



**PENERAPAN ALGORITMA *PARTICLE SWARM OPTIMIZATION*
UNTUK *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS*
PADA KASUS PENDISTRIBUSIAN BARANG**

SKRIPSI

Oleh
Zuhud Mardiyatul Fatimah
NIM 121810101002

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**



**PENERAPAN ALGORITMA *PARTICLE SWARM OPTIMIZATION*
UNTUK *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS*
PADA KASUS PENDISTRIBUSIAN BARANG**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh
Zuhud Mardiyatul Fatimah
NIM 121810101002

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2016**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang telah memberikan semua kemudahan dan kesempurnaan dalam kehidupan ini;
2. Kedua orang tua saya, Ayahanda Muksam Rofiqi dan Ibunda Siti Fatimah yang telah memberikan kasih sayang, dukungan serta doa yang tiada henti;
3. dosen dan guru-guru sejak taman kanak-kanak hingga perguruan tinggi yang telah memberikan ilmu dengan penuh kesabaran;
4. sahabat dan teman-teman yang telah memberikan motivasi serta dukungannya;
5. UKMS TITIK FMIPA Unej yang telah memberikan ruang untuk belajar dan berkarya;
6. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

“Musuh yang paling berbahaya atas dunia ini adalah penakut dan bimbang. Teman yang paling setia, hanyalah keberanian dan keyakinan yang teguh.”

(Andrew Jackson)^{*)}

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

(*Terjemahan Q.S. Al-Baqarah:286*)^{**)}

*) <http://www.maribelajarbkk.web.id/2015/03/contoh-motto-terbaru-dalam-skripsi.html>

***) Departemen Agama Republik Indonesia. 2002. Al-Qur'an dan Terjemahannya. Surabaya: Duta Ilmu Surabaya

PERNYATAAN

Saya yang bertanda-tangan dibawah ini:

nama : Zuhud Mardiyatul Fatimah

NIM : 121810101002

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Penerapan Algoritma *Particle Swarm Optimization* untuk *Vehicle Routing Problem With Time Windows* pada Pendistribusian Barang” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah disebutkan sumbernya, belum pernah diajukan dalam instansi manapun serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Juni 2016

Yang menyatakan,

Zuhud Mardiyatul Fatimah

NIM 121810101002

SKRIPSI

**PENERAPAN ALGORITMA *PARTICLE SWARM OPTIMIZATION*
UNTUK *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS*
PADA KASUS PENDISTRIBUSIAN BARANG**

Oleh

Zuhud Mardiyatul Fatimah
NIM 121810101002

Pembimbing

Dosen Pembimbing Utama : Ahmad Kamsyakawuni, S.Si., M.kom.

Dosen Pembimbing Anggota : M. Ziaul Arif S.Si., M.Sc.

PENGESAHAN

Skripsi yang berjudul “Penerapan Algoritma *Particle Swarm Optimization* untuk *Vehicle Routing Problem With Time Windows* pada Pendistribusian Barang” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Jember

Tim Penguji,

Ketua,

Sekretaris,

Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP 197211291998021001

M. Ziaul Arif S.Si., M.Sc.
NIP 198501112008121002

Penguji I,

Penguji II,

Kosala Dwidja Purnomo, S.Si., M.Si.
NIP 196908281998021001

Dr. Alfian Futuhul Hadi S.Si., M.Si.
NIP 198202162006042002

Mengesahkan

Dekan,

Drs. Sujito, Ph.D.
NIP 196102041987111001

RINGKASAN

Penerapan Algoritma *Particle Swarm Optimization* Untuk *Vehicle Routing Problem* Pada Kasus Pendistribusian Barang; Zuhud Mardiyatul Fatimah, 121810101002; 2016; 100 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

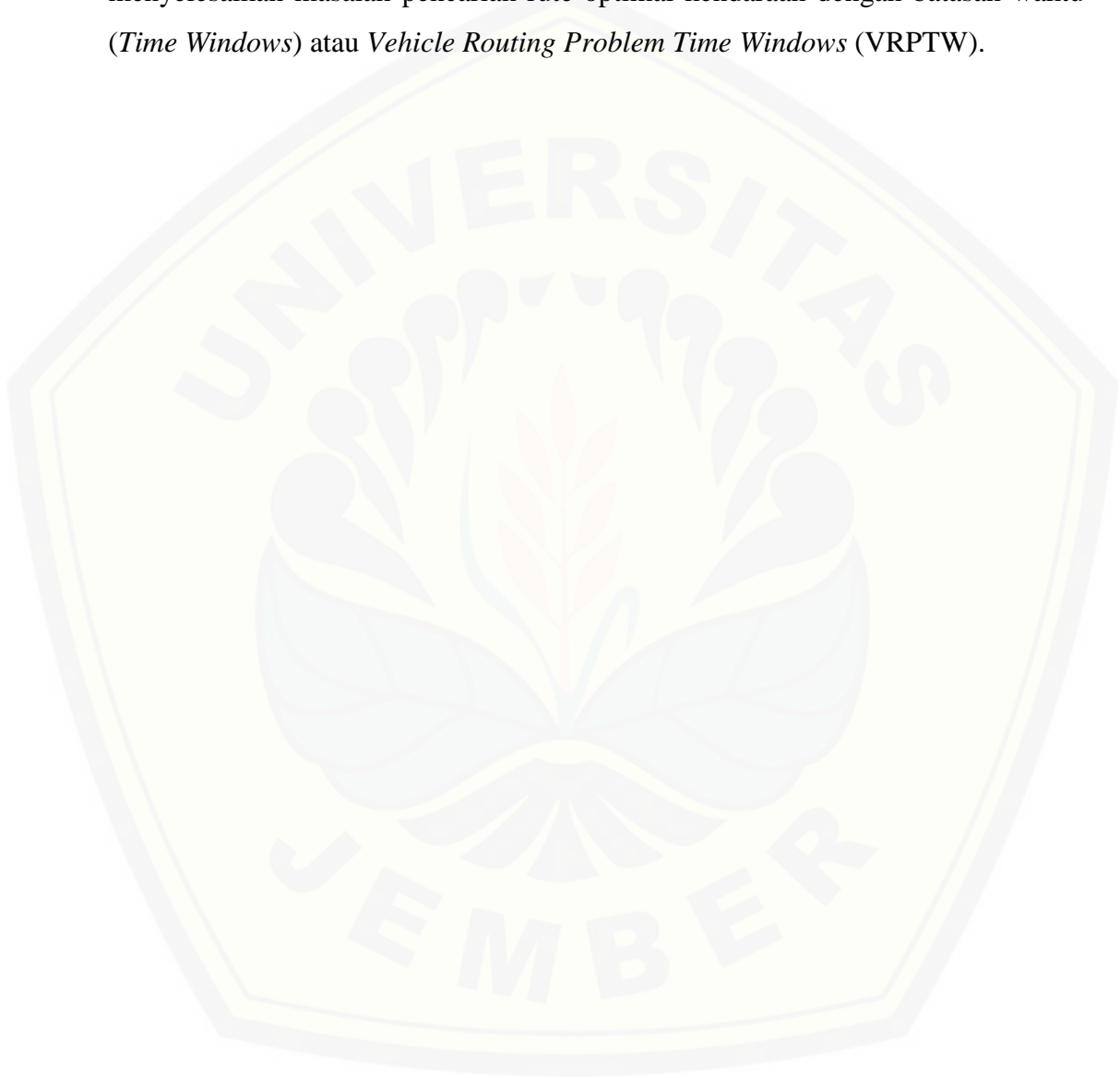
Vehicle Routing Problem (VRP) merupakan permasalahan yang membahas tentang bagaimana memilih beberapa rute yang harus dilalui oleh sejumlah kendaraan pengangkut dalam proses pendistribusian barang yang mengkombinasikan permintaan tiap *retailer* dengan memperhatikan kapasitas angkut. Salah satu jenis masalah VRP yaitu *Vehicle Routing Problem Time Windows* (VRPTW). VRPTW adalah masalah penentuan rute kendaraan dengan biaya minimum untuk melayani seluruh pelanggan dan memenuhi kendala kapasitas kendaraan dengan *time windows* pada masing-masing pelanggan dan depo.

Tujuan dari penulisan tugas akhir ini yaitu menerapkan Algoritma Particle Swarm Optimization untuk Vehicle Routing Problem With Time Windos pada kasus pendistribusian barang, serta mengetahui pengaruh parameter yang digunakan terhadap solusi optimal yang diperoleh.

Penelitian dilakukan melalui beberapa langkah, yaitu diawali dengan *study literature*, pengambilan dan pengumpulan data tentang pendistribusian barang dari PT. Amita Bara Sejahtera Jember, kemudian menerapkan Algoritma PSO sebagai metode pencarian dan metode perhitungan. Selanjutnya membuat program sesuai algoritma yang digunakan dengan menggunakan *software*

MATLAB. Selanjutnya membuat kesimpulan dari hasil yang didapat dari penerapan algoritma tersebut.

Hasil penelitian yang didapat dapat dilihat bahwa algoritma PSO merupakan algoritma yang efisien dan efektif dalam memberikan informasi dan menyelesaikan masalah pencarian rute optimal kendaraan dengan batasan waktu (*Time Windows*) atau *Vehicle Routing Problem Time Windows* (VRPTW).



PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayahNya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Algoritma *Particle Swarm Optimization* Untuk *Vehicle Routing Problem* Pada Kasus Pendistribusian Barang”. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis menyampaikan terimakasih kepada:

1. Ahmad Kamsyakawuni, S.Si., M.Kom., selaku Dosen Pembimbing Utama dan M. Ziaul Arif S.Si., M.Sc., selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan secara intensif serta pengarahan dan saran dalam penyusunan skripsi ini;
2. Kosala Dwidja Purnomo, S.Si., M.Si, dan Dr. Alfian Futuhul Hadi S.Si., M.Si.selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam penyusunan skripsi ini;
3. Seluruh staf pengajar Program Sarjana (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah memberikan ilmunya selama perkuliahan;
4. Ayahanda Muksam Rofiqi, Ibunda Siti Fatimah serta seluruh keluarga besar yang telah memberikan doa dan dorongan semangat demi terselesaikannya skripsi ini;
5. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam menyusun skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi pembaca.

Jember, Juni 2016

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
HALAMAN RINGKASAN	vii
HALAMAN PRAKATA	ix
HALAMAN DAFTAR ISI	xi
HALAMAN DAFTAR GAMBAR	xiii
HALAMAN DAFTAR TABEL	xiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Terminologi Dasar Graf	5
2.2 <i>Vehicle Routing Problem (VRP)</i>	6
2.3 <i>Vehicle Routing Problem With Time Windows (VRPTW)</i>	8
2.4 Algoritma <i>Particle Swarm Optimization (PSO)</i>	10
2.5 Pemrograman <i>MATrix LABORatory (MATLAB)</i>	13

BAB 3. METODE PENELITIAN	15
3.1 Data Penelitian.....	15
3.2 Langkah-langkah Penelitian.....	15
BAB 4. HASIL DAN PEMBAHASAN	18
4.1 Hasil	18
4.1.1 Perhitungan Manual	18
4.1.2 Program PSO untuk VRPTW menggunakan MATLAB.....	26
4.2 Pembahasan	33
4.2.1 Penerapan PSO Untuk VRPTW Pada Kasus Pendistribusian Barang.....	33
4.2.2 Pengaruh Parameter Terhadap Solusi Optimal.....	53
BAB 5. PENUTUP	55
5.1 Kesimpulan	55
5.2 Saran	55
DAFTAR PUSTAKA	56
LAMPIRAN	58

DAFTAR GAMBAR

	Halaman
2.1 (a) Graf dengan satu titik (b) Graf dengan empat titik dan lima sisi	5
2.2 Contoh penyelesaian VRP dengan 3 rute	8
3.1 Skema langkah-langkah penelitian	15
4.1 Tampilan awal program VRPTW	27
4.2 Tampilan Tabel Data	28
4.3 Tampilan <i>Input</i> Parameter Program VRPTW	29
4.4 Tampilan Output Rute Kendaraan	29
4.5 Tampilan Output Total Jarak, Total Waktu dan Total Kapasitas	30
4.6 Tampilan Hasil Setiap Toko untuk Setiap Rute Kendaraan	30
4.7 Tampilan Hasil Record untuk beberapa kali <i>running</i> /proses	31
4.8 Tampilan Grafik Kekonvergenan	31

DAFTAR TABEL

	Halaman
4.1 Jumlah Permintaan Toko/pangkalan.....	18
4.2 Data Jarak Toko/Pangkalan (Km)	19
4.3 Jalur Toko/pangkalan	21
4.4 Jalur Solusi Awal	22
4.5 Hasil Percobaan dengan <i>swarm</i> 10 dan iterasi 100 pada data pertama.....	33
4.6 Hasil Percobaan dengan <i>swarm</i> 20 dan iterasi 100 pada data pertama.....	34
4.7 Hasil Percobaan dengan <i>swarm</i> 30 dan iterasi 100 pada data pertama.....	36
4.8 Hasil Percobaan dengan <i>swarm</i> 10 dan iterasi 500 pada data pertama.....	37
4.9 Hasil Percobaan dengan <i>swarm</i> 20 dan iterasi 500 pada data pertama.....	38
4.10 Hasil Percobaan dengan <i>swarm</i> 30 dan iterasi 500 pada data pertama....	40
4.11 Hasil Percobaan dengan <i>swarm</i> 10 dan iterasi 1000 pada data pertama.	41
4.12 Hasil Percobaan dengan <i>swarm</i> 20 dan iterasi 100 pada data kedua.....	42
4.13 Hasil Percobaan dengan <i>swarm</i> 30 dan iterasi 1000 pada data kedua....	44
4.14 Hasil Percobaan dengan <i>swarm</i> 10 dan iterasi 100 pada data kedua.....	45
4.15 Hasil Percobaan dengan <i>swarm</i> 20 dan iterasi 100 pada data kedua	46
4.16 Hasil Percobaan dengan <i>swarm</i> 30 dan iterasi 100 pada data kedua.....	48
4.17 Hasil Percobaan dengan <i>swarm</i> 10 dan iterasi 500 pada data kedua.....	49
4.18 Hasil Percobaan dengan <i>swarm</i> 20 dan iterasi 500 pada data kedua.....	51
4.19 Hasil Percobaan dengan <i>swarm</i> 30 dan iterasi 500 pada data kedua.....	52
4.20 Hasil Percobaan dengan <i>swarm</i> 10 dan iterasi 1000 pada data kedua.....	54
4.21 Hasil Percobaan dengan <i>swarm</i> 20 dan iterasi 1000 pada data kedua.....	55
4.22 Hasil Percobaan dengan <i>swarm</i> 30 dan iterasi 1000 pada data kedua.....	56

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pencarian rute optimal dalam kehidupan sehari-hari saat ini sangat dibutuhkan untuk meminimalkan waktu dan biaya yang dikeluarkan, terutama bagi perusahaan besar yang setiap harinya mendistribusikan hasil produknya dengan menggunakan kendaraan. Masalah pencarian rute optimal kendaraan atau *Vehicle Routing Problem* (VRP) merupakan permasalahan yang membahas tentang bagaimana memilih beberapa rute yang harus dilalui oleh sejumlah kendaraan pengangkut dalam proses pendistribusian barang yang mengkombinasikan permintaan tiap *retailer* dengan memperhatikan kapasitas angkut. VRP dapat diformulasikan dalam bentuk graf yaitu depo ke setiap *retailer* dinyatakan sebagai titik, sedangkan jalan yang menghubungkan depo ke setiap *retailer* maupun antar *retailer* dinyatakan sebagai sisi. Diasumsikan bahwa graf yang terbentuk merupakan graf berarah dan graf berbobot. Dalam hal ini akan dibentuk beberapa subgraf yang merupakan rute pengiriman yang akan dilalui oleh masing-masing kendaraan.

Terdapat beberapa jenis dalam permasalahan VRP, salah satunya yaitu *Vehicle Routing Problem Time Windows* (VRPTW). VRPTW adalah masalah penentuan rute kendaraan dengan biaya minimum untuk melayani seluruh pelanggan dan memenuhi kendala kapasitas kendaraan dengan *time windows* pada masing-masing pelanggan dan depo. *Time windows* terdiri dari waktu buka dan waktu tutup *customer* dan *central depot*. VRPTW dapat diselesaikan dengan pendekatan metaheuristik, yaitu metode untuk mencari solusi yang memadukan interaksi antara prosedur pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik lokal optimal dan melakukan pencarian diruang solusi untuk menemukan solusi global. Contoh dari pendekatan metaheuristik diantaranya algoritma genetika, algoritma *tabu search*,

algoritma *ant colony system*, algoritma *simulated annealing*, *particle swarm optimization*, dan lain-lain.

VRPTW dengan Algoritma *Tabu Search* telah dibahas oleh Kuncoro, dkk (2011) dengan mengambil data dari PT. Kompas. Dalam penelitiannya dijelaskan bahwa algoritma *Tabu Search* dapat digunakan untuk menyelesaikan permasalahan dalam pencarian rute terbaik untuk pengangkutan surat kabar ke agen-agen. Selanjutnya optimasi VRPTW dengan Algoritma Genetika telah dibahas oleh Sundarningsih, dkk (2014) dengan mengambil studi kasus Pendistribusian Air Minum Kemasan. Dalam penelitiannya dijelaskan bahwa Algoritma Genetika dapat diterapkan pada pencarian rute optimal pada distribusi air minum dengan kendala *time window*, serta hasil pengujian dengan metode *Elitis* menghasilkan nilai *fitness* yang lebih baik daripada menggunakan seleksi *Roulette wheel*.

Penerapan Algoritma *Ant Colony System* (ACS) pada VRPTW telah dibahas oleh Utami, dkk (2012) dengan mengambil data tipe *cluster* dan *random customer*. Dalam penelitiannya dijelaskan bahwa semakin banyak jumlah semut yang diberikan maka akan memberikan solusi yang banyak. Pada penelitian Pornsing (2014) tentang VRP dengan algoritma *Particle Swarm Optimization* (PSO) dijelaskan bagaimana penerapan algoritma berbasis PSO untuk dipartisi VRP yang berhubungan dengan masalah koleksi daur ulang dan bagaimana mengurangi masalah *stagnasi* sementara pencarian lokal dipertahankan dengan mengeksekusi kedua eksplorasi dan tahap eksploitasi iterasi secara bersamaan.

Berdasarkan penjelasan diatas, penulis tertarik untuk menerapkan algoritma *Particle Swarm Optimization* (PSO) untuk *Vehicle Routing Problem Time Windows* (VRPTW) pada kasus pendistribusian barang. Penerapan ini dilakukan untuk memperoleh rute jalur terpendek dengan memperhatikan posisi partikel (*retailer/toko*) dan kecepatan kendaraan serta dengan biaya minimum untuk melayani seluruh pelanggan dan memenuhi kendala kapasitas kendaraan dan batasan *time windows*. Pendistribusian barang akan dilakukan di PT. Amita Bara Sejahtera Jember yaitu salah satu perusahaan distributor gas LPG 3 kg dengan menggunakan *pick up*. Dalam proses pendistribusian barang PT. Amita

Bara Sejahtera Jember menentukan bahwa tidak ada toko/pangkalan yang terlewatkan dan tidak ada toko/pangkalan yang di datangi dua kali dalam sehari.

1.2 Rumusan Masalah

Dari gambaran yang sudah tertulis diatas, permasalahan yang akan dibahas dalam skripsi ini adalah sebagai berikut:

- a. bagaimana penerapan algoritma *Particle Swarm Optimization* (PSO) untuk *Vehicle Routing Problem with Time Windows* (VRPTW) pada kasus pendistribusian barang di PT. Amita Bara Sejahtera Jember?
- b. bagaimana pengaruh parameter (jumlah toko, jumlah permintaan barang, jumlah *swarm* dan jumlah iterasi) terhadap solusi optimal yang didapatkan?

1.3 Batasan Masalah

Batasan-batasan masalah yang terdapat dalam skripsi ini adalah sebagai berikut :

- a. keberangkatan kendaraan berawal dan berakhir dititik yang sama yaitu di depo;
- b. kecepatan kendaraan saat pengiriman diasumsikan rata-rata 25 km/jam;
- c. kapasitas kendaraan pengangkut dan jenis muatan adalah sama;
- d. rute dan jarak antar toko/pangkalan satu dengan toko/pangkalan yang lain dan sebaliknya diasumsikan sama;
- e. waktu buka dan tutup depo dan toko/pangkalan diasumsikan sama;
- f. waktu pelayanan meletakkan dan mengambil tabung dianggap sama.

1.4 Tujuan

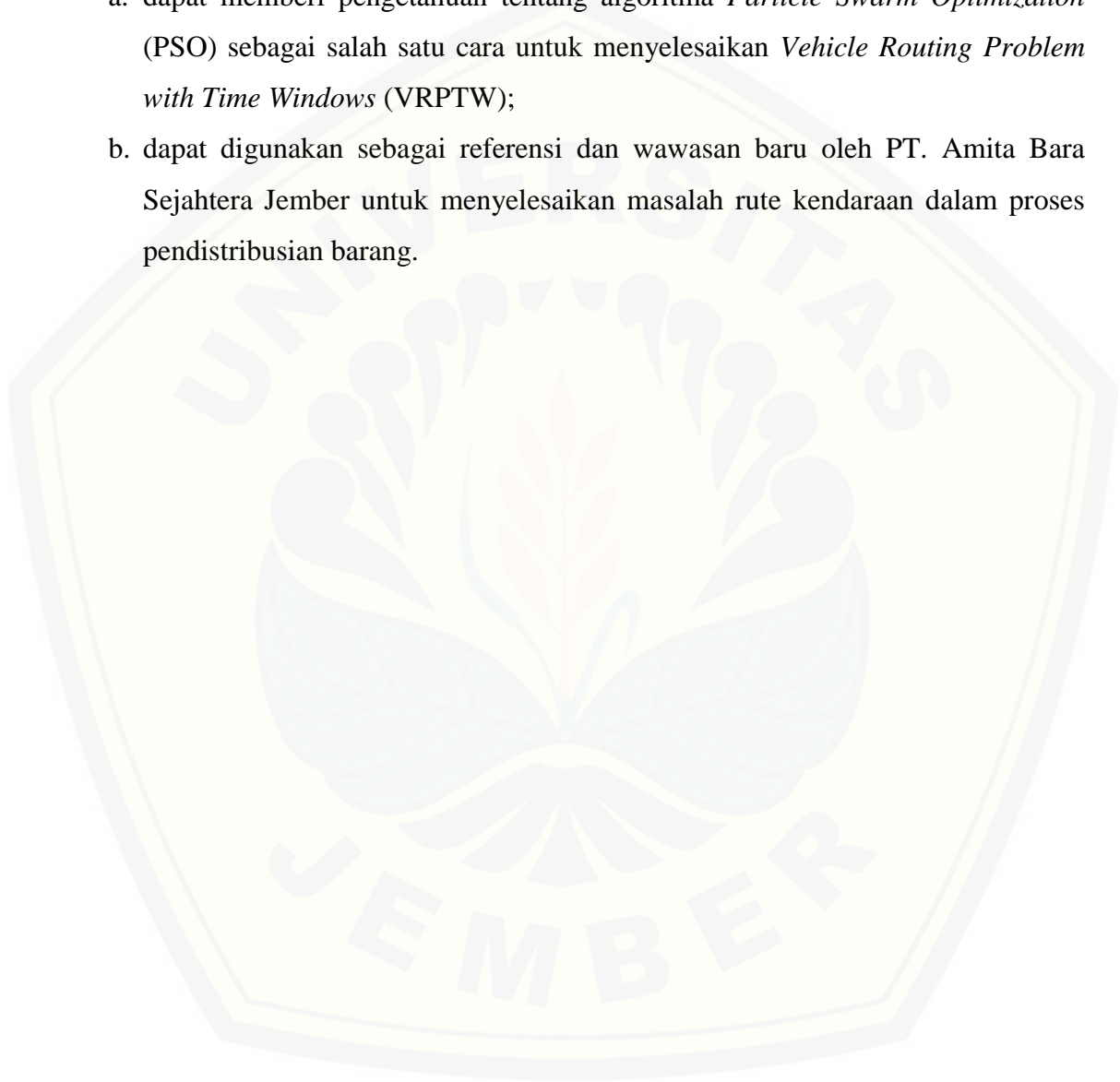
Tujuan dari penelitian ini adalah sebagai berikut:

- a. menerapkan algoritma *Particle Swarm Optimization* (PSO) untuk *Vehicle Routing Problem with Time Windows* (VRPTW) pada kasus pendistribusian barang di PT. Amita Bara Sejahtera Jember;
- b. mengetahui pengaruh parameter (jumlah toko, jumlah permintaan barang, jumlah *swarm* dan jumlah iterasi) terhadap solusi optimal yang didapatkan.

1.5 Manfaat

Manfaat yang dapat diambil dari penulisan penelitian ini adalah sebagai berikut:

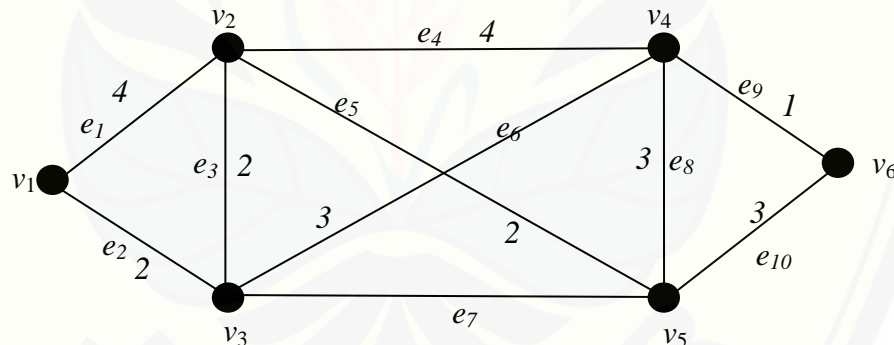
- a. dapat memberi pengetahuan tentang algoritma *Particle Swarm Optimization* (PSO) sebagai salah satu cara untuk menyelesaikan *Vehicle Routing Problem with Time Windows* (VRPTW);
- b. dapat digunakan sebagai referensi dan wawasan baru oleh PT. Amita Bara Sejahtera Jember untuk menyelesaikan masalah rute kendaraan dalam proses pendistribusian barang.



BAB 2. TINJAUAN PUSTAKA

2.1 Terminologi Dasar Graf

Sebuah graf didefinisikan sebagai pasangan himpunan $(V(G), E(G))$, dengan $V(G)$ adalah himpunan tak kosong dari elemen-elemen yang disebut titik (*vertex*) dan $E(G)$ adalah himpunan boleh kosong dari pasangan tak terurut dua titik (u,v) (ditulis $e=(u,v)$) dari titik-titik u,v di V yang disebut sisi (*edges*) (Chartrand and Oellermann, 1993). Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak berarah. Graf berbobot adalah graf yang setiap sisinya dikaitkan dengan bilangan real positif. Bobot dari graf G dinotasikan dengan $w(G)$ yaitu jumlah bobot dari semua sisi digraf G . Gambar 2.1 adalah contoh graf berbobot dengan $w(G)=26$.



Gambar 2.1 Graf Berbobot

Suatu jalan (*walk*) digraf G dinotasikan dengan $W(G)$ adalah suatu barisan berhingga yang diawali dan diakhiri dengan titik, dimana suku-sukunya bergantian antara titik dan sisi yaitu $v_1, e_1, v_2, e_1, v_3, e_1, \dots, v_n$ ($n \geq 1$) dengan titik dan sisi boleh diulang. Suatu jalan dikatakan tertutup jika titik awalnya sama dengan titik akhirnya yaitu $v_1 = v_n$, sedangkan jika $v_1 \neq v_n$ disebut jalan terbuka. Suatu jalan yang sisinya tidak diulang disebut jejak (*trail*). Suatu jalan yang titiknya yang tidak diulang disebut lintasan (*path*). Sedangkan jalan tertutup tanpa

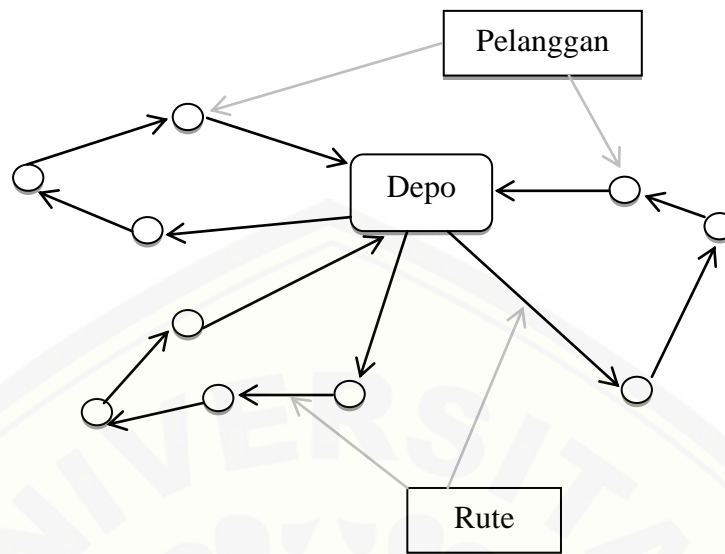
pengulangan titik kecuali titik awal dan titik akhir disebut siklus (*cycle*) (Chartrand and Oellermann, 1993). Suatu graf dikatakan terhubung jika setiap titik yang berlainan dihubungkan dengan sekurang-kurangnya sebuah lintasan.

2.2 Vehicle Routing Problem (VRP)

Vehicle Routing Problem (VRP) atau masalah rute kendaraan merupakan salah satu kajian dalam ilmu matematika dan juga merupakan model masalah transportasi yang mengundang banyak ilmuwan komputer yang bertujuan untuk mencari rute optimal untuk sejumlah kendaraan dalam melayani sejumlah *customer*. VRP juga mempunyai beberapa varian utama dan beberapa metode.

Vehicle Routing Problem (VRP) dapat dimisalkan suatu depo mempunyai n -kendaraan dengan kapasitas angkut yang sama dan melayani sejumlah *retailer*. Depo tersebut ingin membuat m -rute yang akan dilalui setiap kendaraan untuk melakukan pengiriman kepada sejumlah *retailer*. Perjalanan setiap kendaraan berawal dan berakhir pada depo. Permasalahannya adalah memilih beberapa rute dengan mengkombinasikan permintaan tiap *retailer* serta memperhatikan kapasitas angkut kendaraan. Dari rute yang terpilih ditentukan urutan *retailer* yang akan dikunjungi untuk meminimumkan total jarak maupun total waktu.

Dalam *Vehicle Routing Problem* (VRP) terdapat tiga unsur utama yaitu sebuah depo, sejumlah *retailer* yang akan dikunjungi, dan sejumlah kendaraan yang dimiliki depo (Doerner, 2001). *Vehicle Routing Problem* (VRP) didefinisikan kedalam sebuah graf terhubung $G=(V,E)$ dimana $V= \{v_1, v_2, v_3, \dots, v_n\}$ adalah himpunan titik dan $E= \{e_1, e_2, e_3, \dots, e_n\}$ adalah himpunan sisi. Dalam hal ini v_0 melambangkan depo, v_i melambangkan *retailer* i , $e = (v_i, v_j)$ melambangkan jalan yang menghubungkan *retailer* i ke *retailer* j dengan bobot $w(v_i, v_j)$ adalah bilangan real positif yang menyatakan waktu tempuh atau jarak dari *retailer* i ke *retailer* j dan bobot $w(v_0, v_i)$ menyatakan waktu tempuh atau jarak dari depo ke *retailer* i . Tujuan dari VRP adalah untuk meminimalkan total jarak tempuh kendaraan dari rute perjalanan. Contoh penyelesaian permasalahan VRP dapat dilihat pada gambar berikut:



Gambar 2.2 Contoh penyelesaian VRP dengan 3 rute

Menurut Toth dan Vigo (Iskandar, 2010) permasalahan utama VRP dibagi beberapa variasi/jenis, yaitu:

- setiap kendaraan memiliki kapasitas yang terbatas (*Capacitated Vehicle Routing Problem*);
- pelanggan dapat melakukan permintaan berupa pengiriman dan pengambilan barang ke depot (*Vehicle Routing Problem Backhauls*);
- setiap pelanggan harus dikirim barang dalam waktu tertentu (*Vehicle Routing Problem With Time Windows*);
- menggunakan banyak depot untuk mengirim pelanggan (*Multiple Depot Vehicle Routing Problem*);
- pelanggan dapat mengembalikan barang-barang kembali ke depot (*Vehicle Routing Problem with Pick Up and Delivering*);
- pelanggan bisa dilayani dengan kendaraan yang berbeda-beda (*Split Delivery Vehicle Routing Problem*);
- pengiriman dilakukan dalam periode waktu tertentu (*Periodic Vehicle Routing Problem*);
- gabungan dari MDVRP dengan VRPB yaitu kondisi depot sebagai pusat distribusi barang lebih dari satu dan pelanggan dapat melakukan permintaan

berupa pengiriman dan pengambilan barang (*Multiple Depot Vehicle Routing Problem With Backhauls*);

2.3 Vehicle Routing Problem With Time Windows (VRPTW)

Vehicle Routing Problem with Time Windows (VRPTW) adalah turunan dari *Capacitated Vehicle Routing Problem* (CVRP) dengan tambahan properti *Time Windows* yang diasosiasikan pada tiap-tiap *customer* dan *central depot*. *Time Windows* yang diasosiasikan pada tiap *customer* dapat menerima barang kiriman. *Time Windows* yang diasosiasikan pada *central depot* adalah *range* waktu antara semua kendaraan memulai rute dan waktu paling lambat untuk mengakhiri rute. *Time windows* terdiri dari waktu buka dan waktu tutup *customer* dan *central depot*. Jika dimisalkan suatu *customer* buka pukul 7-10 pagi, maka kendaraan harus sampai ke *customer* tersebut sebelum jam 10 pagi. Jika kendaraan tersebut sampai sebelum pukul 7 pagi, maka kendaraan tersebut harus menunggu sampai waktu itu untuk mulai melayani *customer*. Pada varian ini juga diperkenalkan istilah *service time* (waktu pelayanan), yang merepresentasikan waktu yang dibutuhkan untuk melayani suatu *customer*. *Service time* dapat berupa waktu yang dibutuhkan untuk menurunkan barang dari suatu kendaraan. Dengan demikian dapat dikatakan bahwa *Vehicle Routing Problem with Time Windows* (VRPTW) merupakan permasalahan yang terjadi pada sistem logistik dan juga permasalahan bagaimana sebuah depo, pusat distribusi barang dengan sejumlah kendaraan berkapasitas tertentu, melayani sejumlah *customer* pada titik-titik lokasi terpisah dengan permintaan dan batasan *time window* tertentu dengan tujuan meminimalkan total biaya perjalanan tanpa mengabaikan batasan kapasitas kendaraan dan *time window* depot (Maryati dkk, 2010).

Dari perspektif graf, VRPTW dapat dinyatakan sebagai berikut : misalkan $G(V, A)$ menjadi sebuah graf yang diarahkan dengan simpul $V = (v_0, v_1, \dots, v_n)$ dan busur $A = \{(c_i, c_j) : i \neq j, c_i, c_j \in C\}$. dalam model graf ini, c_0 adalah depot, $c_i = (i = 1, 2, \dots, n)$ adalah pelanggan. Untuk setiap arc (c_i, c_j) terkait nilai t_{ij} mewakili waktu perjalanan dari c_i ke c_j . Sebuah rute didefinisikan sebagai mulai dari depot, akan melalui sejumlah pelanggan dan berakhir di depot; setiap

pelanggan c_i ($i=0, 1, \dots, n$) harus dikunjungi tepat satu kali. Ada K kendaraan $V_c=\{0, 1, \dots, K-1\}$ dan jumlah rute tidak dapat melebihi K . Setiap kendaraan melayani sebagian pelanggan pada rute, dan kendaraan mempunyai kapasitas yang sama (Q). Jumlah total tuntutan pelanggan yang dilayani oleh kendaraan pada rute tidak dapat melebihi Q . Kedala tambahan bahwa waktu layanan dimulai dari pelanggan (c_i) harus lebih besar dari atau sama dengan waktu awal yang tersedia untuk pelayanan (b_i). Dalam hal waktu kedatangan kurang dari b_i , kendaraan harus menunggu sampai waktu awal sebelum memulai pelayanan pelanggan. Tujuannya adalah untuk menemukan satu set rute yang dapat menjamin setiap pelanggan untuk dilayani oleh satu kendaraan dalam interval waktu tertentu dan kemudian memenuhi keterbatasan kapasitas kendaraan. Selain itu, ukuran dari himpunan harus kurang dari jumlah kendaraan yang dibutuhkan dan jarak perjalanan total harus diminimalkan. Formulasi matematika dari VRPTW dapat disajikan seperti berikut :

$$\text{Min } z = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^{K-1} c_{ij} x_{ijk} \quad (2.1)$$

dengan kendala,

1. Menetapkan bahwa tidak ada lebih dari rute K akan keluar dari depot

$$\sum_{k=0}^{K-1} \sum_{j=1}^n x_{ijk} \leq K \quad (i = 0) \quad (2.2)$$

2. Memastikan bahwa satu kendaraan masuk dan keluar dari pelanggan persis

$$\sum_{k=0}^{K-1} \sum_{j=0}^n x_{ijk} = 1 \quad (i = 1, 2, \dots, n; i \neq j) \quad (2.3)$$

3. Kendala kapasitas

$$\sum_{i=0}^n d_i \sum_{j=0}^n x_{ijk} \leq Q \quad (i \neq j, \forall k \in [0, K - 1]) \quad (2.4)$$

4. Waktu jendela terjamin

$$t_0 = 0 \quad (2.5)$$

$$t_i + t_{ij} + s_i - M(1 - x_{ijk}) \leq t_j \quad (i, j \in [l, n]; i \neq j \in [0, K - 1]) \quad (2.6)$$

$$b_i \leq t_i \leq e_i \quad (2.7)$$

5. Kendala arus konservasi yang menggambarkan jalur kendaraan

$$\sum_{i=0}^n x_{ihk} - \sum_{j=0}^n x_{hjk} = 0 \quad (i = 0, 1, 2, \dots, n; j \in V, k \in V_c) \quad (2.8)$$

dimana,

x_{ijk} = variabel keputusan, yaitu :

$$x_{ijk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ dari pelanggan } i \text{ ke pelanggan } j \\ 0, & \text{jika lainnya} \end{cases}$$

t_i = waktu pelayanan kendaraan dimulai pada pelanggan i

M = konstanta besar. Fungsi tujuan yang menyatakan bahwa total biaya yang harus diminimalkan

dengan indeks sebagai berikut :

i = indeks toko awal

j = indeks toko tujuan

k = indeks kendaraan

N = indeks banyak toko.

(Xu, 2015).

2.4 Particle Swarm Optimization (PSO)

Algoritma *Particle Swarm Optimization* (PSO) pertama kali dikenalkan oleh Eberhart dan Kennedy ditahun 1995 dalam sebuah konferensi jaringan syaraf di Perth, Australia. Algoritma *Particle Swarm Optimization* (PSO) merupakan teknik optimasi berbasis *stochastic* yang diinspirasi oleh tingkah laku sosial sekawanan burung atau sekumpulan ikan.

Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak disuatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik yaitu posisi dan kecepatan. Setiap partikel bergerak dalam ruang atau space tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing.

Berdasarkan penjelasan diatas dapat dikatakan bahwa algoritma *Particle Swarm Optimization* (PSO) adalah metode pencarian yang didasarkan populasi dan merupakan algoritma optimasi global yang dihubungkan dengan masalah yang mana solusi terbaik dapat direpresentasikan sebagai titik atau *surface* diarea *n-dimensional*. Meskipun PSO telah digunakan untuk optimasi selama hampir dua

dekade, namun periode inidikatakan relatif singkat jika dibandingkan dengan metode lain seperti *Artificial Neural Networks* (ANN), Algoritma Genetika (GA) atau *Ant Colony Optimization* (ACO) karena keuntungan dari PSO. Keuntungan PSO antara lain yaitu konvergensi yang cepat menuju optimal, kemudahan dalam *encoding* dan *decoding*, mudah dalam perhitungandan telah diterapkan di berbagai penelitian seperti optimasi global, pelatihan jaringan syaraf tiruan, sistem kontrol *fuzzy*, rekayasa optimasi desain serta logistik dan sebagainya. Meskipun demikian, banyak peneliti telah mencatat bahwa PSO cenderung konvergen lebih cepat pada optimal lokal (Pornsing, 2014).

PSO diinisialisasi dengan populasi solusi acak dan mencari solusi optimal dengan memperbaiki posisi partikel. Setiap partikel adalah solusi yang layak dan ditetapkan nilai *fitness* dengan fungsi tujuan. Setiap partikel menjaga jalur dari koordinat dalam ruang masalah yang terkait dengan solusi terbaik (*fitness*) yang telah dicapai. Nilai fitness disebut p_{best} . Ketika partikel mengambil semua populasi sebagai tetangga topologinya, posisi terbaik adalah global terbaik yang disebut g_{best} . Melalui p_{best} dan g_{best} , partikel memperbaiki diri untuk menghasilkan generasi berikutnya dari kawanan.

Pemilihan fungsi *fitness* tergantung pada tujuan penelitian. Fungsi *fitness* untuk mengevaluasi individu selalu terkait dengan fungsi tujuan. Untuk VRPTW, total biaya dapat dilihat sebagai nilai *fitness*. Kebalikan dari total biaya yang digunakan untuk mewakili kebugaran dari individu-individu, dan kemudian fungsi kebugaran didefinisikan sebagai berikut:

$$f_{fitnes} = \frac{1}{z} \quad (2.9)$$

Dalam algoritma PSO partikel merupakan solusi potensial untuk masalah ini, dan kawanan terdiri dari partikel P . Setiap partikel p dapat direpresentasikan melalui vektor n -dimensi: vektor pertama didefinisikan sebagai $X_p^t = (x_{p1}^t, x_{p2}^t, \dots, x_{pn}^t)$ dengan $p = (1, 2, \dots, p)$ yang menunjukkan posisi partikel p dalam ruang pencarian di iterasi t . Vektor kedua dituliskan $V_p^t = (v_{p1}^t, v_{p2}^t, \dots, v_{pn}^t)$ yang merepresentasikan kecepatan dengan partikel p bergerak. Vektor ketiga dituliskan $P_{best_p}^t = (p_{best_{p1}}^t, p_{best_{p2}}^t, \dots, p_{best_{pn}}^t)$ menunjukkan posisi terbaik dari

partikel ke- p dan vektor terakhir dituliskan $G_{best_p}^t = (g_{best_{p1}}^t, g_{best_{p2}}^t, \dots, g_{best_{pn}}^t)$ yang merepresentasikan posisi terbaik secara global dalam kawanan sampai iterasi ke- t . Kawanan diperbarui oleh persamaan berikut :

$$v_{pn}^{t+1} = v_{pn}^t + k_1 \times rand1() \times (p_{best_{pn}}^t - x_{pn}^t) + k_2 \times rand2() \times (g_{best_{pn}}^t - x_{pn}^t) \quad (2.10)$$

$$x_{pn}^{t+1} = x_{pn}^t + v_{pn}^{t+1} \quad (2.11)$$

dimana, k_1 dan k_2 adalah parameter, $rand1()$ dan $rand2()$ adalah bilangan acak yang berdistribusi seragam dalam selang $[0, 1]$. Koefisien k_1 dan k_2 adalah konstanta positif untuk mengontrol seberapa jauh sebuah partikel akan bergerak dalam iterasi tunggal. Nilai-nilai yang rendah memungkinkan partikel untuk menjelajah jauh dari daerah sasaran sebelum menarik kembali, sementara nilai-nilai yang tinggi mengakibatkan gerakan tiba-tiba terhadap atau masa lalu daerah sasaran. Biasanya adalah kedua nilai 2.0, meskipun pemberian nilai berbeda untuk k_1 dan k_2 kadang-kadang mengarah ke peningkatan kinerja.

Konstanta v_{max} digunakan untuk membatasi kecepatan dari partikel v_{pn}^t dan meningkatkan resolusi pencarian. Ketika v_{max} besar, kecepatan partikel juga besar, hal ini adalah kondusif untuk pencarian global, yang mungkin terbang melauhi solusi optimal. Ketika v_{max} kecil, kecepatan partikel juga kecil, hal itu mengarah ke pencarian terbaik di wilayah tertentu, tetapi mudah untuk jatuh ke optimum lokal. Secara singkat, efisiensi pencarian tergantung pada v_{max} .

Setiap partikel bergerak dalam ruang pencarian dengan kecepatan sesuai dengan solusi terbaik sendiri dan solusi terbaik kelompok sebelumnya. Dalam persamaan (2.10) kecepatan v_{pn}^{t+1} terdiri dari tiga bagian yaitu bagian pertama adalah v_{pn}^t menunjukkan kecepatan partikel sebelumnya. Bagian kedua $k_1 \times rand1()$ menunjukkan proses penyelesaian dari pengalaman individu. Bagian ketiga $k_2 \times rand2()$ menunjukkan proses penyelesaian dari pengalaman orang lain, yang mewakili berbagi informasi dan kerjasama sosial antar partikel. Keseimbangan antara bagian – bagian ini menentukan kinerja algoritma PSO. (Xu, 2015).

Secara garis besar, algoritma *Particle Swarm Optimization* (PSO) dapat dijabarkan sebagai berikut:

- a. inialisasi posisi awal (C_i) dan kecepatan awal partikel (V_i), dengan $i = 1, 2, \dots, S$ dan S adalah ukuran *swarm*.
- b. evaluasi nilai fungsi tujuan untuk setiap partikel ($f(C_i)$).
- c. tentukan p_{best} awal dan g_{best} awal.
- d. update kecepatan dengan persamaan

$$V_{il} = \omega V_{il} + c_1 r_1 (p_{ibest} - C_{il}) + c_2 r_2 (g_{best} - C_{il})$$

dengan :

1. p_{ibest} adalah individu dengan nilai partikel terbaik dari masing-masing individu.
2. g_{best} adalah individu dengan nilai partikel terbaik dari semua individu dalam *swarm*.
3. ω adalah koefisien berat inersia, dengan rumus :

$$\omega = \omega_{max} - \text{Iterasi} \times \frac{\omega_{max} - \omega_{min}}{\text{Maksimum Iterasi}}$$
 ω_{max} dan ω_{min} adalah nilai maksimum dan nilai minimum dari ω
4. c_1 dan c_2 adalah koefisien akselerasi yang berupa konstanta positif.
5. r_1 dan r_2 adalah bilangan random antara 0 dan 1.
- e. update posisi individu baru dengan persamaan : $C'_{il} = C_{il} + V_{il}$
- f. evaluasi kembali $f(C_i)$, jika $f(C_i) \leq f(p_{ibest})$ maka $p_{ibest} = C_i$, setelah mendapatkan p_{ibest} baru, maka didapatkan $f(p_{ibest})$ baru.

$$g_{best} = \begin{cases} p_{ibest}, & f(p_{ibest}) < f(g_{best}) \\ g_{best}, & f(p_{ibest}) \geq f(g_{best}) \end{cases}$$

- g. Jika iterasi sudah maksimum maka algoritma berhenti, jika tidak maka kembali ke langkah (d).

2.5 Pemrograman MATrix LABoratory (MATLAB)

MATLAB merupakan bahasa pemrograman yang dikembangkan oleh The Mathwork. Inc (<http://www.mathworks.com>). Bahasa pemrograman ini banyak digunakan untuk perhitungan numerik keteknikan, komputasi, simbolik, visualisasi, grafis, analisis data matematis, statistika, simulasi, pemodelan, dan desain GUI.

GUI merupakan MATLAB *script file* yang dibuat untuk menunjukkan analisa suatu permasalahan khusus agar dapat memberikan hasil yang baik dan sederhana dalam penggunaannya serta mudah untuk memasukkan data dan melihat hasilnya. Dalam merancang GUI terdapat 2 cara yaitu dengan metode sederhana dan dengan menggunakan tool khusus untuk merancang sesuatu yang diinginkan (Arhami, 2005).

Ada beberapa karakteristik MATLAB, yaitu sebagai berikut :

- a. Bahasa pemrogramannya didasarkan pada matriks (baris dan kolom)
- b. Lambat (dibandingkan dengan *Fortran* atau C) karena bahasanya langsung diartikan. Sebagai contoh, tidak diperlukan *pre-compiled*. Menghindari kalang *for (for loops)*. Setiap saat menggunakan bentuk-bentuk vektor.
- c. *Automatic memorymanagement*, misalnya kita tidak harus mendeklarasikan *arrays* terlebih dahulu
- d. Tersusun rapi (seperti pengaturan *array* di *Fortran-90*)
- e. Memiliki waktu pengembangan program yang lebih cepat dibandingkan bahasa pemrograman tradisional seperti *Fortran* atau C
- f. Dapat diubah ke bahasa C lewat MATLAB *Compiler* untuk efisiensi yang lebih baik
- g. Tersedia banyak toolbox untuk aplikasi-aplikasi khusus.

(Prasetyo, 2003)

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini adalah data yang diambil penulis dari PT. Amita Bara Sejahtera Jember, yang merupakan salah satu perusahaan distributor gas LPG 3 kg dan yang mengirimkan barang ke toko/pangkalan yang ada dalam satu lingkup kota Jember. Data yang digunakan sebagai berikut:

- a. jumlah toko/pangkalan yaitu 124 toko/pangkalan se Kabupaten Jember, namun banyaknya toko/pangkalan yang dikunjungi setiap hari berbeda sesuai dengan banyaknya permintaan barang;
- b. jarak dari depo ke posisi toko/pangkalan dan jarak antara toko/pangkalan dengan toko/pangkalan yang lain (Lampiran A);
- c. besarnya atau banyaknya permintaan barang setiap toko/pangkalan (Lampiran A);
- d. jumlah kendaraan yang dimiliki perusahaan sebanyak 4 kendaraan, dimana kapasitas angkut setiap kendaraan maksimal 200 tabung gas LPG;
- e. waktu pelayanan barang datang setiap satu kali angkut dari kendaraan ke tempat toko/pangkalan yaitu 30 detik;
- f. waktu buka dan tutup depo dalam proses pendistribusian barang selama 8 jam yang dimulai dari pukul 07.00 sampai 15.00.

3.2 Langkah-langkah Penelitian

Langkah-langkah yang akan dilakukan untuk menyelesaikan *Vehicle Routing Problem with Time Windows* (VRPTW) pada kasus pendistribusian barang dengan algoritma *Particle Swarm Optimization* (PSO) seperti berikut:

- a. Studi Literatur

Langkah awal yang dilakukan adalah mengumpulkan berbagai literatur tentang VRP, VRPTW, dan algoritma *Particle Swarm Optimization* (PSO) dari internet maupun buku-buku.

b. Pengambilan dan Pengumpulan Data

Pada langkah ini, dilakukan pengambilan dan pengumpulan data tentang pendistribusian barang dari PT. Amita Bara Sejahtera Jember ke setiap toko/pelanggan yang ada di Kabupaten Jember. Data diambil langsung dengan cara wawancara dengan bagian *Distribution Center* (DC) dari PT. Amita Bara Sejahtera Jember.

c. Penerapan Algoritma *Particle Swarm Optimization* (PSO) untuk kasus VRPTW

Data yang diperoleh selanjutnya dibawa ke kasus VRPTW dan diolah menggunakan algoritma PSO dengan langkah-langkah sebagai berikut:

1) Inisialisasi parameter

Parameter yang digunakan berupa jumlah toko, jumlah permintaan barang, jumlah partikel (*swarm*), dan jumlah iterasi.

2) Menerapkan algoritma *Particle Swarm Optimization* (PSO) pada data yang sudah dikumpulkan dan diidentifikasi dengan langkah-langkah pada subbab (2.4) dengan memperhatikan kendala-kendala yang ada pada subbab (2.3).

d. Pembuatan program dengan menggunakan *software* MATLAB R2009a

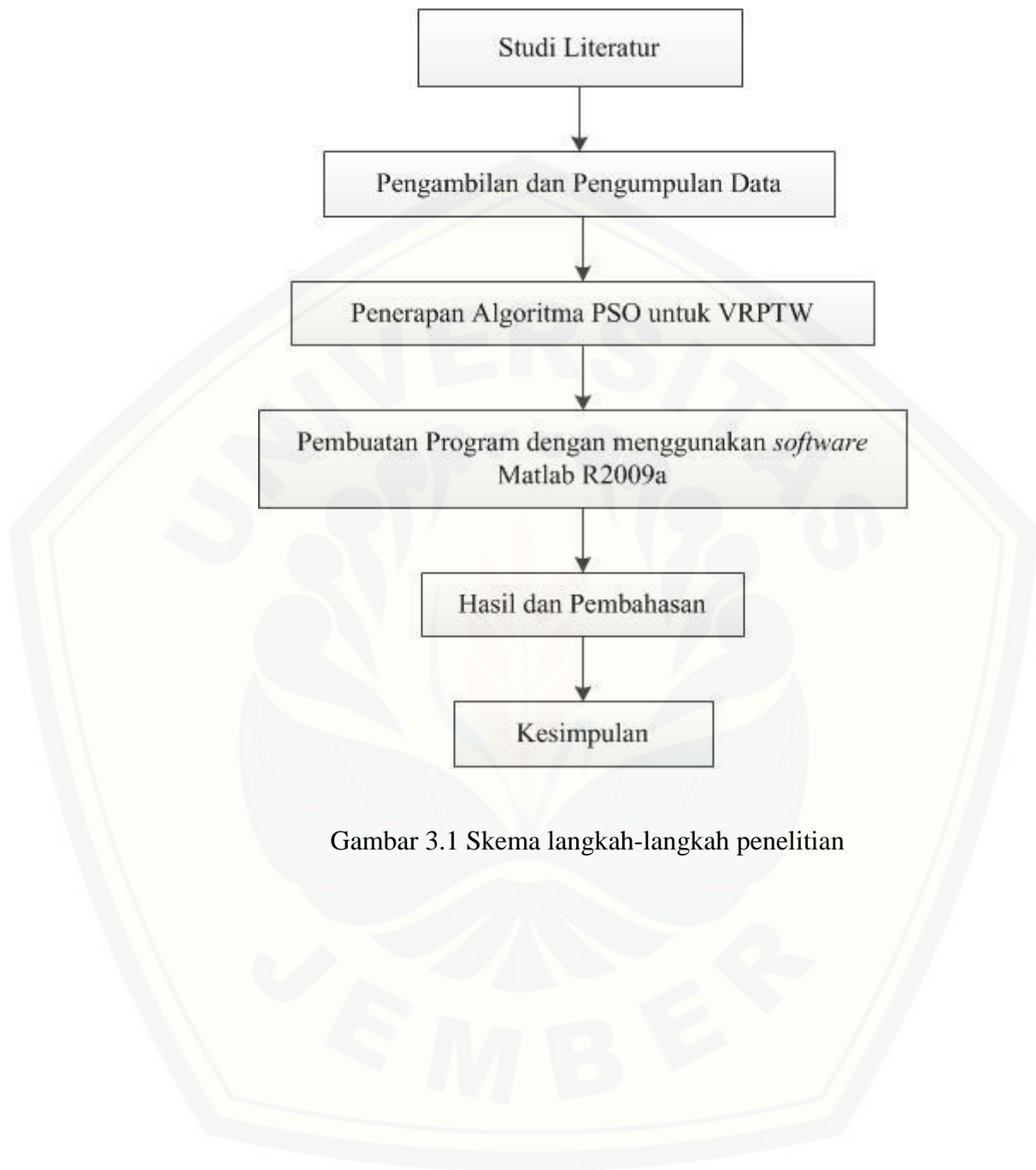
Langkah penelitian berikutnya adalah pembuatan program yang menggunakan *software* matematika yaitu MATLAB R2009a. Pada langkah ini penulis akan membuat skrip program berupa tampilan GUI untuk penyelesaian VRPTW dengan Algoritma *Particle Swarm Optimization* (PSO).

e. Hasil

Dalam penelitian ini, parameter yang dapat di ubah-ubah yaitu variabel jumlah partikel (*swarm*), jumlah toko/pangkalan, jumlah permintaan barang, dan jumlah iterasi. Hasil dari langkah ini berupa rute kendaraan, total waktu perjalanan dan waktu pelayanan, total jarak dan total jarak yang ditempuh dari setiap rute yang dihasilkan untuk penyelesaian VRPTW Algoritma *Particle Swarm Optimization* (PSO).

f. Membuat kesimpulan berdasarkan hasil langkah (c) yang telah dilakukan.

Berikut skema dari langkah – langkah penelitian di atas :



Gambar 3.1 Skema langkah-langkah penelitian

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil Penerapan Algoritma *Particle Swarm Optimization* untuk *Vehicle Routing Problem With Time Windows* pada PT. Amita Bara Sejahtera yaitu salah satu perusahaan distributor gas LPG 3 kg menggunakan program yang telah dibuat dengan bahasa pemrograman MATLAB.

4.1 Hasil

Data yang digunakan dalam penelitian ini adalah data dari PT. Amita Bara Sejahtera Jember yaitu salah satu perusahaan distributor gas LPG 3 kg berupa data lokasi toko/pangkalan serta banyaknya permintaan setiap toko/pangkalan, jumlah kendaraan yang dimiliki perusahaan, kapasitas kendaraan yang digunakan, waktu pelayanan barang datang, serta waktu buka tutup depo dan toko/pangkalan. Data jarak depot ke pelanggan dan jarak pelanggan ke pelanggan dicari menggunakan *Google Maps* (lihat Lampiran A.1).

4.1.1 Perhitungan Manual

Untuk memberikan gambaran tentang langkah-langkah Algoritma *Particle Swarm Optimization* untuk penyelesaian VRPTW maka diberikan contoh penyelesaian manual dengan menggunakan sampel data berupa jumlah permintaan 9 toko/pangkalan (dapat dilihat pada tabel 4.1) dan data jarak dari depo ke setiap toko/pangkalan maupun jarak antar toko/pangkalan (dapat dilihat pada tabel 4.2), serta dimisalkan jumlah kendaraan yang dimiliki depo sebanyak 3 kendaraan *pick up* dengan kapasitas setiap kendaraan sebanyak 200 tabung dan batasan waktu (*time wondows*) selama 8 jam.

Tabel 4.1 Jumlah Permintaan Toko/pangkalan

Kode Toko/Pangkalan	10	11	18	24	29	38	117	125	126
Jumlah Permintaan	17	17	11	3	6	32	15	200	200

Tabel 4.2 Data Jarak Toko/Pangkalan (Km)

Kode Toko/ Pangkalan	1	11	12	19	25	30	39	118	126	127
1	0	18,6	16,2	14	14,5	13,3	17	1,1	12,6	19,9
11	18,6	0	3,5	5,1	4,3	5,3	6,2	16,6	28,8	36,1
12	16,2	3,5	0	2,8	2,5	2,9	2,9	14,2	26,5	33,8
19	14	5,1	2,8	0	1,1	0,7	4,9	12	24,3	31,6
25	14,5	4,3	2,5	1,1	0	1,3	4,6	12,5	24,8	32,1
30	13,3	5,3	2,9	0,7	1,3	0	5,1	11,3	23,6	30,9
39	17	6,2	2,9	4,9	4,6	5,1	0	15,2	27,5	34,8
118	1,1	16,6	14,2	12	12,5	11,3	15,2	0	13,7	21
126	12,6	28,8	26,5	24,3	24,8	23,6	27,5	13,7	0	10,9
127	19,9	36,1	33,8	31,6	32,1	30,9	34,8	21	10,9	0

Berikut ini adalah langkah-langkah penyelesaian VRPTW dengan menggunakan Algoritma *Particle Swarm Optimization* sebagai berikut:

a. Inisialisasi parameter

Parameter yang digunakan yaitu:

Jumlah toko/pangkalan = 7 toko/pangkalan

Jumlah kendaraan = 3 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 5

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 5 iterasi

b. Proses Algoritma PSO untuk VRPTW

1. Inisialisasi posisi awal (C_i) dan kecepatan awal (V_0)

- a) Bangkitkan posisi awal (C_i) secara *random* antara [0,1] dengan jumlah baris sama dengan jumlah swarm dan jumlah kolom sama dengan jumlah toko/pangkalan \times jumlah kendaraan seperti berikut:

$$\text{Pop } (C_i) = \begin{bmatrix} 0,49 & 0,63 & 0,19 & 0,85 & 0,64 & 0,69 & 0,49 & 0,95 & 0,15 & 0,06 \\ 0,02 & 0,81 & 0,84 & 0,54 & 0,43 & 0,17 & 0,38 & 0,91 & 0,12 & 0,84 \\ 0,95 & 0,64 & 0,66 & 0,32 & 0,89 & 0,82 & 0,34 & 0,02 & 0,74 & 0,41 \\ 0,31 & 0,25 & 0,21 & 0,76 & 0,36 & 0,46 & 0,95 & 0,47 & 0,86 & 0,91 \\ 0,13 & 0,38 & 0,16 & 0,27 & 0,04 & 0,55 & 0,81 & 0,96 & 0,47 & 0,34 \\ 0,71 & 0,82 & 0,81 & 0,12 & 0,51 & 0,33 & 0,91 & 0,31 & 0,43 & 0,11 \\ 0,21 & 0,82 & 0,55 & 0,88 & 0,65 & 0,06 & 0,84 & 0,03 & 0,87 & 0,32 \\ 0,19 & 0,81 & 0,12 & 0,71 & 0,91 & 0,44 & 0,93 & 0,39 & 0,27 & 0,04 \\ 0,48 & 0,61 & 0,05 & 0,01 & 0,59 & 0,24 & 0,46 & 0,27 & 0,94 & 0,13 \\ 0,41 & 0,01 & 0,48 & 0,71 & 0,42 & 0,21 & 0,67 & 0,39 & 0,19 & 0,09 \\ 0,46 & 0,93 & 0,85 & 0,78 & 0,21 & 0,34 & 0,56 & & & \\ 0,12 & 0,67 & 0,82 & 0,04 & 0,38 & 0,68 & 0,93 & & & \\ 0,92 & 0,22 & 0,39 & 0,46 & 0,49 & 0,34 & 0,17 & & & \\ 0,97 & 0,86 & 0,61 & 0,52 & 0,68 & 0,82 & 0,44 & & & \\ 0,81 & 0,62 & 0,56 & 0,45 & 0,69 & 0,31 & 0,97 & & & \end{bmatrix}$$

Untuk setiap partikel swarm, urutkan bilangan random dari yang terkecil ke yang terbesar.

$$\text{Pop } (C_i)_{ur} = \begin{bmatrix} 0,06 & 0,11 & 0,12 & 0,15 & 0,19 & 0,21 & 0,31 & 0,33 & 0,34 & 0,43 \\ 0,02 & 0,03 & 0,04 & 0,06 & 0,12 & 0,12 & 0,17 & 0,21 & 0,32 & 0,38 \\ 0,02 & 0,04 & 0,12 & 0,17 & 0,19 & 0,22 & 0,27 & 0,32 & 0,34 & 0,34 \\ 0,01 & 0,05 & 0,13 & 0,21 & 0,24 & 0,25 & 0,27 & 0,31 & 0,36 & 0,44 \\ 0,01 & 0,04 & 0,09 & 0,13 & 0,16 & 0,19 & 0,21 & 0,27 & 0,31 & 0,34 \\ 0,46 & 0,49 & 0,49 & 0,51 & 0,56 & 0,63 & 0,64 & 0,69 & 0,71 & 0,78 \\ 0,38 & 0,43 & 0,54 & 0,55 & 0,65 & 0,67 & 0,68 & 0,81 & 0,82 & 0,82 \\ 0,39 & 0,39 & 0,41 & 0,44 & 0,46 & 0,49 & 0,64 & 0,66 & 0,71 & 0,74 \\ 0,46 & 0,46 & 0,47 & 0,48 & 0,52 & 0,59 & 0,61 & 0,61 & 0,68 & 0,76 \\ 0,38 & 0,39 & 0,41 & 0,42 & 0,45 & 0,47 & 0,48 & 0,55 & 0,56 & 0,62 \\ 0,81 & 0,82 & 0,85 & 0,85 & 0,91 & 0,93 & 0,95 & & & \\ 0,84 & 0,84 & 0,84 & 0,87 & 0,88 & 0,91 & 0,93 & & & \\ 0,81 & 0,82 & 0,89 & 0,91 & 0,92 & 0,93 & 0,95 & & & \\ 0,82 & 0,86 & 0,86 & 0,91 & 0,94 & 0,95 & 0,97 & & & \\ 0,67 & 0,69 & 0,71 & 0,81 & 0,81 & 0,96 & 0,97 & & & \end{bmatrix}$$

Setelah masing-masing partikel di urutkan, maka akan didapat rute yang terbentuk dari posisi bilangan random tersebut sebelum di urutkan. Urutan Pop (C_i) secara mendatar dalam matriks yaitu (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3), (4,1), (4,2), (4,3), (5,1), (5,2), (5,3), (6,1), (6,2), (6,3), (7,1), (7,2), (7,3), (8,1), (8,2), (8,3), (9,1), (9,2), (9,3) dan urutan tersebut digunakan untuk menentukan urutan rute toko/pangkalan dengan kendaraan.

Sebagai ilustrasi misalkan *Swarm* partikel pertama sebelum di urutkan adalah

$$SP_1 = [0,06 \quad 0,11 \quad 0,12 \quad 0,15 \quad 0,19 \quad 0,21 \quad 0,31 \quad 0,33 \quad 0,34 \quad 0,43 \\ 0,46 \quad 0,49 \quad 0,49 \quad 0,51 \quad 0,56 \quad 0,63 \quad 0,64 \quad 0,69 \quad 0,71 \quad 0,78 \\ 0,81 \quad 0,82 \quad 0,85 \quad 0,85 \quad 0,91 \quad 0,93 \quad 0,95]$$

Dari urutan bilangan random tersebut menghasilkan rute toko/pangkalan dengan kendaraan (4,1) – (7,2) – (5,2) – (3,3) – (1,3) – (9,1) – (6,3) – (6,1) – (9,2) – (7,1) – (7,3) – (1,1) – (3,1) – (5,3) – (9,3) – (1,2) – (2,2) – (2,3) – (4,2) – (8,3) – (5,1) – (4,3) – (2,1) – (8,2) – (6,2) – (8,1) – (3,2), untuk vektor solusi partikel yang lain dilakukan cara yang sama untuk mendapatkan rute toko/pangkalan dengan kendaraan (dapat dilihat pada tabel 4.3).

Tabel 4.3 Jalur Toko/pangkalan

Vektor Solusi Partikel (VSP)	Jalur Toko/pangkalan
1	(4,1), (7,2), (5,2), (3,3), (1,3), (9,1), (6,3), (6,1), (9,2), (7,1), (7,3), (1,1), (3,1), (5,3), (9,3), (1,2), (2,2), (2,3), (4,2), (8,3), (5,1), (4,3), (2,1), (8,2), (6,2), (8,1), (3,2)
2	(1,1), (6,3), (8,3), (6,1), (3,3), (7,3), (2,3), (4,2), (7,2), (3,1), (9,1), (2,2), (2,1), (5,1), (5,3), (8,1), (9,2), (1,2), (4,3), (8,2), (1,3), (4,1), (6,2), (7,1), (5,2), (3,2), (9,3)
3	(3,2), (7,2), (5,1), (9,3), (4,2), (8,1), (7,1), (2,1), (3,1), (9,2), (6,3), (8,2), (4,1), (6,1), (8,3), (9,1), (1,2), (1,3), (5,2), (3,3), (4,3), (2,3), (2,2), (5,3), (7,3), (6,2), (1,1)
4	(5,2), (5,1), (7,2), (1,3), (6,1), (1,2), (6,3), (1,1), (2,2), (9,3), (2,3), (6,2), (3,2), (4,2), (8,3), (5,3), (4,3), (8,2), (9,1), (2,1), (9,2), (3,3), (8,1), (4,1), (7,1), (3,1), (7,3)
5	(4,3), (2,2), (7,2), (1,1), (3,1), (7,1), (6,1), (2,1), (9,2), (4,1), (1,2), (6,3), (4,2), (5,3), (8,3), (3,3), (5,1), (3,2), (8,2), (8,1), (6,2), (9,1), (5,2), (7,3), (3,1), (3,2), (9,3)

Perhitungan total jarak, kapasitas kendaraan dan total waktu (perjalanan dan pelayanan) *Swarm* ke-1 dapat dilihat pada Lampiran B. Perhitungan *Swarm* ke-2

sampai dengan *Swarm* ke-5 dilakukan dengan cara yang sama, dengan syarat memenuhi kendala (2.2), (2.3) sampai (2.8)

Untuk menghitung total waktu langkah-langkahnya sebagai berikut:

$$\text{Untuk rute (4,1)} \quad T_{ij} = \frac{\text{jarak}}{\text{kecepatan kendaraan}} = \frac{14,5 \text{ km}}{25 \text{ km/jam}} = 0,58 \text{ jam}$$

$$S_i = \frac{\text{jumlah permintaan}}{4} \times 30 \text{ detik} = \frac{3}{4} \times 30 \text{ detik} = 30 \text{ detik}$$

$$\text{Jadi total waktu} = T_{ij} + S_i = 0,58 \text{ jam} + 0,0083 \text{ jam} = 0,5883 \text{ jam}$$

Solusi awal yaitu *Swarm* ke-1 sampai *Swarm* ke-5 terdapat pada Tabel 4.4

Tabel 4.4 Jalur Solusi Awal

Swarm ke-	Jalur pangkalan	Total Jarak (km)			Kapasitas (tabung)			Total Waktu (jam)		
		K1	K2	K3	K1	K2	K3	K1	K2	K3
1	K1 = 0 - 4 - 9 - 0 K2 = 0 - 7 - 5 - 2 - 0 K3 = 0 - 3 - 1 - 6 - 8 - 0	66,5	29,3	76,2	203	38	260	3,08	1,509	3,465
2	K1 = 0 - 1 - 9 - 5 - 0 K2 = 0 - 4 - 0 K3 = 0 - 6 - 8 - 3 - 7 - 2 - 0	98,9	29	111	223	3	275	3,97	1,168	4,49
3	K1 = 0 - 5 - 8 - 2 - 0 K2 = 0 - 3 - 7 - 4 - 1 - 0 K3 = 0 - 9 - 6 - 0	79,6	61,4	71,7	223	46	232	3,23	2,498	2,935
4	K1 = 0 - 6 - 0 K2 = 0 - 5 - 7 - 2 - 3 - 4 - 0 K3 = 0 - 1 - 9 - 8 - 0	34	53,3	78,2	32	52	417	1,43	2,296	3,545
5	K1 = 0 - 1 - 3 - 6 - 0 K2 = 0 - 2 - 7 - 9 - 0 K3 = 0 - 4 - 5 - 8 - 0	45,6	71,3	52	60	232	209	1,89	3,269	2,497

➤ Untuk memproses berikan kecepatan awal (V_0) misalkan seperti berikut:

$$V_0 = \begin{bmatrix} 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \end{bmatrix}$$

1) Evaluasi fungsi tujuan untuk setiap partikel ($f(C_i)$)

Dari Tabel 4.4 diketahui solusi awal yang harus di evaluasi setiap partikelnya, solusi akan optimal jika memenuhi kendala (2.4) sampai (2.8). Namun pada perhitungan manual ini belum ditemukan solusi awal yang optimal karena belum memenuhi kendala-kendala tersebut, maka perlu dilakukan pembangkitan bilangan *random* lagi. Setelah solusi awal memenuhi kendala-kendala tersebut maka tentukan $p_{i \text{ best}}$ awal dan $g_{i \text{ best}}$ awal untuk mencari nilai individu dan global yang baik.

2. Tentukan $p_{i \text{ best}}$ awal dan $g_{i \text{ best}}$ awal

a) Untuk setiap partikel, $p_{i \text{ best}}$ awal akan sama dengan nilai partikel. Dari contoh perhitungan manual diperoleh :

$$p_{\text{best}1} = [0,49 \quad 0,63 \quad 0,19 \quad 0,85 \quad 0,64 \quad 0,69 \quad 0,49 \quad 0,95 \quad 0,15 \\ 0,06 \quad 0,71 \quad 0,82 \quad 0,81 \quad 0,12 \quad 0,51 \quad 0,33 \quad 0,91 \quad 0,31 \\ 0,43 \quad 0,11 \quad 0,46 \quad 0,93 \quad 0,85 \quad 0,78 \quad 0,21 \quad 0,34 \quad 0,56]$$

$p_{\text{best}2}$ sampai $p_{\text{best}5}$ sama dengan nilai partikel *swarm* ke-2 sampai partikel *swarm* ke-5

b) Untuk $g_{i \text{ best}}$ awal diperoleh setelah mengevaluasi setiap partikel dan memperoleh solusi yang sesuai dengan kendala dari VRPTW yaitu jarak yang minimum, waktu perjalanan dan pelayanan yang minimum dan

sesuai batasan waktu (*time window*) yang ditentukan serta kapasitas kendaraan. Pada perhitungan ini diperoleh $g_{i\ best}$ saat *swarm* ke-4 yaitu:

$$g_{1\ best} = \begin{bmatrix} 0,31 & 0,25 & 0,21 & 0,76 & 0,36 & 0,46 & 0,95 & 0,47 & 0,86 \\ 0,91 & 0,48 & 0,61 & 0,05 & 0,01 & 0,59 & 0,24 & 0,46 & 0,27 \\ 0,94 & 0,13 & 0,97 & 0,86 & 0,61 & 0,52 & 0,68 & 0,82 & 0,44 \end{bmatrix}$$

3. Update kecepatan dengan menggunakan persamaan:

$$\text{Misal } r_1 = 0,2 \quad c_1 = 2 \quad \omega_{max} = 0,9$$

$$r_2 = 0,4 \quad c_2 = 2 \quad \omega_{min} = 0,1$$

$$\omega = \omega_{max} - \text{Iterasi} \times \frac{\omega_{max} - \omega_{min}}{\max \text{ Iterasi}}$$

$$\omega = 0,9 - 1 \times \frac{0,9 - 0,1}{5}$$

$$\omega = 0,016$$

$$V'_1(C_1) = \omega V_0 + c_1 r_1 (p_{best1} - C_1) + c_2 r_2 (g_{1\ best} - C_1)$$

$$V'_1(C_1) = 0,016 \begin{bmatrix} 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 \end{bmatrix} + (2)(0,2) \begin{bmatrix} 0,49 & 0,63 & 0,19 & 0,85 & 0,64 & 0,69 \\ 0,49 & 0,95 & 0,15 & 0,06 & 0,71 & 0,82 & 0,81 & 0,12 & 0,51 \\ 0,33 & 0,91 & 0,31 & 0,43 & 0,11 & 0,46 & 0,93 & 0,85 & 0,78 \\ 0,21 & 0,34 & 0,56 \end{bmatrix} - \begin{bmatrix} 0,49 & 0,63 & 0,19 & 0,85 & 0,64 & 0,69 \\ 0,49 & 0,95 & 0,15 & 0,06 & 0,71 & 0,82 & 0,81 & 0,12 & 0,51 \\ 0,33 & 0,91 & 0,31 & 0,43 & 0,11 & 0,46 & 0,93 & 0,85 & 0,78 \\ 0,21 & 0,34 & 0,56 \end{bmatrix} + (2)(0,4) \begin{bmatrix} 0,31 & 0,25 & 0,21 & 0,76 & 0,36 \\ 0,46 & 0,95 & 0,47 & 0,86 & 0,91 & 0,48 & 0,61 & 0,05 & 0,01 \\ 0,59 & 0,24 & 0,46 & 0,27 & 0,94 & 0,13 & 0,97 & 0,86 & 0,61 \\ 0,52 & 0,68 & 0,82 & 0,44 \end{bmatrix} - \begin{bmatrix} 0,49 & 0,63 & 0,19 & 0,85 & 0,64 \\ 0,69 & 0,49 & 0,95 & 0,15 & 0,06 & 0,71 & 0,82 & 0,81 & 0,12 \\ 0,51 & 0,33 & 0,91 & 0,31 & 0,43 & 0,11 & 0,46 & 0,93 & 0,85 \\ 0,78 & 0,21 & 0,34 & 0,56 \end{bmatrix}$$

$$V'_1(C_1) = \begin{bmatrix} -0,1424 & -0,3024 & 0,0176 & -0,0704 & -0,2224 & -0,1824 \\ 0,3696 & -0,3834 & 0,5696 & 0,6816 & -0,1824 & -0,1664 \end{bmatrix}$$

$$\begin{bmatrix} -0,6064 & -0,0864 & 0,0656 & -0,0704 & -0,3584 & -0,0304 \\ 0,4096 & 0,0176 & 0,4096 & -0,0544 & -0,1904 & -0,2064 \\ 0,3776 & 0,3856 & -0,0944 & & & \end{bmatrix}$$

Untuk perhitungan $V'_2 (C_2)$ sampai $V'_5 (C_5)$ sama seperti perhitungan $V'_1 (C_1)$

4. Update posisi partikel baru

untuk setiap partikel nilainya diupdate dengan

$$C'_1 = C_1 + V'_1$$

$$C'_1 = \begin{bmatrix} 0,49 & 0,63 & 0,19 & 0,85 & 0,64 & 0,69 & 0,49 & 0,95 & 0,15 & 0,06 \\ 0,71 & 0,82 & 0,81 & 0,12 & 0,51 & 0,33 & 0,91 & 0,31 & 0,43 & 0,11 \\ 0,46 & 0,93 & 0,85 & 0,78 & 0,21 & 0,34 & 0,56 & & & \\ -0,3024 & 0,0176 & -0,0704 & -0,2224 & -0,1824 & 0,3696 & & & & \\ -0,3834 & 0,5696 & 0,6816 & -0,1824 & -0,1664 & -0,6064 & & & & \\ -0,0864 & 0,0656 & -0,0704 & -0,3584 & -0,0304 & 0,4096 & & & & \\ 0,0176 & 0,4096 & -0,0544 & -0,1904 & -0,2064 & 0,3776 & & & & \\ 0,3856 & -0,0944 & & & & & & & & \end{bmatrix}$$

$$C'_1 = \begin{bmatrix} 0,3476 & 0,3276 & 0,2076 & 0,7796 & 0,4176 & 0,5076 & 0,8596 \\ 0,5676 & 0,7196 & 0,7416 & 0,5276 & 0,6536 & 0,2036 & 0,0336 \\ 0,5756 & 0,2596 & 0,5516 & 0,2796 & 0,8396 & 0,1276 & 0,8696 \\ 0,8756 & 0,6596 & 0,5736 & 0,5876 & 0,7256 & 0,4656 & \end{bmatrix}$$

Untuk perhitungan C'_2 sampai C'_5 sama seperti perhitungan C'_1 .

Kemudian untuk menentukan posisi partikel baru didapat dari nilai baru dibagi dengan jumlah totalnya setiap partikel. Dalam hal ini jumlah total dari $C'_1 = 14,2852$, sehingga nilai baru setelah setiap partikel dibagi dengan jumlah totalnya adalah

$$C'_{1\text{ baru}} = \begin{bmatrix} 0,0243 & 0,0229 & 0,0145 & 0,0546 & 0,0292 & 0,0355 \\ 0,0602 & 0,0397 & 0,0504 & 0,0519 & 0,0369 & 0,0457 \\ 0,0142 & 0,0023 & 0,0403 & 0,0182 & 0,0386 & 0,0196 \\ 0,0588 & 0,0089 & 0,0609 & 0,0613 & 0,0462 & 0,0401 \\ 0,0411 & 0,0508 & 0,0326 & & & \end{bmatrix}$$

Untuk perhitungan $C'_{2\text{ baru}}$ sampai $C'_{5\text{ baru}}$ sama dengan perhitungan $C'_{1\text{ baru}}$. Kemudian urutkan $C'_{1\text{ baru}}$ tersebut dari yang terkecil ke yang besar, seperti berikut:

$$C'_{1\text{ baru}} = \begin{bmatrix} 0,0023 & 0,0089 & 0,0142 & 0,0145 & 0,0182 & 0,0196 \\ 0,0229 & 0,0243 & 0,0292 & 0,0326 & 0,0355 & 0,0369 \\ 0,0386 & 0,0397 & 0,0401 & 0,0403 & 0,0411 & 0,0457 \\ 0,0462 & 0,0504 & 0,0508 & 0,0519 & 0,0546 & 0,0588 \\ 0,0602 & 0,0606 & 0,0613 \end{bmatrix}$$

Untuk $C'_{2\text{ baru}}$ sampai $C'_{5\text{ baru}}$ di urutkan sama dengan $C'_{1\text{ baru}}$. Sehingga rute yang terbentuk adalah

$$C'_{1\text{ baru}} = \begin{bmatrix} 5,2 & 7,2 & 5,1 & 1,3 & 6,1 & 6,3 & 1,2 & 1,1 & 2,2 & 9,3 & 2,3 \\ 4,2 & 6,2 & 3,2 & 8,3 & 5,3 & 9,1 & 4,3 & 8,2 & 3,3 & 9,2 & 4,1 \\ 2,1 & 7,1 & 3,1 & 7,3 & 8,1 \end{bmatrix}$$

Setelah dicari rute baru dari $C'_{1\text{ baru}}$ sampai $C'_{5\text{ baru}}$ lakukan perhitungan kembali seperti pada lampiran B.

5. Evaluasi kembali $f(C_i)$, jika $f(C_i) \leq f(p_{i\text{ best}})$ maka $p_{i\text{ best}} = C_i$, setelah mendapat $p_{i\text{ best}}$ baru, maka didapat g_{best} baru.
6. Jika iterasi sudah optimal, maka algoritma berhenti. Jika tidak maka kembali ke langkah (4). Dalam perhitungan manual ini, iterasi pertama belum optimal sehingga harus melanjutkan ke iterasi berikutnya.

4.1.2 Program Algoritma PSO untuk VRPTW dengan menggunakan MATLAB

Pada skripsi ini, dihasilkan sebuah program penyelesaian VRPTW dengan menggunakan *software* MATLAB. Tujuan dari pembuatan program ini adalah untuk mendapatkan rute pengiriman barang dengan total jarak minimum dengan memperhatikan batasan kapasitas kendaraan dan batasan waktu pengiriman. Tampilan awal program ini disajikan pada gambar 4.1.



Gambar 4.1 Tampilan awal program VRPTW

Tampilan awal pada gambar 4.1 merupakan tampilan awal GUI program algoritma Particle Swarm Optimization, program ini mempunyai beberapa tombol sebagai berikut:

1. Panel data, terdiri dari tabel jumlah permintaan, tabel jarak, dan tabel Input data. Tabel-tabel ini adalah tempat untuk menampilkan data.
2. Pada tabel *input data*, terdiri dari *Edit Text* yang digunakan untuk memasukkan beberapa parameter yaitu jumlah toko, jumlah kendaraan, kapasitas, jumlah swarm, kecepatan kendaraan, waktu pelayanan, jumlah iterasi, waktu buka toko dan *epsilon*.
3. Tombol Proses, untuk memulai proses algoritma *Particle Swarm Optimization*.
4. Menu open file, berisi data yang sebelumnya sudah disimpan dalam program untuk diproses.
5. Menu Hitung rute, untuk memunculkan perhitungan manual atau perhitungan yang lebih detail.
6. Menu Hasil *record*, berisi semua hasil beberapa kali running/ proses

7. Menu Grafik konvergen, untuk mengetahui letak kekonvergenan dari hasil tersebut.
8. Tombol Close, digunakan untuk menutup tampilan program.

Adapun langkah-langkah Penerapan Algoritma *Particle Swarm Optimization* untuk *Vehicle Routing Problem With Time Windows* menggunakan program yang telah dibuat sebagai berikut:

a. *Input Data*

Untuk menginputkan data ada dua cara yang dapat dilakukan oleh pengguna, caranya sebagai berikut:

1. Cara pertama yaitu menggunakan data manual, isikan angka pada Panel Data (*Input Data*) sesuai yang dibutuhkan lalu klik tombol “*Input Data*”. Setelah itu isikan data yang akan dicari pada tabel jumlah permintaan dan tabel jarak. Kemudian klik “*Proses*” untuk diproses.
2. Cara kedua yaitu menginputkan data dengan klik pada menu “*Open File*”, kemudian pilih data yang di inginkan, selanjutnya klik tombol “*Open*” maka data yang dipilih akan dimunculkan pada jendela Panel Data dalam bentuk tabel.

Jumlah Permintaan

	Toko 1	Toko 4	Toko 6	Toko 7	Toko 10	Toko 11	Toko 14	Toko 17	Toko 18	Toko 2
1	32	11	4	5	10	7	9	18	10	

	Depot	Toko 1	Toko 4	Toko 6	Toko 7	Toko 10	Toko 11	Toko 14	Toko 1
Depot	0	5.9000	13.3000	5.6000	5.4000	18.6000	16.2000	16.4000	9.1
Toko 1	5.9000	0	8.8000	1.4000	1.7000	14.1000	11.8000	12	4.5
Toko 4	13.3000	8.8000	0	1.9000	11.2000	5.3000	3	3.2000	
Toko 6	5.6000	1.4000	1.9000	0	0.2600	15.5000	13.2000	13.4000	5.8
Toko 7	5.4000	1.7000	11.2000	0.2600	0	15.8000	13.4000	13.6000	

Input data

Jumlah Toko: Kecepatan:

Jumlah Kendaraan: Waktu Pelayanan:

Kapasitas: Jumlah Iterasi:

Jumlah Swarm: Waktu Toko Buka:

Epsilon:

Gambar 4.2 Tampilan Tabel Data

b. Proses

Sebelum data di proses, parameter-parameter yang dibutuhkan algoritma harus diisi terlebih dahulu. Setelah itu untuk memulai proses klik tombol “Proses”

Jumlah Permintaan

	Toko 1	Toko 4	Toko 6	Toko 7	Toko 10	Toko 11	Toko 14	Toko 17	Toko 18	Toko 2
1	32	11	4	5	10	7	9	18	10	

Depot	Depot	Toko 1	Toko 4	Toko 6	Toko 7	Toko 10	Toko 11	Toko 14	Toko 1
Depot	0	5.9000	13.3000	5.6000	5.4000	18.6000	16.2000	16.4000	9.1
Toko 1	5.9000	0	8.8000	1.4000	1.7000	14.1000	11.8000	12	4.5
Toko 4	13.3000	8.8000	0	1.9000	11.2000	5.3000	3	3.2000	
Toko 6	5.6000	1.4000	1.9000	0	0.2600	15.5000	13.2000	13.4000	5.8
Toko 7	5.4000	1.7000	11.2000	0.2600	0	15.8000	13.4000	13.6000	

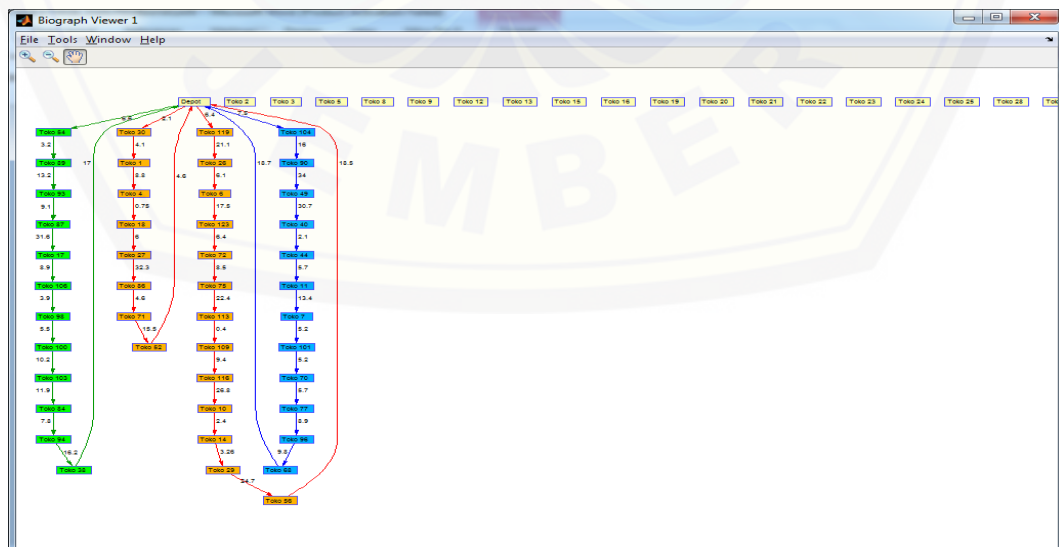
Input data

Jumlah Toko:	45	Kecepatan :	25
Jumlah Kendaraan:	4	Waktu Pelayanan:	30
Kapasitas :	200	Jumlah Iterasi:	500
Jumlah Swarm :	10	Waktu Toko Buka:	8
Epsilon :	0.01		

Gambar 4.3 Tampilan *Input* Parameter Program VRPTW

c. Output

Setelah di klik “Proses”, program akan memproses data hingga memperoleh hasil yang optimal. Output dari hasil program VRPTW berupa rute kendaraan, total jarak yang ditempuh, total waktu yang dibutuhkan selama perjalanan dan pelayanan dan kapasitas barang(dapat dilihat pada gambar 4.4 dan gambar 4.5).

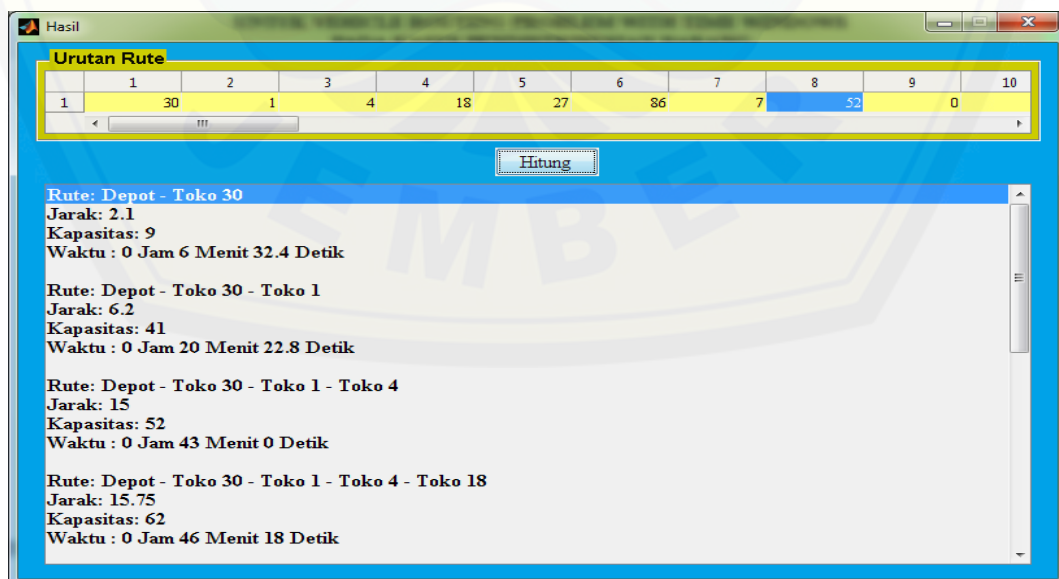


Gambar 4.4 Tampilan Output Rute Kendaraan

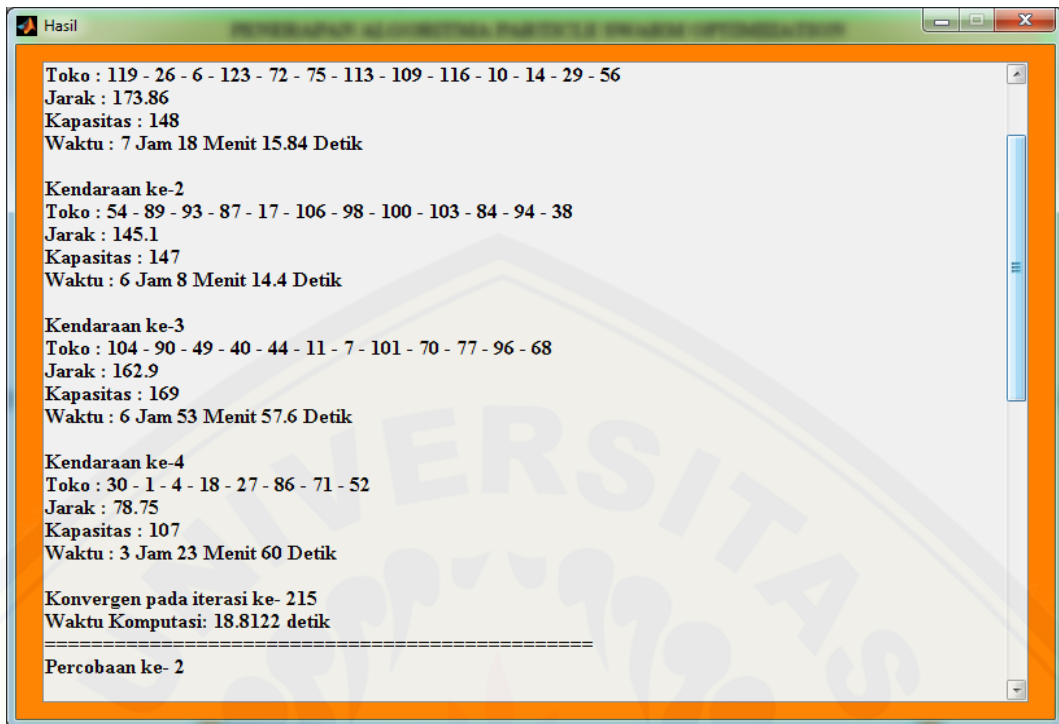


Gambar 4.5 Tampilan Output Total Jarak, Total Waktu dan Total Kapasitas

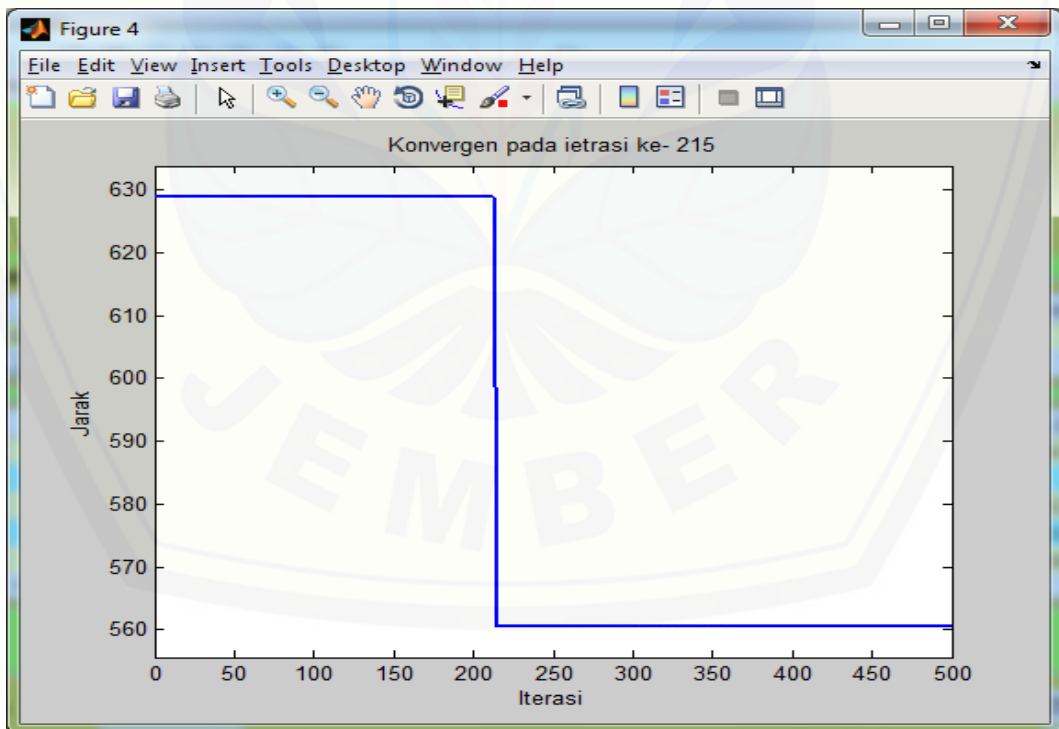
Pada tampilan output juga dapat ditampilkan hasil jarak, kapasitas dan waktu setiap toko/pangkalan untuk setiap rute kendaraan dengan menekan tombol menu “Hitung Rute” (dapat dilihat pada gambar 4.6). Sedangkan menu “Hasil Record” ini untuk memperlihatkan hasil dari beberapa kali *running*/proses (dapat dilihat pada gambar 4.7) dan menu “Grafik Konvergen” untuk memperlihatkan grafik kekonvergenan dari hasil tersebut, dalam grafik tersebut terdiri dari jarak dan iterasi (dapat dilihat pada gambar 4.8).



Gambar 4.6 Tampilan Hasil Setiap Toko untuk Setiap Rute Kendaraan



Gaambar 4.7 Tampilan Hasil Record untuk beberapa kali *running*/proses



Gambar 4.8 Tampilan Grafik Kekonvergenan

4.2 Pembahasan

4.2.1 Penerapan algoritma PSO untuk VRPTW pada kasus pendistribusian barang

Pada bagian ini akan dibahas penerapan Algoritma *Particle Swarm Optimization* untuk penyelesaian VRPTW pada perusahaan distributor gas LPG 3 kg PT. Amita Bara Sejahtera Jember yang akan menghasilkan rute optimal dengan memperhatikan kapasitas dan batasan waktu (*Time Windows*). Data tersebut diproses dengan menggunakan program Matlab dengan ukuran data yang berbeda dan perubahan parameter algoritma. Masing-masing percobaan dilakukan 10 kali *running/proses* dengan nilai parameter yang sama. Hasil percobaannya sebagai berikut:

- a. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 10

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 100 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running/proses* yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.5 Hasil Percobaan dengan jumlah *swarm* 10 dan iterasi 100 Hasil pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	633,05	571	26 Jam 38 Menit 19,2 Detik	1	3,9886
2	634,8	571	26 Jam 42 Menit 31,2 Detik	100	4,0968
3	619,5	571	26 Jam 5 Menit 48 Detik	1	7,2877
4	664,5	571	27 Jam 53 Menit 48 Detik	1	8,0895
5	636,2	571	26 Jam 45 Menit 52,8 Detik	97	3,8223
6	711,6	571	29 Jam 46 Menit 50,4 Detik	23	4,21
7	652,2	571	27 Jam 24 Menit 16,8 Detik	99	3,838
8	650,9	571	27 Jam 21 Menit 9, Detik	97	4,8111
9	631,27	571	26 Jam 34 Menit 2,88 Detik	1	6,0254
10	591,76	571	24 Jam 59 Menit 13,44 Detik	92	5,9128

Berdasarkan Tabel 4.5 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 591,76 km, dengan total kapasitas 571 tabung, dalam total waktu 24 jam 59 menit 13,44 detik, konvergen pada iterasi ke-92, dan *running time* sebesar 5,9128 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke-10 (hasil lebih rinci lihat lampiran C1).

- b. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 20

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 100 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.6 Hasil Percobaan dengan jumlah *swarm* 20 dan iterasi 100 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	596,7	571	25 Jam 11 Menit 4,8 Detik	91	9,539
2	666	571	27 Jam 57 Menit 24 Detik	1	13,9053
3	657,85	571	27 Jam 37 Menit 50,4 Detik	89	8,5908
4	578,42	571	24 Jam 27 Menit 12,48 Detik	95	7,4442
5	593	571	25 Jam 2 Menit 12 Detik	91	9,041
6	530,5	571	22 Jam 32 Menit 12 Detik	32	10,8121
7	625,66	571	26 Jam 20 Menit 35,04 Detik	1	7,3285
8	630,2	571	26 Jam 31 Menit 28,8 Detik	99	7,2382
9	673,2	571	28 Jam 14 Menit 40,8 Detik	1	11,4342
10	620,61	571	26 Jam 8 Menit 27,84 Detik	96	11,8384

Berdasarkan Tabel 4.6 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 530,5 km, dengan total kapasitas 571 tabung, dalam total waktu 22 jam 32 menit 12 detik, konvergen pada iterasi ke-32, dan *running time* sebesar 10,8121detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke-6 (hasil lebih rinci lihat lampiran C1).

- c. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 30

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 100 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.7 Hasil Percobaan dengan jumlah *swarm* 30 dan iterasi 100 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	591,7	571	24 Jam 59 Menit 4,8 Detik	27	11,0219
2	635	571	26 Jam 43 Menit 0 Detik	99	10,921
3	651,3	571	27 Jam 22 Menit 7,2 Detik	99	13,6904
4	575,55	571	24 Jam 20 Menit 19,2 Detik	96	11,1472
5	579,15	571	24 Jam 28 Menit 57,6 Detik	90	10,8584
6	600,5	571	25 Jam 20 Menit 12 Detik	95	12,192
7	614,7	571	25 Jam 54 Menit 16,8 Detik	92	12,7164
8	625,3	571	26 Jam 19 Menit 43,2 Detik	29	12,6046
9	473,8	571	20 Jam 16 Menit 7,2 Detik	42	11,5162
10	605,85	571	25 Jam 33 Menit 2,4 Detik	91	10,9337

Berdasarkan Tabel 4.7 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 473,8 km, dengan total kapasitas 571 tabung, dalam total waktu 20 jam 16 menit 7,2 detik, konvergen pada iterasi ke-42, dan *running time* sebesar 11,5162detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke-9 (hasil lebih rinci lihat lampiran C1).

- d. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 10
 Kecepatan rata-rata = 25 km/jam
 Waktu pelayanan = 30 detik
 Jumlah iterasi = 500 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.8 Hasil Percobaan dengan jumlah *swram* 10 dan iterasi 500 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	599,72	571	25 Jam 18 Menit 19,68 Detik	258	19,4608
2	651,2	571	27 Jam 21 Menit 52,8 Detik	16	17,6626
3	544,96	571	23 Jam 6 Menit 54,24 Detik	299	17,9818
4	579,46	571	24 Jam 29 Menit 42,24 Detik	252	18,9051
5	638,3	571	26 Jam 50 Menit 55,2 Detik	21	18,4362
6	564,3	571	23 Jam 53 Menit 19,2 Detik	238	19,0492
7	620,16	571	26 Jam 7 Menit 23,04 Detik	253	20,3755
8	650,2	571	27 Jam 19 Menit 28,8 Detik	230	18,0013
9	588,4	571	24 Jam 51 Menit 9,6 Detik	210	19,7281
10	666,06	571	27 Jam 57 Menit 32,64 Detik	213	20,02,59

Berdasarkan Tabel 4.8 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 544,96 km, dengan total kapasitas 571 tabung, dalam total waktu 23 jam 6 menit 54,24 detik, konvergen pada iterasi ke- 299, dan *running time* sebesar 17,9818 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke-3 (hasil lebih rinci lihat lampiran C1).

- e. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 20

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 500 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.9 Hasil Percobaan dengan jumlah *swram* 20 dan iterasi 500 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	604,16	571	25 Jam 28 Menit 59,04 Detik	219	35,0824
2	614,76	571	25 Jam 54 Menit 25,44 Detik	15	37,9438
3	524,41	571	22 Jam 17 Menit 35,04 Detik	273	35,3932
4	512,35	571	21 Jam 48 Menit 38,4 Detik	43	37,5513
5	625,1	571	26 Jam 19 Menit 14,4 Detik	249	37,3344
6	606,85	571	25 Jam 35 Menit 26,4 Detik	25	33,9749
7	500,8	571	21 Jam 20 Menit 55,2 Detik	46	34,9506
8	584,1	571	24 Jam 40 Menit 50,4 Detik	251	34,499
9	542,6	571	23 Jam 1 Menit 14,4 Detik	248	34,7302

10	558,95	571	23 Jam 40 Menit 28,8 Detik	221	35,0267
----	--------	-----	-------------------------------	-----	---------

Berdasarkan Tabel 4.9 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 500,8 km, dengan total kapasitas 571 tabung, dalam total waktu 21 jam 20 menit 55,2 detik, konvergen pada iterasi ke- 46, dan *running time* sebesar 34,9506 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke-7 (hasil lebih rinci lihat lampiran C1).

- f. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 30

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 500 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running/proses* yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.10 Hasil Percobaan dengan jumlah *swarm* 30 dan iterasi 500 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	538,3	571	22 Jam 50 Menit 55,2 Detik	252	54,9412
2	575,86	571	24 Jam 21 Menit 3,84 Detik	246	53,8052
3	576,8	571	24 Jam 23 Menit 19,2 Detik	219	51,7833
4	530,01	571	22 Jam 31 Menit 1,44 Detik	231	54,0902
5	619,2	571	26 Jam 5 Menit 4,8 Detik	22	52,6226
6	547,5	571	23 Jam 13 Menit 0 Detik	223	51,942
7	558,65	571	23 Jam 39 Menit 45,6 Detik	239	51,9166
8	603,26	571	25 Jam 26 Menit 49,44 Detik	44	54,2065
9	492,9	571	21 Jam 1 Menit 57,6 Detik	218	53,2096
10	583,96	571	24 Jam 40 Menit 30,24 Detik	236	55,0771

Berdasarkan Tabel 4.10 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 492,9 km, dengan total kapasitas 571 tabung, dalam total waktu 21 jam 1 menit 57,6 detik, konvergen pada iterasi ke- 218, dan *running time* sebesar 53,2096 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 9 (hasil lebih rinci lihat lampiran C1).

g. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 10

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 1000 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.11 Hasil Percobaan dengan jumlah *swarm* 10 dan iterasi 1000 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	575,9	571	24 Jam 21 Menit 9,6 Detik	407	35,2031
2	292,7	571	249 Jam 1 Menit 28,8 Detik	1	34,7283
3	589,1	571	24 Jam 52 Menit 50,4 Detik	219	51,7833
4	530,01	571	22 Jam 31 Menit 1,44 Detik	1	36,4185
5	527,35	571	22 Jam 24 Menit 38,4 Detik	356	36,1643
6	675,4	571	28 Jam 19 Menit 57,6 Detik	1	35,9816
7	600,4	571	25 Jam 19 Menit 57,6 Detik	19	38,5253
8	640,45	571	26 Jam 56 Menit 4m8 detik	393	41,6427
9	580,6	571	24 Jam 32 Menit 24 Detik	308	34,9241
10	648,4	571	27 Jam 15 Manit 9,6 Detik	370	36,1729

Berdasarkan Tabel 4.11 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 527,35 km, dengan total kapasitas 571 tabung, dalam total waktu 22 jam 24 menit 38,4 detik, konvergen pada iterasi ke- 356, dan *running time* sebesar 36,1643 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 5 (hasil lebih rinci lihat lampiran C1).

h. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak

dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 20

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 1000 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.12 Hasil Percobaan dengan jumlah *swram* 20 dan iterasi 1000 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	561,9	571	23 Jam 47 Menit 33,6 Detik	513	70,1874
2	596,6	571	25 Jam 10 Menit 540,4 Detik	37	71,4717
3	562,4	571	23 Jam 48 Menit 45,6 Detik	347	72,0845
4	552,2	571	23 Jam 24 Menit 16,8 Detik	389	71,2475
5	542,46	571	23 Jam 0 Menit 54,24 Detik	357	75,6274
6	589,5	571	24 Jam 53 Menit 48 Detik	376	71,9883
7	575,5	571	24 Jam 20 Menit 12 Detik	21	71,2234
8	613,76	571	25 Jam 52 Menit 1,44 Detik	361	75,4912
9	600,85	571	25 Jam 21 Menit 2,4 Detik	367	75,43,89
10	576,96	571	24 Jam 23 Menit 42,24 Detik	24	72,1457

Berdasarkan Tabel 4.12 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 542,46 km, dengan total kapasitas 571 tabung, dalam total waktu 23 jam 0 menit 54,24 detik, konvergen pada iterasi ke- 357, dan *running time* sebesar 75,6274 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 5 (hasil lebih rinci lihat lampiran C1).

- i. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 5 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A1.

Jumlah toko/pangkalan = 45 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 30

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 1000 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running/proses* yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.13 Hasil Percobaan dengan jumlah *swarm* 30 dan iterasi 1000 pada data tanggal 5 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	512,8	571	21 Jam 49 Menit 43,2 Detik	426	115,9802
2	522,4	571	22 Jam 12 Menit 45,6 Detik	375	109,2459
3	561	571	23 Jam 45 Menit 24 Detik	369	111,8673
4	554,35	571	23 Jam 29 Menit 26,4 Detik	328	112,037
5	538,8	571	22 Jam 52 Menit 7,2 Detik	329	110,4427
6	553,7	571	23 Jam 27 Menit 52,8 Detik	381	117,9408
7	553,7	571	23 Jam 27 Menit 52,8 Detik	381	117,9408
8	582,6	571	24 Jam 37 Menit 14,4 Detik	25	109,3713
9	568,3	571	24 Jam 2 Menit 55,2 Detik	422	105,7292
10	520,26	571	22 Jam 7 Menit 37,44 Detik	23	105,6368

Berdasarkan Tabel 4.13 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 520,26 km, dengan total kapasitas 571 tabung, dalam total waktu 22 jam 7 menit 37,44 detik, konvergen pada iterasi ke- 23, dan *running time* sebesar 105,6368 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 10 (hasil lebih rinci lihat lampiran C1).

- j. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 10

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 100 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.14 Hasil Percobaan dengan jumlah *swarm* 10 dan iterasi 100 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	320,792	445	13 Jam 53 Menit 24,048 Detik	1	5,6619
2	337,9	445	14 Jam 34 Menit 27,6 Detik	90	5,189
3	333,4	445	14 Jam 23 Menit 39,6 Detik	4	5,3055
4	299,982	445	13 Jam 3 Menit 27,408 Detik	16	5,2516
5	276,75	445	12 Jam 7 Menit 42 Detik	34	5,2383
6	305,542	445	13 Jam 16 Menit 48,048 Detik	89	5,3502
7	284	445	12 Jam 25 Menit 6 Detik	21	5,3411
8	274,432	445	12 Jam 2 Menit 8,208 Detik	95	5,2988
9	300,15	445	13 Jam 3 Menit 51,6 Detik	20	5,3572
10	271,2	445	11 Jam 54 Menit 22,8 Detik	13	5,2932

Berdasarkan Tabel 4.14 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 271,2km, dengan total kapasitas 445 tabung, dalam total waktu 11 jam 54 menit 22,8 detik, konvergen pada iterasi ke- 13, dan *running time* sebesar 5,2932 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 10 (hasil lebih rinci lihat lampiran C2).

k. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak

dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 20

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 100 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.15 Hasil Percobaan dengan jumlah *swram* 20 dan iterasi 100 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	279,45	445	12 Jam 14 Menit 10,8 Detik	22	10,5902
2	302,76	445	13 Jam 10 Menit 7,44 Detik	2	10,5107
3	269,932	445	11 Jam 51 Menit 20,208 Detik	17	10,4129
4	275,7	445	12 Jam 5 Menit 10,8 Detik	28	11,3103
5	275,2	445	12 Jam 3 Menit 58,8 Detik	19	11,5574
6	282,832	445	12 Jam 22 Menit 17,808 Detik	14	11,2422
7	271,542	445	11 Jam 55 Menit 12,048 Detik	13	10,7848
8	288,032	445	12 Jam 34 Menit 46,608 Detik	37	10,5734
9	263,6	445	11 Jam 36 Menit 8,4 Detik	32	10,8284
10	285,51	445	12 Jam 28 Menit 43,44 Detik	14	10,5918

Berdasarkan Tabel 4.15 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 263,6 km, dengan total kapasitas 445 tabung, dalam total waktu 11 jam 36 menit 8,4 detik, konvergen pada iterasi ke- 32, dan *running time* sebesar 10,8284 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 9 (hasil lebih rinci lihat lampiran C2).

1. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 30

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 100 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running/proses* yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.16 Hasil Percobaan dengan jumlah *swarm* 30 dan iterasi 100 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	257,082	445	11 Jam 20 Menit 29,808 Detik	23	17,7705
2	266,8	445	11 Jam 43 Menit 49,2 Detik	28	16,5774
3	259,6	445	11 Jam 26 Menit 32,4 Detik	29	15,8673
4	283,66	445	12 Jam 24 Menit 17,4 Detik	86	15,8145
5	285	445	12 Jam 27 Menit 30 Detik	1	15,9577
6	244,95	445	10 Jam 51 Menit 22,8 Detik	16	16,2006
7	281,15	445	12 Jam 18 Menit 15,6 Detik	15	16,3028
8	256	445	11 Jam 17 Menit 54 Detik	44	15,9674
9	265,75	445	11 Jam 41 Menit 18 Detik	26	15,9144
10	301,042	445	13 Jam 6 Menit 0,048 Detik	100	16,1079

Berdasarkan Tabel 4.16 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 244,95 km, dengan total kapasitas 445 tabung, dalam total waktu 10 jam 51 menit 22,8 detik, konvergen pada iterasi ke- 16, dan *running time* sebesar 16,2006 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 6 (hasil lebih rinci lihat lampiran C2).

m. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 10

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 500 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.17 Hasil Percobaan dengan jumlah *swarm* 10 dan iterasi 500 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	295,75	445	12 Jam 53 Menit 18 Detik	20	17,0858
2	311,22	445	13 Jam 30 Menit 25,68 Detik	23	15,1006
3	315,45	445	13 Jam 40 Menit 34,8 Detik	19	17,5054
4	283,85	445	12 Jam 24 Menit 44,4 Detik	37	14,9217
5	313,5	445	13 Jam 35 Menit 54 Detik	29	15,1395
6	307,85	445	13 Jam 22 Menit 20,4 Detik	19	15,8454
7	316,16	445	13 Jam 42 Menit 17,04 Detik	16	15,4881
8	307,71	445	13 Jam 22 Menit 0,24 Detik	16	14,9249
9	265,95	445	11 Jam 41 Menit 46,8 Detik	242	15,7253
10	286,15	445	12 Jam 30 Menit 15,6 Detik	231	15,5641

Berdasarkan Tabel 4.17 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 265,95 km, dengan total kapasitas 445 tabung, dalam total waktu 11 jam 41 menit 46,8 detik, konvergen pada iterasi ke- 242, dan *running time* sebesar 15,7253 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 9 (hasil lebih rinci lihat lampiran C2).

- n. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 20

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 500 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running/proses* yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.18 Hasil Percobaan dengan jumlah *swarm* 20 dan iterasi 500 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	313,95	445	13 Jam 36 Menit 58,8 Detik	16	31,286
2	265,46	445	11 Jam 40 Menit 36,24 Detik	60	30,3302
3	283,3	445	12 Jam 23 Menit 25,2 Detik	16	30,3466
4	320,3	445	13 Jam 52 Menit 13,2 Detik	3	29,9787
5	307,4	445	13 Jam 21 Menit 15,6 Detik	2	30,6851
6	261,16	445	11 Jam 30 Menit 17,04 Detik	45	30,5082
7	303,982	445	13 Jam 13 Menit 3,408 Detik	18	31,0236
8	278,9	445	12 Jam 12 Menit 51,6 Detik	22	30,3592
9	319,15	445	13 Jam 49 Menit 27,6 Detik	14	30,9152
10	254,742	445	11 Jam 14 Menit 52,848 Detik	39	30,3055

Berdasarkan Tabel 4.18 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 254,742 km, dengan total kapasitas 445 tabung, dalam total waktu 11 jam 14 menit 52,848 detik, konvergen pada iterasi ke- 39, dan *running time* sebesar 30,3055detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 10 (hasil lebih rinci lihat lampiran C2).

- o. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 30

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 500 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.19 Hasil Percobaan dengan jumlah *swarm* 30 dan iterasi 500 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	306,75	445	13 Jam 19 Menit 42 Detik	251	45,7584
2	288,96	445	12 Jam 37 Menit 0,24 Detik	10	47,1879
3	271,082	445	11 Jam 54 Menit 5,808 Detik	19	47,3385
4	298,56	445	13 Jam 0 Menit 2,64 Detik	18	45,7997
5	297,05	445	12 Jam 56 Menit 25,2 Detik	11	46,3003
6	288,2	445	12 Jam 35 Menit 10,8 Detik	24	47,2429
7	249,86	445	11 Jam 3 Menit 9,84 Detik	22	51,1685
8	291,76	445	12 Jam 43 Menit 43,44 Detik	13	47,3821
9	258,032	445	11 Jam 22 Menit 46,608 Detik	32	46,9458
10	290,7	445	12 Jam 41 Menit 10,8 Detik	14	47,1573

Berdasarkan Tabel 4.19 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 249,86 km, dengan total kapasitas 445 tabung, dalam total waktu 11 jam 3 menit 9,84 detik, konvergen pada iterasi ke- 22, dan *running time* sebesar 51,1685 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 7 (hasil lebih rinci lihat lampiran C2).

- p. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 10

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 1000 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running/proses* yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.20 Hasil Percobaan dengan jumlah *swarm* 10 dan iterasi 1000 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	288,36	445	12 Jam 35 Menit 33,84 Detik	19	33,0596
2	269,25	445	11 Jam 49 Menit 42 Detik	34	30,5382
3	273,51	445	11 Jam 59 Menit 55,44 Detik	58	30,5102
4	308,9	445	13 Jam 24 Menit 51,6 Detik	2	30,2471
5	284,442	445	12 Jam 26 Menit 9,648 Detik	27	30,3421
6	290,25	445	12 Jam 40 Menit 6 Detik	43	30,9035
7	309,532	445	13 Jam 26 Menit 22,608 Detik	8	30,2579
8	282,65	445	12 Jam 21 Menit 51,6 Detik	23	30,0449
9	290,85	445	12 Jam 41 Menit 32,4 Detik	23	29,9364
10	277,71	445	12 Jam 100 Menit 0,24 Detik	389	32,0789

Berdasarkan Tabel 4.20 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 269,25 km, dengan total kapasitas 445 tabung, dalam total waktu 11 jam 49 menit 42 detik, konvergen pada iterasi ke- 34, dan *running time* sebesar 30,5382 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 2 (hasil lebih rinci lihat lampiran C2).

q. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 20

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 1000 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.21 Hasil Percobaan dengan jumlah *swarm* 20 dan iterasi 1000 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	287,8	445	12 Jam 34 Menit 13,2 Detik	9	61,4824
2	286,65	445	12 Jam 31 Menit 27,6 Detik	19	61,8068
3	276,7	445	12 Jam 7 Menit 34,8 Detik	119	60,0018
4	273,55	445	12 Jam 0 Menit 1,2 Detik	21	65,969
5	286,7	445	12 Jam 31 Menit 34,8 Detik	15	62,32,32
6	273,2	445	11 Jam 59 Menit 10,8 Detik	15	64,7781
7	255,76	445	11 Jam 17 Menit 19,44 Detik	52	63,5561
8	241,06	445	10 Jam 42 Menit 2,64 Detik	347	60,8553
9	262,4	445	11 Jam 33 Menit 15,6 Detik	24	62,5773
10	252,47	445	11 Jam 9 Menit 25,68 Detik	380	61,9555

Berdasarkan Tabel 4.21 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 241,06 km, dengan total kapasitas 445 tabung, dalam total waktu 10 jam 42 menit 2,64 detik, konvergen pada iterasi ke- 347, dan *running time* sebesar 60,8553 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 8 (hasil lebih rinci lihat lampiran C2).

r. Ditentukan percobaan parameter Algoritma *Particle Swarm Optimization* yaitu jumlah permintaan yang digunakan pada tanggal 14 April 2016 dan data jarak

dari depo ke toko/pangkalan maupun jarak antar toko/pangkalan ditunjukkan pada lampiran A2.

Jumlah toko/pangkalan = 40 toko/pangkalan

Jumlah kendaraan = 4 kendaraan

Kapasitas kendaraan = 200 tabung

Jumlah swarm = 30

Kecepatan rata-rata = 25 km/jam

Waktu pelayanan = 30 detik

Jumlah iterasi = 1000 iterasi

Nilai $c_1 = c_2 = 2$

Nilai r_1 dan r_2 adalah *random* antara 0 dan 1

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running*/proses yang hasilnya dapat dilihat pada lampiran A1.

Tabel 4.22 Hasil Percobaan dengan jumlah *swram* 30 dan iterasi 1000 pada data tanggal 14 April 2016

Running ke-	Total Jarak (Km)	Toatal Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
1	294,882	445	12 Jam 51 Menit 13,008 Detik	37	97,7626
2	277,45	445	12 Jam 9 Menit 22,8 Detik	17	94,8806
3	289,35	445	12 Jam 37 Menit 56,4 Detik	119	60,0018
4	288,192	445	12 Jam 35 Menit 9,648 Detik	2	95,9976
5	253,832	445	11 Jam 12 Menit 41,808 Detik	421	100,7064
6	261	445	11 Jam 30 Menit 8,4 Detik	20	93,4709
7	247,382	445	10 Jam 57 Menit 13,008 Detik	382	93,8841
8	307,75	445	13 Jam 12 Menit 6 Detik	16	94,1468
9	306,282	445	13 Jam 18 Menit 34,68 Detik	11	95,3138
10	277,5	445	12 Jam 9 Menit 30 Detik	16	96,4363

Berdasarkan Tabel 4.22 dapat dilihat bahwa algoritma *Particle Swarm Optimization* mencapai hasil optimal dengan total jarak 247,382 km, dengan total kapasitas 445 tabung, dalam total waktu 10 jam 57 menit 13,008 detik, konvergen pada iterasi ke- 382, dan *running time* sebesar 93,8841 detik. Sehingga pada data ini terpilih hasil rute yang lebih optimal pada *running time* ke- 7 (hasil lebih rinci lihat lampiran C2).

Berdasarkan beberapa percobaan yang telah dilakukan pada Algoritma *Particle Swarm Optimization*, secara umum terlihat bahwa total jarak minimum yang dihasilkan dari setiap *running* tidak sama. Hal ini dikarenakan populasi yang dibangkitkan menggunakan bilangan *random*. Jika nilai bilangan *random* mendukung untuk mendapatkan jarak minimum maka akan memperoleh hasil yang optimal, namun jika nilai bilangan *random* tidak mendukung untuk mendapatkan jarak minimum maka tidak akan memperoleh hasil yang optimal.

Pada percobaan di atas menggunakan dua data yang berbeda dan setiap data diproses dengan memasukkan jumlah *swarm* dan jumlah iterasi yang berbeda. Dari kelimabelas percobaan tersebut diperoleh total jarak yang minimal, total waktu yang diperlukan untuk proses pengiriman barang dan setiap kendaraan telah memenuhi kendala dari VRPTW (kapasitas kendaraan, *time windows* dan rute). Dengan demikian Penerapan Algoritma *Particle Swarm Optimization* sangat efektif untuk menyelesaikan *Vehicle Routing Problem With Time Windows* pada kasus pendistribusian barang karena dapat mengoptimalkan jarak minimum dengan batasan waktu tertentu (*Time Windows*) sehingga dapat meminimalkan jumlah kendaraan dan meminimalkan biaya yang dikeluarkan.

4.2.2 Pengaruh Parameter terhadap solusi optimal yang diperoleh

Parameter yang digunakan dan dapat berubah-ubah dalam masalah ini yaitu jumlah toko, jumlah permintaan barang, jumlah *swarm* dan jumlah iterasi. Pengaruh dari setiap parameter tersebut sebagai berikut:

- a. Jika jumlah toko dan jumlah permintaan semakin besar tidak sesuai dengan kendaraan yang dimiliki depo, maka solusi yang diperoleh tidak akan optimal

karena kapasitas kendaraan tidak memenuhi jumlah permintaan yang semakin besar dan jumlah toko yang banyak.

- b. Pada Algoritma *Particle Swarm Optimization* ukuran/jumlah swarm sangat diperlukan untuk memperoleh suatu solusi yang optimal. Swarm dapat diartikan sebagai suatu vektor solusi atau banyaknya kemungkinan untuk menemukan posisi partikel yang terbaik/optimal. Semakin kecil ukuran swarm maka kemungkinan untuk menemukan posisi partikel yang terbaik/optimal semakin sedikit. Jika semakin besar ukuran/jumlah swarm maka membuat perhitungan jadi panjang namun akan menemukan posisi partikel yang terbaik/optimal seperti pada percobaan pertama dengan percobaan kedua (Tabel 4.5 dan Tabel 4.6), percobaan ketiga dengan percobaan keempat (Tabel 4.8 dan Tabel 4.9), percobaan kesebelas dengan percobaan keduabelas (Tabel 4.15 dan Tabel 4.16) dan lain-lain.
- c. Semakin besar iterasi yang digunakan maka hasil yang diperoleh akan semakin optimal/valid, seperti pada data pertama hasil percobaan pertama, percobaan keempat, dan percobaan ke tujuh menggunakan jumlah swarm yang sama tapi jumlah itersi yang berbeda-beda. Namun jika iterasi semakin besar maka *running time* juga semakin lain.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada tugas akhir ini, maka dapat diambil kesimpulan sebagai berikut:

- a. Penerapan Algoritma *Particle Swarm Optimization* untuk *Vehicle Routing Problem With Time Windows* pada kasus pendistribusian barang di PT. Amita Bara Sejahtera Jember yaitu perusahaan distributor gas LPG 3 kg, dilakukan sebanyak 9 percobaan untuk data pertama dan data kedua. Pada percobaan tersebut mendapatkan hasil total jarak yang optimal dan memenuhi kendala dari VRPTW. Sehingga Algoritma PSO sangat efektif untuk menyelesaikan VRPTW pada kasus pendistribusian barang.
- b. Pada kasus ini parameter (jumlah swarm dan jumlah iterasi) sangat berpengaruh untuk memperoleh hasil pencarian rute kendaraan yang optimal, karena semakin besar jumlah *swarm* maka kemungkinan solusi optimal yang akan diperoleh semakin banyak, dan semakin besar iterasi yang digunakan maka akan memperoleh hasil total jarak yang lebih optimal.

5.2 Saran

Penelitian lebih lanjut dapat dilakukan untuk mengembangkan dan bisa mengkombinasikan Algoritma *Particle Swarm Optimization* dengan Algoritma lain dalam menyelesaikan *Vehicle Routing Problem With Time Windows* maupun *Vehicle Routing Problem With Multi Time Windows*. Selain itu, dapat menerapkan algoritma lain pada variasi VRP yang lain seperti *capacitated vehicle routing problem with pickup and delivery*, *distance constrained capacitated vehicle routing problem* dan lain-lain.

DAFTAR PUSTAKA

- Amini, S., Javanshir, H., dan Tavakkoli-Moghaddam, R. 2010. *A PSO Approach for Solving VRPTW with Relat Case Study*. IJRRAS.
- Arhami, M., dan Desiani, A. 2005. *Pemrograman MATLAB*. Yogyakarta: ANDI
- Chartrand, G dan O.R Oellermann. 1993. *Applied and Algorithmic Graph Theory*. New York : McGraw Hill, Inc.
- Doerner, K., Gronalt, M., Hartl, R.F., Reimann, M., Strauss, C., dan Stummer, M. 2001. *Savings Ants for the Vehicle Routing Problem*. Austria : Institute of Management Science, University of Vienna.
- Iskandar. 2010. *Model Optimasi Vehicle Routing Problem dan Implementasinya*. Tidak Diterbitkan. Tesis. Bogor: Program Pascasarjana IPB.
- Kuncoro, H.A, Prasetyaningrum, I., dan Asmara, R. 2011. *Penentuan Rute Pendistribusian Surat Kabar Dengan Time Window, Aplikasi Algoritma Tabu Search*. Surabaya : Jurusan Teknik Informatika, ITS Surabaya.
- Maryati, I., Gunawan, Pickerling, C., dan Wibowo, H.K. 2010. *Vehicle Routing Problem Berbasis Ant Colony System Untuk Optimasi Penentuan Rute Kendaraan Pada Sistem Distribusi Barang Dan Jasa*. Surabaya : Teknik Informatika, Sekolah Tinggi Teknik Surabaya.
- Munir, R. 2001. *Matematika Diskrit CV*. Bandung : Informatika.
- Pornsing, C. 2014. *A Particle Swarm Optimization For The Vehicle Routing Problem*. Rhode Island : University Of Rhode Island.
- Prasetyo, dan Wahyu Agung. 2003. *Analisis dan Desain Sistem Kontrol dengan MATLAB*. Yogyakarta: ANDI.

Setiawan, Bayu Arif. 2007. *Optimasi Proses Distribusi Barang Berdasarkan Algoritma TILLMAN & COCHRAN*. Jember: Universitas Jember.

Sundarningsih, D., Mahmudy, W.F., dan Sutrisno. 2014. *Penerapan Algoritma Genetika untuk Optimasi Vehicle Routing Problem With Time Window (VRPTW) Studi Kasus Air Minum Kemasan*. Malang : Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Utami, R.A, dan Suyanto. 2012. *Penerapan Algoritma Ant Colony System (ACS) Pada Vehicle Routing Problem With Time Windows (VRPTW)*. Malang : Fakultas Teknik Informatika, Universitas Telkom.

Xu,S.H, Liu,J.P, Zhang,F.H, Wang,L., dan Sun,L.J. 2015. *A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem With Time Windows*. Switzerland : Licensee MDPI, Basel, Switzerland.

LAMPIRAN

LAMPIRAN A. Data Permintaan dan Data Jarak perhari

Lampiran A1.

DATA PERMINTAAN GAS LPG 3 Kg (Jum'at, 5 April 2016)

Kode Toko/ Pangkalan	1	4	6	7	10	11	14	17	18	26	27	29	30	38	40	44	49	52	54	56	68	70	71	72	75
Jumlah Permintaan	32	11	4	5	10	7	9	18	10	14	7	8	9	19	30	32	9	22	4	12	16	11	7	5	31
	77	84	86	87	89	90	93	94	96	98	100	101	103	104	106	109	113	116	119	123					
	6	8	9	15	19	24	15	6	7	7	12	10	5	12	19	6	9	6	19	15					

DATA JARAK DARI DEPO KE SETIAP TOKO DAN ANTAR TOKO

Kode Toko/ Pangkalan	1 (depo)	2	5	7	8	11	12	15	18	19	27	28	30	31	39	41	45	50	53	55	57
1	0	5,9	13,3	5,6	5,4	18,6	16,2	16,4	9,1	14	9	14,3	13,3	2,1	17	11,5	11,2	30	4,6	6,6	18,5
2	5,9	0	8,8	1,4	1,7	14,1	11,8	12	4,5	9,5	4,8	10,9	8,8	4,1	13,5	7	6,8	26,6	1,4	5,9	15,2
5	13,3	8,8	0	1,9	11,2	5,3	3	3,2	6	0,75	8,2	5,2	0,06	10,4	5,1	1,8	2,7	35,4	9,6	7,9	24,6
7	5,6	1,4	1,9	0	0,26	15,5	13,2	13,4	5,8	11	6,1	12,3	10,2	4,7	15	8,4	8,2	24,5	2,9	7,3	13,7
8	5,4	1,7	11,2	0,26	0	15,8	13,4	13,6	6	11,2	6,3	12,6	10,5	4,4	15,2	8,7	8,4	24,3	3,1	7,6	13,5
11	18,6	14,1	5,3	15,5	15,8	0	3,5	2,4	11,4	5,1	13,4	8,9	5,3	15,8	6,2	7,1	8	32	12,5	13,3	30
12	16,2	11,8	3	13,2	13,4	3,5	0	2,4	9	2,8	11	5,6	2,9	13,4	2,9	4,8	5,7	38,4	12,6	10,9	27,6
15	16,4	12	3,2	13,4	13,6	2,4	2,4	0	9,2	3,95	11,4	15,5	3,26	13,6	8,3	5	5,9	38,6	12,8	11,1	27,8
18	9,1	4,5	6	5,8	6	11,4	9	9,2	0	6,75	2	4,1	6,1	2,5	7,2	7,5	4	29,1	9,5	8,1	22,6
19	14	9,5	0,75	11	11,2	5,1	2,8	3,95	6,75	0	8,8	6	0,7	5,2	4,9	2,4	3,5	35,9	10,1	8,4	25,1
27	9	4,8	8,2	6,1	6,3	13,4	11	11,4	2	8,8	0	10,1	8,1	8,5	8,7	6,3	7,6	37,1	13,7	9,5	23,7
28	14,3	10,9	5,2	12,3	12,6	8,9	5,6	15,5	4,1	6	10,1	0	5,3	11,1	3,1	4,1	3,2	34,1	14,9	10	24,2
30	13,3	8,8	0,06	10,2	10,5	5,3	2,9	3,26	6,1	0,7	8,1	5,3	0	10,4	5,1	1,9	2,8	26,7	9,7	8	24,7
31	2,1	4,1	10,4	4,7	4,4	15,8	13,4	13,6	2,5	5,2	8,5	11,1	10,4	0	19,1	13,5	13,2	32	6,6	8,6	20,5
39	17	13,5	5,1	15	15,2	6,2	2,9	8,3	7,2	4,9	8,7	3,1	5,1	19,1	0	6,9	6	36,7	16,2	12,3	25,7
41	11,5	7	1,8	8,4	8,7	7,1	4,8	5	7,5	2,4	6,3	4,1	1,9	13,5	6,9	0	2,1	30,7	11,5	10,2	24,6
45	11,2	6,8	2,7	8,2	8,4	8	5,7	5,9	4	3,5	7,6	3,2	2,8	13,2	6	2,1	0	29	9,8	8,5	23
50	30	26,6	35,4	24,5	24,3	32	38,4	38,6	29,1	35,9	37,1	34,1	26,7	32	36,7	30,7	29	0	27	32,1	16,4

53	4,6	1,4	9,6	2,9	3,1	12,5	12,6	12,8	9,5	10,1	13,7	14,9	9,7	6,6	16,2	11,5	9,8	27	0	6,8	15,9
55	6,6	5,9	7,9	7,3	7,6	13,3	10,9	11,1	8,1	8,4	9,5	10	8	8,6	12,3	10,2	8,5	32,1	6,8	0	15,9
57	18,5	15,2	24,6	13,7	13,5	30	27,6	27,8	22,6	25,1	23,7	24,2	24,7	20,5	25,7	24,6	23	16,4	15,9	15,9	0
69	18,7	15,3	24,8	13,9	13,7	30,2	27,8	25	27,2	25,3	28,2	26,1	24,9	20,7	30,3	28,7	27	16,5	16	21	4,5
71	7,6	6,6	8,2	8	8,3	11,1	11,3	11,4	7,4	8,7	8,8	9,2	8,3	9,6	12	9,6	8	32,1	7,7	2,5	24
72	18,1	14,8	24,2	13,3	13,1	27,1	27,3	27,4	25,7	24,7	26,2	25,1	24,3	20,1	27,9	27,7	26	20,4	15,5	16,2	3,8
73	20,3	17,3	16,8	15,9	15,7	32,3	29,8	20	27,5	17,3	28,2	25,8	16,9	22,3	26,2	29,4	27,7	23,2	18	16,7	4,2
76	17,7	14,4	23,8	12,9	12,7	39,2	26,8	26	23,7	24,3	24,8	25,6	23,9	19,7	27,9	25,7	24	21,2	15,1	4,7	5,1
78	14,4	11,8	21,3	10,4	10,1	26,6	24,3	24,5	25,8	21,8	26,8	24,5	21,4	16,4	26,6	27,8	26,1	38,2	12,3	5,2	5,9
85	7,1	6,2	7,8	7,6	7,8	13,2	10,8	11	5,6	8,6	10,7	7,6	7,9	9,1	9,4	7,6	5,9	37,2	6,9	4,7	24,9
87	22,7	19,3	28,8	17,9	17,6	33,3	31	34	30	29,2	34,1	32,3	28,9	24,7	34,2	32	30,3	8,9	20	25	17,3
88	21,7	18,3	27,8	16,9	16,6	32,3	30	31	31,6	28,1	35,2	33,1	27,8	23,7	35,8	33,6	31,9	77	19,3	23,9	17,9
90	8,5	4,4	6,9	5,8	6	12,2	9,9	11,1	4,7	7,6	7,9	6,7	6,9	10,5	8,6	6,7	5	37,9	5,1	3,2	23,6
91	10,8	16,3	21,6	4,9	14,7	26,9	24,6	24,8	21,1	22,1	25,1	26,2	22,7	12,8	29	22,1	20,4	34	17	7,1	10,5
94	32,6	29,2	38,7	27,8	27,6	43	40,7	41	31,9	38,2	32,9	33,9	38,8	34,6	36,1	33,9	32,2	11	28,1	34,9	15,9
95	3,1	7,5	13,3	8,9	9,2	18,6	16,3	15,5	11,3	13,8	24,3	14,6	13,4	5,1	16,2	13,3	11,6	40,2	8,2	7,4	23,2
97	17,8	14,8	23,9	13	12,8	28,2	26,9	27,1	20,5	24,4	25,1	26,2	24	19,8	27,9	22,5	10,8	37,6	13,5	10,7	8,9
99	6,9	2,4	7,2	3,9	4,1	12,5	10,2	10,4	6,7	7,7	11,7	8,9	7,3	8,9	11,2	8,7	7	39,2	3,1	2,5	26,5
101	2,6	3,6	11,6	4,2	3,9	16,9	14,6	14,8	9,7	12,1	14,7	13,2	11,7	4,6	13,2	12,7	11	27,3	4,3	6,9	18,5
102	2,3	3,6	9,9	5	5,2	14,5	12,9	14,1	10,6	10,4	15,6	12,6	10	4,3	14,9	12,6	10,9	28,2	4,3	7,6	17,9
104	19,3	15,9	25,4	14,5	14,3	30,6	28,4	28,6	24,3	25,9	29,5	26,4	25,5	21,3	28,1	26,3	24,6	21,2	16,6	11,2	5,4
105	7,5	5,4	7	6,8	7	12,3	10	10,2	10,1	7,5	15,2	8,9	7,1	9,5	11,7	12,2	10,5	38,2	6,1	4,7	19,2
107	4,6	1,3	9,5	2,8	3	14,7	12	12,7	8,9	10	13,5	12,3	9,6	6,6	14,9	10,9	9,2	28,9	2	6	14,5

110	13	16,4	22	17,8	17,9	28,6	25	15,2	20,1	22,1	23,6	24,1	22,1	15	26,2	22,1	20,4	41,3	17,1	17,9	21
114	12,6	16	21,6	17,4	17,9	26,8	23,6	24,7	19,7	22,9	23,3	24,9	21,7	14,6	26,5	21,7	20	42,5	16,7	18,4	21,9
117	10,8	16	21,7	17,5	17,5	26,8	24,7	23,8	20,5	22,2	25,2	24,2	21,8	12,8	26,5	22,5	20,8	40,4	16,7	18,9	20,7
120	6,4	11,6	17,3	13	13,2	22,6	20,3	20,6	16,1	17,8	21,1	18,2	17,4	8,4	23,9	18,1	16,4	37,9	12,9	13,9	16,5
124	22,3	18,9	28,4	17,5	17,2	33,5	31,4	31,6	26,7	28,9	29,9	29,4	28,5	24,3	32,1	28,7	27	8,1	19,4	24,6	11,2

69	71	72	73	76	78	85	87	88	90	91	94	95	97	99	101	102	104	105	107	110	114	117	120	124
18,7	7,6	18,1	20,3	17,7	14,4	7,1	22,7	21,7	8,5	10,8	32,6	3,1	17,8	6,9	2,6	2,3	19,3	7,5	4,6	13	12,6	10,8	6,4	22,3
15,3	6,6	14,8	17,3	14,4	11,8	6,2	19,3	18,3	4,4	16,3	29,2	7,5	14,8	2,4	3,6	3,6	15,9	5,4	1,3	16,4	16	16	11,6	18,9
24,8	8,2	24,2	16,8	23,8	21,3	7,8	28,8	27,8	6,9	21,6	38,7	13,3	23,9	7,2	11,6	9,9	25,4	7	9,5	22	21,6	21,7	17,3	28,4
13,9	8	13,3	15,9	12,9	10,4	7,6	17,9	16,9	5,8	4,9	27,8	8,9	13	3,9	4,2	5	14,5	6,8	2,8	17,8	17,4	17,5	13	17,5
13,7	8,3	13,1	15,7	12,7	10,1	7,8	17,6	16,6	6	14,7	27,6	9,2	12,8	4,1	3,9	5,2	14,3	7	3	17,9	17,9	17,5	13,2	17,2
30,2	11,1	27,1	32,3	39,2	26,6	13,2	33,3	32,3	12,2	26,9	43	18,6	28,2	12,5	16,9	14,5	30,6	12,3	14,7	28,6	26,8	26,8	22,6	33,5
27,8	11,3	27,3	29,8	26,8	24,3	10,8	31	30	9,9	24,6	40,7	16,3	26,9	10,2	14,6	12,9	28,4	10	12	25	23,6	24,7	20,3	31,4
25	11,4	27,4	20	26	24,5	11	34	31	11,1	24,8	41	15,5	27,1	10,4	14,8	14,1	28,6	10,2	12,7	15,2	24,7	23,8	20,6	31,6
27,2	7,4	25,7	27,5	23,7	25,8	5,6	30	31,6	4,7	21,1	31,9	11,3	20,5	6,7	9,7	10,6	24,3	10,1	8,9	20,1	19,7	20,5	16,1	26,7
25,3	8,7	24,7	17,3	24,3	21,8	8,6	29,2	28,1	7,6	22,1	38,2	13,8	24,4	7,7	12,1	10,4	25,9	7,5	10	22,1	22,9	22,2	17,8	28,9
28,2	8,8	26,2	28,2	24,8	26,8	10,7	34,1	35,2	7,9	25,1	32,9	24,3	25,1	11,7	14,7	15,6	29,5	15,2	13,5	23,6	23,3	25,2	21,1	29,9
26,1	9,2	25,1	25,8	25,6	24,5	7,6	32,3	33,1	6,7	26,2	33,9	14,6	26,2	8,9	13,2	12,6	26,4	8,9	12,3	24,1	24,9	24,2	18,2	29,4
24,9	8,3	24,3	16,9	23,9	21,4	7,9	28,9	27,8	6,9	22,7	38,8	13,4	24	7,3	11,7	10	25,5	7,1	9,6	22,1	21,7	21,8	17,4	28,5
20,7	9,6	20,1	22,3	19,7	16,4	9,1	24,7	23,7	10,5	12,8	34,6	5,1	19,8	8,9	4,6	4,3	21,3	9,5	6,6	15	14,6	12,8	8,4	24,3
30,3	12	27,9	26,2	27,9	26,6	9,4	34,2	35,8	8,6	29	36,1	16,2	27,9	11,2	13,2	14,9	28,1	11,7	14,9	26,2	26,5	26,5	23,9	32,1
28,7	9,6	27,7	29,4	25,7	27,8	7,6	32	33,6	6,7	22,1	33,9	13,3	22,5	8,7	12,7	12,6	26,3	12,2	10,9	22,1	21,7	22,5	18,1	28,7

27	8	26	27,7	24	26,1	5,9	30,3	31,9	5	20,4	32,2	11,6	10,8	7	11	10,9	24,6	10,5	9,2	20,4	20	20,8	16,4	27
16,5	32,1	20,4	23,2	21,2	38,2	37,2	8,9	77	37,9	34	11	40,2	37,6	39,2	27,3	28,2	21,2	38,2	28,9	41,3	42,5	40,4	37,9	8,1
16	7,7	15,5	18	15,1	12,3	6,9	20	19,3	5,1	17	28,1	8,2	13,5	3,1	4,3	4,3	16,6	6,1	2	17,1	16,7	16,7	12,9	19,4
21	2,5	16,2	16,7	4,7	5,2	4,7	25	23,9	3,2	7,1	34,9	7,4	10,7	2,5	6,9	7,6	11,2	4,7	6	17,9	18,4	18,9	13,9	24,6
4,5	24	3,8	4,2	5,1	5,9	24,9	17,3	17,9	23,6	10,5	15,9	23,2	8,9	26,5	18,5	17,9	5,4	19,2	14,5	21	21,9	20,7	16,5	11,2
0	24,6	4,6	4,8	5,9	6,5	25,5	4	3	24	11,6	13,9	23,9	9,8	27,6	16,7	18,9	6,7	20,9	15,8	22,4	22,8	21,5	17,3	3,6
24,6	0	15,8	15,5	4,9	5,7	0,65	24,7	23,4	4,5	7,9	33,6	8,1	11,8	4,4	7,3	5,2	12,5	1,2	6,6	12,9	18,6	18,5	14,5	25,5
4,6	15,8	0	5,4	6,5	7,1	26,1	4,6	3,6	24,6	12,2	14,5	24,5	10,4	18,2	17,3	19,4	7,3	21,5	16,4	23	23,4	22	17,9	4,2
4,8	15,5	5,4	0	8,5	9,1	28,1	6,6	5,6	25,6	14,2	16,5	26,5	12,4	20,2	19,3	21,4	9,3	23,5	18,4	25	25,4	24	19,9	6,4
5,9	4,9	6,5	8,5	0	6,1	15,1	3,6	2,6	22,6	21,2	13,5	23,5	9,4	17,2	16,3	18,4	6,3	20,5	15,4	22	22,4	21	16,9	3,4
6,5	5,7	7,1	9,1	6,1	0	4,8	3,9	2,8	21	20,7	12,7	21,9	8,9	16,5	15,9	19,2	5,8	20,9	14,9	22,7	22,9	2,9	15,9	4,3
25,5	0,65	26,1	28,1	15,1	4,8	0	24,2	22	1,8	7,4	32,9	7,8	11,1	4,1	7	4,6	11,9	0,8	6	13,6	13,6	17,9	13,9	24,9
4	24,7	4,6	6,6	3,6	3,9	24,2	0	1	38,1	24,9	9,9	30,1	20,1	37,7	20,7	20,9	11,6	35,9	28,9	40,7	40,4	41,2	30,1	0,4
3	23,4	3,6	5,6	2,6	2,8	22	1	0	10,9	24,1	9,1	30,8	20,6	37,1	19,7	19,9	10,6	34,9	27,9	39,7	39,4	40,2	29,1	1,1
24	4,5	24,6	25,6	22,6	21	1,8	38,1	10,9	0	15,1	13,2	17,1	4,6	18	17,7	17,4	4,3	15,8	19,9	26,9	26,5	25,9	20,1	10,1
11,6	7,9	12,2	14,2	21,2	20,7	7,4	24,9	24,1	15,1	0	28,2	12,6	10,1	18	16,6	16,9	11	16	16,9	11,7	11,9	12,1	7,1	22,6
13,9	33,6	14,5	16,5	13,5	12,7	32,9	9,9	9,1	13,2	28,2	0	28	27,1	32,7	30,6	30,9	12,5	28,9	23,7	31,6	31,9	32,1	30	9,5
23,9	8,1	24,5	26,5	23,5	21,9	7,8	30,1	30,8	17,1	12,6	28	0	13,1	16,3	13,3	13,9	16,1	13	12,8	9,7	9,9	4,8	7,8	29,6
9,8	11,8	10,4	12,4	9,4	8,9	11,1	20,1	20,6	4,6	10,1	27,1	13,1	0	18,4	17,5	19,6	7,5	21,7	16,6	23,2	23,6	22,2	18,1	4,4
27,6	4,4	18,2	20,2	17,2	16,5	4,1	37,7	37,1	18	18	32,7	16,3	18,4	0	5,5	4,3	15,2	3,4	3,9	16,4	16	14,9	10,1	41,1
16,7	7,3	17,3	19,3	16,3	15,9	7	20,7	19,7	17,7	16,6	30,6	13,3	17,5	5,5	0	3,4	10,2	11,7	12,5	14,2	14,7	13,9	12,1	20,3
18,9	5,2	19,4	21,4	18,4	19,2	4,6	20,9	19,9	17,4	16,9	30,9	13,9	19,6	4,3	3,4	0	7,3	5,1	2,2	13,6	13,2	16,9	11,9	10,3
6,7	12,5	7,3	9,3	6,3	5,8	11,9	11,6	10,6	4,3	11	12,5	16,1	7,5	15,2	10,2	7,3	0	10,1	6,8	15,4	15	14,4	11,9	12,7
20,9	1,2	21,5	23,5	20,5	20,9	0,8	35,9	34,9	15,8	16	28,9	13	21,7	3,4	11,7	5,1	10,1	0	5,1	14,2	13,8	15,9	11,2	43,1
15,8	6,6	16,4	18,4	15,4	14,9	6	28,9	27,9	19,9	16,9	23,7	12,8	16,6	3,9	12,5	2,2	6,8	5,1	0	15,1	14,7	10,1	7,1	40,2

22,4	12,9	23	25	22	22,7	13,6	40,7	39,7	26,9	11,7	31,6	9,7	23,2	16,4	14,2	13,6	15,4	14,2	15,1	0	0,4	9,4	4,1	39,6
22,8	18,6	23,4	25,4	22,4	22,9	13,6	40,4	39,4	26,5	11,9	31,9	9,9	23,6	16	14,7	13,2	15	13,8	14,7	0,4	0	9,9	4,5	40,1
21,5	18,5	22	24	21	2,9	17,9	41,2	40,2	25,9	12,1	32,1	4,8	22,2	14,9	13,9	16,9	14,4	15,9	10,1	9,4	9,9	0	10,5	35,3
17,3	14,5	17,9	19,9	16,9	15,9	13,9	30,1	29,1	20,1	7,1	30	7,8	18,1	10,1	12,1	11,9	11,9	11,2	7,1	4,1	4,5	10,5	0	26,2
3,6	25,5	4,2	6,4	3,4	4,3	24,9	0,4	1,1	10,1	22,6	9,5	29,6	4,4	41,1	20,3	10,3	10,3	43,1	40,2	39,6	40,1	35,3	26,2	0

Lampiran A2.

DATA PERMINTAAN GAS LPG 3 Kg (Jum'at, 14 April 2016)

Kode Toko/ Pangkalan	1	2	4	7	8	10	11	14	15	17	18	23	24	25	27	29	30	32	33	34	35	36	38
Jumlah Permintaan	20	4	7	8	3	18	22	9	13	20	10	8	5	10	17	5	9	3	5	15	7	5	39
41	42	43	50	66	70	81	82	89	92	94	98	99	101	104	107	118							
8	20	5	14	8	9	13	10	13	4	5	7	7	14	4	14	28							

DATA JARAK DARI DEPO KE SETIAP TOKO DAN ANTAR TOKO

Kode Toko/ Pangkalan	1 (depo)	2	3	5	8	9	11	12	15	16	18	19	24	25	26	28	30	31	33	34	35	36	37
1	0	5,9	7.5	13.3	5.4	5.2	18.6	16.2	16.4	5.1	9.1	14.0	4.9	14.5	4.8	14.3	13.3	2.1	5.4	4.5	7.3	5.1	5.8
2	5,9	0	2.9	8.8	1.7	2.6	14.1	11.8	12.0	2.7	4.5	9.5	2.3	10.1	2.4	10.9	8.8	4.1	1.6	2.7	4.0	1.9	3.2
3	7.5	2.9	0	7.7	4.4	4.4	13.0	10.7	10.8	4.5	1.7	8.4	4.7	9.0	4.8	9.8	7.7	5.6	4.3	5.1	4.3	4.6	3.8
5	13.3	8.8	7.7	0	11.2	12.1	5.3	3.0	3.2	12.2	6.0	0.75	11.8	1.3	11.9	5.2	0.06	10.4	11.1	12.2	12.0	11.4	11.4
8	5.4	1.7	4.4	11.2	0	1.0	15.8	13.4	13.6	1.1	6.0	11.2	0.65	11.8	0.75	12.6	10.5	4.4	0.082	1.05	3.1	0.382	1.6
9	5.2	2.6	4.4	12.1	1.0	0	16,8	14,4	14,6	0,1	6	12,2	0,35	12,7	0,36	13,5	11,5	4,2	1,1	0,35	2,1	1,4	0,6
11	18.6	14.1	13.0	5.3	15.8	16,8	0	3,5	2,4	17,6	11,4	5,1	17,1	4,3	17,2	8,9	5,3	15,8	16,4	17,1	17,4	16,7	16,8
12	16.2	11.8	10.7	3.0	13.4	14,4	3,5	0	2,8	15,2	9	2,8	14,8	2,5	14,9	5,6	2,9	13,4	14,1	14,8	15	14,4	14,4
15	16.4	12.0	10.8	3.2	13.6	14,6	2,4	2,8	0	15,3	9,2	3	15,3	2,2	15,4	8,1	3,1	13,6	14,3	15,3	15,2	14,6	14,6
16	5.1	2.7	4.5	12.2	1.1	0,1	17,6	15,2	15,3	0	6,1	12,2	0,45	12,8	0,46	13,6	11,6	4,3	1,2	0,45	2,2	1,5	0,7
18	9.1	4.5	1.7	6.0	6.0	6	11,4	9	9,2	6,1	0	6,8	6,4	7,3	6,5	6,3	6,1	2,5	5,9	6,8	6	6,2	5,4
19	14.0	9.5	8.4	0.75	11.2	12,2	5,1	2,8	3	12,2	6,8	0	12,6	1,1	12,7	6	0,7	5,2	11,9	13	12,8	12,2	12,2
24	4.9	2.3	4.7	11.8	0.65	0,35	17,1	14,8	15,3	0,45	6,4	12,6	0	12,4	0,1	13,2	11,1	3,9	0,7	0,4	2,5	1	0,95
25	14.5	10.1	9.0	1.3	11.8	12,7	4,3	2,5	2,2	12,8	7,3	1,1	12,4	0	12,5	6,6	1,3	11,7	12,4	12,8	13,3	12,7	12,8
26	4.8	2.4	4.8	11.9	0.75	0,36	17,2	14,9	15,4	0,46	6,5	12,7	0,1	12,5	0	13,3	11,2	4	0,8	0,3	2,6	1,1	1,05
28	14.3	10.9	9.8	5.2	12.6	13,5	8,9	5,6	8,1	13,6	6,3	6	13,2	6,6	13,3	0	5,3	11,1	12,4	13,6	13,3	12,7	14,1
30	13.3	8.8	7.7	0.06	10.5	11,5	5,3	2,9	3,1	11,6	6,1	0,7	11,1	1,3	11,2	5,3	0	10,4	11,1	11,5	12,1	11,4	12,8
31	2.1	4.1	5.6	10.4	4.4	4,2	15,8	13,4	13,6	4,3	2,5	5,2	3,9	11,7	4	11,1	10,4	0	4,5	4,3	6,4	4,8	4,9
33	5.4	1.6	4.3	11.1	0.082	1,1	16,4	14,1	14,3	1,2	5,9	11,9	0,7	12,4	0,8	12,4	11,1	4,5	0	1,1	2,9	0,3	1,7
34	4.5	2.7	5.1	12.2	1.05	0,35	17,1	14,8	15,3	0,45	6,8	13	0,4	12,8	0,3	13,6	11,5	4,3	1,1	0	2,9	1,4	1,35

35	7.3	4.0	4.3	12.0	3.1	2,1	17,4	15	15,2	2,2	6	12,8	2,5	13,3	2,6	13,3	12,1	6,4	2,9	2,9	0	3,1	1,5
36	5.1	1.9	4.6	11.4	0.382	1,4	16,7	14,4	14,6	1,5	6,2	12,2	1	12,7	1,1	12,7	11,4	4,8	0,3	1,4	3,1	0	2
37	5.8	3.2	3.8	11.4	1.6	0,6	16,8	14,4	14,6	0,7	5,4	12,2	0,95	12,8	1,05	14,1	12,8	4,9	1,7	1,35	1,5	2	0
39	17.0	13.5	12.4	5.1	15.2	16,2	6,2	2,9	5,4	16,3	26,2	4,9	15,9	6,8	16	3,1	5,1	19,1	15,1	16,3	16,8	15,4	16,2
42	5.2	3.0	5.1	12.5	1.05	0,75	17,9	15,5	15,7	0,85	6,7	13,1	0,35	13,6	0,45	13,7	12,4	4,4	1,3	0,75	2,7	1,6	6,5
43	7.9	4.6	4.9	12.6	4.2	2,7	17,9	15,6	15,7	2,8	6,5	13,3	3	13,9	3,1	14	12,6	6,9	3,4	3,5	2,2	3,7	6,3
44	7.1	12.3	13.9	17.9	13.8	13,7	23,3	20,9	21,1	13,8	15,7	18,7	13,3	19,2	13,4	19,3	18	9,2	10,2	13,7	12,7	10,3	9,9
51	9.2	4.7	3.6	4.1	6.4	7,4	9,5	7,1	7,3	7,5	2,9	4,9	8	5,5	8,1	5,5	4,2	8,3	7,2	8,4	8,9	8,7	7,4
67	6.0	1.6	3.2	9.1	3.3	4,2	12,8	11,2	10,6	4,2	3,4	9,9	3,9	10,4	4	10,5	9,2	9,5	3,2	4,3	8,6	3,5	7,9
71	7.6	6.6	7.7	8.2	8.3	9,3	11,1	11,3	11,4	9,4	7,4	8,7	8,9	9,6	9	9,2	8,3	9,6	8,2	9,3	10,6	8,5	8,1
82	3.3	9.1	10.7	14.8	8.6	8,5	20,1	17,8	17,9	8,6	12,6	15,5	8,1	16,1	8,2	16,1	15,8	5,4	9,2	8,5	10,4	8,8	8,6
83	8.8	5.6	5.1	5.7	7.9	8,2	11	8,7	8,8	8,3	3,5	6,4	7,9	7	8	7,1	8,9	6,8	7,8	8,3	9,8	8,6	8,2
90	32,6	4.4	4.8	6.9	6.0	7,1	12,2	9,9	11,1	7,2	4,7	7,6	6,5	8,2	6,6	6,7	6,9	10,5	8,4	6,9	9,1	7,2	7,6
93	3.3	7.7	9.2	13.3	9.3	9	18,7	16,3	16,5	9,1	11,1	14,1	8,7	14,6	8,8	14,7	9,1	4,5	9,3	9,1	8,9	9,4	9,2
95	3.1	7.5	9.1	13.3	9.2	8,9	18,6	16,3	15,5	9	11,3	13,8	8,6	14,5	8,7	14,6	13,4	5,1	9,5	9	8,7	9,3	9
99	6.9	2.4	3.9	7.2	4.1	5,1	12,5	10,2	10,4	5,2	6,7	7,7	11,3	8,5	4,9	8,9	7,3	8,9	8,2	5,2	8,8	5,5	5,2
100	0.9	5.9	8.4	13.7	6.3	6,1	19	16,7	16,9	6,2	11,2	14	5,8	15	5,9	15	15,1	4,7	7,9	6,2	5,2	6,5	6,2
102	2.3	3.6	5.1	9.9	5.2	4,9	14,5	12,9	14,1	5	10,6	10,4	4,6	11,3	4,7	12,6	10	4,3	7,5	5	5,6	5,3	5
105	7.5	5.4	6.5	7.0	7.0	8	12,3	10	10,2	8,1	10,1	7,5	7,7	8,3	7,8	8,9	7,1	9,5	10,4	8,1	9,3	8,4	8,2
108	8.6	4.8	5.9	6.5	6.5	7,5	11,8	9,5	19,6	7,6	4,3	7,2	7,1	7,8	7,2	6,9	14,7	5,7	9,1	7,5	9,7	7,8	7,6
119	4.5	5.6	7.2	11.3	7.2	7	16,6	14,3	14,4	7,1	9,1	12	6,7	12,6	6,8	12,7	15,4	5,6	9,4	12,9	6,2	13,2	13

39 42 43 44 51 67 71 82 83 90 93 95 99 100 102 105 108 119

17.0	5.2	7.9	7.1	9.2	6.0	7.6	3.3	8.8	8.5	3.3	3.1	6.9	0.9	2.3	7.5	8.6	4.5
13.5	3.0	4.6	12.3	4.7	1.6	6.6	9.1	5.6	4.4	7.7	7.5	2.4	5.9	3.6	5.4	4.8	5.6
12.4	5.1	4.9	13.9	3.6	3.2	7.7	10.7	5.1	4.8	9.2	9.1	3.9	8.4	5.1	6.5	5.9	7.2
5.1	12.5	12.6	17.9	4.1	9.1	8.2	14.8	5.7	6.9	13.3	13.3	7.2	13.7	9.9	7.0	6.5	11.3
15.2	1.05	4.2	13.8	6.4	3.3	8.3	8.6	7.9	6.0	9.3	9.2	4.1	6.3	5.2	7.0	6.5	7.2
16.2	0,75	2,7	13,7	7,4	4,2	9,3	8,5	8,2	7,1	9	8,9	5,1	6,1	4,9	8	7,5	7
6,2	17,9	17,9	23,3	9,5	12,8	11,1	20,1	11	12,2	18,7	18,6	12,5	19	14,5	12,3	11,8	16,6
2,9	15,5	15,6	20,9	7,1	11,2	11,3	17,8	8,7	9,9	16,3	16,3	10,2	16,7	12,9	10	9,5	14,3
5,4	15,7	15,7	21,1	7,3	10,6	11,4	17,9	8,8	11,1	16,5	15,5	10,4	16,9	14,1	10,2	19,6	14,4
16,3	0,85	2,8	13,8	7,5	4,2	9,4	8,6	8,3	7,2	9,1	9	5,2	6,2	5	8,1	7,6	7,1
26,2	6,7	6,5	15,7	2,9	3,4	7,4	12,6	3,5	4,7	11,1	11,3	6,7	11,2	10,6	10,1	4,3	9,1
4,9	13,1	13,3	18,7	4,9	9,9	8,7	15,5	6,4	7,6	14,1	13,8	7,7	14	10,4	7,5	7,2	12
15,9	0,35	3	13,3	8	3,9	8,9	8,1	7,9	6,5	8,7	8,6	11,3	5,8	4,6	7,7	7,1	6,7
6,8	13,6	13,9	19,2	5,5	10,4	9,6	16,1	7	8,2	14,6	14,5	8,5	15	11,3	8,3	7,8	12,6
16	0,45	3,1	13,4	8,1	4	9	8,2	8	6,6	8,8	8,7	4,9	5,9	4,7	7,8	7,2	6,8
3,1	13,7	14	19,3	5,5	10,5	9,2	16,1	7,1	6,7	14,7	14,6	8,9	15	12,6	8,9	6,9	12,7
5,1	12,4	12,6	18	4,2	9,2	8,3	15,8	8,9	6,9	9,1	13,4	7,3	15,1	10	7,1	14,7	15,4
19,1	4,4	6,9	9,2	8,3	9,5	9,6	5,4	6,8	10,5	4,5	5,1	8,9	4,7	4,3	9,5	5,7	5,6
15,1	1,3	3,4	10,2	7,2	3,2	8,2	9,2	7,8	8,4	9,3	9,5	8,2	7,9	7,5	10,4	9,1	9,4
16,3	0,75	3,5	13,7	8,4	4,3	9,3	8,5	8,3	6,9	9,1	9	5,2	6,2	5	8,1	7,5	12,9
16,8	2,7	2,2	12,7	8,9	8,6	10,6	10,4	9,8	9,1	8,9	8,7	8,8	5,2	5,6	9,3	9,7	6,2
15,4	1,6	3,7	10,3	8,7	3,5	8,5	8,8	8,6	7,2	9,4	9,3	5,5	6,5	5,3	8,4	7,8	13,2
16,2	6,5	6,3	9,9	7,4	7,9	8,1	8,6	8,2	7,6	9,2	9	5,2	6,2	5	8,2	7,6	13
0	16,3	16,6	18,9	7,5	10,9	12	16,5	7,6	8,6	14,6	16,2	11,2	15,6	14,9	11,7	7,5	13
16,3	0	3,6	11,2	9,2	5,7	10,6	8,4	8,2	6,7	8,9	8,8	11,5	6	4,9	8	7,4	7

16,6	3,6	0	11,5	9,5	7,8	11,2	8,5	8,3	6,9	8,7	8,9	11	6,3	5,2	8,3	7,2	7,5
18,9	11,2	11,5	0	14,8	11,2	13,1	11,2	13,4	10,5	12,4	15,7	11,4	10,8	12,9	13,6	10,5	9,8
7,5	9,2	9,5	14,8	0	5,2	5,1	6,5	8,4	2,7	9,2	6,7	8,9	8,6	8,8	10,1	10	10,4
10,9	5,7	7,8	11,2	5,2	0	5	8,5	5,5	10,1	6,9	10,5	0,65	7,3	3,5	3,8	10,2	4,8
12	10,6	11,2	13,1	5,1	5	0	12,7	4,5	2,2	8	8,1	4,4	8,9	5,2	1,2	10,1	5,9
16,5	8,4	8,5	11,2	6,5	8,5	12,7	0	11,9	12,4	13,2	13,5	17,1	14,1	11,3	10,6	12,1	11,4
7,6	8,2	8,3	13,4	8,4	5,5	4,5	11,9	0	3,2	5,8	7,9	7,9	9,5	9,7	4,3	4,7	8,8
8,6	6,7	6,9	10,5	2,7	10,1	2,2	12,4	3,2	0	7,9	17,1	18	9,8	17,4	15,8	10,1	9,8
14,6	8,9	8,7	12,4	9,2	6,9	8	13,2	5,8	7,9	0	7,9	7,9	2,4	4,8	8,3	9,8	2,5
16,2	8,8	8,9	15,7	6,7	10,5	8,1	13,5	7,9	17,1	7,9	0	16,3	3,5	13,9	13	9,7	3,9
11,2	11,5	11	11,4	8,9	0,65	4,4	17,1	7,9	18	7,9	16,3	0	8,1	4,3	3,4	8,3	5,7
15,6	6	6,3	10,8	8,6	7,3	8,9	14,1	9,5	9,8	2,4	3,5	8,1	0	3,3	8,8	9,7	3,5
14,9	4,9	5,2	12,9	8,8	3,5	5,2	11,3	9,7	17,4	4,8	13,9	4,3	3,3	0	5,1	9,8	2,8
11,7	8	8,3	13,6	10,1	3,8	1,2	10,6	4,3	15,8	8,3	13	3,4	8,8	5,1	0	5,4	5,9
7,5	7,4	7,2	10,5	10	10,2	10,1	12,1	4,7	10,1	9,8	9,7	8,3	9,7	9,8	5,4	0	7,1
13	7	7,5	9,8	10,4	4,8	5,9	11,4	8,8	9,8	2,5	3,9	5,7	3,5	2,8	5,9	7,1	0

LAMPIRAN B

Data perhitungan total jarak, kapasitas kendaraan dan total waktu

Pangkalan	K1			K2			K3					
	Jarak (Km)		Kapasitas (Tabung)	Waktu(jam)	Jarak (Km)		Kapasitas (Tabung)	Waktu (jam)	Jarak (Km)		Kapasitas (Tabung)	Waktu
	Start	End			Start	End			Start	End		
Pangkalan 1									14	19,1	17	0,8057

Pangkalan 2				12,4	13,1	17	0,941				
Pangkalan 3								0	14	11	0,585
Pangkalan 4	0	14,5	3	0,5883							
Pangkalan 5				1,1	12,4	6	0,513				
Pangkalan 6								19,1	36,1	32	1,511
Pangkalan 7				0	1,1	15	0,077				
Pangkalan 8								36,1	63,6	200	2,961
Pangkalan 9	14,5	46,6	200	2,281							
Jumlah		46,6+19,9 = 66,5	203	2,281+0,796 = 3,077	13,1+16,2 = 29,3	38	0,941+0,648 = 1,589	63,6+12,6 = 76,2	260	2,961+0,504 = 3,465	

LAMPIRAN C. Hasil Percobaan setiap rute dengan Dua Data Berbeda

C1. Hasil Percobaan dengan data tanggal 5 April 2016

Jumlah Swarm	Jumlah Iterasi	Rute Kendaraan ke -	Total Jarak (Km)	Total Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
10	100	Rute K1 = Depo - Toko ke 90 - 119 - 30 - 7 - 106 - 1 - 77 - 109 - 93 - 27 - 11 - 18 - Depo	K1 = 157,4	K1 = 159	K1 = 6 Jam 39 Menit 45,6 Detik	92	5,9128
		Rute K2 = Depo - Toko ke 44-10-4-75-71-72-94-17-26-87-104 - Depo	K2 = 177,6	K2 = 161	K2 = 7 Jam 28 Menit 14,4 Detik		
		Rute K3 = Depo - Toko ke 6-96-89-103-54-98-14-29-38-49-86-68 - Depo	K3 = 128,26	K3 = 116	K3 = 5 Jam 24 Menit 19,44 Detik		
		Rute K4 = Depo - Toko ke 100-40-116-113-84-123-56-52-70-101 - Depo	K4 = 128,5	K4 = 135	K4 = 5 Jam 26 Menit 54 Detik		
500	500	Rute K1 = Depo - Toko ke 54-96-52-4-44-49-10-87-103-90-100 - Depo	K1 = 177,2	K1 = 151	K1 = 7 Jam 25 Menit 46,8 Detik	299	17,9818
		Rute K2 = Depo - Toko ke 38-11-29-72-77-93-106-71-123-14-109-116-94 - Depo	K2 = 169,9	K2 = 128	K2 = 7 Jam 6 Menit 15,6 Detik		
		Rute K3 = Depo - Toko ke 119-84-101-18-27-56-75-86-68-113 - Depo	K3 = 113,6	K3 = 131	K3 = 4 Jam 50 Menit 38,4 Detik		
		Rute K4 = Depo - Toko ke 17-30-89-104-98-7-6-26-1-70-40 - Depo	K4 = 84,26	K4 = 161	K4 = 3 Jam 44 Menit 13,44 Detik		
1000	1000	Rute K1 = Depo - Toko ke 100-4-98-119-75-30-94-44-40 - Depo	K1 = 98,4	K1 = 157	K1 = 4 Jam 17 Menit 9,6 Detik	356	36,1643
		Rute K2 = Depo - Toko ke 109-104-106-17-84-70-10-38-56-68-101-11-29-113 - Depo	K2 = 163,95	K2 = 165	K2 = 6 Jam 55 Menit 58,8 Detik		

		Rute K3 = Depo - Toko ke52-7-96-89-123-54-90-116-103-93-77-1-6 - Depo	K3 = 137,4	K3 = 164	K3 = 5 Jam 52 Menit 15,6 Detik		
		Rute K4 = Depo - Toko ke 27-18-26-71-49-86-87-72-14Depo	K4 = 127,6	K4 = 85	K4 = 5 Jam 19 Menit 14,4 Detik		
20	100	Rute K1 = Depo - Toko ke 54 - 44-30-116-68-75-96-89-11-27 - Depo	K1 = 112,3	K1 = 138	K1 = 4 Jam 48 Menit 1,2 Detik	32	10,8121
		Rute K2 = Depo - Toko ke 52-6-26-17-29-106-90-94-109-84-113-10Depo	K2 = 143,1	K2 = 148	K2 = 6 Jam 3 Menit 56,4 Detik		
		Rute K3 = Depo - Toko ke 56-71-123-49-4-18-14-101-1-98-119 - Depo	K3 = 125,9	K3 = 158	K3 = 5 Jam 24 Menit 9,6 Detik		
		Rute K4 = Depo - Toko ke 40-93-87-86-77-38-7-72-104-70 - Depo	K4 = 149,2	K4 = 127	K4 = 6 Jam 16 Menit 4,8 Detik		
	500	Rute K1 = Depo - Toko ke 49-123-68-71-93-89-4-40-7-113- Depo	K1 = 121,9	K1 = 136	K1 = 5 Jam 11 Menit 33,6 Detik	46	34,9506
		Rute K2 = Depo - Toko ke 38-103-77-1-104-109-94-18-11-119 - Depo	K2 = 135,3	K2 = 122	K2 = 5 Jam 41 Menit 43,2 Detik		
		Rute K3 = Depo - Toko ke 6-72-86-87-75-96-70-26-90-54-52Depo	K3 = 105,3	K3 = 146	K3 = 4 Jam 32 Menit 43,2 Detik		
		Rute K4 = Depo - Toko ke 84-27-29-98-101-30-106-116-100-44-14-10-17-56 - Depo	K4 = 138,3	K4 = 167	K4 = 5 Jam 54 Menit 55,2 Detik		
	1000	Rute K1 = Depo - Toko ke 100-30-29-40-18-70-26-119-113-116-11-54-4-27-17 - Depo	K1 = 136,8	K1 = 175	K1 = 5 Jam 52 Menit 49,2 Detik	357	75,6274
		Rute K2 = Depo - Toko ke 87-75-109-84-98-6-7-93-103-71-101 - Depo	K2 = 157,06	K2 = 134	K2 = 6 Jam 35 Menit 56,64 Detik		
		Rute K3 = Depo - Toko ke10-86-123-1-96-89-44 -	K3 = 106,8	K3 = 124	K3 = 4 Jam 32 Menit 49,2		

		Depo			Detik		
		Rute K4 = Depo - Toko ke 68-72-56-90-94-52-106-14-77-38 - Depo	K4 = 141,8	K4 = 138	K4 = 5 Jam 59 Menit 19,2 Detik		
30	100	Rute K1 = Depo - Toko ke 10-101-7-87-72-52-44-100-109-77-6-4-30 - Depo	K1 = 161	K1 = 147	K1 = 6 Jam 47 Menit 24 Detik	42	11,5162
		Rute K2 = Depo - Toko ke 1-70-14-68-93-86-49-123-7 - Depo	K2 = 112	K2 = 123	K2 = 4 Jam 45 Menit 48 Detik		
		Rute K3 = Depo - Toko ke 94-116-103-96-75-56-113-90-84-17-26Depo	K3 = 102,1	K3 = 140	K3 = 4 Jam 24 Menit 32,4 Detik		
		Rute K4 = Depo - Toko ke 106-89-18-40-11-38-27-119-98-104-29-54 - Depo	K4 = 98,7	K4 = 161	K4 = 4 Jam 18 Menit 22,8 Detik		
	500	Rute K1 = Depo - Toko ke 11-14-40-18-101-100-70-27-38-89-116- Depo	K1 = 104,7	K1 = 140	K1 = 4 Jam 30 Menit 46,8 Detik	218	53,2096
		Rute K2 = Depo - Toko ke 98-17-26-6-104-10-29-119 - Depo	K2 = 69,9	K2 = 92	K2 = 3 Jam 0 Menit 15,6 Detik		
		Rute K3 = Depo - Toko ke 72-86-49-56-77-113-109-123-68-7-1-54-44-52Depo	K3 = 168,8	K3 = 182	K3 = 7 Jam 10 Menit 37,2 Detik		
		Rute K4 = Depo - Toko ke 94-84-96-75-87-103-90-30-71-93-106-4 - Depo	K4 = 149,5	K4 = 157	K4 = 6 Jam 20 Menit 18 Detik		
	1000	Rute K1 = Depo - Toko ke 90 - 119 - 30 - 7 - 106 - 1 - 77 - 109 - 93 - 27 - 11 - 18 - Depo	K1 = 105,6	K1 = 105	K1 = 4 Jam 28 Menit 26,4 Detik	23	105,637
		Rute K2 = Depo - Toko ke 44-10-4-75-71-72-94-17-26-87-104 - Depo	K2 = 129,9	K2 = 160	K2 = 5 Jam 33 Menit 45,6 Detik		
		Rute K3 = Depo - Toko ke 6-96-89-103-54-98-14-29-38-49-86-68 - Depo	K3 = 165,7	K3 = 144	K3 = 6 Jam 57 Menit 40,8 Detik		

Rute K4 = Depo - Toko ke 100-40-116-113-84-123-56-52-70-101 - Depo

K4 = 119,06

K4 = 162

K4 = 5 Jam 7 Menit 44,64 Detik

C2. Hasil Percobaan dengan data tanggal 14 April 2016

Jumlah Swarm	Jumlah Iterasi	Rute Kendaraan ke -	Total Jarak (Km)	Total Kapasitas (Tabung)	Total Waktu	Iterasi Konvergen	Running Time (detik)
10	100	Rute K1 = Depo - Toko ke 50-99-92-81-43-11-10-29-33 - Depo	K1 = 90,3	K1 = 93	K1 = 3 Jam 50 Menit 43,2 Detik	13	5,3
		Rute K2 = Depo - Toko ke 104-15-18-14-38-4-89 - Depo	K2 = 56,7	K2 = 95	K2 = 2 Jam 29 Menit 34,8 Detik		
		Rute K3 = Depo - Toko ke 101-66-17-107-98-1-82-7-2-118-94-8-41-25 - Depo	K3 = 68,1	K3 = 159	K3 = 3 Jam 4 Menit 56,4 Detik		
		Rute K4 = Depo - Toko ke 30-70-27-24-36-34-23-42-32-35 - Depo	K4 = 56,1	K4 = 98	K4 = 2 Jam 29 Menit 8,4 Detik		
500	500	Rute K1 = Depo - Toko ke 99-118-25-89-4-8-17-36-81-30 - Depo	K1 = 64,3	K1 = 115	K1 = 2 Jam 50 Menit 49,2 Detik	242	16
		Rute K2 = Depo - Toko ke 15-42-7-23-70-43-27-50-29-10-66-101 - Depo	K2 = 87,65	K2 = 139	K2 = 3 Jam 50 Menit 21,6 Detik		
		Rute K3 = Depo - Toko ke 92-104-33-32-41-1-94Depo	K3 = 35,7	K3 = 49	K3 = 1 Jam 32 Menit 40,8 Detik		
		Rute K4 = Depo - Toko ke 34-82-38-107-2-35-14-11-24-18-98 - Depo	K4 = 78,3	K4 = 142	K4 = 3 Jam 27 Menit 55,2 Detik		
1000	1000	Rute K1 = Depo - Toko ke 34-42-94-30-36-17-107-92-104-81 - Depo	K1 = 70,1	K1 = 109	K1 = 3 Jam 3 Menit 44,4 Detik	34	31

		Rute K2 = Depo - Toko ke 1-50-29-35-23-2-25-7-15 - Depo	K2 = 43,65	K2 = 89	K2 = 1 Jam 57 Menit 15,6 Detik		
		Rute K3 = Depo - Toko ke 98-18-38-14-101-10-66-99 - Depo	K3 = 74,5	K3 = 112	K3 = 3 Jam 14 Menit 18 Detik		
		Rute K4 = Depo - Toko ke 118-43-89-70-32-33-8-8-41-82-4-11-27-24 - Depo	K4 = 81	K4 = 135	K4 = 3 Jam 34 Menit 24 Detik		
20	100	Rute K1 = Depo -43-18-8-33-34-30-17-27-29-11-82-94-36-98 - Depo	K1 = 102,35	K1 = 138	K1 = 4 Jam 26 Menit 38,4 Detik	32	11
		Rute K2 = Depo - Toko ke 89-7-25-81-50-4-1-66-2-23 - Depo	K2 = 57,25	K2 = 105	K2 = 2 Jam 31 Menit 54 Detik		
		Rute K3 = Depo - Toko ke 101-70-32-35-41-104-107-92 - Depo	K3 = 44,1	K3 = 63	K3 = 1 Jam 54 Menit 50,4 Detik		
		Rute K4 = Depo - Toko ke 99-38-24-14-10-118-42-15Depo	K4 = 59,9	K4 = 139	K4 = 2 Jam 42 Menit 45,6 Detik		
	500	Rute K1 = Depo - Toko ke 15-2107-38-50-24-10-101-118-17-99- Depo	K1 = 78,8	K1 = 176	K1 = 3 Jam 33 Menit 7,2 Detik	39	30
		Rute K2 = Depo - Toko ke 35-18-82-94-8-33-66-1-23-30 - Depo	K2 = 55,05	K2 = 85	K2 = 2 Jam 24 Menit 37,2 Detik		
		Rute K3 = Depo - Toko ke42-104-11-14-4-29-89-70-98 - Depo	K3 = 52,66	K3 = 96	K3 = 2 Jam 20 Menit 23,04 Detik		
		Rute K4 = Depo - Toko ke 92-27-81-43-41-25-32-7-34-36 - Depo	K4 = 68,232	K4 = 88	K4 = 2 Jam 56 Menit 45,408 Detik		
	1000	Rute K1 = Depo - Toko ke 43-70-82-17-24-89-50-81 - Depo	K1 = 56,2	K1 = 89	K1 = 2 Jam 28 Menit 22,8 Detik	347	61
		Rute K2 = Depo - Toko ke 30-42-23-8-66-15-33-	K2 = 53,11	K2 = 103	K2 = 2 Jam 22 Menit 27,84 Detik		

		7-4-29-27 - Depo						
		Rute K3 = Depo - Toko ke 32-41-25-94-92-118-2-1-107-35 - Depo	K3 = 54,05	K3 = 103	K3 = 2 Jam 23 Menit 43,2 Detik			
		Rute K4 = Depo - Toko ke 38-10-14-18-11-99-98-104-34-36-101 - Depo	K4 = 77,7	K4 = 150	K4 = 3 Jam 27 Menit 28,8 Detik			
30	100	Rute K1 = Depo - Toko ke 30-32-118-94-81-42-23-8-1-4-14-104 - Depo	K1 = 77,55	K1 = 129	K1 = 3 Jam 24 Menit 7,2 Detik	16		16
		Rute K2 = Depo - Toko ke 7-70-25-36-35-50-17-34 - Depo	K2 = 50,65	K2 = 88	K2 = 2 Jam 14 Menit 3,6 Detik			
		Rute K3 = Depo - Toko ke 101-15-33-41-10-11-38-82-92-66-98 - Depo	K3 = 60,65	K3 = 148	K3 = 2 Jam 46 Menit 3,6 Detik			
		Rute K4 = Depo - Toko ke 99-43-107-24-18-29-27-89-2 - Depo	K4 = 56,1	K4 = 80	K4 = 2 Jam 27 Menit 8,4 Detik			
	500	Rute K1 = Depo - Toko ke 30-101-1-98-104-23-17-92-70-50-81 - Depo	K1 = 63,9	K1 = 122	K1 = 2 Jam 50 Menit 21,6 Detik	13		47
		Rute K2 = Depo - Toko ke 33-34-10-4-27-66-36-15 - Depo	K2 = 59,5	K2 = 88	K2 = 2 Jam 35 Menit 48 Detik			
		Rute K3 = Depo - Toko ke 89-2-8-25-35-32-41-42-7-14-118-94 -Depo	K3 = 63,56	K3 = 118	K3 = 2 Jam 49 Menit 2,64 Detik			
		Rute K4 = Depo - Toko ke 99-38-82-18-24-29-11-107-43 - Depo	K4 = 62,9	K4 = 117	K4 = 2 Jam 47 Menit 57,6 Detik			
	1000	Rute K1 = Depo - Toko ke 17-82-104-24-10-14-94-41-25-66-92 - Depo	K1 = 70,85	K1 = 101	K1 = 3 Jam 4 Menit 32,4 Detik	382		94
		Rute K2 = Depo - Toko ke 99-36-81-35-7-42 - Depo	K2 = 36,982	K2 = 60	K2 = 1 Jam 37 Menit 15,408 Detik			

Rute K3 = Depo - Toko ke 101-8-33-23-34-32-30-
4-89-1-98-2-29-11-50 - Depo

K3 = 72,75

K3 = 149

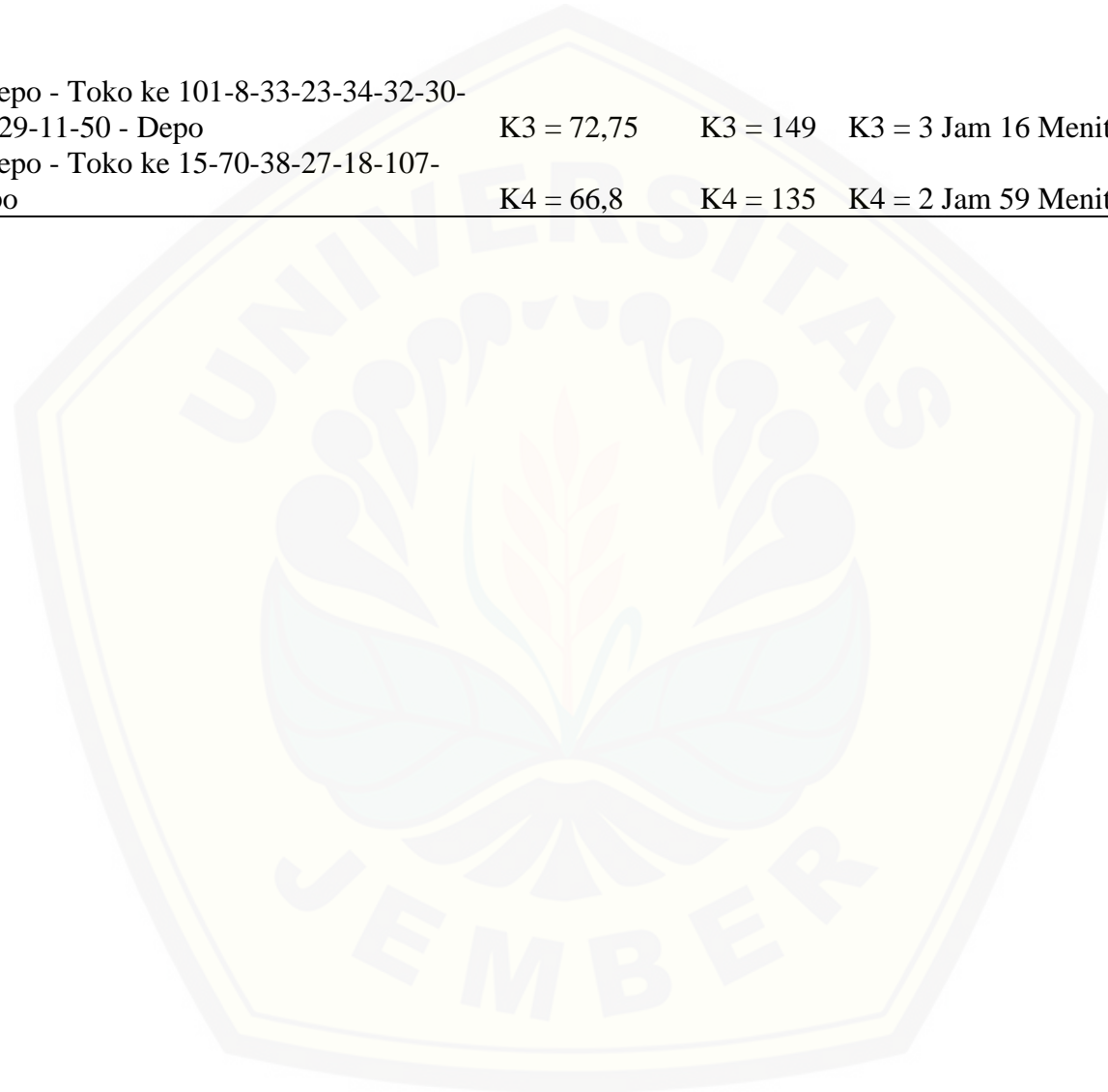
K3 = 3 Jam 16 Menit 6 Detik

Rute K4 = Depo - Toko ke 15-70-38-27-18-107-
118-43 - Depo

K4 = 66,8

K4 = 135

K4 = 2 Jam 59 Menit 19,2 Detik



LAMPIRAN D. Hasil Penelitian di PT. Amita Bara Sejahtera

D1. Data Jumlah Permintaan Barang Setiap Pangkalan PT. Amita Bara Sejahtera Jember perHari dalam 1 Bulan April 2016 (Satuan perTabung)

Tanggal	Nama Pangkalan ke-																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	17	17	0	0	0	0	0	0	11	0	0	0	0	0	3
2	15	0	0	0	0	6	10	4	0	0	0	0	0	0	9	0	11	0	0	6	0	0	10	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	0	0	0	18
5	32	0	0	11	0	4	5	0	0	10	7	0	0	9	0	0	18	10	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	21	7	0	0	0	0	12	0	0	0	0	0	19	0	13	0	13	0	33	8	14	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	6	0	11	0	0	0	0	0	20	13	0	0	15	0	0	19	11	0	0	0	5	0	7
12	25	6	0	0	8	4	15	5	0	0	0	0	0	0	13	0	15	0	0	8	0	0	4	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	20	4	0	7	0	0	8	3	0	18	22	0	0	9	13	0	20	10	0	0	0	0	8	5
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	20	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	9	0	7	0	0	0	0	8	15	12	0	32	7	0	0	13	5	0	12	0	0	0	16
19	25	0	0	0	0	4	10	5	0	0	0	5	0	0	11	0	0	0	0	0	7	0	0	0

0	15	0	0	13	0	0	0	0	0	0	5	0	0	0	0	0	29	19	13	34	13	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5	0	12	0	0	0	0	0	6	6	0	29	0	0	0	11	0	0	0	0	10
18	0	138	19	0	0	0	0	0	0	4	3	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0
0	14	0	0	4	0	0	14	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	17	14	0
0	4	0	0	7	0	0	0	26	0	3	0	9	0	13	0	0	15	19	12	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0
8	0	0	21	0	11	17	0	0	7	0	0	0	12	0	0	10	0	25	15	0	0	0	0	10
0	15	0	0	10	0	0	0	0	0	0	10	0	0	0	0	0	15	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	15	0	0	6	15	0
0	12	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	15	28	0	0	0	0	0	10
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0
0	0	0	0	0	0	0	0	0	0	5	0	0	0	23	0	0	0	0	9	0	0	0	0	0
0	14	0	0	12	0	13	0	0	5	0	14	0	10	0	15	0	29	29	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	9	0	0	12	83	0	0	0	7
0	0	134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	11	0	0	0
0	11	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	14	21	13	0	0	11	14	0

LAMPIRAN D1. Data Pangkalan PT. Amita Bara Sejahtera Jember dan Alamat setiap pangkalan

NO	NAMA PANGKALAN	ALAMAT	KELURAHAN	KECAMATAN
1	Ampera	JL. LETJEN DI PANJAITAN RT/RW 004/023	SUMBERSARI	SUMBERSARI
2	Aris	JL. TEUKU UMAR	KEPATIHAN	
3	Baru tk (Sifong)	JL. TEUKU UMAR	KEPATIHAN	
4	Bulan Bintang Jubung	DUSUN KRAJAN RT/RW 005/00 JUBUNG	KRAJAN	
5	Cahaya Baru	JL. PANJAITAN	KEBONSARI	SUMBERSARI
6	Dani Sukorejo	JL. LETJEN S. PARMAN	KARANGREJO	SUMBERSARI
7	Hadi Sutarto (Sukorejo)	JL. S. PARMAN RT/RW 001/001	KARANGREJO	SUMBERSARI
8	Harto	JL. KAPTEN TENDEAN	SUMBERSARI	SUMBER SARI
9	Hasan Basuki Rahmat	JL. BASUKI RAHMAT	KEBONSARI	
10	Imam Rowotamtu	DUSUN KRAJAN RT/RW 002/004	RAMBIPUJI	RAMBIPUJI
11	Kaspil Jubung	DUSUN GAYAM RT/RW 001/024	RAMBIGUNDAM	RAMBIPUJI
12	Kurnia Tk.	JL. LETJEN SUTOYO	SUMBERSARI	
13	Lestari Pakem	JL. BASUKI RAHMAT 291	TEGAL BESAR	KALIWATES
14	Lestari Rambli	JL. DARMAWANGSA RT/RW 003/003	KALIWINING	RAMBIPUJI
15	Machriyatun (Tun)	JL. KAPTEN TENDEAN	KARANGREJO	
16	Mahsus	PERUM GRIYA MANGLI INDAH RT/RW 001/004	MANGLI	KALIWATES
17	Maskuri Tk.	JL. IMAM BONJOL RT/RW 003/005	TEGAL BESAR	KALIWATES
18	Misro Jubung	DSN KRAJAN RT/RW 006/001	JUBUNG	SUKORAMBI
19	Mitra Lancar	JL. BASUKI RAHMAT Gang Nuri No.53	KRANJINGAN	SUMBERSARI
20	Nursumi B. Rahmat	JL. BASUKI RAHMAT	KEBONSARI	
21	Parni MT Haryono	JL. MT. HARYONO	KARANGREJO	SUMBERSARI

22	Raung Motor	MANGLI		
23	Ridwan	JL KAPTEN P. TANDEAN 57	KARANGREJO	SUMBERSARI
24	Roni Rambli	JL. DARMAWANGSA RT/RW 002/003 Gang Putra Gayam	KALIWINING	RAMBIPUJI
25	Saha	JL. KAPTEN TENDEAN	KARANGREJO	
26	Sederhana Tk.	JL. BASUKI RAHMAT 58	KRANJINGAN	KALIWATES
27	Sofyan Mangli	PANTI, MANGLI, JEMBER		
28	Suari	JL. MT HARYONO	WIROLEGI	SUMBERSARI
29	Suher	JL. BRAWIJAYA DSN KRAJAN RT/RW 001/002	JUBUNG	SUKORAMBI
30	Sumbersari Tk.	Jl. BATURADEN	TEGAL GEDE	
31	Sumber Mas Sukorejo	JL. KAPTEN TENDEAN 61	KARANGREJO	SUMBERSARI
32	H. Suroto	Jl. Piere Tendean RT/RW 002/004	SUMBERSARI	SUMBER SARI
33	B. Tatik	JL. PIERE TENDEAN 57	KARANGREJO	SUMBERSARI
34	Taufik Kranjingan	KRANJINGAN	KARANGREJO	
35	B. Ton	JAL. KAPTEN PIEREE TENDEAN		SUMBERSARI
36	Umam Kranjingan	JL. KAPTEN TENDEAN 66	KARANGREJO	SUMBERSARI
37	Urip Tendean	Jl. KP. Tendean Gg. Remaja 103 RT/RW 003/009 No. 6	SUMBERSARI	SUMBER SARI
38	Vivi Mangli	JL. PB SUDIRMAN, PANTI, MANGLI	SUMBERSALAK	LEDOK OMBO
39	Warung Pincuk	JL. HAYAM WURUK	SEMPUSARI	
40	Wijaya Mangli	JL. HAYAM WURUK 189 RT/RW 003/009, MANGLI	SEMPUSARI	KALIWATES
41	Wiji	JL. MT. HARYONO	KARANGREJO	SUMBERSARI
42	Zein Market (Lin)	JL. YOS SUDARSO	KARANGREJO	SUMBER SARI
43	Abdollah / Iva	DUSUN KRAJAN BARAT RT/RW 01/01	CANDIJATI	ARJASA
44	Adi Jl. Lumba-Lumba	Jl. Lumba-Lumba No. 97 RT/RW 003/001	SEMPUSARI	KALIWATES

45	Adim Dr. Subandi	JL. Dr. SUBANDI	PATRANG	
46	Adip S	DUSUN TENGIR TIMUR RT/RW 004/001	TENGIR TIMUR	JELBUK
47	Agus Kejayan	DUSUN MRAPEN RT/RW 03/10	SUMBER KEJAYAN	MAYANG
48	Aida Mangku bumi	JL. TRUNOJOYO, PERUM MANGKUBUMI	JEMBER KIDUL	KALIWATES
49	Akbar	DUSUN GARAHAN JATI RT/RW 002/003	GARAHAN	SILO
50	Ali G. Mada	JL. GAJAH MADA		KALIWATES
51	Aneka Cell	JL. RAYA RAUNG KRAJAN TEGAL REJO RT/RW 006/001	KRAJAN	MAYANG
52	Anton	JL. SUMATRA 2/2 RT/RW 01/05 Gang VII	SUMBERSARI	SUMBER SARI
53	Ar	DUSUN KOJUK RT/RW 01/08	SUKOKERTO	SUKOWONO
54	Asep	Jl. MERAK	GEBANG	PATRANG
55	Ayu Ilham	JL. KOPTU BERLIAN	ANTIROGO	SUMBERSARI
56	Ayu / Mayang	MAYANG		
57	Barokah	JL. KRAJAN	SUMBER WRINGIN	SUKOWONO
58	Baru Ajung / Hartono	JL. MH. THAMRIN RT/RW 1/7	AJUNG	KALISAT
59	Baru Sempolan	JL. PB SUDIRMAN RT/RW 06/02	SEMPOLAN	SILO
60	Baru Sukosari	JL. PAHLAWAN	SUKOREJO	SUKOWONO
61	Bary Slawu	JL. MANYAR	SLAWU	PATRANG
62	BB Jaya Cempaka Tk	JL. CEMPAKA RT/RW 003/002	GEBANG	PATRANG
63	Bintang Mas	Toko Bintang, JL. RAYA RAUNG RT/RW 01/01	TEGALREJO	MAYANG
64	Candra PR. Kebon Agung	PERUM KEBON AGUNG INDAH	KEBONAGUNG	KALIWATES
65	Dita	JL. RASAMALA, BARATAN, PATRANG		
66	Doremon Trunojoyo	JL. TRUNOJOYO		KALIWATES
67	Eko	JL. TAWANGMANGU RT/RW 001/008	SUMBERSARI	SUMBERSARI

68	Erfan	JL BANYUWANGI	SUMBER KEJAYAN	MAYANG
69	Erna	DUSUN KRAJAN RT/RW 05/05	SUMBER WRINGIN	SUKOWONO
70	Evi	Jl. Kasuari Lingk. Kedawung Lor RT/RW 002/008	GEBANG	PATRANG
71	Fausan	JL. MRAPEN, SUMBERKEJAYAN, MAYANG	SUMBERKEJAYAN	MAYANG
72	Fian	DUSUN TEGALAN SUMBER KEJAYAN	SUMBERKEJAYAN	MAYANG
73	Fuji Ayu	JL. TAPAK SIRING	ANTIROGO	SUMBERSARI
74	Husen H.	SUMBERWRINGIN RT/RW 007/004	SUMBERWRINGIN	SUKOWONO
75	Hariyanto	DUSUN KLAYU RT/RW 03/02	TEGAL WARU	MAYANG
76	Hotija	JL. CEMPEDAK	TEGAL WARU	MAYANG
77	Ido	JL> PAHLAWAN, DESA TEGAL WARU DUSUN WUYU	TEGAL WARU	MAYANG
78	Imel Cell	JL. KRAJAN, DUSUN KRAJAN RT/RW 001/002		SUKOWONO
79	Ismail L. Suprpto	JL. LETJEN SUPRAPTO		
80	Ita	DUSUN TENGGIR TIMUR	JELBUK	JELBUK
81	Jamik	DUSUN KRAJAN UTARA	PATEMON	PAKUSARI
82	Jamilan	JL. TERATAI JEMBER		
83	Jery	PERUM KEBON AGUNG	KAEBON AGUNG	KALIWATES
84	Kedawung Kaswari	JL. KASUARI ,GANG IV, LINGK.KEDAWUNG KIDUL RT/RW 003/001	GEBANG	PATRANG
85	Kijang Mas	JL. H. SAMANHUDI LINGK. KULON PASAR	JEMBER KIDUL	KALIWATES
86	Komariyah	DUSUN ONJUR RT/RW 004/008 SEMPOLAN		
87	Like	JL. BANYUWANGI, DUSUN KRAJAN	SEMPOLAN	SILO
88	Marin UD.	Perum. ISTANA TEGAL BESAR, JL. Moh. YAMIN		KALIWATES
89	Maryam Cempaka	JL. KACA PIRING	GEBANG	PATRANG
90	Mayang Jaya	JLN RAYA RAUNG, no 108, KALISAT	MAYANG	MAYANG

91	Muhtar	DSN DUK LENGKONG RT/RW 002/015	PATEMON	PAKUSARI
92	Narto	BARATAN		
93	Nora	DUSUN KRAJAN SUMBERJATI SILO		
94	Novi/Tk.Robi	Gang. ASOKA, BARATAN	BARATAN	PATRANG
95	Posi / Ahmad Fauzy	JL IMAM MUSTOFA, DUSUN KOJUK RT/RW 01/02	SUKOKERTO	SUKOWONO
96	Pran	DUSUN MRAPEN		
97	Putra Jaya	JL. SRIWIJAYA	KARAGREJO	SUMBERSARI
98	Rais Kedawung	KOMPLEKS PASAR TANJUNG 245-253, JEMBER	GEBANG	PATRANG
99	Rejeki Baru	JL. PANGANDARAN RT/RW 2/2	ANTIROGO	SUMBER SARI
100	Ribut	JL. KALIURANG	TEGALGEDE	SUMBERSARI
101	Ri	JL. DANAUTOBA	TEGALGEDE	SUMBERSARI
102	Riro	JL SLAMET RIYADI GG SENTAL	BARATAN	PATRANG
103	Rus	JL. KH. ABDUL AZIZ, DUSUN MRAPEN RT/RW 01/01	SUMBER KEJAYAN	MAYANG
104	Slamet Kaswari	JL CEMPAKA	GEBANG	PATRANG
105	Stevanus	JL. MASTRIP		SUMBERSARI
106	Sudarmono	JL. SUMATRA		SUMBERSARI
107	Sulis	JL. KACA PIRING III		
108	Tanjung Jaya	JL. KARIMATA		SUMBERSARI
109	Tatik	DUSUN SUMBER TENGAH RT/RW 002/001 PANDUMAN		
110	Temu	JL. GAJAH MADA		
111	Tila	JL. PANGANDARAN RT/RW 2/2	ANTIROGO	SUMBERSARI
112	Titi Manggar	JL. MANGGAR LINGK. GEBANG PORENG		
113	Tres	PANDOMAN, JELBUK	JELBUK	JELBUK

114	Veri Antirogo	JL. TAWANG MANGU	SUMBERSARI	SUMBERSARI
115	Veri Jelbuk	SUCO PANGEPOK		JELBUK
116	Waris	SUKOWIRYO		
117	Weny	JL. TAWANGMANGU LING. KRAJAN BARAT RT/RW 02/06	TEGAL GEDE	SUMBER SARI
118	Wili Srikoyo	JL. SRIKOYO GG. TEGAL BATU		PATRANG
119	Wiwik	BITING	BITING	
120	Yahya (Asak)	JL DIPONEGORO	KALISAT	KALISAT
121	Yanto Anggur	JL. ANGGUR		PATRANG
122	Yanto Sempolan	JL. CUMEDAK	PLATANGAN	KALISAT
123	Yosdi	DUSUN KRAJAN RT/RW 03/03	SEMPOLAN	SILO
124	Yudi	JL. TAWANG MANGU		