1.9	140, 200, 78	85, 100, 55	549.43
	2.2	1.9	103, 74, 177	50, 46, 71	494.39
	2.2	1.8	84, 164, 113	50, 100, 74	443.72
160	2	2	193, 82, 100	100, 61, 77	414.93
	2.05	2.05	152, 67, 140	66, 40, 58	477.2
	2	2.1	200, 68, 118	97, 37, 70	448.96
	2.1	2	200, 144, 123	100, 100, 100	512.63
	1.9	2.1	131, 133, 89	88, 58, 65	486.94
	1.9	2.2	190, 189, 98	100, 88, 69	609.96
	1.8	2.2	110, 104, 115	74, 89, 89	394.01
	2.1	1.9	135, 133, 91	100, 100, 75	428.46
	2.2	1.9	123, 91, 119	35, 48, 53	481.43
	2.2	1.8	79, 125, 142	10, 100, 100	462.04
180	2	2	200, 120, 90	100, 100, 72	458.4
	2.05	2.05	152, 113, 158	38, 60, 95	555.97
	2	2.1	105, 90, 136	52, 75, 81	395.23
	2.1	2	116, 117, 200	89, 100, 80	591.91
	1.9	2.1	200, 66, 122	81, 49, 92	442.9
	1.9	2.2	84, 102, 174	31, 87, 70	511.61
	1.8	2.2	72, 113, 200	50, 100, 100	501.88
	2.1	1.9	189, 52, 197	100, 36, 100	513.58
	2.2	1.9	81, 111, 158	46, 100, 100	426.31
	2.2	1.8	166, 117, 144	99, 100, 100	470.62

Keterangan : Warna Abu-abu = Percobaan terbaik

Pemilihan percobaan terbaik berdasarkan nilai total biaya inventori (TC_{VMI}) yang terendah. Dari hasil percobaan pada Tabel 4.7 diatas dapat diketahui nilai parameter c_1 dan c_2 yang optimal untuk masing-masing populasi. Parameter c_1 dan c_2 yang optimal ini selanjutnya digunakan pada percobaan 5, 8, dan 10 jenis produk. Dari percobaan terbaik untuk tiap populasi kemudian dipilih hasil percobaan yang terbaik untuk 3 jenis produk yaitu pada percobaan dengan populasi = 120, $c_1 = 2.1$, dan $c_2 = 1.9$ yang menghasilkan TC_{VMI} sebesar \$ 384.59. Proses optimasi PSO hingga mencapai konvergen untuk hasil TC_{VMI} yang terbaik pada 3 jenis produk dapat ditunjukkan pada Gambar 4.1 dan untuk hasil grafik konvergen dari semua percobaan untuk 3 jenis produk dapat ditunjukkan pada Lampiran 3.



Gambar 4.1 Grafik Konvergen untuk $n = 3$

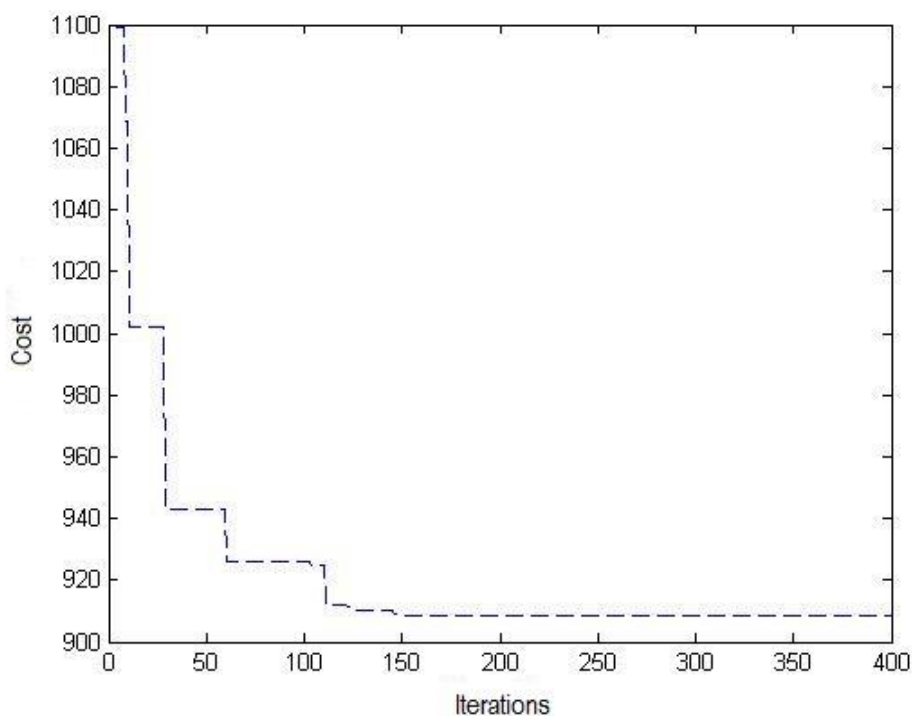
4.4.2 Optimasi pada $n = 5$

Pada percobaan $n = 5$ ini dilakukan menggunakan parameter c_1 dan c_2 yang optimum pada masing-masing populasi berdasarkan percobaan sebelumnya. Adapun hasil percobaan yang telah dilakukan dapat ditunjukkan pada Tabel 4.8 sebagai berikut :

Tabel 4.8 Hasil Percobaan Optimasi PSO ($n = 5$)					
<i>Npop</i>	c_1	c_2	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
80	2.1	2	170, 159, 206, 166, 225	109, 112, 66, 61, 134	1046.65
100	2.2	1.8	249, 151, 250, 250, 115	150, 125, 150, 150, 71	994.71
120	2.1	1.9	134, 154, 195, 217, 250	50, 129, 150, 50, 125	944.44
140	1.9	2.1	250, 100, 212, 250, 184	78, 50, 150, 100, 109	1009.62
160	1.8	2.2	168, 161, 150, 224, 250	114, 131, 107, 118, 150	908.52
180	2	2.1	169, 196, 152, 241, 214	140, 148, 107, 70, 127	954.72

Keterangan : Warna Abu-abu = Percobaan terbaik

Percobaan untuk 5 jenis produk yang terbaik adalah pada percobaan dengan parameter populasi 160, $c_1 = 1.8$, dan $c_2 = 2.2$. Pada percobaan terbaik ini dihasilkan nilai TC_{VMI} sebesar \$ 908.52. Adapun proses optimasi PSO hingga mencapai konvergen untuk percobaan terbaik ini dapat ditunjukkan pada Gambar 4.2 dan untuk hasil grafik konvergen dari semua percobaan untuk 5 jenis produk dapat ditunjukkan pada Lampiran 4.



Gambar 4.2 Grafik Konvergen untuk $n = 5$

4.4.3 Optimasi pada $n = 8$

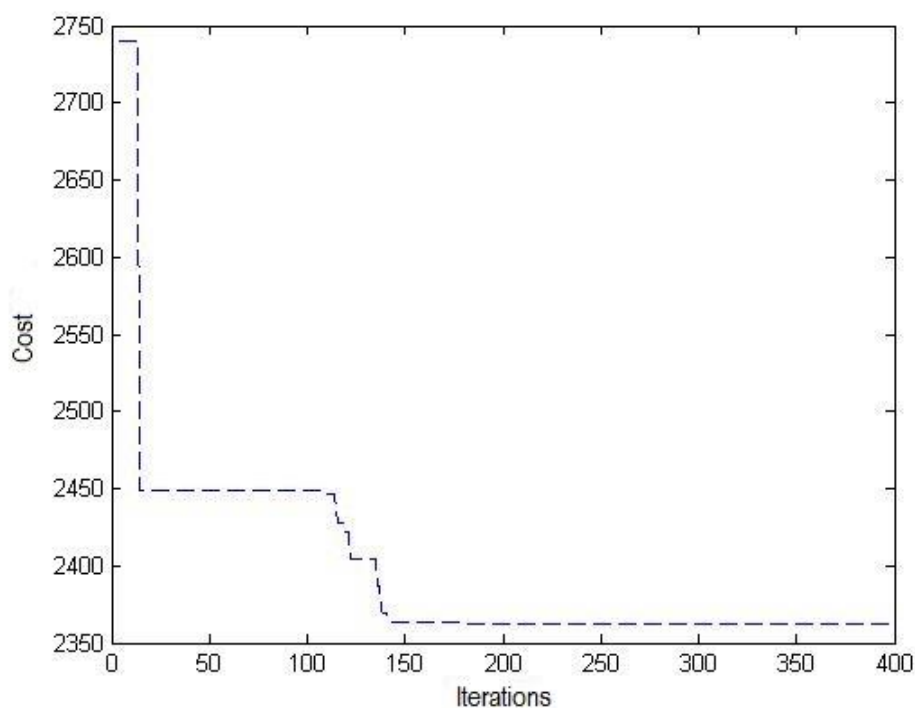
Pada percobaan $n = 8$ ini dilakukan menggunakan parameter c_1 dan c_2 yang optimum pada masing-masing populasi berdasarkan percobaan sebelumnya. Adapun hasil percobaan yang telah dilakukan dapat ditunjukkan pada Tabel 4.9 sebagai berikut :

N_{pop}	c_1	c_2	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
80	2.1	2	301, 297, 320, 349, 333, 257, 400, 272	200, 157, 217, 109, 104, 217, 147, 176	2655.78
100	2.2	1.8	286, 286, 279, 259, 400, 316, 357, 299	168, 242, 190, 149, 250, 207, 177, 121	2362.19
120	2.1	1.9	265, 314, 387, 200, 342, 316, 324, 300	132, 229, 250, 134, 246, 193, 219, 100	2411.2

N_{pop}	c_1	c_2	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
140	1.9	2.1	400, 283, 400, 325, 400, 225, 400, 200	240, 250, 250, 105, 250, 150, 250, 100	2475.41
160	1.8	2.2	400, 200, 272, 400, 397, 200, 400, 400	100, 125, 243, 213, 238, 135, 229, 200	2602.52
180	2	2.1	297, 310, 221, 372, 325, 319, 304, 342	154, 250, 162, 149, 103, 225, 121, 172	2401.48

Keterangan : Warna Abu-abu = Percobaan terbaik

Percobaan untuk 8 jenis produk yang terbaik adalah pada percobaan dengan parameter populasi 100, $c_1 = 2.2$, dan $c_2 = 1.8$. Pada percobaan terbaik ini dihasilkan nilai TC_{VMI} sebesar \$ 2362.19. Adapun proses optimasi PSO hingga mencapai konvergen untuk percobaan terbaik ini dapat ditunjukkan pada Gambar 4.3 dan untuk hasil grafik konvergen dari semua percobaan untuk 8 jenis produk dapat ditunjukkan pada Lampiran 5.



Gambar 4.3 Grafik Konvergen untuk $n = 8$

4.4.4 Optimasi pada $n = 10$

Pada percobaan $n = 10$ ini dilakukan menggunakan parameter c_1 dan c_2 yang optimum pada masing-masing populasi berdasarkan percobaan sebelumnya. Adapun hasil percobaan yang telah dilakukan dapat ditunjukkan pada Tabel 4.10 sebagai berikut :

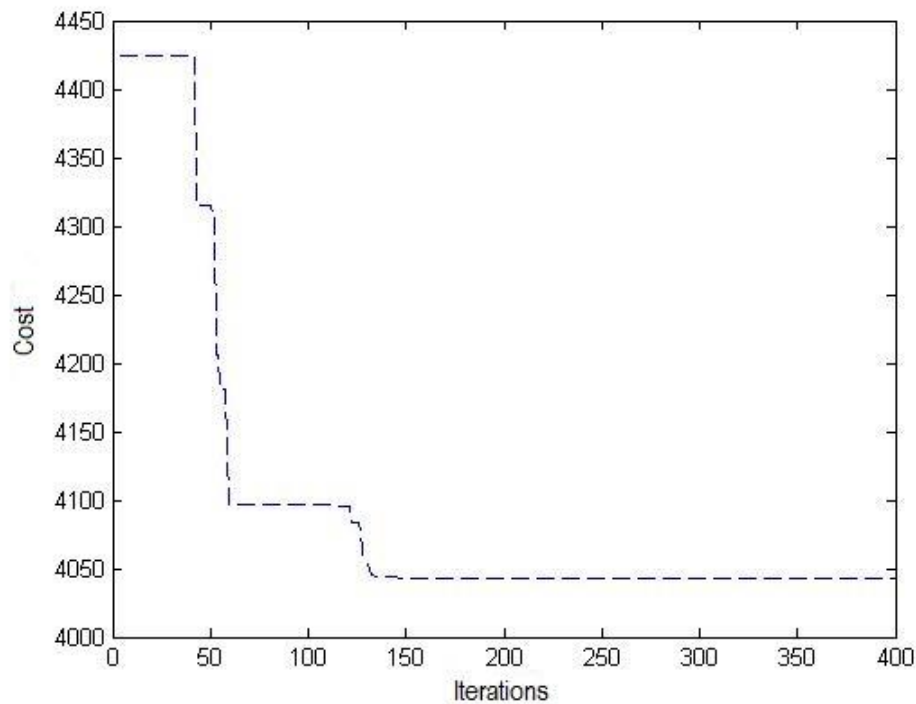
Tabel 4.10 Hasil Percobaan Optimasi PSO ($n = 10$)

N_{pop}	c_1	c_2	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
80	2.1	2	521, 270, 439, 259, 600, 600, 300, 600, 260, 411	207, 157, 241, 150, 150, 194, 200, 250, 150, 242	4666.55
100	2.2	1.8	595, 600, 338, 397, 367, 250, 298, 600, 380, 481	250, 250, 221, 150, 221, 150, 204, 224, 250, 249	4557.34
120	2.1	1.9	425, 305, 284, 559, 360, 514, 279, 516, 477, 424	204, 250, 150, 198, 163, 245, 185, 186, 165, 178	4042.18
140	1.9	2.1	507, 268, 264, 475, 373, 390, 540, 275, 468, 536	150, 250, 150, 216, 249, 241, 250, 150, 191, 236	4176.77
160	1.8	2.2	448, 250, 434, 539, 586, 359, 455, 564, 312, 264	245, 187, 237, 186, 165, 247, 179, 150, 205, 157	4189.55
180	2	2.1	591, 356, 263, 484, 360, 348, 328, 328, 497, 592	245, 250, 176, 237, 168, 175, 205, 201, 158, 187	4380.62

Keterangan : Warna Abu-abu = Percobaan terbaik

Percobaan untuk 10 jenis produk yang terbaik adalah pada percobaan dengan parameter populasi 120, $c_1 = 2.1$, dan $c_2 = 1.9$. Pada percobaan terbaik ini dihasilkan

nilai TC_{VMI} sebesar \$ 4042.18. Adapun proses optimasi PSO hingga mencapai konvergen untuk percobaan terbaik ini dapat ditunjukkan pada Gambar 4.4 dan untuk hasil grafik konvergen dari semua percobaan untuk 10 jenis produk dapat ditunjukkan pada Lampiran 6.



Gambar 4.4 Grafik Konvergen untuk $n = 10$

4.5 Hasil Optimasi PSO

Optimasi menggunakan PSO dilakukan untuk mengoptimalkan jumlah Q dan b sehingga didapatkan total biaya inventori yang minimum. Berdasarkan percobaan yang telah dilakukan, hasil optimasi PSO untuk 3, 5, 8, dan 10 jenis produk dapat ditunjukkan pada Tabel 4.11 sebagai berikut :

Tabel 4.11 Hasil Optimasi PSO

n	Q_j (Unit)	b_j (Unit)	TC_{VMI} (\$)
3	127, 88, 118	75, 72, 90	384.59
5	168, 161, 150, 224, 250	114, 131, 107, 118, 150	908.52
8	286, 286, 279, 259, 400, 316, 357, 299	168, 242, 190, 149, 250, 207, 177,121	2362.19
10	425, 305, 284, 559, 360, 514, 279, 516, 477, 424	204, 250, 150, 198, 163, 245, 185, 186, 165, 178	4042.18

4.6 Verifikasi Hasil Optimasi PSO

Pada sub bab ini dilakukan verifikasi berdasarkan hasil optimasi PSO yang telah dilakukan. Verifikasi dilakukan dengan menghitung secara manual yang akan dibagi menjadi 2 yaitu pertama adalah menghitung jumlah Q terhadap kendala-kendala (F dan M) agar dapat diketahui apakah jumlah tersebut memenuhi batasan model atau tidak. Kemudian yang kedua adalah melakukan perhitungan secara manual terhadap hasil total biaya inventori (TC_{VMI}). Verifikasi jumlah Q dan TC_{VMI} dilakukan pada 3, 5, 8 dan 10 jenis produk. Adapun proses perhitungannya dapat dilihat sebagai berikut :

4.6.1 Verifikasi Kendala

Verifikasi kendala dilakukan terhadap 2 kendala yaitu kapasitas penyimpanan (F) sebesar 18000 dan jumlah pesanan (M) sebanyak 12 kali. Verifikasi kendala ini dengan melakukan perhitungan secara manual untuk 3, 5, 8 dan 10 jenis produk (n). Hasil perhitungan dapat ditunjukkan pada Tabel 4.12 sebagai berikut :

Tabel 4.12 Perhitungan Verifikasi Kendala untuk Hasil Optimasi PSO

N	$\sum_{j=1}^n f_j q_j \leq 18000$	$\sum_{j=1}^n \frac{D_j}{Q_j} \leq 12$	Keterangan
3	$(3 \times 127) + (2 \times 88) + (3 \times 118) = \mathbf{911}$	$(420/127) + (360/88) + (540/118) = \mathbf{11,97}$	Memenuhi batasan
5	$(3 \times 168) + (2 \times 161) + (3 \times 150) + (1 \times 224) + (4 \times 250) = \mathbf{2500}$	$(420/168) + (360/161) + (540/150) + (390/224) + (480/250) = \mathbf{11,99}$	Memenuhi batasan
8	$(3 \times 286) + (2 \times 286) + (3 \times 279) + (1 \times 259) + (4 \times 400) + (3 \times 316) + (2 \times 357) + (1 \times 299) = \mathbf{6087}$	$(420/286) + (360/286) + (540/279) + (390/259) + (480/400) + (510/316) + (530/357) + (380/299) = \mathbf{11,73}$	Memenuhi batasan
10	$(3 \times 425) + (2 \times 305) + (3 \times 284) + (1 \times 559) + (4 \times 360) + (3 \times 514) + (2 \times 279) + (1 \times 516) + (3 \times 477) + (4 \times 424) = \mathbf{10479}$	$(420/425) + (360/305) + (540/284) + (390/559) + (480/360) + (510/514) + (530/279) + (380/516) + (430/477) + (580/424) = \mathbf{11,99}$	Memenuhi batasan

Dari Tabel 4.12 diatas menunjukkan bahwa untuk 3, 5, 8, dan 10 jenis produk telah memenuhi batasan. Sehingga total biaya hasil optimasi PSO selanjutnya dapat dibandingkan dengan hasil optimasi GA.

4.6.2 Verifikasi Total Biaya

Verifikasi total biaya dilakukan secara perhitungan manual yang kemudian hasilnya dibandingkan dengan hasil total biaya yang dioptimasi menggunakan PSO. Verifikasi dilakukan untuk 3, 5, 8 dan 10 jenis produk (n). Adapun hasil perhitungan dengan menggunakan persamaan 4.1 dapat ditunjukkan sebagai berikut :

1. Untuk $n = 3$

$$\begin{aligned}
 TC_{VMI} &= \left(\frac{420}{127(1+3)} + \frac{4}{2 \times 127} (127 - 75)^2 + \frac{3 \times 75^2}{2 \times 127} + \frac{0 \times 75 \times 420}{127} \right) \\
 &+ \left(\frac{360}{88(2+2)} + \frac{9}{2 \times 88} (88 - 72)^2 + \frac{3 \times 72^2}{2 \times 88} + \frac{0 \times 72 \times 360}{88} \right) \\
 &+ \left(\frac{540}{118(3+1)} + \frac{7}{2 \times 118} (118 - 90)^2 + \frac{3 \times 90^2}{2 \times 118} + \frac{0 \times 90 \times 540}{118} \right) \\
 &= 384,59164 \text{ (Sesuai)}
 \end{aligned}$$

2. Untuk $n = 5$

$$\begin{aligned}
 TC_{VMI} &= \left(\frac{420}{168(1+3)} + \frac{4}{2 \times 168} (168 - 114)^2 + \frac{3 \times 114^2}{2 \times 168} + \frac{0 \times 114 \times 420}{168} \right) \\
 &+ \left(\frac{360}{161(2+2)} + \frac{9}{2 \times 161} (161 - 131)^2 + \frac{3 \times 131^2}{2 \times 161} + \frac{0 \times 131 \times 360}{161} \right) \\
 &+ \left(\frac{540}{150(3+1)} + \frac{7}{2 \times 150} (150 - 107)^2 + \frac{3 \times 107^2}{2 \times 150} + \frac{0 \times 107 \times 540}{150} \right) \\
 &+ \left(\frac{390}{224(5+4)} + \frac{2}{2 \times 224} (224 - 118)^2 + \frac{3 \times 118^2}{2 \times 224} + \frac{0 \times 118 \times 390}{224} \right) \\
 &+ \left(\frac{480}{250(2+2)} + \frac{4}{2 \times 250} (250 - 150)^2 + \frac{3 \times 150^2}{2 \times 250} + \frac{0 \times 150 \times 480}{250} \right) \\
 &= 908,519234 \text{ (Sesuai)}
 \end{aligned}$$

3. Untuk $n = 8$

$$TC_{VMI} = \left(\frac{420}{286(1+3)} + \frac{4}{2 \times 286} (286 - 168)^2 + \frac{3 \times 168^2}{2 \times 286} + \frac{0 \times 168 \times 420}{286} \right)$$

$$\begin{aligned}
& + \left(\frac{360}{286(2+2)} + \frac{9}{2 \times 286} (286 - 242)^2 + \frac{3 \times 242^2}{2 \times 286} + \frac{0 \times 242 \times 360}{286} \right) \\
& + \left(\frac{540}{279(3+1)} + \frac{7}{2 \times 279} (279 - 190)^2 + \frac{3 \times 190^2}{2 \times 279} + \frac{0 \times 190 \times 540}{279} \right) \\
& + \left(\frac{390}{259(5+4)} + \frac{2}{2 \times 259} (259 - 149)^2 + \frac{3 \times 149^2}{2 \times 259} + \frac{0 \times 149 \times 390}{259} \right) \\
& + \left(\frac{480}{400(2+2)} + \frac{4}{2 \times 400} (400 - 250)^2 + \frac{3 \times 250^2}{2 \times 400} + \frac{0 \times 250 \times 480}{400} \right) \\
& + \left(\frac{510}{316(4+2)} + \frac{6}{2 \times 316} (316 - 207)^2 + \frac{3 \times 207^2}{2 \times 316} + \frac{0 \times 207 \times 510}{316} \right) \\
& + \left(\frac{530}{357(1+3)} + \frac{5}{2 \times 357} (357 - 177)^2 + \frac{3 \times 177^2}{2 \times 357} + \frac{0 \times 177 \times 530}{357} \right) \\
& + \left(\frac{380}{299(2+1)} + \frac{3}{2 \times 299} (299 - 121)^2 + \frac{3 \times 121^2}{2 \times 299} + \frac{0 \times 121 \times 380}{299} \right) \\
& = 2362,191868 \text{ (Sesuai)}
\end{aligned}$$

4. Untuk $n = 10$

$$\begin{aligned}
TC_{VMI} & = \left(\frac{420}{425(1+3)} + \frac{4}{2 \times 425} (425 - 204)^2 + \frac{3 \times 204^2}{2 \times 425} + \frac{0 \times 204 \times 420}{425} \right) \\
& + \left(\frac{360}{305(2+2)} + \frac{9}{2 \times 305} (305 - 250)^2 + \frac{3 \times 250^2}{2 \times 305} + \frac{0 \times 250 \times 360}{305} \right) \\
& + \left(\frac{540}{284(3+1)} + \frac{7}{2 \times 284} (284 - 150)^2 + \frac{3 \times 150^2}{2 \times 284} + \frac{0 \times 150 \times 540}{284} \right) \\
& + \left(\frac{390}{559(5+4)} + \frac{2}{2 \times 559} (559 - 198)^2 + \frac{3 \times 198^2}{2 \times 559} + \frac{0 \times 198 \times 390}{559} \right) \\
& + \left(\frac{480}{360(2+2)} + \frac{4}{2 \times 360} (360 - 163)^2 + \frac{3 \times 163^2}{2 \times 360} + \frac{0 \times 163 \times 480}{360} \right)
\end{aligned}$$

$$\begin{aligned}
& + \left(\frac{510}{514(4+2)} + \frac{6}{2 \times 514} (514 - 245)^2 + \frac{3 \times 245^2}{2 \times 514} + \frac{0 \times 245 \times 510}{514} \right) \\
& + \left(\frac{530}{279(1+3)} + \frac{5}{2 \times 279} (279 - 185)^2 + \frac{3 \times 185^2}{2 \times 279} + \frac{0 \times 185 \times 530}{279} \right) \\
& + \left(\frac{380}{516(2+1)} + \frac{3}{2 \times 516} (516 - 186)^2 + \frac{3 \times 186^2}{2 \times 516} + \frac{0 \times 186 \times 380}{516} \right) \\
& + \left(\frac{430}{477(3+4)} + \frac{2}{2 \times 477} (477 - 165)^2 + \frac{3 \times 165^2}{2 \times 477} + \frac{0 \times 165 \times 430}{477} \right) \\
& + \left(\frac{580}{424(4+2)} + \frac{8}{2 \times 424} (424 - 178)^2 + \frac{3 \times 178^2}{2 \times 424} + \frac{0 \times 178 \times 580}{424} \right) \\
& = 4042,182114 \text{ (Sesuai)}
\end{aligned}$$

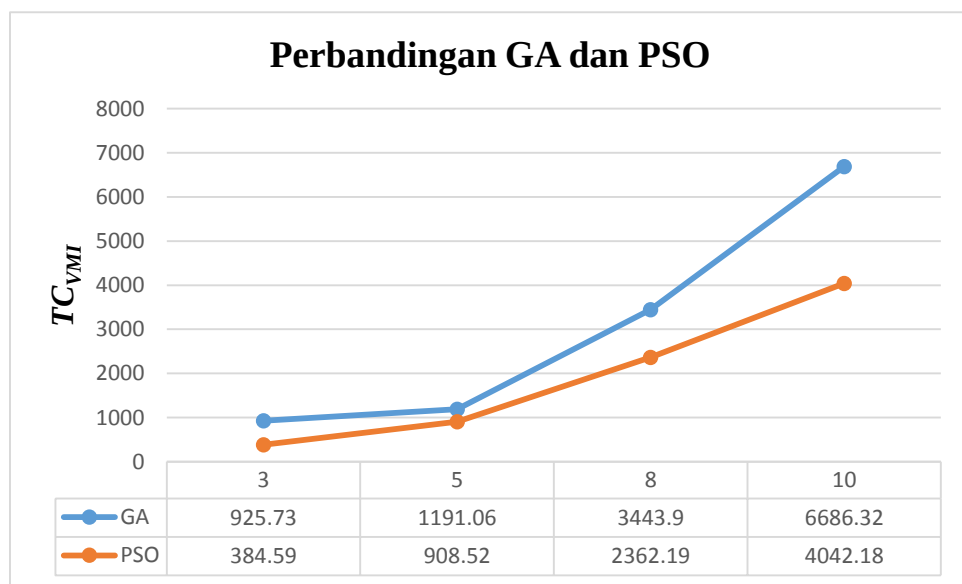
Berdasarkan hasil perhitungan manual dengan menggunakan persamaan 4.1 untuk verifikasi kendala dan total biaya, hasilnya menunjukkan bahwa hasil optimasi dengan PSO yang telah dilakukan sudah memenuhi batasan model dan total biaya sudah sesuai dengan hasil perhitungan yang telah dilakukan. Sehingga untuk Q_j , b_j , dan TC_{VMI} pada 3, 5, 8, dan 10 jenis produk dapat dibandingkan dengan hasil optimasi GA yang dilakukan Pasandideh et al. (2011).

4.7 Perbandingan Hasil Optimasi GA dan PSO

Setelah dilakukan percobaan untuk 3, 5, 8, dan 10 jenis produk ternyata hasil optimasi PSO lebih baik dibandingkan hasil optimasi GA. Hal ini dapat ditunjukkan dari total biaya inventori yang lebih minimum. Untuk lebih rinci mengenai perbandingan hasil optimasi GA dan PSO dapat ditunjukkan pada Tabel 4.13 dan Gambar 4.5 sebagai berikut:

Tabel 4.13 Perbandingan Hasil Optimasi GA dan PSO

n	TC_{VMI} (\$)		Total	Persentase
	GA	PSO	Pengurangan	Pengurangan
3	925.73	384.59	541.14	58.55 %
5	1191.06	908.52	282.54	23.72 %
8	3443.9	2362.19	1081.71	31.41 %
10	6686.32	4042.18	2644.14	39.54 %



Gambar 4.5 Grafik Perbandingan Hasil Optimasi GA dan PSO

Pada Tabel 4.13 dan Gambar 4.5 menunjukkan total biaya inventori hasil optimasi PSO lebih kecil dibandingkan hasil optimasi GA untuk 3, 5, 8, dan 10 jenis produk. Total pengurangan untuk 3 jenis produk sebesar \$ 541.14 atau 58.55 %, untuk 5 jenis produk sebesar \$ 282.54 atau 23.72 %, untuk 8 jenis produk \$ 1081.71 atau 31.41 %, dan untuk 10 jenis produk \$ 2644.14 atau 39.54 %.

BAB V

PEMBAHASAN

Pada bab ini akan didiskusikan hasil penelitian yang dilakukan sesuai dengan tujuan penelitian sebagai berikut :

5.1 Optimasi Total Biaya Inventori

Berdasarkan penelitian sebelumnya yang dilakukan oleh Yao et al. (2007) yang menyatakan bahwa model VMI dapat mengurangi biaya simpan baik pada *supplier* maupun *retailer*. Kemudian Zavanella & Zanoni (2009) meneliti sistem VMI yang diterapkan pada level *single-vendor multi-buyer* yang hasilnya menunjukkan bahwa kerjasama pengelolaan inventori memberikan peningkatan pada manfaat ekonomi. Kemudian Darwish & Odah (2010) menerapkan model VMI pada level *single-vendor multi-retailer* yang bertujuan meminimumkan total biaya inventori sehingga diperoleh keuntungan yang optimal. Kemudian Pasandideh et al. (2011) mengembangkan model VMI multi-produk multi-kendala pada level *single-supplier single-retailer* dengan metode *Genetic Algorithm*. Kemudian Yu et al. (2013) menggunakan model *Stackelberg Game* dalam menyeleksi *retailer* untuk membentuk sistem VMI yang optimal. Hingga Saleh et al. (2014) yang melanjutkan penelitian Yu et al. (2013) mengoptimasi menggunakan metode *Genetic Algorithm* dan hasilnya memberikan peningkatan keuntungan kepada manufaktur sebesar 37,14%. Maka model VMI pada penelitian-penelitian sebelumnya ini dapat dijadikan kajian penelitian yang sangat mendukung. Selain itu para peneliti-peneliti sebelumnya telah merekomendasikan

untuk menggunakan metode selain *Genetic Algorithm* dalam mengoptimasi model VMI. Hal ini telah dilakukan oleh Mousavi et al. (2013) dan Sadeghi et al. (2014) yaitu dengan menggunakan *Particle Swarm Optimization* dalam mengoptimasi model VMI. Sehingga memunculkan ide untuk mengoptimasi model VMI multi-produk multi-kendala pada level *single-supplier single-retailer* yang dikembangkan Pasandideh et al. (2011) dengan metode *Particle Swarm Optimization*. Metode *Particle Swarm Optimization* merupakan metode optimasi yang lebih baru dibandingkan metode *Genetic Algorithm*. Penelitian ini dilakukan agar dapat mengisi lebih banyak data base keilmuan khususnya dalam kajian VMI dan metode optimasi.

Metode PSO dalam mengoptimasi model dan data memiliki beberapa tahapan-tahapan. Pada awalnya dilakukan verifikasi dengan menghitung ulang secara manual model dan data tersebut. Verifikasi dibagi menjadi 2 yaitu pertama adalah menghitung jumlah Q terhadap kendala-kendala (F dan M) agar dapat diketahui apakah jumlah tersebut memenuhi batasan model atau tidak. Kemudian yang kedua adalah melakukan perhitungan secara manual terhadap hasil total biaya inventori (TC_{VMI}). Verifikasi jumlah Q dan TC_{VMI} dilakukan pada 3, 5, 8 dan 10 jenis produk. Dari hasil verifikasi menunjukkan bahwa jumlah Q memenuhi batasan-batasan model dan besarnya TC_{VMI} sudah sesuai dengan hasil yang dipaparkan dalam artikel. Setelah model dan data diverifikasi selanjutnya adalah mengoptimasi menggunakan PSO. Optimasi menggunakan PSO dilakukan untuk mencari Q_j dan b_j yang optimum pada 3, 5, 8, dan 10 jenis produk (n) sehingga didapatkan hasil TC_{VMI} yang minimum. Model dan data diprogram kedalam *software* Matlab R2013b sebagai fungsi objektif. Kemudian PSO mengoptimasi fungsi objektif tersebut menggunakan parameter jumlah populasi (N_{pop}), iterasi, parameter kognitif (c_1) dan parameter sosial (c_2) sehingga didapatkan

berbagai hasil percobaan. Untuk jumlah N_{pop} yang digunakan adalah sebanyak 80, 100, 120, 140, 160, dan 180. Untuk iterasi yang digunakan adalah sebanyak 400 iterasi. Sedangkan untuk nilai c_1 dan c_2 yang digunakan didapatkan berdasarkan percobaan yang terbaik pada percobaan $n = 3$. Sehingga pada percobaan dengan n sebesar 5, 8 dan 10 menggunakan nilai c_1 dan c_2 yang sudah optimal. Keseluruhan percobaan dipilih menjadi yang terbaik berdasarkan nilai TC_{VMI} yang paling minimum. Dari hasil percobaan untuk 3 jenis produk didapatkan percobaan terbaik dengan populasi = 120, $c_1 = 2.1$, dan $c_2 = 1.9$ yang menghasilkan TC_{VMI} sebesar \$ 384.59. Kemudian untuk 5 jenis produk percobaan terbaik adalah dengan populasi = 160, $c_1 = 1.8$, dan $c_2 = 2.2$ yang menghasilkan TC_{VMI} sebesar \$ 908.52. Untuk 8 jenis produk percobaan terbaik adalah dengan populasi = 100, $c_1 = 2.2$, dan $c_2 = 1.8$ yang menghasilkan TC_{VMI} sebesar \$ 2362.19. Dan untuk 10 jenis produk percobaan terbaik adalah dengan populasi = 120, $c_1 = 2.1$, dan $c_2 = 1.9$ yang menghasilkan TC_{VMI} sebesar \$ 4042.18.

5.2 Analisis Perbandingan GA dan PSO

Pada penelitian ini dilakukan perbandingan dua metode optimasi yaitu *Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO). Metode GA merupakan metode yang paling dikenal dan sering dipakai dalam memecahkan masalah-masalah optimasi. GA pertama kali diperkenalkan oleh Jhon Holland pada tahun 1970-an. Sedangkan PSO merupakan metode optimasi yang lebih baru dibandingkan GA. PSO pertama kali dikemukakan oleh Kennedy dan Eberhart pada tahun 1995.

PSO memiliki banyak kemiripan dengan GA, di mana sistem diawali dengan suatu populasi yang terbentuk dari solusi-solusi acak (*random solutions*) kemudian

sistem mencari optimalitas melalui pembaharuan generasi secara acak. Namun demikian, PSO tidak memiliki *evolution operators*, seperti mutasi dan *crossover* (persilangan). Sebaliknya, *potential solutions* yakni individu-individu atau yang disebut sebagai *particles*, ‘terbang’ mengikuti individu-individu yang optimum saat ini (*current optimum particles*). Setiap individu menyimpan jejak-jejak posisinya dalam dimensi masalah. Jejak-jejak posisi tersebut diartikan sebagai solusi terbaik atau *fitness* dalam GA, yang diperolehnya sejauh ini. Nilainya yakni *fitness value* yang disebut *pbest*, juga turut disimpan. Jika suatu individu memperhitungkan semua individu di dalam populasi di mana dia berada sebagai individu di sekitarnya, maka nilai terbaik yang dimaksud adalah nilai terbaik umum (*global best*) yang kemudian disebut dengan *gbest*. Selanjutnya, terjadi akselerasi pada lokasi *pbest* dari setiap individu. Akselerasi ini diberi bobot berupa bilangan acak. Dengan demikian, mekanisme berbagi informasi (*information sharing mechanism*) yang dimiliki PSO berbeda secara signifikan dengan yang dimiliki GA. Dalam GA, setiap individu yang disebut kromosom, berbagi informasi satu sama lain, sehingga keseluruhan populasi bergerak sebagai sebuah kesatuan menuju optimalitas. Dalam PSO, hanya *gbest* yang memberi informasi kepada yang lain. Ini adalah sebuah mekanisme berbagi informasi satu arah. Proses evolusi hanya mencari solusi yang terbaik. Oleh karena itu, seluruh individu yang disebut partikel, bergerak konvergen secara cepat ke solusi terbaik.

Dari hasil perbandingan total biaya inventori yang dioptimasi menggunakan metode GA dan PSO menunjukkan bahwa hasil optimasi PSO lebih baik. Hal ini dapat dibuktikan dengan adanya penurunan total biaya inventori. Penurunan yang terjadi yaitu untuk 3 jenis produk 58,55%, untuk 5 jenis produk 23,72%, untuk 8 jenis produk 31,41%, dan untuk 10 jenis produk 39,54%.

BAB VI

PENUTUP

Pada bab ini akan dipaparkan kesimpulan hasil penelitian dan saran yang perlu diberikan untuk penelitian di masa yang akan datang.

6.1 Kesimpulan

Berdasarkan diskusi dari hasil penelitian maka dapat disimpulkan sebagai berikut :

1. Total biaya inventori model VMI multi-produk multi-kendala pada level *single-supplier* dan *single-retailer* yang dioptimasi dengan PSO memberikan hasil untuk 3 jenis produk \$ 384,59, untuk 5 jenis produk \$ 908,52, untuk 8 jenis produk \$ 2362,19 dan untuk 10 jenis produk \$ 4042,18.
2. Dengan model VMI dan data yang sama ternyata optimasi PSO memberikan hasil lebih baik dibandingkan dengan optimasi GA, dengan efisiensi penurunan total biaya inventori untuk 3 jenis produk 58,55%, untuk 5 jenis produk 23,72%, untuk 8 jenis produk 31,41%, dan untuk 10 jenis produk 39,54%.

6.2 Saran

Penelitian lanjutan perlu dilakukan untuk memperkaya keilmuan khususnya dalam bidang VMI dalam *supply chain*. Dari hasil penelitian yang telah dilakukan hal tersisa yang dimungkinkan untuk diteliti adalah :

1. Model VMI pada level *single-vendor multi-retailer* yang dioptimasi dengan GA dan PSO.
2. Mengkombinasi atau *hybrid* terhadap metode GA dan PSO maupun metode-metode optimasi lainnya.

DAFTAR PUSTAKA

- Aichlmayr, M. (2000). DC mart: who manages inventory in a value chain? *Transportation & Distribution* 41, 60-68.
- Bahagia, S. N. (2006). *Sistem Inventori*. Bandung : Kampus ITB.
- Buzzell, R. D., & Ortmeyer, G. (1995). Channel Partnerships Streamline Distribution. *Sloan Management Review* 36, 85.
- Chairul Saleh., & Muhammad Ridwan, A. P., (2013). *Metodologi Penelitian : Sebuah Petunjuk Praktis*. Edisi Ke-2.
- Challener, C. (2000). Taking the VMI step to collaborative commerce. *Chemical Market Reporter*, 258(21), 11–12.
- Chase, R. B., Aquilano, N. J., & Jacobs, F. R. (2001). *Operations Management for Competitive Advantage*. Irwin/McGraw-Hill.
- Chopra, S., & Meindl, P. (2004). *Supply Chain Management: Strategy, Planning and Control*. Pearson Education. Upper Saddle River, NJ.
- Clerc, M. (1999). The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. *Proc. 1999 Congress on Evolutionary Computation*, Washington, DC, pp 1951–1957. Piscataway, NJ: IEEE Service Center
- Darwish, M. a., & Odah, O. M. (2010). Vendor managed inventory model for single-vendor multi-retailer supply chains. *European Journal of Operational Research*, 204(3), 473–484. doi:10.1016/j.ejor.2009.11.023
- Disney, S. M., & Towill, D. R. (2002). A procedure for the optimization of the dynamic response of a Vendor Managed Inventory system. *Computers & Industrial Engineering* (Vol. 43, pp. 27–58). doi:10.1016/S0360-8352(02)00061-X
- Disney, S. M., & Towill, D. R. (2003). The effect of vendor managed inventory (VMI) dynamics on the Bullwhip Effect in supply chains. *International Journal of Production Economics*, 85(2), 199–215. doi:10.1016/S0925-5273(03)00110-5
- Dong, Y., & Xu, K. (2002). A supply chain model of vendor managed inventory. *Transportation Research Part E: Logistics and Transportation Review*, 38(2), 75–95. doi:10.1016/S1366-5545(01)00014-X
- Eberhart, R. C., & Shi, Y. (2001). Particle Swarm Optimization : Developments , Applications and Resources. *Proc. Congress on Evolutionary Computation 2001*, Seoul, Korea. pp 81–86. Piscataway, NJ: IEEE Service Center
- Elsayed, E. A., & Boucher, T. O. (1994). *Analysis and control of production systems*. Prentice Hall.
- Ganeshan, R., & Harrison, T. P. (1995). *An Introduction to Supply Chain Management*. Penn State University, 5–22.
- Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. John Wiley & Sons.

- Indrajit, R. E., & Djokopranoto, R. (2002). Konsep Manajemen Supply Chain. *Cara Baru Memandang Mata Rantai Penyediaan Barang*. Jakarta: GRASINDO.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of 1995 IEEE International Conference on Neural Networks*. pp 1942–1948.
- Kwak, C., Choi, J. S., Kim, C. O., & Kwon, I.-H. (2009). Situation reactive approach to Vendor Managed Inventory problem. *Expert Systems with Applications*, 36(5), 9039–9045. doi:10.1016/j.eswa.2008.12.018
- Lamming, R. C., Caldwell, N. D., Harrison, D. A., & Phillips, W. (2001). Transparency in supply relationships: concept and practice. *Journal of Supply Chain Management*, 37(3), 4–10.
- Lee, C. C., & Chu, W. H. J. (2005). Who should control inventory in a supply chain? *European Journal of Operational Research*, 164(1), 158–172. doi:10.1016/j.ejor.2003.11.009
- Mentzer, J. T., Keebler, J. S., Nix, N. W., Smith, C. D., & Zacharia, Z. G. (2001). *Journal Of Business Logistics*, Vol.22, No. 2, 2001 1, 22(2), 1–25.
- Mousavi, S. M., Hajipour, V., Niaki, S. T. A., & Aalifar, N. (2013). A multi-product multi-period inventory control problem under inflation and discount: a parameter-tuned particle swarm optimization algorithm. *The International Journal of Advanced Manufacturing Technology*, 70(9-12), 1739–1756. doi:10.1007/s00170-013-5378-y
- Nachiappan, S. P., & Jawahar, N. (2007). A genetic algorithm for optimal operating parameters of VMI system in a two-echelon supply chain. *European Journal of Operational Research*, 182(3), 1433–1452. doi:10.1016/j.ejor.2006.09.040
- Pasandideh, S. H. R., & Niaki, S. T. A. (2008). A genetic algorithm approach to optimize a multi-products EPQ model with discrete delivery orders and constrained space. *Applied Mathematics and Computation*, 195(2), 506–514. doi:10.1016/j.amc.2007.05.007
- Pasandideh, S. H. R., Niaki, S. T. A., & Nia, A. R. (2011). A genetic algorithm for vendor managed inventory control system of multi-product multi-constraint economic order quantity model. *Expert Systems with Applications*, 38(3), 2708–2716. doi:10.1016/j.eswa.2010.08.060
- Purba, R., Halim, A. & Ridwan, A., 2014. Fragile Image Watermarking Berbasis Dct Dengan Operator Evolusi Hybrid Of Particle. *ISSN. 1412-0100*, 15(1), pp.11–20.
- Rangkuti, F. (2004). *Manajemen Persediaan: Aplikasi di bidang bisnis*. Jakarta: Fajar Interpratama Offset.
- Sadeghi, J., Sadeghi, S., & Niaki, S. T. A. (2014). Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm. *Information Sciences*, 272, 126–144. doi:10.1016/j.ins.2014.02.075
- Saleh, C., Hassan, A., Deros, B. M., Rahman, M. N. A., Leuveano, R. A. C., & Adiyoga, A. (2014). Parameter Optimization of VMI system in a Manufacturer and Multi Retailer using Genetic Algorithm. *International Conference on Advanced Manufacturing and Materials Engineering*.

- Santosa, B. (2006). *Tutorial Particle Swarm Optimization*. Surabaya: Kampus ITS.
- Schonberger, R. J. (1996). Strategic collaboration: Breaching the castle walls. *Business Horizons*, 39(2), 20–26.
- Simchi-Levi, D. (2000). *Designing and managing the supply chain*. McGraw-Hill College.
- Skousen, K. F., Stice, E. K., & Stice, J. D. (2000). *Intermediate Accounting*. South-Western College Pub.
- Teng, J.-T. (2009). A simple method to compute economic order quantities. *European Journal of Operational Research*, 198(1), 351–353. doi:10.1016/j.ejor.2008.05.019
- Tersine, R. J. (1994). *Principles Of Inventory And Materials Management*. Prentice Hall. Englewood Cliffs, NJ.
- Tuegeh, M., Soeprijanto & Purnomo, M.H., (2009). Modified Improved Particle Swarm Optimization for Optimal. *Seminar Nasional Aplikasi Teknologi Informatika 2009 (SNATI 2009)*, 85-90.
- Tyan, J., & Wee, H. (2003). Vendor managed inventory : a survey of the Taiwanese grocery industry. *Journal of Purchasing & Supply Management*, 9, 11–18.
- Vanany, I., Zailani, S., & Pujawan, N. (2009). Supply chain risk management: literature review and future research. *International Journal of Information Systems and Supply Chain Management (IJISSCM)*, 2(1), 16–33.
- Wati, D. A. R. (2011). *Sistem Kendali Cerdas*. Graha Ilmu.
- Yao, Y., Evers, P. T., & Dresner, M. E. (2007). Supply chain integration in vendor-managed inventory. *Decision Support Systems*, 43(2), 663–674. doi:10.1016/j.dss.2005.05.021
- Yu, Y., Hong, Z., Zhang, L. L., Liang, L., & Chu, C. (2013). Optimal selection of retailers for a manufacturing vendor in a vendor managed inventory system. *European Journal of Operational Research*, 225(2), 273–284. doi:10.1016/j.ejor.2012.09.044
- Yu, Y., Wang, Z., & Liang, L. (2012). A vendor managed inventory supply chain with deteriorating raw materials and products. *International Journal of Production Economics*, 136(2), 266–274. doi:10.1016/j.ijpe.2011.11.029
- Zavanella, L., & Zanoni, S. (2009). A one-vendor multi-buyer integrated production-inventory model: The “Consignment Stock” case. *International Journal of Production Economics*, 118(1), 225–232. doi:10.1016/j.ijpe.2008.08.044

LAMPIRAN

Lampiran 1. Program Matlab PSO

```
% Particle Swarm Optimization - PSO
% Haupt & Haupt
% 2004

clear
clc
ff = 'TC'; % Objective function

% Initializing variables
popsize = 80; % size of the swarm
npar = 6; % dimension of the problem
maxit = 400; % maximum number of iterations
c1 = 2; % cognitive parameter
c2 = 2; % social parameter
% constriction factor
C = 2/abs(2-(c1+c2)-sqrt(((c1+c2)^2)-(4*(c1+c2))));

% Initializing swarm and velocities
par = rand(popsize,npar); % random population of continuous
                           values
vel = rand(popsize,npar); % random velocities

% evaluate initial population
cost=feval(ff,par); % calculate population cost using ff
minc(1)=min(cost); % min cost
meanc(1)=mean(cost); % mean cost
globalmin=minc(1); % initialize global minimum

% Initialize local minimum for each particle
localpar = par; % location of local minima
localcost = cost; % cost of local minima

% Finding best particle in initial population
[globalcost, indx] = min(cost);
globalpar=par(indx,:);

%-----Start Iteration-----

iter = 0; % counter

while iter < maxit
    iter = iter + 1;

    % update velocity = vel
    w=(maxit-iter)/maxit; % inertia weight
    r1 = rand(popsize,npar); % random numbers
```

```

    r2 = rand(popsize,npar); % random numbers
    vel = C*(w*vel + c1*r1.*(localpar-par) +
c2*r2.*(ones(popsize,1)*globalpar-par));

% update particle positions
par = par + vel; % update particle position
overlimit=par<=1;
underlimit=par>=0;
par=par.*overlimit+not(overlimit);
par=par.*underlimit;

% Evaluate the new swarm
cost = feval(ff,par); % evaluate cost of swarm

% updating the best local position for each particle
bettercost = cost < localcost;
localcost = localcost.*not(bettercost) + cost.*bettercost;
localpar(find(bettercost),:) = par(find(bettercost),:);

% Updating index g
[temp, t] = min(localcost);
if temp<globalcost
    globalpar=par(t,:); indx=t; globalcost=temp;
end
[iter globalpar globalcost] %print out each iteration
minc(iter+1)=min(cost); % min for this iteration
globalmin(iter+1)=globalcost; % best min so far
meanc(iter+1)=mean(cost); % avg. cost for this iteration

end % while

figure(1)
iters=0:length(minc)-1;
plot(iters,globalmin,'--');
xlabel('Iterations');ylabel('Cost');

```

Lampiran 2. Program Fungsi Total Cost

```
function cost=TC(par)
%% Inisialisasi Data
Dj = [420 360 540 390 480 510 530 380 430 580];
Ajs = [1 2 3 5 2 4 1 2 3 4];
Ajr = [3 2 1 4 2 2 3 1 4 2];
hjr = [4 9 7 2 4 6 5 3 2 8];
fj = [3 2 3 1 4 3 2 1 3 4];
F = 18000;
M = 12;
phi= 3;

%% Perhitungan
[n,k]=size(par);
MTC=[];
Fsum=[];
Msum=[];
for p=1:n;

    bbQ=1;
    baQ=600;
    bbb=1;
    bab=300;

    Qj=round(bbQ+(baQ-bbQ)*par(p,1:k/2));
    bj=round(bbb+(bab-bbb)*par(p,(k/2)+1:k));

    for y=1:k/2;

        Fhitung=fj(y)*Qj(y);
        Fsum=[Fsum Fhitung];

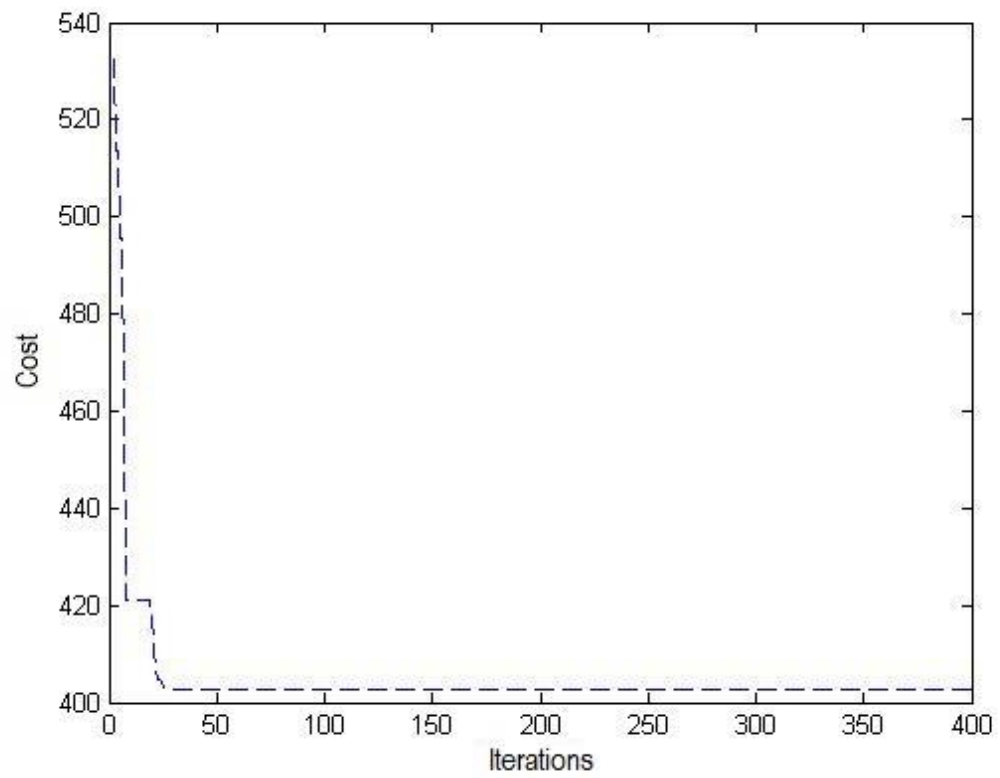
        Mhitung=Dj(y)/Qj(y);
        Msum=[Msum Mhitung];

    end

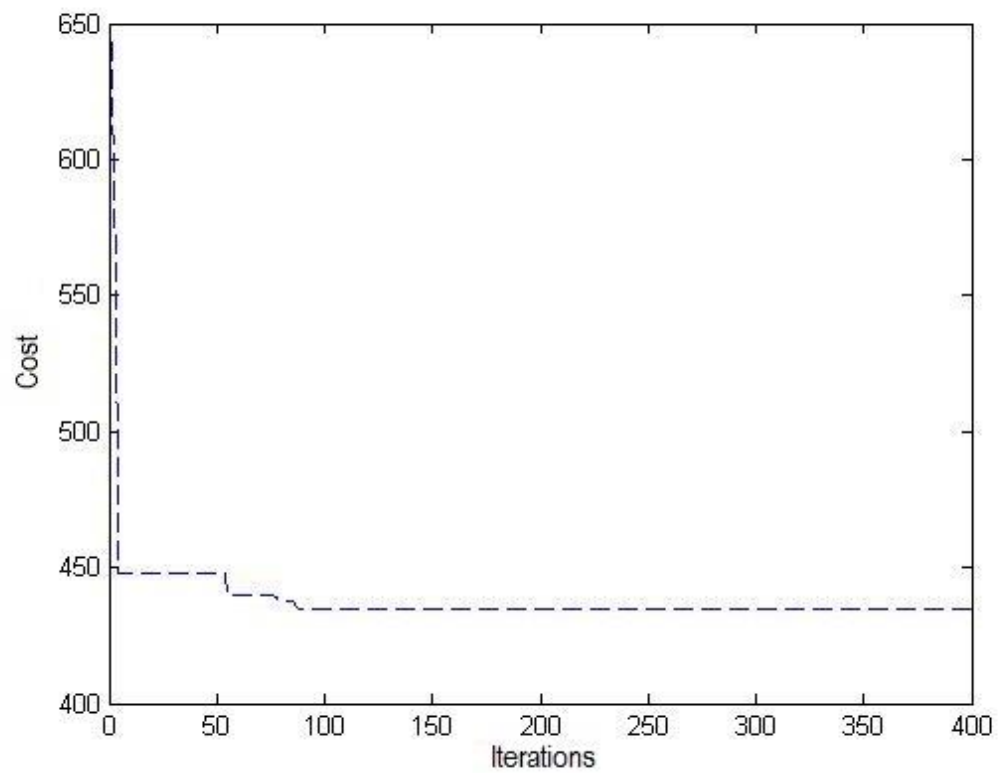
    TC=(Dj(y)/Qj(y)*(Ajs(y)+Ajr(y)))+(hjr(y)/(2*Qj(y)))*((Qj(y)-
    bj(y))^2)+(phi*(bj(y)^2)/(2*Qj(y)));
    MTC=[MTC TC];
end

if sum(Fsum)<=F && sum(Msum)<=M
    cost(p)=sum(MTC);
else
    cost(p)=9999;
end
end
```

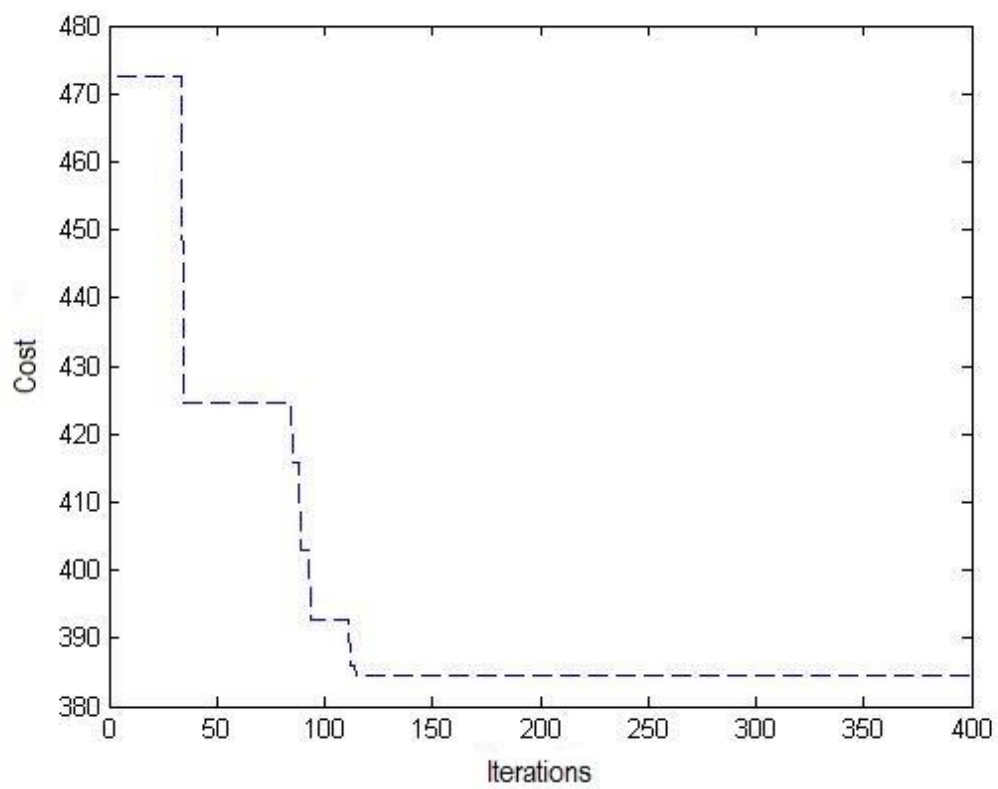
Lampiran 3. Grafik Konvergen Hasil Percobaan $n = 3$



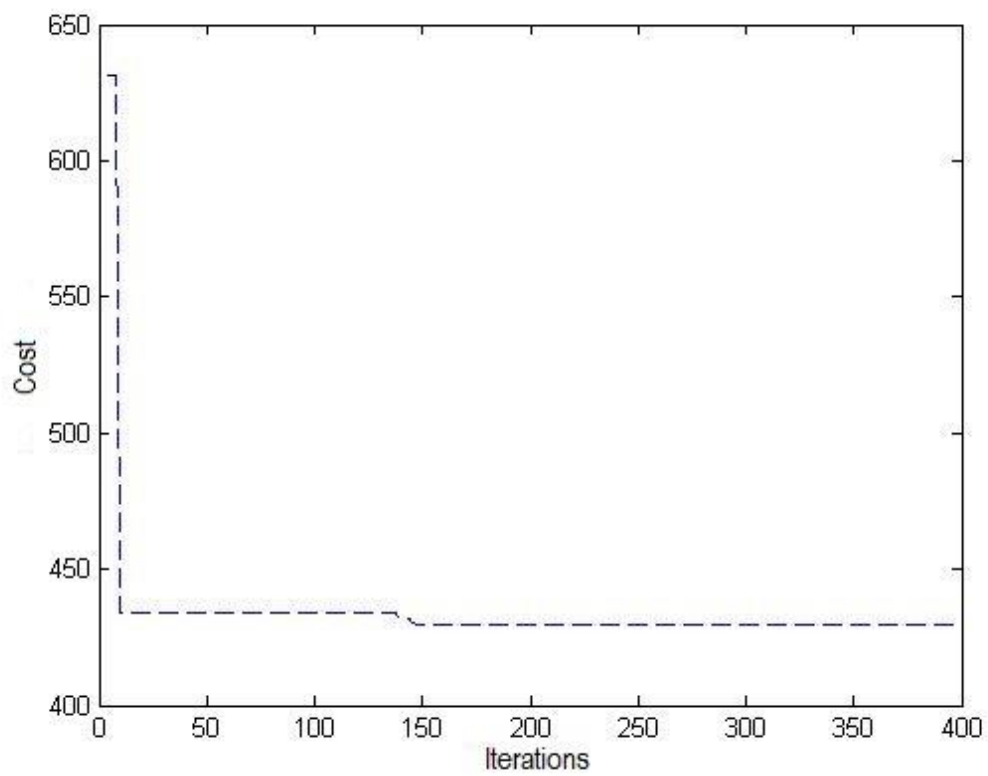
Popsiz = 80



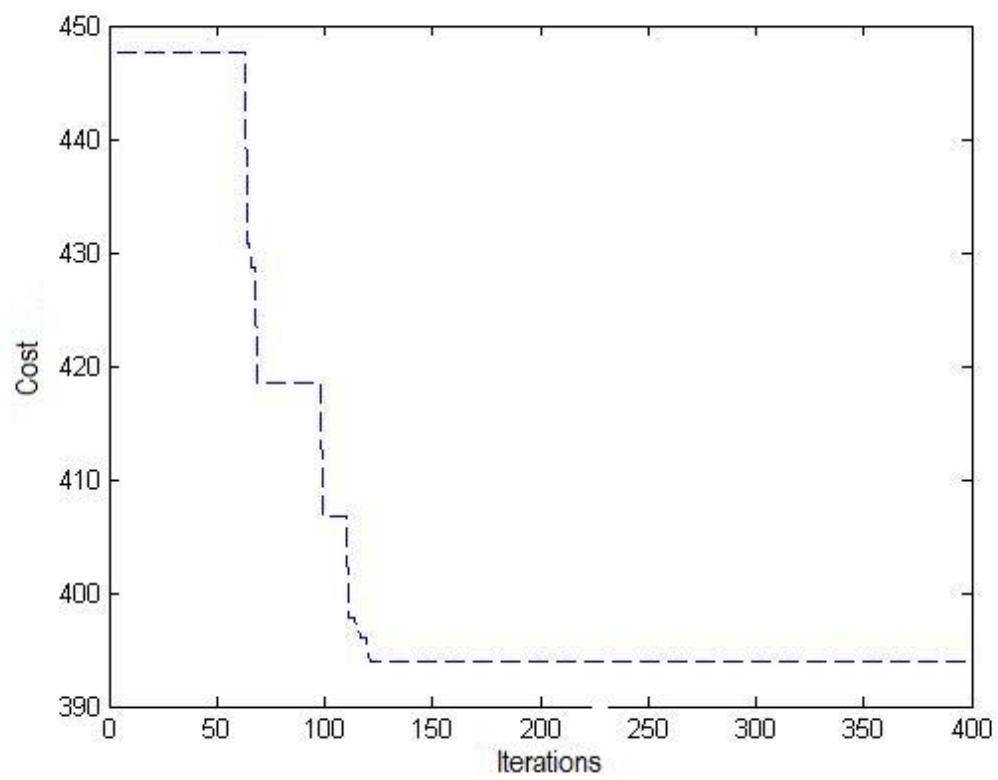
Popsiz = 100



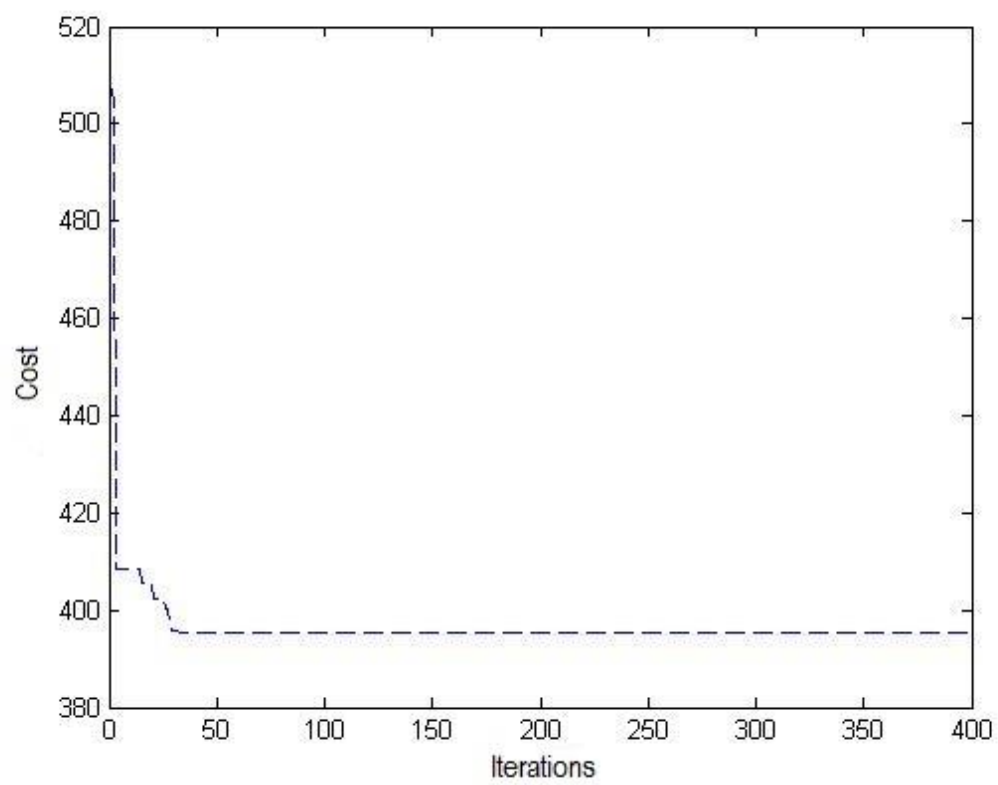
Popsiz = 120



Popsiz = 140

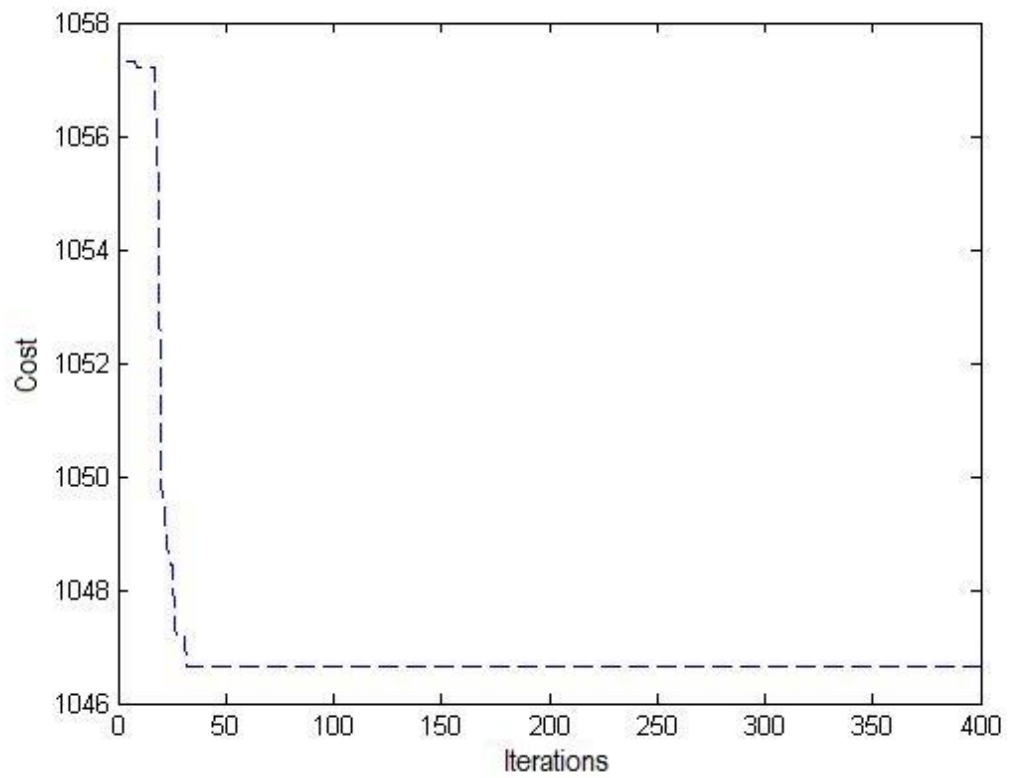


Popsiz = 160

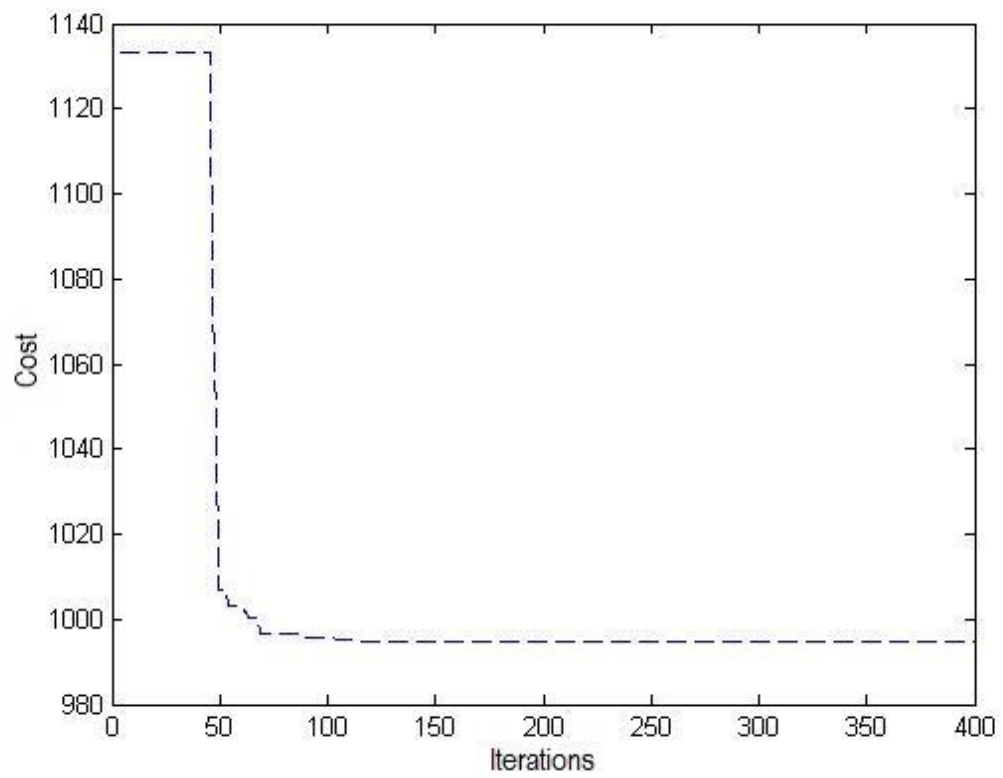


Popsiz = 180

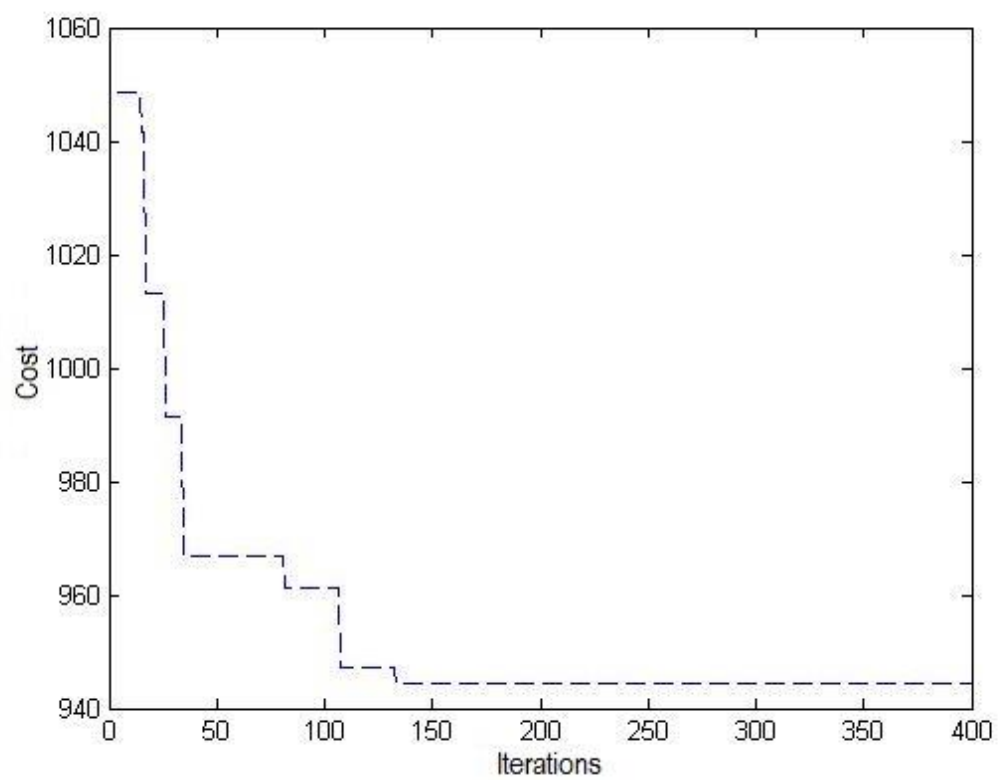
Lampiran 4. Grafik Konvergen Hasil Percobaan $n = 5$



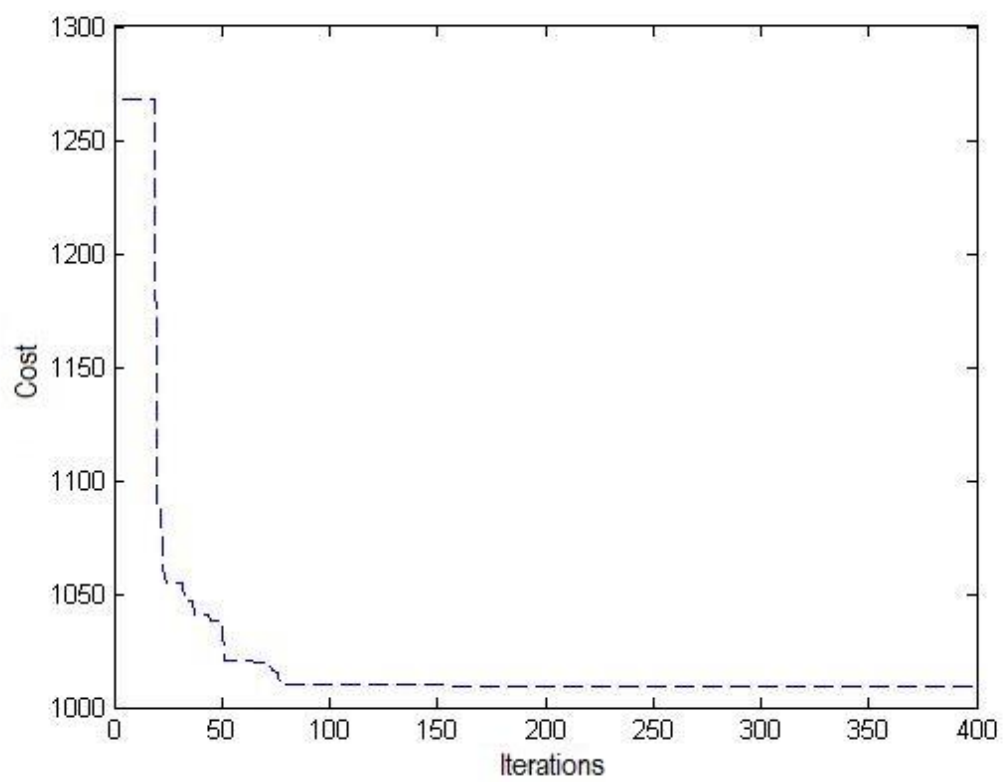
Popsiz = 80



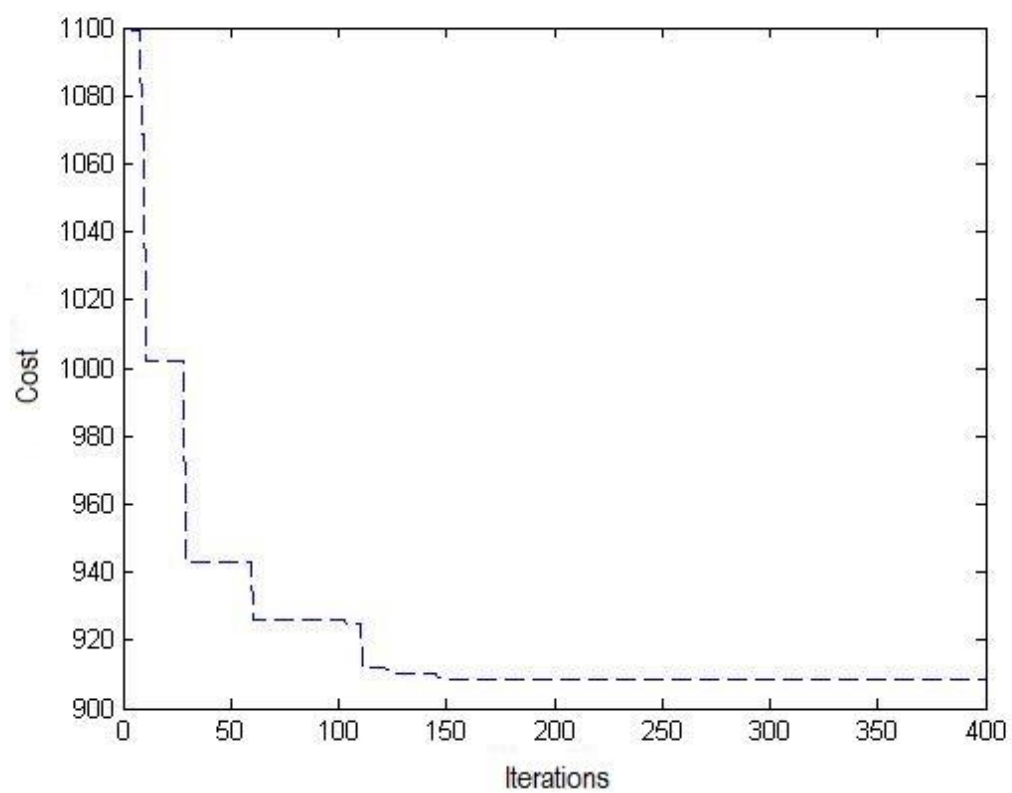
Popsiz = 100



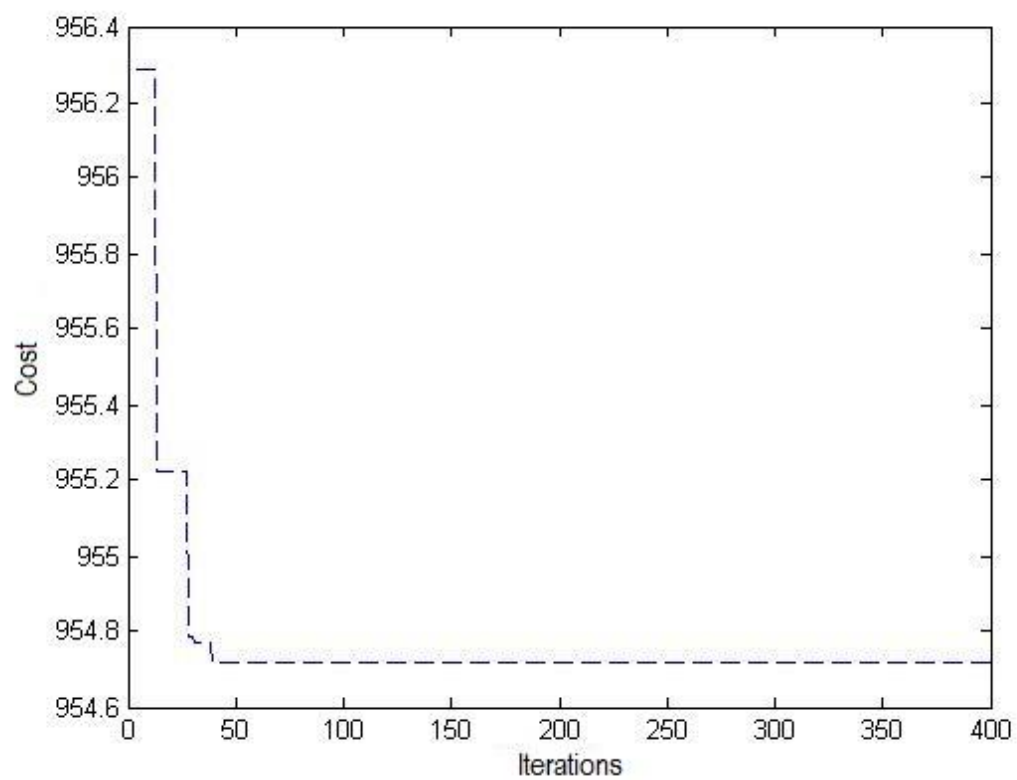
Popsiz = 120



Popsiz = 140

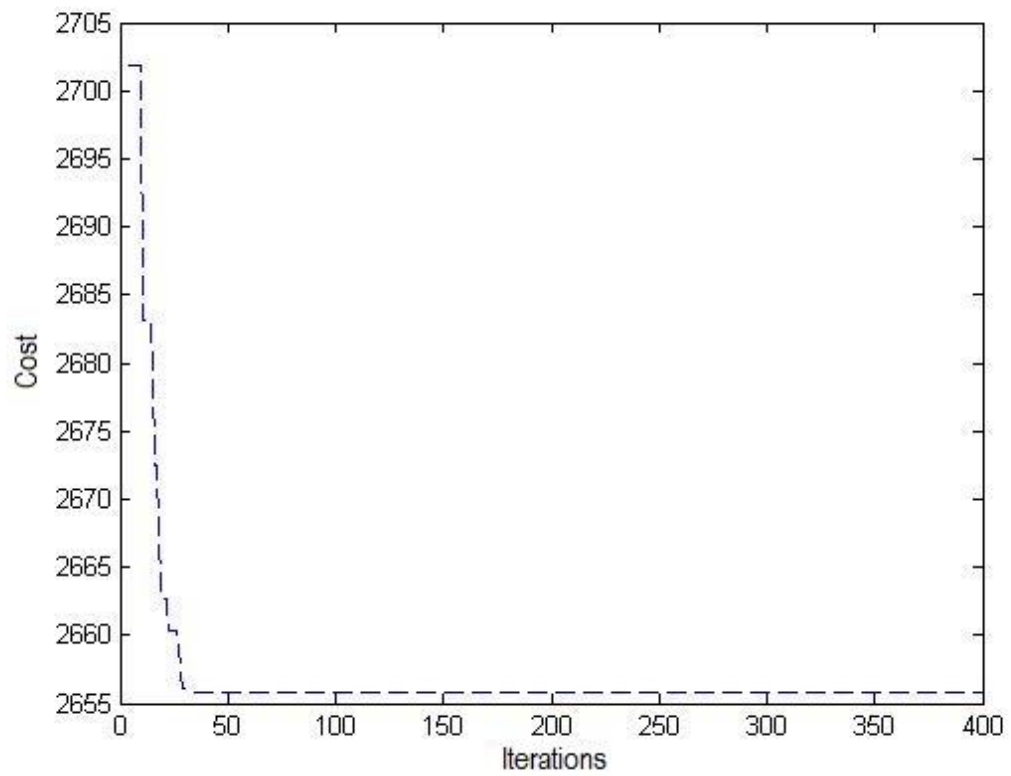


Popsiz = 160

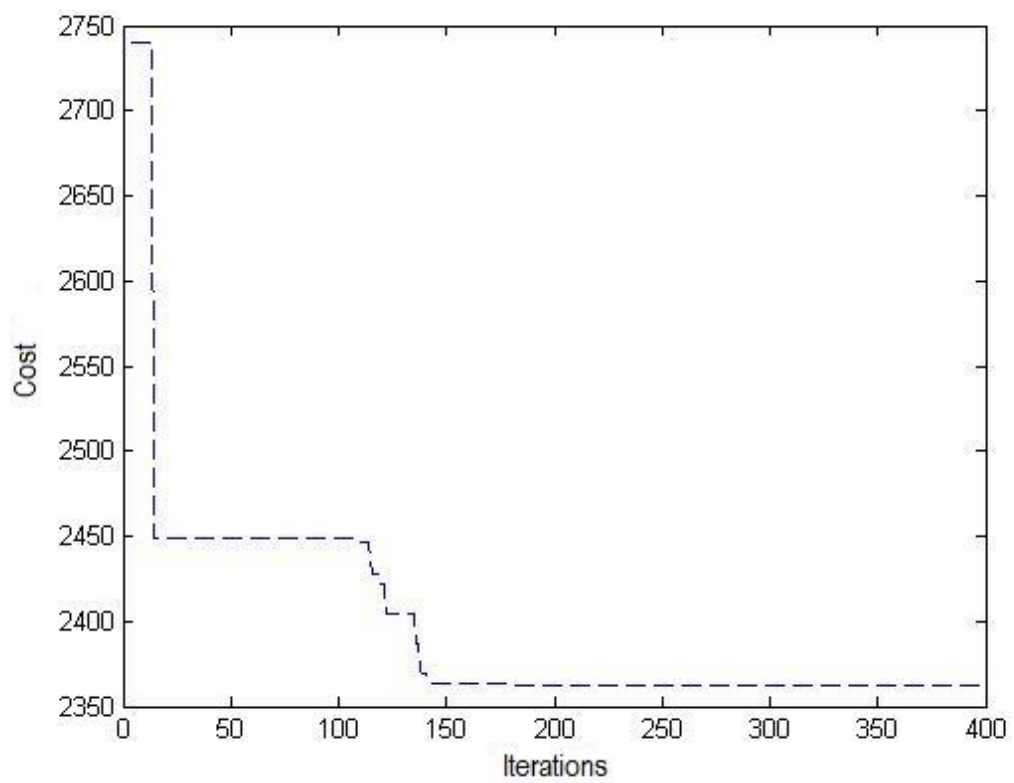


Popsiz = 180

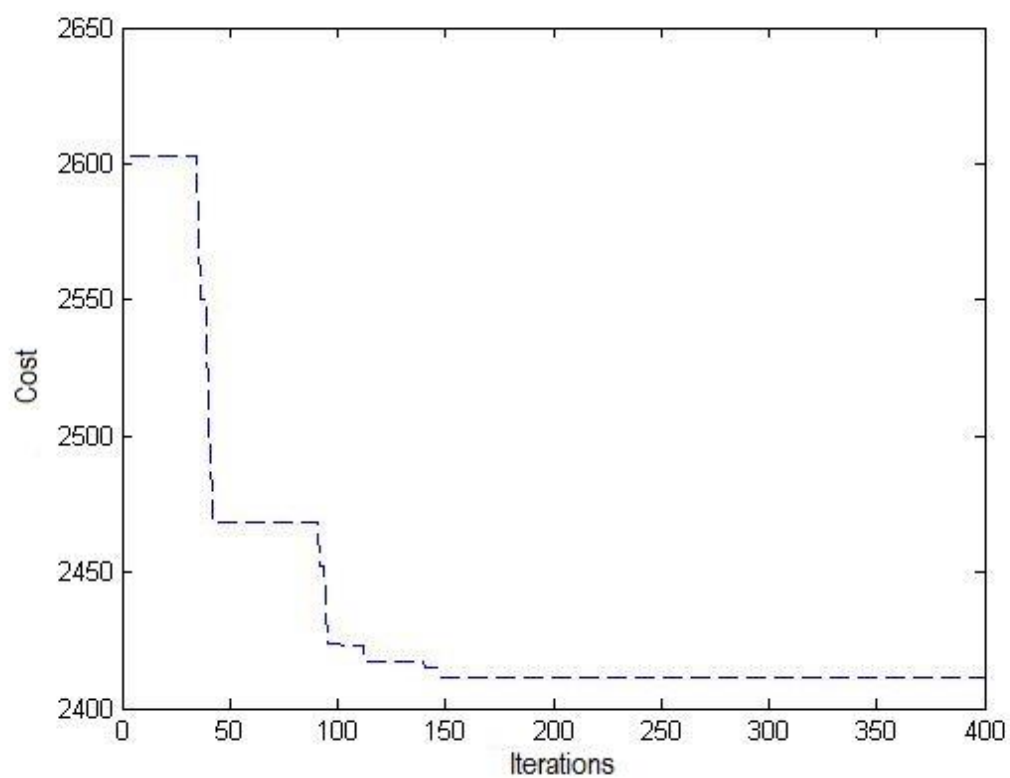
Lampiran 5. Grafik Konvergen Hasil Percobaan $n = 8$



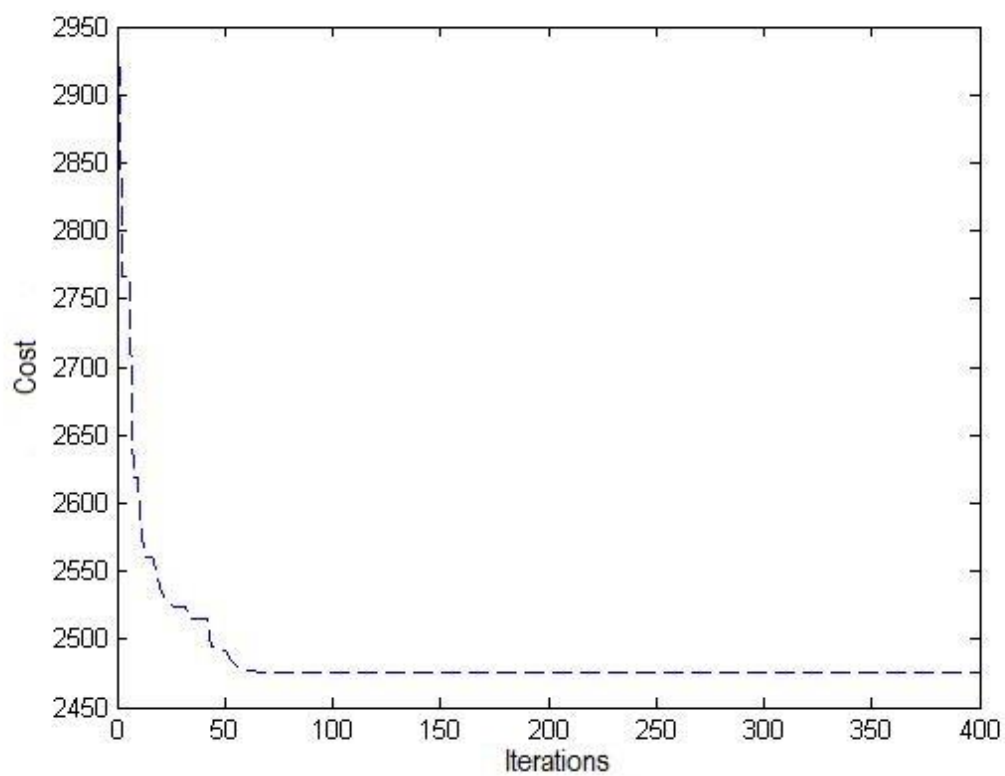
Popsiz = 80



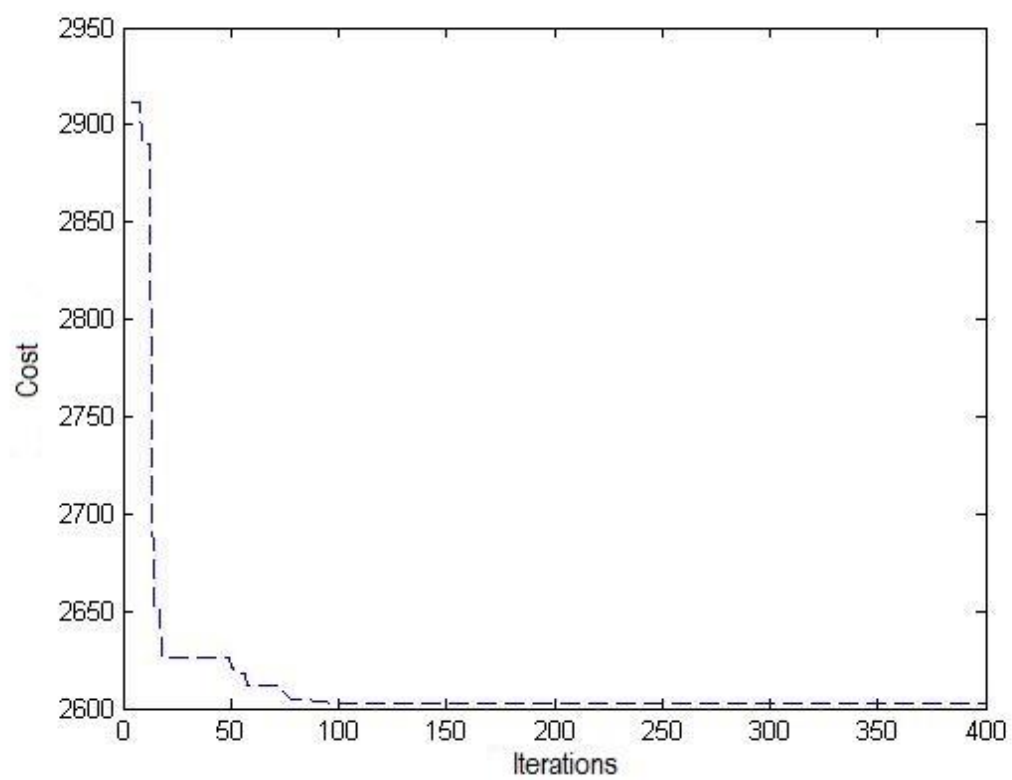
Popsiz = 100



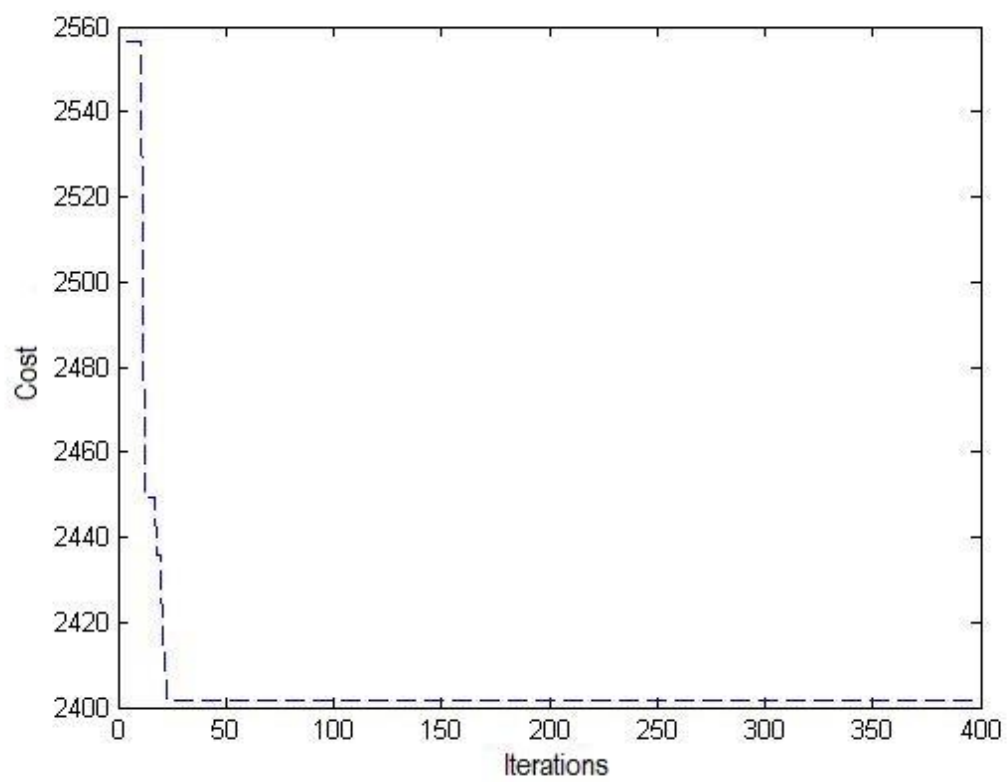
Popsiz = 120



Popsiz = 140

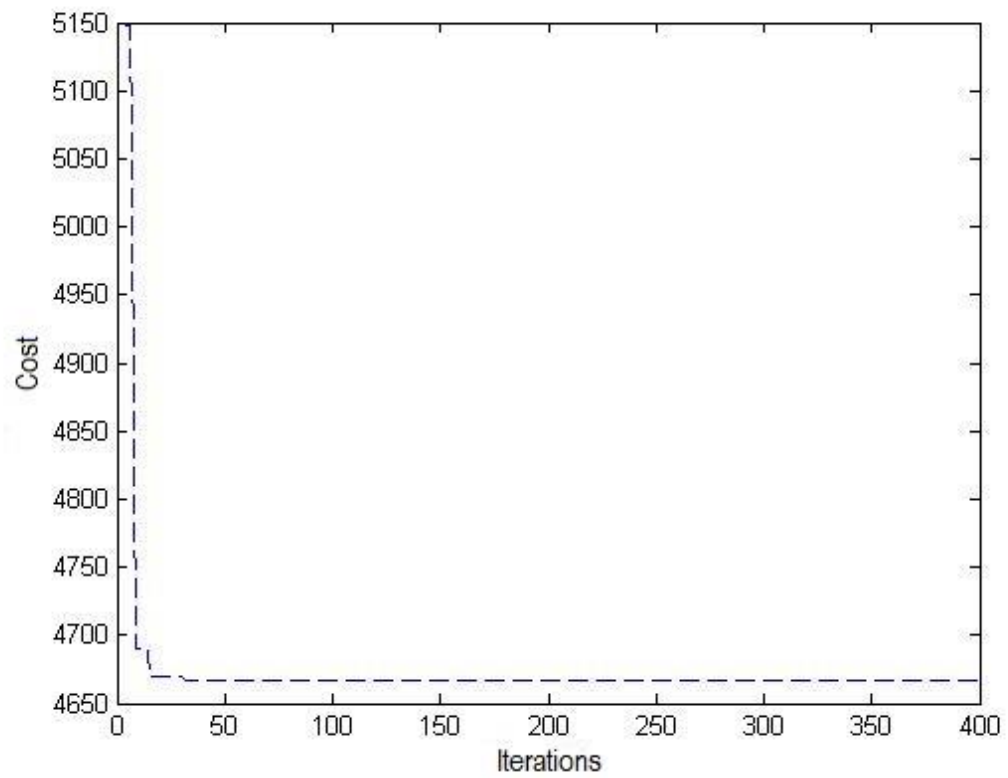


Popsiz = 160

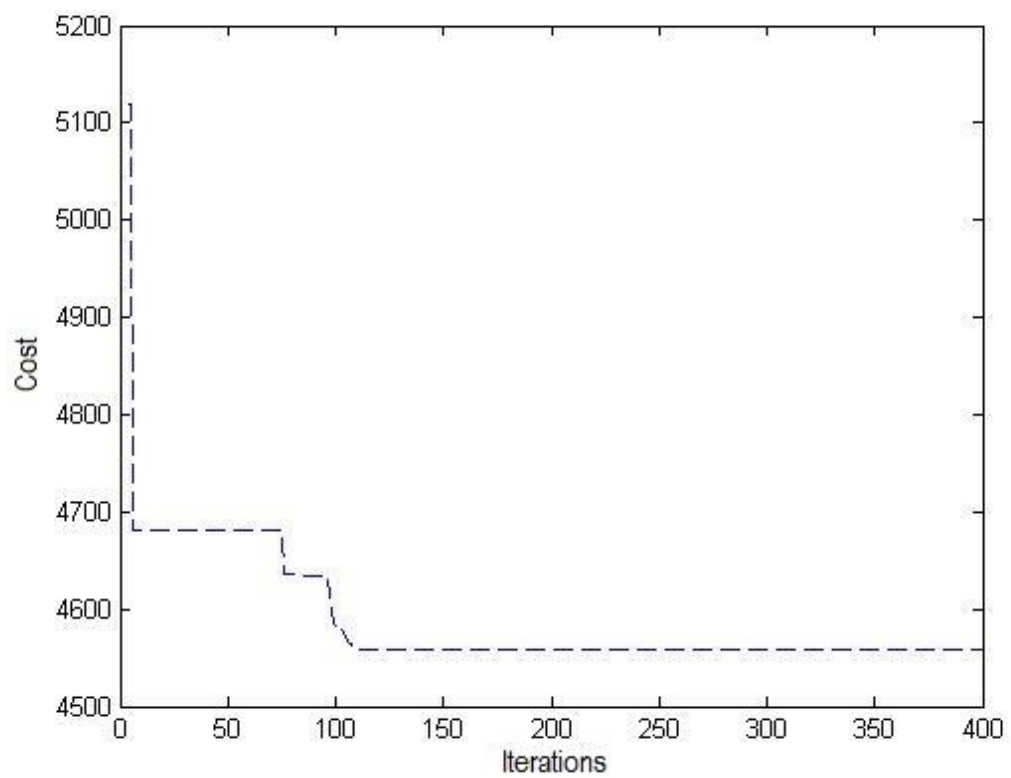


Popsiz = 180

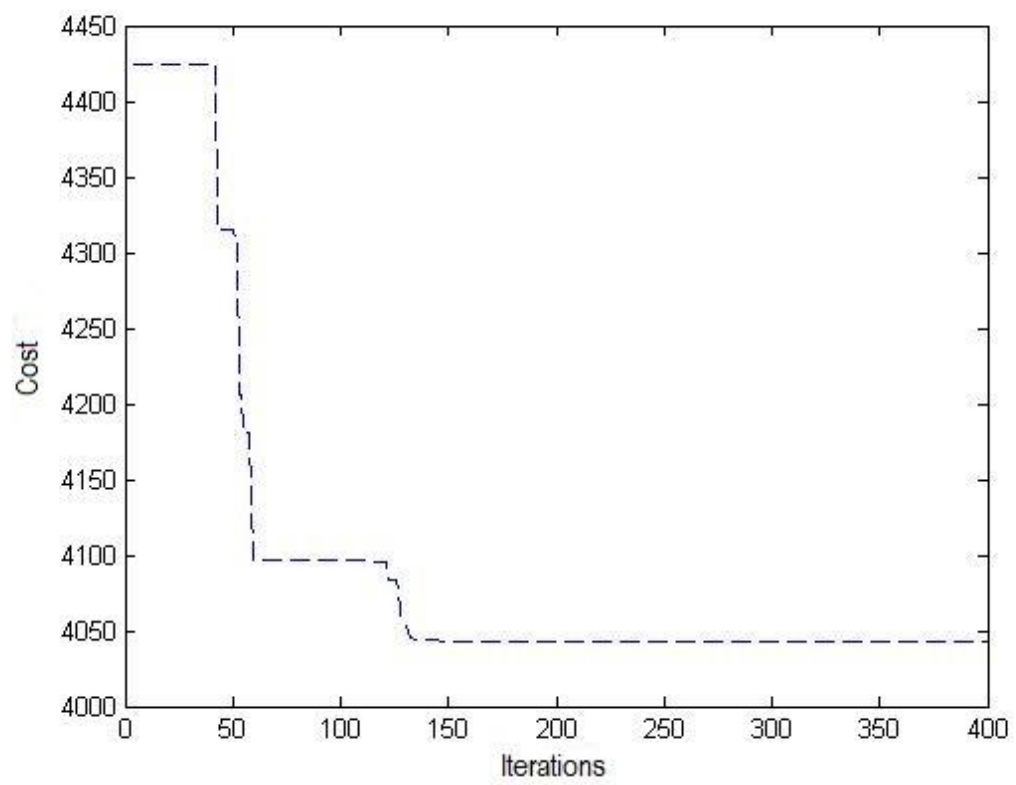
Lampiran 6. Grafik Konvergen Hasil Percobaan $n = 10$



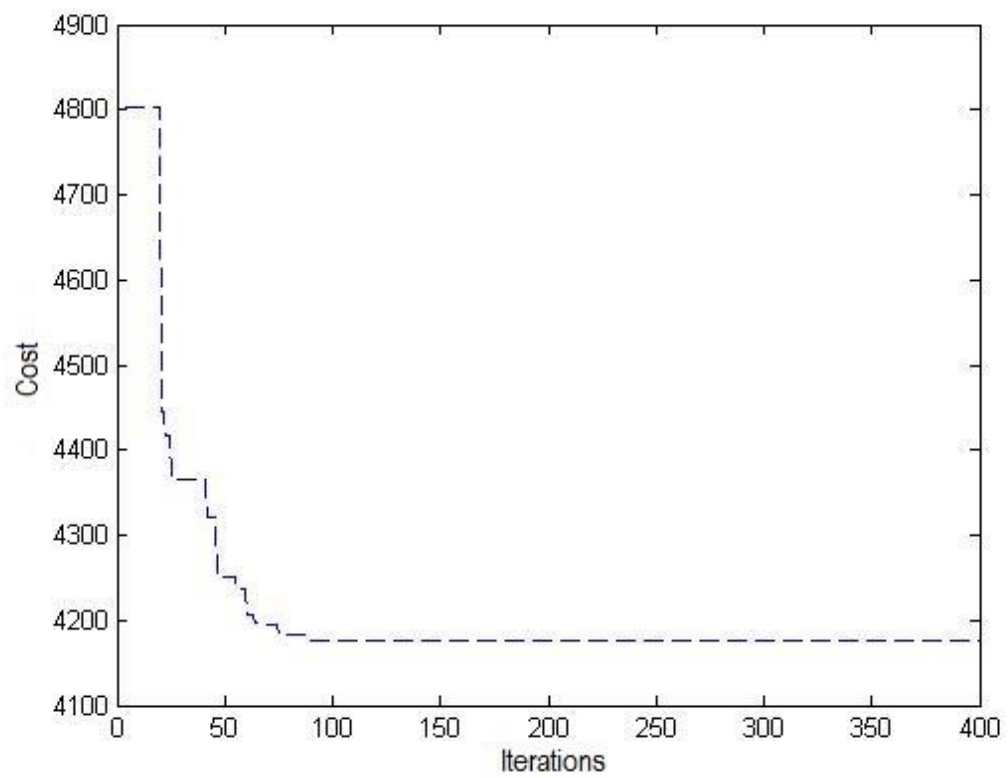
Popsiz = 80



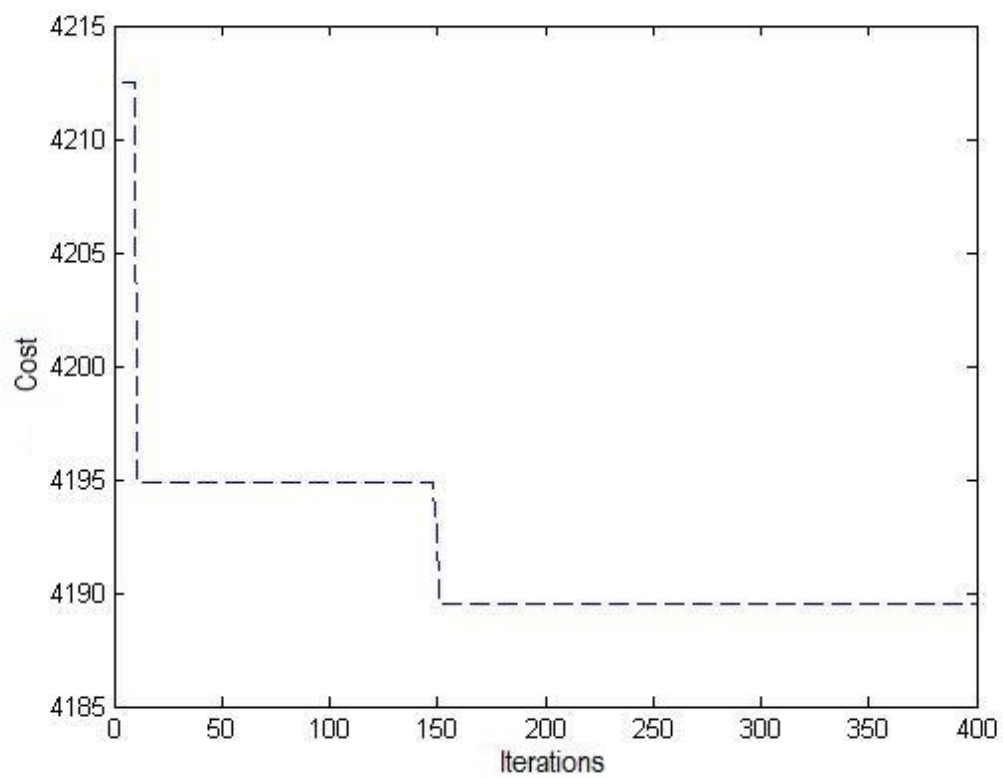
Popsiz = 100



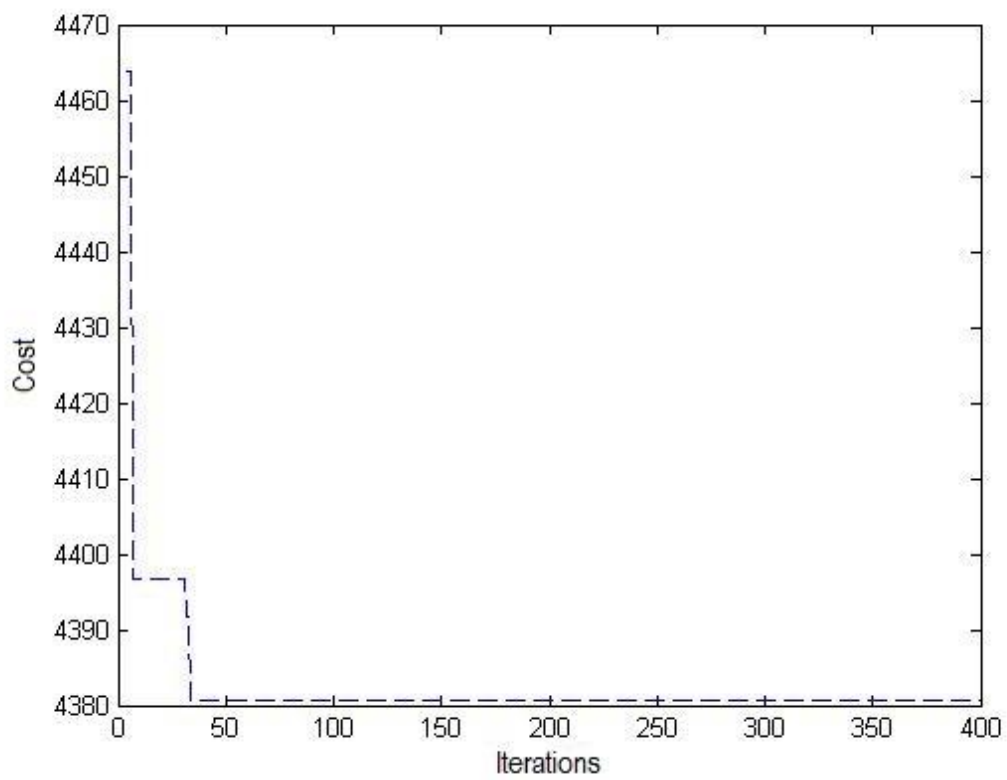
Popsiz = 120



Popsiz = 140



Popsiz = 160



Popsiz = 180