ANALISIS *INERTIA WEIGHT* PADA ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO) UNTUK OPTIMALISASI DAN PEMODELAN SISTEM TERHADAP PERSOALAN *VEHICLE ROUTING PROBLEM WITH TIME WINDOW* (VRPTW)

SKRIPSI

UIN SUNAN AMPEL
S U R A B A Y A

Disusun Oleh

AYU NURUL ATIQOH

H72216049

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA

2020

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama	:	AYU NURUL ATIQOH
NIM	:	H72216049
Program Studi	:	Matematika
Angkatan	:	2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul " ANALISIS *INERTIA WEIGHT* PADA ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO) UNTUK OPTIMALISASI DAN PEMODELAN SISTEM TERHADAP PERSOALAN *VEHICLE ROUTING PROBLEM WITH TIME WINDOW* (VRPTW)". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 15 Juli 2020

Yang menyatakan,

AYU NURUL ATIQOH

NIM. H72216049

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh

Nama	:	AYU NURUL ATIQOH
NIM	:	H72216049
Judul Skripsi	:	ANALISIS *INERTIA WEIGHT* PADA *PARTICLE SWARM OPTIMIZATION* (PSO) UNTUK OPTIMALISASI DAN PEMODELAN SISTEM TERHADAP PERSOALAN *VEHICLE ROUTING PROBLEM WITH TIME WINDOW* (VRPTW)

telah diperiksa dan disetujui untuk diujikan.

Surabaya, 15 Juli 2020

Pembimbing

Yuniar Farida, M.T

NIP. 197905272014032002

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi oleh

Nama	:	AYU NURUL ATIQOH
NIM	:	H72216049
Judul Skripsi	:	ANALISIS *INERTIA WEIGHT* PADA *PARTICLE SWARM OPTIMIZATION* (PSO) UNTUK OPTIMALISASI DAN PEMODELAN SISTEM TERHADAP PERSOALAN *VEHICLE ROUTING PROBLEM WITH TIME WINDOW* (VRPTW)

Telah dipertahankan di depan Tim Penguji
pada tanggal 24 Juli 2020

Mengesahkan,
Tim Penguji

Penguji I

Yuniar Farida, M.T
NIP. 197905272014032002

Penguji II

Aris Fanani, M.Kom
NIP. 198701271014031002

Penguji III

Putroue Keumala Intan, M.Si
NIP. 198805282018012001

Penguji IV

Wika Dianita Utami, M.Sc
NIP. 199206102018012003

Mengetahui,
Plt. Dekan Fakultas Sains dan Teknologi
UIN Sunan Ampel Surabaya

Dr. Evi Fatimatur Rusydiyah, M.Ag
NIP. 197312272005012003

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : Ayu Nurul Atiqoh
NIM : H72216049
Fakultas/Jurusan : Saintek/Matematika
E-mail address : ayunurulatiqoh72@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :
☑ Skripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :

ANALISIS *INERTIA WEIGHT* PADA *PARTICLE SWARM OPTIMIZATION* (PSO)
UNTUK OPTIMALISASI DAN PEMODELAN SISTEM TERHADAP PERSOALAN
VEHICLE ROUTING PROBLEM WITH TIME WINDOW (VRPTW)

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 18 Agustus 2020

Penulis

(AYU NURUL ATIQOH)

ABSTRAK

ANALISIS *INERTIA WEIGHT* PADA ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO) UNTUK OPTIMALISASI DAN PEMODELAN SISTEM TERHADAP PERSOALAN *VEHICLE ROUTING PROBLEM WITH TIME WINDOW* (VRPTW)

Permasalahan terkait kegiatan distribusi memiliki aspek yang perlu diperhatikan yaitu bagaimana mekanisme menyalurkan barang ke sejumlah konsumen dengan tepat. Suatu perusahaan harus mampu menyusun penjadwalan yang efektif dan rute pengiriman yang tepat sehingga dapat meminimalkan jarak dan biaya yang dikeluarkan. Permasalahan terkait distribusi tersebut dikenal dengan *Vehicle Routing Problem* (VRP). Pada penelitian ini menggunakan tipe persoalan *Vehicle Routing Problem with Time Window* (VRPTW). VRPTW memiliki tambahan fungsi kendala dengan batasan model *time window* di setiap nodenya sehingga permasalahan semakin kompleks. Tujuan dari VRPTW adalah untuk meminimalkan total jarak dengan mempertimbangkan batasan *time window* tersebut. Penelitian ini terinspirasi dari permasalahan benchmark Solomon pada persoalan VRPTW. Metode optimalisasi yang digunakan adalah algoritma *Particle Swarm Optimization* (PSO) yang merupakan metode optimasi metaheuristik dan bersifat independen sehingga dapat diimplementasikan untuk persoalan VRPTW . Tujuan pada penelitian ini adalah menganalisis penerapan algoritma PSO yang dirancang untuk menguji validitas model pada suatu persoalan VRPTW. Selain itu, algoritma PSO akan dimodifikasi berdasarkan perhitungan nilai dari tipe *inertia weight* yang digunakan dalam penelitian. Solusi optimum penelitian menunjukkan jarak terpendek diperoleh dari penerapan algoritma PSO dengan modifikasi strategi *random inertia weight* dengan total jarak 3821 satuan jarak. Sedangkan, total jarak terpanjang diperoleh berdasarkan hasil modifikasi algoritma PSO dengan strategi *Global-Local Inertia Weight* dengan total jarak 3959 satuan jarak.

Kata kunci: optimasi, distribusi, algoritma PSO, VRPTW

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

ABSTRACT

ANALYSIS OF INERTIA WEIGHT IN PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM TO OPTIMIZATIOAN AND MODELING SYSTEM ON VEHICLE ROUTING PROBLEM WITH TIME WINDOW (VRPTW)

Problems related to distribution activities have the necessary aspects note that the mechanism for distributing goods to a number consumers to be exact. The company must be able to arrange scheduling effective and precise shipping routes that minimize distance and costs incurred. Problems with distribution are known with Vehicle Routing Problem (VRP). In this study using types Vehicle Routing Problems with Time Window (VRPTW) problems. VRPTW has additional constraint functions with time window model constraints on each node so the problem is more complex. The purpose of VRPTW is to minimize the total distance by considering the window timeout. This research was inspired by Solomon's benchmark problem in VRPTW problem. The optimization method used is the Particle algorithm Swarm Optimization (PSO) which is a metaheuristic optimization method and is independent so that it can be implemented for VRPTW problems. The purpose of this study is to analyze the application of the PSO algorithm designed to test the validity of the model on VRPTW problems. Other than that, The PSO algorithm will be modified based on the calculation of the type of inertia Weight used in research. The optimal solution shown by this research the shortest distance is obtained from the application of the PSO algorithm with modification random inertia weight strategy with a total distance of 3821 distance units. While the longest distance is obtained based on the modification of the PSO algorithm with the Global-Local Weight Inertia strategy with a total distance of 3959 distance units.

Keywords: optimation, distribution, PSO algorithm, VRPTW

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR ISI

HALAMAN JUDUL . . . . . **i**
LEMBAR PERSETUJUAN PEMBIMBING . . . . . **ii**
PENGESAHAN TIM PENGUJI SKRIPSI . . . . . **iii**
HALAMAN PERNYATAAN KEASLIAN . . . . . **iv**
MOTTO . . . . . **v**
HALAMAN PERSEMBAHAN . . . . . **vi**
KATA PENGANTAR . . . . . **vii**
DAFTAR ISI . . . . . **viii**
DAFTAR TABEL . . . . . **x**
DAFTAR GAMBAR . . . . . **xi**
DAFTAR LAMBANG . . . . . **xiii**
ABSTRAK . . . . . **xiv**
ABSTRACT . . . . . **xv**
I PENDAHULUAN **1**
1.1. Latar Belakang Masalah 1
1.2. Rumusan Masalah 10
1.3. Tujuan Penelitian 10
1.4. Manfaat Penelitian 11
1.5. Batasan Masalah 11
1.6. Sistematika Penulisan 12
II TINJAUAN PUSTAKA **14**
2.1. *Vehicle Routing Problem* (VRP) 14
2.2. *Vehicle Routing Problem with Time Window* (VRPTW) 17
2.3. Optimasi 22
2.4. *Particle Swarm Optimization* (PSO) 26
2.4.1. Dasar Algoritma *Particle Swarm Optimization* (PSO) 26
2.4.2. *Personal Best Particle Swarm Optimization* (PSO) 29

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.4.3. *Global Best Particle Swarm Optimization* (PSO) 30
2.4.4. *Update Velocity* (Kecepatan) dan *Update* Posisi 31
2.4.5. Parameter Algoritma *Particle Swarm Optimization* 32
2.4.6. Tahapan Algoritma *Particle Swarm Optimization* 39
III METODE PENELITIAN 42
3.1. Jenis Penelitian 42
3.2. Data Penelitian 43
3.3. Perancangan Persoalan VRPTW 44
3.4. Perancangan Algoritma *Particle Swarm Optimization* (PSO) 46
3.4.1. Tahapan dan Proses Algoritma *Particle Swarm Optimization* 48
IV HASIL DAN PEMBAHASAN 51
4.1. Implementasi Algoritma PSO dalam Proses Perhitungan Secara Manual Terhadap Persoalan VRPTW 51
4.2. Analisis Hasil Perhitungan *Algoritma Particle Swarm Optimization* (PSO) 72
4.3. Perbandingan Hasil Algoritma PSO Berdasarkan Strategi *Inertia Weight* Terhadap Persoalan VRPTW 106
V PENUTUP 109
5.1. Simpulan 109
5.2. Saran 111
DAFTAR PUSTAKA 112
A SKRIP PROGRAM MATLAB 119
B TABEL DATA BENCHMARK SOLOMON 128
C TABEL HASIL PERCOBAAN ALGORITMA PSO 130
D TABEL SOLUSI OPTIMAL PERSOALAN VRPTW 154

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR TABEL

2.1 Strategi *Inertia Weight* 37

3.1 Data Random Cluster 101 (RC101) dengan 3 Titik 44

4.1 Data Benchmark Solomon Random Cluster 101 (RC101) 52

4.2 Data Jarak Antar Pelanggan 54

4.3 Data Waktu Tempuh Antar Pelanggan 56

4.4 Posisi Awal Partikel Pada Setiap *Swarm* 57

4.5 Hasil Inisialisasi Kecepatan Awal Partikel Pada Setiap *Swarm* 58

4.6 Set Parameter Algoritma PSO 73

4.7 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Constant Inertia Weight* 75

4.8 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Random Inertia Weight* 80

4.9 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Linear Decreasing Inertia Weight* 85

4.10 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Chaotic Inertia Weight* 91

4.11 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Global-Local Inertia Weight* 96

4.12 Set Parameter *Constriction Coefficient* 100

4.13 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Constriction Coefficient* 102

4.14 Hasil Percobaan Terbaik Penerapan Algoritma PSO 107

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR GAMBAR

2.1 Contoh Representasi Graf Berdasarkan Definisi VRP 16
2.2 Diagram Klasifikasi Metode Optimisasi 25

3.1 *Flowchart* algoritma PSO 46
3.2 *Flowchart* Pbest dan Gbest PSO 47
3.3 *Flowchart* Tahapan Penelitian 50

4.1 Grafik Analisis Berdasarkan Jumlah Swarm 76
4.2 Grafik Analisis Berdasarkan Iterasi Maksimum 77
4.3 Grafik Analisis Berdasarkan *Learning rates* 78
4.4 Ilustrasi Solusi Optimal Percobaan ke 193 79
4.5 Grafik Analisis Berdasarkan Jumlah Swarm 81
4.6 Grafik Analisis Berdasarkan Jumlah Iterasi 82
4.7 Grafik Analisis Berdasarkan Learning rates 83
4.8 Ilustrasi Solusi Optimal Percobaan ke 150 84
4.9 Grafik Analisis Berdasarkan Jumlah Swarm 86
4.10 Grafik Analisis Berdasarkan Jumlah Iterasi 87
4.11 Grafik Analisis Berdasarkan Learning rates 88
4.12 Ilustrasi Solusi Optimal Percobaan ke 198 89
4.13 Grafik Analisis Berdasarkan Jumlah Swarm 92
4.14 Grafik Analisis Berdasarkan Jumlah Iterasi 93
4.15 Grafik Analisis Berdasarkan Learning rates 94
4.16 Ilustrasi Solusi Optimal Percobaan ke 114 95
4.17 Grafik Analisis Berdasarkan Jumlah Swarm 97
4.18 Grafik Analisis Berdasarkan Jumlah Iterasi 98
4.19 Grafik Analisis Berdasarkan Learning rates 99
4.20 Ilustrasi Solusi Optimal Percobaan ke 18 100
4.21 Grafik Analisis Berdasarkan Jumlah Swarm 103

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.22 Grafik Analisis Berdasarkan Jumlah Iterasi 104
4.23 Grafik Analisis Berdasarkan Parameter c_1 dan c_2 104
4.24 Grafik Analisis Berdasarkan Parameter ω 105
4.25 Ilustrasi Solusi Optimal Percobaan ke 50 106

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR LAMBANG

$\sum_{i=1}^{n} x_i$	: penjumlahan $x_1 + x_2 + \cdots + x_n$
$\imath \in N$	: i anggota himpunan N
$j \in D$	: j anggota himpunan D
$t \in T$	: t anggota himpunan T
$U(0,1)$	: distribusi *uniform* (random nilai 0 sampai 1)
ω	: bobot inersia
χ	: *constiction coefficient*
φ	: parameter *learning rates*
$\varepsilon \in [0,1]$	: ε bernilai random dari 0 sampai 1

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Distribusi merupakan bentuk aspek kegiatan yang bermanfaat dalam sektor ekonomi dan sering dilakukan oleh perusahaan. Secara umum, distribusi adalah pergerakan dari komoditas barang dan jasa dari satu tempat menuju tempat yang lainnya dalam *supply chain management* dan juga meliputi transportasi (pengangkutan) serta pergudangan. Distribusi barang yang efektif akan memperlancar penyaluran barang ke konsumen sehingga memberikan dampak positif seperti kemudahan mendapatkan barang atau produk tersebut, serta bertambahnya keuntungan bagi perusahaan (Fuadiyah, 2018). Berdasarkan hal tersebut, saluran distribusi yang efektif dan efisien telah menjadi salah satu aset paling penting yang diperlukan oleh suatu perusahaan (Zhang et al., 2017).

Dalam perspektif Islam dijelaskan terkait distribusi yang tercantum pada QS. Al-Hasyar ayat 7.

مَّآ أَفَآءَ ٱللَّهُ عَلَىٰ رَسُولِهِۦ مِنْ أَهْلِ ٱلْقُرَىٰ فَلِلَّهِ وَلِلرَّسُولِ وَلِذِى ٱلْقُرْبَىٰ وَٱلْيَتَـٰمَىٰ
وَٱلْمَسَـٰكِينِ وَٱبْنِ ٱلسَّبِيلِ كَىْ لَا يَكُونَ دُولَةًۢ بَيْنَ ٱلْأَغْنِيَآءِ مِنكُمْ ۚ وَمَآ ءَاتَىٰكُمُ
ٱلرَّسُولُ فَخُذُوهُ وَمَا نَهَىٰكُمْ عَنْهُ فَٱنتَهُوا۟ ۚ وَٱتَّقُوا۟ ٱللَّهَ ۖ إِنَّ ٱللَّهَ شَدِيدُ ٱلْعِقَابِ ﴿٧﴾

Artinya : ” *Apa saja harta rampasan (fai-i) yang diberikan Allah kepada RasulNya (dari harta benda) yang berasal dari penduduk kota-kota Maka adalah*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

untuk Allah, untuk rasul, kaum kerabat, anak-anak yatim, orang-orang miskin dan orang-orang yang dalam perjalanan, supaya harta itu jangan beredar di antara orang-orang Kaya saja di antara kamu. apa yang diberikan Rasul kepadamu, Maka terimalah. dan apa yang dilarangnya bagimu, Maka tinggalkanlah. dan bertakwalah kepada Allah. Sesungguhnya Allah amat keras hukumannya ".

Pada QS. Al-Hasyar ayat 7 menjelaskan makna distribusi dalam perspektif Islam seperti yang tercantum pada Tafsir Ayat-Ayat Ekonomi Al-Qur'an. Berkenaan dengan tafsir ayat tersebut, M. Quraish Shihab menjelaskan harta rampasan (*fa'i*) yang diperoleh dari musuh tanpa melalui peperangan. Oleh karena itu, kata *ma afa,allahu 'ala rasulihi* yang terdapat pada ayat di atas berarti "apa saja yang telah dikembalikan oleh Allah kepada Rasulnya, maka kemudian akan diedarkan kepada orang-orang yang beriman sehingga Allah telah mengembalikannya kepada pemilik yang sah. Sedangkan, substansi distribusi menurut Afzalur Rahman ada dua, yaitu pemerataan dan adil. Berdasarkan penjelasan ayat pada QS. Al Hasyar ayat 7 di atas maka makna distribusi telah ada sejak zaman Nabi yang disebut dengan distribusi kekayaan dan dalam kaitannya terhadap ekonomi distribusi yang efektif dan efisien memaksimalkan bagaimana cara menyaluran barang kepada konsumen secara merata (Tarigan, 2012).

Aspek yang diperhatikan dalam kegiatan distribusi barang adalah bagaimana sistem atau mekanisme menyalurkan barang ke sejumlah pelanggan dengan tepat (Hutomo dan Sari, 2017). Bagian terpenting dalam kegiatan distribusi adalah pada transportasinya. Transportasi membutuhkan biaya yang cenderung tinggi. Selain sarana transportasi dalam kegiatan distribusi, perusahaan juga harus mampu menyusun penjadwalan yang efektif dan rute pengiriman yang tepat sehingga dapat meminimalkan jarak dan waktu tempuh. Oleh karena itu,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dibutuhkan tindakan yang disebut dengan optimasi (Lawrence, 2008). Optimasi atau optimisasi adalah suatu proses untuk menemukan solusi penyelesaian yang layak dalam hal melakukan upaya secara tepat, efektif, dan efisien (Alif, 2014). Optimasi juga merupakan suatu tindakan tersistematis yang dapat dilakukan untuk menemukan solusi terbaik. Optimasi berperan dalam berbagai bidang diantaranya bidang perdagangan, transportasi, ekonomi, arsitektur, telekomunikasi, jaringan komputer dan lain-lain (Budi, 2013).

Optimasi secara abstrak dapat digambarkan dalam ilmu matematika. Secara umum optimasi dibagi menjadi dua jenis yang meliputi optimasi kombinatorial dan pemrograman matematika. Optimasi kombinatorial berhubungan dengan teori kompleksitas komputasi, teori algoritma, dan riset operasi (Moengin, 2011). Optimasi terkait masalah rute terpendek (*The Shortest Route Problem*) dapat dijelaskan dalam riset operasi yang berhubungan dengan model arus jaringan. Komponen model arus jaringan pada riset operasi menggunakan teori graf sebagai dasarnya. Teori graf merupakan salah satu cabang dari ilmu matematika yang berhubungan dengan masalah graf (Putra, 2015). Graf merupakan himpunan yang terdiri atas simpul (*vertex*) dan sisi (*edge*). Graf yang digunakan pada trasportasi terkait pencarian rute terpendek merupakan jenis graf lengkap berbobot dan tidak memiliki arah dengan simpul (*vertex*) mengasumsikan kota dan sisi (*edge*) mengasumsikan lintasan atau jalan yang menghubungkan antara kota satu dengan lainnya. Selain itu, dalam graf bobot diasumsikan sebagai biaya, dan jarak tempuh atau waktu perjalanan yang dibutuhkan oleh kendaraan(Fatmawati, 2015).

Secara garis besar, pada kegiatan distribusi barang dalam permasalahan transportasi terkait pencarian rute terpendek disebut dengan *Vehicle Routing Problem* (VRP) (Lawrence, 2008). *Vehicle Routing Problem* (VRP) adalah salah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

satu persoalan dalam riset operasi yang berkaitan dengan optimasi kombinatorial atau disebut juga dengan masalah dalam pemrograman integer (Zhang et al., 2017). *Vehicle Routing Problem* (VRP) mencakup dua masalah yaitu masalah penentuan rute atau *Travellling Salesman Problem* (TSP) dan masalah kapasitas atau *Bin Packing Problem* (BIN) sehingga termasuk ke dalam kategori *NP-Hard Problem* yang berarti waktu komputasi semakin rumit dan lama (Iskandar, 2010). *Travellling Salesman Problem* (TSP) merupakan masalah tentang menemukan rute terpendek mulai dari satu kota dan kembali ke kota yang sama. *Travelling Salesman Problem* (TSP) mempertimbangkan hanya satu yang melewati setiap kota (titik, *node* atau komponen) dimana jarak antara masing-masing kota tersebut telah diketahui. Pada TSP juga memiliki sebuah batasan (*constraint*), salah satunya adalah untuk setiap *node* atau titik hanya boleh dikunjungi tepat satu kali (Kaya, 2018).

Vehicle Routing Problem (VRP) merupakan masalah yang terkenal dalam penelitian operasional dimana permintaan dari pelanggan akan dipasok oleh satu atau beberapa depot (pusat distribusi). Tujuan dari *Vehicle Routing Problem* (VRP) adalah untuk menemukan serangkaian rute pengiriman yang memenuhi beberapa persyaratan atau kendala dan memberikan total biaya yang minimal (Hadhiatma , 2017). VRP telah menarik minat yang sangat besar dari banyak peneliti terutama pada penelitian riset operasi dan komputasi karena perannya dalam perencanaan sistem distribusi dan logistik di banyak sektor. Beberapa contoh permasalahan terkait VRP seperti pengumpulan sampah, pengiriman surat atau barang, pengurutan tugas, dan lain sebagainya. *Vehicle Routing Problem* (VRP) juga memiliki berbagai tipe dalam penyelesaian permasalahan. Tipe-tipe tersebut meliputi *Vehicle Routing Problem with Time Window* (VRPTW), VRP dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pick-Up dan *Delivery*, *Capasitated Vehicle Routing Problem* (CVRP) dan lain sebagainya.

Pada penelitian ini digunakan tipe *Vehicle Routing Problem with Time Window* (VRPTW). Secara umum, *Vehicle Routing Problem with Time Window* (VRPTW) merupakan tipe atau variasi dari VRP yang masalah pada rute kendaraannya meliputi, setiap klien atau *customer* memiliki batasan waktu selama permintaannya harus dipenuhi, tujuannya adalah untuk meminimalkan jumlah kendaraan yang digunakan dan total jarak tempuh sembari mempertimbangkan batasan waktu (Belachgar, 2017). *Vehicle Routing Problem with Time Window* (VRPTW) merupakan turunan dari *Capasitated Vehicle Routing Problem* (CVRP) dimana *time window* yang dihubungkan pada setip customer dan pusat distribusi atau depot (Maryati et al., 2018). Karena tingkat kompleksitas yang tinggi dari VRPTW dan penerapannya yang luas pada situasi kehidupan nyata, dibutuhkan teknik yang mampu menghasilkan solusi berkualitas tinggi dalam kurun waktu yang terbatas (Omar et al, 2018).

Metode optimisasi yang dapat digunakan dalam permasalahan *Vehicle Routing Problem* (VRP) dibagi menjadi dua kelompok yaitu metode eksak (*exact method*) dan teknik pendekatan (*approximate method*). Pada metode eksak contohnya adalah metode *branch and bound* dan metode *dynamic programming*. Sedangkan pada teknik pendekatan terbagi lagi menjadi metode pendekatan dan algoritma heuristik. Algortima heuristik terbagi menjadi dua jenis yaitu spesifik heuristik (biasanya disebut dengan heuristik saja) dan metaheuristik. Contoh pada spesifik heuristik yaitu metode *saving based*, metode *matching based*, metode *multoroute improvement heuristic* dan metode *nearest neighborhood*. Sedangkan contoh pada metaheuristik yaitu meliputi algoritma pencarian lokal (*local search*),

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Algoritma Genetika (*Genetic Algorithm*), *Simulated Annealing*, *Tabu Search*, Algoritma Koloni Semut (*Ant Colony Algoritm*), dan *Particle Swarm Optimisation* (PSO) (Lawrence, 2008).

Pada penelitian ini menggunakan teknik pendekatan dengan metaheuristik karena pada metode eksak membutuhkan waktu yang lama untuk memperoleh solusi optimal. Metaheuristik merupakan suatu teknik pendekatan bersifat "*problem independent*" yang berarti dapat diimplementasikan diberbagai macam permasalahan yang terkait dengan masalah optimisasi. Selain itu, metaheuristik didefinisikan sebagai salah satu variasi metode yang didasarkan pada teknik pendekatan algoritma heuristik. Berdasarkan hal tersebut, maka metaheuristik dapat diimplementasikan dalam persoalan *Vehicle Routing problem with Time Window* (VRPTW) dimana tingkat kompleksitas permasalahan yang tinggi dari persoalan tersebut.

Beberapa penelitian dalam persoalan VRPTW diantaranya dilakukan oleh Y.W Zhao dkk (2004) berjudul "*Particle Swarm Optimization for Vehicle Routing Problem with Time Windows*". Pada penelitian tersebut menunjukkan bahwa penerapan algoritma PSO dapat memperoleh solusi yang baik untuk masalah VRPTW. Namun penelitian ini memodifikasi PSO dengan banyak aspek diantaranya GN-PSO, GS-PSO, GI-PSO, LN-PSO, LS-PSO, dan LI-PSO. Metodologi ini berguna untuk memecahkan VRPTW secara *real time* untuk persoalan VRP dan penjadwalan secara dinamis. Hasil penelitian ini menunjukkan bahwa versi global PSO (GN-PSO) lebih baik dibandingkan dengan versi lokal PSO (LN-PSO) (Zhao et al., 2004).

Nur Adhita Rahmawati (2012) juga melakukan penelitian dengan judul "Implementasi *Particle Swarm Optimisation* (PSO) pada *Vehicle Routing Problem*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

with Time Window (VRPTW)". Hasil penelitian tersebut yaitu penerapan dari algortima PSO pada kasus VRPTW mampu memberikan solusi yang mendekati optimal. Algoritma PSO menghasilkan *Relative Percentage Deviation* (RPD) rata-rata presentasenya kurang dari 5,3% dengan menggunakan dataset sejumlah 25 *node* (Rahmawati, 2012).

Penelitian lainnya yang juga dilakukan oleh Yong Zhang dkk (2017) yang berjudul " *Analysis of an Automated Vehicle Routing Problem in Logistics Considering Path Interruption*". Pada penelitian tersebut masalah perutean kendaraan otomatis digunakan model *Automated Vehicle Routing Problem with Time Window* (AVRPTW) dengan salah satu metode metaheuristik yaitu Algoritma *Particle Swarm Optimisation* (PSO). Hasil penelitiannya menjelaskan bahwa total biaya minimum yang dikeluarkan adalah 11.328. Selain itu, total kendaraan yang digunakan adalah empat kendaraan otomatis dengan presentase kapasitas optimumnya adalah 76%, 76%, 80% dan 100% untuk rute atau jalur berbeda. Oleh karena itu, hasil penelitian menunjukkan bahwa sumber daya transportasi yang dimiliki perusahaan telah digunakan secara maksimal (Zhang et al., 2017).

Pada penelitian ini, penulis akan menggunakan persoalan mengenai permasalahan benchmark Solomon terkait *Vehicle Routing Problem with Time Window*(VRPTW). Permasalahan benchmark Solomon tersebut dikembangkan oleh Profesor Solomon untuk pencarian rute kendaraan dengan batasan jendela waktu yang disebut dengan VRPTW. Marius M. Solomon adalah seorang Profesor di Kelompok Informasi, Riset Operasi dan Analisis di Northeastern University, Boston, Massachusetts. Profesor Solomon memiliki minat penelitian dan konsultasi dalam pengembangan dan implementasi metodologi desain, perencanaan, dan eksekusi yang mengarah pada peningkatan ketangkasan rantai

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

pasokan. Fokusnya adalah pada pengembangan pendekatan optimisasi untuk aspek sensitif waktu dari rantai pasokan yang menguntungkan berbagai sektor industri, publik dan layanan. Profesor Solomon telah menerbitkan banyak penelitian yang berhubungan dengan perencanaan armada kendaraan dengan batasan jendela waktu (*time window*), pencarian rute dan penjadwalan (*scheduling*), desain rantai pasokan, serta manajemen sistem manufaktur dan pergudangan yang maju secara teknologi (Solomon, 2005).

Penelitian terkait permasalahan benchmark Solomon telah dilakukan oleh Niklas Kohl dkk (1999) dengan judul "*2-Path Cuts for the Vehicle Routing Problem with Time Window*". Penelitian tersebut terdapat perpotongan 2 jalur menghasilkan batas bawah yang lebih baik untuk masalah rute kendaraan dengan (*time window*). Selain itu batasan jumlah kendaraan mengharuskan setiap pelanggan untuk dilayani. Subproblem dalam masalah jalur terpendek adalah batasan waktu (*time window*) dan kapasitas kendaraan. Dalam penelitian ini menggunakan algoritma *branch and bound* untuk mendapatkan solusi optimal. Algoritma tersebut telah diimplementasikan dan diuji pada masalah VRPTW hingga 100 pelanggan dari dataset benchmark Solomon. Hasilnya menunjukkan bahwa algoritma tersebut dapat menyelesaikan beberapa masalah yang sebelumnya tidak terpecahkan dan mencapai optimalitas hingga 150 pelanggan (Niklas Kohl et al., 1999).

Penelitian lainnya terkait VRPTW pada permasalahan benchmark Solomon juga dilakukan oleh Olli Braysy dan Michael Gendreu (2005) dengan judul "*Vehicle Routing Problem with Time Window, Part II : Metaheuristics*". Penelitian tersebut mensurvei tentang penggunaan salah satu metode dari teknik pendekatan metaheuristik yaitu algoritma *Tabu search* dengan tujuan untuk menyelesaikan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

persoalan *Vehicle Routing Problem with Time Window* (VRPTW). VRPTW dapat didefinisikan sebagai suatu permasalahan dalam merancang rute dengan biaya paling rendah dari satu depot ke satu set titik yang tersebar secara geografis. Rute harus dirancang untuk setiap titik dikunjungi tepat satu kali oleh satu kendaraan dalam interval waktu tertentu; semua krndaraan dengan rutenya dimulai dan kembali lagi di depot, dan total tuntutan untuk setiap titik pada satu rute tertentu tidak boleh melebihi kapasitas kendaraan (Olli Bräysy dan Michel Gendreau, 2005).

Berdasarkan penjabaran diatas mengenai beberapa penelitian tentang *Vehicle Routing Problem* (VRP), maka penulis akan melakukan penelitian pada permasalahan terkait pencarian rute atau jalur terpendek menggunakan algoritma *Particle Swarm Optimization* (PSO). Algoritma *Particle Swarm Optimization* (PSO) adalah salah satu teknik pendekatan metaheuristik yang dapat digunakan untuk menyelesaikan permasalahan transportasi terkait masalah pencarian rute terpendek. Algoritma tersebut dipilih karena memiliki sifat independen yang dapat diimplementasikan pada permasalahan VRP. Algoritma PSO didasarkan pada *intelligence* yang dapat diterapkan baik dalam penelitian ilmiah mauoun teknik. Selain itu, perhitungan dari algoritma PSO cukup sederhana dibandingkan dengan perhitungan pada algoritma lainnya serta algoritma PSO memiliki tingkat optimasi yang tinggi (Zhang et al., 2017).

Pada penelitian ini juga akan dimodifikasi berdasarkan parameter strategi *inertia weight*. Modifikasi parameter strategi *inertia weight* pertama kali diperkenalkan oleh Y.Shi dan Eberhart (2002) yang disebut dengan *constant inertia weight*. Parameter *inertia weight* kemudian berkembang dengan berbagai macam bentuk persamaan yang digunakan untuk menghitung nilai *inertia weight*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tersebut. Penelitian ini menggunakan modifikasi strategi *inertia weight* terbaik berdasarkan kriteria tertentu dari penelitian Bansal (2011). Penelitian ini menambahkan kendala batasan waktu (*time window*) yang disebut dengan *Vehicle Routing Problem with Time Window* (VRPTW). Pada kasus model VRPTW ini selain memperhatikan masalah kapasitas kendaran, masalah *time window* juga harus dipenuhi. Oleh karena itu, penerapan algoritma *Particle Swarm Optimization* (PSO) dengan modifikasi parameter *inertia weight* akan digunakan dalam menyelesaikan model *Vehicle Routing Problem with Time Window* (VRPTW).

1.2. Rumusan Masalah

1. Bagaimana penerapan algoritma *Particle Swarm Optimization* (PSO) untuk menyelesaikan persoalan *Vehicle Routing Problem with Time Window* (VRPTW) terhadap permasalahan terkait penentuan rute terpendek pada distribusi barang dengan batasan *time window*?

2. Bagaimana analisis hasil berdasarkan modifikasi parameter strategi *inertia weight* pada penerapan algoritma *Particle Swarm Optimization* (PSO) terhadap persoalan *Vehicle Routing Problem with Time Window* (VRPTW)?

3. Bagaimana perbandingan hasil dari algoritma *Particle Swarm Optimization* (PSO) berdasarkan modifikasi strategi *inertia weight*?

1.3. Tujuan Penelitian

Tujuan yang diharapkan pada penelitian ini adalah untuk memperoleh rute pengiriman barang yang optimal berdasarkan penerapan dari Algoritma *Particle Swarm Optimization* (PSO) untuk menyelesaikan permasalahan *Vehicle Routing Problem with Time Window* (VRPTW), menganalisis hasil, dan melakukan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

perbandingan hasil yang diperoleh berdasarkan penggunaan modifikasi parameter strategi *inertia weight* pada algoritma *Particle Swarm Optimization* (PSO).

1.4. Manfaat Penelitian

Manfaat penelitian yang diharapkan pada penelitian ini diantaranya adalah sebagai berikut.

1. Secara teoritis, penelitian ini dapat digunakan sebagai penambah wawasan, ilmu pengetahuan, sumber referensi, studi literatur, dan tinjauan pustaka bagi pembaca atau referensi perbandingan untuk penelitian serupa.
2. Secara praktis, penelitian ini diharapkan dapat memberikan rujukan dalam menentukan rute pengiriman paling optimal berdasarkan penerapan dari Algoritma *Particle Swarm Optimization* (PSO) dengan modifikasi *inertia weight* untuk menyelesaikan persoalan *Vehicle Routing Problem with Time Window* (VRPTW).

1.5. Batasan Masalah

Batasan masalah pada penelitian ini digunakan untuk penyelesaian masalah yang akan diteliti menjadi lebih terarah dan terpusat sehingga penelitian dapat dilakukan sesuai rencana dan memberikan hasil yang optimal. Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan dengan menerapkan salah satu algoritma metaheuristik yaitu Algoritma *Particle Swarm Optimization* (PSO).
2. Model atau kasus permasalahannya adalah *Vehicle Routing Problem with Time Window* (VRPTW).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3. Batasan kendalanya yaitu mencakup kapasitas kendaraan ditambah dengan batasan jendela waktu (*time window*) yang meliputi *earlist time* (waktu awal) dan *latest time* (waktu akhir) pada setiap pelanggan.

4. Data yang digunakan merupakan permasalahan dari benchmark Solomon.

5. Parameter modifikasi algoritma PSO dengan *inertia weight* yang digunakan meliputi *constant inertia weight*, *random inertia weight*, *linear decreasing inertia weight*, *chaotic inertia wight*, *global-local inertia weight*, dan *constriction coefficient inertia weight*.

1.6. Sistematika Penulisan

Pada penelitian ini sistematika penulisan digunakan sebagai gambaran secara menyeluruh, dan secara garis besar dapat dituliskan sebagai berikut:

BAB I : PENDAHULUAN

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

BAB II : TINJAUAN PUSTAKA

Pada bab II tinjauan pustaka berisi bahan referensi untuk teori yang terkait persoalam optimasi, *Travelling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP), *Vehicle Routing Problem with Time Window* (VRPTW)dan Algoritma *Particle Swarm Optimization* (PSO).

BAB III : METODE PENELITIAN

Pada bab III metode penelitian berisi tentang jenis penelitian, jenis dan sumber data penelitian, pemodelan trkait persoalan *Vehicle Routing Problem with Time Window* (VRPTW), perancangan sistem terdiri dari deskripsi sistem, analisa kebutuhan sistem. Selain itu, perancangan dari algoritma *Particle Swarm*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Optimization (PSO) yang meliputi algoritma PSO, tahapan metode algoritma PSO, dan yang terakhir terdapat tahapan penelitian secara menyeluruh serta diagram alur (*flowchart*) penelitian dari algoritma PSO pada model VRPTW.

BAB IV: HASIL DAN PEMBAHASAN

Pada bab IV hasil dan pembahasan menjelaskan tentang implementasi algoritma PSO dalam proses perhitungan secara manual terhadap persoalan VRPTW, analisis hasil perhitungan algoritma PSO yang meliputi penjelsan set parameter yang digunakan dalam penelitian dan analisis hasil algoritma PSO berdasarkan strategi *inertia weight*. Strategi *inertia weight* tersebut meliputi *constant inertia weight*, *random inertia weight*, *linear decreasing inertia weight*, *chaotic inertia wight*, *global-local inertia weight*, dan *constriction coefficient inertia weight*.

BAB V : PENUTUP

Pada bab V berisi tentang kesimpulan dari hasil penelitian dan saran yang membangun setelah mengimplementasikan dan menganalisis algoritma PSO terhadap persoalan VRPTW.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB II

TINJAUAN PUSTAKA

Topik yang akan dibahas dalam penelitian ini adalah permasalahan transportasi terkait pencarian rute terpendek dengan model *Vehicle Routing Problem with Time Window* (VRPTW) menggunakan metode optimasi. Dalam hal ini, metode optimasi yang digunakan adalah dengan teknik pendekatan metaheuristik salah satunya algoritma *Particle Swarm Optimization*.

2.1. *Vehicle Routing Problem* (VRP)

Vehicle Routing Prolem (VRP) pertama kali ditemukan pada tahun 1959 oleh seorang ilmuwan bernama Dantzig dan Ramser. Dalam penelitian yang ditulis oleh Dantzig dan Ramser tersebut, rumusan permasalahannya adalah bagaimana memaksimalkan kapasitas kendaraan truk dan meminimalkan jarak tempuh yang dicakup oleh armada kendaraan. Permasalahan tersebut kemudian berkembang pada tahun 1960 menjadi masalah yang penting dalam bidang distribusi barang dan logistik gama. *Vehicle Routing Problem* (VRP) adalah suatu masalah yang berkaitan dengan optimasi kombinatorial atau disebut juga dengan masalah dalam pemrograman integer (Zhang et al., 2017).

Dantzig dan Ramser menggambarkan bagaimana VRP dapat dianggap sebagai generalisasi dari masalah salesman keliling atau yang disebut dengan *Travelling Salesman Problem* (TSP). Dalam buku Hoffman dan Wolfe menggambarkan pentingnya TSP yang termasuk kedalam permasalahan optimasi kombinatorial. Dantzig bersama dengan ilmuwan Fulkerson dan Johnson

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

mengembangkan pendekatan yang tepat untuk persoalan TSP. Dalam tulisan tersebut berisi tentang beberapa hal diantaranya sebagai berikut.

1. Setiap *tour* vektor akan diberi indeks pada setiap jalur antar kota. Sebuah vektor akan bernilai 1 jika jalan antara sepasang kota dilalui atau dilewati, dan akan bernilai 0 jika tidak dilewati. Selain itu, dalam representasi semua *tour* maka persamaan linier pada setiap kota hanya akan dikunjungi tepat satu kali.

2. Fungsi objektif linier menyatakan biaya *tour* sebagai jumlah jarak tempuh dari antar kota secara berturut-turut dalam *tour*.

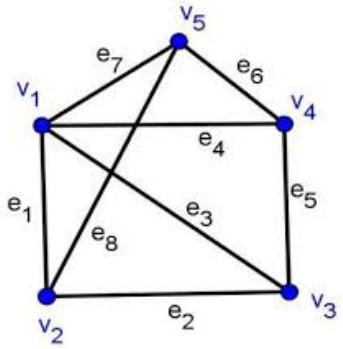
3. Masalah pemrograman linier dari nilai integer tersebut merupakan masalah pemrograman dengan mengidentifikasi batasan linier tambahan pada vektor untuk memastikan bahwa akan diperoleh nilai minimum yang diasumsikan oleh beberapa *tour*. Hal tersebut mengarah pada pengenalan eliminasi *subtour* fungsi kendala, dengan pengecualian solusi dimana kota-kota hanya akan dikunjungi tepat sekali, tetapi dalam satu set terputus *subtour*. Namun, ditunjukkan juga bahwa ada jenis kendala lain yang kadang harus ditambahkan selain kendala eliminasi *subtour* untuk mengecualikan solusi yang melibatkan vektor entri fraksional(Belachgar, 2017).

Vehicle Routing Problem (VRP) merupakan jenis permasalahan dalam bidang distribusi atau penyaluran barang untuk menentukan himpunan dari rute-rute yang harus dilalui oleh suatu kendaraan. Pada setiap kendaraan tersebut memulainya dari pusat distribusi (*distribution center*) yang disebut juga dengan depot, kemudian melakukan pelayanan kepada konsumen atau pelanggan, dan akan kembali pada pusat distribusi (*distribution center*) atau depot yang sama.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tujuan dari penyelesaian VRP adalah menemukan lintasan yang dapat melewati semua titik agar memperoleh rute yang optimal (Gamayanti et al., 2015).

Dalam VRP klasik dapat didefinisikan sebagai berikut: Misalkan $G = (V, E)$ merupakan suatu graf G dimana $V = 1, 2, 3, ..., n$ adalah himpunan simpul yang mewakili kota dengan pusat distribusi (depot), dan E adalah himpunan sisi (himpunan yang mungkin kosong) dengan setiap sisi $(i, j) i \neq j$ dikaitkan dengan matriks jarak tidak negatif $C = (c_{ij})$. Pada beberapa konteks, c_{ij} dapat diartikan sebagai biaya perjalanan atau sebagai waktu perjalanan. Ketika C simetris, maka sangat mudah untuk mengganti nilai elemen e dengan set E dari sisi yang tidak terarah (Laporte, 1992). VRP yang digambarkan sebagai digraf terhubung terdiri dari satu set n + 1 simpul. Setiap simpul atau *node* ke - i akan meminta pelayanan untuk setiap permintaan pelanggan (*demand*) yang dinotasikan dengan q_i, kecuali *node* untuk depot atau pusat distribusi $q_i = 0$ (Belachgar, 2017). Berikut ini merupakan contoh gambar graf sederhana sebagai dasar untuk direpresentasikan sesuai dengan definisi VRP klasik.



Gambar 2.1 Contoh Representasi Graf Berdasarkan Definisi VRP

Berdasarkan gambar di atas menjelaskan bahwa himpunan E direpresentasikan sebagai jalur dan himpunan V direpresentasikan sebagai kota dalam model VRP. VRP terdiri dari merancang serangkaian rute kendaraan paling

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

optimal sehingga :

1. Setiap kota di V akan dikunjungi tepat satu oleh satu kendaraan.
2. Semua rute kendaraan dimulai dan berakhir pada pusat distribusi (depot).
3. Memenuhi beberapa kendala lainnya, seperti jendela waktu (*time window*).

Vehicle Routing Problem (VRP) memiliki beberapa tipe atau jenis permasalahan yang berbeda-beda sesuai dengan kendala dan batasan masing-masing. Jenis-jenis VRP tersebut yang terkenal dan secara umum banyak persoalannya diantaranya adalah *Capacitated Vehicle Routing Problem* (CVRP) mencakup kendala batasan kapasitas dari kendaraan, *Vehicle Routing Problem with Time Window* (VRPTW) dengan batasan kapasitas dan jendela waktu (*time window*), *Multi Depot Vehicle Routing Problem with Time Window* (MDVRPTW) dengan batasan depot yang lebih dari satu dan jendela waktu (*time window*), dan masih banyak jenis VRP yang lainnya (Belachgar, 2017).

2.2. *Vehicle Routing Problem with Time Window* (VRPTW)

Vehicle Routing Problem with Time Window(VRPTW) merupakan generalisasi dari VRP yang terkenal dan merupakan turunan dari tipe CVRP. Ini dapat ditinjau sebagai gabungan masalah rute dan penjadwalan kendaraan yang sering muncul di banyak aplikasi dalam dunia nyata. VRPTW merupakan persoalan dengan tujuan untuk mengoptimalkan penggunaan armada kendaraan dimana harus membuat sejumlah pemberhentian untuk melayani satu set pelanggan, dan menentukan pelanggan mana yang harus dilayani terlebih dahulu oleh setiap kendaraan serta bagaimana urutannya sehingga dapat meminimalkan biaya. Selain itu, VRPTW bergantung pada kapasitas kendaraan dan pembatasan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

waktu layanan (*time window*). Persoalan dari VRPTW melibatkan penugasan kendaraan untuk perjalanan sehingga biaya penugasan dan biaya rute yang sesuai akan minimal (Ellabibet al., 2002).

Vehicle Routing Problem (VRPTW) merupakan persoalan dengan satu pusat distribusi barang atau depot yang memiliki sejumlah kendaraan beserta kapasitasnya untuk melakukan pelayanan kepada sejumlah *customers* pada *node* atau titik terpisah, serta dengan memperhatikan permintaan *customers* dan batasan waktu (*time window*). Pada *time window* tersebut terdapat waktu awal yang disebut *earlist time* dan waktu akhir yang disebut *latest time*. *Time window* dimaksudkan untuk setiap pelanggan harus dilayani saat atau setelah waktu awal(*earlist time*) dan sebelum waktu akhir(*latest time*). Apabila kendaraan tiba di lokasi *customer* pada saat sebelum waktu awal(*earlist time*) maka akan menghasilkan waktu tunggu. Sedangkan, apabila kendaraan tiba di lokasi *customer* pada saat setelah waktu akhir(*latest time*) maka akan menghasilkan waktu keterlambatan. Selain itu, masing-masing kendaraan memiliki waktu pelayanan (*service time*) untuk melakukan pelayanan pada setiap *customers* (Rahmawati, 2012).

VRPTW telah menjadi subjek dari upaya penelitian intensif untuk pendekatan optimalisasi heuristik yang tepat. Survei awal teknik solusi untuk VRPTW dapat ditemukan di Golden and Assad (1986), Desrochers et al. (1988), dan Solomon dan Desrosiers (Chiang, Russell 1996). Fokus utama dalam Desrosiers et al. (1995) dan Cordeau et al. (2001) adalah teknik solusi yang tepat. Rincian lebih lanjut tentang metode yang tepat ini dapat ditemukan di Larsen (1999) dan Cook and Rich (1999). Karena tingkat kompleksitas yang tinggi dari VRPTW dan penerapannya yang luas untuk situasi kehidupan nyata, maka dibutuhkan teknik solusi yang mampu menghasilkan suatu solusi paling optimal.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Berikut ini merupakan model matematika dalam persoalan *Vehicle Routing Problem with Time Window* (VRPTW) mulai dari fungsi tujuan beserta fungsi kendalanya.

Fungsi Tujuan :

Min

$$\sum_{k=1}^{K}\sum_{j=1}^{n}\sum_{i=0}^{n} C_{ij}X_{ijk} \tag{2.1}$$

$$X_{ijk} = \begin{cases} 1, & \text{jika kendaraan k mengunjungi titik j setelah titik i, i} \neq \text{j} \\ 0, & \text{untuk yang lain} \end{cases}$$

dengan fungsi kendala :

$$\sum_{k=1}^{K}\sum_{i=1}^{n} X_{ijk} \;=\; 1, j = 1, 2, 3, ..., n. \tag{2.2}$$

$$\sum_{j=1}^{n}\sum_{i=0}^{n} q_i X_{ijk} \leq Q, k = 1, 2, 3, ..., K. \tag{2.3}$$

$$\sum_{i=0}^{n} X_{i0k} = 1, k = 1, 2, 3, ..., K. \tag{2.4}$$

$$\sum_{j=1}^{n} X_{0jk} = 1, k = 1, 2, 3, ..., K. \tag{2.5}$$

$$\sum_{i=0}^{n} X_{ijk} - \sum_{i=0}^{n} X_{ijk} = 0, j = 1, 2, 3, ..., n, k = 1, 2, 3, ..., K. \tag{2.6}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$t_i + f_i + t_{ij} - M(1 - X_{ijk}) \leq t_j \tag{2.7}$$

$$i = 1, 2, 3, ..., n; j = 1, 2, 3, ..., n; k = 1, 2, 3, ..., K.$$

$$X_{ijk} = \{0, 1\} \tag{2.8}$$

$$t_i \geq 0; i = 1, 2, 3, ..., n; k = 1, 2, 3, ..., K. \tag{2.9}$$

$$e_i \leq t_i \leq l_i; i = 1, 2, 3, ..., n. \tag{2.10}$$

Penjelasan untuk setiap persamaan di atas adalah sebagai berikut.

a. Persamaan 2.1 fungsi tujuan digunakan untuk meminimalkan biaya tetap, biaya transportasi, dan biaya pinalti.

b. Persamaan 2.2 mengasumsikan bahwa setiap konsumen hanya dilayani oleh satu kendaraan atau setiap pelanggan hanya dikunjungi tepat satu kali.

c. Persamaan 2.3 menunjukkan total permintaan setiap kota dalam rute yang dilalui oleh setiap kendaraan tidak melebihi kapasitas kendaraan (keterbatasan kapasitas masing-masing kendaraan).

d. Persamaan 2.4 mensyaratkan bahwa jumlah total kendaraan tidak dapat melebihi jumlah kendaraan yang dimiliki pusat distribusi (depot).

e. Persamaan 2.5 setiap rute awal dimulai dari pusat distribusi atau depot.

f. Persamaan 2.6 menunjukkan bahwa jumlah kendaraan yang memasuki lokasi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sama dengan jumlah kendaraan yang meninggalkan lokasi.

g. Persamaan 2.7 mendefinisikan jendela waktu atau *time window*.

h. Persamaan 2.8 mendefinisikan keputusan variabel sebagai nilai biner.

i. Persamaan 2.9 menunjukkan bahwa waktu tidak negatif.

k. Persamaan 2.10 mendefinisikan waktu tiba kendaraan hanya dapat dilayani sesuai dengan *time window*.

Keterangan notasi :

i, j : menunjukkan jumlah konsumen yang akan dilayani (jumlah node), dimana node 0 didefinisikan sebagai pusat distribusi.

k : menunjukkan jumlah kendaraan.

X_{ijk} : kendaraan k yang melayani setiap pelanggan j setelah mengunjungi pelanggan i.

Q : Kapasitas maksimum kendaraan.

q_i : permintaan(*demand*) pada setiap pelanggan i.

t_{ij} : waktu tempuh kendaraan dari pelanggan$i$ ke pelanggan j.

e_i : waktu paling awal (*earliest time*) untuk melakukan pelayanan pada pelanggan i.

l_i : waktu paling akhir (*latest time*) untuk melakukan pelayanan pada pelanggan i.

f_i : waktu pelayanan kepada pelanggan i.

t_i : waktu kedatangan kendaraan pada pelanggan i.

K : banyak kendaraan.

n : banyak pelanggan.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.3. Optimasi

Optimasi atau optimisasi merupakan suatu proses untuk menemukan solusi satu atau lebih penyelesaian yang layak dalam rangka melakukan upaya secara tepat, efektif, dan efisien (Alif, 2014). Pada optimasi dilakukan suatu tindakan tersistematis untuk menemukan solusi terbaik. Menurut Wardy I. S.(2007) menjelaskan bahwa tujuan umum yang akan dicapai pada optimasi tersebut adalah meminimumkan usaha atau dengan kata lain memaksimumkan manfaat yang diinginkan (Wardy, 2007). Oleh karena itu, optimasi didefinisikan sebagai proses untuk menemukan suatu kondisi minimum atau maksimum dari suatu kasus seperti dalam sebuah fungsi, peluang, maupun pencarian nilai lainnya (Budi, 2013).

Secara umum, metode optimisasi dinyatakan seperti dibawah ini sebagai contoh ketika kita mengambil fungsi minimal.

Min f(x)

dengan kendala C (x)

dimana,

f(.) : fungsi objektif atau fungsi tujuan

x : variabel vektor

C(.) : fungsi kendala atau constraint

Berdasarkan hal ini, fungsi kendala didefinisikan sebagai fungsi yang mengendalikan nilai variabel keputusan. Sedangkan, fungsi objektif dalam metode optimisasi dapat berupa fungsi maksimum atau fungsi minimum.

Dalam ilmu matematika optimasi mengacu pada suatu kasus untuk mencari nilai minimum atau maksimum dari suatu fungsi riil (Randhani, 2015). Optimasi berperan dalam berbagai bidang diantaranya bidang perdagangan, transportasi, ekonomi, arsitektur, telekomunikasi, jaringan komputer dan lain-lain (Budi, 2013).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Berikut ini merupakan permasalahan optimasi yang umum terjadi dalam kehidupan nyata yang meliputi: (i) Penentuan jumlah pekerja yang minimum untuk menekan pengeluaran biaya pekerja namun hasil produksi tetap maksimal, (ii)Menemukan lintasan atau rute terpendek, dan (iii) *Scheduling* atau penjadwalan rute kendaraan umum dengan tujuan agar semua lokasi bisa dijangkau.

Pada optimisasi memiliki tahapan pengambilan keputusan yang meliputi empat tahap, diantaranya adalah sebagai berikut.

1. Tahap merumuskan

   Pada tahap merumuskan memiliki fungsi untuk merumuskan suatu permasalahan, melakukan identifikasi masalah dari faktor-faktor internal maupun eksternal suatu permasalahan yang akan diselesaikan.

2. Tahap memodelkan

   Pada tahap memodelkan memiliki fungsi untuk memodelkan secara matematis. Dalam bidang optimisasi, untuk mencari model yang sesuai dengan permasalahan dapat dilakukan dengan cara mencari referensi terkait dan studi literatur mengenai model-model yang mirip dengan permsalahan yang akan diselesaikan.

3. Tahap mengoptimalkan

   Pada tahap mengoptimalkan digunakan metode optimisasi untuk menyelesaikan suatu permasalahan baik hasil penyelesaiannya optimal maupun sub-optimal (hasil penyelesaian yang dimaksud adalah penyelesaian permasalahan dengan model matematis yang dilakukan).

4. Tahap mengimplementasikan Pada tahap mengimplementasikan atau tahap penerapan dilakukan uji solusi optimal yang diperoleh yaitu dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menerapkannya dalam suatu studi kasus atau kedalam permasalahan yang sesungguhnya (Talbi, 2009).

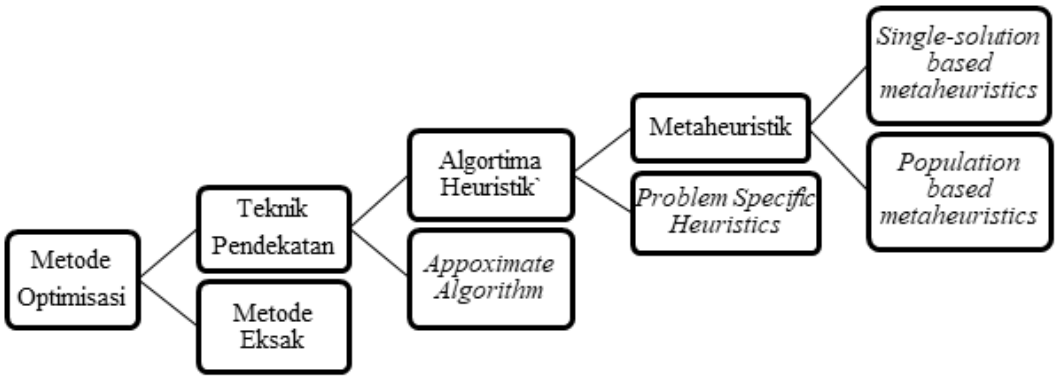
Secara garis besar, metode optimasi dibagi menjadi 2 kategori diantaranya adalah optimasi kombinatorial dan pemrograman dalam matematika (Budi, 2013). Optimasi kombinatorial adalah salah satu tipe optimasi yang seing digunakan dalam ilmu komputer teoritis dan matematika terapan dengan tujuan untuk mencari solusi biaya terkecil. Sebagai contoh permasalahan umum optimasi kombinatorial yaitu *Travelling Salesman Problem* (TSP), *Minimum Spanning Tree Problem*, *Vehicle Routing Problem* (VRP) dan lain sebagainya. Sedangkan, teknik pemrograman dalam matematika adalah suatu tipe optimasi dengan metode pencarian titik optimum. Kedua tipe metode optimasi tersebut merupakan bagian dari permasalahan pada riset operasi. Selain itu, metode optimasi juga memiliki peranan penting dalam bidang lainnya diantaranya bidang matematika, kecerdasan buatan, rekayasa perangkat lunak dan lain-lain (Moengin, 2011). Berdasarkan hal tersebut, pemecahan solusi optimasi dapat diselesaikan dengan berbagai macam metode yang telah dikembangkan untuk dapat menyelesaikannya (Budi, 2013).

Permasalahan pada optimasi kombinatorial salah satu contohnya adalah pada persoalan *Vehicle Routing Problem* (VRP) dengan tujuan mencari solusi nilai optimal. Pengertian dari solusi nilai optimal adalah suatu solusi yang diperoleh paling optimum dan menjadi solusi nilai terbaik. Pada VRP memiliki tujuan untuk mencari rute terpendek atau rute yang paling optimal. Solusi pemecahan pada permasalahan metode optimasi telah banyak dikembangkan diantaranya terbagi kedalam dua kelompok yaitu metode eksak (*exact method*) dan teknik pendekatan *(approximate method)* (Randhani, 2015).

Teknik pendekatan *(approximate method)* pada metode optimasi dibagi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menjadi dua yaitu algoritma heuristik dan algoritma pendekatan. Dalam algoritma heuristik terbagi lagi menjadi problem spesifik heuristik atau biasa disebut heuristik saja dan metaheuristik. Pada heuristik sebagai contohnya yaitu metode *nearest neighborhood*, *hill climbing*, *best first search* dan *comsoal*, sedangkan pada metaheuristik sebagai contohnya adalah algoritma genetika, algoritma koloni semut, algoritma kelelawar, *Particle Swarm Optimization* (PSO), *Cat Swarm Optimization* (CSO) dan lain sebagainya. Berikut ini merupakan diagram dari klasifikasi metode optimisasi.



Gambar 2.2 Diagram Klasifikasi Metode Optimisasi

Berbagai penerapan dari varian heuristik pada persoalan VRPTW dapat ditemukan di persoalan Solomon (1987), Potvin dan Rousseau (1993), Bramel dan Simchi-Levi (1996), dan Dullaert dan Braysy (2003). Pada jangka waktu kurang lebih 20 tahun, metaheuristik telah berkembang pesat karena metode tersebut mampu menyelesaikan permasalahan optimasi secara efektif dan efisien. Penerapan pada metode tersebut meliputi penjadwalan (scheduling), design (misalnya design pesawat, jembatan, dll), *Supply Chain Management* (SCM), manufaktur, dan hampir semua bidang optimasi lainnya. Metaheuristik merupakan metode yang banyak terinspirasi dari suatu fenomena alam. Seperti contoh pada

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

algoritma Particle Swarm Optimization (PSO) yang terinspirasi dari pergerakan sekawanan burung dan masih banyak lagi yang lainnya. Selain itu algoritma *Particle Swarm Opimization* (PSO) termasuk ke dalam jenis *population based metaheuristics* (Parsopoulos, 2015).

2.4. *Particle Swarm Optimization* (PSO)

Metode optimasi *Particle Swarm Optimization* (PSO) yang meliputi proses terbentuknya PSO, dasar dari algoritma PSO, parameter dalam algoritma PSO, dan langkah-langkah algoritma PSO dapat dijelaskan sebagai berikut.

2.4.1. Dasar Algoritma *Particle Swarm Optimization* (PSO)

Particle Swarm Optimization (PSO) pertama kali diperkenalkan oleh Dr. Eberhart dan Dr. Kennedy sejak tahun 1995. *Particle Swarm Optimization* (PSO) terinspirasi pada kejadian fenomena alam yang berawal dari pengamatan perilaku sekawanan burung dan ikan. Perilaku seekor burung dan ikan dalam kawanan tersebut disebut dengan swarm yang dipengaruhi oleh perilaku sosial dari organisme. Pengertian dari perilaku sosial yang dimaksud adalah perilaku atau tindakan individu dan juga tindakan kelompoknya (Kennedy dan Eberhart, 1995). Dalam algoritma *Particle Swarm Optimization* (PSO) seekor burung dalam sekelompok kawanan burung disebut sebagai partikel. Setiap individu memiliki perilaku dengan cara menggunakan kecerdasannya (*intelligence*) sendiri serta dipengaruhi oleh perilaku kelompok (Budi, 2013).

Dalam PSO, tiap satu partikel atau satu swarm akan bergerak dalam ruang tertentu untuk menemukan jalan yang tepat atau posisi terbaik yang pernah dilaluinya untuk menuju pada sumber makanan (dapat direpresentasikan sebagai nilai fungsi objektif), kemudian untuk partikel atau swarm lainnya yang masih

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tersisa akan mengikuti jalan tersebut walaupun lokasinya sangat jauh berdasarkan sumber informasi yang diperoleh. Meskipun pada kawanan burung relatif kurang canggih dengan kemampuan terbatas yang dimilikinya (kecerdasannya), namun kemampuan tersebut mampu mencapai tujuan akhir dari penyelesaian pada persoalan terkait pola perilaku dan interaksi melalui berbagai sumber informasi (Budi Santosa, 2011).

Pada algoritma *Particle Swarm Optimization* (PSO), pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel. Inisialisasi populasi dibangkitkan secara random dengan batasan nilai terkecil dan terbesar. Setiap partikel merepresentasikan posisi yang melakukan pencarian solusi optimal dengan melintasi dimensi atau ruang pencarian. Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut yang disebut dengan *personal best* dan penyesuaian terhadap posisi partikel terbaik dari seluruh kumpulan kawanan *swarm* yang disebut dengan *global best* selama melintasi ruang pencarian (Budi Santosa, 2011).

Dalam algoritma PSO, penyebaran terkait informasi terjadi di dalam partikel tersebut dan juga terjadi antara setiap suatu partikel dengan partikel terbaik dari seluruh kawanan selama proses pencarian solusi. Kemudian, akan dilakukan proses pencarian dengan tujuan untuk mencari posisi terbaik *(personal best)* pada setiap partikel dalam sejumlah iterasi tertentu sampai didapatkan posisi yang relatif stabil atau mencapai batas iterasi yang telah ditetapkan. Pada setiap iterasi dalam PSO, setiap solusi yang direpresentasikan oleh posisi partikel, akan dievaluasi performansinya dengan cara memasukkan solusi tersebut kedalam fungsi *fitness* (Kennedy dan Eberhart, 1995).

Posisi partikel dalam *Particle Swarm Optimization* (PSO) diperlakukan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

seperti sebuah titik pada suatu dimensi D dalam ruang tertentu. Dimensi D tersebut merupakan jumlah parameter yang harus dioptimalkan. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel (Kennedy dan Eberhart, 1995). Sistem kerja dari algoritma PSO adalah setiap partikel cenderung akan bergerak kearah tujuan dan faktor dari gerakan ini disebut kecepatan (*velocity*) (Yang, 2013). PSO dikembangkan terutama untuk pengoptimalan dalam ruang pencarian bernilai real. Oleh karena itu untuk pernyataan tersebut, masalah pengoptimalan terbatas pada dimensi D dalam ruang tertentu (Parsopoulos, 2015).

Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu :

$$x_i^{(t)} = (x_{i1}^{(t)}, x_{i2}^{(t)}, \ldots, x_{id}^{(t)})^T \in X \tag{2.11}$$

$$v_i^{(t)} = (v_{i1}^{(t)}, v_{i2}^{(t)}, \ldots, v_{id}^{(t)})^T \tag{2.12}$$

dengan

$$i = 1, 2, 3, \ldots, n \quad d = 1, 2, 3, \ldots, n$$

Keterangan:

x : posisi partikel

v : kecepatan partikel

i : indeks partikel

t : iterasi ke - t

d : dimensi ruang

Pada Persamaan 2.11 mendefinisikan bahwa posisi saat ini dari partikel

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dalam X pada iterasi ke - t. Istilah partikel digunakan untuk mendefinisikan hanya posisi pada saat ini yang dinotasikan dengan x_i. Pada Persamaan 2.12 mendefinisikan pergeseran posisi yang dapat disesuaikan dan mengacu pada perpindahan partikel yang disebut juga dengan kecepatan *(velocity)* partikel (Parsopoulos, 2015).

2.4.2. *Personal Best Particle Swarm Optimization* (PSO)

Personal Best Particle Swarm Optimization (PSO) adalah suatu vektor PSO yang mendefinisikan posisi terbaik untuk partikel ke-i yang didasarkan pada nilai fitness terbaik sampai dengan iterasi ke-t. *Personal best* PSO biasa juga disebut dengan *local best* dari PSO dimana hanya memungkinkan setiap partikel untuk dipengaruhi oleh partikel paling cocok yang dipilih dari lingkungannya. Dalam hal ini informasi akan dipertukarkan dalam lingkungan partikel. *Personal Best Particle Swarm Optimization* (PSO) dapat dinotasikan sebagai berikut (Parsopoulos, 2015).

$$p_i^{(t)} = (p_{i1}^{(t)}, p_{i2}^{(t)}, \ldots, p_{id}^{(t)})^T \in X \tag{2.13}$$

dengan,

$$i = 1, 2, 3, \ldots, n \quad d = 1, 2, 3, \ldots, n$$

keterangan :

p : posisi partikel terbaik

i : indeks partikel

t : iterasi ke - t

d : dimensi ruang

Pada Persamaan 2.13 di atas mendefinisikan posisi partikel terbaik, misalnya

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

posisi terbaik yang pernah dikunjungi di X hingga iterasi ke - t. Untuk P_{best} dalam metode *Particle Swarm Optimization* (PSO) maka kecepatan dapat diihitung dengan persamaan update kecepatan (*velocity*).

2.4.3. *Global Best Particle Swarm Optimization* (PSO)

Global best Particle Swarm Optimization (PSO) adalah suatu partikel terbaik dimana berisi kumpulan dari posisi terbaik dari keseluruhan *personal best* pada iterasi ke-t. *Global best* merupakan suatu posisi dari setiap partikel yang dipengaruhi oleh partikel paling cocok di seluruh kawanan. Partikel tersebut menggunakan informasi yang diperoleh dari semua partikel di dalam seluruh kawanan. Dalam hal ini setiap partikel individu, $i \in 1, 2, 3, ..., n$ dimana $n > 1$, memiliki posisi saat ini di ruang pencarian (x_i), memiliki kecepatan saat ini (v_i), dan posisi terbaik individu di ruang pencarian $(P_{(best,i)})$. Posisi *personal best* $(P_{(best,i)})$ sesuai dengan posisi di ruang pencarian dimana partikel i memiliki nilai terkecil sebagaimana telah ditentukan oleh fungsi objektif (f), dengan mempertimbangkan masalah minimisasi. Selain itu, posisi yang menghasilkan nilai terendah di antara semua yang terbaik $(P_{(best,i)})$ disebut sebagai posisi terbaik global *(global best)* yang dilambangkan dengan G_{best}. Persamaan 2.14 dan 2.15 berikut ini menentukan bagaimana nilai posisi terbaik dan global posisi terbaik dari masing-masing partikel yang akan diperbarui. Dalam masalah minimisasi, maka posisi terbaik $(P_{(best,i)})$ pada langkah berikutnya yaitu $t + 1$ dimana $t \in 0, 1, 2, 3, ..., n$ akan dihitung sebagai berikut (Talukder, 2011).

$$P_{best,i}^{(t+1)} = \begin{cases} P_{best,i}^{(t)}, & \text{jika } f(x_i^{(t+1)}) > P_{best,i}^{(t)} \\ x_i^{(t+1)}, & \text{jika } f(x_i^{(t+1)}) \leqslant P_{best,i}^{(t)}) \end{cases} \tag{2.14}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Dimana f merupakan fungsi *fitness*. Global posisi terbaik (G_{best}) pada saat langkah t akan dihitung sebagai sebagai berikut.

$$G_{best} = \min P_{best,i} \text{ dimana } i \in 1, 2, 3, ..., n \text{ dan } n > 1. \tag{2.15}$$

Oleh karena itu, untuk posisi individu yang terbaik adalah posisi terbaik yang dikunjungi oleh individu partikel sejak langkah pertama kali. Di sisi lain, global posisi terbaik adalah posisi terbaik yang ditemukan oleh salah satu partikel di seluruh kawanan (Andries, 2007). Untuk G_{best} dalam metode *Particle Swarm Optimization* (PSO) maka kecepatan dapat dihitung dengan menggunakan persamaan *update* kecepatan (*velocity*).

2.4.4. *Update Velocity* (Kecepatan) dan *Update* Posisi

Partikel bergerak sepanjang ruang solusi sesuai dengan kecepatan *(velocity)*. Kecepatan berfungsi untuk menentukan arah dari suatu partikel duntuk berpindah dan kemudian memperbaiki posisinya sehingga partikel tersebut dapat menemukan ruang solusi terbaiknya. Pada setiap kecepatan akan diperbarui pada iterasinya dengan memperhatikan beberapa hal, yang meliputi kecepatan sebelumnya, pengaruh *personal best* dan *global best* pada iterasi sebelumnya. Model matematika yang menggambarkan mekanisme *updating status* partikel atau perubahan pada setiap komponen partikel yang memperbarui kecepatan dan posisinya (Kennedy dan Eberhart, 1995):

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + c_1 r_{1j}^{(t)} (P_{best,i}^{(t)} - X_{ij}^{(t)}) + c_2 r_{2j}^{(t)} (G_{best,i} - X_{ij}^{(t)}) \tag{2.16}$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t)} \tag{2.17}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dimana,

$v_{ij}^{(t)}$: vektor kecepatan partikel ke – i dalam dimensi j pada waktu t.

$X_{ij}^{(t)}$: vektor posisi partikel ke – i dalam dimensi j pada waktu t.

$P_{best,i}^{(t)}$: posisi terbaik dari individu partikel ke – i dalam dimensi j yang ditemukan dari inisialisasi pada waktu t.

G_{best} : posisi terbaik dari global partikel ke – i dalam dimensi j yang ditemukan dari inisialisasi pada waktu t.

c_1, c_2 : parameter *learning rates* digunakan untuk mengontrol pengaruh *personal best* dan *global best*.

$r_{1j}^{(t)}, r_{2j}^{(t)}$: angka acak dari distribusi *Uniform* kontinue dari 0 sampai dengan 1 pada iterasi ke-t.

Persamaan 2.16 digunakan untuk menghitung kecepatan partikel yang baru berdasarkan kecepatan sebelumnya, jarak antara posisi saat ini dengan posisi terbaik partikel *(local best)*, dan jarak antara posisi saat ini dengan posisi terbaik kawanan *(global best)*. Kemudian partikel terbang menuju posisi yang baru berdasarkan Persamaan 2.17. Setelah algoritma PSO ini dijalankan dengan sejumlah iterasi tertentu hingga mencapai kriteria pemberhentian, maka akan didapatkan solusi yang terletak pada *global best* (Talukder, 2011).

2.4.5. Parameter Algoritma *Particle Swarm Optimization*

Algoritma PSO memiliki beberapa parameter yang dapat mempengaruhi kinerjanya. Pada setiap masalah optimasi, beberapa nilai dan pilihan parameter memiliki dampak besar terhadap efisiensi metode PSO. Parameter dasar PSO adalah ukuran *swarm*, jumlah iterasi, dan koefisien percepatan. Selain itu, PSO

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

juga dipengaruhi oleh bobot inersia (*inertia weight*) dan *velocity constriction* (Anthony et.al, 2001).

1. Ukuran *Swarm*

   Ukuran *Swarm* merupakan jumlah populasi dari suatu algoritma. Ukuran swarm menghasilkan bagian yang lebih besar dari ruang pencarian yang akan dicakup per iterasi. Sebagian besar jumlah partikel dapat mengurangi jumlah iterasi yang diperlukan untuk mencapai hasil yang optimal. Sedangkan, pada sebagian besar partikel juga dapat meningkatkan kompleksitas komputasi per iterasi, dan membutuhkan waktu yang lebih lama. Beberapa penelitian dan karya ilmiah tentang implementasi PSO menggunakan interval $n \in [20, 60]$ untuk ukuran *swarm* (Talukder, 2011).

2. Jumlah Iterasi

   Jumlah iterasi dilakukan untuk mendapatkan hasil yang baik juga tergantung pada setiap permasalahan. Jika jumlah iterasi yang rendah dapat menghentikan proses pencarian sebelum waktunya, sementara jika terlalu besar jumlah iterasi juga memiliki konsekuensi dari kompleksitas komputasi tambahan yang tidak perlu dan lebih banyak waktu yang dibutuhkan (Andries, 2007).

3. Kecepatan (*Velocity*) Partikel

   Kecepatan *velocity* merupakan vektor penggerak pada proses optimisasi yang menentukan arah dari suatu partikel berpindah dan untuk memperbaiki posisinya semula. Eberhart dan Kennedy pertama kali memperkenalkan *velocity clamping*atau batas penentuan kecepatan. *Velocity clamping* membantu partikel untuk tetap dalam batas dan untuk mengambil ukuran langkah yang wajar dalam ruang pencarian. Jika tanpa menggunakan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

velocity clamping maka proses pencarian akan mudah meledak dan posisi partikel berubah dengan cepat (Bratton, 2007). Kecepatan maksimum mengontrol dalam ruang pencarian dengan *velocity clamping* dan menciptakan keseimbangan yang lebih baik antara eksplorasi global dan eksploitasi lokal. Jika kecepatan maksimum terlalu besar, maka partikel dapat bergerak tidak menentu dan melompati solusi optimal. Sedangkan jika terlalu kecil, maka pergerakan partikel terbatas dan kawanan *swarm* mungkin tidak cukup mengeksplorasi atau terjebak dalam lokal optimal. Kecepatan maksimum dapat didefinisikan sebagai berikut.

$$V_{max} = \varepsilon(X_{max} - X_{min}) \tag{2.18}$$

dimana, X_{max} dan X_{min} batas atas dan batas bawah dari x (inisialisasi posisi) dan nilai ε menggunakan interval nilai random $\varepsilon \in [0, 1]$.

4. *Learning Rates* (c_1 dan c_2)

   Learning rates terbagi menjadi *cognitive parameter dan social parameter*. *Cognitive parameter* merupakan parameter yang digunakan untuk menilai suatu partikel dengan mengontrol pengaruh *personal best* terhadap posisi partikel untuk iterasi sebelumnya(Wati, 2011). *Cognitive parameter* dinotasikan dengan c_1 (Hassan , 2004). Sedangkan, *social parameter* merupakan parameter untuk menilai kemampuan sosial dari suatu *swarm* dengan mengontrol pengaruh dari *global best* terhadap posisi partikel untuk iterasi sebelumnya (Wati, 2011). *Social parameter* dinotasikan dengan c_2(Hassan , 2004). Ada beberapa jenis dari c_1 dan c_2 :

   a. Jika $c_1 = c_2 = 0$, maka seluruh partikel akan terus terbang dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

kecepatan partikel tersebut hingga mencapai batas ruang pencarian.

b. Jika $c_1 > 0$ dan $c_2 = 0$, maka semua partikel bersifat independen (tidak terikat). Sedangkan, jika $c_1 = 0$ dan $c_2 > 0$, maka semua partikel akan tertarik pada satu titik.

c. Jika $c_1 = c_2$, maka semua partikel tertarik ke arah rata-rata dari $P_{best,i}^{(t)}$ dan G_{best}.

d. Jika $c_1 < c_2$, maka setiap partikel lebih kuat dipengaruhi oleh posisi terbaik dan menghasilkan pencarian yang berlebihan. Sedangkan jika $c_1 > c_2$, maka semua partikel jauh lebih dipengaruhi oleh *Global best*, sehingga menyebabkan semua partikel berjalan sebelum waktunya optimal (Van, 2006).

Namun, normalnya c_1 dan c_2 bersifat statis(tidak bergerak) dengan nilai yang dioptimalkan diperoleh secara empiris(dalam penemuan atau percobaan penelitian). Inisialisasi yang salah dari c_1 dan c_2 dapat menyebabkan divergen(dalam keadaan menyebar). Berdasarkan dari berbagai penelitian, telah diusulkan bahwa dua *learning rate* dapat menggunakan nilai $c_1 = c_2 = 2$ (Andries, 2007).

5. *Inertia weight*

Inertia weight merupakan parameter yang digunakan untuk mengontrol efek dari kecepatan partikel sebelumnya pada kecepatan yang sekarang dan dinotasikan dengan ω. *Inertia weight* pertama kali diperkenalkan oleh Y.Shi dan Eberhart pada tahun 1998 sehingga disebut dengan modifikasi PSO(Y.Shi dan Eberhart , 1998). Modifikasi PSO tersebut menggunakan Strategi *Constant Inertia Weight*. Oleh karena itu, modifikasi dari memperbarui kecepatan dalam algoritma PSO tersebut berdasarkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Persamaan 2.16 berubah menjadi:

$$v_{ij}^{(t+1)} = \omega v_{ij}^{(t)} + c_1 r_{1j}^{(t)}(P_{best,i}^{(t)} - X_{ij}^{(t)}) + c_2 v_{2j}^{(t)}(G_{best,i} - X_{ij}^{(t)}) \qquad (2.19)$$

Terdapat beberapa penelitian terkait ketentuan nilai yang digunakan pada parameter ω diantaranya adalah sebagai berikut.

a. Jika $\omega \geq 1$, maka kecepatan meningkat dari waktu ke waktu dan tidak dapat mengubah arahnya untuk bergerak kembali ke arah yang optimal sehingga *swarm* akan bersifat divergen.

b. Jika $\omega < 1$, maka momentum kecil hanya disimpan dari langkah sebelumnya dan perubahan arah yang cepat diatur dalam proses pencarian ruang.

c. Jika $\omega = 0$, maka kecepatan partikel menghilang dan semua partikel bergerak tanpa sepengetahuan kecepatan sebelumnya di setiap proses pencarian ruang.

Tabel 2.1 menunjukkan strategi terbaik untuk bobot inersia berdasarkan penelitian yang dilakukan oleh Bansal (2011) dengan kriteria tertentu dari beberapa penelitian.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 2.1 Strategi *Inertia Weight*

Strategi	Definisi *Inertia Weight*
Costant inertia weight	$\omega(t) = \omega = const$
Random inertia weight	$\omega(t) = 0.5 + \frac{r}{2}$ with $r \sim U(0,1)$
Lineary decreasing inertia weight	$\omega(t) = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} t$
Chaotic inertia weight	$\omega(t) = (\omega(0) - \omega(t_{max}))(\frac{t_{max} - t}{t_{max}}) + \omega(t_{max})z$ $z = 4r(1-r)$ with $r \sim U(0,1)$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

6. *Consriction Coefficient*

Consriction coefficient dikembangkan oleh Kennedy dan Clerc (2002) dengan notasi χ (Clerc dan Kennedy, 2002). Pada koefisien ini sangat penting untuk mengontrol *trade-off* eksplorasi dan eksploitasi untuk memastikan perilaku konvergensi, serta untuk mengecualikan bobot inersia (ω) dan kecepatan maksimum (V_{max}) (Kennedy dan Tim Blackwell, 2007). Persamaan pembaruan kecepatan yang diajukan oleh Clerc dihitung sebagai berikut:

$$v_{ij}^{(t+1)} = \chi(v_{ij}^{(t)} + c_1 r_{1j}^{(t)}(P_{best,i}^{(t)} - X_{ij}^{(t)}) + c_2 v_{2j}^{(t)}(G_{best,i} - X_{ij}^{(t)})) \quad (2.20)$$

dimana,

$$\chi = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (2.21)$$

dengan nilai k adalah $0 \leq k \leq 1$, untuk φ_1 dan φ_2 diperoleh berdasarkan nilai parameter c_1 dan c_2 yang digunakan dalam algoritma PSO sehingga $\varphi_1 = c_1$ dan $\varphi_2 = c_2$. Sedangkan, untuk nilai φ diperoleh dari hasil penjumlahan antara nilai φ_1 dan φ_2 dengan syarat hasil penjumlahan tersebut harus lebih dari atau sama dengan 4, sehingga $\varphi = \varphi_1 + \varphi_2 \geq 4$. Jika nilai $\varphi < 4$, maka semua partikel perlahan-lahan akan berputar menuju di sekitar solusi terbaik pada ruang pencarian tanpa jaminan konvergensi. Jika $\varphi > 4$, maka semua partikel menyatu dengan cepat dan terjamin sehingga lebih disarankan jika nilai $\varphi > 4$ (Bratton, 2007). Algoritma PSO dengan *constriction coefficient* setara dengan algoritma PSO dengan bobot inersia sehingga pada Persamaan 2.19 sama dengan 2.20 (Kennedy dan Tim Blackwell, 2007).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.4.6. Tahapan Algoritma *Particle Swarm Optimization*

Model algoritma *Particle Swarm Optimization* akan disimulasikan pada dimensi dalam ruang tertentu sesuai dengan jumlah iterasi sehingga pada setiap iterasi, posisi partikel akan semakin mengarah menuju target (fungsi minimum atau maksimum). Proses tersebut akan dilakukan hingga mencapai iterasi maksimum atau berdasarkan pada kriteria penghentian (*stopping criteria*). Algoritma *Particle Swarm Optimization* (PSO) meliputi tahapan dan proses sebagai berikut.

1. Inisialisasi posisi awal dalam algoritma PSO nilai akan dibangkitkan secara random dengan menentukan variabel minimum (X_{min}) dan variabel maximum (X_{max}) (Chen dan Shih, 2013).

2. Inisialisasi kecepatan awal yang dibangkitkan secara random berdasarkan Persamaan 2.18 sebagai kecepatan maksimum (V_{max}), sedangkan untuk kecepatan minimum nilainya adalah $V_{min} = -V_{max}$. Kecepatan maksimum (V_{max}) adalah batas atas untuk menentukan inisialisasi nilai random kecepatan awal, sedangan kecepatan minimum (V_{min}) adalah batas bawah untuk menentukam inisialisasi nilai random kecepatan awal.

3. Evaluasi nilai *fitness* dari masing-masing partikel berdasarkan posisinya. Jika nilai *fitness* posisi partikel saat ini lebih baik dari nilai *Pbest*, maka nilai *Pbest* yang baru diambil dari nilai *fitness* posisi partikel pada saat ini.

4. Menentukan partikel dengan nilai *fitness* terbaik dan membandingkannya dengan *Gbest*. Jika *Gbest* yang terbaik maka *Gbest* yang akan di *update* kecepatannya.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

5. Menggunakan *Pbest* dan *Gbest* yang ada, kemudian memperbarui kecepatan pada setiap partikel dengan menggunakan Persamaan 2.16. Setelah kecepatan baru diperoleh, maka langkah selanjutnya adalah memperbarui posisi baru setiap partikel menggunakan Persamaan 2.17.

6. Cek apakah solusi yang sekarang sudah konvergen. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka langkah 2 diulang dengan memperbarui iterasi $i = i + 1$, dengan cara menghitung nilai baru dari *Pbest,j* dan *Gbest*. Proses iterasi ini dilanjutkan sampai semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian (*stopping criteria*), misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil.

7. Kondisi Penghentian *stopping criteria* memiliki beberapa aspek meliputi sebagai berikut.

   a. Kondisi berhenti jika pada algoritma *Particle Swarm Optimization* (PSO) terjadi *covergent premature* (memusat sebelum waktunya) dimana menyebabkan solusi menjadi tidak optimal.

   b. Kondisi berhenti harus melindungi dari kondisi *oversampling* pada nilainya. Pengertian dari kondisi *oversampling* merupakan mekanisme untuk menyeimbangkan distribusi kelas dengan cara replikasi *instance* kelas minoritas secara acak. Jika kondisi berhenti memerlukan perhitungan yang terus menerus maka kerumitan dari proses pencarian akan meningkat.

   c. Kondisi berhenti jika jumlah iterasi telah mencapai iterasi maksimum

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

yang diperbolehkan.

d. Kondisi berhenti jika solusi yang diterima ditemukan.

e. Kondisi berhenti jika sudah tidak ada perkembangan setelah beberapa iterasi (Engelbrecht, 2006).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB III

METODE PENELITIAN

Metode penelitian dalam penerapan algoritma *Particle Swarm Optimization* (PSO) untuk menyelesaikan persoalan *Vehicle Routing Problem with Time Windows* (VRPTW) adalah sebagai berikut.

3.1. Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian terapan atau *applied research*. Pengertian dari penelitian terapan atau *applied research* adalah salah satu jenis penelitian yang hasilnya dapat secara langsung diimplementasikan atau diterapkan dalam suatu permasalahan. Secara umum, tujuan utama dari penelitian terapan pada tulisan ini adalah untuk mencari solusi dari pemecahan masalah yang dihadapi sehingga dapat diterapkan dalam pemanfaatannya oleh kepentingan individu maupun kepentingan kelompok (Sukardi, 2013). Pada penelitian ini juga tergolong kedalam pendekatan penelitian kuantitatif. Penelitian kuantitatif adalah penelitian yang urut dan sistematis (terstruktur), menggunakan data numerik (angka) serta mengkuantifikasi data untuk digeneralisasikan (Kurniawan dan Puspitaningtyas, 2016). Pendekatan kuantitatif memiliki tujuan untuk menjelaskan, menguji keterkaitan antar variabel, dan juga menguji suatu teori-teori (Dinas Kependidikan, 2008).

Pada penelitian ini menguji manfaat dari metode *Particle Swarm Optimization* (PSO) serta mengetahui bagaimana hubungan empiris dan menganalisis dalam permasalahan yang muncul pada bidang transportasi.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Permasalahan terkait dengan mencari rute terpendek disebut dengan Vehicle Routing Problem (VRP). Penelitian ini menggunakan salah satu tipe dari VRP yaitu tipe *Vehicle Routing Problem with Time Window* (VRPTW). VRPTW memiliki batasan batasan waktu (*time window*) yang harus dipenuhi. Tujuan yang diharapkan adalah untuk menemukan rute terpendek sehingga menekan atau dapat meminimumkan biaya.

3.2. Data Penelitian

Dalam penelitian ini, untuk melakukan pengujian dari penerapan algoritma *Particle Swarm Optimization* (PSO) maka penulis menggunakan data sekunder. Data sekunder merupakan sumber data yang tidak langsung memberikan data kepada pengumpul data melainkan data mentah yang telah diolah oleh orang lain (Hasan, 2002). Data sekunder yang digunakan pada penelitian ini bersumber dari permasalahan *Vehicle Routing Problem with Time Window* (VRPTW). Permasalahan tersebut merupakan permasalahan benchmark Solomon pada kasus VRPTW yang berstandar internasional.

Data yang disediakan oleh Marius M. Solomon tersebut dapat diakses dan diunduh dari http://w.cba.neu.edu/ msolomon/problems.htm. Marius M. Solomon adalah Profesor di Kelompok Informasi, Operasi dan Analisis di Northeastern University, Boston, Massachusetts. Dataset yang tersedia pada website tersebut terbagi atas beberapa kelompok, diantaranya adalah dataset dengan node yang letaknya acak/random (R), dataset dengan node yang letaknya sudah diklaster (C), dan dataset dengan node yang letaknya random dan sudah terklaster (RC). Bentuk data yang akan digunakan pada penelitian ini, yakni dataset dengan node yang letaknya random dan terkluster (RC). Data yang digunakan untuk penelitian adalah RC101 sebanyak 100 *custumers*, Tabel 3.1 menunjukkan contoh data 3 pelanggan.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 3.1 Data Random Cluster 101 (RC101) dengan 3 Titik

Pelanggan ke-	x	y	*Demand* (q_i)	*Earliest Time* (e_i)	*Latest Time* (l_i)	*Service Time* (f_i)
1	40	50	0	0	240	0
2	25	85	20	145	175	10
3	22	75	30	50	80	10

Berdasarkan Tabel 3.1 menunjukkan contoh data yang akan digunakan dalam penelitian. Keterangan pada Tabel 3.1 ditunjukkan sebagai berikut.

x : titik koordinat x

y : titik koordinat y

Demand : permintaan setiap pelanggan

Earliest Time : waktu awal setiap pelanggan

Latest Time : waktu akhir setiap pelanggan

Service Time : waktu pelayanan setiap pelanggan

3.3. Perancangan Persoalan VRPTW

Pada penelitian ini, penerapan dari algoritma *Particle Swarm Optimization* (PSO) yang akan dirancang pada sistem. Sistem dibangun untuk menyelesaikan permasalahan pada distribusi barang terutama pada bidang transportasi. Permasalahan tersebut terkait pada optimasi yaitu pencarian jalur atau rute yang paling optimum yang termasuk kedalam bentuk permasalahan *Vehicle Routing Problem with Time Windows* (VRPTW).

Vehicle Routing Problem with Time Windows (VRPTW) merupakan salah satu tipe dari Vehicle Routing Problem (VRP) dengan memiliki batasan pada kapasitas kendaraan dan tambahan *time window* yang harus dipenuhi. Pada *time*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

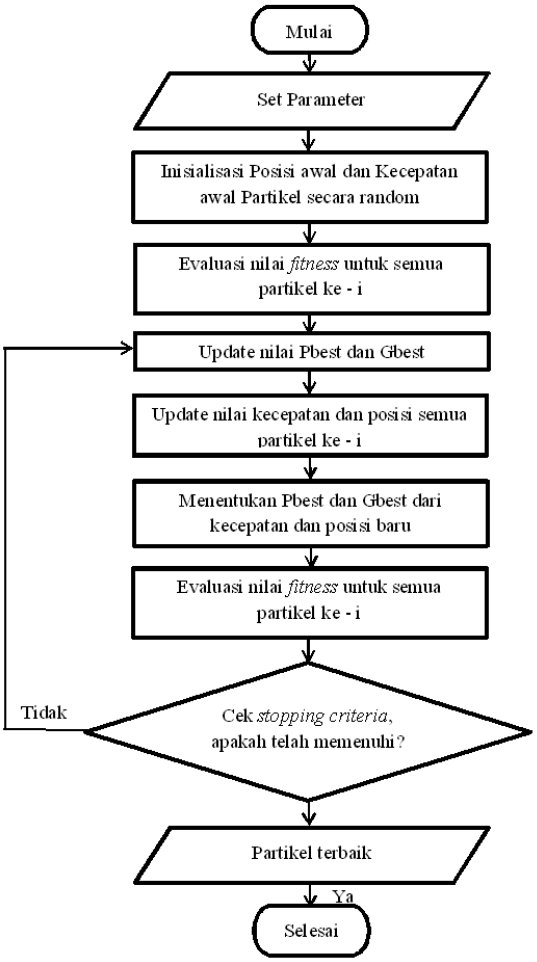
window tersebut meliputi waktu awal (*earliest time*), waktu akhir (*latest time*), dan waktu pelayanan (*service time*) pada setiap node. Secara umum, permasalahan VRPTW pada penelitian ini dapat dideskripsikan sebagai berikut.

1. Terdapat satu pusat distribusi (*distribution center*) yang disebut juga dengan depot. Selain itu terdapat beberapa kendaraan dengan kapasitas tertentu untuk melayani permintaan *customers* ke sejumlah node.
2. Pada tiap-tiap kendaraan, untuk rutenya harus dimulai dan diakhiri pada pusat distribusi atau depot.
3. Jumlah total permintaan *customers* yang akan dilayani pada tiap-tiap kendaraan tidak boleh melebihi kapasitas kendaraan (Q).
4. Setiap satu *customers* hanya akan dilayani satu kali *service* dan sesuai dengan batas waktu atau *time windows*.
5. Setiap pelayanan yang dilakukan untuk semua node disesuaikan dengan *time window* depot [e_0, l_0], dengan waktu awal untuk kendaraan berangkat dari pusat distribusi atau depot adalah e_0 dan waktu akhir saat kendaraan kembali ke pusat distribusi atau depot adalah l_0.
6. Setiap kendaraan diizinkan datang sebelum sebelum waktu awal (e_i) dengan syarat harus menunggu sampai batas waktu awal pelayanan tiap node tersebut sehingga dapat memenuhi permintaan node.
7. Pada setiap node ke-i maka akan memiliki permintaan q_i, waktu pelayanan (*service time*) f_i, dan batasan waktu (*time window*) [e_i, l_i].

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3.4. Perancangan Algoritma *Particle Swarm Optimization* (PSO)

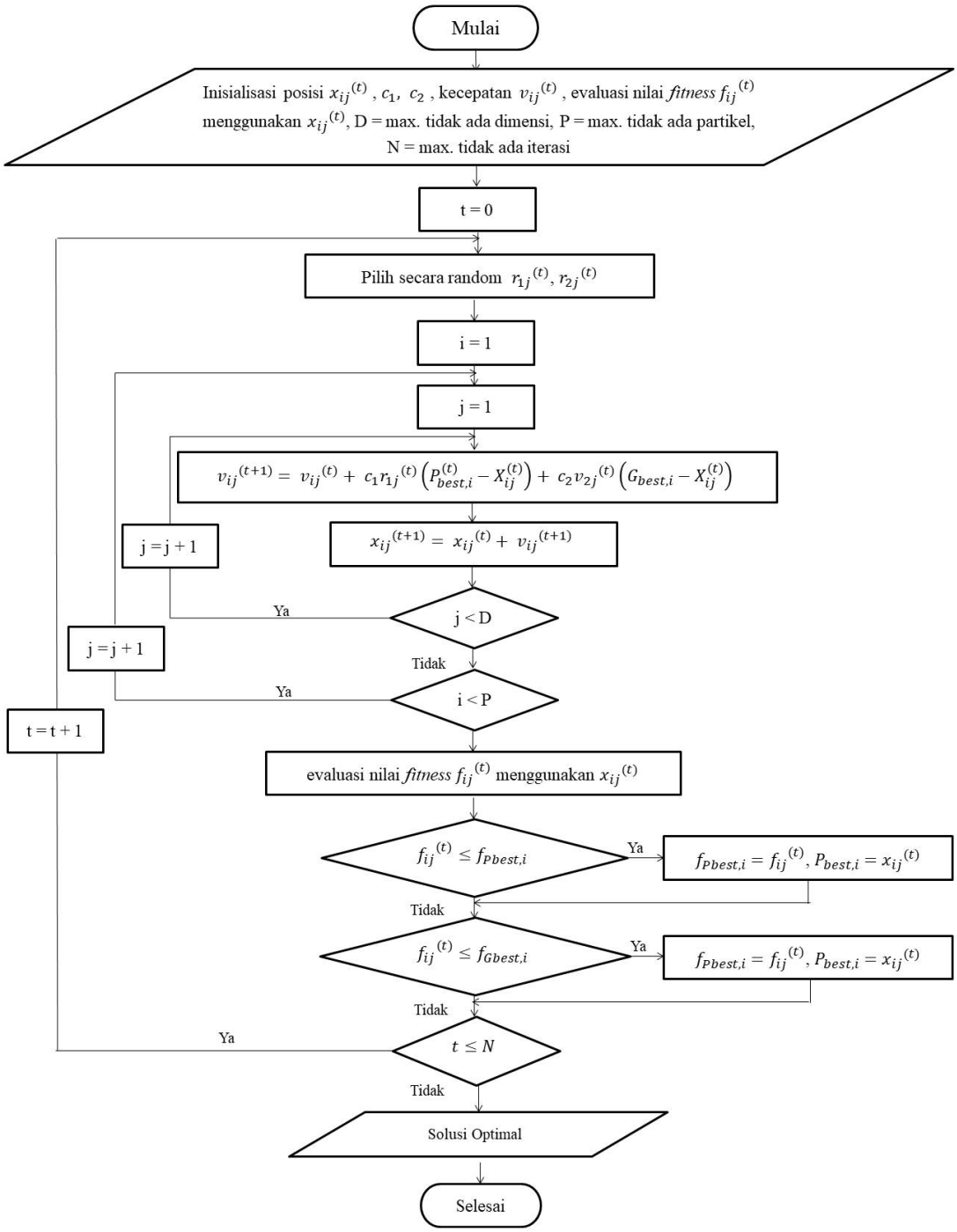
Perancangan pada algoritma *Particle Swarm Optimization* dalam menyelesaikan persoalan *Vehicle Routing Problem with Time Windows* (VRPTW) bertujuan memperoleh solusi paling optimal. Proses penyelesaian persoalan dengan algoritma PSO akan ditunjukkan dengan *flowchart* pada gambar dibawah ini.



Gambar 3.1 *Flowchart* algoritma PSO

Untuk tahapan mencari *Personal Best* (Pbest) PSO dan *Global Best* (Gbest) PSO akan digambarkan pada *flowchart* sebagai berikut.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 3.2 *Flowchart* Pbest dan Gbest PSO

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3.4.1. Tahapan dan Proses Algoritma *Particle Swarm Optimization*

Langkah-langkah implementasi algoritma *Particle Swarm Optimization* (PSO) dalam menyelesaikan permasalahan *Vehicle Routing Problem with Time Windows* (VRPTW) adalah sebagai berikut.

1. Melakukan tinjauan pustaka mengenai permasalahan yang berhubungan dengan penerapan dari Algoritma *Particle Swarm Optimization* (PSO), penyelesaian permasalahan *Vehicle Routing Problem with Time Window* (VRPTW).
2. Mengolah data VRPTW dengan cara menghitung jarak dan waktu antar pelanggan pada data RC101 dalam Lampiran B. Jarak tersebut dihitung berdasarkan persamaan Euclidean dengan melibatkan nilai koordinat x dan koordinat y pada data RC101. Sedangkan, untuk menentukan waktu antar pelanggan dilakukan dengan cara membagi jarak antar pelanggan dengan inisialisasi kecepatan rata-rata.
3. Tahapan dari penerapan algoritma *Particle Swarm Optimization* (PSO) adalah sebagai berikut :
    a. Input data VRPTW yang terdapat pada Lampiran B dan parameter algoritma PSO yang akan digunakan. Parameter tersebut meliputi jumlah *swarm*, parameter *learning rates* yaitu c_1 dan c_2, serta strategi dari *inertia weight* dengan menggunakan persamaan pada Tabel 2.1 dan Persamaan 2.20.
    b. Inisialisasi populasi posisi awal secara random dengan menentukan variabel minimum X_{min} dan variabel maximum X_{max}. Selain itu, kecepatan (*velocity*) partikel awal dilakukan secara random dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menentukan kecepatan maksimum V_{max} berdasarkan Persamaan 2.18. Sedangkan untuk kecepatan minimum V_{min} nilainya sama dengan $-V_{max}$.

c. Menghitung nilai fitnes pada setiap partikel. Pada penelitian ini nilai fitnes tersebut dihitung berdasarkan persamaan fungsi tujuan VRPTW yaitu dengan cara menghitung total jarak.

d. Memilih nilai fitnes terbaik yaitu *personal best* (*Pbest*) dan *global best* (*Gbest*) dari partikel. *Pbest* adalah hasil terbaik dari setiap partikel dan *Gbest* adalah hasil terbaik dari *Pbest*.

e. Mengupdate kecepatan (*velocity*) baru menggunakan Persamaan 2.16.

f. Mengupdate posisi baru setiap partikel dengan menggunakan Persamaan 2.17.

g. Menentukan *personal best* (*Pbest*) dan *global best* (*Gbest*) dari posisi baru tersebut.

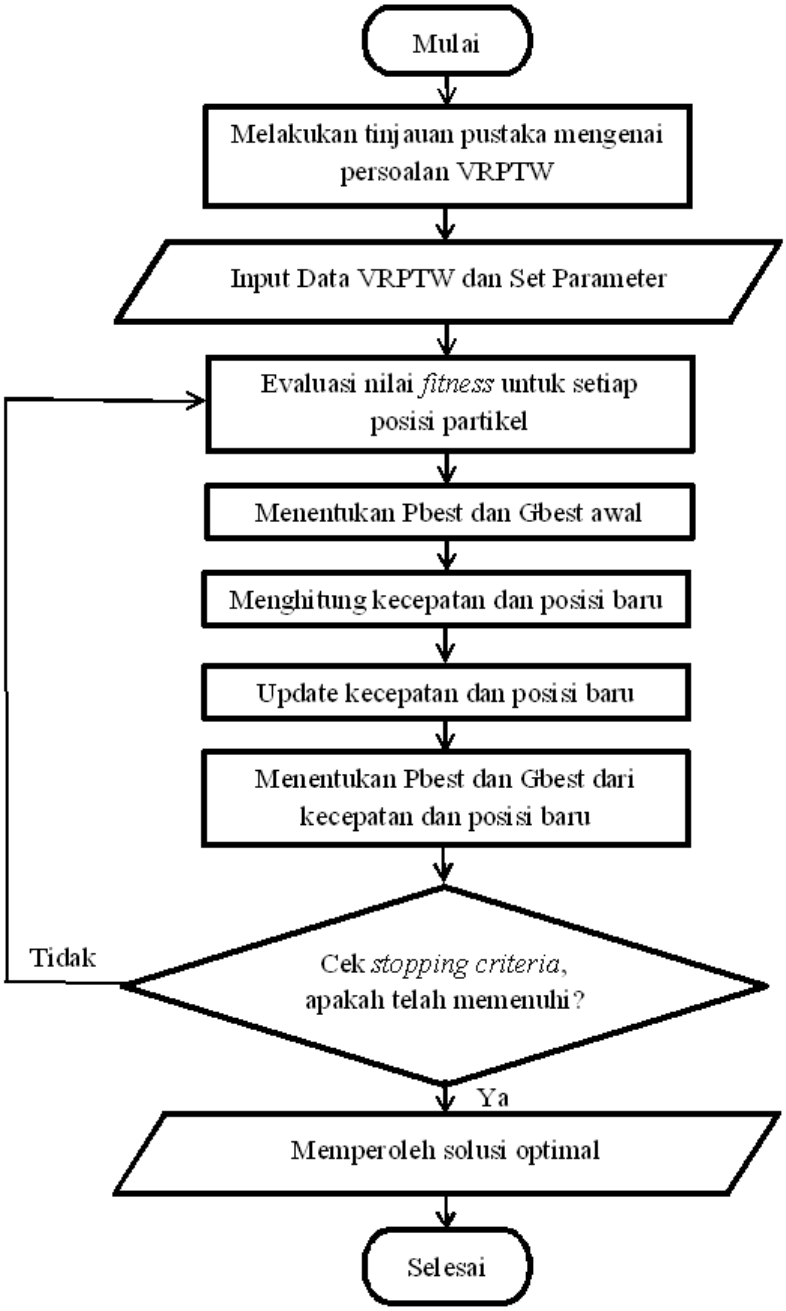
h. Ulangi pada langkah poin c hingga memperoleh iterasi yang maksimal.

i. Memperoleh solusi optimum berdasarkan posisi terbaik dari algoritma *Particle Swarm Optimization* (PSO).

j. Selesai.

Berikut ini merupakan gambar *flowchart* yang menunjukkan tahapan penelitian.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 3.3 *Flowchart* Tahapan Penelitian

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB IV

HASIL DAN PEMBAHASAN

Pada bab hasil dan pembahasan akan dijelaskan bagaimana penerapan algoritma *Particle Swarm Optimization* (PSO) dalam menyelesaikan persolan *Vehicle Routing Problem with Time Window* (VRPTW). Persoalan VRPTW adalah suatu persoalan dalam menentukan rute kendaraan yang paling optimal dimana terdapat satu pusat distribusi (depot) yang memiliki sejumlah kendaaran dengan kapasitas tertentu untuk melakukan pelayanan pengiriman kepada sejumlah pelanggan. Pada penentuan rute tersebut dibentuk dengan memperhatikan kendala batasan waktu (*time window*) dan maksimum kapasitas kendaraan. Pada bab ini juga akan dijelaskan hasil dari analisis perbandingan berdasarkan jumlah data yang digunakan, dan pengaruh parameter-parameter algoritma *Particle Swarm Optimization* (PSO).

4.1. Implementasi Algoritma PSO dalam Proses Perhitungan Secara Manual Terhadap Persoalan VRPTW

Proses perhitungan secara manual pada persoalan *Vehicle Routing Problem with Time Window* (VRPTW) menggunakan algoritma *Particle Swarm Optimization* diantaranya adalah sebagai berikut.

Langkah 1 : Input Data VRPTW dan Inisialisasi Parameter

Langkah pertama pada perhitungan manual persoalan *Vehicle Routing Problem with Time Windows* (VRPTW) menerapkan algoritma *Particle Swarm*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Optimization (PSO) menggunakan dataset benchmark Solomon dengan tipe data Random Cluster 101 (RC101) yang secara rinci terdapat pada Lampiran B. Pada data RC101 tersebut berjumlah 101 dengan jumlah pelanggan sebanyak 100 dan terdapat 1 pusat distribusi (depot). Dataset RC101 untuk implementasi perhitungan secara manual dapat digambarkan pada Tabel 4.1.

Tabel 4.1 Data Benchmark Solomon Random Cluster 101 (RC101)

Pelanggan ke-	x	y	*Demand* (q_i)	*Earliest Time* (e_i)	*Latest Time* (l_i)	*Service Time* (f_i)
(Depot)1	40	50	0	0	240	0
2	25	85	20	145	175	10
3	22	75	30	50	80	10
4	22	85	10	109	139	10
5	20	80	40	141	171	10
6	20	85	20	41	71	10
⋮	⋮	⋮	⋮	⋮	⋮	⋮
101	31	67	3	180	210	10

Pada Tabel 4.1 menunjukkan dataset RC101 dengan depotnya adalah 1 (terdapat pada baris pertama) dan jumlah pelanggan ditunjukkan pada pelanggan ke-2 sampai pelanggan ke-101. Sedangkan, untuk inisialisasi parameter dari algoritma *Particle Swarm Optimization* (PSO) meliputi:

jumlah *swarm* $= 30$

banyaknya pelanggan $= 100$

kapasitas maksimum kendaraan $= 200$

iterasi maksimum $= 1000$

$r_1 = 0.2$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$r_2 = 0.4$

$c_1 = 2$

$c_2 = 2$

$\omega = 0.7$

Selain dataset RC101 pada Tabel 4.1 di atas, kemudian dilakukan olah data dengan menghitung data jarak antar pelanggan dan data waktu tempuh (waktu lama perjalanan) yang digambarkan pada Tabel 4.2 dan Tabel 4.3. Data jarak tersebut diperoleh dengan Persamaan Euclidean berdasarkan data koordinat x dan koordinat y pada Tabel 4.1.

$$C_{ij} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Keterangan :

C_{ij} : jarak antara pelanggan ke-i menuju pelanggan ke-j

x : koordinat titik x

y : koordinat titik y

Sebagai contoh akan dilakukan satu kali perhitungan misalnya pada jarak antara depot (1) menuju pelanggan ke-2.

Diketahui :

$x_1 = 40$; $y_1 = 50$ (titik koordinat x dan koordinat y depot (1))

$x_2 = 25$; $y_2 = 85$ (titik koordinat x dan koordinat y pelanggan ke-2)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
C_{ij} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\
&= \sqrt{(25 - 40)^2 + (85 - 50)^2} \\
&= \sqrt{(-15)^2 + (35)^2} \\
&= \sqrt{225 + 1225} \\
&= \sqrt{1450} \\
&= 38.08
\end{aligned}
$$

Proses perhitungan dari jarak antar pelanggan lainnya akan dilakukan dengan cara yang sama seperti proses perhitungan dari jarak antara depot (1) menuju pelanggan ke-2. Tabel 4.2 menunjukkan hasil perhitungan jarak antar pelanggan.

Tabel 4.2 Data Jarak Antar Pelanggan

Pelanggan ke-	1	2	3	4	5	6	$\cdots$	101
1	0	38.08	30.81	39.36	36.06	40.31	$\cdots$	19.24
2	38.08	0	10.44	3	7.07	5	$\cdots$	18.97
3	30.81	10.44	0	10	5.39	10.2	$\cdots$	12.04
4	39.36	3	10	0	5.39	2	$\cdots$	20.12
5	36.06	7.07	5.39	5.39	0	5	$\cdots$	17.03
6	40.31	5	10.2	2	5	0	$\cdots$	21.1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
101	19.24	18.97	12.04	20.12	17.03	21.1	$\cdots$	0

Berdasarkan Tabel 4.2 menunjukkan gambaran hasil perhitungan untuk jarak antar pelanggan dengan jumlah pelanggan sebanyak 100. Sedangkan untuk data waktu diperoleh dengan cara mengkonversi dari data jarak antar pelanggan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

yang ditunjukkan pada Tabel 4.2. Inisialisasi untuk kecepatan rata-rata yang digunakan adalah 60 Km/jam. Selain itu, satuan jam dikonversi dalam bentuk menit sehingga akan dikali dengan 60. Perhitungannya dilakukan dengan menggunakan Persamaan dibawah ini.

$$\text{waktu tempuh}(t_{ij}) = \frac{\text{jarak(Km/jam)}}{\text{kecepatan(Km)}} \times 60$$

Sebagai contoh akan dilakukan satu kali perhitungan misalnya pada waktu tempuh antara depot (1) menuju pelanggan ke-2.

Diketahui :

$C_{ij} = C_{1,2} = 38.08$ Km (jarak depot (1) menuju pelanggan ke-2)

kecepatan = 60 Km/jam

$$\begin{aligned}\text{waktu tempuh}(t_{1,2}) &= \frac{\text{jarak(Km/jam)}}{\text{kecepatan(Km)}} \\ &= \frac{38.08}{60} \times 60 \\ &= 38.08\end{aligned}$$

Proses perhitungan konversi jarak ke waktu tempuh untuk antar pelanggan dilakukan dengan cara yang sama seperti waktu tempuh dari depot (1) menuju pelanggan ke-2. Tabel 4.3 menunjukkan hasil perhitungan waktu tempuh antar pelanggan.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.3 Data Waktu Tempuh Antar Pelanggan

Pelanggan ke-	1	2	3	4	5	6	$\cdots$	101
1	0	38.08	30.81	39.36	36.06	40.31	$\cdots$	19.24
2	38.08	0	10.44	3	7.07	5	$\cdots$	18.97
3	30.81	10.44	0	10	5.39	10.2	$\cdots$	12.04
4	39.36	3	10	0	5.39	2	$\cdots$	20.12
5	36.06	7.07	5.39	5.39	0	5	$\cdots$	17.03
6	40.31	5	10.2	2	5	0	$\cdots$	21.1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
101	19.24	18.97	12.04	20.12	17.03	21.1	$\cdots$	0

Berdasarkan Tabel 4.3 menunjukkan gambaran hasil perhitungan untuk waktu tempuh antar pelanggan dengan jumlah pelanggan sebanyak 100. Karena kecepatan rata-rata yang digunakan adalah 60 Km/jam maka hasil konversi jarak ke waktu tempuh hasilnya sama.

Langkah 2 : Membangkitkan Posisi Awal dan Kecepatan Awal Secara Random pada Algoritma *Particle Swarm Optimization* (PSO)

Posisi awal akan dibangkitkan secara random pada partikel-partikelnya dengan variabel minimum $X_{min} = -5$ dan variabel maximum $X_{max} = 5$ sebanyak jumlah pelanggan yang mewakili posisi dari setiap *swarm*. Pada perhitungan ini menggunakan jumlah *swarm* sebanyak 30.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.4 Posisi Awal Partikel Pada Setiap *Swarm*

Swarm ke-	Pelanggan ke-							
	1	2	3	4	5	6	$\cdots$	101
1	1.054	-3.581	-1.797	4.973	2.624	4.78	$\cdots$	3.292
2	-1.320	0.855	-3.702	0.699	4.098	-1.732	$\cdots$	-0.978
3	-0.845	1.202	-4.336	-1.606	0.835	4.472	$\cdots$	0.314
4	3.324	-4.015	-3.019	-2.229	-2.466	-3.492	$\cdots$	-1.938
5	-1.753	3.093	-4.518	0.448	-3.561	2.951	$\cdots$	-1.572
6	1.335	3.031	-0.806	0.865	1.638	4.424	$\cdots$	4.013
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
30	-4.510	4.641	3.550	3.485	0.273	-4.318	$\cdots$	2.217

Selain itu, dibangkitkan pula kecepatan awal yang diinisialisasi berdasarkan Persamaan 2.18.

Misal : $\varepsilon = 0.2$

$V_{max} = \varepsilon(X_{max} - X_{min})$

$V_{max} = 0.2(5 - (-5))$

$V_{max} = 2$

Sehingga diperoleh kecepatan maksimum sebesar 2, sedangkan untuk kecepatan minimum $V_{min} = -V_{max} = -2$. Proses membangkitkan posisi dan kecepatan awal tersebut dilakukan pula sebanyak jumlah *swarm* dimana pada perhitungan manual ini menggunakan 30 *swarm*. Kecepatan awal partikel-partikel tersebut digambarkan pada Tabel 4.5.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.5 Hasil Inisialisasi Kecepatan Awal Partikel Pada Setiap *Swarm*

Swarm ke-	Pelanggan ke-							
	1	2	3	4	5	6	···	101
1	-1.491	-1.064	-1.301	-1.829	1.608	1.069	···	-0.721
2	-0.430	1.381	-1.835	-0.590	0.479	-1.467	···	-1.897
3	-1.828	1.364	1.974	0.297	-0.516	0.289	···	1.888
4	-1.090	1.049	-1.792	1.54	-1.832	-0.289	···	-0.529
5	-1.116	-1.706	1.073	-0.930	1.387	1.268	···	0.204
6	-0.713	1.280	0.992	0.278	-1.471	-0.500	···	-0.086
⋮	⋮	⋮	⋮	⋮	⋮	⋮	···	⋮
30	-1.089	-1.488	1.607	-1.718	1.760	1.730	···	-0.152

Pada Tabel 4.5 menunjukkan hasil pembangkitan inisialisasi kecepatan awal pada setiap *swarm*. Pada setiap *swarm* memiliki kecepatan awal sesuai dengan jumlah pelanggan sebanyak 100.

Langkah 3 : Evaluasi Nilai Fitnes

Pada langkah ini, evaluasi nilai fitnes dapat dilakukan untuk setiap *swarm* dengan cara melakukan perhitungan berdasarkan fungsi tujuan VRPTW pada masing-masing *swarm* tersebut. Namun, sebelum menghitung fungsi tujuan, maka harus dilakukan pengurutan dahulu dari mulai yang terkecil hingga yang terbesar pada bilangan ramdom dalam inisialisasi posisi awal yang terdapat dalam Tabel 4.4. Proses tersebut dilakukan dengan cara memberi nomor urut atau indeks (nilai random tersebut diubah menjadi bentuk permutasi) pada masing-masing partikel *swarm* yang mewakili suatu posisi. Berdasarkan hal tersebut hasil pengurutan posisi yang membentuk solusi awal dapat digambarkan pada Tabel 4.6.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.6 Indeks Hasil Pengurutan Posisi Awal Setiap *Swarm*

Swarm ke-	Solusi Awal yang Terbentuk
1	1 - 33 - 54 - 48 - 74 - 45 - 28 - 73 - 46 - 35 - 60 - 94 - 17 - 81 - 31 - 69 - 70 -
	41 - 3 - 58 - 64 - 85 - 14 - 59 - 22 - 83 - 65 - 93 - 20 - 92 - 96 - 72 - 55 - 16 -
	84 - 12 - 91 - 13 - 4 - 23 - 39 - 77 - 37 - 95 - 78 - 61 - 71 - 66 - 75 - 10 - 87 -
	9 - 89 - 88 - 34 - 8 - 21 - 30 - 19 - 68 - 32 - 79 - 100 - 44 - 63 - 38 - 98 - 2 -
	47 - 40 - 86 - 99 - 25 - 56 - 36 - 24 - 52 - 27 - 6 - 62 - 90 - 57 - 101 - 26 - 76 -
	50 - 15 - 29 - 43 - 42 - 80 - 11 - 18 - 67 - 51 - 49 - 7 - 82 - 53 - 5 - 97 - 1
2	1 - 68 - 26 - 78 - 99 - 27 - 23 - 66 - 95 - 10 - 61 - 75 - 69 - 22 - 94 - 54 - 98 -
	9 - 18 - 4 - 30 - 84 - 73 - 63 - 100 - 25 - 17 - 60 - 90 - 34 - 57 - 21 - 56 - 52 -
	70 - 12 - 91 - 39 - 51 - 33 - 7 - 13 - 20 - 2 - 59 - 8 - 64 - 44 - 40 - 101 - 32 -
	41 - 96 - 97 - 48 - 80 - 14 - 37 - 65 - 62 - 28 - 72 - 5 - 11 - 3 - 16 - 53 - 77 -
	93 - 15 - 58 - 82 - 47 - 50 - 45 - 85 - 83 - 86 - 79 - 55 - 38 - 43 - 92 - 76 - 24 -
	42 - 31 - 88 - 6 - 49 - 71 - 81 - 35 - 29 - 67 - 46 - 74 - 36 - 19 - 89 - 87 - 1
3	1 - 47 - 80 - 31 - 99 - 77 - 17 - 4 - 69 - 59 - 52 - 45 - 63 - 53 - 38 - 70 - 19 -
	23 - 79 - 93 - 32 - 41 - 50 - 83 - 13 - 60 - 65 - 37 - 76 - 85 - 35 - 10 - 61 - 5 -
	56 - 11 - 81 - 22 - 27 - 2 - 26 - 24 - 48 - 25 - 88 - 82 - 21 - 75 - 71 - 101 - 43 -
	8 - 62 - 12 - 6 - 84 - 3 - 15 - 51 - 42 - 14 - 28 - 54 - 39 - 67 - 73 - 96 - 78 -
	66 - 18 - 44 - 33 - 20 - 95 - 91 - 55 - 46 - 87 - 16 - 74 - 90 - 49 - 100 - 68 - 64 -
	94 - 98 - 58 - 97 - 34 - 40 - 57 - 92 - 7 - 89 - 30 - 29 - 86 - 72 - 9 - 36 - 1
4	1 - 67 - 33 - 93 - 73 - 49 - 27 - 29 - 74 - 36 - 3 - 37 - 20 - 94 - 18 - 30 - 75 -
	22 - 45 - 58 - 7 - 32 - 100 - 77 - 4 - 53 - 97 - 90 - 56 - 6 - 78 - 39 - 87 - 5 -
	51 - 101 - 72 - 99 - 41 - 76 - 42 - 14 - 70 - 25 - 19 - 98 - 26 - 46 - 68 - 84 - 83 -
	71 - 88 - 89 - 85 - 38 - 63 - 65 - 50 - 52 - 64 - 86 - 81 - 15 - 69 - 82 - 44 - 55 -
	91 - 57 - 12 - 80 - 92 - 17 - 13 - 16 - 28 - 10 - 43 - 95 - 23 - 48 - 47 - 35 - 79 -
	2 - 34 - 62 - 54 - 96 - 60 - 40 - 59 - 31 - 24 - 11 - 21 - 66 - 8 - 61 - 9 - 1
5	1 - 52 - 68 - 30 - 61 - 26 - 49 - 4 - 33 - 43 - 29 - 11 - 46 - 8 - 85 - 94 - 44 -
	14 - 6 - 51 - 40 - 72 - 69 - 54 - 79 - 50 - 89 - 39 - 23 - 27 - 12 - 78 - 80 - 76 -
	37 - 2 - 96 - 101 - 81 - 32 - 95 - 73 - 74 - 35 - 71 - 42 - 57 - 28 - 59 - 31 - 93 -
	77 - 56 - 84 - 91 - 22 - 5 - 21 - 55 - 99 - 25 - 97 - 90 - 53 - 66 - 87 - 20 - 83 -
	64 - 75 - 70 - 62 - 15 - 82 - 65 - 34 - 16 - 100 - 67 - 92 - 7 - 3 - 10 - 98 - 18 -
	19 - 9 - 13 - 58 - 63 - 41 - 88 - 36 - 17 - 86 - 38 - 45 - 60 - 48 - 47 - 24 - 1
6	1 - 41 - 100 - 39 - 64 - 31 - 28 - 58 - 16 - 70 - 62 - 24 - 10 - 57 - 91 - 22 - 48 -
	84 - 96 - 36 - 93 - 56 - 8 - 90 - 30 - 60 - 21 - 35 - 23 - 52 - 47 - 11 - 49 - 99 -
	87 - 45 - 86 - 29 - 66 - 89 - 15 - 4 - 83 - 12 - 14 - 33 - 32 - 54 - 43 - 42 - 44 -
	18 - 73 - 5 - 71 - 26 - 53 - 38 - 2 - 25 - 34 - 94 - 92 - 75 - 76 - 46 - 6 - 65 -
	97 - 61 - 69 - 80 - 78 - 37 - 88 - 27 - 68 - 9 - 50 - 3 - 79 - 82 - 77 - 98 - 59 -
	51 - 20 - 55 - 81 - 95 - 101 - 19 - 74 - 17 - 7 - 63 - 13 - 67 - 72 - 40 - 85 - 1
⋮	⋮ ⋮ ⋮
30	1 - 80 - 87 - 54 - 16 - 96 - 2 - 7 - 99 - 68 - 74 - 47 - 43 - 25 - 8 - 38 - 59 -
	76 - 56 - 50 - 73 - 58 - 48 - 34 - 39 - 71 - 36 - 70 - 11 - 63 - 41 - 79 - 40 - 37 -
	95 - 78 - 42 - 57 - 27 - 86 - 53 - 33 - 20 - 49 - 66 - 88 - 26 - 6 - 97 - 24 - 46 -
	32 - 19 - 100 - 82 - 14 - 28 - 45 - 18 - 64 - 13 - 77 - 85 - 65 - 60 - 29 - 35 - 55 -
	101 - 69 - 15 - 30 - 9 - 61 - 62 - 93 - 44 - 51 - 67 - 91 - 81 - 5 - 75 - 4 - 94 -
	23 - 90 - 21 - 10 - 12 - 22 - 72 - 83 - 89 - 52 - 17 - 3 - 84 - 98 - 31 - 92 - 1

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Berdasarkan Tabel 4.6 menunjukkan indeks hasil pengurutan posisi awal sesuai dengan banyaknya jumlah *swarm*. Proses pengurutan tersebut dilakukan pada *swarm* ke-1 sampai dengan *swarm* ke-30 sehingga solusi awal akan terbentuk. Pada tahap selanjutnya, akan dilakukan perhitungan fungsi tujuan VRPTW yang diperoleh berdsarkan hasil total jarak dengan melakukan proses pembentukan lintasan pada setiap *swarm*. Selain itu, proses tersebut dilakukan dengan memperhitungkan konsep dasar pada VRP yaitu setiap node (titik) dimulai dari depot (pusat distribusi) yang disimbolkan dengan angka 1 dan kembali lagi pada depot tersebut. Sedangkan, pada setiap permintaan pelanggan yang akan dikirim tidak boleh melebihi jumlah maksimum kapasitas kendaraan. Pada penelitian ini, tipe VRPTW memiliki tambahan fungsi kendala yaitu batasan waktu (*time window*). Sebagai gambaran akan dilakukan satu kali perhitungan pada salah satu *swarm* sebagai berikut.

$$1-33-54-48-74-45-28-73-46-35-60-94-17-81-31-69-70-41-3-58-64-85-14-59-22-83-65-93-20-92-96-72-55-16-84-12-91-13-4-23-39-77-37-95-78-61-71-66-75-10-87-9-89-88-34-8-21-30-19-68-32-79-100-44-63-38-98-2-47-40-86-99-25-56-36-24-52-27-6-62-90-57-101-26-76-50-15-29-43-42-80-11-18-67-51-49-7-82-53-5-97-1$$

Posisi awal partikel pada *swarm* ke-1 dalam persoalan *Vehicle Routing Problem with Time Window* (VRPTW) yang dimulai dari depot (1) menuju pelanggan ke-33, pelanggan ke-54, pelanggan ke-48, pelanggan ke-74, pelanggan ke-45, pelanggan ke-28, hingga berlanjut menuju pelanggan ke-97 dan akan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

kembali lagi pada depot (1) sehinggan akan membentuk dalam beberapa rute. Langkah selanjutnya yaitu dengan menjumlahkan *demand* (permintaan) dari setiap pelanggan tersebut dan solusi rute awal akan terbentuk setelah kapasitas maksimum kendaraan terpenuhi. Model matematika fungsi kendala VRPTW untuk *demand* (permintaan) setiap pelanggan pada Persamaan 2.3 adalah sebagai berikut.

$$\sum_{j=1}^{n}\sum_{i=0}^{n} q_i X_{ijk} \leq Q$$

$$\sum_{j=1}^{n}\sum_{i=0}^{n} q_i X_{ijk} \leq 200$$

Berdasarkan dari fungsi kendala VRPTW di atas, maka total *demand*(permintaan) setiap pelanggan tidak boleh melebihi batas kapasitas maksimum kendaraan dimana kapasitas maksimumnya adalah sebesar 200. Pada *swarm* ke-1, kendaraan mengunjugi pelanggan ke-33 terlebih dahulu yang kemudian dilakukan perhitungan untuk waktu tempuh kendaraan dari depot (1) menuju pelanggan ke-33 berdasarkan Tabel 4.3.

Langkah selanjutnya, dengan memperhatikan batasan waktu (*time window*) yang didasarkan pada salah satu fungsi kendala VRPTW pada Persamaan 2.7. *Time windows* terdiri dari *earlist time* (e_i) atau waktu awal pelanggan dan *latest time* (l_i) atau waktu akhir pelanggan. Pada pelanggan ke-33, memiliki *time window* $[e_33, l_{33}]$ dengan $e_{33} = 131$ dan $l_{33} = 161$. Jika waktu tempuh kendaraan dari depot (1) menuju pelanggan ke-6 ($t_{1,33}$) memiliki nilai kurang dari waktu awal $e_{33} = 131$, maka waktu kedatangan pelanggan ke-33 (t_{33}) sama dengan 131 karena kendaraan harus menunggu sampai pelanggan ke-33 dapat menerima pelayanan. Jika waktu tempuh kendaraan dari depot (1) menuju pelanggan ke-33 ($t_{1,33}$) memiliki nilai diantara waktu awal $(e_{33} = 131)$ dan waktu akhir $(l_{33} = 161)$, maka

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

waktu kedatangan pelanggan ke-33 (t_{33}) sama dengan waktu tempuh kendaraan dan langsung dapat melakukan pelayanan kepada pelanggan ke-33. Sedangkan, jika waktu tempuh dari depot (1) menuju pelanggan ke-33 ($t_{1,33}$) memiliki nilai lebih dari $l_{33} = 161$, maka waktu kedatangan pelanggan ke-33 (t_{33}) tidak memenuhi *time window* sehingga harus kembali ke depot (1). Kemudian, dilakukan perhitungan waktu selesai atau total waktu pada pelanggan ke-33. Berikut merupakan gambaran perhitungan secara rinci dari *swarm* ke-1.

1. Perhitungan Rute ke-1

   a. Depot (1) menuju pelanggan ke-33

      1) permintaan pelanggan ke-33 $(q_{33}) = 10$

      2) $[e_{33}, l_{33}] = [131, 161]$

      3) waktu tempuh $(t_{1,33}) = 51.08$

      4) waktu kedatangan $(t_{33}) = 131$

      5) waktu selesai 1

         = waktu kedatangan (t_{33}) + waktu pelayanan (f_{33})

         $= 131 + 10$

      6) Total jarak $1 = 51.08$

      7) Total waktu $1 = 141$

      8) Total Kapasitas $1 = 10$

   b. Pelanggan ke-33 menuju pelanggan ke-54

      1) permintaan pelanggan ke-54 $(q_{54}) = 5$

      2) $[e_{54}, l_{54}] = [91, 121]$

      3) waktu tempuh $(t_{33,54})$

         = total waktu $1 +$ waktu tempuh$(t_{33,54})$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$= 141 + 69.92$

$= 210.92$

4) waktu kedatangan $(t_{54}) = 210.92$

Karena waktu kedatangan melebihi waktu akhir (*latest time*) $l_{54} = 121$, maka kendaraan tidak dapat melakukan pelayanan sehingga harus kembali ke depot (1).

c. Pelanggan ke-33 kembali Depot (1)

1) Total jarak 2

$=$ Total jarak $1 +$ jarak pelanggan ke-33 menuju Depot (1) $(C_{33,1})$

$= 51.08 + 51.08 = 102.16$

2) Total waktu 2

$=$ Total waktu $1 +$ waktu tempuh$(t_{33,1})$

$= 141 + 51.08$

$= 192.08$

3) Total Kapasitas $2 = q_{33} = 10$.

Sehingga, diperoleh rute 1 pada *swarm* ke-4 yaitu $1 - 33 - 1$ dengan total jarak sebesar 102.16, total waktu sebesar 192.08, dan total kapasitas sebesar 10. Langkah selanjutnya adalah menghitung rute yang ke-2.

2. Perhitungan Rute ke-2

a. Depot (1) menuju pelanggan ke-54

1) permintaan pelanggan ke-54 $(q_{54}) = 5$

2) $[e_{54}, l_{54}] = [91, 121]$

3) waktu tempuh $(t_{1,54}) = 20$

4) waktu kedatangan $(t_{54}) = 91$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

5) waktu selesai 1

$=$ waktu kedatangan $(t_{54} +$ waktu pelayanan (f_{54})

$= 91 \ + \ 10$

6) Total jarak $1 = 20$

7) Total waktu $1 = 101$

8) Total Kapasitas $1 = 5$

b. Pelanggan ke-54 menuju pelanggan ke-48

1) permintaan pelanggan ke-48 $(q_{48}) = 10$

2) $[e_{48}, l_{48}] = [45, 75]$

3) waktu tempuh $(t_{54,48})$

$=$ waktu selesai $1 +$ waktu tempuh$(t_{25,26})$

$= 101 + 18.68$

$= 119.68$

4) waktu kedatangan $(t_{48}) = 119.68$

Karena waktu kedatangan melebihi waktu akhir (*latest time*) $l_{48} = 75$, maka kendaraan tidak dapat melakukan pelayanan sehingga harus kembali ke depot (1).

c. Pelanggan ke-54 menuju kembali ke Depot(1)

1) Total jarak 2

$=$ Total jarak $1 +$ jarak pelanggan ke-54 menuju Depot (1) $(C_{54,1})$

$= 20 + 20 = 40$

2) Total waktu 2

$=$ Total waktu $1 +$ waktu tempuh$(t_{54,1})$

$= 101 + 20$

$= 121$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3) Total kapasitas $3 = q_{54} = 5$.

Sehingga, diperoleh rute 2 pada *swarm* ke-1 yaitu $1-54-1$ dengan total jarak sebesar 40, total waktu sebesar 121, dan total kapasitas sebesar 5. Langkah selanjutnya adalah menghitung rute yang ke-3.

3. Perhitungan Rute ke-3

   a. Depot (1) menuju pelanggan ke-48

      1) permintaan pelanggan ke-48 $(q_{48}) = 10$

      2) $[e_{48}, l_{48}] = [45, 75]$

      3) waktu tempuh $(t_{1,48}) = 38.33$

      4) waktu kedatangan $(t_{48}) = 45$

      5) waktu selesai 1

         $=$ waktu kedatangan (t_{48}) $+$ waktu pelayanan (f_{48})

         $= 45 + 10$

         $= 55$

      6) Total jarak $1 = 38.33$

      7) Total waktu $1 = 55$

      8) Total kapasitas $1 = 10$

   b. Pelanggan ke-48 menuju pelanggan ke-74

      1) permintaan pelanggan ke-74 $(q_{74}) = 5$

      2) $[e_{74}, l_{74}] = [77, 107]$

      3) waktu tempuh $(t_{48,74})$ $=$ waktu selesai 1 $+$ waktu tempuh$(t_{48,74})$

         $= 55 + 15 = 70$

      4) waktu kedatangan $(t_{74}) = 77$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

5) waktu selesai 2

$=$ waktu kedatangan $t_{74}) +$ waktu pelayanan (f_{74})

$= 77 + 10$

$= 87$

6) Total jarak 2

$=$ Total jarak $1 +$ jarak pelanggan ke-48 menuju pelanggan ke-74 $(C_{48,74})$

$= 38.33 + 15$

$= 53.33$

7) Total waktu $2 =$ waktu selesai $2 = 87$

8) Total kapasitas $2 = q_{48} + q_{74} = 10 + 5 = 15$

c. Pelanggan ke-74 menuju pelanggan ke-45

1) permintaan pelanggan ke-45 $(q_{45}) = 10$

2) $[e_{45}, l_{45}] = [64, 94]$

3) waktu tempuh $(t_{74,45}) =$ waktu selesai $2 +$ waktu tempuh$(t_{74,45})$

$= 87 + 57.39 = 144.39$

4) waktu kedatangan $(t_{45}) = 144.39$

Karena waktu kedatangan melebihi waktu akhir (*latest time*) $l_{45} = 94$, maka kendaraan tidak dapat melakukan pelayanan sehingga harus kembali ke depot (1).

c. Pelanggan ke-74 kembali Depot (1)

1) Total jarak 3

$=$ Total jarak $2 +$ jarak pelanggan ke-33 menuju Depot (1) $(C_{74,1})$

$= 53.33 + 39.29 = 92.62$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2) Total waktu 2

$= \text{Total waktu } 1 + \text{waktu tempuh}(t_{74,1})$

$= 141 + 39.29$

$= 180.29$

3) Total Kapasitas $2 = q_{48} + q_{74} = 10 + 5$

$= 15.$

Sehingga, diperoleh rute 3 pada *swarm* ke-1 yaitu $1-48-74-1$ dengan total jarak sebesar 92.62, total waktu sebesar 180.29, dan total kapasitas sebesar 15.

Proses perhitungan setiap rute pada *swarm* ke-1 diatas dapat dilanjutkan dengan cara yang sama sepertia pada proses perhitungan pada rute ke-1, rute ke-2, dan rute ke-3, sehingga dapat diperoleh solusi awal dari *swarm* ke-1. Setelah melakukan perhitungan pada *swarm* ke-1 tersebut, dengan cara yang sama dilakukan perhitungan serupa pada *swarm* ke-2, *swarm* ke-3, hingga pada *swarm* ke-30.

Langkah 4 : Menentukan $P_{best(i)}$ awal dan $G_{best(i)}$ awal dari Partikel

1. Pada setiap partikel dari masing-masing *swarm*, maka $P_{best(i)}$ awal nilainya akan sama dengan partikel awal (posisi awal) yang terdapat pada Tabel 4.4. Berdasarkan penerapan algoritma PSO pada perhitungan manual tersebut, maka diperoleh nilai $P_{best(i)}$ awal sebagai berikut.

   $P_{best(1)} = \begin{bmatrix} 1.054 & -3.581 & -1.797 & 4.973 & 2.624 & 4.78 & \cdots & 3.292 \end{bmatrix}$

   Pembentukan $P_{best(2)}$ sampai dengan $P_{best(30)}$ dilakukan dengan cara yang sama untuk setiap partikel yang terdapat pada *swarm* ke-2 sampai dengan *swarm* ke-30.

2. Pada penentuan $G_{best(i)}$ awal dari partikel dilakukan setelah proses evaluasi fungsi tujuan untuk setiap partikel. Selain itu, diperoleh solusi yang sesuai

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan kendala VRPTW yang meliputi jarak minimum, waktu tempuh perjalanan dan waktu pelayanan yang juga minimum, serta menyesuaikan dengan batasan waktu (*time windows*) dan kapasitas kendaraan. Pada perhitungan manual ini, maka diperoleh $G_{best(i)}$ pada *swarm* ke-20 atau $P_{best(23)}$ sebagai berikut.

$$G_{best(i)} = \begin{bmatrix} 2.392 & 3.072 & 1.279 & 1.467 & 0.7656 & -2.634 & \cdots & 1.423 \end{bmatrix}$$

Langkah 5 : Mengupdate Nilai Kecepatan dan Posisi Partikel ke-i pada Setiap Swarm

1. Update Nilai Kecepatan

   Pada langkah ini, update nilai kecepatan dilakukan untuk proses update posisi dan memasukkan nilai inisialisasi set parameter dari algoritma *Particle Swarm Optimization* (PSO) pada langkah ke-1.

   Misal :

   $r_1 = 0.2$

   $r_2 = 0.4$

   $c_1 = 2$

   $c_2 = 2$

   $\omega = 0.7$ (*constant inertia weight*)

   iterasi maksimum $= 1000$

   Setelah melakukan inisialisasi set parameter, tahapan selanjutnya adalah mengupdate kecepatan baru berdasarkan Persamaan 2.19.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$v_{ij}^{(t+1)} = \omega v_{ij}^{(t)} + c_1 r_{1j}^{(t)}(P_{best(i)}^{(t)} - X_{ij}^{(t)}) + c_2 v_{2j}^{t}(G_{best,i} - X_{ij}^{(t)})$$

$$v_1(X_1) = \omega v_0 + c_1 r_1([P_{best(1)}] - [X_1]) + c_2 r_2([G_{best,1}] - [X_1])$$

$$\begin{aligned} v_1(X_1) = {} & 0.7\,[-1.491 \;\; -1.064 \;\; -1.301 \;\; -1.829 \;\; 1.608 \;\; 1.069 \;\; \cdots \;\; -0.721] \\ & + \; (2)\,(0.2)\,([1.054 \;\; 3.581 \;\; -1.797 \;\; 4.973 \;\; 2.624 \;\; 4.78 \;\; \cdots \;\; 3.292] \\ & - \; [1.054 \;\; 3.581 \;\; -1.797 \;\; 4.973 \;\; 2.624 \;\; 4.78 \;\; \cdots \;\; 3.292]) \\ & + \; (2)\,(0.4)\,([2.392 \;\; 3.072 \;\; 1.279 \;\; 1.467 \;\; 0.7656 \;\; -2.634 \;\; \cdots \;\; 1.423] \\ & - \; [1.054 \;\; 3.581 \;\; -1.797 \;\; 4.973 \;\; 2.624 \;\; 4.78 \;\; \cdots \;\; 3.292]) \end{aligned}$$

$$\begin{aligned} v_1(X_1) = {} & [-1.044 \;\; -0.745 \;\; -0.911 \;\; -1.280 \;\; 1.126 \;\; 0.748 \;\; \cdots \;\; -0.505] \\ & + \; [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \cdots \;\; 0] \; + \; [1.070 \;\; 5.322 \;\; 2.461 \;\; -2.085 \;\; -1.487 \\ & -5.391 \;\; -3.868 \cdots \;\; -1.496] \end{aligned}$$

$$v_1(X_1) = [0.294 \;\; 5.907 \;\; 2.165 \;\; -4.786 \;\; -0.732 \;\; -6.666 \;\; \cdots \;\; -2.374]$$

Pada perhitungan $v_1(X_2)$ dan seterusnya dapat dilakukan dengan cara yang sama seperti perhitungan pada $v_1(X_1)$, sehingga akan diperoleh nilai kecepatan baru.

2. Update Posisi Partikel ke-i

   Setelah diperoleh nilai kecepatan baru, maka langkah selanjutnya adalah mengupdate posisi baru partikel ke-i. Setiap partikel nilainya di update dengan menggunakan Persamaan 2.17, sehingga diperoleh perhitungan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

manualnya pada *swarm* ke-1 sebagai berikut.

$$X_1' = v_1(X_1) + (X_1)$$

$$X_1' = [0.294 \;\; 5.907 \;\; 2.165 \;\; -4.786 \;\; -0.732 \;\; -6.666 \;\; \cdots \;\; -2.374]$$

$$+ \; [1.054 \;\; 3.581 \;\; -1.797 \;\; 4.973 \;\; 2.624 \;\; 4.78 \;\; \cdots \;\; 3.292]$$

$$X_1' = [1.348 \;\; 2.3271 \;\; 0.369 \;\; 0.187 \;\; 1.981 \;\; -1.886 \;\; \cdots \;\; 0.198]$$

Berdasarkan perhitungan diatas, maka untuk perhitungan partikel X_2' sampai X_5' dilakukan dengan cara yang sama. Langkah selanjutnya, untuk menentukan posisi partikel baru dilakukan dengan cara membagi nilai baru tersebut X_i' dengan jumlah totalnya pada setiap partikel. Sebagai contoh, pada X_1' memiliki jumlah total sebesar -20.056 sehingga diperoleh nilai baru tersebut sebagai berikut.

$$X_{1baru}' = [-0.067 \;\; -0.116 \;\; -0.018 \;\; -0.009 \;\; -0.094 \;\; 0.094 \;\; -0.046]$$

Untuk perhitungan X_{2baru}' dan seterusnya dilakukan proses perhitungan yang sama yaitu membagi dengan jumlah totalnya. Setelah diperoleh nilai update posisi baru tersebut, maka langkah selanjutnya adalah dengan mengurutkan nilai-nilai tersebut mulai dari yang terkecil hingga yang terbesar. Hasil pengurutan dari X_{1baru}' adalah sebagai berikut.

$$X_{1baru}' = [-0.284 \;\; -0.270 \;\; -0.263 \;\; -0.261 \;\; -0.248 \;\; -0.245 \;\; \cdots \;\; 0.320]$$

Kemudian dari hasil pengurutan tersebut diubah dalam bentuk permutasi(urutan indeks posisi partikel baru) berdasarkan dari nilai X_{1baru}'

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sehingga diperoleh :

$$X'_{1baru} = 1 - 65 - 32 - 88 - 96 - 70 - 15 - 38 - 86 - 21 - 73 - 17 - 51 -$$
$$63 - 91 - 33 - 41 - 12 - 47 - 55 - 9 - 78 - 94 - 14 - 3 - 20 -$$
$$81 - 77 - 6 - 56 - 26 - 23 - 97 - 58 - 10 - 79 - 19 - 2 - 18 -$$
$$48 - 46 - 89 - 16 - 101 - 92 - 95 - 27 - 68 - 4 - 5 - 67 - 72 -$$
$$29 - 64 - 52 - 80 - 44 - 98 - 61 - 40 - 100 - 37 - 30 - 57 -$$
$$36 - 31 - 7 - 11 - 50 - 45 - 39 - 35 - 82 - 75 - 76 - 42 - 54 -$$
$$34 - 87 - 49 - 66 - 74 - 84 - 22 - 62 - 93 - 69 - 85 - 99 - 28 -$$
$$24 - 60 - 43 - 25 - 71 - 90 - 8 - 59 - 13 - 83 - 53 - 1$$

Untuk perhitungan X'_{2baru} dan seterusnya dapat dilakukan dengan caryaa ng sama yaitu dengan mengurutkan dan mengubahnya kedalam bentuk permutasi sebanyak inisialisasi jumlah *swarm*. Indeks permutasi diatas disebut juga dengan solusi awal yang diperoleh dari proses perhitungan dengan menerapkan langkah-langkah tersistematis dari algoritma PSO dengan menggunakan strategi *constant inertia weight*.

Langkah 6: Menentukan P_{best} dan G_{best} dari Nilai Kecepatan dan Posisi Baru Setiap Partikel

Setelah langkah 5 selesai, maka proses selanjutnya adalah evaluasi fungsi tujuan atau nilai fitness seperti pada langkah ke-3. Tujuan dari evaluasi nilai fitness tersebut adalah jika fungsi tujuan $f(C_{ij}) \leq P_{best,i}$, $P_{best,i} = C_{ij}$. Namun, jika $P_{best,i}$ baru tidak lebih baik dari $P_{best,i}$ lama, maka yang diambil adalah $P_{best,i}$ lama sehingga diperoleh $P_{best,i}$ baru dan $G_{best,i}$ baru. Pada perhitungan manual ini,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$P_{best,1}$ lama $=$ X_1 dan $P_{best,1}$ baru $=$ X'_{1baru}. Fungsi tujuan pada persoalan VRPTW ini berdasarkan dari hasil total jarak. Pada $P_{best,1}$ lama memiliki hasil total jarak sebesar 5411.38, sedangkan $P_{best,1}$ baru memiliki hasil total jarak sebesar 5222.89. Berdasarkan hal tersebut $P_{best,1}$ baru memiliki nilai total jarak yang lebih kecil dibandingkan dengan $P_{best,1}$ lama maka yang diambil adalah $P_{best,1}$ baru.

Langkah 7 : Cek *Stopping Criteria*

Pada langkah ini, terdapat kondisi pemberhentian atau (*Stopping Criteria*) seperti yang telah dijelaskan pada Bab 2(Tinjauan Pustaka). Proses perhitungan akan berhenti jika telah mencapai nilai optimal atau mencapai iterasi maksimum. Jika belum optimal, maka kembali pada langkah ke-5. Dalam proses perhitungan manual ini, hanya dilakukan satu kali iterasi sehingga belum mencapai titik optimal. Oleh karena itu, harus dilakukan iterasi berikutnya dengan menggunakan program.

4.2. Analisis Hasil Perhitungan *Algoritma Particle Swarm Optimization* (PSO)

Algoritma *Particle Swarm Optimization* (PSO) dapat diterapkan pada beberapa persoalan. Dalam hal ini, algoritma PSO diimplementasikan terhadap persoalan *Vehicle Routing Problem with Time Window* (VRPTW). Pada perhitungan untuk menentukan solusi optimal, maka akan dilakukan uji coba berdasarkan modifikasi parameter-parameter PSO. Hasilnya akan dianalisis bagaimana pengaruh dari parameter tersebut terhadap persoalan VRPTW. Pada penelitian ini untuk tujuan perbandingan dari hasil algoritma PSO, maka semua simulasi menggunakan set parameter yang sama kecuali bobot inersia (ω).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.2.1 Set Parameter Algoritma PSO dan Strategi *Inertia Weight* (ω)

Set parameter algoritma *Particle Swarm Optimization* (PSO) dalam penelitian ini akan dilakukan beberapa nilai untuk uji coba. Algortima PSO memiliki beberapa parameter diantaranya jumlah *swarm*, *learning rate* (c_1 dan c_2), nilai random (r_1 dan r_2), dan modifikasi dari strategi *inertia weight*. Berikut ini merupakan Tabel set parameter algoritma PSO yang akan digunakan dalam penelitian.

Tabel 4.6 Set Parameter Algoritma PSO

Jumlah Swarm	Iterasi Maksimum	c_1	c_2
30	1000	0.5	1
40	2000	0.7	1.2
60	4000	1	1.4
		1.5	1.6
			1.8
			2

Pada Tabel 4.6 menunjukkan kombinasi parameter yang akan digunakan dalam penelitian ini. Dalam Tabel tersebut jumlah *swarm* yang akan digunakan adalah 30, 40, dan 60. Jumlah *swarm* yang akan digunakan dalam penelitian berdasarkan referensi Talukder (2011). Sedangkan untuk parameter c_1 dan c_2 digunakan nilai uji coba dari 0,5 sampai dengan 2. Berdasarkan penelitian Andries (2007), secara umum nilai yang sering digunakan untuk parameter c_1 dan c_2 adalah 2. Selain itu, strategi *inertia weight* yang akan digunakan berdasarkan hasil penelitian oleh bansal (2011) yang ditunjukkan pada Tabel 2.1. Namun, dalam penelitian ini akan menambahkan strategi *global-local inertia weight*. Berdasarkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

penelitian Arumugam (2006) maka Persamaan strategi *global-local inertia weight* menggunakan persamaan dalam penelitian tersebut(Arumugam et al., 2006).

$$\omega = 1.1\frac{G_{best(i)}}{P_{best(i)}} \tag{4.1}$$

Sedangkan, untuk modifikasi ω dengan model *constriction coefficient* menggunakan Persamaan 2.21. *Constriction coefficient* menggambarkan nilai χ yang setara dengan nilai ω.

4.2.2 Analisis Hasil Implementasi Data RC101

Algoritma *Particle Swarm Optimization* (PSO) dapat diterapkan dalam menyelesaikan persoalan optimasi *Vehicle Routing Problem With Time Window* (VRPTW). Analisis hasil penerapan algoritma PSO tersebut diantaranya analisis hasil berdasarkan jumlah *swarm*, jumlah iterasi maksimum, parameter c_1 dan c_2, dan modifikasi PSO dengan strategi *inertia weight* (ω). Analisis hasil tersebut akan dijelaskan secara merinci berdasarkan penggunaan modifikasi strategi *inertia weight* (ω).

1. Analisis Hasil Berdasarkan Strategi *Constant Inertia Weight*

Implementasi data RC101 dengan menggunakan modifikasi strategi *constant inertia weight* ditunjukkan dalam Lampiran C. Pada Tabel 4.10 di bawah ini menunjukkan 5 hasil percobaan terbaik dari 216 kombinasi percobaan berdasarkan total jarak terpendek.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

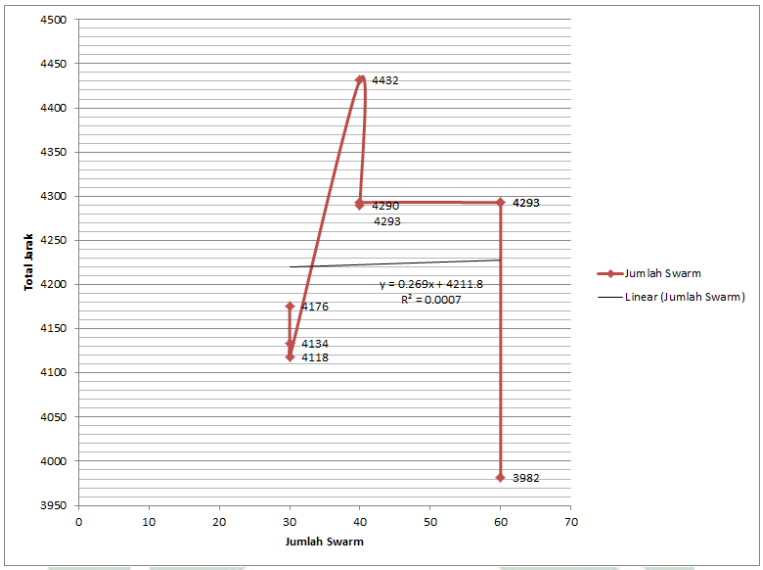
Tabel 4.7 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Constant Inertia Weight*

Percobaan ke-	Jumlah Swarm	Iterasi Maksimum	c_1	c_2	ω	Jumlah Lintasan	Kapasitas Maksimum	Total Jarak	Total Waktu
193	60	4000	0.5	1	0.7	49	200	3879	7899
138	40	4000	1	2	0.7	49	200	3956	8024
195	60	4000	0.5	1.4	0.7	46	200	3967	7787
198	60	4000	0.5	2	0.7	46	200	3967	7544
181	60	2000	1	1	0.7	46	200	3974	7641

Percobaan di atas dilakukan *running* sebanyak 10 kali menggunakan bahasa pemrograman. Inisialisasi posisi awal ditentukan $X_{min} = -5$ dan $X_{max} = 5$. Sedangkan, pada inisialisasi kecepatan awal dilakukan berdasarkan Persamaan 2.18 dengan menggunakan nilai $\varepsilon = 0.2$ sehingga diperoleh $V_{min} = -2$ dan $V_{max} = 2$. Berdasarkan Tabel 4.7 menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke 193, 138, 195, 198, dan 181. Pada percobaan ke 193 menunjukkan jarak terpendek berdasarkan modifikasi *constant inertia weight* sebesar 3879 satuan jarak dengan total waktu 7899 satuan waktu.

Analisis hasil dalam penerapan algoritma PSO untuk menyelesaikan persoalan optimasi VRPTW dilakukan berdasarkan jumlah *swarm*, jumlah iterasi, parameter c_1 dan c_2, serta modifikasi *inertia weight*. Pada Tabel 4.7 menunjukkan jumlah *swarm* 60 yang cenderung memiliki pengaruh terhadap total jarak terpendek. Analisis hasil percobaan berdasarkan jumlah *swarm* digambarkan dalam grafik pada Gambar 4.1.

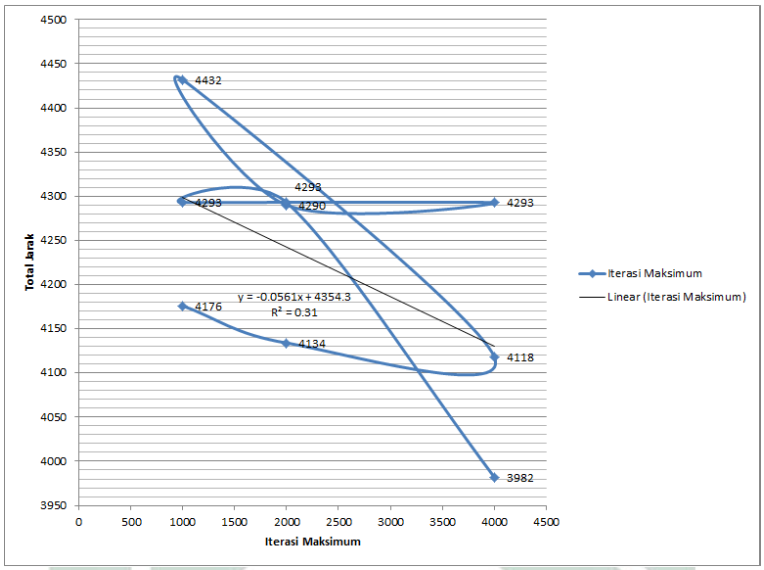
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.1 Grafik Analisis Berdasarkan Jumlah Swarm

Berdasarkan Gambar 4.1 di atas, jumlah *swarm* yang semakin besar tidak memberikan pengaruh yang signifikan terhadap pengoptimalan persoalan *vehicle Routing Problem With Time Window* (VRPTW). Namun, analisis hasil PSO dengan strategi *constant inertia weight* jumlah *swarm* 30 dan 60 mampu menghasilkan solusi optimal. Selain itu, analisis berdasarkan iterasi maksimum yang digunakan dalam penelitian. Iterasi tersebut memberikan pengaruh yang cukup signifikan terhadap total jarak, semakin besar iterasi yang digunakan maka hasilnya akan semakin optimum. Namun, jumlah iterasi yang besar akan berdampak pada waktu komputasi yang semakin lama. Gambar 4.2 di bawah ini menunjukkan grafik analisis hasil percobaan berdasarkan dari iterasi yang digunakan.

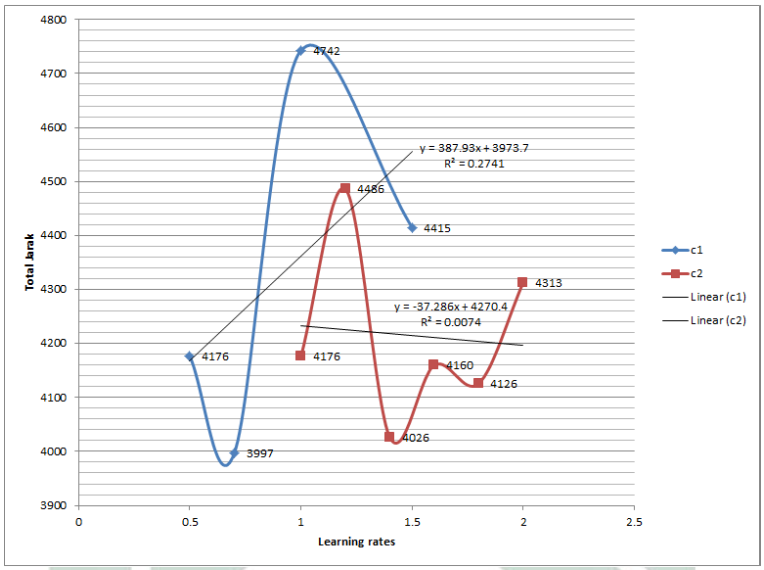
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.2 Grafik Analisis Berdasarkan Iterasi Maksimum

Berdasarkan grafik pada Gambar 4.2 di atas menunjukkan semakin besar jumlah iterasi maka semakin kecil jarak yang terbentuk. Pada iterasi 4000 menunjukkan pengaruh yang signifikan dalam meminimumkan total jarak sehingga dapat memperoleh solusi yang optimal. Analisis hasil PSO berdasarkan parameter *learning rates* yaitu c_1 dan c_2 menggunakan nilai seperti yang ditunjukkan pada Tabel 4.6. Parameter untuk c_1 dengan nilai 0.5 dan 1 memberikan kombinasi dan pengaruh terhadap meminimumkan total jarak. Sedangkan, parameter c_2 cenderung memberikan hasil terbaiknya dengan nilai 1 dan 2. Analisis *learning rates* (c_1,c_2) dijelaskan pada Gambar 4.3.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



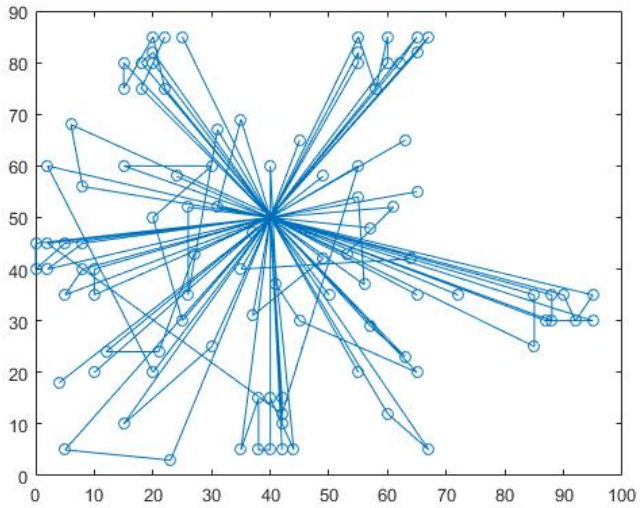
Gambar 4.3 Grafik Analisis Berdasarkan *Learning rates*

Grafik pada Gambar 4.3 di atas menggambarkan analisis hasil PSO untuk parameter c_1 dan c_2. Pada parameter c_1 menunjukkan semakin kecil nilai c_1 maka berpengaruh signifikan terhadap total jarak terpendek. Namun, nilai $c_1 = 1.5$ juga memberikan pengaruh untuk menemukan solusi optimal seperti pada Tabel 4.7 dengan $c_1 = 1.5$ memperoleh percobaan terbaik. Sedangkan, untuk parameter c_2 dengan nilai 1, 1.4, dan 2 memberikan pengaruh yang cukup signifikan terhadap total jarak. Sebagai contoh, ditunjukkan pada Tabel 4.7 menggunakan nilai parameter 1, 1.4, dan 2.

Nilai yang digunakan untuk *constant inertia weight* adalah 0,7. Berdasarkan analisis hasil pada Tabel 4.7 secara keseluruhan kombinasi pada percobaan ke 193 dan 138 memberikan pengaruh signifikan untuk meminimumkan total jarak. Namun, jumlah lintasan yang dihasilkan lebih besar sehingga menyebabkan total waktu cenderung lebih besar pula. Penerapan algoritma *Particle Swarm Optimization*(PSO) dengan modifikasi *constant inertia weight* memperoleh hasil terbaik pada percobaan ke 193. Secara rinci solusi optimal

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

diperoleh pada *swarm* ke 30 dengan rute yang terbentuk sebanyak 49 rute yang terlampir pada Lampiran D. Gambar 4.1 sebagai berikut menggambarkan ilustrasi rute yang terbentuk berdasarkan solusi optimal dari percobaan ke 193.



Gambar 4.4 Ilustrasi Solusi Optimal Percobaan ke 193

Berdasarkan Gambar 4.4 menunjukkan gambar ilustrasi untuk 100 titik (*node*) dengan 1 pusat distribusi (depot) dan simpul menggambarkan lintasan sehingga dapat membentuk rute berdasarkan hasil percobaan ke 193.

2. Analisis Hasil Strategi *Random Inertia Weight*

Implementasi data RC101 dengan menggunakan modifikasi strategi *random inertia weight* ditunjukkan dalam Lampiran C. Persamaan yang digunakan untuk *random inertia weight* tersebut terdapat pada Tabel 2.1. Sebagai contoh akan dilakukan satu kali perhitungan untuk nilai *random inertia weight*.

Misal :

$r = 0.6$ (nilai random)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$t = 1$ ($t =$ iterasi)

$$\omega(t) = 0.5 + \frac{r}{2}$$
$$\omega(1) = 0.5 + \frac{.6}{2}$$
$$\omega(1) = 0.5 + 0.3$$
$$\omega(1) = 0.8$$

Pada perhitungan nilai ω seterusnya dilakukan dengan cara yang sama sebanyak jumlah iterasi. Pada Tabel 4.11 di bawah ini menunjukkan 5 hasil terbaik berdasarkan total jarak terpendek dengan strategi *random inertia weight*.

Tabel 4.8 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Random Inertia Weight*

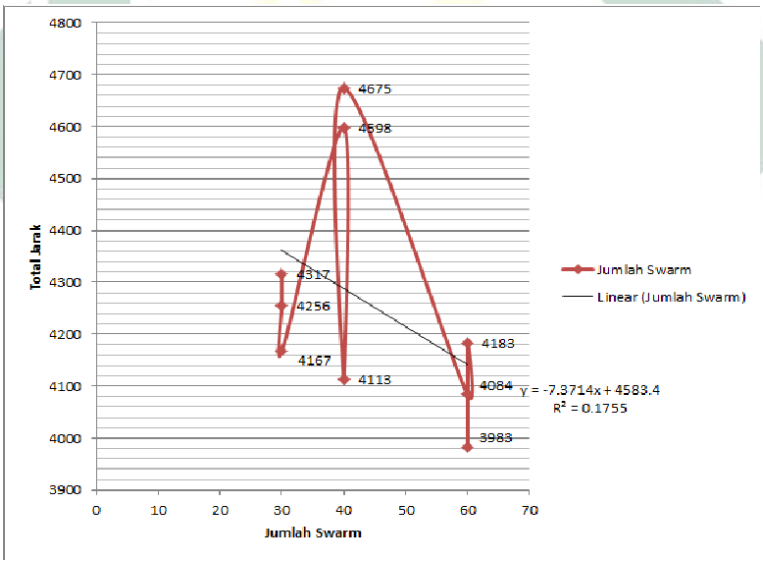
Percobaan ke-	Jumlah Swarm	Iterasi Maksimum	c_1	c_2	Jumlah Lintasan	Kapasitas Maksimum	Total Jarak	Total Waktu
150	60	1000	0.5	2	45	200	3821	7412
195	60	4000	0.5	1.4	47	200	3876	7709
180	60	2000	0.7	2	48	200	3894	7709
67	30	4000	1.5	1	45	200	3939	7502
96	40	1000	1.5	2	47	200	3941	7674

Tabel 4.8 di atas merupakan hasil percobaan yang dilakukan *running* kurang lebih sebanyak 10 kali dengan menggunakan bahasa pemrograman. Persamaan untuk *random inertia weight* dapat dilihat pada Tabel 2.1. Inisialisasi posisi awal ditentukan $X_{min} = -5$ dan $X_{max} = 5$. Sedangkan, pada inisialisasi kecepatan awal dilakukan berdasarkan Persamaan 2.18 dengan menggunakan nilai $\varepsilon = 0.2$ sehingga diperoleh $V_{min} = -2$ dan $V_{max} = 2$. Berdasarkan Tabel 4.8

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke 150,195, 180, 67, dan 96. Hasil percobaan dengan jarak terpendek adalah pada percobaan ke 150 yang menghasilkan total jarak sebesar 3821 satuan jarak, dan total waktu 7412 satuan waktu.

Analisis hasil berdasarkan jumlah *swarm* dapat dilihat pada Tabel 4.8 yang menunjukkan dari 5 percobaan terbaik 3 diantaranya menggunakan jumlah *swarm* 60 yang mempunyai pengaruh terhadap total jarak terpendek. Analisis hasil percobaan berdasarkan jumlah *swarm* digambarkan dalam grafik pada Gambar 4.5.



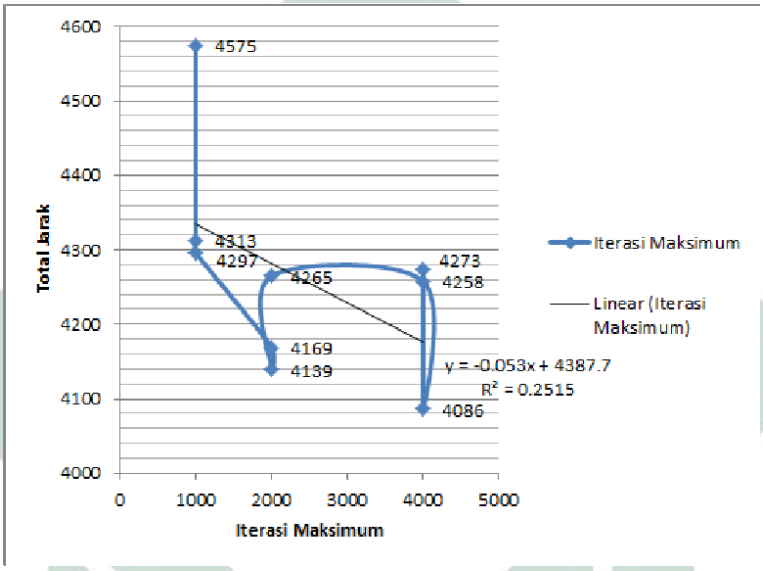
Gambar 4.5 Grafik Analisis Berdasarkan Jumlah Swarm

Berdasarkan Gambar 4.5 di atas, jumlah *swarm* yang semakin besar cenderung memberikan pengaruh yang signifikan terhadap pengoptimalan persoalan *vehicle Routing Problem With Time Window* (VRPTW). Pada grafik tersebut juga menggambarkan jumlah *swarm* 30 dan 40 dapat menghasilkan solusi optimal jika diimbangi iterasi yang besar. Oleh karena itu, analisis hasil PSO dengan strategi *constant inertia weight* jumlah *swarm* 30, 40 dan 60 mampu

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menghasilkan solusi optimal seperti pada Tabel 4.8.

Analisis hasil berdasarkan jumlah iterasi yang digunakan yaitu 1000 dan 4000 memiliki solusi optimum terhadap total jarak terpendek. Pada strategi *random inertia weight* ini, jumlah iterasi 1000 dapat menghasilkan jarak paling kecil. Gambar 4.6 berikut ini menunjukkan grafik analisis hasil percobaan berdasarkan dari iterasi pada strategi *random inertia weight*.



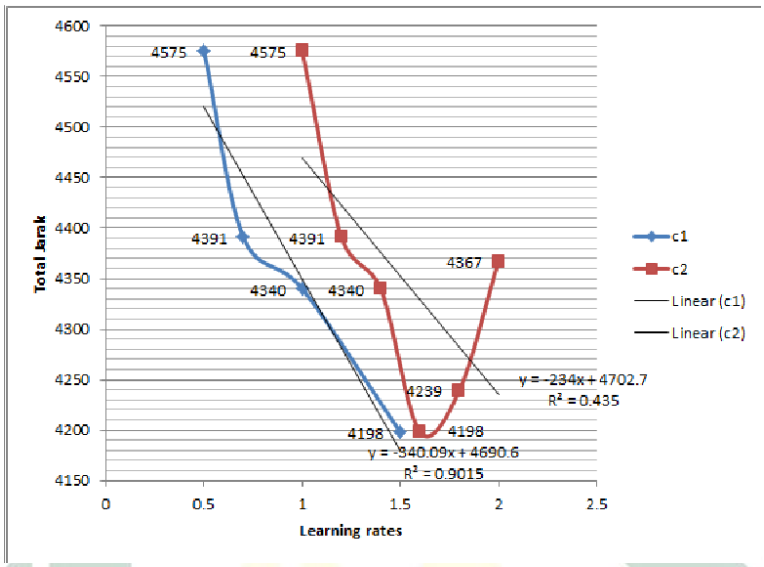
Gambar 4.6 Grafik Analisis Berdasarkan Jumlah Iterasi

Berdasarkan grafik pada Gambar 4.6 di atas menunjukkan semakin besar jumlah iterasi maka semakin kecil jarak yang terbentuk. Namun, pada iterasi 1000 menunjukkan pengaruh dalam meminimumkan total jarak jika diimbangi dengan parameter-parameter pendukung lainnya sehingga dapat memperoleh solusi yang optimal.

Analisis hasil PSO berdasarkan parameter *learning rates* yaitu c_1 dan c_2 menggunakan nilai seperti yang ditunjukkan pada Tabel 4.6. Parameter untuk c_1 dengan nilai 0.5 dan 1.5 memberikan kombinasi dan pengaruh cukup signifikan terhadap meminimumkan total jarak. Sedangkan pada parameter c_2 memberikan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

hasil terbaiknya dengan nilai 1 dan 2. Analisis hasil berdasarkan *learning rates* (c_1,c_2) dijelaskan pada Gambar 4.3.

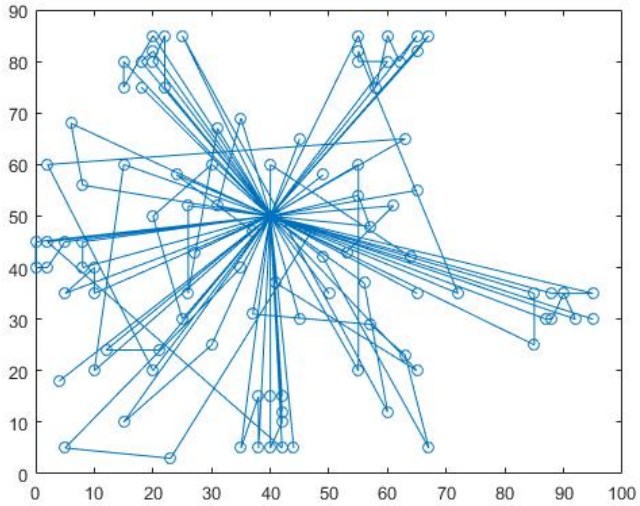


Gambar 4.7 Grafik Analisis Berdasarkan Learning rates

Grafik pada Gambar 4.7 di atas menggambarkan analisis hasil PSO untuk parameter c_1 dan c_2. Pada parameter c_1 menunjukkan semakin besar nilai c_1 maka berpengaruh signifikan terhadap total jarak terpendek. Sedangkan, untuk parameter c_2 dengan nilai semakin besar nilai c_2 maka akan memberikan pengaruh yang cukup signifikan terhadap total jarak. Sebagai contoh, ditunjukkan pada Tabel 4.8 menggunakan nilai parameter 1, 1.4, dan 2. Sebagai contoh, pada percobaan ke 150 yang menggunakan kombinasi parameter $c_1 = 0.5$ dan $c_2 = 2$ dengan strategi *ramdom inertia weight* memperoleh hasil total jarak paling kecil.

Penerapan algoritma *Particle Swarm Optimization*(PSO) dengan modifikasi *random inertia weight* memperoleh hasil terbaik pada percobaan ke 150. Secara rinci solusi optimal diperoleh pada *swarm* ke 9 dengan rute yang terbentuk sebanyak 45 rute yang terlampir pada Lampiran D. Gambar 4.8 sebagai berikut menggambarkan ilustrasi rute yang terbentuk berdasarkan solusi optimal dari percobaan ke 150.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.8 Ilustrasi Solusi Optimal Percobaan ke 150

Berdasarkan Gambar 4.8 menunjukkan gambar ilustrasi untuk 100 titik (*node*)dengan 1 pusat distribusi (depot) dan simpul menggambarkan lintasan sehingga dapat membentuk rute berdasarkan hasil percobaan ke 150.

3. Analisis Hasil Berdasarkan Strategi *Linear Decreasing Inertia Weight*

Implementasi data RC101 dengan menggunakan modifikasi strategi *linear decerasing inertia weight* ditunjukkan dalam Lampiran C. Persamaan yang digunakan untuk strategi *linear decerasing inertia weight* ditunjukkan pada Tabel 2.1 untuk modifikasi nilai *inertia weight*. Sebagai contoh akan dilakukan satu kali perhitungan untuk nilai *linear decreasing inertia weight*.

Misal :

$\omega_{max} = 0.9$

$\omega_{min} = 0.4$

$t = 1$ ($t =$ iterasi)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$t_{max} = 1000$ (iterasi maksimum)

$$\omega(t) = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} \times t$$
$$\omega(1) = 0.9 - \frac{0.9 - 0.4}{1000} \times 5$$
$$\omega(1) = 0.9 - \frac{0.5}{1000} \times 1$$
$$\omega(1) = 0.9 - 0.0005$$
$$\omega(1) = 0.8995$$

Pada perhitungan nilai ω seterusnya dilakukan dengan cara yang sama sebanyak jumlah iterasi. Pada Tabel 4.12 di bawah ini menunjukkan 5 hasil terbaik berdasarkan total jarak terpendek dengan strategi *linear decreasing inertia weight*.

Tabel 4.9 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Linear Decreasing Inertia Weight*

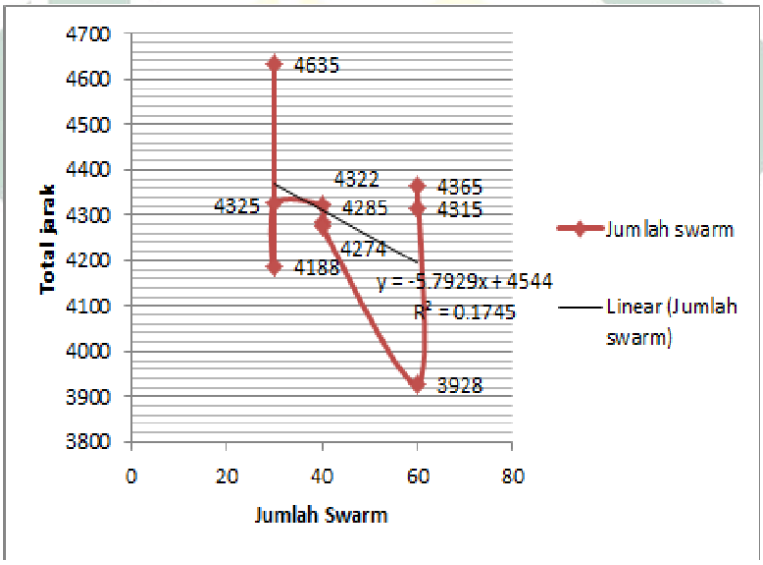
Percobaan ke-	Jumlah Swarm	Iterasi Maksimum	c_1	c_2	Jumlah Lintasan	Kapasitas Maksimum	Total Jarak	Total Waktu
198	60	4000	0.5	2	49	200	3870	7876
146	60	1000	0.5	1.2	49	200	3928	7923
177	60	2000	0.7	1.4	48	200	3964	7914
142	40	4000	1.5	1.6	48	200	3971	7882
193	60	4000	0.5	1	46	200	3983	7725

Berdasarkan Tabel 4.9 di atas merupakan hasil percobaan yang dilakukan *running* kurang lebih sebanyak 10 kali dengan menggunakan bahasa pemrograman. Inisialisasi posisi awal ditentukan $X_{min} = -5$ dan $X_{max} = 5$. Sedangkan, pada inisialisasi kecepatan awal dilakukan berdasarkan Persamaan 2.18 dengan menggunakan nilai $\varepsilon = 0.2$ sehingga diperoleh

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$V_{min} = -2$ dan $V_{max} = 2$. Tabel 4.9 menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke 198,146, 177, 142, dan 193. Hasil percobaan dengan jarak terpendek adalah pada percobaan ke 198 yang menghasilkan total jarak sebesar 3870 satuan jarak, dan total waktu 7876 satuan waktu.

Analisis hasil berdasarkan jumlah *swarm* dapat dilihat pada Tabel 4.9 yang menunjukkan dari 5 percobaan terbaik 3 diantaranya menggunakan jumlah *swarm* 60 yang mempunyai pengaruh terhadap total jarak terpendek. Analisis hasil percobaan berdasarkan jumlah *swarm* digambarkan dalam grafik pada Gambar 4.9.



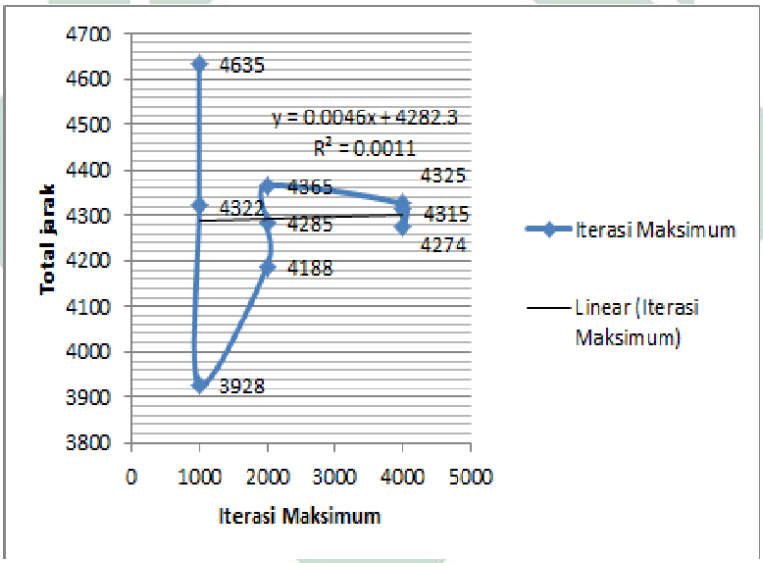
Gambar 4.9 Grafik Analisis Berdasarkan Jumlah Swarm

Berdasarkan Gambar 4.9 di atas, jumlah *swarm* yang semakin besar cenderung memberikan pengaruh yang signifikan terhadap pengoptimalan persoalan *vehicle Routing Problem With Time Window* (VRPTW). Pada grafik tersebut juga menggambarkan jumlah *swarm* 30 dan 40 dapat menghasilkan solusi optimal jika diimbangi iterasi yang besar. Namun pada jumlah *swarm* 60 dapat menghasilkan hasil paling optimum dengan menggunakan jumlah iterasi yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

besar. Oleh karena itu, analisis hasil PSO dengan strategi *linear decreasing inertia weight* jumlah *swarm* 30, 40 dan 60 mampu menghasilkan solusi optimal seperti pada Tabel 4.9.

Analisis hasil berdasarkan jumlah iterasi yang digunakan yaitu 1000 dan 4000 memiliki solusi optimum terhadap total jarak terpendek. Pada strategi *linear decreasing inertia weight*, jumlah iterasi 1000 dapat menghasilkan jarak paling dengan kombinasi jumlah *swarm* yang besar. Gambar 4.10 berikut ini menunjukkan grafik analisis hasil percobaan berdasarkan dari iterasi pada strategi *linear decreasing inertia weight*.



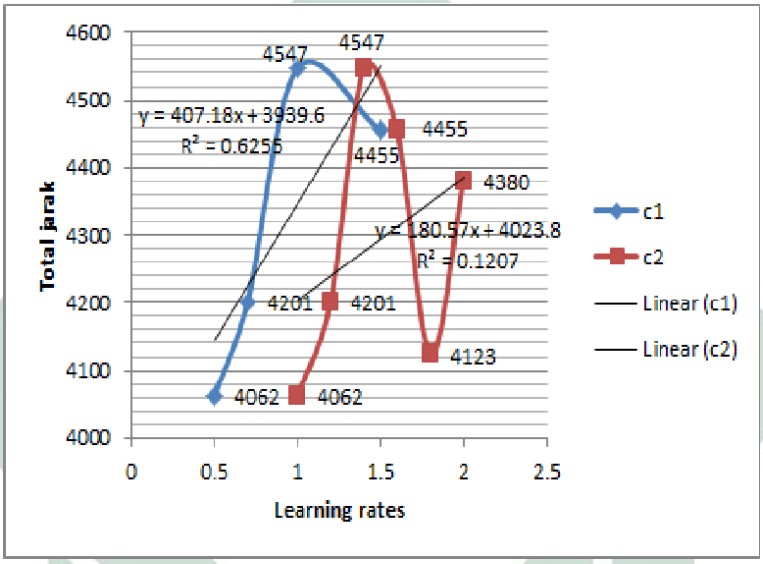
Gambar 4.10 Grafik Analisis Berdasarkan Jumlah Iterasi

Berdasarkan grafik pada Gambar 4.10 di atas menunjukkan semakin besar jumlah iterasi maka semakin kecil jarak yang terbentuk. Namun, pada iterasi 1000 menunjukkan pengaruh yang signifikan dalam meminimumkan total jarak jika diimbangi dengan jumlah *swarm*yang besar sehingga dapat memperoleh solusi yang optimal.

Analisis hasil PSO berdasarkan parameter *learning rates* yaitu c_1 dan c_2 menggunakan nilai seperti yang ditunjukkan pada Tabel 4.6. Parameter untuk c_1

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan nilai 0.5 yang terdapat pada Tabel 4.9 memberikan kombinasi dan pengaruh terhadap meminimumkan total jarak. Sedangkan pada parameter c_2 pada *linear decreasing inertia weight* hampir semua set parameter c_2 pada Tabel 4.6 mempengaruhi total jarak yang tidak dominan pada 1 atau 2 nilai parameter saja. Analisis hasil berdasarkan *learning rates* (c_1,c_2) secara keseluruhan dijelaskan pada Gambar 4.11.



Gambar 4.11 Grafik Analisis Berdasarkan Learning rates

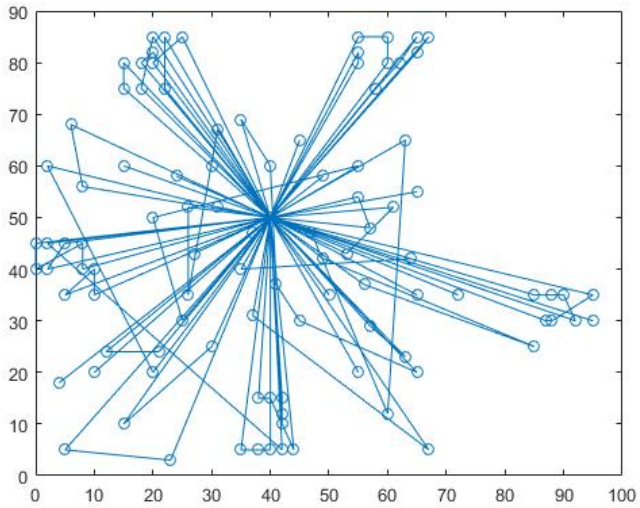
Grafik pada Gambar 4.1 di atas menggambarkan analisis hasil PSO untuk parameter c_1 dan c_2. Pada parameter c_1 menunjukkan semakin kecil nilai c_1 maka berpengaruh signifikan terhadap total jarak terpendek. Sedangkan, untuk parameter c_2 dengan nilai 1, 1.2, dan 1.8 memberikan pengaruh yang cukup signifikan terhadap total jarak. Sebagai contoh, ditunjukkan pada Tabel 4.9 menggunakan nilai parameter 1, 1.4, dan 1.8.

Analisis hasil berdasarkan strategi *linear decreasing inertia weight* memberikan pengaruh yang cukup signifikan, namun total jarak terpendeknya cenderung lebih besar dibandingkan menggunakan strategi *random inertia weight*.

Penerapan algoritma *Particle Swarm Optimization*(PSO) dengan modifikasi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

random inertia weight memperoleh hasil terbaik pada percobaan ke 150. Secara rinci solusi optimal diperoleh pada *swarm* ke 2 dengan rute yang terbentuk sebanyak 49 rute yang terlampir pada Lampiran D. Gambar 4.12 sebagai berikut menggambarkan ilustrasi rute yang terbentuk berdasarkan solusi optimal dari percobaan ke 198.



Gambar 4.12 Ilustrasi Solusi Optimal Percobaan ke 198

Berdasarkan Gambar 4.12 menunjukkan gambar ilustrasi untuk 100 titik (*node*)dengan 1 pusat distribusi (depot) dan simpul menggambarkan lintasan sehingga dapat membentuk rute berdasarkan hasil percobaan ke 198.

4. Analisis Hasil Berdasarkan Strategi *Chaotic Decreasing Inertia Weight*

Implementasi data RC101 dengan menggunakan modifikasi strategi *chaotic inertia weight* ditunjukkan dalam Lampiran C. Persaman yang digunakan untuk strategi *Chaotic Inertia Weight* terdapat pada Tabel 2.1. Sebagai contoh akan dilakukan satu kali perhitungan untuk nilai *chaotic inertia weight*.

Misal :

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$r \ = \ 0.6$ (nilai random)

$\omega(0) \ = \ 0.9$

$\omega(t_{max}) \ = \ 0.4$

$t \ = \ 1$ ($t =$ iterasi)

$t_{max} \ = \ 1000$ (iterasi maksimum)

dengan nilai z sebagai berikut:

$z \ = \ 4r(1-r)$

$z \ = \ 4(0.6)\ (1 \ - \ 0.6)$

$z \ = \ (0.24)\ (0.4)$

$z \ = \ 0.096$

sehingga diperoleh:

$$\omega(t) = (\omega(0) - \omega(t_{max}))(\frac{t_{max} - t}{t_{max}}) + \omega(t_{max})z$$

$$\omega(1) = (0.9 - 0.4)(\frac{1000 - 1}{1000}) + (0.4)0.096$$

$$\omega(1) = 0.499 + 0.038$$

$$\omega(1) = 0.537$$

Pada perhitungan nilai ω seterusnya dilakukan dengan cara yang sama sebanyak jumlah iterasi. Sedangkan, pada Tabel 4.13 di bawah ini menunjukkan 5 hasil terbaik berdasarkan total jarak terpendek.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

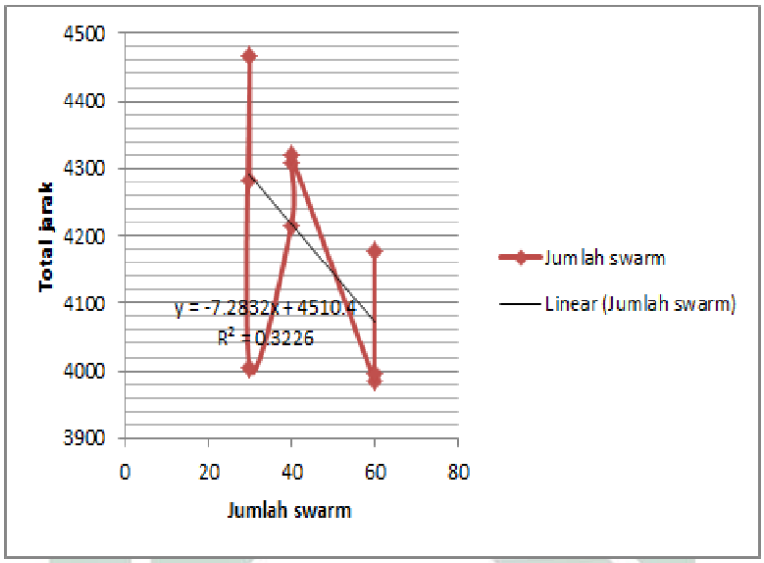
Tabel 4.10 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Chaotic Inertia Weight*

Percobaan ke-	Jumlah Swarm	Iterasi Maksimum	c_1	c_2	Jumlah Lintasan	Kapasitas Maksimum	Total Jarak	Total Waktu
114	40	2000	1	2	45	200	3942	7396
202	60	4000	0.7	1.6	46	200	3944	7702
146	60	1000	0.5	1.2	48	200	3986	7688
12	30	1000	0.7	2	48	200	3993	7814
170	60	2000	0.5	1.2	47	200	3995	7685

Berdasarkan Tabel 4.10 di atas merupakan hasil percobaan yang dilakukan *running* kurang lebih sebanyak 10 kali dengan menggunakan bahasa pemrograman. Inisialisasi posisi awal ditentukan $X_{min} = -5$ dan $X_{max} = 5$. Sedangkan, pada inisialisasi kecepatan awal dilakukan berdasarkan Persamaan 2.18 dengan menggunakan nilai $\varepsilon = 0.2$ sehingga diperoleh $V_{min} = -2$ dan $V_{max} = 2$. Tabel 4.10 menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke 114, 202, 146, 12, dan 170. Hasil percobaan dengan jarak terpendek adalah pada percobaan ke 114 yang menghasilkan total jarak sebesar 3942 satuan jarak, dan total waktu 7342 satuan waktu.

Analisis hasil berdasarkan jumlah *swarm* dapat dilihat pada Tabel 4.10 yang menunjukkan dari 5 percobaan terbaik 3 diantaranya menggunakan jumlah *swarm* 60 yang mempunyai pengaruh terhadap total jarak terpendek. Analisis hasil percobaan berdasarkan jumlah *swarm* digambarkan dalam grafik pada Gambar 4.13.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

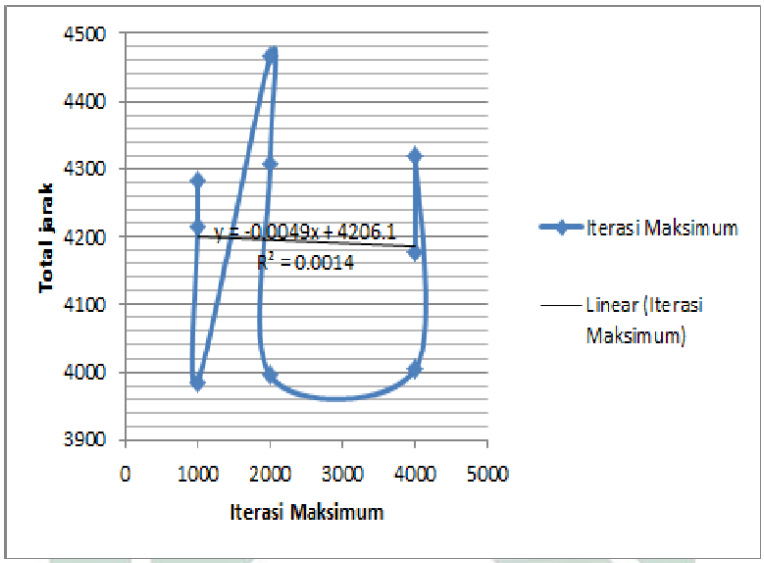


Gambar 4.13 Grafik Analisis Berdasarkan Jumlah Swarm

Berdasarkan Gambar 4.13 di atas, jumlah *swarm* yang semakin besar memberikan pengaruh yang signifikan terhadap pengoptimalan persoalan *vehicle Routing Problem With Time Window* (VRPTW). Pada grafik tersebut juga menggambarkan jumlah *swarm* 30 dan 40 dapat menghasilkan solusi optimal jika diimbangi iterasi yang besar. Oleh karena itu, analisis hasil PSO dengan strategi *linear decreasing inertia weight* jumlah *swarm* 30, 40 dan 60 mampu menghasilkan solusi optimal seperti pada Tabel 4.10.

Analisis hasil berdasarkan jumlah iterasi yang digunakan yaitu 1000 dan 2000 pada Tabel 4.10 mampu memperoleh solusi optimum terhadap total jarak terpendek. Pada strategi *chaotic decreasing inertia weight*, jumlah iterasi 1000 mampu menghasilkan jarak paling pendek dengan kombinasi jumlah *swarm* yang besar. Gambar 4.14 berikut ini menunjukkan grafik analisis hasil percobaan berdasarkan dari iterasi pada strategi *chaotic inertia weight*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

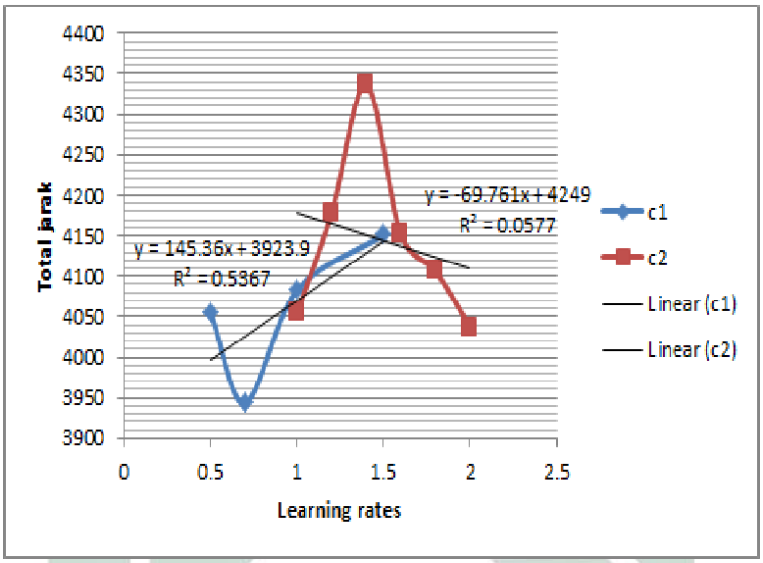


Gambar 4.14 Grafik Analisis Berdasarkan Jumlah Iterasi

Berdasarkan grafik pada Gambar 4.14 di atas menunjukkan semakin besar jumlah iterasi maka semakin kecil jarak yang terbentuk. Namun, pada iterasi 1000 dan 2000 menunjukkan pengaruh yang signifikan dalam meminimumkan total jarak jika diimbangi dengan jumlah *swarm* yang besar sehingga dapat memperoleh solusi yang optimal.

Analisis hasil PSO berdasarkan parameter *learning rates* yaitu c_1 dan c_2 menggunakan nilai seperti yang ditunjukkan pada Tabel 4.6. Parameter untuk c_1 dengan nilai 0.5 dan 0.7 yang terdapat pada Tabel 4.10 merupakan kombinasi yang memberi pengaruh terhadap total jarak. Sedangkan pada parameter c_2 pada *chaotic inertia weight* nilai 1.2 dan 2 dominan memberikan pengaruh terhadap total jarak. Analisis hasil berdasarkan *learning rates* (c_1,c_2) secara keseluruhan dijelaskan pada Gambar 4.15.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

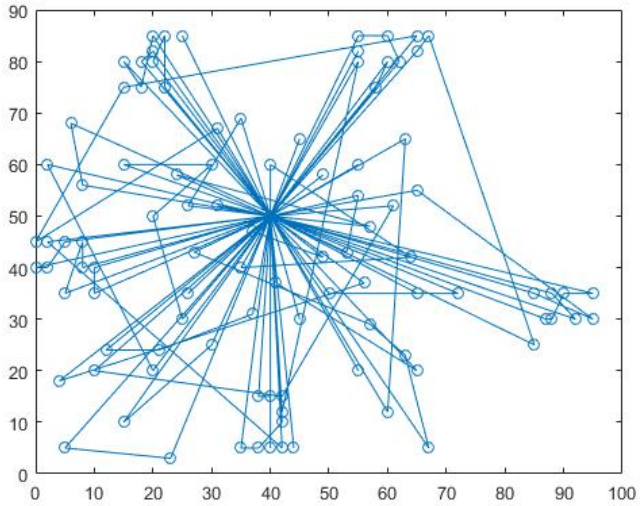


Gambar 4.15 Grafik Analisis Berdasarkan Learning rates

Grafik pada Gambar 4.15 di atas menggambarkan analisis hasil PSO untuk parameter c_1 dan c_2. Pada parameter c_1 menunjukkan semakin kecil nilai c_1 maka berpengaruh signifikan terhadap total jarak terpendek. Sedangkan, untuk parameter c_2 dengan nilai 1 dan 2 memberikan pengaruh yang cukup signifikan terhadap total jarak.

Penerapan algoritma *Particle Swarm Optimization*(PSO) dengan modifikasi *random inertia weight* memperoleh hasil terbaik pada percobaan ke 114. Secara rinci solusi optimal diperoleh pada *swarm* ke 39 dengan rute yang terbentuk sebanyak 45 rute yang terlampir pada Lampiran D. Gambar 4.16 sebagai berikut menggambarkan ilustrasi rute yang terbentuk berdasarkan solusi optimal dari percobaan ke 114.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.16 Ilustrasi Solusi Optimal Percobaan ke 114

Berdasarkan Gambar 4.16 menunjukkan gambar ilustrasi untuk 100 titik (*node*)dengan 1 pusat distribusi (depot) dan simpul menggambarkan lintasan sehingga dapat membentuk rute berdasarkan hasil percobaan ke 114.

5. Analisis Hasil Berdasarkan Strategi *Global-Local Inertia Weight*

Implementasi data RC101 dengan menggunakan modifikasi strategi *global-local inertia weight* ditunjukkan dalam Lampiran C. Persaman yang digunakan untuk strategi *Global-Local Inertia Weight* terdapat pada Persamaan 4.1. Dalam Persamaan 4.1 tersebut melibatkan nilai personal best (*Pbest*) dan global best (*Gbest*) dengan *Pbest* adalah hasil terbaik dari setiap partikel dan *Gbest* adalah hasil terbaik yang diperoleh dari *Pbest*. Pada Tabel 4.14 di bawah ini menunjukkan 5 hasil terbaik berdasarkan total jarak terpendek.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

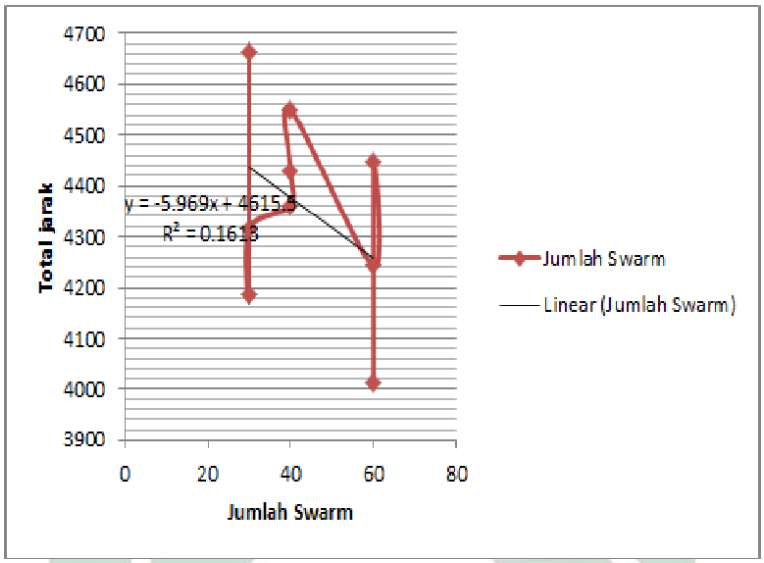
Tabel 4.11 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Global-Local Inertia Weight*

Percobaan ke-	Jumlah Swarm	Iterasi Maksimum	c_1	c_2	Jumlah Lintasan	Kapasitas Maksimum	Total Jarak	Total Waktu
18	30	1000	1	2	49	200	3959	7809
194	60	4000	0.5	1.2	49	200	4011	8023
160	60	1000	1	1.6	50	200	4030	7942
65	30	4000	1	1.8	45	200	4049	7587
198	60	4000	0.5	2	49	200	4061	8027

Berdasarkan Tabel 4.11 merupakan hasil percobaan yang dilakukan *running* kurang lebih sebanyak 10 kali dengan menggunakan bahasa pemrograman. Persamaan untuk *random inertia weight* dapat dilihat pada Tabel 2.1. Inisialisasi posisi awal ditentukan $X_{min} = -5$ dan $X_{max} = 5$. Sedangkan, pada inisialisasi kecepatan awal dilakukan berdasarkan Persamaan 2.18 dengan menggunakan nilai $\varepsilon = 0.2$ sehingga diperoleh $V_{min} = -2$ dan $V_{max} = 2$. Tabel 4.11 menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke 18, 194, 160, 65, dan 198. Hasil percobaan dengan jarak terpendek adalah pada percobaan ke 18 yang menghasilkan total jarak sebesar 3959 satuan jarak, dan total waktu 7809 satuan waktu.

Analisis hasil berdasarkan jumlah *swarm* dapat dilihat pada Tabel 4.11 yang menunjukkan dari 5 percobaan terbaik 3 diantaranya menggunakan jumlah *swarm* 30 dan 60 mampu memperoleh hasil optimal jarak terpendek. Analisis hasil percobaan berdasarkan jumlah *swarm* digambarkan dalam grafik pada Gambar 4.17.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

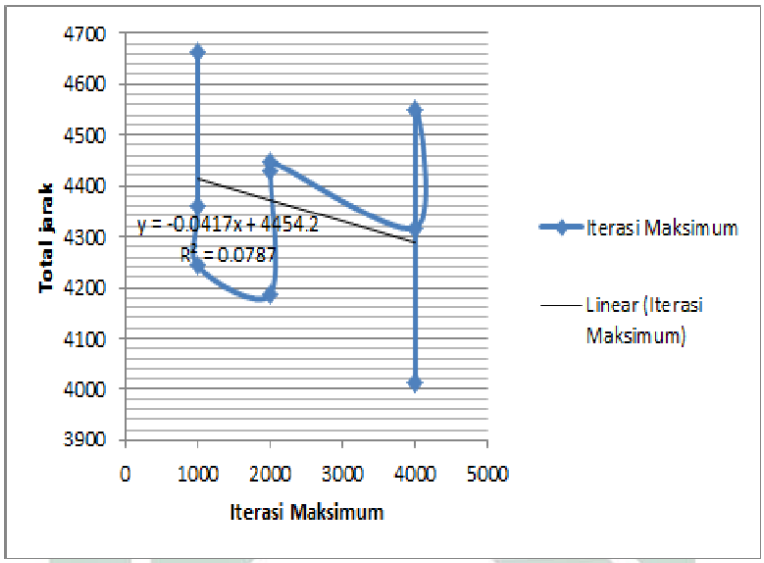


Gambar 4.17 Grafik Analisis Berdasarkan Jumlah Swarm

Berdasarkan Gambar 4.17 di atas, jumlah *swarm* yang semakin besar dapat memberikan pengaruh yang signifikan terhadap pengoptimalan persoalan *vehicle Routing Problem With Time Window* (VRPTW). Namun,grafik tersebut menggambarkan jumlah *swarm* 30 dan 60 yang dapat menghasilkan solusi optimal jika diimbangi dengan iterasi yang besar seperti contoh hasil percobaan pada Tabel 4.11.

Analisis hasil berdasarkan jumlah iterasi yang digunakan yaitu 1000 dan 4000 pada Tabel 4.11 mampu memperoleh solusi optimum terhadap total jarak terpendek. Pada strategi *global-local decreasing inertia weight*, jumlah iterasi 1000 mampu menghasilkan solusi optimum dengan kombinasi jumlah *swarm* yang besar. Gambar 4.14 berikut ini menunjukkan grafik analisis hasil percobaan berdasarkan dari iterasi pada strategi *chaotic inertia weight*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

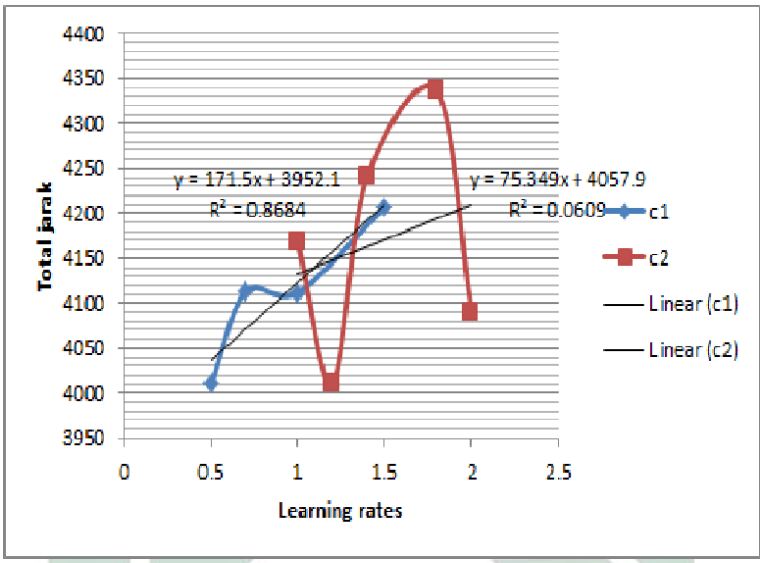


Gambar 4.18 Grafik Analisis Berdasarkan Jumlah Iterasi

Berdasarkan grafik pada Gambar 4.14 di atas menunjukkan semakin besar jumlah iterasi belum tentu menghasilkan total jarak yang minimum. Namun, pada iterasi 1000, 2000 maupun 4000 menunjukkan pengaruh yang signifikan dalam meminimumkan total jarak jika diimbangi dengan jumlah *swarm* yang besar sehingga dapat memperoleh solusi yang optimal.

Analisis hasil PSO berdasarkan parameter *learning rates* yaitu c_1 dan c_2 menggunakan nilai seperti yang ditunjukkan pada Tabel 4.6. Parameter untuk c_1 dengan nilai 0.5 dan 1 yang terdapat pada Tabel 4.11 merupakan kombinasi yang memberi pengaruh terhadap total jarak. Sedangkan pada parameter c_2 pada *chaotic inertia weight* nilai 1.2 dominan memberikan pengaruh terhadap total jarak. Analisis hasil berdasarkan *learning rates* (c_1,c_2) secara keseluruhan dijelaskan pada Gambar 4.19.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

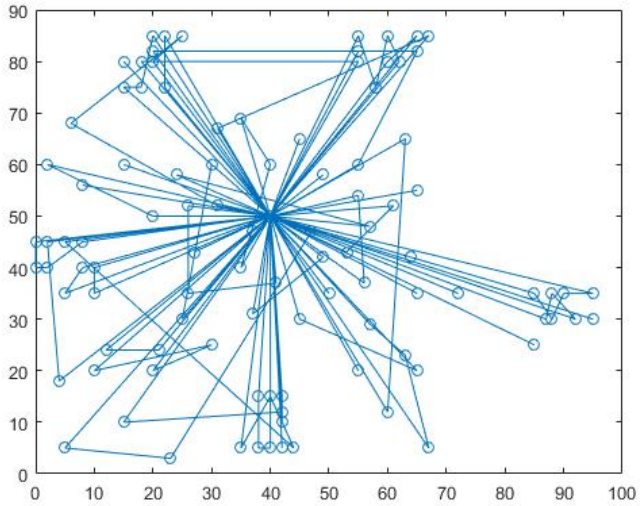


Gambar 4.19 Grafik Analisis Berdasarkan Learning rates

Grafik pada Gambar 4.19 di atas menggambarkan analisis hasil PSO untuk parameter c_1 dan c_2. Pada parameter c_1 menunjukkan semakin kecil nilai c_1 maka berpengaruh signifikan terhadap total jarak terpendek. Sedangkan, untuk parameter c_2 dengan nilai 1.2 dan 2 memberikan pengaruh yang cukup signifikan terhadap total jarak.

Penerapan algoritma *Particle Swarm Optimization* (PSO) dengan modifikasi *global-local inertia weight* memperoleh hasil terbaik pada percobaan ke 114. Secara rinci solusi optimal diperoleh pada *swarm* ke 11 dengan rute yang terbentuk sebanyak 49 rute yang terlampir pada Lampiran D. Gambar 4.20 sebagai berikut menggambarkan ilustrasi rute yang terbentuk berdasarkan solusi optimal dari percobaan ke 114.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.20 Ilustrasi Solusi Optimal Percobaan ke 18

Berdasarkan Gambar 4.20 menunjukkan gambar ilustrasi untuk 100 titik (*node*)dengan 1 pusat distribusi (depot) dan simpul menggambarkan lintasan sehingga dapat membentuk rute berdasarkan hasil percobaan ke 18.

6. Analisis Hasil Berdasarkan *Inertia Weight* dengan modifikasi *Constriction Coefficient*

Implementasi data RC101 dengan menggunakan modifikasi *constriction coefficient inertia weight* menggunakan Persamaan 2.21. Tabel 4.15 di bawah ini menunjukkan set parameter yang digunakan untuk menghitung proses nilai ω.

Tabel 4.12 Set Parameter *Constriction Coefficient*

Iterasi Maksimum	φ_1	φ_2	k	ω_{damp}
100	2.05	2.3	0.7	0.99
1000	2.55	2.8	0.9	
2000	2.8	3.05	1	
	3.05			

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Berdasarkan Tabel 4.12 maka akan menghasilkan 216 kombinasi hasil percobaan. Sebagai contoh akan dilakukan satu kali proses perhitungan nilai ω berdasarkan Persamaan 2.21.

Misal :

$\varphi_1 = 2.05$; $\varphi_2 = 2.3$

$k = 0.7$

sehingga :

$\varphi = \varphi_1 + \varphi_2$

$\varphi = 2.05 + 2.3$

$\varphi = 4.35$

$$\omega = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}$$

$$\omega = \frac{2(0.7)}{|2 - 4.35 - \sqrt{4.35^2 - 4(4.35)}|}$$

$$\omega = \frac{1.4}{|2 - 4,35 - \sqrt{18.9225 - 17.4}|}$$

$$\omega = \frac{1.4}{|2 - 4.35 - 1.23|}$$

$$\omega = \frac{1.4}{|-3.58|}$$

$$\omega = \frac{1.4}{3.58}$$

$$\omega = 0.39$$

Pada perhitungan nilai ω seterusnya dilakukan dengan cara yang sama. Hasil percobaan keseluruhan ditunjukkan dalam Lampiran C. Pada Tabel 4.16 di bawah ini menunjukkan 5 hasil terbaik berdasarkan total jarak terpendek.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

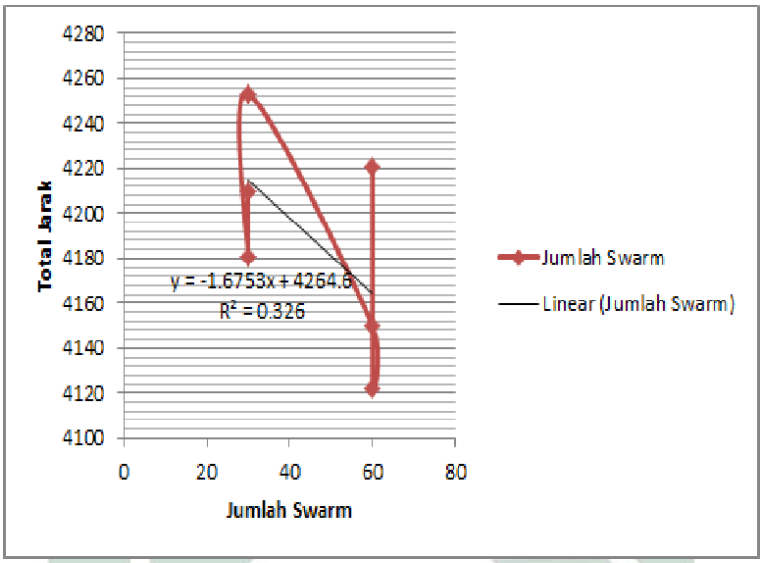
Tabel 4.13 Hasil Percobaan Terbaik Penerapan Algoritma PSO Dengan Strategi *Constriction Coefficient*

Percobaan ke-	Jumlah Swarm	Iterasi Maksimum	ω	Jumlah Lintasan	Kapasitas Maksimum	Total Jarak	Total Waktu
50	30	1000	0.9	45	200	3942	7396
210	60	2000	1	46	200	3944	7702
171	60	1000	1	48	200	3986	7688
125	60	100	0.9	48	200	3993	7814
110	60	100	0.9	47	200	3995	7685

Berdasarkan Tabel 4.13 merupakan hasil percobaan yang dilakukan *running* kurang lebih sebanyak 10 kali dengan menggunakan bahasa pemrograman. Inisialisasi posisi awal ditentukan $X_{min} = -5$ dan $X_{max} = 5$. Sedangkan, pada inisialisasi kecepatan awal dilakukan berdasarkan Persamaan 2.18 dengan menggunakan nilai $\varepsilon = 0.2$ sehingga diperoleh $V_{min} = -2$ dan $V_{max} = 2$. Tabel 4.13 menggambarkan 5 percobaan terbaik dari 216 kombinasi hasil percobaan diantaranya adalah percobaan ke 50, 210, 171, 125, dan 110. Hasil percobaan dengan jarak terpendek adalah pada percobaan ke 50 yang menghasilkan total jarak sebesar 3942 satuan jarak, dan total waktu 7396 satuan waktu.

Analisis hasil berdasarkan jumlah *swarm* dapat dilihat pada Tabel 4.13 yang menunjukkan dari 5 percobaan terbaik cenderung menggunakan jumlah *swarm* 60. Analisis hasil percobaan berdasarkan jumlah *swarm* digambarkan dalam grafik pada Gambar 4.21.

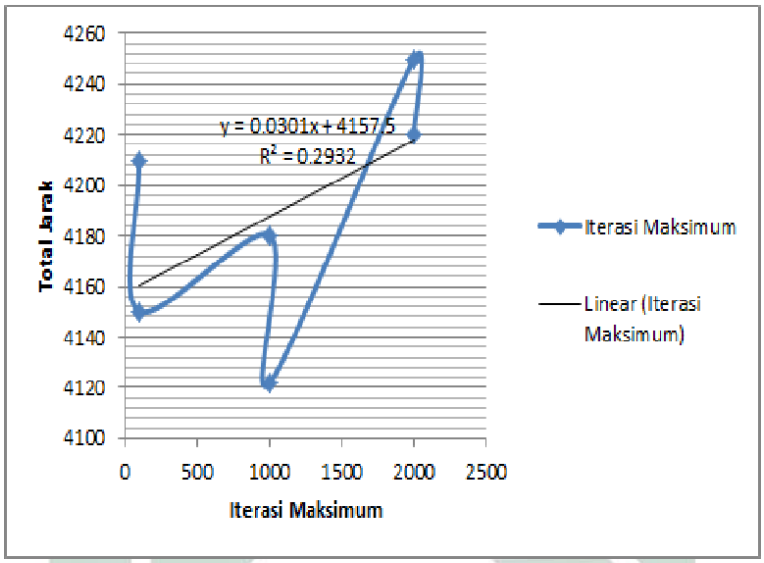
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.21 Grafik Analisis Berdasarkan Jumlah Swarm

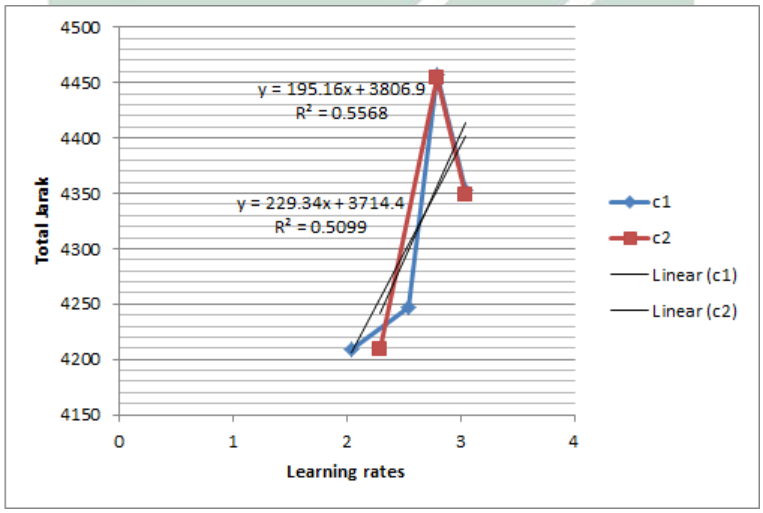
Berdasarkan Gambar 4.21 di atas, jumlah *swarm* yang semakin besar dapat memberikan pengaruh yang signifikan terhadap pengoptimalan persoalan *Vehicle Routing Problem With Time Window* (VRPTW). Namun,grafik tersebut menggambarkan jumlah *swarm* 30 menghasilkan solusi optimal jika diimbangi dengan iterasi yang besar seperti contoh hasil percobaan ke 50 pada Tabel 4.13. Analisis hasil berdasarkan jumlah iterasi yang digunakan yaitu 1000 pada Tabel 4.13 mampu memperoleh solusi optimum terhadap total jarak terpendek. Gambar 4.22 berikut ini menunjukkan grafik analisis hasil percobaan berdasarkan dari iterasi pada strategi *constriction coefficient inertia weight*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 4.22 Grafik Analisis Berdasarkan Jumlah Iterasi

Berdasarkan grafik pada Gambar 4.22 di atas menunjukkan bahwa iterasi 100 dan 1000 mempunyai pengaruh yang signifikan dalam meminimumkan total jarak jika diimbangi dengan jumlah *swarm* 60 sehingga dapat memperoleh solusi yang optimal. Analisis hasil PSO berdasarkan parameter *learning rates* yaitu c_1 dan c_2 menggunakan nilai seperti yang ditunjukkan pada Gambar 4.23.

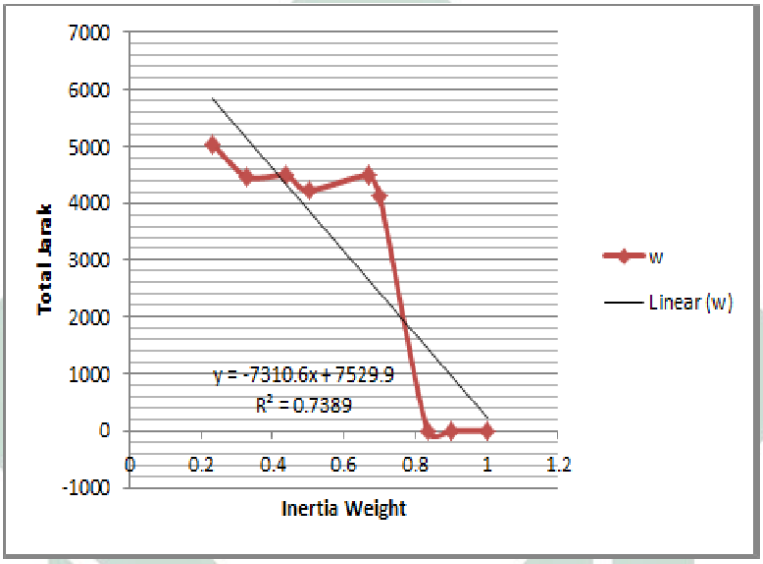


Gambar 4.23 Grafik Analisis Berdasarkan Parameter c_1 dan c_2

Berdasarkan Gambar 4.23 di atas menunjukkan bahwa semakin kecil nilai

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

parameter c_1 dan c_2 yang digunakan maka semakin minimum total jarak yang dihasilkan. Oleh karena itu, *construction coefficient* dalam algoritma PSO ini menghasilkan total jarak terpendek jika menggunakan parameter c_1 dan c_2 yang kecil. Pada modifikasi *inertia weight* dengan *constriction coefficient* untuk χ setara dengan ω sehingga diperoleh hasil perhitungan nilai ω tersebut. Analisis hasil berdasarkan nilai *inertia weight* dapat dijelaskan pada Gambar 4.25.



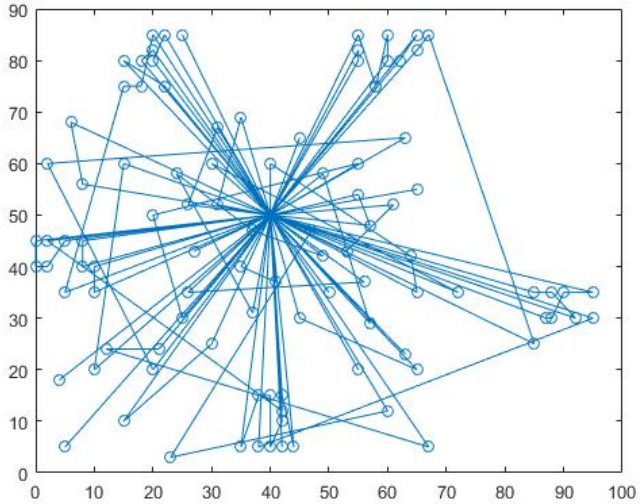
Gambar 4.24 Grafik Analisis Berdasarkan Parameter ω

Berdasarkan pada Gambar 4.25 di atas menunjukkan semakin besar nilai *inertia weight* yang digunakan maka akan menghasilkan total jarak yang minimum. Pada Gambar 4.25 tersebut mempunyai titik optimal dengan nilai ω 0.8, 0.9, dan 1. Sehingga pada persoalan VRPTW berdasarkan modifikasi *inertia weight* dengan *constriction coefficient* mampu menghasilkan solusi optimum pada nilai ω 0.8, 0.9, dan 1.

Penerapan algoritma *Particle Swarm Optimization* (PSO) dengan modifikasi *constriction coefficient inertia weight* memperoleh hasil terbaik pada percobaan ke 50. Secara rinci solusi optimal diperoleh pada *swarm* ke 9 dengan rute yang terbentuk sebanyak 45 rute yang terlampir pada Lampiran D. Gambar

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.25 sebagai berikut menggambarkan ilustrasi rute yang terbentuk berdasarkan solusi optimal dari percobaan ke 50.



Gambar 4.25 Ilustrasi Solusi Optimal Percobaan ke 50

Berdasarkan Gambar 4.25 menunjukkan gambar ilustrasi untuk 100 titik (*node*)dengan 1 pusat distribusi (depot) dan simpul menggambarkan lintasan sehingga dapat membentuk rute berdasarkan hasil percobaan ke 50.

4.3. Perbandingan Hasil Algoritma PSO Berdasarkan Strategi *Inertia Weight* Terhadap Persoalan VRPTW

Algoritma *Particle Swarm Optimization* (PSO) untuk menyelesaikan persoalan *Vehicle Routing Problem With Time Window* (VRPTW) dilakukan dengan beberapa modifikasi pada parameter PSO. Modifikasi PSO tersebut menggunakan parameter dari strategi *inertia weight*. Pada penelitian ini digunakan 6 modifikasi strategi *inertia weight* meliputi *constant inertia weight*, *random inertia weight*, *linear decreasing inertia weight*, *chaotic inertia wight*, *global-local inertia weight*, dan *constriction coefficient inertia weight*. Perbandingan hasil

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

solusi optimal yang dihasilkan berdasarkan jarak terpendek dari 6 modifikasi *inertia weight* digambarkan pada Tabel 4.16.

Tabel 4.14 Hasil Percobaan Terbaik Penerapan Algoritma PSO

Strategi *Inertia Weight*	Percobaan ke-	Jumlah Lintasan	Total Jarak	Total Waktu
Constant Inertia Weight	193	49	3879	7899
Random Inertia Weight	150	45	3821	7412
Linear Decreasing Inertia Weight	198	49	3870	7876
Chaotic Inertia Weight	114	45	3942	7396
Global-Local Inertia Weight	18	49	3959	7809
Constriction Coefficient	50	45	3942	7396

Berdasarkan Tabel 4.14 menunjukkan hasil percobaan dengan total jarak paling kecil dari 216 hasil percobaan pada masing-masing strategi *inertia weight*. Solusi optimal tersebut menghasilkan jarak terpendek pada modifikasi algoritma PSO dengan strategi *random inertia weight* dengan total jarak sebesar 3821 satuan jarak. Selain jarak terpendek yang dihasilkan berdasarkan strategi *random inertia weight* jumlah lintasan atau rute yang diperoleh juga minimum yaitu ada 45 lintasan, sehingga waktu yang dihasilkan akan lebih optimal yaitu sebesar 7412 satuan waktu.

Solusi optimal dengan jarak terpendek lainnya adalah dengan modifikasi *linear decreasing inertia weight* yang menghasilkan total jarak minimum sebesar 3870 satuan jarak. Namun, jumlah lintasan atau rute yang dihasilkan mencapai 49 lintasan sehingga total waktu yang dihasilkan cenderung lebih besar yaitu 7876 satuan waktu. Sedangkan, solusi optimal dengan total jarak yang sama yaitu sebesar 3942 satuan jarak dihasilkan berdasarkan strategi *chaotic inertia weight*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dan *constriction coefficient*. Jumlah lintasannya sebesar 45 lintasan sehingga waktu yang dihasilkan cenderung lebih kecil dibandingkan *linear decreasing inertia weight*.

Pada dasarnya, modifikasi algoritma PSO dengan 6 strategi *inertia weight* kurang mempengaruhi secara signifikan jika diperbandingkan dengan hasil solusi optimalnya. Oleh karena itu, total jarak yang diperoleh dari modifikasi algoritma PSO terhadap persoalan VRPTW tersebut cenderung tidak jauh berbeda. Berdasarkan total jarak dari hasil modifikasi algoritma PSO dengan total jarak terpendeknya diperoleh dari modifikasi *random inertia weight*, sedangkan total jarak tertinggi diperoleh dari modifikasi *global-local inertia weight*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB V

PENUTUP

Pada bab ini akan diberikan simpulan dan saran-saran yang dapat diambil berdasarkan penelitian yang telah dibahas pada bab-bab sebelumnya.

5.1. Simpulan

Simpulan yang dapat diambil penulis setelah menyelesaikan pembuatan skripsi ini adalah :

1. Algoritma *Particle Swarm Optimization* dapat diterapkan dalam menyelesaikan persoalan *Vehicle Routing Problem With Time Window*(VRPTW). Pada penelitian ini ditambahkan modifikasi dari strategi *inertia weight*. Hasil total jarak yang minimum diantaranya adalah sebagai berikut.

    a. Strategi *Constant Inertia Weight* menghasilkan total jarak 3879 satuan jarak.

    b. Strategi *Random Inertia Weight* menghasilkan total jarak 3821 satuan jarak.

    c. Strategi *Linear Decreasing Inertia Weight* menghasilkan total jarak 3870 satuan jarak.

    d. Strategi *Chaotic Inertia Weight* menghasilkan total jarak 3942 satuan jarak.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

e. Strategi *Global-Local Inertia Weight* menghasilkan total jarak 3959 satuan jarak.

f. Strategi *Constriction Coefficient Inertia Weight* menghasilkan total jarak 3942 satuan jarak.

2. Analisis hasil berdasarkan parameter-parameter pada algoritma *Particle Swarm Optimization* (PSO) diantaranya adalah sebagai berikut.

   a. Analisis hasil berdasarkan jumlah *swarm* cenderung menunjukkan jumlah *swarm* 60 untuk menyelesaikan VRPTW dengan jumlah titik 100 mampu memberikan solusi yang optimum.

   b. Analisis hasil berdasarkan iterasi maksimum menunjukkan semakin besar iterasi yang digunakan maka semakin kecil total jarak yang diperoleh.

   c. Analisis hasil berdasarkan parameter c_1 dan c_2 menunjukkan bahwa nilai rentang nilai antara 1 sampai dengan 2 memberikan hasil yang minimum.

   d. Analisis hasil berdasarkan strategi *inertia weight* diantaranya dengan rentang nilai 0.7 sampai dengan 1 memberikan pengaruh yang signifikan terhadap total jarak terpendek. Namun, semakin kecil nilai ω maka total jarak cenderung memiliki nilai yang tinggi.

3. Perbandingan hasil dari modifikasi algoritma PSO dengan 6 strategi *inertia weight* memperoleh hasil total jarak terpendek adalah dengan menggunakan strategi *random inertia weight*. Namun, pada modifikasi *inertia weight* tersebut kurang mempengaruhi secara signifikan jika diperbandingkan dengan hasil solusi optimal berdasarkan total jarak terpendek. Oleh karena

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

itu, total jarak yang diperoleh dari modifikasi 6 strategi *inertia weight* pada algoritma PSO terhadap persoalan VRPTW tersebut cenderung tidak jauh berbeda.

5.2. Saran

Setelah mengimplementasikan dan menganalisis algoritma *Particle Swarm Optimization* (PSO) untuk menyelesaikan persoalan *Vehicle Routing Problem With Time Window* (VRPTW) dengan modifikasi strategi *inertia weight* yang kurang mempengaruhi secara signifikan, penulis ingin menyampaikan beberapa saran.

1. Implementasi dengan *Fitness Distance Ratio* (FDR) berbasis algoritma *Particle Swarm Optimization* (PSO)
2. Menggunakan tipe algoritma *Particle Swarm Optimization* (PSO) lainnya seperti algoritma Multi-Objective PSO.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR PUSTAKA

Alif, A., 2014, *Ekspedisi Darat Menggunakan Algoritma Tabu Search Berbasis Android Studi Kasus : PT. Antess*, UIN Maulana Malik Ibrahim, Malang.

Andries, P., Engelbrecht, 2007, Computational Intelligence: An Introduction, *John Wiley and Sons*, ch. 16, pp. 289-358.

Anthony, C., dan Gerry, D., 2001, An Off-The-Shelf PSO," *in Workshop Particle Swarm Optimization*, Indianapolis, 2001.

Belachgar, K., 2017, *Vehicle Routing Problem with Distance Constraints and Clustering*, School of Science and Engineering, Al Akhawayn University.

Bonabeau, E., Dorigo, M., dan Theraulaz, G., (1999), *Swarm Intelligence from Natural to Artificial Systems*, New York : Oxford University Press.

Bratton, D., dan Kennedy, J., 2007, "Defining a Standard for Particle Swarm Optimization ," *in IEEE Swarm Intelligence Symposium*, pp. 120-127.

Bräysy, O., Gendreau, M., 2005, *Vehicle Routing Problem with Time Windows, Part II: Metaheuristics*, *Journal of Transportation Science*, 39(1) : 119-139.

Budi, W. P. S., 2013, *Optimasi Traveling Salesman Problem Dengan Algoritma Genetika Menggunakan Operator Partially Matched Crossover*, UIN Maulana Malik Ibrahim , Malang.

Chen, R.M., dan Shih, H.F., 2013, Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search, *Journal of Article Algorithms*, 6, pp. 227-244, ISSN 1999-4893.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Clerc, M. and Kennedy, J., 2002, The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, 6, 58-73.

Dinas Kependidikan, (2008), Pendekatan, Jenis, dan Metode Penelitian Pendidikan, http://staff.uny.ac.id/sites/default/files/PENELITIAN%20PENDIDIKAN.pdf, diakses pada 5 November 2019.

Ellabib, I., Otman, A. B., dan Calamai, P., 2002, An Experimental Study of a Simple Ant Colony System for the Vehicle Routing Problem with Time WIndows, *ANTS, LNCS*, 53(64).

Engelbrecht, S., 2006, *Motivation and Burnout in Human Service Work: The Case of Midwifery in Denmark*, Copenhagen:Faculty of Psychology, Philosophy and Science Studies, Roskilde University.

F. van den bergh, 2006,An Analysis of Perticle Swarm Optimizers. PhD thesis, *Department of Computer Science*, University of Pretoria, South Africa.

Fatmawati, Prihandono, B., Noviani, E., 2015, Penyelesaian *Travelling Salesman Problem* Dengan Metode Tabu Search. *Bimaster*(1) : 17-24.

Feng Y., Teng G.F., Wang, A.X., Yao, Y.M., 2007, Chaotic Inertia Weight in Particle Swarm Optimization, *IEEE press*, New York, p. 475.

Ferdian, F., 2009, *Penyelesaian Travelling Salesman Problem dengan Algoritma Heuristik*, Institut Teknologi Bandung.

Fuady A., Fariza, A., dan Prasetyaningrum, I., 2015, *Aplikasi Gis Berbasis J2me Pencarian Jalur Terpendek Menggunakan Algoritma Particle Swarm*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Optimization (PSO) Di Kabupaten Bangkalan, Politeknik Elektronika Negeri Surabaya.

Fuadiyah, M., 2018, *Algoritma Particle Swarm Optimization dan Aplikasinya Pada Masalah Capasitated Vehicle Routing Problem (CVRP)*, UIN Sunan Kalijag, Yogyakarta.

Fuaidah, A., 2012, *Hybrid Algoritma Particle Swarm Optimization (PSO) Dengan Simulated Annealing (SA) Pada Traveling Salesman Problem (TSP)*, Universitas Airlangga.

Gamayanti, N., Alkaff, A., dan Mangatas, R., 2015, Optimisasi Multi Depot Vehicle Routing Problem (MDVRP) dengan Variabel Travel Time Menggunakan Algoritma Particle Swarm Optimization, *JAVA Journal of Electronics Engineering*, 13(1).

Hadhiatma, A., dan Purbo, A., 2017, *Vehicle Routing Problem* Untuk Distribusi Barang Menggunakan Algoritma Semut,*Jurnal Prosiding SNATIF Ke -4 Tahun 2017*, pp. 139–145.

Hassan, R., 2004, *Particle Swarm Optimization : Method and Applications*, Engineering System Division, Massachusetts Institude of Technology.

Hutomo, H., and Sari, E. R., 2017, Penyelesaian *Capacitated Vehicle Routing Problem* Menggunakan Algoritma Genetika Dan *Nearest Neighbour Solution Of Capacitated Vehicle Routing Problem Using Genetic*,*Jurnal Matematika*, 6(2): 52–62.

Iskandar, 2010, *Model Optimasi Vehicle Routing Problem Dan Implementasinya*, Institut Pertanian Bogor.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

J. Xin, G. Chen, Y. Hai, 2009, A particle swarm optimizer with multistage linearlydecreasing inertia weight, *Proceedings of the International Joint Conference on Computational Sciences and Optimization (CSO 2009)*, IEEE press, New York, vol.1, pp. 505–508.

Kaya, I.,Hacizade, U., 2018, *GA Based Traveling Salesman Problem Solution and its Application to Transport Routes Optimization,IFAC-PapersOnLine* , 51(30): 620–625.

Kennedy, J., dan Tim Blackwell Riccardo Poli, 2007, *Particle swarm optimization An overview,Swarm Intelligence*, vol. 1, no. 1, pp. 33–57.

Kennedy, J., dan Eberhart, R., 1995, Particle Swarm Optimization, *IEEE Conference on Neural Networks*, (Perth, Australia), Piscataway, NJ, IV, pp. 1942-1948,.

Kohl, K., Desrosiers, J., Madsen, O.B.G., Solomon, M.M., Soumis, F., 1999, *2-Path Cuts for the Vehicle Routing Problem with Time Windows*, *Journal of Transportation Science* (1):103-116.

Kurniawan, A. W., dan Z, Puspitaningtyas, 2016, Metode Penelitian Kuantitatif, Yogyakarta: Pandiva Buku.

Laporte, G., 1992, The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms, *European Journal of Operational Research*, 59, 345-358.

Lawrence, R., 2008, *Penjadwalan Dan Rute Pengiriman Daging Beku Menggunakan Model Vehicle Routing Problem Dengan Metode Algoritma Tabu Search*, Universitas Indonesia.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

M.S. Arumugam, M.V.C. Rao, 2006, *On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems*, Discrete Dyn. Nat. Soc. (2006) 79295.

Maryati, I., Pickerling, C., Wibowo, H. K., 2018, *Vehicle Routing Problem* Berbasis *Ant Colony System* Untuk Optimasi Penentuan Rute Kendaraan Pada Sistem Distribusi Barang Dan Jasa, *Jurnal Teknik Informatika*, Sekolah Tinggi Teknologi Surabaya.

Moengin, P., 2011, Metode Optimasi. Bandung: Muara Indah.

Nabilla, A.R., 2016, *Penerapan Vehicle Routing Problem With Time Windows (VRPTW) Menggunakan Algoritma Kelelawar*, Universitas Airlangga.

Napitipulu, C., 2018, Implementasi Algoritma Genetika Untuk Penyelesaian Traveling Salesman Problem (TSP) Berbasis Android, pp. 01-70.

Omar, K., Yeuni, L.C., Ismail, W.R., dan Zirour, M., 2018, Vehicle Routing Problem Models and Solutions, *Journal of Quality Measurement and Analysis*, 4(1): 205-218.

Palit, H.C., dan Sherly, 2012, *Vehicle Routing Problem With Time Windows* (VRPTW) Pada Distributor Bahan Makanan, *Jurnal Prosiding Seminar Nasional Manajemen Teknologi XV*, *Petra Christian University*.

Parsopoulos, K.E., 2015, *Particle Swarm Methods*, Springer International Publishing AG, University of Ioannina, Ioannina, Greece.

Santosa, B., dan Willy, P., 2011, *Metoda Metaheuristik Konsep dan Implementasi*, Graha Ilmu, Surabaya.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Putra, R. P., 2015, Penyelesaian *Travelling Salesman Problem* Dengan Algoritma *Artificial Bee Colony*, UIN Sunan Kalijaga Yogyakarta.

Rahmawati, N. A., dan Suyanto, 2012, Implementasi *Particle Swarm Optimization* (PSO) Pada *Vehicle Routing Problem With Time Windows* (VRPTW),*Jurnal Teknik Informatika*, Institut Teknologi Telkom Bandung.

Randhani, A., 2015, *Aplikasi Algoritma Cat Swarm Optimization Pada Travelling Salesman Problem*, Universitas Airlangga.

Solomon, M.M, 2005, VRPTW Benchmark Problems, http://w.cba.neu.edu/ msolomon/problems.htm, diakses pada 7 Oktober 2019.

Sukardi, 2013, *Metodologi Penelitian Pendidikan Kompetensi dan Praktiknya*. Jakarta: Bumi Aksara.

Suyanto, 2010, *Algoritma Optimasi Deterministik atau Probabilistik*, Yogyakarta : Graha ilmu.

Talbi, E. G., 2009, Metaheuristic : From Design to Implementation, *John Wiley and Son, Inc. Hoboken*, New Jersey.

Talukder, S., 2011, *Mathematical Modelling and Applications of Particle Swarm Optimization, Mathematical Modelling and Simulation*, Blekinge Institute of Technology.

Tarigan, A. A., 2012, *Tafsir Ayat-Ayat Ekonomi (Sebuah Eksplorasi Melalui Kata-Kata Kunci dalam Al-Qur'an)*, Citapusaka Media Perintis, Bandung.

Uysal, O. dan Bulkan, S., 2008, Comparison of Genetic Algorithm and Particle Swarm Optimization for Bicriteria Permutation Flowshop Scheduling Problem,

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

International Journal of Computational Intelligence Research, ISSN: 0973-1873, Vol. 4, No. 2, pp. 159-175.

Wardy, I. S., 2007, *Penggunaan Graph Dalam Algoritma Semut Untuk Melakukan Optimisasi*, Program studi Teknik Informatika, ITB, Bandung.

Wati, Ratna, D.A., 2011, *Sistem Kendali Cerdas*, Bandung.

Shi, Y., Eberhart, R., C., 1998, A modified particle swarm optimizer, *Proceedings of IEEE 1998 International Conference on Evolutionary Computation*, IEEE press, New York, vol.1, pp. 69–73.

Shi, Y., Eberhart, R., C., 2001, Tracking and optimizing dynamic systems with particle swarms, *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE press, New York, vol.1, pp. 94–100.

Zhang, Y., Shi, L., Chen, J., and Li, X., 2017, Analysis of an Automated Vehicle Routing Problem in Logistics considering Path Interruption, *Journal of Advanced Transportation*.

Zhang, J., Yang F., dan Weng, X., 2018, An Evolutionary Scatter Search Particle Swarm Optimization Algorithm for the Vehicle Routing Problem With Time Windows, *IEEE*, Vol.6, ISSN : 2169-3536.

Zhao, Y., W., Wu, B., Wang, W., L., Ma, Y., L., Wang, W., A., dan Sun, H., 2004, Particle Swarm Optimization for Vehicle Routing Problem with Time Windows, *Materials Science Forum*, Vol 471-472, pp : 801-805.

...

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id