



PENYELESAIAN *BIOBJECTIVE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS* DENGAN ALGORITMA GENETIKA DAN ALGORITMA *CAT SWARM OPTIMIZATION*

SKRIPSI

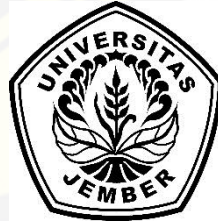
Oleh :

Latifah Angraini Suyono

NIM 141810101006

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

2018



PENYELESAIAN *BIOBJECTIVE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS* DENGAN ALGORITMA GENETIKA DAN ALGORITMA *CAT SWARM OPTIMIZATION*

SKRIPSI

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

Oleh :

**Latifah Anggraini Suyono
141810101006**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Orang tuaku tercinta Ayah Suyono, Mama Wiwik Harinastuti, serta Ibu Debbi Andriana yang telah memberikan limpahan kasih sayang, doa, dukungan yang tiada henti, serta memotivasi untuk setiap langkah saya. Tidak ada yang dapat membalas semua jasa dan kasih sayang yang telah kalian berikan kecuali Allah SWT;
2. seluruh dosen Jurusan FMIPA Matematika, terimakasih atas bimbingan, ilmu, dan motivasi selama saya menjadi mahasiswa;
3. almamater tercinta, Jurusan Matematika FMIPA Universitas Jember, bapak dan ibu guru SMA Negeri 1 Sampang, SMP Negeri 1 Sampang, dan SDN Gunungsekar 1 Sampang yang telah berjasa membimbing dan memberikan ilmunya kepada penulis;
4. sahabatku Nuraini Rahila dan Ayu Mayasari yang telah menjadi sahabat terbaik, serta memberikan dukungan, semangat, dan motivasi dalam hidup saya;
5. sahabat-sahabatku Novi Intansari, Niya Liyani, serta Hanifah Husnul Umami yang telah memberikan semangat dan dukungannya., terimakasih untuk semuanya;
6. teman-teman seperjuangan Extreme'14 yang selalu membantu dan memberi dukungan;

MOTTO

"Sesungguhnya sesudah kesulitan itu ada kemudahan, maka apabila kamu selesai (dari suatu urusan) kerjakanlah dengan sungguh-sungguh urusan yang lain, dan hanya kepada Allah lah hendaknya kamu berharap."

(Surat Asy Syarh ayat 6-8)^{*)}

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya."

(Surat Al Baqarah: 286)^{*)}

*) Departemen Agama Republik Indonesia. 2004. Al-Quran dan Terjemahnya.
Bandung: J-ART

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

nama : Latifah Anggraini Suyono

NIM : 141810101006

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penyelesaian *Biobjective Vehicle Routing Problem with Time Windows* dengan Algoritma Genetika dan Algoritma *Cat Swarm Optimization*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan 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 ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juli 2018

Yang menyatakan,

Latifah Anggraini Suyono

141810101006

SKRIPSI

PENYELESAIAN *BIOBJECTIVE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS* DENGAN ALGORITMA GENETIKA DAN ALGORITMA *CAT SWARM OPTIMIZATION*

Oleh
Latifah Angraini Suyono
NIM 141810101006

Pembimbing:

Dosen Pembimbing Utama : Ahmad Kamsyakawuni, S.Si., M.Kom
Dosen Pembimbing Anggota : Abduh Riski, S.Si., M.Si

PENGESAHAN

Skripsi berjudul “Penyelesaian *Biobjective Vehicle Routing Problem with Time Windows* dengan Algoritma Genetika dan Algoritma *Cat Swarm Optimization*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

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

Abduh Riski, S.Si., M.Si.
NIP. 199004062015041001

Anggota II,

Anggota III,

Drs. Rusli Hidayat, M.Sc.
NIP. 196610121993031001

Bagus Juliyanto, S.Si., M.Si.
NIP. 198007022003121001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Penyelesaian *Biobjective Vehicle Routing Problem with Time Windows* dengan Algoritma Genetika dan Algoritma *Cat Swarm Optimization*; Latifah Anggraini Suyono, 141810101006; 2018; 123 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Jember.

Perkembangan jumlah penduduk yang semakin meningkat menyebabkan jumlah kebutuhan hidup yang harus dipenuhi juga semakin banyak. Para produsen barang atau jasa saling berlomba untuk dapat memberikan pelayanan terbaik dalam memenuhi kebutuhan masyarakat. Di era persaingan usaha yang sangat ketat, para produsen dituntut untuk dapat memberikan pelayanan dalam memenuhi kebutuhan masyarakat. Untuk menjangkau pasar yang luas dalam memenuhi keinginan pelanggan, setiap perusahaan harus membuat strategi pemasaran produknya yaitu dengan membuat jalur distribusi yang efisien. Para produsen barang akan berusaha secara maksimal untuk menekan biaya distribusi barang sampai seminimal mungkin untuk memenuhi kebutuhan pelanggannya yang tersebar di beberapa lokasi. Permasalahan optimasi pelayanan kepada pelanggan seperti ini disebut *Vehicle Routing Problem (VRP)*. *Vehicle Routing Problem (VRP)* merupakan permasalahan distribusi yang di dalamnya terdapat sebuah *problem* dimana terdapat sejumlah rute yang harus dipilih dan dilewati sejumlah kendaraan untuk sampai kepada pelanggan atau konsumennya. Bentuk variasi lainnya dari jenis VRP ini yaitu salah satunya *Vehicle Routing Problem with Time Windows (VRPTW)*. *Vehicle Routing Problem with Time Windows* merupakan variasi permasalahan VRP dimana terdapat tambahan kendala yaitu batasan interval waktu pelayanan. Permasalahan VRPTW ini untuk mencapai solusi yang lebih optimal dapat menekan pada dua fungsi tujuan khusus yaitu meminimalkan jumlah kendaraan dan jumlah total jarak tempuh yang disebut juga dengan *Biobjective VRPTW*.

Penelitian ini akan menyelesaikan permasalahan *Biobjective Vehicle Routing Problem with Time Windows* menggunakan algoritma genetika dan

algoritma *Cat Swarm Optimization* (CSO) dan untuk mengetahui hasil dan waktu komputasi dari penerapan algoritma tersebut. Data yang digunakan pada penelitian ini adalah data yang diambil dari skripsi Faridah (2017) yang bersumber dari perusahaan air minum kemasan PT. Tujuh Impian Bersama AMDK (Air Minum Dalam Kemasan) Al-Qodiri Jember. Data-data tersebut antara lain nama barang, jumlah barang, harga beli, dan harga jual. Penyelesaian dibantu *software* MATLAB R2015b.

Hasil penelitian menunjukkan bahwa Solusi terbaik yang mampu ditemukan oleh kedua algoritma untuk data 32 pelanggan yaitu untuk algoritma genetika Rp 817.900,00 dengan jumlah kendaraan 3 dan total jarak tempuh 435,8km, serta untuk algoritma CSO Rp 777.500,00 dengan jumlah kendaraan 3 dan total jarak tempuh 355 km. Solusi terbaik yang mampu ditemukan oleh kedua algoritma untuk data 57 pelanggan yaitu untuk algoritma genetika Rp 2.358.300,00 dengan jumlah kendaraan 10 dan total jarak tempuh 716,6 km, serta untuk algoritma CSO Rp 2.153.250,00 dengan jumlah kendaraan 9 dan total jarak tempuh 706,5 km. Berdasarkan hasil tersebut, terlihat bahwa solusi dari CSO lebih minimal daripada solusi dari algoritma genetika, atau dapat dikatakan CSO lebih efektif untuk menemukan solusi optimal dibandingkan algoritma genetika. Hal ini dikarenakan pada setiap iterasi algoritma CSO mendapatkan solusi baru sejumlah kucing *tracing* ditambah dengan sejumlah kucing *seeking* dikali parameter SMP, sedangkan pada algoritma genetika maksimal hanya dua solusi hasil *crossover* dan tiga solusi hasil mutasi.

PRAKATA

Puji syukur kehadiran Allah SWT. Atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penyelesaian *Biobjective Vehicle Routing Problem with Time Windows* dengan Algoritma Genetika dan Algoritma *Cat Swarm Optimization*”. Skripsi ini disusun untuk memenuhi salah satu syarat 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. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Drs. Sujito, Ph.D. selaku Dekan Fakultas MIPA Universitas Jember dan Kusbudiono S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Jember yang telah memberikan fasilitas-fasilitas dalam tahap penelitian;
2. Ahmad Kamsyakawuni, S.Si., M.Kom selaku Dosen Pembimbing Utama dan Abduh Riski, S.Si., M.Si selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Drs. Rusli Hidayat, M.Sc. selaku Dosen Penguji I dan Bagus Juliyanto, S.Si., M.Si selaku Dosen Penguji II yang telah memberikan kritik serta sarannya terhadap penulisan skripsi ini;
4. Seluruh staf pengajar Jurusan Matematika Fakultas MIPA Universitas Jember yang telah memberikan ilmu serta bimbingannya sehingga penulis dapat menyelesaikan skripsi ini;
5. Ayah, Ibu, dan Mama tercinta yang telah memberikan do’a, kasih sayang dan motivasi yang sangat berharga dalam menyelesaikan skripsi ini;
6. Guru-guruku sejak taman kanak – kanak sampai dengan perguruan tinggi yang telah memberikan ilmu yang bermanfaat;

7. sahabat-sahabat saya yang telah memberikan do'a, motivasi, dan saran yang sangat berharga dalam menyelesaikan skripsi ini;
8. teman-teman seperjuanganku Annisa Yuniar Hidayah, Firda Anisa Fajarini, Elsa Lorensa yang senantiasa memberikan dukungan dan semangat;
9. teman-teman angkatan 2014 (EXTREME) yang telah memberikan dukungan, keceriaan, dan canda-tawa yang telah diberikan;
10. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat dan memberikan sumbangan pengetahuan bagi pembaca.

Jember, Juli 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
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 Teori Graf.....	5
2.2 <i>Vehicle Routing Problem</i> (VRP)	6
2.2.1 <i>Vehicle Routing Problem with Time Windows</i> (VRPTW)....	8
2.2.2 <i>Biobjective VRPTW</i>	10
2.3 Algoritma Genetika.....	11
2.4 Algoritma <i>Cat Swarm Optimization</i> (CSO).....	18
BAB 3. METODE PENELITIAN	24
BAB 4. HASIL DAN PEMBAHASAN	29
4.1 Hasil	29

4.4.1 Perhitungan Manual.....	29
4.4.2 Program.....	57
4.4.3 Hasil Percobaan	60
4.2 Pembahasan.....	69
BAB 5. KESIMPULAN DAN SARAN	72
5.1 Kesimpulan	72
5.2 Saran	72
DAFTAR PUSTAKA	74
LAMPIRAN.....	76

DAFTAR GAMBAR

	Halaman
2.1 Contoh graf berbobot	5
2.2 Graf komplit dalam permasalahan VRP	7
2.3 Contoh salah satu penyelesaian permasalahan VRP	7
2.4 Contoh <i>two-point order crossover</i>	15
2.5 <i>Flowchart</i> alur proses <i>crossover</i>	15
2.6 Contoh <i>slide mutation</i>	16
2.7 <i>Flowchart</i> alur proses mutasi	17
2.8 <i>Flowchart</i> alur proses algoritma genetika.....	18
2.9 <i>Flowchart</i> alur proses algoritma CSO	23
3.1 Skema alur penelitian	28
4.1 Tampilan awal program <i>Biobjective VRPTW</i>	57
4.2 Tampilan input parameter program <i>Biobjective VRPTW</i>	59
4.2 Tampilan output program <i>Biobjective VRPTW</i>	60

DAFTAR TABEL

	Halaman
4.1 Jarak Antara Depo Dengan Toko Dan Jarak Antar Toko	30
4.2 Jumlah Permintaan Setiap Toko dan Waktu Operasional.....	30
4.3 Kecepatan Rata-Rata Kendaraan.....	30
4.4 Kromosom Awal	32
4.5 Hasil Rute Kromosom Awal	37
4.6 Hasil Kromosom Urut.....	38
4.7 Nilai <i>Fitness</i> Setiap Individu.....	38
4.8 Probabilitas Relatif Setiap Individu	39
4.9 Hasil Probabilitas Kumulatif.....	39
4.10 Hasil Proses <i>Crossover</i>	41
4.11 Induk Mutasi	41
4.12 <i>Offspring</i> Hasil Mutasi	42
4.13 Hasil Proses Mutasi.....	42
4.14 Kromosom Gabungan	43
4.15 Kromosom Gabungan Urut.....	44
4.16 Kromosom Populasi Baru	44
4.17 Populasi Awal Kucing	45
4.18 Kecepatan Awal Kucing	46
4.19 Representasi Permutasi Populasi Awal.....	46
4.20 Hasil Rute Populasi Kucing Awal	47
4.21 Hasil Populasi Awal.....	48
4.22 SPC Populasi Awal	48
4.23 Penempatan Flag	49
4.24 Representasi Permutasi Posisi Kucing Baru	50
4.25 Hasil Proses <i>Tracing Mode</i>	51
4.26 Hasil Solusi Kandidat Kucing 4.....	52
4.27 Hasil Representasi Permutasi Kandidat Solusi Kucing 4	53
4.28 Hasil <i>Seeking Memory Pool</i> Dari Kucing 4	53

4.29 Hasil Probabilitas Terpilih Kandidat Solusi Kucing 4.....	55
4.30 Hasil Probabilitas Kumulatif Kandidat Solusi Kucing 4	55
4.31 Hasil Proses <i>Seeking Mode</i>	56
4.32 Hasil Perbandingan Nilai Fitness	56
4.33 Hasil Percobaan Pengaruh Parameter P_c	61
4.34 Hasil Percobaan Pengaruh Parameter P_m	61
4.35 Hasil Percobaan Pengaruh Parameter MR	62
4.36 Hasil Percobaan Pengaruh Parameter c	62
4.37 Hasil Percobaan Pengaruh Parameter CDC	63

DAFTAR LAMPIRAN

	Halaman
A1. Jumlah Permintaan dan Waktu Operasional untuk 32 Toko.....	76
A2. Jarak 32 Toko.....	77
A3. Kecepatan Rata-Rata Kendaraan untuk 32 Toko.....	79
A4. Jumlah Permintaan dan Waktu Operasional untuk 57 Toko.....	82
A5. Jarak 57 Toko.....	83
A6. Kecepatan Rata-rata Kendaraan untuk 57 Toko.....	90
B1. Data Uji Percobaan Perubahan Parameter Algoritma Genetika pada 32 Toko.....	99
B2. Data Uji Percobaan Perubahan Parameter Algoritma <i>Cat Swarm</i> <i>Optimization</i> pada 32 Toko.....	101
B3. Data Uji Percobaan Perubahan Parameter Algoritma Genetika pada 57 Toko.....	104
B4. Data Uji Percobaan Perubahan Parameter Algoritma <i>Cat Swarm</i> <i>Optimization</i> pada 57 Toko.....	106
B5. Data Uji Percobaan Algoritma Genetika pada 32 Toko.....	110
B6. Data Uji Percobaan Algoritma <i>Cat Swarm Optimization</i> pada 32 Toko.....	111
B7. Data Uji Percobaan Algoritma Genetika pada 57 Toko.....	113
B8. Data Uji Percobaan Algoritma <i>Cat Swarm Optimization</i> pada 57 Toko.....	116
B9. Script Program.....	120

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan jumlah penduduk yang semakin meningkat menyebabkan jumlah kebutuhan hidup yang harus dipenuhi juga semakin banyak. Kebutuhan hidup adalah segala sesuatu yang sangat diperlukan oleh manusia untuk mempertahankan hidupnya. Selain dapat dipenuhi sendiri, kebutuhan hidup manusia sebagian besar dipenuhi oleh perusahaan-perusahaan yang bergerak di bidang jasa maupun barang dengan kuantitas tertentu dan waktu tertentu sesuai kebutuhan.

Para produsen barang atau jasa saling berlomba untuk dapat memberikan pelayanan terbaik dalam memenuhi kebutuhan masyarakat. Di era persaingan usaha yang sangat ketat, para produsen dituntut untuk dapat memberikan pelayanan dalam memenuhi kebutuhan masyarakat dengan cepat, tepat, dan murah seiring dengan perkembangan teknologi. Untuk menjangkau pasar yang luas dalam memenuhi keinginan pelanggan, setiap perusahaan harus membuat strategi pemasaran produknya. Strategi pemasaran yang dilaksanakan dengan membuat jalur distribusi yang efisien. Strategi yang diterapkan untuk setiap jenis konsumen berbeda.

Distribusi merupakan salah satu permasalahan yang cukup penting pada manajemen perusahaan karena berkaitan dengan biaya. Biaya yang dibutuhkan dalam pendistribusian barang memiliki pengaruh yang besar karena biaya ditentukan oleh banyaknya jumlah kendaraan dan jarak tempuh ke konsumen. Para produsen barang akan berusaha secara maksimal untuk menekan biaya distribusi barang sampai seminimal mungkin untuk memenuhi kebutuhan pelanggannya yang tersebar di beberapa titik atau lokasi. Permasalahan optimasi pelayanan kepada pelanggan seperti ini disebut *Vehicle Routing Problem* (VRP).

VRP merupakan permasalahan distribusi yang di dalamnya terdapat sebuah *problem* dimana terdapat sejumlah rute yang harus dipilih dan dilewati sejumlah kendaraan untuk sampai kepada pelanggan atau konsumennya. Bentuk variasi lainnya dari jenis VRP ini yaitu salah satunya *Vehicle Routing Problem with Time*

Windows (VRPTW). Vehicle Routing Problem with Time Windows (VRPTW) merupakan suatu permasalahan menentukan rute perjalanan pendistribusian barang dengan mempertimbangkan waktu yang ada, baik dari *customer* maupun perusahaan. Permasalahan VRPTW ini untuk mencapai solusi yang lebih optimal dapat menekan pada dua fungsi tujuan khusus yaitu meminimalkan jumlah kendaraan dan jumlah total jarak tempuh yang disebut juga dengan *Biobjective VRPTW*.

Menurut Taqwiya (2017), algoritma genetika dapat menghasilkan total jarak tempuh yang minimal pada permasalahan VRPTW. Menurut Ombuki *et al* (2006), penerapan algoritma genetika yang diusulkan untuk menyelesaikan VRPTW menghasilkan solusi yang cukup efektif dalam meminimalkan jumlah kendaraan dan total biaya jarak. Penelitian lain tentang *Traveling Salesman Problem (TSP)* menggunakan algoritma CSO dan PSO yang telah dilakukan oleh Vitaloka (2016). Menurut Vitaloka (2016), algoritma CSO lebih baik untuk menentukan total jarak tempuh yang minimum daripada algoritma PSO. Selain penelitian tersebut, juga terdapat penelitian tentang Biobjektif pada permasalahan VRPTW yang telah dilakukan oleh Najera dan Bullinaria (2009). Penelitian tersebut menyelesaikan permasalahan khusus yaitu meminimalkan jumlah rute dan total biaya kendaraan. Najera dan Bullinaria mengusulkan metode pengukuran kesamaan rute dan memasukkannya dalam algoritma evolusi untuk menyelesaikan *Biobjective VRPTW*.

Berdasarkan penjelasan di atas, penulis tertarik untuk melakukan penelitian menyelesaikan permasalahan *Biobjective VRPTW* dengan dua fungsi tujuan khusus yaitu meminimalkan jumlah kendaraan dan meminimalkan total jarak tempuh yang diterapkan pada algoritma genetika dan *Cat Swarm Optimization (CSO)*. Hasil penerapan dari kedua algoritma tersebut akan dibandingkan berdasarkan solusi total kendaraan dan total jarak tempuh yang paling minimum. Selain dilihat berdasarkan kedua solusi tersebut juga akan dibandingkan berdasarkan hasil *running time* untuk mendapatkan solusi yang lebih baik.

1.2 Rumusan Masalah

Berdasarkan gambaran yang telah tertulis diatas, maka rumusan permasalahan yang akan dibahas yaitu:

- a. Bagaimana menerapkan algoritma Genetika dan algoritma *Cat Swarm Optimization* (CSO) dalam menyelesaikan permasalahan *Biobjective Vehicle Routing Problem with Time Windows* (VRPTW) untuk memperoleh total jarak tempuh dan jumlah kendaraan yang minimum?
- b. Bagaimana perbandingan hasil algoritma Genetika dan algoritma *Cat Swarm Optimization* (CSO) dilihat dari biaya yang dihitung berdasarkan jumlah kendaraan dan total jarak tempuh?

1.3 Batasan Masalah

Permasalahan dalam penelitian ini dibatasi oleh beberapa masalah yaitu:

- a. Barang yang didistribukan homogen
- b. Setiap kendaraan yang digunakan dianggap memiliki kapasitas yang sama
- c. Waktu pelayanan dianggap sama untuk setiap barang
- d. Jarak antar titik dianggap simetris
- e. Kondisi jalan dan kendaraan dianggap tidak ada kendala
- f. Waktu buka dan tutup toko hanya sekali

1.4 Tujuan Penelitian

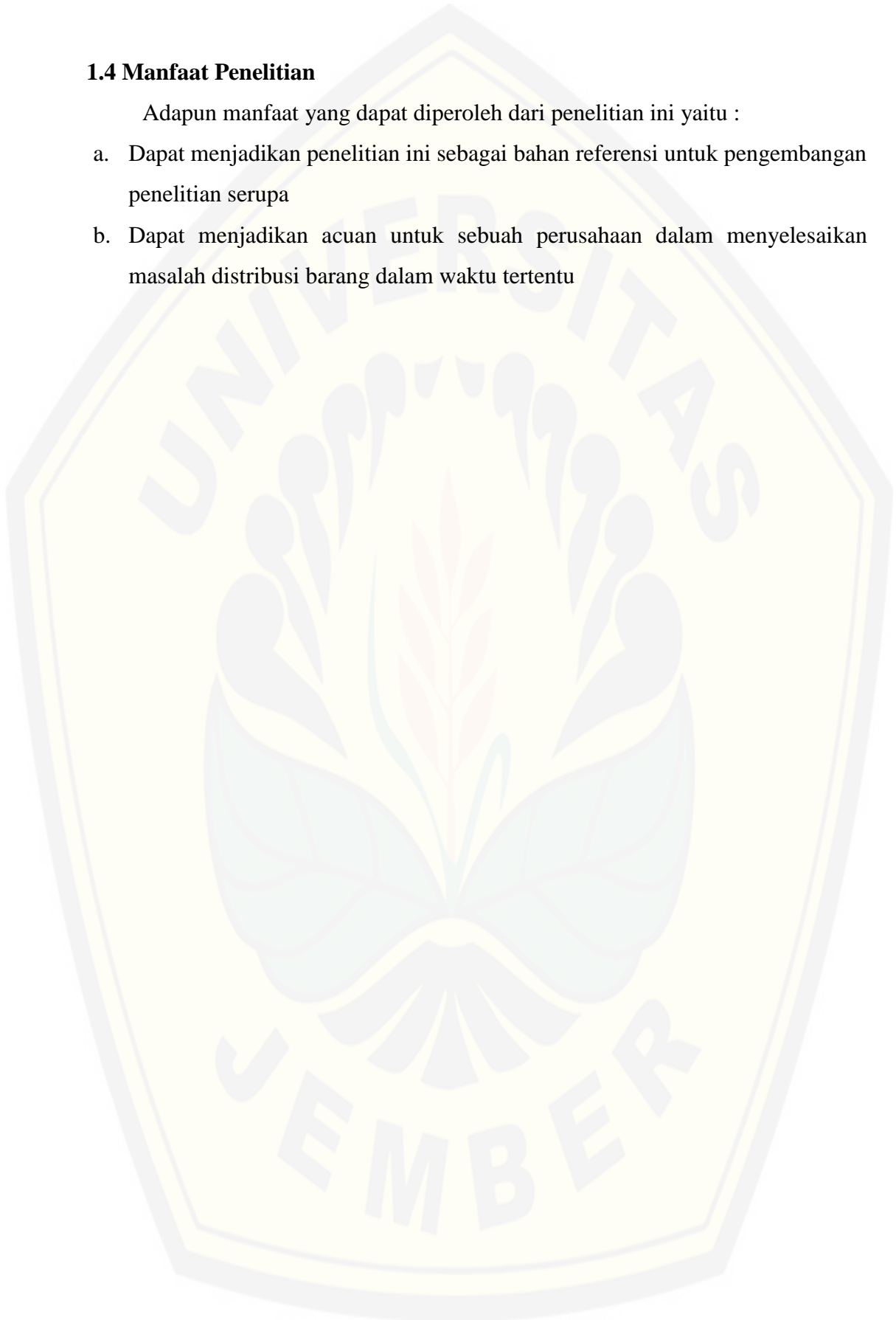
Adapun tujuan dari penelitian ini yaitu :

- a. Menerapkan algoritma Genetika dan *Cat Swarm Optimisation* (CSO) dalam menyelesaikan *Biobjective Vehicle Routing Problem with Time Windows* (VRPTW) untuk memperoleh total jarak tempuh dan jumlah kendaraan yang minimum;
- b. Mengetahui perbandingan hasil penerapan algoritma Genetika dan *Cat Swarm Optimisation* (CSO) dilihat dari biaya yang dihitung berdasarkan jumlah kendaraan dan total jarak tempuh.

1.4 Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini yaitu :

- a. Dapat menjadikan penelitian ini sebagai bahan referensi untuk pengembangan penelitian serupa
- b. Dapat menjadikan acuan untuk sebuah perusahaan dalam menyelesaikan masalah distribusi barang dalam waktu tertentu

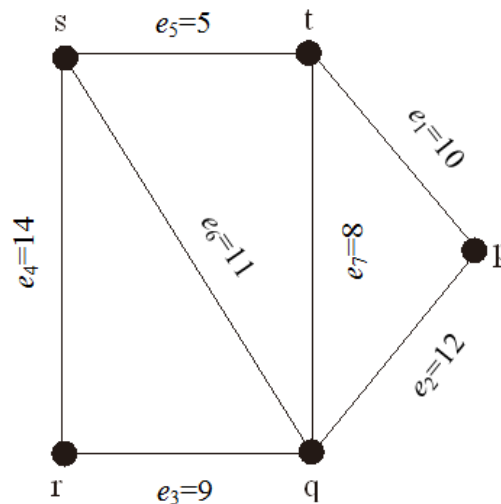


BAB 2. TINJAUAN PUSTAKA

2.1 Teori Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$ yang terdiri atas sekumpulan objek $V = \{v_1, v_2, \dots, v_n\}$ yang dalam hal ini V adalah himpunan tak kosong dari titik (*vertices*), dan E adalah himpunan sisi (*edge*) yaitu $E = \{e_1, e_2, \dots, e_n\}$ yang menghubungkan sepasang titik (Munir, 2005).

Graf yang setiap sisinya memiliki arah atau panah disebut graf berarah. Sebaliknya, graf yang setiap sisinya tidak memiliki arah atau panah disebut graf tak berarah. Graf berbobot merupakan graf yang setiap sisinya dikaitkan dengan bilangan *real* positif. Jumlah bobot dari semua sisi dalam graf G dinotasikan dengan $w(G)$. Berikut merupakan contoh graf berbobot (Gambar 2.1).



Gambar 2.1 Contoh graf berbobot

Suatu graf disebut graf terhubung jika terdapat lintasan yang menghubungkan setiap dua titik berbeda. Perjalanan (*walks*) dalam graf $G = (V, E)$ merupakan barisan berhingga dalam bentuk $W = \{v_1, e_1, v_2, e_2, v_3, e_3, \dots, e_{n-1}, v_n\}$ dimana sisi e_t menghubungkan titik v_i dengan v_{i+1} . Contoh perjalanan dalam graf sesuai Gambar 2.1 adalah $\{p, e_2, q, e_3, r, e_4, s, e_5, t, e_1\}$. Perjalanan yang melewati titik yang berbeda-beda

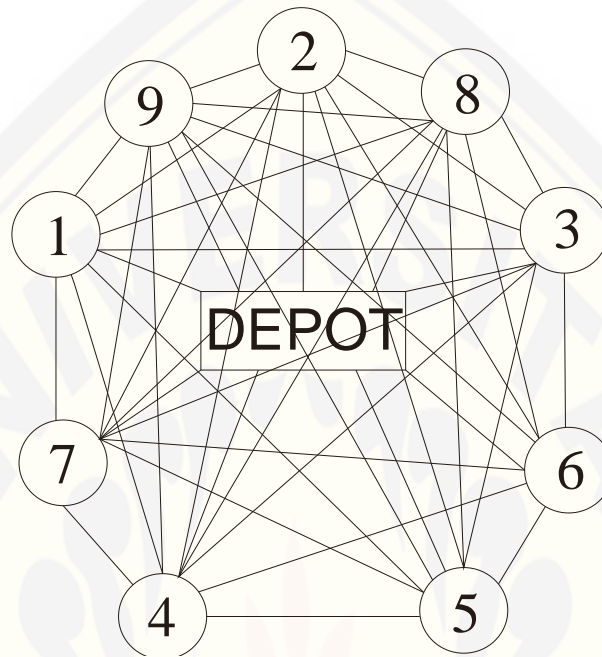
disebut lintasan (*trails*). Lintasan yang berawal dan berakhir pada titik yang sama disebut sirkuit (*circuit*) (Chartrand and Oellermann, 1993).

2.2 Vehicle Routing Problem (VRP)

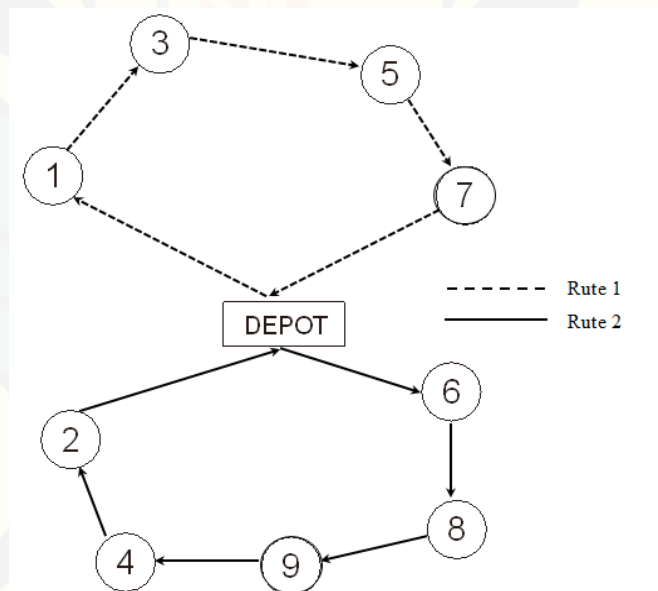
Vehicle Routing Problem (VRP) pertama kali dipelajari oleh Dantzig dan Ramser pada tahun 1959 dalam bentuk rute dan penjadwalan truk. VRP merupakan sebuah permasalahan optimasi dan merancang rute kendaraan dengan biaya minimal dimana setiap kendaraan berawal dan berakhir di depot yang sama. Permasalahan ini berhubungan dengan menjemput atau mengantar barang pada sejumlah tempat.

Permasalahan VRP merupakan generalisasi kasus permasalahan *Travelling Salesman Problem* (TSP). Kasus VRP ini juga sering disebut *Multiple Travelling Salesman Problem* (m-TSP) dimana dalam kasus ini terdapat lebih dari satu salesman yang mengunjungi setiap kota yang memulai perjalanannya dari suatu depot dan pada akhir perjalanannya harus kembali ke depot tersebut. Tujuan kasus VRP ini yaitu untuk meminimalkan total biaya dari rute perjalanan. Dalam hal ini, total biaya mempunyai beberapa representasi seperti jumlah kendaraan, jarak tempuh, ataupun waktu.

Vehicle Routing Problem (VRP) didefinisikan dalam bentuk sebuah graf terhubung. Graf $G = (V, E)$ dimana $V = \{v_1, v_2, \dots, v_n\}$ merupakan himpunan titik, sedangkan $E = \{e_1, e_2, \dots, e_n\}$ merupakan himpunan sisi yang menghubungkan antar titik. Dalam hal ini, v_o menunjukkan sebuah depot, v_i menunjukkan pelanggan i , $e = (v_i, v_j)$ menunjukkan jalan yang menghubungkan antara pelanggan i dan pelanggan j dengan bobot $w(v_i, v_j)$ yang merupakan jarak dari depot ke toko atau jarak antar toko. Contoh penyelesaian permasalahan VRP dapat dilihat pada Gambar (2.2) dan Gambar (2.3).



Gambar 2.2 Graf Komplit dalam permasalahan VRP



Gambar 2.3 Contoh salah satu penyelesaian permasalahan VRP (Ombuki *et al*, 2006)

Vehicle Routing Problem (VRP) memiliki banyak variasi permasalahan antara lain sebagai berikut (Toth dan Vigo, 2002) :

- a. *Capacitated Vehicle Routing Problem* (CVRP), merupakan permasalahan VRP yang ditambah dengan kendala kapasitas kendaraan yang terbatas.
- b. *Distance Constrained Vehicle Routing Problem* (DCVRP), merupakan permasalahan VRP yang ditambah dengan kendala batasan jarak rute yang ditempuh
- c. *Vehicle Routing Problem with Time Windows* (VRPTW), merupakan permasalahan VRP dimana setiap pelanggan mempunyai batasan interval waktu pelayanan
- d. *Vehicle Routing Problem with Pick Up and Delivery* (VRPPD), merupakan permasalahan VRP dimana setiap pelanggan mendapatkan pelayanan campuran, yaitu pengiriman dan pengambilan barang dalam satu rute
- e. *Vehicle Routing Problem with Backhauls* (VRPB), merupakan permasalahan VRP dengan aturan setiap pengambilan baru dapat dilakukan setelah semua pengiriman selesai.

2.2.1 *Vehicle Routing Problem with Time Windows* (VRPTW)

Vehicle Routing Problem with Time Windows (VRPTW) merupakan permasalahan optimasi untuk menemukan rute dengan meminimalkan biaya untuk sejumlah kendaraan dengan batasan waktu tertentu, baik dari *customer* maupun perusahaannya. Selain batasan waktu, batasan kapasitas armada juga menjadi batas yang perlu diperhatikan. Kendaraan melakukan pengiriman dari depot awal (pusat) kemudian berakhir pada depot awal (pusat) tersebut. Kendaraan juga tidak boleh melanggar batasan waktu yang telah ditetapkan (Yansyah, 2016).

Dalam permasalahan VRPTW, sebuah kendaraan tidak diperbolehkan tiba di pelanggan setelah waktu terakhir untuk memulai layanan. Tetapi, apabila kendaraan tiba terlalu awal di pelanggan, maka diperbolehkan untuk menunggu sampai batas waktu paling awal memulai layanan (Lenstra dan Kan, 1981). *Vehicle Routing Problem With Time Windows* (VRPTW) merupakan turunan dari *Capacitated Vehicle Routing Problem* (CVRP) dengan tambahan kendala yaitu

time windows yang diasosiasikan pada setiap *customer* dan pusat depot. *Time windows* terdiri dari waktu buka tutup depot, waktu buka tutup toko (pelanggan), dan batas waktu perjalanan. Dalam kasus VRPTW juga diperkenalkan istilah *service time* yang mempresentasikan waktu yang dibutuhkan untuk melayani suatu *customer*.

Formula untuk permasalahan VRPTW ini dibentuk dalam model matematis yang mempunyai fungsi tujuan meminimalkan total jarak tempuh dengan diikuti beberapa kendala. Berikut merupakan model matematisnya, jika Z merupakan fungsi tujuan, maka :

$$Z = \min \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} \sum_{k=1}^K c_{ij} x_{ij}^k \quad (2.1)$$

dengan kendala,

$$\sum_{i=1}^N q_i \sum_{j=0}^{N+1} x_{ij}^k \leq Q, \text{ untuk } k = 1, 2, \dots, K \quad (2.2)$$

$$\sum_{k=1}^K \sum_{j=0}^{N+1} x_{ij}^k = 1, \text{ untuk } i = 1, 2, \dots, N \quad (2.3)$$

$$t_0 = 0 \quad (2.4)$$

$$\sum_{i=1}^N t_i + \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} t_{ij}^k \leq T_w \quad (2.5)$$

$$t_{ij}^k = \frac{c_{ij}}{v_{ij}} \quad (2.6)$$

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

$$\sum_{j=0}^{N+1} x_{0j}^k = 1, \text{ untuk } k = 1, 2, \dots, K \quad (2.8)$$

$$\sum_{i=0}^{N+1} x_{ih}^k - \sum_{j=0}^{N+1} x_{hj}^k = 0, \text{ untuk } h = 1, 2, \dots, N; k = 1, 2, \dots, K \quad (2.9)$$

$$\sum_{i=0}^{N+1} x_{i,N+1}^k = 1, \text{ untuk } k = 1, 2, \dots, K \quad (2.10)$$

Dimana x_{ij}^k merupakan variabel keputusan, yaitu :

$$x_{ij}^k = \begin{cases} 1, & \text{jika setiap kendaraan } k \text{ bergerak dari titik } i \text{ ke titik } j \\ 0, & \text{jika setiap kendaraan } k \text{ tidak bergerak dari titik } i \text{ ke titik } j \end{cases}$$

Keterangan:

N = jumlah *customer*

K = jumlah kendaraan

c_{ij} = jarak antara pelanggan i ke j

q_i = jumlah permintaan pelanggan i

Q = kapasitas maksimum kendaraan

t_i = waktu pelayanan pada pelanggan i

t_{ij}^k = waktu perjalanan dari pelanggan i ke pelanggan j

v_{ij} = kecepatan kendaraan dari pelanggan i ke pelanggan j

T_w = total waktu operasional

b_i = waktu buka toko

e_i = waktu tutup toko

(Xu *et al*, 2015).

Persamaan (2.1) menjelaskan fungsi permasalahan untuk kasus VRPTW yaitu total jarak tempuh minimal. Persamaan (2.2) menunjukkan total permintaan *customer* tidak boleh melebihi kapasitas kendaraan. Persamaan (2.3) menunjukkan setiap *customer*nya dikunjungi sekali oleh sebuah kendaraan. Persamaan (2.4) menunjukkan waktu awal berangkat. Persamaan (2.5) menunjukkan waktu perjalanan tidak boleh melebihi batas total waktu operasional. Persamaan (2.6) menunjukkan waktu perjalanan yang ditempuh bergantung pada kecepatan yang ditempuh. Persamaan (2.7) menunjukkan batas waktu pelayanan tidak boleh melebihi antara waktu buka dan waktu tutup dari pelanggan. Persamaan (2.8) sampai (2.10) menunjukkan setiap kendaraan berangkat dari depot ($N = 0$), dan setelah melayani pelanggan, kendaraan tersebut akan pergi, serta pada akhirnya kendaraan tersebut akan kembali ke depot ($N + 1$). Depot ($N = 0$) merupakan depot yang sama dengan depot ($N + 1$) (Kallehauge *et al*, 2001).

2.2.2 Biobjective Vehicle Routing Problem with Time Windows

Permasalahan VRPTW secara umum mempunyai fungsi tujuan yaitu untuk mencari total jarak tempuh minimal dengan batasan waktu yang telah disediakan dari *customer* maupun perusahaannya. Permasalahan *Vehicle Routing Problem with Time Windows* (VRPTW) sebagai *Biobjective Optimization Problem* merupakan bentuk spesifikasi dari permasalahan *Multiobjective VRPTW* dimana lebih dikhususkan untuk menyelesaikan dua fungsi tujuan yaitu meminimalkan kendaraan dan meminimalkan jumlah total jarak tempuh dengan batasan waktu

yang telah disediakan (Najera dan Bullinaria, 2009). Model matematis untuk *Biobjective* VRPTW yaitu (Murata dan Itai, 2005):

untuk meminimalkan jumlah kendaraan

$$z_1 = K \quad (2.11)$$

untuk meminimalkan jumlah total jarak tempuh

$$z_2 = \sum_{k=1}^K \sum_{i=0}^{N+1} \sum_{j=0}^{N+1} c_{ij} x_{ij}^k \quad (2.12)$$

dibatasi kendala-kendala pada persamaan (2.2) sampai (2.10).

Fungsi z_1 merupakan fungsi untuk meminimalkan jumlah kendaraan. Fungsi z_2 merupakan fungsi untuk meminimalkan total jarak tempuh. Fungsi ini dapat menghasilkan solusi minimal apabila jumlah kendaraan yang digunakan minimal dan tidak melebihi batas kurang dari jumlah pelanggan ($K < N$). Apabila ($K = N$) maka setiap kendaraan hanya mengunjungi satu pelanggan.

2.3 Algoritma Genetika

Algoritma genetika diciptakan pertama kali oleh John Holland pada tahun 1975. Setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam bentuk terminologi genetika. Algoritma ini didasari oleh konsep biologi, dan dapat memberikan solusi alternatif atas suatu masalah yang hendak diselesaikan. Algoritma genetika menawarkan suatu solusi pemecahan masalah yang terbaik, dengan memanfaatkan metode seleksi, *crossover*, dan mutasi (Kusumadewi, 2003).

Algoritma genetika merupakan suatu bentuk teknik pencarian secara *stochastic* berdasarkan mekanisme yang ada pada seleksi alam dan genetik secara natural. Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut kromosom. Tahap awal pencarian dalam algoritma genetika dimulai dari himpunan penyelesaian acak (*random*) yang disebut populasi. Sebuah kromosom dibentuk dari komponen-komponen penyusun sebagai gen. Kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan generasi. Setiap kromosom dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang diselesaikan menggunakan ukuran yang disebut nilai *fitness*. Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya disebut proses

seleksi. Kromosom yang mempunyai nilai *fitness* tinggi akan memiliki peluang lebih besar untuk terpilih pada generasi selanjutnya. Generasi baru itu dikenal dengan anak (*offspring*) yang terbentuk dengan cara melakukan perkawinan antara kromosom-kromosom dalam satu generasi yang disebut proses *crossover*. Kromosom dalam populasi yang mengalami proses *crossover* akan ditentukan oleh parameter. Kromosom juga dapat dimodifikasi dengan menggunakan proses mutasi. Jumlah gen yang mengalami proses mutasi ditentukan oleh parameter mutasi. Beberapa generasi tersebut menghasilkan kromosom-kromosom yang nilai gen-gennya konvergen ke suatu nilai tertentu yang merupakan solusi terbaik yang dihasilkan oleh algoritma genetika (Goldberg, 1989).

Beberapa parameter yang berperan dalam proses algoritma genetika yaitu (Gen *et al*, 1997):

a. Jumlah Generasi

Jumlah generasi juga berpengaruh karena semakin besar jumlah generasi maka semakin besar pula kemungkinan didapatkan solusi optimum.

b. Probabilitas *crossover* (P_c)

Nilai probabilitas *crossover* berkisar antara selang $[0,1]$. Semakin besar nilai P_c maka semakin besar kemungkinan ditemukannya solusi optimum. Probabilitas *crossover* (P_c) juga sangat berpengaruh terhadap proses ini karena probabilitas ini digunakan untuk menentukan apakah dapat dilakukan proses *crossover* atau tidak.

c. Probabilitas mutasi (P_m)

Nilai probabilitas mutasi berkisar antara selang $[0,1]$. Probabilitas mutasi menentukan gen yang akan dimutasi. Semakin besar nilai probabilitas mutasi, maka kromosom yang muncul akan semakin banyak yang kemungkinan tidak memiliki potensi dalam pencapaian solusi optimum.

Langkah-langkah dalam penerapan algoritma genetika sebagai berikut :

a. Teknik Pengkodean

Teknik pengkodean ini meliputi pengkodean gen dari kromosom yang mewakili posisi dalam sebuah rangkaian. Gen dapat direpresentasikan dalam bentuk *string bit*, pohon, *array*, bilangan *real*, daftar aturan, elemen

permutasi, elemen program dan sebagainya. Dalam kasus *Biobjective VRPTW* menggunakan pengkodean permutasi dengan representasi gen sebagai angka yang mewakili toko yang dikunjungi.

b. Inisialisasi Populasi Awal

Membangkitkan sejumlah kromosom (sesuai dengan ukuran populasi) untuk dijadikan anggota populasi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang mempresentasi solusi yang diinginkan. Inisialisasi kromosom dilakukan secara acak, namun harus tetap mempertimbangkan domain solusi dan kendala yang ada

c. Evaluasi Nilai *Fitness*

Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dengan menggunakan persamaan :

$$Fitness = \frac{1}{\alpha \cdot K + \beta \sum_{k \in K} D_k} \quad (2.13)$$

$$D_k = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k$$

Keterangan :

α = biaya supir

β = biaya bahan bakar

K = jumlah kendaraan

D_k = jumlah total jarak tempuh

(Ombuki et al, 2006).

Setiap kromosom akan dihitung nilai *fitness*-nya , kemudian mengurutkannya berdasarkan fungsi *fitness* dari terkecil ke terbesar.

d. Proses Seleksi

Seleksi merupakan pemilihan dua kromosom untuk dijadikan sebagai induk yang dilakukan secara proporsional sesuai dengan nilai *fitness*-nya. Masing-masing individu akan diberikan probabilitas setiap individu tergantung dari nilai *fitness* dirinya sendiri dengan nilai *fitness* semua individu dalam seleksi tersebut. Nilai *fitness* ini yang akan digunakan pada tahap selanjutnya. Proses

seleksi yang digunakan dalam kasus ini akan menggunakan metode *Roulette Wheel*.

Metode *Roulette Wheel* merupakan metode seleksi berdasarkan kualitas individual. Semakin berkualitas individu semakin besar kemungkinan individu tersebut terpilih untuk menjadi anggota pada populasi yang baru. Langkah-langkah menerapkan metode ini sebagai berikut:

- 1) Menghitung nilai *fitness* dari masing-masing individu.
- 2) Menghitung total *fitness* dari semua individu.
- 3) Menghitung probabilitas setiap individu (P_k) tersebut dengan menggunakan persamaan:

$$P_k = \frac{f(i)}{\sum f(i)} \quad (2.14)$$

dimana P_k merupakan probabilitas individu ke- i dan $f(i)$ merupakan nilai *fitness* setiap individu i .

- 4) Menghitung probabilitas kumulatif (q_k), dengan

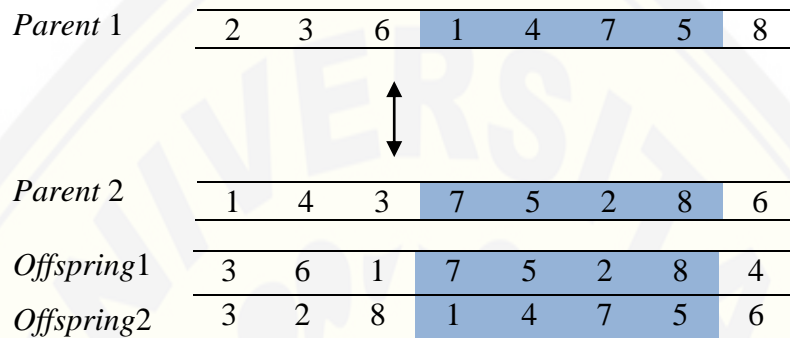
$$q_k = q_{k-1} + p_k \text{ dan } q_0 = 0 \quad (2.15)$$

- 5) Membangkitkan nilai *random* (r_i), $0 \leq r_i \leq 1$ sebanyak jumlah induk yang akan dibangkitkan
- 6) Memilih kromosom untuk dijadikan *parent* pada proses *crossover* dengan syarat $q_{k-1} \leq r_i \leq q_k$, dengan $k = 1, 2, \dots, N$ dan $i = 1, 2, \dots, N$. Semakin besar selang, kemungkinan nilai *fitness* tertinggi pada selang itu semakin besar.

e. Proses *Crossover*

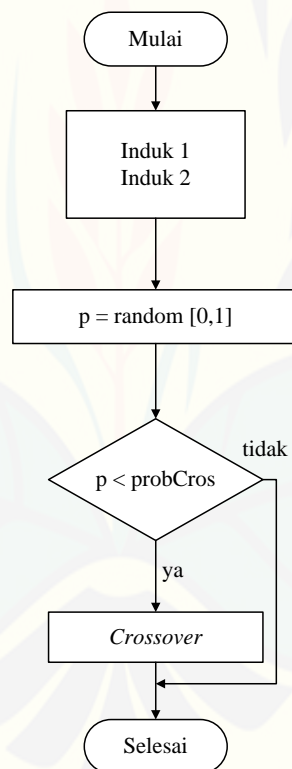
Proses ini merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Teknik ini dilakukan dengan melakukan pertukaran gen dari dua induk kromosom. Proses *crossover* akan terjadi pada sepasang kromosom jika suatu bilangan yang dibangkitkan secara random (r), $0 \leq r \leq 1$, nilainya kurang dari atau sama dengan P_c . Terdapat beberapa tipe proses *crossover* yaitu *one cut point crossover*, *order based crossover*, dan *two point order crossover*. Dalam kasus ini, proses *crossover* yang digunakan yaitu tipe *two point order*

crossover. Metode *two point order crossover* dapat menjadi bagian awal dan akhir dari kromosom dan hanya menukar bagian tengahnya. Dibawah ini merupakan contoh tipe *two point order crossover* (Gambar 2.4).



Gambar 2.4 Contoh *two-point order crossover*

Gambar 2.5 di bawah ini merupakan alur tahapan proses *crossover*:



Gambar 2.5 *Flowchart* alur proses *crossover*

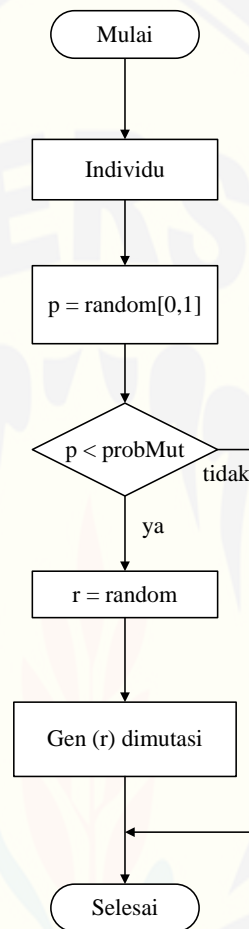
f. Proses Mutasi

Proses mutasi akan terjadi pada suatu gen, jika suatu bilangan yang dibangkitkan secara random (r), $0 \leq r \leq 1$, nilainya kurang dari atau sama dengan probabilitas mutasi (P_m). Proses ini dilakukan pada hasil crossover dan setelah proses *crossover* dengan memilih kromosom yang akan dimutasi yang diambil dari hasil crossover dan kromosom terbaik. Mutasi ini dilakukan dengan cara membangkitkan random sebanyak jumlah ukuran data. Kemudian, menentukan titik gen untuk dilakukan proses mutasi pada kromosom tersebut dengan cara mengevaluasi apakah nilai random dari gen tersebut kurang dari P_m atau tidak. Banyaknya gen yang akan mengalami mutasi dihitung berdasarkan probabilitas mutasi yang telah ditentukan. Proses mutasi yang digunakan dalam kasus ini yaitu *swap mutation* (menukar). Dibawah ini merupakan contoh metode *slide mutation* (Gambar 2.6).

<i>Parent</i>	1	2	3	4	5	6	7	8
<i>Offspring1</i>	1	7	8	4	5	6	2	8

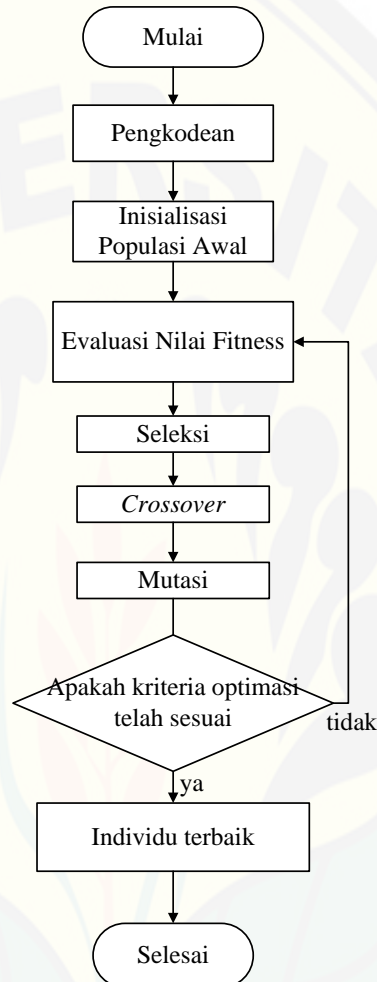
Gambar 2.6 Contoh *Swap Mutation*

Gambar 2.7 Di bawah ini merupakan alur tahapan proses mutasi:



Gambar 2.7 *Flowchart* alur proses mutasi

Proses yang telah dijelaskan di atas dapat dilakukan dalam proses penerapan algoritma dengan tahapan alur sebagai berikut (Gambar 2.8).



Gambar 2.8 *Flowchart* alur proses algoritma genetika

2.4 Algoritma *Cat Swarm Optimization* (CSO)

Algoritma *Cat Swarm Optimization* (CSO) pertama kali ditemukan oleh Shu-Chuan Chu, Pei-wei Tsai, dan Jeng-Shyang Pan pada tahun 2006. Algoritma ini pada umumnya diperoleh dari perilaku sekelompok kucing. Namun, kucing yang dimaksud dalam algoritma ini adalah kelas binatang dari *felidae*. Kucing sendiri mempunyai berbagai spesies antara lain *lion*, *tiger*, *leopard*, kucing dan sebagainya.

Beberapa spesies kucing ini mempunyai lingkungan yang berbeda, tetapi binatang dalam kelas *felidae* ini memiliki perilaku yang sama. Keterampilan

berburu pada jenis kucing ini menjamin pasokan makanan dan kelangsungan hidup spesiesnya. Kucing ini mempunyai keterampilan berburu dan rasa ingin tahu yang besar untuk menangkap atau mencari mangsanya. Kucing juga tampak seolah tidak aktif (diam) ketika mereka terjaga. Tetapi, dalam kondisi istirahat kucing juga sebenarnya mempunyai tingkat kewaspadaan yang tinggi artinya kucing akan mengamati dengan seksama dan mencari apabila terdapat mangsa yang dapat diburu. Matanya yang lebih besar dapat mengamati lingkungan sekitarnya meskipun sedang beristirahat atau bermalas-malasan tidur. Sergapan akan objek buruannya dilakukan setelah kucing tersebut menemukan mangsanya. Artinya pada saat kucing melakukan pengamatan, dia sedang melakukan kalkulasi langkah apa yang dilakukannya kelak secara efektif. Kucing menyusun sebuah strategi untuk menangkap mangsanya dengan cara yang singkat dan efektif. Oleh karena itu, algoritma CSO mempunyai dua model dalam menyelesaikan permasalahan optimisasi yaitu seeking mode dan tracing mode (Chu *et al*, 2015).

2.4.1 *Seeking Mode*

Model ini digunakan untuk memodelkan situasi kucing yang sedang beristirahat, melihat keadaan sekitar dan berpindah mencari posisi berikutnya. Model seeking ini mempunyai empat faktor penting yaitu *seeking memory pool* (SMP), *seeking range of the selected dimension* (SRD), *counts of dimension to change* (CDC), dan *self-position considering* (SPC) (Bouzidi dan Riffi, 2013).

SMP digunakan untuk mendefinisikan ukuran dari pencarian memori untuk setiap kucing, yang menunjukkan titik yang dicari oleh kucing. Kucing dapat memilih sebuah titik dari kelompok memori berdasarkan aturan yang telah ditetapkan. SRD menyatakan rasio mutatif untuk dimensi yang dipilih. Dalam mode pencarian, jika sebuah dimensi dipilih untuk dimodifikasi atau mutasi, perbedaan antara nilai baru dan lama tidak akan keluar dari jangkauan yang telah didefinisikan oleh SRD. CDC berperan penting dalam model ini karena tahap ini untuk menghitung berapa banyak dimensi yang dimodifikasi atau mutasi. SPC merupakan variabel Boolean, yang menentukan titik dimana posisi kucing harus

dipertahankan atau diubah. Adapun tahap-tahap yang harus dilakukan pada metode *seeking mode* yaitu sebagai berikut (Bouzidi dan Riffi, 2013):

- a. Membangkitkan j tiruan dari posisi cat_k , dimana $j = SMP$. Jika posisi cat_k mempunyai nilai fitness tertinggi dengan nilai SPC 1, maka pertahankan posisi kucing sekarang sebagai salah satu kandidat. Sedangkan, jika posisi cat_k mempunyai nilai SPC 0, maka bangkitkan posisi sebanyak jumlah SMP.
- b. Untuk setiap tiruan disesuaikan berdasarkan CDC, kemudian secara acak persentase SRD kurangkan atau jumlahkan dengan nilai sekarang dan ganti yang sebelumnya. Untuk menghitung jumlah dimensi yang akan dimodifikasi dengan persamaan :

$$\text{Jumlah modifikasi} = CDC * n \quad (2.17)$$

- c. Menghitung nilai *fitness* dari semua titik kandidat
- d. Jika semua nilai *fitness* tidak sama persis, maka hitung probabilitas dari setiap titik kandidat dengan persamaan berikut:

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}}, \text{ dimana } 0 < i < j \quad (2.18)$$

jika tujuan dari fungsi fitness untuk menentukan solusi minimum, $FS_b = .FS_{max}$, sebaliknya untuk menentukan solusi maksimum $FS_b = .FS_{min}$

- e. Memilih titik untuk memindahkan titik kandidat dan mengganti posisi cat_k menggunakan seleksi *roulette wheel* dengan mencari probabilitas relatif dari masing-masing kandidat sesuai dengan langkah yang telah dijelaskan dalam proses seleksi *roulette wheel* dalam algoritma genetika. Tetapi, untuk probabilitas relatif menggunakan persamaan (2.19)

$$P_{relatif_i} = \frac{\text{Probabilitas Terpilih}_i}{\text{Total Probabilitas}} \quad (2.19)$$

2.4.2 *Tracing Mode*

Tracing Mode merupakan bagian model yang menggambarkan kasus kucing yang sedang mengikuti jejak target mangsanya. Ketika masuk dalam *tracing mode*, kucing akan bergerak sesuai dengan kecepatannya sendiri setiap dimensi. Tahap yang dilakukan dalam *tracing mode* ini yaitu sebagai berikut (Bouzidi dan Riffi, 2013):

- a. Memperbarui kecepatan untuk setiap dimensi ($v'_{k,d}$) sesuai persamaan berikut:

$$v'_{k,d} = v_{k,d} + r \times c \times (x_{best,d} - x_{k,d}); d = 1, 2, \dots, M \quad (2.20)$$

Keterangan :

$x_{k,d}$ = posisi cat_k

$x_{best,d}$ = posisi kucing yang memiliki nilai fitness terbaik

c_1 = konstanta

r_1 = nilai acak dalam rentang [0,1]

- b. Memperbarui posisi cat_k sesuai persamaan (2.21)

$$x'_{k,d} = x_{k,d} + v_{k,d} \quad (2.21)$$

Penggabungan dua model tersebut dapat dilakukan dengan mendefinisikan *mixture ratio* (MR) atau rasio campuran. MR yang digunakan juga harus bernilai kecil Sehingga, proses algoritma CSO dapat dilakukan dengan tahapan sebagai berikut (Chu *et al*, 2006) :

- Bangkitkan N kucing dalam proses
- Sebarkan kucing secara acak ke dalam ruang solusi M -dimensi dan pilih nilai secara acak yang berada dalam kisaran kecepatan maksimum untuk menjadi kecepatan dari setiap kucing. Kemudian, pilih jumlah kucing sembarang lalu masukkan dalam metode *tracking mode* sesuai MR dan lainnya masukkan dalam metode *seeking mode*

Untuk jumlah *tracing* menggunakan persamaan :

$$\text{Jumlah tracing} = MR * m \quad (2.22)$$

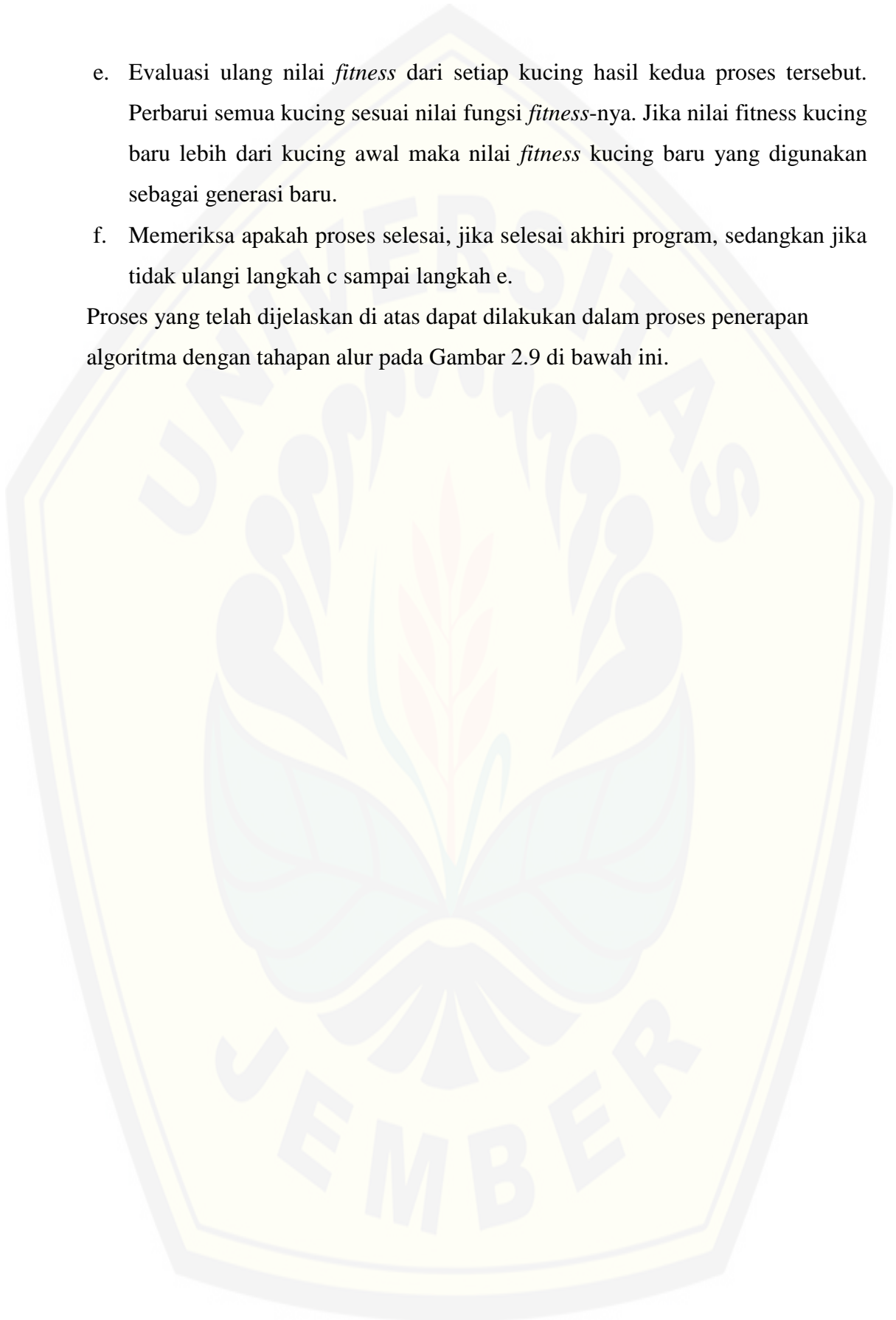
Untuk jumlah *seeking* menggunakan persamaan :

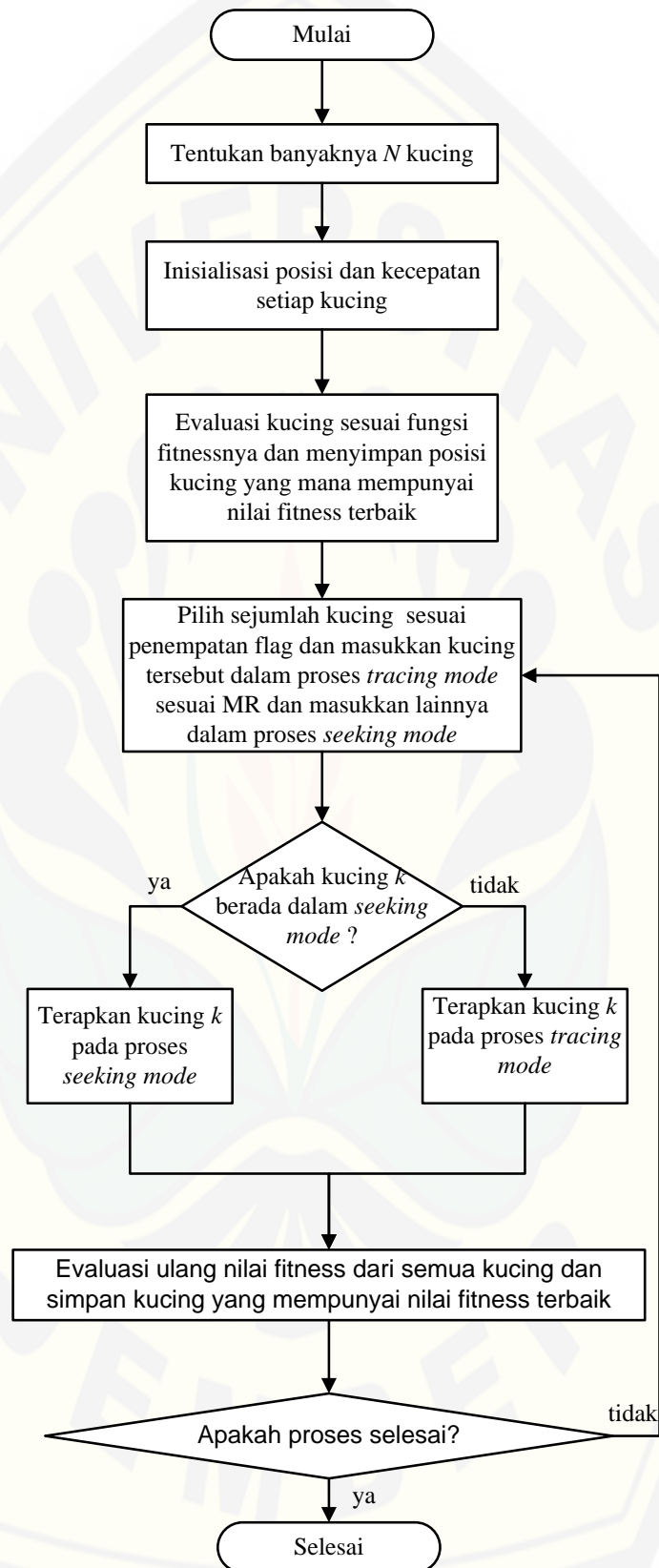
$$\text{Jumlah seeking} = m - \text{jumlah tracing} \quad (2.23)$$

- Evaluasi nilai *fitness* setiap kucing dengan menerapkan posisi kucing dalam fungsi *fitness* yang mewakili kriteria tujuan, dan simpan kucing terbaik dalam memori.
- Pindahkan kucing berdasarkan kelompok mereka, jika kucing dalam *seeking mode* maka terapkan kucing pada proses *seeking mode*, serta begitu juga sebaliknya.

- e. Evaluasi ulang nilai *fitness* dari setiap kucing hasil kedua proses tersebut. Perbarui semua kucing sesuai nilai fungsi *fitness*-nya. Jika nilai *fitness* kucing baru lebih dari kucing awal maka nilai *fitness* kucing baru yang digunakan sebagai generasi baru.
- f. Memeriksa apakah proses selesai, jika selesai akhiri program, sedangkan jika tidak ulangi langkah c sampai langkah e.

Proses yang telah dijelaskan di atas dapat dilakukan dalam proses penerapan algoritma dengan tahapan alur pada Gambar 2.9 di bawah ini.





Gambar 2.9 Flowchart alur proses algoritma CSO

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini yaitu data yang diambil dari skripsi Faridah (2017). Skripsi tersebut menggunakan data dari perusahaan air minum kemasan PT. Tujuh Impian Bersama AMDK (*Air Minum dalam Kemasan*) Al-Qodiri Jember. Data yang digunakan adalah sebagai berikut.

- a. Terdapat data distribusi 32 dan 57 pelanggan.
- b. Jumlah permintaan barang dan waktu operasional depot dan setiap toko untuk 32 pelanggan (lihat Lampiran A1) dan 57 pelanggan (lihat Lampiran A4)
- c. Jarak antara depo dengan toko 32 pelanggan (lihat Lampiran A2) dan 57 pelanggan (lihat Lampiran A5)
- d. Kecepatan rata-rata kendaraan depo ke setiap toko dan jarak antar toko untuk 32 pelanggan (lihat Lampiran A3) dan 57 pelanggan (lihat Lampiran A6)
- e. Waktu pelayanan setiap satu kali angkut per karton dari kendaraan ke toko yaitu 20 detik dengan penambahan konstanta untuk setiap angkut barang yaitu 2 menit
- f. Setiap kendaraan yang disediakan memiliki kapasitas 600 karton
- g. Waktu operasional depo dalam proses pengiriman selama 10 jam yaitu pada pukul 07.00 – 17.00
- h. Biaya bahan bakar minyak diasumsikan per kilometer sebesar Rp 650,00
- i. Upah sopir setiap pengiriman sebesar Rp 200.000,00

3.2 Prosedur Penelitian

Adapun tahap-tahap yang dilakukan dalam penelitian ini sebagai berikut:

- a. Studi Literatur

Tahap yang dilakukan terlebih dahulu pada penelitian ini yaitu melakukan studi literatur dari berbagai referensi seperti buku, jurnal, artikel, dan lain-lain mengenai algoritma genetika, algoritma *Cat Swarm Optimisation* (CSO),

serta permasalahan tentang *Biobjective Vehicle Routing Problem with Time Windows*.

b. Mengumpulkan Data

Data yang digunakan dalam penelitian ini merupakan data yang diambil dari skripsi Faridah (2017). Skripsi tersebut menggunakan data dari perusahaan air minum kemasan PT. Tujuh Impian Bersama AMDK (*Air Minum dalam Kemasan*) Al-Qodiri Jember dengan penambahan data *time windows*.

c. Penerapan Algoritma Genetika dan Algoritma CSO

Pada tahap ini, langkah-langkah untuk menerapkan algoritma genetika dan CSO akan disesuaikan dengan permasalahan *Biobjective CVRP* yang diselesaikan.

1) Langkah-langkah penerapan algoritma genetika

a) Inisialisasi Parameter

Parameter yang digunakan berupa jumlah toko, jumlah permintaan barang, jumlah populasi, dan jumlah generasi.

b) Pembangkitan Kromosom Awal

Pembangkitan kromosom dilakukan secara acak sejumlah populasi

c) Evaluasi Fitness

Setiap kromosom yang telah dibangkitkan sebelumnya berupa TSP diubah dalam bentuk VRPTW dengan mempertimbangkan beberapa kendala yang ada. Kemudian, setiap kromosom dievaluasi dengan cara menghitung nilai fitness dengan menggunakan persamaan (2.13) dan mengurutkannya dari terkecil ke terbesar.

d) Proses Seleksi

Memilih kromosom yang akan dijadikan sebagai induk untuk dilakukan proses *crossover*. Metode seleksi yang digunakan dalam penelitian ini yaitu metode *Roulette Wheel*.

e) Proses *Crossover*

Proses *crossover* dilakukan dengan melakukan pertukaran gen dari dua induk kromosom. Metode *crossover* yang digunakan dalam penelitian ini yaitu metode *two point order crossover*.

f) Proses Mutasi

Proses mutasi dilakukan dengan memilih kromosom yang akan dimutasi yang diambil dari kromosom terbaik dan hasil *crossover*. Metode mutasi yang digunakan dalam penelitian ini yaitu metode *slide mutation*.

g) Pengecekan Iterasi

Proses akan berhenti apabila iterasi yang dilakukan sudah mencapai iterasi maksimal.

2) Langkah-langkah penerapan algoritma CSO

a) Inisialisasi Parameter

Menentukan parameter yang digunakan dalam algoritma CSO yaitu jumlah populasi (m), jumlah titik yang dikunjungi (n), SMP, CDC, MR, c .

b) Pembangkitan populasi awal

Setiap kucing merupakan representasi dari satu set solusi yang terdiri atas n dimensi. Untuk pembangkitan populasi awal dibuat dalam bentuk bilangan acak yang mempunyai nilai dengan selang $[0,1]$.

c) Pembangkitan kecepatan awal

Setiap dimensi dari setiap kucing juga memiliki inisialisasi lain yaitu kecepatan awal. Pembangkitan kecepatan awal dibuat dalam bentuk bilangan acak yang mempunyai nilai dengan selang $[0,1]$.

d) Representasi Permutasi

Langkah ini bertujuan untuk menghitung nilai fungsi tujuan dari setiap solusi awal yang telah dibangkitkan. Tahap ini dilakukan dengan cara mengubah bilangan *real* ke bentuk permutasi yang diurutkan dari terkecil ke terbesar.

e) Evaluasi *Fitness*

Setiap kucing yang telah dibangkitkan sebelumnya berupa TSP diubah dalam bentuk VRPTW dengan mempertimbangkan beberapa kendala yang ada. Kemudian, setiap kucing dievaluasi dengan cara

menghitung nilai fitness dengan menggunakan persamaan (2.13) dan mengurutkannya dari terbesar ke terkecil.

f) Menghitung *Self Position Considering* (SPC)

Penempatan *Self Position Considering* (SPC) dilakukan dengan cara penempatan label $SPC=1$ terhadap kucing yang mempunyai nilai *fitness* tertinggi dan sisanya mempunyai label $SPC=0$.

g) Menentukan Flag

Jumlah kucing yang dimasukkan ke dalam masing-masing proses dengan nilai MR.

h) Proses Tracing Mode

Setiap kucing akan mengubah representasi kecepatan yang lama menjadi kecepatan baru menggunakan persamaan (2.21) dan (2.22).

i) Proses Seeking Mode

Proses ini dilakukan untuk melakukan modifikasi terhadap set solusi dari tiruan kucing. Solusi tiruan kucing yang telah dimodifikasi tersebut dipilih untuk dijadikan solusi kucing terbaik dengan menggunakan metode *Roulette Wheel*.

j) Pengecekan Iterasi

Proses akan berhenti apabila iterasi yang dilakukan sudah mencapai iterasi maksimal.

d. Pembuatan Program

Peneliti akan membuat program untuk menyelesaikan permasalahan *Biobjective Vehicle Routing Problem With Time Windows* dengan menggunakan kedua algoritma tersebut. Program yang digunakan merupakan program *software* MATLAB dan dibuat dalam bentuk GUI

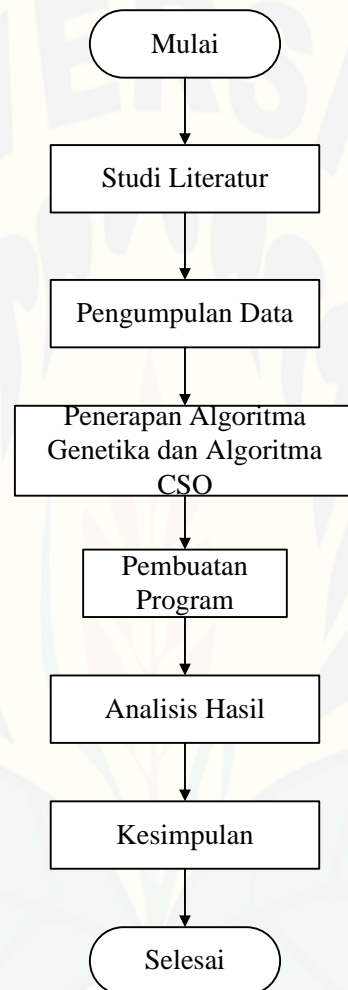
e. Analisis Hasil

Data penelitian tersebut akan dilakukan percobaan menggunakan program yang telah dibuat sebelumnya. Hasil percobaan yang diperoleh akan dianalisis untuk membandingkan solusi total kendaraan dan total jarak tempuh dari kedua algoritma tersebut dan biaya yang dihitung dari jumlah kendaraan dan total jarak tempuh.

f. Kesimpulan

Hasil kedua algoritma tersebut akan ditarik kesimpulan untuk menentukan solusi yang lebih baik.

Skema alur penelitian yang dapat dilakukan dapat dilihat pada Gambar 3.1



Gambar 3.1 Skema Alur Penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada tugas akhir dapat disimpulkan sebagai berikut:

- a. Algoritma Genetika dan algoritma CSO dapat diterapkan dalam kasus *Biobjective VRPTW*. Kedua algoritma tersebut dapat menyelesaikan data berukuran kecil maupun besar yaitu 32 toko dan 57 toko. Algoritma CSO dapat menghasilkan solusi yang lebih baik dibandingkan algoritma genetika. Algoritma genetika menghasilkan solusi terbaik yaitu sebanyak 3 kendaraan dengan jumlah jarak tempuh 435,8 km untuk data 32 toko, dan 10 kendaraan dengan jumlah jarak tempuh 716,6 km untuk data 57 toko. Sedangkan, pada algoritma CSO menghasilkan solusi terbaik yaitu sebanyak 3 kendaraan dengan jumlah jarak tempuh 366 km untuk data 32 toko, dan 9 kendaraan dengan jumlah jarak tempuh 706,5 km untuk data 57 toko.
- b. Berdasarkan solusi yang diperoleh dari kedua algoritma tersebut diperoleh bahwa algoritma genetika menghasilkan biaya minimum yaitu sebesar Rp 817.900,00 untuk data 32 toko, dan Rp 2.358.300,00 untuk data 57 toko. Sedangkan, pada algoritma CSO menghasilkan biaya minimum yaitu sebesar Rp 777.500,00 untuk data 32 toko, dan Rp 2.153.250,00 untuk data 57 toko. Algoritma CSO lebih efektif dalam penyelesaian permasalahan *Biobjective VRPTW* dibandingkan dengan algoritma genetika, karena dapat menemukan solusi yang lebih minimum.

5.2 Saran

Pada penelitian ini kendaraan yang digunakan memiliki kapasitas yang sama, namun pada kehidupan nyata jarang sekali perusahaan yang menggunakan satu jenis kendaraan. Sehingga, untuk penelitian selanjutnya diharapkan dapat menyelesaikan permasalahan *Biobjective Vehicle Routing Problem with Time Windows* dengan kapasitas kendaraan yang berbeda. Selain itu, pada penelitian ini menggunakan permasalahan *Biobjective Vehicle Routing Problem with Time*

Windows, untuk penelitian lebih lanjut diharapkan dapat mengembangkan pada permasalahan yang lebih kompleks seperti *Multiobjective Vehicle Routing Problem with Time Windows*.



DAFTAR PUSTAKA

- Bouizidi, A dan Riffi, M.E. 2013. Discrete Cat Swarm Optimization to Resolve the Traveling Salesman Problem. *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*. 3(9): 13-18.
- Chartrand, G dan Oellermann, O.R. 1993. *Applied and Algorithmic Graph Theory*. New York: McGraw Hill, Inc.
- Chu, S.C., Tsai, P.W., dan Pan, J.S. 2006. Cat Swarm Optimization. *Pacific Rim International Conference on Artificial Intelligence*. 7-11 Agustus 2006. Springer: 854-858.
- Dantzig, G.B dan Ramser, J.H. 1959. The Truck Dispatching Problem. *INFORMS*. 6: 80-91.
- Faridah, Y.N. 2017. Implementasi Algoritma Codeq dan Modifikasi Algoritma Codeq pada *Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
- Gen, M dan Cheng, R. 1997. *Genetic Algorithms and Engineering design*. New York: John Wiley & Sons.
- Goldberg, D.E dan Holland, J.H. 1988. Genetic Algorithms and Machine Learning. *Machine Learning*. 3(2): 95-99.
- Kallehauge, B., J.K. Larsen, dan O.B. Madsen. 2001. Lagrangian Duality Applied on Vehicle Routing with Time Windows. *Computers & Operations Research*. 5(33):1464-1487.
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Lenstra, J.K. dan G.H.A Kan. 1981. Complexity of Vehicle Routing and Scheduling Problems. *Combinatorics in Operations Research*. 11: 221-227.
- Munir, R. 2005. *Matematika Diskrit*. Bandung: Informatika.

- Murata, T. dan R. Itai. 2005. Multi-objective Vehicle Routing Problem Using Two-fold EMO Algorithms to Enhance Solution Similarity on Non-dominated Solutions. *Evolutionary multi-criterion optimization*: 885-896.
- Najera, A.G dan J.A. Bullinaria. 2009. Bi-objective Optimization for the Vehicle Routing Problem with Time Windows: Using Route Similarity to Enhance Performance. *Evolutionary Multi-Criterion Optimization*: 275-289.
- Ombuki, B., B.J. Ross, dan F.Hanshar. 2006. Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. *Applied Intelligence*. 24 (1): 17-30.
- Taqwiya, U.A.C. 2017. Aplikasi Algoritma Genetika pada Kasus *Vehicle Routing Problem with Time Windows*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
- Toth, P. dan D. Vigo. 2002. *The Vehicle Routing Problem*. Philadelphia: Society for Industrial and Applied Mathematics.
- Vitaloka, A.R.D. 2016. Penerapan *Cat Swarm Optimization (CSO)* dan *Particle Swarm Optimization (PSO)* dalam Penyelesaian *Travelling Salesman Problem*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
- Yansyah, M.D. 2016. Aplikasi Algoritma *Cheapest Insertion Heuristic (CIH)* with *Saving Method* dalam kasus *Vehicle Routing Problem with Time Windows* Pada Pengangkutan Sampah. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
- Xu, S.H., J.P. Liu, Zhang, F.H. Zhang, L. Wang, dan L.J. Sun. 2015. A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows. *Sensors*. 15(9): 21033-21053.

LAMPIRAN

Lampiran A. Data Jumlah Permintaan, Waktu Oprasional dan Jarak dari Depo ke Setiap Toko dan Antar Toko Per Hari

A1. Data Jumlah Permintaan Barang dan Waktu Operasional dengan depo dan 32 Toko

Kode Toko	Alamat Toko	Jumlah Permintaan	Waktu Operasional		Kode Toko	Alamat Toko	Jumlah Permintaan	Waktu Operasional	
			a_i	b_i				a_i	b_i
0	Depot	0	07.00	17.00	16	Glagah Wero	13	11.00	18.30
1	Gebang	115	08.30	15.30	17	Arjasa	103	15.00	21.00
2	Slawu	5	10.00	17.00	18	Candijati	13	12.30	19.30
3	Kaliwates	48	11.30	20.00	19	Darsono	22	15.00	21.00
4	Kepatihan	91	10.30	20.00	20	Biting	13	07.30	14.00
5	Tegal Besar	11	13.30	20.00	21	Karang Anyar	117	12.00	19.00
6	Sempusari	5	13.30	20.30	22	Ledokombo	2	14.30	21.00
7	Jember Kidul	10	09.30	18.00	23	Suren	2	09.00	17.30
8	Kebonsari	5	10.30	19.30	24	Sumber Lesung	2	14.30	21.00
9	Tegal Gede	56	12.30	20.00	25	Pace	17	14.00	21.00
10	Sumber Sari	64	11.30	20.00	26	Silo	44	08.30	17.00
11	Antirogo	2	08.00	14.30	27	Sumberjati	4	13.00	20.30
12	Dukuh Mencek	2	07.00	15.30	28	Balung Lor	160	08.30	17.00
13	Kalisat	19	08.00	17.00	29	Balung Kulon	1	08.00	17.30
14	Ajung-Kalisat	19	08.30	15.30	30	Suger Kidul	6	09.30	16.00
15	Sumber Ketempah	2	10.30	18.00	31	Jelbuk	121	09.30	17.30
					32	Bangsalsari	11	10.00	16.30

A2. Jarak Antara Depo Dengan Toko dan Jarak Antar Toko Untuk 32 Pelanggan (km)

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 (depo)	0	1,2	2,1	6,4	3,7	6,2	6,8	3,6	5,7	4	4,5	6,5	9,2	14,5	18,9	20,2
1	1,2	0	2,3	5,9	4,3	5,6	6,3	3	6,2	5,2	5,5	7,7	9,4	15,7	20,2	21,5
2	2,1	2,3	0	7,4	5,8	7,2	7,9	4,6	7,8	6,1	6,6	8,6	10,2	16,5	21	22,3
3	6,4	5,9	7,4	0	5,9	2,7	2,4	3,9	4,4	8	7,1	10,5	5,7	18,5	23	24,3
4	3,7	4,3	5,8	5,9	0	5	5,9	2,4	3,3	4,3	3,4	6,9	9,3	14,9	19,4	20,7
5	6,2	5,6	7,2	2,7	5	0	4,7	3,5	3,6	8,4	7,1	10,9	8,1	18,9	23	24,3
6	6,8	6,3	7,9	2,4	5,9	4,7	0	4,3	6,4	8,4	7,5	10,9	3,6	18,9	23,4	24,7
7	3,6	3	4,6	3,9	2,4	3,5	4,3	0	4,8	4,9	4	7,4	6,9	15,5	19,9	21,2
8	5,7	6,2	7,8	4,4	3,3	3,6	6,4	4,8	0	6,2	4,1	8,7	9,8	17,5	19,8	21
9	4	5,2	6,1	8	4,3	8,4	8,4	4,9	6,2	0	4,1	3,1	12,7	12,3	17,1	17,8
10	4,5	5,5	6,6	7,1	3,4	7,1	7,5	4	4,1	4,1	0	4,2	12,5	13,1	18,1	19,1
11	6,5	7,7	8,6	10,5	6,9	10,9	10,9	7,4	8,7	3,1	4,2	0	15,2	9,5	14	25,3
12	9,2	9,4	10,2	5,7	9,3	8,1	3,6	6,9	9,8	12,7	12,5	15,2	0	21,5	26	27,3
13	14,5	15,7	16,5	18,5	14,9	18,9	18,9	15,5	17,5	12,3	13,3	9,5	21,5	0	5,5	6,8
14	18,9	20,2	21	23	19,4	23	23,4	19,9	19,8	17,1	18,1	14	26	5,5	0	2,3
15	20,2	21,5	22,3	24,3	20,7	24,3	24,7	21,2	21	17,8	19,1	15,2	27,3	6,8	2,3	0
16	15,8	17	17,9	19,4	15,8	18,6	20,3	16,8	15,4	13,9	13,7	10,5	22,9	2,8	4,9	6,1
17	11,5	12,9	14,1	15,6	12,4	16	16,4	12,5	15	10,1	11,1	9,2	19	11,2	15,7	16,4
18	10,7	12	12,8	14,8	11,2	15,2	15,2	11,7	13,8	8,9	11,3	7,9	17,8	7,2	11,7	13,5
19	11,5	12,7	12,7	15,6	11,9	15,9	15,9	12,5	14,5	10,4	11,5	9,5	18,5	12,4	16,9	17,5
20	10,1	10,4	12,2	14,3	10,5	14,6	14,5	11,1	13,1	8,2	10,6	7,3	17,1	5,1	9,6	10,2
21	24,9	24,3	25,9	20,4	24	20	19,2	21,6	22,8	27,4	26,3	29,9	19,1	38,5	41,3	42,6
22	27,8	29	29,9	29,8	27,3	29	31,1	28,7	25,7	26,1	24	21,5	33,7	15,2	35,7	13,9
23	25,2	25,7	27,3	26,5	22,8	25,7	27,7	24,3	22,4	22,8	20,2	22,4	30,4	16,6	31,1	14,5
24	25,5	26,7	27,5	27,4	23,8	26,7	28,7	25,3	23,4	23,8	21,7	19,2	31,3	12,9	33,4	11,6
25	36,4	36,9	38,5	37,7	34	36,9	39	35,5	33,6	34	31,9	32,6	41,6	26,3	23,6	24,9
26	42,4	42,9	44,4	43,6	40	42,8	44,9	41,4	39,5	39,9	37,9	38,1	47,5	31,8	29,2	30,4
27	33,5	34	35,5	34,7	31,1	33,9	36	32,5	30,6	31	29	29,2	38,6	24,8	20,3	21,5
28	25	24,4	26	20,5	24,1	22,9	19,3	21,7	24,6	27,5	26,4	30	19,2	38,1	42,5	43,2

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
29	29,3	28,7	30,3	24,8	28,3	27,1	23,5	25,9	28,8	31,7	30,7	34,3	23,5	42,3	46,8	47,4
30	17,8	19	19,8	21,8	18,2	22,2	22,2	18,8	20,8	15,9	18,3	15	24,8	14,4	17,3	16
31	16,2	17,4	18,2	20,2	16,6	20,6	20,6	17,2	19,2	14,3	16,7	13,4	23,2	13,3	17,7	15,6
32	24,1	23,5	25,1	19,6	23,1	22	18,4	20,8	23,7	26,6	25,5	29,1	18,3	37,1	41,6	42,3

Kode Toko	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0 (depo)	15,8	11,5	10,7	11,5	10,1	24,9	27,8	25,2	25,5	36,4	42,4	33,5	25	29,3	17,8	16,2	24,1
1	17	12,9	12	12,7	10,4	24,3	29	25,7	26,7	36,9	42,9	34	24,4	28,7	19	17,4	23,5
2	17,9	14,1	12,8	12,7	12,2	25,9	29,9	27,3	27,5	38,5	44,4	35,5	26	30,3	19,8	18,2	25,1
3	19,4	15,6	14,8	15,6	14,3	20,4	29,8	26,5	27,4	37,7	43,6	34,7	20,5	24,8	21,8	20,2	19,6
4	15,8	12,4	11,2	11,9	10,5	24	27,3	22,8	23,8	34	40	31,1	24,1	28,3	18,2	16,6	23,1
5	18,6	16	15,2	15,9	14,6	20	29	25,7	26,7	36,9	42,8	33,9	22,9	27,1	22,2	20,6	22
6	20,3	16,4	15,2	15,9	14,5	19,2	31,1	27,7	28,7	39	44,9	36	19,3	23,5	22,2	20,6	18,4
7	16,8	12,5	11,7	12,5	11,1	21,6	28,7	24,3	25,3	35,5	41,4	32,5	21,7	25,9	18,8	17,2	20,8
8	15,4	15	13,8	14,5	13,1	22,8	25,7	22,4	23,4	33,6	39,5	30,6	24,6	28,8	20,8	19,2	23,7
9	13,9	10,1	8,9	10,4	8,2	27,4	26,1	22,8	23,8	34	39,9	31	27,5	31,7	15,9	14,3	26,6
10	13,7	11,1	11,3	11,5	10,6	26,3	24	20,2	21,7	31,9	37,9	29	26,4	30,7	18,3	16,7	25,5
11	10,5	9,2	7,9	9,5	7,3	29,9	21,5	22,4	19,2	32,6	38,1	29,2	30	34,3	15	13,4	29,1
12	22,9	19	17,8	18,5	17,1	19,1	33,7	30,4	31,3	41,6	47,5	38,6	19,2	23,5	24,8	23,2	18,3
13	2,8	11,2	7,2	12,4	5,1	38,5	15,2	16,6	12,9	26,3	31,8	24,8	38,1	42,3	14,4	13,3	37,1
14	4,9	15,7	11,7	16,9	9,6	41,3	35,7	31,1	33,4	23,6	29,2	20,3	42,5	46,8	17,7	17,3	41,6
15	6,1	16,4	13,5	17,5	10,2	42,6	13,9	14,5	11,6	24,9	30,4	21,5	43,2	47,4	16	15,6	42,3
16	0	12,5	8,6	13,7	6,4	36,9	12,9	14,9	10,6	23,9	29,4	20,5	38,2	42,4	16,3	14,7	37,3
17	12,5	0	7,5	5,7	6,8	35,5	24,5	26,6	22,2	34,5	41,1	32,2	35,6	39,8	12,2	10,6	34,6
18	8,6	7,5	0	8,7	2,6	34,2	20,6	22,6	18,2	31,6	37,1	28,2	34,3	38,6	8,4	6,8	33,4
19	13,7	5,7	8,7	0	8	35	25,7	27,8	23,4	36,7	42,2	33,3	35,1	39,3	15,7	14,1	34,1
20	6,4	6,8	2,6	8	0	33,6	18,4	20,4	16	29,4	34,9	26	33,7	37,9	10,5	8,9	32,7
21	36,9	35,5	34,2	35	33,6	0	46,1	42,8	43,7	48,5	59,9	51	16,8	16,6	40,5	38,9	33,2
22	12,9	24,5	20,6	25,7	18,4	46,1	0	6,2	4,6	18,9	24,4	10,8	48,5	52,8	26	27,1	47,6

Kode Toko	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
23	14,9	26,6	22,6	27,8	20,4	42,8	6,2	0	7,8	17,9	25,9	17	43,5	47,7	22,4	22	42,6
24	10,6	22,2	18,2	23,4	16	43,7	4,6	7,8	0	19	24,5	14,5	46,2	50,4	23,7	24,4	45,3
25	23,9	34,5	32,6	36,7	29,4	48,5	18,9	17,9	19	0	31,3	22,4	56,7	61	38,5	38,1	55,8
26	29,4	41,1	37,1	42,2	34,9	59,9	24,4	25,9	24,5	31,3	0	23,2	62,7	66,9	44	43,6	61,7
27	20,5	32,2	28,2	33,3	26	51	10,8	17	14,5	22,4	23,2	0	53,8	58	35,1	34,7	52,8
28	38,2	35,6	34,3	35,1	33,7	16,8	48,5	43,5	46,2	56,7	62,7	53,8	0	4,3	40,3	38,7	17,7
29	42,4	39,8	38,6	29,3	37,9	16,6	52,8	47,7	50,4	61	66,9	58	4,3	0	44,1	42,5	17,6
30	16,3	12,2	8,4	15,7	10,5	40,5	26	22,4	23,7	38,5	44	35,1	40,3	44,1	0	10,5	40,4
31	14,7	10,6	6,8	14,1	8,9	38,9	27,1	22	24,4	38,1	43,6	34,7	38,7	42,5	10,5	0	38,8
32	37,3	34,6	33,4	34,1	32,7	33,2	47,6	42,6	45,3	55,8	61,7	52,8	17,7	17,6	40,4	38,8	0

A3. Data Kecepatan Rata-Rata Kendaraan dari Depo ke Setiap Toko dan Antar Toko untuk 32 Pelanggan (km/jam)

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 (depo)	0	26	26	26	27	30	27	25	26	29	27	28	26	34	27	33
1	26	0	29	28	26	30	27	26	25	28	26	26	27	35	34	39
2	26	29	0	26	27	27	29	28	26	29	30	28	29	34	36	38
3	26	28	26	0	30	30	29	26	26	26	28	34	27	28	30	31
4	27	26	27	30	0	30	29	30	28	26	26	27	27	35	27	38
5	30	30	27	30	30	0	27	29	28	29	29	25	26	28	35	31
6	27	27	29	29	29	27	0	27	26	26	28	27	27	31	37	34
7	25	26	28	26	30	29	27	0	25	29	29	30	25	32	26	33
8	26	25	26	26	28	28	26	25	0	30	27	26	29	35	34	30
9	29	28	29	26	26	29	26	29	30	0	27	29	27	35	34	33
10	27	26	30	28	26	29	28	29	27	27	0	29	33	26	34	29
11	28	26	28	34	27	25	27	30	26	29	29	0	33	29	29	37
12	26	27	29	27	27	26	27	25	29	27	33	33	0	35	39	36
13	34	35	34	28	35	28	31	32	35	35	26	29	35	0	28	26

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14	27	34	36	30	27	35	37	26	34	34	34	29	39	28	0	29
15	33	39	38	31	38	31	34	33	30	33	29	37	36	26	29	0
16	25	30	27	34	28	31	33	29	27	29	28	26	33	27	29	29
17	29	28	34	26	34	32	26	25	33	26	35	29	32	34	34	34
18	29	28	27	32	30	29	30	31	28	28	26	29	26	25	26	27
19	35	27	26	31	35	28	31	30	34	35	33	25	28	28	28	32
20	28	31	28	27	26	34	33	28	26	27	29	28	31	30	25	34
21	32	32	33	34	36	29	31	30	40	32	31	39	27	36	45	49
22	31	31	31	38	30	37	40	37	32	39	40	32	37	33	45	29
23	31	33	32	38	33	33	35	34	31	32	37	39	44	26	35	26
24	39	32	33	38	35	38	35	40	36	30	38	27	44	34	40	27
25	43	41	43	40	37	36	38	43	44	44	45	37	42	37	34	34
26	49	45	49	43	41	43	48	44	43	45	41	43	43	36	38	44
27	44	44	39	41	40	40	40	41	44	42	30	38	38	34	37	34
28	32	34	34	36	38	31	31	40	34	37	35	37	27	41	40	44
29	37	36	37	34	31	40	36	39	36	40	37	43	32	46	48	45
30	28	26	33	32	28	34	31	30	37	31	31	26	38	28	27	26
31	33	25	27	35	26	40	32	34	27	26	29	27	37	32	31	30
32	38	31	30	27	35	36	28	38	39	37	36	32	30	39	46	48

Kode Toko	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0 (depo)	25	29	29	35	28	32	31	31	39	43	49	44	32	37	28	33	38
1	30	28	28	27	31	32	31	33	32	41	45	44	34	36	26	25	31
2	27	34	27	26	28	33	31	32	33	43	49	39	34	37	33	27	30
3	34	26	32	31	27	34	38	38	38	40	43	41	36	34	32	35	27
4	28	34	30	35	26	36	30	33	35	37	41	40	38	31	28	26	35
5	31	32	29	28	34	29	37	33	38	36	43	40	31	40	34	40	36
6	33	26	30	31	33	31	40	35	35	38	48	40	31	36	31	32	28

Kode Toko	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
7	29	25	31	30	28	30	37	34	40	43	44	41	40	39	30	34	38
8	27	33	28	34	26	40	32	31	36	44	43	44	34	36	37	27	39
9	29	26	28	35	27	32	39	32	30	44	45	42	37	40	31	26	37
10	28	35	26	33	29	31	40	37	38	45	41	30	35	37	31	29	36
11	26	29	29	25	28	39	32	39	27	37	43	38	37	43	26	27	32
12	33	32	26	28	31	27	37	44	44	42	43	38	27	32	38	37	30
13	27	34	25	28	30	36	33	26	34	37	36	34	41	46	28	32	39
14	29	34	26	28	25	45	45	35	40	34	38	37	40	48	27	31	46
15	29	34	27	32	34	49	29	26	27	34	44	34	44	45	26	30	48
16	0	26	27	35	27	42	33	31	27	30	32	35	38	42	30	31	43
17	26	0	28	27	27	35	34	32	35	43	45	35	38	36	29	26	39
18	27	28	0	26	26	42	38	40	34	37	37	39	42	36	27	26	44
19	35	27	26	0	29	42	30	40	36	36	43	43	37	35	26	28	42
20	27	27	26	29	0	36	30	31	31	35	41	38	40	36	32	29	39
21	42	35	42	42	36	0	40	40	43	49	44	41	28	27	44	42	42
22	33	34	38	30	30	40	0	26	26	35	37	29	41	45	36	38	42
23	31	32	40	40	31	40	26	0	29	27	39	35	45	43	32	31	42
24	27	35	34	36	31	43	26	29	0	31	38	30	43	46	38	32	48
25	30	43	37	36	35	49	35	27	31	0	40	35	46	43	39	44	49
26	32	45	37	43	41	44	37	39	38	40	0	39	46	42	47	44	42
27	35	35	39	43	38	41	29	35	30	35	39	0	44	48	38	42	42
28	38	38	42	37	40	28	41	45	43	46	46	44	0	27	41	38	32
29	42	36	36	35	36	27	45	43	46	43	42	48	27	0	44	44	25
30	30	29	27	26	32	44	36	32	38	39	47	38	41	44	0	27	41
31	31	26	26	28	29	42	38	31	32	44	44	42	38	44	27	0	42
32	43	39	44	42	39	42	42	42	48	49	42	42	32	25	41	42	0

A4. Data Jumlah Permintaan Barang dan Waktu Operasional dengan depo dan 57 Toko

Kode Toko	Alamat Toko	Jumlah Permintaan	Waktu Operasional		Kode Toko	Alamat Toko	Jumlah Permintaan	Waktu Operasional	
			a_i	b_i				a_i	b_i
0	Depot	0	08.00	16.00	30	Pontang	70	11.00	18.30
1	Gebang1	600	09.00	16.00	31	Karang Anyar	117	12.00	20.30
2	Gebang2	254	11.00	19.30	32	Ambulu	600	08.00	19.00
3	Patrang	33	11.30	21.00	33	Tempurejo	2	12.00	15.00
4	Slawu	25	10.30	18.00	34	Ledokombo	51	08.30	19.30
5	Mangli	83	14.00	21.30	35	Suren	2	10.00	15.30
6	Kaliwates	121	09.30	17.30	36	Sumberlesung	2	11.00	17.00
7	Kepatihan1	600	12.00	19.00	37	Garahan	83	14.00	17.0
8	Kepatihan2	518	09.30	18.00	38	Pace	17	12.00	21.00
9	Tegal Besar	23	09.00	16.00	39	Silo	5	12.00	19.30
10	Sempusari	8	10.00	18.30	40	Sumberjati	4	13.30	21.00
11	Jember Kidul	104	13.30	20.00	41	Balung Lor	160	08.00	17.00
12	Wirolegi	10	11.30	18.00	42	Tutul	4	11.00	17.30
13	Karangrejo	22	11.30	20.30	43	Balung Kulon	1	14.00	21.00
14	Kranjingan	33	09.00	16.00	44	Suger Kidul	18	08.00	16.0
15	Kebonsari	27	08.00	14.30	45	Jelbuk	121	12.00	19.00
16	Tegal Gede	112	12.00	19.30	46	Bangsalsari	11	07.30	14.00
17	Sumbersari	271	15.00	21.30	47	Tamansari	5	10.00	18.30
18	Antirogo	2	08.00	15.00	48	Kertosari	10	12.30	20.00
19	Sukorambi	15	08.30	17.30	49	Pakusari	5	08.00	15.00
20	Dukuh Mencek	2	07.00	14.30	50	Sumberpinang	4	13.30	20.30
21	Kalisat	19	13.30	20.00	51	Mangaran	4	12.00	18.30
22	Ajung Kalisat	19	11.00	18.00	52	Ajung	3	14.00	21.00
24	Glagahwero	13	14.00	22.00	53	Karang Kedawung	69	12.00	19.30
25	Gumuksari	52	10.30	18.30	54	Suco	50	10.30	17.00
26	Arjasa	144	11.30	18.30	55	Kemuningsari Kidul	2	12.00	19.30
27	Candijati	13	12.30	19.30	56	Jatimulyo	13	13.30	21.00
28	Darsono	22	07.30	17.00	57	Jenggawah	105	08.00	17.00
29	Biting	13	11.00	20.30					

A5. Jarak Antara Depo dengan Toko dan Jarak Antar Toko Untuk 57 Pelanggan (km)

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12
0 (depo)	0	1,2	1,2	3,5	2,1	10	6,4	3,7	3,7	6,2	6,8	3,6	8,5
1	1,2	0	0	4,6	2,3	8,5	5,9	4,3	4,3	5,6	6,3	3	9,1
2	1,2	0	0	4,6	2,3	8,5	5,9	4,3	4,3	5,6	6,3	3	9,1
3	3,5	4,6	4,6	0	2,6	12,5	9,8	6,2	6,2	9,6	10,2	6,9	11
4	2,1	2,3	2,3	2,6	0	11	7,4	5,8	5,8	7,2	7,9	4,6	10,6
5	10	8,5	8,5	12,5	11	0	4,6	8,5	8,5	7	3,4	6,6	13,3
6	6,4	5,9	5,9	9,8	7,4	4,6	0	5,9	5,9	2,7	2,4	3,9	10,7
7	3,7	4,3	4,3	6,2	5,8	8,5	5,9	0	0	5	5,9	2,4	6,1
8	3,7	4,3	4,3	6,2	5,8	8,5	5,9	0	0	5	5,9	2,4	6,1
9	6,2	5,6	5,6	9,6	7,2	7	2,7	5	5	0	4,7	3,5	9
10	6,8	6,3	6,3	10,2	7,9	3,4	2,4	5,9	5,9	4,7	0	4,3	11,1
11	3,6	3	3	6,9	4,6	6,6	3,9	2,4	2,4	3,5	4,3	0	7,6
12	8,5	9,1	9,1	11	10,6	13,3	10,7	6,1	6,1	9	11,1	7,6	0
13	5,8	8,1	8,1	8,5	7,9	12,3	9,7	5,1	5,1	8	10,1	6,6	3,1
14	8,8	9,3	9,3	11,2	10,8	11,3	7,8	6,4	6,4	7,2	9,8	7,8	4,1
15	5,7	6,2	6,2	8,1	7,8	8,7	4,4	3,3	3,3	3,6	6,4	4,8	5,6
16	4	5,2	5,2	6,7	6,1	10,6	8	4,4	4,4	8,4	8,4	4,9	6,1
17	4,5	5,5	5,5	7,7	6,6	9,7	7,1	4,3	4,3	7,1	7,5	4	4,1
18	6,5	7,7	7,7	8,3	8,6	13,2	10,5	6,9	6,9	10,9	10,9	7,4	5,3
19	8,7	8,7	8,7	8,3	7,5	9,9	9,2	10,5	10,5	11,5	9,6	8,1	15
20	9,2	9,4	9,4	11,7	10,2	2,6	5,7	9,3	9,3	8,1	3,6	6,9	13,7
21	14,5	15,7	15,7	14,4	16,5	21,2	18,5	14,9	14,9	18,9	18,9	15,5	11,6
22	18,9	20,2	20,2	18,9	21	25,7	23	19,4	19,4	23	23,4	19,9	14,6
23	20,2	21,5	21,5	19,6	22,3	27	24,3	20,7	20,7	24,3	24,7	21,2	15,8
24	15,8	17	17	15,8	17,9	22,5	19,4	15,8	15,8	18,6	20,3	16,8	10,1
25	12,4	13,7	13,7	12,4	14,5	19,1	16,5	12,9	12,9	16,9	16,9	13,4	9,9
26	11,5	12,9	12,9	10,4	14,1	18,7	15,6	12,4	12,4	16	16,4	12,5	16,6
27	10,7	12	12	10,7	12,8	17,5	14,8	11,2	11,2	15,2	15,2	11,7	14,8
28	11,5	12,7	12,7	10,3	12,7	18,2	15,6	11,9	11,9	15,9	15,9	12,5	17

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12
29	10,1	10,4	10,4	10	12,2	16,8	14,3	10,5	10,5	14,6	14,5	11,1	12,6
30	31,2	30,6	30,6	34,9	32,2	24,2	26,7	30,2	30,2	26,2	25,4	27,8	30,9
31	24,9	24,3	24,3	28,7	25,9	18	20,4	24	24	20	19,2	21,6	27,4
32	47	46,4	46,4	50,8	48	40,1	42,5	46,1	46,1	42,6	41,3	43,7	44,4
33	40,9	41,4	41,4	43,3	43	37,8	39,1	38,5	38,5	36,4	39	41,4	38,2
34	27,8	29	29	27,8	29,9	33,3	29,8	27,3	27,3	29	31,1	28,7	20,5
35	25,2	25,7	25,7	28,6	27,3	30	26,5	22,8	22,8	25,7	27,7	24,3	17,2
36	25,5	26,7	26,7	25,4	27,5	31	27,4	23,8	23,8	26,7	28,7	25,3	18,2
37	30,2	30,7	30,7	32,1	32,2	35	31,4	27,8	27,8	30,6	32,7	29,2	22,1
38	36,4	36,9	36,9	38,9	38,5	41,2	37,7	34	34	36,9	39	35,5	28,4
39	42,4	42,9	42,9	44,3	44,4	47,2	43,6	40	40	42,8	44,9	41,4	34,3
40	33,5	34	34	35,4	35,5	38,3	34,7	31,1	31,1	33,9	36	32,5	25,4
41	25	24,4	24,4	28,8	26	18	20,5	24,1	24,1	22,9	19,3	21,7	28,5
42	28,3	27,7	27,7	32,1	29,3	21,3	23,8	27,4	27,4	26,2	22,6	25	31,8
43	29,3	28,7	28,7	33,1	30,3	22,2	24,8	28,3	28,3	27,1	23,5	25,9	32,8
44	17,8	19	19	17,7	19,8	24,5	21,8	18,2	18,2	22,2	22,2	18,8	22,4
45	16,2	17,4	17,4	16,1	18,2	22,9	20,2	16,6	16,6	20,6	20,6	17,2	20,8
46	24,1	23,5	23,5	27,9	25,1	17,1	19,6	23,1	23,1	22	18,4	20,8	27,6
47	35	34,4	34,4	38,8	36	27,9	30,5	34	34	32,8	29,2	31,6	38,5
48	10,9	11,4	11,4	12,8	13	15,7	13,1	8,5	8,5	11,4	13,5	10	2,9
49	13	13,5	13,5	15,5	15,1	17,8	15,2	10,6	10,6	13,5	15,5	12,1	5
50	12	12,4	12,4	11,2	13,3	17,9	15,3	11,8	11,8	14,6	15,7	12,2	4,2
51	19,3	18,8	18,8	23,1	20,3	12,3	14,8	18,4	18,4	14,4	13,6	16	21,8
52	13,5	12,9	12,9	17,3	14,5	6,6	9	12,5	12,5	8,6	7,7	10,1	15,9
53	22,5	23	23	24,9	24,5	22	20,6	20,1	20,1	17,9	22,6	21,5	14,4
54	19,6	20,2	20,2	22,1	21,7	19,3	17,9	17,2	17,2	15,2	19,9	18,7	17
55	24,1	23,5	23,5	27,9	25,1	17,2	19,6	23,2	23,2	19,2	18,4	20,8	26,6
56	26	25,4	25,4	29,8	27	19,1	21,5	25,1	25,1	21,6	20,3	22,7	23,4
57	24,6	24	24	28,4	25,6	17,7	20,1	23,7	23,7	19,7	18,9	21,3	26

Kode Toko	13	14	15	16	17	18	19	20	21	22	23	24	25
0 (depo)	5,8	8,8	5,7	4	4,5	6,5	8,7	9,2	14,5	18,9	20,2	15,8	12,4
1	8,1	9,3	6,2	5,2	5,5	7,7	8,7	9,4	15,7	20,2	21,5	17	13,7
2	8,1	9,3	6,2	5,2	5,5	7,7	8,7	9,4	15,7	20,2	21,5	17	13,7
3	8,5	11,2	8,1	6,7	7,7	8,3	8,3	11,7	14,4	18,9	19,6	15,8	12,4
4	7,9	10,8	7,8	6,1	6,6	8,6	7,5	10,2	16,5	21	22,3	17,9	14,5
5	12,3	11,3	8,7	10,6	9,7	13,2	9,9	2,6	21,2	25,7	27	22,5	19,1
6	9,7	7,8	4,4	8	7,1	10,5	9,2	5,7	18,5	23	24,3	19,4	16,5
7	5,1	6,4	3,3	1,9	4,3	6,9	10,5	9,3	14,9	19,4	20,7	15,8	12,9
8	5,1	6,4	3,3	4,4	4,3	6,9	10,5	9,3	14,9	19,4	20,7	15,8	12,9
9	8	7,2	3,6	8,4	7,1	10,9	11,5	8,1	18,9	23	24,3	18,6	16,9
10	10,1	9,8	6,4	8,4	7,5	10,9	9,6	3,6	18,9	23,4	24,7	20,3	16,9
11	6,6	7,8	4,8	4,9	4	7,4	8,1	6,9	15,5	19,9	21,2	16,8	13,4
12	3,1	4,1	5,6	6,1	4,1	5,3	15	13,7	11,6	14,6	15,8	10,1	9,9
13	0	4,1	4,7	3,2	1,1	5,3	13,9	12,7	12,8	17,2	18,5	12,8	12,2
14	4,1	0	3,7	7,1	5	9,2	14,6	12,6	15	18,2	19,5	13,8	13,6
15	4,7	3,7	0	6,2	4,1	8,7	12,1	9,8	17,5	19,8	21	15,4	15,1
16	3,2	7,1	6,2	0	1,9	3,1	12,7	12,7	12,3	17,1	17,8	13,9	10,4
17	1,1	5	4,1	4,1	0	4,2	13,3	12,5	13,3	18,1	19,1	13,7	9,9
18	5,3	9,2	8,7	3,1	4,2	0	15,2	15,2	9,5	14	15,2	10,5	6,6
19	13,9	14,6	12,1	12,7	13,3	15,2	0	9,1	22,6	27,1	28,9	24,4	21,1
20	12,7	12,6	9,8	12,7	12,5	15,2	9,1	0	21,5	26	27,3	22,9	19,5
21	12,8	15	17,5	12,3	13,3	9,5	22,6	21,5	0	5,5	6,8	2,8	3,4
22	17,2	18,2	19,8	17,1	18,1	14	27,1	26	5,5	0	2,3	4,9	7,9
23	18,5	19,5	21	17,8	19,1	15,2	28,9	27,3	6,8	2,3	0	6,1	8,6
24	12,8	13,8	15,4	13,9	13,7	10,5	24,4	22,9	2,8	4,9	6,1	0	4,8
25	12,2	13,6	15,1	10,4	9,9	6,6	21,1	19,5	3,4	7,9	8,6	4,8	0
26	12,3	16,6	15	10,1	11,1	9,2	20,7	19	11,2	15,7	16,4	12,5	9,2
27	12,4	15,8	13,8	8,9	11,3	7,9	19,4	17,8	7,2	11,7	13,5	8,6	5,2
28	12,6	16,5	14,5	10,4	11,5	9,5	18,4	18,5	12,4	16,9	17,5	13,7	10,3
29	10,4	15,1	13,1	8,2	10,6	7,3	18,8	17,1	5,1	9,6	10,2	6,4	3
30	33,4	28,3	29	33,6	32,6	36,1	32,4	25,4	42,3	45	46,3	40,6	42,1

Kode Toko	13	14	15	16	17	18	19	20	21	22	23	24	25
31	27,1	24,3	22,8	27,4	26,3	29,9	26,2	19,1	38,5	41,3	42,6	36,9	35,9
32	45,1	41,9	42,8	47,6	46,1	50,1	48,3	41,2	55,8	58,6	59,9	54,2	53,4
33	38,9	35,7	36,6	41,4	39,8	43,9	46	39	49	51,8	53	47,4	47,2
34	23,1	24,1	25,7	26,1	24	21,5	34,7	33,7	15,2	14	13,9	12,9	17,1
35	19,8	20,8	22,4	22,8	20,2	22,4	31,4	30,4	16,6	16	14,5	14,9	17,6
36	20,8	21,8	23,4	23,8	21,7	19,2	32,4	31,3	12,9	12	11,6	10,6	14,7
37	24,8	25,8	27,3	27,7	25,7	27,4	36,4	35,3	21,5	17	18,2	17,2	22,5
38	27,6	32	33,6	34	31,9	32,6	42,6	41,6	26,3	23,6	24,9	23,9	15,4
39	37	38	39,5	39,9	37,9	38,1	48,5	47,5	31,8	29,2	30,4	29,4	28
40	28,1	29,1	30,6	31	29	29,2	39,6	38,6	24,8	20,3	21,5	20,5	25,8
41	27,5	27,4	24,6	27,5	26,4	30	26,3	19,2	38,1	42,5	43,2	38,2	36
42	30,8	30,7	27,9	30,8	29,7	33,3	29,6	22,5	41,3	45,8	46,5	41,5	39,3
43	31,8	31,7	28,8	31,7	30,7	34,3	30,5	23,5	42,3	46,8	47,4	42,4	40,2
44	19,4	22,8	20,8	15,9	18,3	15	26,4	24,8	14,4	17,7	16	16,3	12,9
45	17,8	21,2	19,2	14,3	16,7	13,4	24	23,2	13,3	17,3	15,6	14,7	11,3
46	26,6	26,5	23,7	26,6	25,5	29,1	25,3	18,3	37,1	41,6	42,3	37,3	35,1
47	37,5	38,5	34,5	37,4	37,2	40	36,2	29,2	48	24	53,1	48,1	45,9
48	5,5	6,5	8,1	8,5	6,4	6,7	17,1	16,1	8,7	18,1	14,5	9	7
49	7,6	8,6	10,2	10,6	8,5	9,5	19,2	18,1	7,8	20,2	11,8	6,1	6,2
50	5	9,8	11,4	8,7	5,5	5	19,3	18,3	7	21,4	12,8	8	5,4
51	21,6	18,7	17,2	21,8	21,6	24,3	20,6	13,5	32,4	35,7	37,5	31,3	30,3
52	15,7	12,8	11,3	15,9	15,7	18,5	14,7	7,7	26,5	29,8	31,6	25,4	24,5
53	17,1	18,1	18	20	18	19,7	28,6	23,1	13,8	16,6	17,9	12,2	14,8
54	17,6	14,4	15,3	20,1	18,5	22,7	25,8	20,4	21,4	24,1	25,4	19,7	22,3
55	26,6	23,5	22	26,6	26,4	29,1	25,4	18,3	37,2	10,9	41,8	36,1	35,1
56	24,1	20,9	21,8	26,6	25	29,1	27,3	20,2	34,8	12,8	38,9	33,2	32,4
57	27,1	20,8	22,5	27,1	26,9	29,6	25,9	18,8	34,8	11,4	38,8	33,1	35,6

Kode Toko	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
0 (depo)	11,5	10,7	11,5	10,1	31,2	24,9	47	40,9	27,8	25,2	25,5	30,2	36,4	42,4	33,5	25
1	12,9	12	12,7	10,4	30,6	24,3	46,4	41,4	29	25,7	26,7	30,7	36,9	42,9	34	24,4
2	12,9	12	12,7	10,4	30,6	24,3	46,4	41,4	29	25,7	26,7	30,7	36,9	42,9	34	24,4
3	10,4	10,7	10,3	10	34,9	28,7	50,8	43,3	27,8	28,6	25,4	32,1	38,9	44,3	35,4	28,8
4	14,1	12,8	12,7	12,2	32,2	25,9	48	43	29,9	27,3	27,5	32,2	38,5	44,4	35,5	26
5	18,7	17,5	18,2	16,8	24,2	18	40,1	37,8	33,3	30	31	35	41,2	47,2	38,3	18
6	15,6	14,8	15,6	14,3	26,7	20,4	42,5	39,1	29,8	26,5	27,4	31,4	37,7	43,6	34,7	20,5
7	12,4	11,2	11,9	10,5	30,2	24	46,1	38,5	27,3	22,8	23,8	27,8	34	40	31,1	24,1
8	12,4	11,2	11,9	10,5	30,2	24	46,1	38,5	27,3	22,8	23,8	27,8	34	40	31,1	24,1
9	16	15,2	15,9	14,6	26,2	20	42,6	36,4	29	25,7	26,7	30,6	36,9	42,8	33,9	22,9
10	16,4	15,2	15,9	14,5	25,4	19,2	41,3	39	31,1	27,7	28,7	32,7	39	44,9	36	19,3
11	12,5	11,7	12,5	11,1	27,8	21,6	43,7	41,4	28,7	24,3	25,3	29,2	35,5	41,4	32,5	21,7
12	16,6	14,8	17	12,6	30,9	27,4	44,4	38,2	20,5	17,2	18,2	22,1	28,4	34,3	25,4	28,5
13	12,3	12,4	12,6	10,4	33,4	27,1	45,1	38,9	23,1	19,8	20,8	24,8	27,6	37	28,1	27,5
14	16,6	15,8	16,5	15,1	28,3	24,3	41,9	35,7	24,1	20,8	21,8	25,8	32	38	29,1	27,4
15	15	13,8	14,5	13,1	29	22,8	42,8	36,6	25,7	22,4	23,4	27,3	33,6	39,5	30,6	24,6
16	10,1	8,9	10,4	8,2	33,6	27,4	47,6	41,4	26,1	22,8	23,8	27,7	34	39,9	31	27,5
17	11,1	11,3	11,5	10,6	32,6	26,3	46,1	39,8	24	20,2	21,7	25,7	31,9	37,9	29	26,4
18	9,2	7,9	9,5	7,3	36,1	29,9	50,1	43,9	21,5	22,4	19,2	27,4	32,6	38,1	29,2	30
19	20,7	19,4	18,4	18,8	32,4	26,2	48,3	46	34,7	31,4	32,4	36,4	42,6	48,5	39,6	26,3
20	19	17,8	18,5	17,1	25,4	19,1	41,2	39	33,7	30,4	31,3	35,3	41,6	47,5	38,6	19,2
21	11,2	7,2	12,4	5,1	42,3	38,5	55,8	49	15,2	16,6	12,9	21,5	26,3	31,8	24,8	38,1
22	15,7	11,7	16,9	9,6	45	41,3	58,6	51,8	14	16	12	17	23,6	29,2	20,3	42,5
23	16,4	13,5	17,5	10,2	46,3	42,6	59,9	53	13,9	14,5	11,6	18,2	24,9	30,4	21,5	43,2
24	12,5	8,6	13,7	6,4	40,6	36,9	54,2	47,4	12,9	14,9	10,6	17,2	23,9	29,4	20,5	38,2
25	9,2	5,2	10,3	3	42,1	35,9	53,4	47,2	17,1	17,6	14,7	22,5	28	33,6	25,8	36
26	0	7,5	5,7	6,8	41,7	35,5	56,4	50,2	24,5	26,6	22,2	28,9	34,5	41,1	32,2	35,6
27	7,5	0	8,7	2,6	40,5	34,2	55,2	48,9	20,6	22,6	18,2	24,9	31,6	37,1	28,2	34,3
28	5,7	8,7	0	8	41,2	35	55,9	49,7	25,7	27,8	23,4	30	36,7	42,2	33,3	35,1
29	6,8	2,6	8	0	39,8	33,6	54,5	48,3	18,4	20,4	16	22,7	29,4	34,9	26	33,7
30	41,7	40,5	41,2	39,8	0	7	20,6	23,8	48,3	45	46	50	46,2	62,2	53,3	17,8

Kode Toko	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
31	35,5	34,2	35	33,6	7	0	26,1	29,2	46,1	42,8	43,7	47,7	48,5	59,9	51	16,8
32	56,4	55,2	55,9	54,5	20,6	26,1	0	31,8	64,2	60,9	61,9	65,9	62,1	78,1	69,2	36,9
33	50,2	48,9	49,7	48,3	23,8	29,2	31,8	0	57,9	54,7	55,6	59,7	55,3	71,9	63	40,1
34	24,5	20,6	25,7	18,4	48,3	46,1	64,2	57,9	0	6,2	4,6	12,2	18,9	24,4	10,8	48,5
35	26,6	22,6	27,8	20,4	45	42,8	60,9	54,7	6,2	0	7,8	13,7	17,9	25,9	17	43,5
36	22,2	18,2	23,4	16	46	43,7	61,9	55,6	4,6	7,8	0	12,3	19	24,5	14,5	46,2
37	28,9	24,9	30	22,7	50	47,7	65,9	59,7	12,2	13,7	12,3	0	19,2	13,2	11	50,5
38	34,5	31,6	36,7	29,4	46,2	48,5	62,1	55,3	18,9	17,9	19	19,2	0	31,3	22,4	56,7
39	41,1	37,1	42,2	34,9	62,2	59,9	78,1	71,9	24,4	25,9	24,5	13,2	31,3	0	23,2	62,7
40	32,2	28,2	33,3	26	53,3	51	69,2	63	10,8	17	14,5	11	22,4	23,2	0	53,8
41	35,6	34,3	35,1	33,7	17,8	16,8	36,9	40,1	48,5	43,5	46,2	50,5	56,7	62,7	53,8	0
42	38,9	37,6	38,4	37	21,3	20,3	40,4	43,6	51,8	46,8	49,5	53,7	60	65,9	57	4,7
43	39,8	38,6	39,3	37,9	17,6	16,6	36,8	39,9	52,8	47,7	50,4	54,7	61	66,9	58	4,3
44	12,2	8,4	15,7	10,5	46,2	40,5	62,4	56	26	22,4	23,7	31,8	38,5	44	35,1	40,3
45	10,6	6,8	14,1	8,9	45	38,9	60,6	54,4	27,1	22	24,4	31,4	38,1	43,6	34,7	38,7
46	34,6	33,4	34,1	32,7	38,2	33,2	55,1	52,8	47,6	42,6	45,3	49,5	55,8	61,7	52,8	17,7
47	45,5	44,3	45	43,6	14	13	33,1	36,3	54,9	54,5	57,2	60,8	67,7	72,3	63,4	10,6
48	13,8	12,5	14,1	10	30,7	29,8	46,6	40,4	19,3	14,3	17	22	27,5	33,4	24,6	31,5
49	15,4	11,4	16,9	9,2	32,8	31,9	48,7	42,5	16,5	11,4	14,1	20,1	25,6	31,6	22,7	33,6
50	12,1	10,9	12,4	8,4	34	33	49,9	43,7	17,8	12,8	15,5	25	30,5	36,5	27,6	33,8
51	29,9	28,6	29,4	28	17,1	10,9	33,7	31,5	41,6	37,8	40,5	45,5	48,5	57	45,5	13,5
52	24	22,8	23,5	22,1	18,1	11,9	33,8	31,6	35,8	32	34,7	39,6	42,6	48,6	42,2	22,4
53	23,8	19,8	25	17,6	26,8	29,1	42,7	35,9	21,1	15,4	19,8	22,3	20,2	34,5	25,6	37,8
54	31,3	27,3	28,4	25,2	21,6	25,4	37,5	30,7	28,7	22,9	27,3	29,8	27,7	42	33,1	35,1
55	34,7	33,4	34,2	32,8	13,4	7,2	32,5	35,1	46,4	41,4	44,1	49,1	53,3	59,2	51,9	8,4
56	35,4	34,2	34,9	33,5	5,2	9	22,4	24,2	42,1	36,4	40,8	44	41,2	55,5	46,6	20
57	35,2	33,9	34,7	33,3	14	7,8	27,3	27,1	46	36,4	40,7	44	41,2	60	51	15,8

Kode Toko	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
0 (depo)	28,3	29,3	17,8	16,2	24,1	35	10,9	13	12	19,3	13,5	22,5	19,6	24,1	26	24,6
1	27,7	28,7	19	17,4	23,5	34,4	11,4	13,5	12,4	18,8	12,9	23	20,2	23,5	25,4	24
2	27,7	28,7	19	17,4	23,5	34,4	11,4	13,5	12,4	18,8	12,9	23	20,2	23,5	25,4	24
3	32,1	33,1	17,7	16,1	27,9	38,8	12,8	15,5	11,2	23,1	17,3	24,9	22,1	27,9	29,8	28,4
4	29,3	30,3	19,8	18,2	25,1	36	13	15,1	13,3	20,3	14,5	24,5	21,7	25,1	27	25,6
5	21,3	22,2	24,5	22,9	17,1	27,9	15,7	17,8	17,9	12,3	6,6	22	19,3	17,2	19,1	17,7
6	23,8	24,8	21,8	20,2	19,6	30,5	13,1	15,2	15,3	14,8	9	20,6	17,9	19,6	21,5	20,1
7	27,4	28,3	18,2	16,6	23,1	34	8,5	10,6	11,8	18,4	12,5	20,1	17,2	23,2	25,1	23,7
8	27,4	28,3	18,2	16,6	23,1	34	8,5	10,6	11,8	18,4	12,5	20,1	17,2	23,2	25,1	23,7
9	26,2	27,1	22,2	20,6	22	32,8	11,4	13,5	14,6	14,4	8,6	17,9	15,2	19,2	21,6	19,7
10	22,6	23,5	22,2	20,6	18,4	29,2	13,5	15,5	15,7	13,6	7,7	22,6	19,9	18,4	20,3	18,9
11	25	25,9	18,8	17,2	20,8	31,6	10	12,1	12,2	16	10,1	21,5	18,7	20,8	22,7	21,3
12	31,8	32,8	22,4	20,8	27,6	38,5	2,9	5	4,2	21,8	15,9	14,4	17	26,6	23,4	26
13	30,8	31,8	19,4	17,8	26,6	37,5	5,5	7,6	5	21,6	15,7	17,1	17,6	26,6	24,1	27,1
14	30,7	31,7	22,8	21,2	26,5	38,5	6,5	8,6	9,8	18,7	12,8	18,1	14,4	23,5	20,9	20,8
15	27,9	28,8	20,8	19,2	23,7	34,5	8,1	10,2	11,4	17,2	11,3	18	15,3	22	21,8	22,5
16	30,8	31,7	15,9	14,3	26,6	37,4	8,5	10,6	8,7	21,8	15,9	20	20,1	26,6	26,6	27,1
17	29,7	30,7	18,3	16,7	25,5	37,2	6,4	8,5	5,5	21,6	15,7	18	18,5	26,4	25	26,9
18	33,3	34,3	15	13,4	29,1	40	6,7	9,5	5	24,3	18,5	19,7	22,7	29,1	29,1	29,6
19	29,6	30,5	26,4	24	25,3	36,2	17,1	19,2	19,3	20,6	14,7	28,6	25,8	25,4	27,3	25,9
20	22,5	23,5	24,8	23,2	18,3	29,2	16,1	18,1	18,3	13,5	7,7	23,1	20,4	18,3	20,2	18,8
21	41,3	42,3	14,4	13,3	37,1	48	8,7	7,8	7	32,4	26,5	13,8	21,4	37,2	34,8	34,8
22	45,8	46,8	17,7	17,3	41,6	24	18,1	20,2	21,4	35,7	29,8	16,6	24,1	10,9	12,8	11,4
23	46,5	47,4	16	15,6	42,3	53,1	14,5	11,8	12,8	37,5	31,6	17,9	25,4	41,8	38,9	38,8
24	41,5	42,4	16,3	14,7	37,3	48,1	9	6,1	8	31,3	25,4	12,2	19,7	36,1	33,2	33,1
25	39,3	40,2	12,9	11,3	35,1	45,9	7	6,2	5,4	30,3	24,5	14,8	22,3	35,1	32,4	35,6
26	38,9	39,8	12,2	10,6	34,6	45,5	13,8	15,4	12,1	29,9	24	23,8	31,3	34,7	35,4	35,2
27	37,6	38,6	8,4	6,8	33,4	44,3	12,5	11,4	10,9	28,6	22,8	19,8	27,3	33,4	34,2	33,9
28	38,4	39,3	15,7	14,1	34,1	45	14,1	16,9	12,4	29,4	23,5	25	28,4	34,2	34,9	34,7
29	37	37,9	10,5	8,9	32,7	43,6	10	9,2	8,4	28	22,1	17,6	25,2	32,8	33,5	33,3
30	21,3	17,6	46,2	45	38,2	14	30,7	32,8	34	17,1	18,1	26,8	21,6	13,4	5,2	14

Kode Toko	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
31	20,3	16,6	40,5	38,9	33,2	13	29,8	31,9	33	10,9	11,9	29,1	25,4	7,2	9	7,8
32	40,4	36,8	62,4	60,6	55,1	33,1	46,6	48,7	49,9	33,7	33,8	42,7	37,5	32,5	22,4	27,3
33	43,6	39,9	56	54,4	52,8	36,3	40,4	42,5	43,7	31,5	31,6	35,9	30,7	35,1	24,2	27,1
34	51,8	52,8	26	27,1	47,6	54,9	19,3	16,5	17,8	41,6	35,8	21,1	28,7	46,4	42,1	46
35	46,8	47,7	22,4	22	42,6	54,5	14,3	11,4	12,8	37,8	32	15,4	22,9	41,4	36,4	36,4
36	49,5	50,4	23,7	24,4	45,3	57,2	17	14,1	15,5	40,5	34,7	19,8	27,3	44,1	40,8	40,7
37	53,7	54,7	31,8	31,4	49,5	60,8	22	20,1	25	45,5	39,6	22,3	29,8	49,1	44	44
38	60	61	38,5	38,1	55,8	67,7	27,5	25,6	30,5	48,5	42,6	20,2	27,7	53,3	41,2	41,2
39	65,9	66,9	44	43,6	61,7	72,3	33,4	31,6	36,5	57	48,6	34,5	42	59,2	55,5	60
40	57	58	35,1	34,7	52,8	63,4	24,6	22,7	27,6	45,5	42,2	25,6	33,1	51,9	46,6	51
41	4,7	4,3	40,3	38,7	17,7	10,6	31,5	33,6	33,8	13,5	22,4	37,8	35,1	8,4	20	15,8
42	0	3,5	43,7	42,1	16,6	8,2	34,9	37	37,1	18,4	25,8	41,2	38,5	11,9	23,5	19,3
43	3,5	0	44,1	42,5	17,6	4,4	35,3	37,4	37,5	18,2	26,2	41,6	35,3	11,4	22,5	18,8
44	43,7	44,1	0	10,5	40,4	51,3	19,6	19,1	17,9	35,7	29,8	27,5	35,1	40,4	41,2	40,9
45	42,1	42,5	10,5	0	38,8	49,7	18	17,5	16,3	34,1	28,2	25,9	33,5	38,9	39,6	39,3
46	16,6	17,6	40,4	38,8	0	31,3	30,7	32,8	32,9	27,4	21,6	37	34,3	24,5	34,1	32,7
47	8,2	4,4	51,3	49,7	31,3	0	41,8	43,9	44	22	24	41,2	37,5	17,3	18,9	20
48	34,9	35,3	19,6	18	30,7	41,8	0	3	3,3	22,7	16,8	13,2	20,8	28,8	25,6	25,6
49	37	37,4	19,1	17,5	32,8	43,9	3	0	6,1	24,8	18,9	11,4	18,9	30,9	27,7	27,7
50	37,1	37,5	17,9	16,3	32,9	44	3,3	6,1	0	27	19,8	16,3	23,8	31,8	28,6	28,6
51	18,4	18,2	35,7	34,1	27,4	22	22,7	24,8	27	0	6,2	24,1	20,8	7,4	12,7	10,5
52	25,8	26,2	29,8	28,2	21,6	24	16,8	18,9	19,8	6,2	0	18,3	15,6	10,9	12,8	11,4
53	41,2	41,6	27,5	25,9	37	41,2	13,2	11,4	16,3	24,1	18,3	0	8,3	28,1	21,8	21,8
54	38,5	35,3	35,1	33,5	34,3	37,5	20,8	18,9	23,8	20,8	15,6	8,3	0	24,4	16,5	16,5
55	11,9	11,4	40,4	38,9	24,5	17,3	28,8	30,9	31,8	7,4	10,9	28,1	24,4	0	12,2	8
56	23,5	22,5	41,2	39,6	34,1	18,9	25,6	27,7	28,6	12,7	12,8	21,8	16,5	12,2	0	6,3
57	19,3	18,8	40,9	39,3	32,7	20	25,6	27,7	28,6	10,5	11,4	21,8	16,5	8	6,3	0

A6. Data Kecepatan Rata-Rata Kendaraan dari Depo ke Setiap Toko dan Antar Toko untuk 57 Pelanggan (km/jam)

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12
0 (depo)	0	26	26	30	27	25	26	28	27	30	26	26	25
1	26	0	29	29	26	28	29	29	29	29	27	29	28
2	26	29	0	30	26	25	26	29	30	26	29	30	26
3	30	29	30	0	25	33	26	27	26	27	33	30	29
4	27	26	26	25	0	27	29	29	28	29	30	26	30
5	25	28	25	33	27	0	29	26	30	27	26	29	34
6	26	29	26	26	29	29	0	27	27	28	27	26	33
7	28	29	29	27	29	26	27	0	26	29	25	25	29
8	27	29	30	26	28	30	27	26	0	25	29	27	25
9	30	29	26	27	29	27	28	29	25	0	27	29	27
10	26	27	29	33	30	26	27	25	29	27	0	27	27
11	26	29	30	30	26	29	26	25	27	29	27	0	27
12	25	28	26	29	30	34	33	29	25	27	27	27	0
13	27	26	28	26	27	33	27	30	28	26	31	28	27
14	28	27	29	32	34	30	30	28	26	28	25	26	30
15	29	29	27	28	30	27	27	29	27	26	25	29	30
16	27	25	26	29	29	35	28	27	26	25	27	26	27
17	25	26	29	27	27	26	28	27	28	27	28	26	27
18	28	28	28	25	27	28	31	28	29	33	33	28	30
19	28	26	27	30	29	27	29	26	29	27	27	27	29
20	26	27	27	30	33	27	27	27	28	26	26	27	34
21	27	30	26	34	29	35	35	34	26	26	31	26	26
22	32	33	30	34	33	32	34	29	31	38	32	26	29
23	33	39	39	34	34	35	32	35	37	40	33	39	32
24	28	32	27	29	31	39	31	29	33	29	32	29	32
25	29	26	33	29	32	27	32	35	31	26	33	25	26
26	34	35	30	31	33	25	30	30	27	27	26	30	29
27	28	34	28	32	30	27	32	34	34	34	30	34	32
28	33	33	26	28	27	30	34	26	29	26	32	34	31

Kode Toko	0 (depo)	1	2	3	4	5	6	7	8	9	10	11	12
29	30	31	32	26	35	30	34	28	34	26	32	30	32
30	43	45	44	35	38	38	33	39	41	33	37	35	42
31	38	31	33	31	35	31	39	38	30	29	32	32	30
32	49	49	42	43	40	47	42	42	45	47	41	50	43
33	41	48	46	40	41	42	37	37	44	38	39	41	40
34	39	37	35	37	31	40	35	32	32	37	41	31	33
35	38	37	34	34	38	33	31	36	36	36	33	33	26
36	33	36	31	34	32	36	32	32	37	30	38	39	28
37	37	36	39	36	38	40	38	37	33	37	36	33	31
38	40	41	41	43	38	45	38	44	36	41	39	41	31
39	42	43	41	49	41	48	41	42	40	45	48	43	39
40	44	42	42	43	44	42	37	42	44	45	44	44	38
41	33	33	38	31	32	32	33	34	38	40	29	36	39
42	31	33	39	37	34	32	33	37	33	38	38	32	44
43	34	32	36	43	36	33	32	34	37	32	31	39	36
44	30	29	33	29	32	32	31	31	31	37	33	26	39
45	25	29	32	34	26	37	39	33	32	37	35	26	36
46	36	35	37	36	34	31	27	37	37	36	30	36	36
47	41	38	44	37	43	37	43	45	40	44	34	38	44
48	28	34	27	33	26	28	34	26	28	30	26	25	25
49	29	26	34	30	33	26	29	25	33	34	29	34	29
50	27	29	34	27	29	31	26	29	29	25	35	28	27
51	31	34	32	39	36	26	34	34	28	31	33	34	34
52	33	34	29	27	31	29	29	35	32	26	28	30	35
53	39	32	37	36	30	37	32	38	34	32	38	33	35
54	30	36	32	32	36	25	30	27	29	27	30	35	30
55	33	34	39	39	34	27	26	30	36	26	31	38	37
56	32	40	32	36	30	32	39	36	30	33	40	33	30
57	36	38	36	31	37	30	38	37	37	26	32	39	35

Kode Toko	13	14	15	16	17	18	19	20	21	22	23	24	25
0 (depo)	27	28	29	27	25	28	28	26	27	32	33	28	29
1	26	27	29	25	26	28	26	27	30	33	39	32	26
2	28	29	27	26	29	28	27	27	26	30	39	27	33
3	26	32	28	29	27	25	30	30	34	34	34	29	29
4	27	34	30	29	27	27	29	33	29	33	34	31	32
5	33	30	27	35	26	28	27	27	35	32	35	39	27
6	27	30	27	28	28	31	29	27	35	34	32	31	32
7	30	28	29	27	27	28	26	27	34	29	35	29	35
8	28	26	27	26	28	29	29	28	26	31	37	33	31
9	26	28	26	25	27	33	27	26	26	38	40	29	26
10	31	25	25	27	28	33	27	26	31	32	33	32	33
11	28	26	29	26	26	28	27	27	26	26	39	29	25
12	27	30	30	27	27	30	29	34	26	29	32	32	26
13	0	28	27	29	25	28	25	32	30	29	33	27	27
14	28	0	26	29	25	26	34	31	32	26	29	29	26
15	27	26	0	30	27	28	31	30	35	30	34	25	32
16	29	29	30	0	26	25	27	27	31	27	32	32	28
17	25	25	27	26	0	28	28	29	30	25	28	33	27
18	28	26	28	25	28	0	32	31	30	27	26	32	26
19	25	34	31	27	28	32	0	27	33	32	37	36	35
20	32	31	30	27	29	31	27	0	32	32	33	38	29
21	30	32	35	31	30	30	33	32	0	26	26	26	26
22	29	26	30	27	25	27	32	32	26	0	27	25	30
23	33	29	34	32	28	26	37	33	26	27	0	29	28
24	27	29	25	32	33	32	36	38	26	25	29	0	26
25	27	26	32	28	27	26	35	29	26	30	28	26	0
26	33	30	28	31	28	28	39	26	27	26	30	35	25
27	35	29	29	28	28	25	25	29	28	32	31	30	28
28	28	27	33	32	33	25	32	34	32	33	35	34	25
29	32	27	31	28	31	29	33	31	29	27	29	27	27
30	43	33	36	35	39	43	41	38	41	40	43	46	49

Kode Toko	13	14	15	16	17	18	19	20	21	22	23	24	25
31	38	31	33	38	31	31	40	28	36	40	43	43	35
32	43	43	43	48	50	42	50	41	42	43	49	44	45
33	36	37	38	48	35	41	48	36	44	47	48	47	48
34	36	34	37	34	33	39	36	41	28	44	27	33	32
35	26	30	32	34	37	30	40	39	30	40	32	32	27
36	38	32	36	33	36	28	36	43	34	36	25	31	35
37	40	33	33	37	34	36	39	35	33	29	35	35	39
38	39	41	38	39	41	40	41	46	36	33	31	39	32
39	37	36	38	35	35	42	49	43	37	33	36	31	36
40	39	32	40	40	34	32	41	35	33	33	37	35	39
41	31	34	36	33	33	35	40	34	42	50	43	38	44
42	35	35	31	38	38	40	39	32	41	46	47	41	44
43	42	40	30	36	44	45	40	36	50	42	41	43	46
44	26	40	38	30	34	28	30	38	34	34	31	29	29
45	27	39	27	29	28	27	37	32	33	25	26	34	26
46	37	39	37	40	39	36	31	34	36	46	43	37	40
47	38	37	42	40	36	40	40	39	49	37	46	41	44
48	28	28	26	29	27	26	31	28	28	34	25	25	28
49	28	25	33	33	28	27	33	33	26	40	30	29	26
50	27	29	30	27	26	28	31	28	25	32	28	26	28
51	35	25	32	33	31	38	31	26	39	35	38	39	45
52	31	33	28	28	26	26	28	29	40	31	40	38	37
53	29	28	27	33	26	27	39	39	32	27	30	31	34
54	34	26	34	40	29	30	37	38	35	36	31	33	39
55	34	39	36	36	40	37	36	29	43	31	45	40	44
56	39	38	32	39	36	37	36	32	38	33	43	39	40
57	34	31	36	39	33	31	38	30	42	26	42	38	37

Kode Toko	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
0 (depo)	34	28	33	30	43	38	49	41	39	38	33	37	40	42	44	33
1	35	34	33	31	45	31	49	48	37	37	36	36	41	43	42	33
2	30	28	26	32	44	33	42	46	35	34	31	39	41	41	42	38
3	31	32	28	26	35	31	43	40	37	34	34	36	43	49	43	31
4	33	30	27	35	38	35	40	41	31	38	32	38	38	41	44	32
5	25	27	30	30	38	31	47	42	40	33	36	40	45	48	42	32
6	30	32	34	34	33	39	42	37	35	31	32	38	38	41	37	33
7	30	34	26	28	39	38	42	37	32	36	32	37	44	42	42	34
8	27	34	29	34	41	30	45	44	32	36	37	33	36	40	44	38
9	27	34	26	26	33	29	47	38	37	36	30	37	41	45	45	40
10	26	30	32	32	37	32	41	39	41	33	38	36	39	48	44	29
11	30	34	34	30	35	32	50	41	31	33	39	33	41	43	44	36
12	29	32	31	32	42	30	43	40	33	26	28	31	31	39	38	39
13	33	35	28	32	43	38	43	36	36	26	38	40	39	37	39	31
14	30	29	27	27	33	31	43	37	34	30	32	33	41	36	32	34
15	28	29	33	31	36	33	43	38	37	32	36	33	38	38	40	36
16	31	28	32	28	35	38	48	48	34	34	33	37	39	35	40	33
17	28	28	33	31	39	31	50	35	33	37	36	34	41	35	34	33
18	28	25	25	29	43	31	42	41	39	30	28	36	40	42	32	35
19	39	25	32	33	41	40	50	48	36	40	36	39	41	49	41	40
20	26	29	34	31	38	28	41	36	41	39	43	35	46	43	35	34
21	27	28	32	29	41	36	42	44	28	30	34	33	36	37	33	42
22	26	32	33	27	40	40	43	47	44	40	36	29	33	33	33	50
23	30	31	35	29	43	43	49	48	27	32	25	35	31	36	37	43
24	35	30	34	27	46	43	44	47	33	32	31	35	39	31	35	38
25	25	28	25	27	49	35	45	48	32	27	35	39	32	36	39	44
26	0	29	25	26	49	38	49	47	37	34	39	36	38	48	43	45
27	29	0	26	30	44	37	46	44	38	31	25	38	36	45	36	38
28	25	26	0	29	44	37	47	40	34	38	33	36	38	50	35	38
29	26	30	29	0	42	37	44	48	26	36	33	36	37	41	37	40
30	49	44	44	42	0	28	34	31	43	47	42	47	44	40	47	31

Kode Toko	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
31	38	37	37	37	28	0	36	37	49	50	50	45	45	41	43	34
32	49	46	47	44	34	36	0	41	49	41	40	42	46	48	49	36
33	47	44	40	48	31	37	41	0	44	44	42	48	49	47	41	41
34	37	38	34	26	43	49	49	44	0	28	30	34	33	39	28	47
35	34	31	38	36	47	50	41	44	28	0	27	29	28	34	30	43
36	39	25	33	33	42	50	40	42	30	27	0	35	26	39	27	48
37	36	38	36	36	47	45	42	48	34	29	35	0	29	28	31	40
38	38	36	38	37	44	45	46	49	33	28	26	29	0	43	31	47
39	48	45	50	41	40	41	48	47	39	34	39	28	43	0	32	41
40	43	36	35	37	47	43	49	41	28	30	27	31	31	32	0	42
41	45	38	38	40	31	34	36	41	47	43	48	40	47	41	42	0
42	45	42	40	36	37	37	49	41	45	45	46	41	42	46	41	25
43	41	40	39	38	33	28	44	37	47	40	43	46	41	50	43	28
44	25	28	27	33	47	44	45	44	36	31	37	45	37	47	37	44
45	28	28	29	28	43	39	41	43	34	31	39	43	43	41	43	39
46	38	42	43	39	42	40	49	46	41	45	44	43	42	42	47	32
47	42	40	48	49	33	27	37	42	45	49	49	49	49	43	43	32
48	32	33	32	28	38	34	48	43	30	29	34	39	38	36	37	43
49	31	29	29	26	43	44	43	48	30	34	34	33	39	43	34	41
50	34	27	28	28	36	36	47	43	30	28	34	33	35	40	35	40
51	35	39	31	38	34	29	44	41	44	37	42	44	42	42	41	27
52	34	39	39	38	35	26	36	44	42	36	35	42	44	46	47	32
53	36	33	31	30	34	38	49	37	37	31	26	31	34	37	35	44
54	37	36	37	31	33	38	44	40	37	36	37	38	32	42	39	39
55	35	39	41	36	28	29	42	42	45	46	43	43	49	46	48	27
56	37	41	39	41	27	28	37	32	41	43	42	41	41	43	41	30
57	44	40	40	41	33	30	39	38	48	35	41	49	47	45	47	31

Kode Toko	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
0 (depo)	31	34	30	25	36	41	28	29	27	31	33	39	30	33	32	36
1	33	32	29	29	35	38	34	26	29	34	34	32	36	34	40	38
2	39	36	33	32	37	44	27	34	34	32	29	37	32	39	32	36
3	37	43	29	34	36	37	33	30	27	39	27	36	32	39	36	31
4	34	36	32	26	34	43	26	33	29	36	31	30	36	34	30	37
5	32	33	32	37	31	37	28	26	31	26	29	37	25	27	32	30
6	33	32	31	39	27	43	34	29	26	34	29	32	30	26	39	38
7	37	34	31	33	37	45	26	25	29	34	35	38	27	30	36	37
8	33	37	31	32	37	40	28	33	29	28	32	34	29	36	30	37
9	38	32	37	37	36	44	30	34	25	31	26	32	27	26	33	26
10	38	31	33	35	30	34	26	29	35	33	28	38	30	31	40	32
11	32	39	26	26	36	38	25	34	28	34	30	33	35	38	33	39
12	44	36	39	36	36	44	25	29	27	34	35	35	30	37	30	35
13	35	42	26	27	37	38	28	28	27	35	31	29	34	34	39	34
14	35	40	40	39	39	37	28	25	29	25	33	28	26	39	38	31
15	31	30	38	27	37	42	26	33	30	32	28	27	34	36	32	36
16	38	36	30	29	40	40	29	33	27	33	28	33	40	36	39	39
17	38	44	34	28	39	36	27	28	26	31	26	26	29	40	36	33
18	40	45	28	27	36	40	26	27	28	38	26	27	30	37	37	31
19	39	40	30	37	31	40	31	33	31	31	28	39	37	36	36	38
20	32	36	38	32	34	39	28	33	28	26	29	39	38	29	32	30
21	41	50	34	33	36	49	28	26	25	39	40	32	35	43	38	42
22	46	42	34	25	46	37	34	40	32	35	31	27	36	31	33	26
23	47	41	31	26	43	46	25	30	28	38	40	30	31	45	43	42
24	41	43	29	34	37	41	25	29	26	39	38	31	33	40	39	38
25	44	46	29	26	40	44	28	26	28	45	37	34	39	44	40	37
26	45	41	25	28	38	42	32	31	34	35	34	36	37	35	37	44
27	42	40	28	28	42	40	33	29	27	39	39	33	36	39	41	40
28	40	39	27	29	43	48	32	29	28	31	39	31	37	41	39	40
29	36	38	33	28	39	49	28	26	28	38	38	30	31	36	41	41
30	37	33	47	43	42	33	38	43	36	34	35	34	33	28	27	33

Kode Toko	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
31	37	28	44	39	40	27	34	44	36	29	26	38	38	29	28	30
32	49	44	45	41	49	37	48	43	47	44	36	49	44	42	37	39
33	41	37	44	43	46	42	43	48	43	41	44	37	40	42	32	38
34	45	47	36	34	41	45	30	30	30	44	42	37	37	45	41	48
35	45	40	31	31	45	49	29	34	28	37	36	31	36	46	43	35
36	46	43	37	39	44	49	34	34	34	42	35	26	37	43	42	41
37	41	46	45	43	43	49	39	33	33	44	42	31	38	43	41	49
38	42	41	37	43	42	49	38	39	35	42	44	34	32	49	41	47
39	46	50	47	41	42	43	36	43	40	42	46	37	42	46	43	45
40	41	43	37	43	47	43	37	34	35	41	47	35	39	48	41	47
41	25	28	44	39	32	32	43	41	40	27	32	44	39	27	30	31
42	0	27	48	44	27	28	44	37	43	29	35	49	39	31	34	27
43	27	0	49	49	30	26	44	37	42	30	38	40	40	29	30	34
44	48	49	0	35	42	44	32	28	34	37	37	31	38	48	41	43
45	44	49	35	0	43	43	31	32	32	42	33	37	43	42	37	38
46	27	30	42	43	0	38	45	43	42	38	34	43	43	36	35	45
47	28	26	44	43	38	0	48	43	40	30	31	45	38	33	33	32
48	44	44	32	31	45	48	0	29	27	35	35	34	32	40	36	31
49	37	37	28	32	43	43	29	0	27	33	29	28	27	45	38	39
50	43	42	34	32	42	40	27	27	0	39	35	26	39	44	30	32
51	29	30	37	42	38	30	35	33	39	0	30	35	34	25	30	29
52	35	38	37	33	34	31	35	29	35	30	0	28	25	33	35	29
53	49	40	31	37	43	45	34	28	26	35	28	0	27	39	31	40
54	39	40	38	43	43	38	32	27	39	34	25	27	0	35	28	30
55	31	29	48	42	36	33	40	45	44	25	33	39	35	0	32	26
56	34	30	41	37	35	33	36	38	30	30	35	31	28	32	0	28
57	27	34	43	38	45	32	31	39	32	29	29	40	30	26	28	0

Lampiran B. Hasil Uji Parameter Algoritma Genetika dan Algoritma CSO

B1. Data Uji Percobaan Perubahan Parameter Algoritma Genetika pada 32 Toko

- (Probabilitas *Crossover*) P_c

Tabel 4.38 Hasil Percobaan Uji $P_c=0,1; P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	536,9km	Rp 1.268.450,-	481	5,2994
2	4	520km	Rp 1.060.000,-	712	3,6109
3	4	717,3km	Rp 1.158.650,-	911	3,4225
4	5	665,4km	Rp 1.332.700,-	354	3,846
5	5	572,7km	Rp 1.286.350,-	522	3,7802
Rata-rata			Rp 1.221.230,-	596	3,9918

Tabel 4.39 Hasil Percobaan Uji $P_c=0,3; P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	564,4km	Rp 1.282.200,-	780	5,7508
2	5	626,7km	Rp 1.313.350,-	19	5,3887
3	5	571,5km	Rp 1.285.750,-	696	5,6028
4	4	604,3km	Rp 1.102.150,-	877	5,332
5	4	631,7km	Rp 1.115.850,-	406	5,332
Rata-rata			Rp 1.219.860,-	555,6	5,4813

Tabel 4.40 Hasil Percobaan Uji $P_c=0,5; P_m=0,5$

No	Jumlah Kendaraan	Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	654,8km	Rp 1.327.400,-	968	6,9939
2	5	617,4km	Rp 1.308.700,-	998	7,3808
3	4	568km	Rp 1.084.000,-	470	5,9058
4	4	628,8km	Rp 1.114.400,-	513	6,237
5	4	681km	Rp 1.140.500,-	782	7,917
Rata-rata			Rp 1.195.000,-	746,2	6,8869

Tabel 4.41 Hasil Percobaan Uji $P_c=0,7; P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	619,3km	Rp 1.309.650,-	552	10,3566
2	5	614,8km	Rp 1.307.400,-	234	10,9772
3	4	590,3km	Rp 1.095.150,-	954	9,1824
4	4	607,7km	Rp 1.103.850,-	849	9,2185
5	4	602,2km	Rp 1.101.100,-	638	11,025
Rata-rata			Rp 1.183.430,-	645,4	10,1519

Tabel 4.42 Hasil Percobaan Uji $P_c=0,9; P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	685,9km	Rp 1.142.950,-	874	10,7427
2	5	588,3km	Rp 1.294.150,-	429	17,1513
3	4	649,5km	Rp 1.124.750,-	945	10,6175
4	4	607,7km	Rp 1.103.850,-	716	10,373
5	4	661,5km	Rp 1.130.750,-	577	10,4922
Rata-rata			Rp 1.159.290,-	708,2	11,8753

- (Probabilitas Mutasi) P_m

Tabel 4.43 Hasil Percobaan Uji $P_c=0,9; P_m=0,1$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	461,4km	Rp 1.030.700,-	946	13,2165
2	4	424,7km	Rp 1.012.350,-	866	12,6999
3	4	482,6km	Rp 1.041.300,-	883	13,4021
4	4	493,8km	Rp 1.046.900,-	897	15,6213
5	5	467,6km	Rp 1.123.800,-	902	13,0063
Rata-rata			Rp 1.051.010,-	898,8	13,5892

Tabel 4.44 Hasil Percobaan Uji $P_c=0,9; P_m=0,3$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	3	528,4km	Rp 864.200,-	596	12,9005
2	4	606,4km	Rp 1.103.200,-	855	13,28
3	4	602,5km	Rp 1.102.250,-	944	12,8935
4	4	521km	Rp 1.060.500,-	901	12,68
5	4	585,3km	Rp 1.092.650,-	916	12,1122
Rata-rata			Rp 1.044.560,-	842,4	12,7732

Tabel 4.45 Hasil Percobaan Uji $P_c=0,9; P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	584,5km	Rp 1.292.250,-	438	13,4557
2	4	639,5km	Rp 1.119.750,-	455	13,3524
3	5	616,9km	Rp 1.308.450,-	640	12,4928
4	5	608km	Rp 1.304.000,-	898	14,1036
5	4	612,5km	Rp 1.106.250,-	813	11,6652
Rata-rata			Rp 1.226.140,-	648,8	13,0139

Tabel 4.46 Hasil Percobaan Uji $P_c=0,9; P_m=0,7$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	583,9km	Rp 1.291.950,-	140	8,7985
2	5	580,2km	Rp 1.290.100,-	899	12,1148
3	4	707,7km	Rp 1.153.850,-	299	11,7595
4	4	618,1km	Rp 1.109.050,-	596	10,2943
5	5	611,2km	Rp 1.305.600,-	653	11,7082
Rata-rata			Rp 1.230.110,-	517,4	10,9351

Tabel 4.47 Hasil Percobaan Uji $P_c=0,9; P_m=0,9$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	5	637,8km	Rp 1.318.900,-	976	9,2524
2	4	641km	Rp 1.120.500,-	353	11,6476
3	4	644,3km	Rp 1.122.150,-	59	11,2562
4	5	664,3km	Rp 1.332.150,-	677	11,5957
5	5	566,7km	Rp 1.283.350,-	842	11,0119
Rata-rata			Rp 1.235.410,-	581,4	10,9528

B2. Data Uji Percobaan Perubahan Parameter Algoritma *Cat Swarm Optimization* (CSO) pada 32 Toko

- *Mixture Ratio* (MR)

Tabel 4.48 Hasil Percobaan Uji MR=0,1; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	548,8km	Rp 1.074.400,-	299	261,195
2	4	564,5km	Rp 1.082.250,-	638	268,5693
3	4	607km	Rp 1.103.500,-	28	290,1646
4	4	595,2km	Rp 1.097.600,-	874	286,2859
5	4	560km	Rp 1.080.000,-	583	288,4938
Rata-rata			Rp 1.087.550,-	484,4	278,9417

Tabel 4.49 Hasil Percobaan Uji MR=0,3; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	565,3km	Rp 1.082.650,-	995	237,038
2	4	657,5km	Rp 1.128.750,-	739	239,013
3	4	610,2km	Rp 1.105.100,-	596	237,9402
4	4	659,9km	Rp 1.129.950,-	343	235,4485
5	4	601,4km	Rp 1.100.700,-	86	235,6929
Rata-rata			Rp 1.109.430,-	551,8	237,0265

Tabel 4.50 Hasil Percobaan Uji MR=0,5; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	633,1km	Rp 1.116.550,-	507	186,3718
2	4	614,4km	Rp 1.107.200,-	832	186,2477
3	4	668km	Rp 1.134.000,-	973	152,2638
4	4	623,4km	Rp 1.111.700,-	887	166,3272
5	4	687,8km	Rp 1.143.900,-	85	208,6399
Rata-rata			Rp 1.122.670,-	656,8	179,9701

Tabel 4.51 Hasil Percobaan Uji MR=0,7; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	636,9km	Rp 1.118.450,-	662	161,3294
2	4	626,3km	Rp 1.113.150,-	765	175,3932
3	4	630,1km	Rp 1.115.050,-	595	169,4906
4	4	634,1km	Rp 1.117.050,-	448	140,9299
5	4	630,1km	Rp 1.115.050,-	714	150,7381
Rata-rata			Rp 1.115.750,-	636,8	159,5762

Tabel 4.52 Hasil Percobaan Uji $MR=0,9$; $c = 2$; $SMP = 5$; $CDC=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	671,8km	Rp 1.135.900,-	41	88,563
2	4	633,5km	Rp 1.116.750,-	529	139,9411
3	4	608,3km	Rp 1.104.150,-	778	88,6017
4	4	615,5km	Rp 1.107.750,-	565	85,9891
5	4	673,1km	Rp 1.136.550,-	973	87,2621
Rata-rata			Rp 1.120.220,-	577,2	100,0714

- c (Konstanta Tracking)

Tabel 4.53 Hasil Percobaan Uji $MR=0,1$; $c = 0,1$; $SMP = 5$; $CDC=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	607,2km	Rp 1.103.600,-	844	298,2631
2	4	554,8km	Rp 1.077.400,-	223	330,9301
3	4	565,6km	Rp 1.082.800,-	81	367,0177
4	4	575,1km	Rp 1.087.550,-	463	339,3768
5	4	595,9km	Rp 1.097.950,-	268	351,243
Rata-rata			Rp 1.089.860,-	375,8	337,3661

Tabel 4.54 Hasil Percobaan Uji $MR=0,1$; $c = 0,5$; $SMP = 5$; $CDC=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	595,5km	Rp 1.097.750,-	381	350,6778
2	4	600,4km	Rp 1.100.200,-	988	346,176
3	4	594,1km	Rp 1.097.050,-	392	347,5618
4	4	533,4km	Rp 1.066.700,-	919	349,8021
5	4	606,3km	Rp 1.103.150,-	241	321,8444
Rata-rata			Rp 1.092.970,-	584,2	343,2124

Tabel 4.55 Hasil Percobaan Uji $MR=0,1$; $c = 1$; $SMP = 5$; $CDC=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	562,1km	Rp 1.081.050,-	815	355,9943
2	4	584,4km	Rp 1.092.200,-	269	345,3417
3	4	615,5km	Rp 1.107.750,-	413	332,7952
4	4	629,1km	Rp 1.114.550,-	484	329,0014
5	4	583km	Rp 1.091.500,-	263	321,0664
Rata-rata			Rp 1.097.410,-	448,8	336,8398

Tabel 4.56 Hasil Percobaan Uji $MR=0,1$; $c = 1,5$; $SMP = 5$; $CDC=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	567,5km	Rp 1.083.750,-	60	235,348
2	4	587,9km	Rp 1.093.950,-	81	309,6
3	4	597,3km	Rp 1.098.650,-	413	311,7301
4	4	593,6km	Rp 1.096.800,-	760	276,9402
5	4	633,9km	Rp 1.116.950,-	962	311,199
Rata-rata			Rp 1.098.020,-	455,2	288,9635

Tabel 4.57 Hasil Percobaan Uji MR=0,1; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	575,6km	Rp 1.087.800,-	257	244,2025
2	4	618,4km	Rp 1.109.200,-	819	310,0259
3	4	555km	Rp 1.077.500,-	92	308,143
4	4	646,9km	Rp 1.123.450,-	556	232,5955
5	4	602,6km	Rp 1.101.300,-	954	236,4034
Rata-rata			Rp 1.099.850,-	535,6	266,2741

- Counts of Dimension to Change (CDC)

Tabel 4.58 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,1

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	3	360km	Rp 780.000,-	113	228,4418
2	4	369,5km	Rp 984.750,-	698	218,6765
3	3	428,4km	Rp 814.200,-	989	218,4522
4	4	410,9km	Rp 1.005.450,-	400	211,3537
5	3	399km	Rp 799.500,-	153	210,7068
Rata-rata			Rp 876.780,-	470,6	217,5262

Tabel 4.59 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,3

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	473,1km	Rp 1.036.550,-	801	231,111
2	4	494km	Rp 1.047.000,-	956	247,7831
3	3	541,7km	Rp 870.850,-	218	222,1854
4	4	505,9km	Rp 1.052.950,-	974	283,5626
5	4	466,2km	Rp 1.033.100,-	981	289,1699
Rata-rata			Rp 1.007.990,-	786	254,7624

Tabel 4.60 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	562,1km	Rp 1.081.050,-	480	285,6758
2	4	586,2km	Rp 1.093.100,-	923	297,5798
3	4	5852,6km	Rp 1.091.300,-	741	294,3814
4	4	566,4km	Rp 1.083.200,-	468	305,9182
5	4	597,4km	Rp 1.098.700,-	231	301,0963
Rata-rata			Rp 1.089.470,-	568,6	296,9303

Tabel 4.61 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,7

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	641,2km	Rp 1.120.600,-	572	291,6445
2	4	600,2km	Rp 1.100.100,-	958	299,722
3	4	595,8km	Rp 1.097.900,-	870	317,7808
4	4	657,7km	Rp 1.128.850,-	272	303,2987
5	4	606,7km	Rp 1.103.350,-	92	298,9549
Rata-rata			Rp 1.110.160,-	552,8	302,2802

Tabel 4.62 Hasil Percobaan Uji $MR=0,1$; $c = 0,1$; $SMP = 5$; $CDC=0,9$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	4	550,8km	Rp 1.075.400,-	752	311,7493
2	4	612,4km	Rp 1.106.200,-	881	293,788
3	4	649km	Rp 1.124.500,-	915	299,7517
4	4	619,6km	Rp 1.109.800,-	463	293,1007
5	4	526,1km	Rp 1.063.050,-	892	317,9061
Rata-rata			Rp 1.095.790,-	780,6	303,2592

B3. Data Uji Percobaan Perubahan Parameter Algoritma Genetika pada 57 Toko

- (Probabilitas *Crossover*) P_c

Tabel 4.63 Hasil Percobaan Uji $P_c=0,1$; $P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	14	1.278km	Rp 3.439.000,-	953	5,9392
2	13	1.355km	Rp 3.277.500,-	752	5,5518
3	13	1.279,4km	Rp 3.239.700,-	515	5,5
4	14	1.265km	Rp 3.432.500,-	317	5,4324
5	14	1.363,6km	Rp 3.481.800,-	792	6,1046
Rata-rata			Rp 3.374.100,-	665,8	5,7056

Tabel 4.64 Hasil Percobaan Uji $P_c=0,3$; $P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.340,2km	Rp 3.270.100,-	88	9,0655
2	13	1.474,1km	Rp 3.337.050,-	698	9,1566
3	14	1.257,6km	Rp 3.428.800,-	951	9,1072
4	13	1.316,9km	Rp 3.258.450,-	893	9,1918
5	13	1.367,3km	Rp 3.283.650,-	847	8,9712
Rata-rata			Rp 3.315.610,-	695,3	9,0985

Tabel 4.65 Hasil Percobaan Uji $P_c=0,5$; $P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.316,8km	Rp 3.258.400,-	890	12,5582
2	14	1.259,3km	Rp 3.429.650,-	120	13,0424
3	13	1.329,3km	Rp 3.264.650,-	624	12,1869
4	13	1.304,6km	Rp 3.252.300,-	954	12,387
5	13	1.339,6km	Rp 3.269.800,-	766	12,0038
Rata-rata			Rp 3.294.960,-	670,8	12,4357

Tabel 4.66 Hasil Percobaan Uji $P_c=0,7$; $P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.380,9km	Rp 3.290.450,-	829	15,7722
2	13	1.321,5km	Rp 3.260.750,-	857	16,7034
3	13	1.393,4km	Rp 3.296.700,-	170	15,5111
4	13	1.254,3km	Rp 3.227.150,-	682	16,1763
5	13	1.236,5km	Rp 3.218.250,-	321	15,1981
Rata-rata			Rp 3.258.660,-	571,8	15,8722

Tabel 4.67 Hasil Percobaan Uji $P_c=0,9$; $P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	12	1.405km	Rp 3.102.500,-	921	18,2359
2	13	1.438,5km	Rp 3.319.250,-	824	19,0523
3	13	1.325,1km	Rp 3.262.550,-	655	18,0116
4	12	1.355,9km	Rp 3.077.950,-	341	18,8944
5	13	1.226,5km	Rp 3.213.250,-	526	18,4665
Rata-rata			Rp 3.195.100,-	653,4	18,5321

- (Probabilitas Mutasi) P_m

Tabel 4.68 Hasil Percobaan Uji $P_c=0,9$; $P_m=0,1$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	12	1.076,8km	Rp 2.938.400,-	991	14,4798
2	12	1.052,2km	Rp 2.926.100,-	641	14,3731
3	12	1.103,9km	Rp 2.951.950,-	789	14,7816
4	11	1.092km	Rp 2.746.000,-	818	14,6738
5	12	1.132,7km	Rp 2.966.350,-	984	14,723
Rata-rata			Rp 2.905.760,-	844,6	14,6063

Tabel 4.69 Hasil Percobaan Uji $P_c=0,9$; $P_m=0,3$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.216,4km	Rp 3.208.200,-	895	14,8579
2	12	1.204,5km	Rp 3.002.250,-	349	14,4547
3	13	1.260,9km	Rp 3.230.450,-	910	14,5878
4	12	1.319,3km	Rp 3.059.650,-	724	14,7064
5	13	1.270,7km	Rp 3.235.350,-	110	14,813
Rata-rata			Rp 3.147.180,-	597,6	14,6840

Tabel 4.70 Hasil Percobaan Uji $P_c=0,9$; $P_m=0,5$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	14	1.296,2km	Rp 3.448.100,-	364	19,439
2	14	1.285,6km	Rp 3.442.800,-	388	17,6565
3	13	1.256,2km	Rp 3.228.100,-	411	19,3597
4	13	1.316,3km	Rp 3.258.150,-	645	19,5228
5	12	1.297,5km	Rp 3.248.750,-	149	20,3137
Rata-rata			Rp 3.325.180,-	391,4	19,2583

Tabel 4.71 Hasil Percobaan Uji $P_c=0,9$; $P_m=0,7$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	14	1.396,1km	Rp 3.498.050,-	572	19,1341
2	14	1.302,5km	Rp 3.451.250,-	626	19,049
3	13	1.334,9km	Rp 3.267.450,-	945	19,1968
4	14	1.322,7km	Rp 3.461.350,-	43	20,1115
5	12	1.413,7km	Rp 3.106.850,-	902	23,2779
Rata-rata			Rp 3.356.990,-	617,6	20,1539

Tabel 4.72 Hasil Percobaan Uji $P_c=0,9$; $P_m=0,9$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	14	1.292,5km	Rp 3.446.250,-	756	23,5228
2	13	1.353,1km	Rp 3.276.550,-	608	17,6496
3	13	1.204,1km	Rp 3.202.050,-	865	18,6465
4	14	1.279km	Rp 3.439.500,-	454	17,898
5	14	1.305km	Rp 3.452.500,-	119	18,3345
Rata-rata			Rp 3.363.370,-	560,4	19,2103

B4. Data Uji Percobaan Perubahan Parameter Algoritma *Cat Swarm Optimization* (CSO) pada 57 Toko

- *Mixture Ratio* (MR)

Tabel 4.73 Hasil Percobaan Uji MR=0,1; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.349,6km	Rp 3.274.800,-	957	521,1762
2	12	1.356,1km	Rp 3.078.050,-	507	496,6898
3	13	1.271,2km	Rp 3.235.600,-	760	491,7081
4	13	1.233,7km	Rp 3.216.850,-	785	476,0314
5	13	1.293km	Rp 3.246.500,-	572	510,679
Rata-rata			Rp 3.210.360,-	716,2	499,3169

Tabel 4.74 Hasil Percobaan Uji MR=0,3; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.331,7km	Rp 3.265.850,-	86	429,9536
2	13	1.281,8km	Rp 3.240.900,-	433	438,0457
3	12	1.396km	Rp 3.098.000,-	918	436,6969
4	13	1.286,1km	Rp 3.243.050,-	666	426,2699
5	13	1.266,1km	Rp 3.233.050,-	679	442,2289
Rata-rata			Rp 3.216.170,-	556,4	434,6390

Tabel 4.75 Hasil Percobaan Uji MR=0,5; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.287,5km	Rp 3.243.750,-	606	357,8119
2	13	1.356km	Rp 3.278.000,-	472	340,0137
3	14	1.181km	Rp 3.390.500,-	993	335,38
4	13	1.381km	Rp 3.290.500,-	667	339,2951
5	13	1.326,5km	Rp 3.263.250,-	822	341,5966
Rata-rata			Rp 3.293.200,-	712	342,8195

Tabel 4.76 Hasil Percobaan Uji MR=0,7; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.342,5km	Rp 3.271.250,-	599	248,792
2	14	1.267,5km	Rp 3.433.750,-	949	268,1954
3	13	1.420,4km	Rp 3.310.200,-	641	245,1071
4	13	1.346,4km	Rp 3.273.200,-	528	248,9877
5	13	1.294,5km	Rp 3.247.250,-	585	249,5104
Rata-rata			Rp 3.307.130,-	652,4	252,1185

Tabel 4.77 Hasil Percobaan Uji MR=0,9; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	14	1.317,6km	Rp 3.458.800,-	899	161,4837
2	14	1.324,7km	Rp 3.462.350,-	448	155,7504
3	13	1.356,8km	Rp 3.278.400,-	213	153,9741
4	13	1.369,9km	Rp 3.284.950,-	331	160,0647
5	13	1.437,8km	Rp 3.318.900,-	27	160,2449
Rata-rata			Rp 3.360.680,-	383,6	158,3036

- c (Konstanta Tracking)

Tabel 4.78 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.234,2km	Rp 3.217.100,-	916	408,0646
2	12	1.311,3km	Rp 3.055.650,-	566	407,49
3	13	1.182,5km	Rp 3.191.250,-	478	410,5758
4	13	1.253,4km	Rp 3.226.700,-	901	405,7707
5	12	1.338,8km	Rp 3.069.400,-	500	418,7864
Rata-rata			Rp 3.152.020,-	672,2	410,1375

Tabel 4.79 Hasil Percobaan Uji MR=0,1; $c = 0,5$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.284,3km	Rp 3.242.150,-	366	405,7156
2	13	1.231,4km	Rp 3.215.700,-	938	405,2473
3	13	1.218km	Rp 3.209.000,-	753	401,976
4	12	1.360,6km	Rp 3.080.300,-	561	405,819
5	13	1.332,2km	Rp 3.266.100,-	655	403,1616
Rata-rata			Rp 3.202.650,-	654,6	404,3839

Tabel 4.80 Hasil Percobaan Uji MR=0,1; $c = 1$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.182,1km	Rp 3.191.050,-	77	531,6505
2	13	1.234,7km	Rp 3.217.350,-	944	494,4025
3	13	1.275,7km	Rp 3.237.850,-	875	518,928
4	13	1.333,3km	Rp 3.266.650,-	718	541,3005
5	13	1.264,4km	Rp 3.232.200,-	987	553,9958
Rata-rata			Rp 3.229.020,-	720,2	528,0555

Tabel 4.81 Hasil Percobaan Uji MR=0,1; $c = 1,5$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.334,2km	Rp 3.267.100,-	155	521,1341
2	13	1.221,7km	Rp 3.210.850,-	446	518,0231
3	13	1.301,7km	Rp 3.250.850,-	934	527,054
4	13	1.356,4km	Rp 3.278.200,-	848	552,5975
5	13	1.252,2km	Rp 3.226.100,-	695	634,9876
Rata-rata			Rp 3.246.620,-	615,6	550,7593

Tabel 4.82 Hasil Percobaan Uji MR=0,1; $c = 2$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.250km	Rp 3.225.000,-	156	650,5436
2	13	1.324,9km	Rp 3.262.450,-	593	503,3112
3	13	1.334,3km	Rp 3.267.150,-	727	517,5995
4	13	1.285,8km	Rp 3.242.900,-	583	502,3164
5	13	1.335,9km	Rp 3.267.950,-	764	500,9385
Rata-rata			Rp 3.253.090,-	564,6	534,9418

- Counts of Dimension to Change (CDC)

Tabel 4.83 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,1

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	10	771km	Rp 2.385.500,-	928	531,9344
2	10	749,5km	Rp 2.374.750,-	984	542,8234
3	11	756,4km	Rp 2.578.200,-	991	498,4076
4	11	783,6km	Rp 2.591.800,-	956	549,673
5	10	794,9km	Rp 2.397.450,-	772	540,2726
Rata-rata			Rp 2.465.540,-	926,2	532,6222

Tabel 4.84 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,3

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	12	1.329,6km	Rp 3.064.800,-	794	574,1691
2	12	1.281,2km	Rp 3.040.600,-	367	575,331
3	12	1.281,4km	Rp 3.040.700,-	764	574,6919
4	12	1.230,5km	Rp 3.015.250,-	536	497,5248
5	13	1.326,5km	Rp 3.263.250,-	538	476,9673
Rata-rata			Rp 3.084.920,-	599,8	539,7368

Tabel 4.85 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,5

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	12	1.356,8km	Rp 3.078.400,-	710	501,3088
2	12	1.317,5km	Rp 3.058.750,-	238	499,2044
3	13	1.308,5km	Rp 3.254.250,-	399	525,2265
4	13	1.314,7km	Rp 3.257.350,-	680	512,7888
5	13	1.276km	Rp 3.238.000,-	179	510,1846
Rata-rata			Rp 3.177.350,-	441,2	509,7426

Tabel 4.86 Hasil Percobaan Uji MR=0,1; $c = 0,1$; SMP = 5; CDC=0,7

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.273,6km	Rp 3.236.800,-	816	501,1984
2	12	1.297,4km	Rp 3.048.700,-	755	489,7225
3	13	1.239,2km	Rp 3.219.600,-	805	508,5136
4	13	1.303,7km	Rp 3.251.850,-	149	504,6837
5	13	1.324km	Rp 3.262.000,-	128	510,3359
Rata-rata			Rp 3.203.790,-	530,6	502,8908

Tabel 4.87 Hasil Percobaan Uji $MR=0,1$; $c = 0,1$; $SMP = 5$; $CDC=0,9$

No	Jumlah Kendaraan	Total Jarak Tempuh	Total Biaya	Iterasi Konvergen	Running Time
1	13	1.240,5km	Rp 3.220.250,-	354	500,3709
2	13	1.280,6km	Rp 3.240.300,-	27	587,0717
3	13	1.333,9km	Rp 3.266.950,-	602	528,478
4	13	1.262,8km	Rp 3.231.400,-	295	573,668
5	13	1.264,7km	Rp 3.232.350,-	484	571,4132
	Rata-rata		Rp 3.238.250,-	352,4	552,2004

B5. Data Uji Percobaan Algoritma Genetika pada 32 Toko

Tabel 4.88 Data Uji Percobaan Algoritma Genetika pada 32 Toko

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
1	R1: Depo-30-31-10-4-32-1-9-6-3-5-Depo R2: Depo-12-7-20-11-18-16-14-15-24-22-13-8-2-Depo R3: Depo-21-25-23-17-19-Depo R4: Depo-29-28-26-27-Depo	R1: 119,9 R2: 117 R3: 135,1 R4: 153 Total: 525	R1: 528 R2: 107 R3: 261 R4: 209 Total: 1105	R1: 09.07 R2: 11.15 R3: 11.32 R4: 08.51	R1: 16.25 R2: 16.22 R3: 16.58 R4: 13.49	1062500	9693	40,5141
2	R1: Depo-29-28-32-30-31-17-19-2-Depo R2: Depo-16-11-10-8-6-3-21-5-7-Depo R3: Depo-26-14-15-9-18-20-27-23-24-22-4-Depo R4: Depo-1-13-25-12-Depo	R1: 133,3 R2: 90,9 R3: 189,6 R4: 94 Total: 507,8	R1: 429 R2: 275 R3: 248 R4: 153 Total: 1105	R1: 09.53 R2: 11.19 R3: 09.47 R4: 11.59	R1: 16.36 R2: 16.21 R3: 16.51 R4: 15.30	1053900	9857	42,7615
3	R1: Depo-12-32-7-30-31-18-9-2-1-6-3-5-4-Depo R2: Depo-20-14-13-11-28-29-21-8-26-27-22-24-16-Depo R3: Depo-10-23-25-15-17-19-Depo	R1: 121,8 R2: 212,9 R3: 101,1 Total: 435,8	R1: 494 R2: 401 R3: 210 Total: 1105	R1: 08.59 R2: 07.45 R3: 12.11	R1: 16.16 R2: 16.55 R3: 16.41	817900	3480	39,8163
4	R1: Depo-12-3-13-20-16-14-15-27-22-25-9-Depo R2: Depo-1-2-10-17-19-5-Depo R3: Depo-28-21-8-11-18-30-31-6-7-Depo R4: Depo-32-29-4-23-24-26-Depo	R1: 141,3 R2: 49 R3: 128,6 R4: 167,5 Total: 486,4	R1: 195 R2: 320 R3: 439 R4: 151 Total: 1105	R1: 10.53 R2: 13.12 R3: 09.41 R4: 10.34	R1: 16.40 R2: 16.50 R3: 16.38 R4: 16.19	1043200	9845	41,9826
5	R1: Depo-7-12-1-8-28-29-32-21-Depo R2: Depo-30-11-20-23-26-14-13-6-5-3-4-19-Depo R3: Depo-15-22-24-25-27-Depo R4: Depo-2-10-16-9-18-31-17-Depo	R1: 130,7 R2: 176,7 R3: 113,6 R4: 74,1 Total: 495,1	R1: 421 R2: 282 R3: 27 R4: 375 Total: 1105	R1: 09.21 R2: 08.51 R3: 13.21 R4: 11.08	R1: 16.07 R2: 16.19 R3: 16.57 R4: 16.06	1047550	5783	40,8016
6	R1: Depo-11-20-1-2-10-28-29-32-19-Depo R2: Depo-31-16-15-23-26-27-22-13-18-17-Depo R3: Depo-7-9-12-21-6-3-5-8-Depo R4: Depo-30-14-25-24-4-Depo	R1: 127 R2: 152,8 R3: 73,9 R4: 105,2 Total: 458,9	R1: 393 R2: 323 R3: 254 R4: 135 Total: 1105	R1: 09.29 R2: 09.54 R3: 12.05 R4: 11.48	R1: 15.51 R2: 16.48 R3: 16.24 R4: 16.08	1029450	7754	43,0016
7	R1: Depo-11-1-29-28-21-12-6-5-27-14-15-16-18-2-Depo R2: Depo-8-10-9-24-13-7-Depo R3: Depo-30-31-23-25-22-26-Depo R4: Depo-20-4-3-32-19-17-Depo	R1: 177,5 R2: 69,7 R3: 153,9 R4: 97,4 Total: 498,5	R1: 469 R2: 156 R3: 192 R4: 288 Total: 1105	R1: 08.14 R2: 12.23 R3: 10.47 R4: 11.27	R1: 16.58 R2: 15.46 R3: 16.30 R4: 16.22	1049250	9626	41,2973
8	R1: Depo-11-28-32-2-12-31-16-26-22-27-7-Depo R2: Depo-21-29-23-24-18-30-Depo	R1: 228,1 R2: 141,4	R1: 374 R2: 141	R1: 07.46 R2: 11.13	R1: 16.59 R2: 16.27	1061200	9004	42,0681

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R3: Depo-1-5-3-6-17-19-10-Depo R4: Depo-8-4-9-20-14-15-13-25-Depo	R3: 50 R4: 102,9 Total: 522,4	R3: 368 R4: 222 Total: 1105	R3: 12.50 R4: 11.23	R3: 16.52 R4: 15.59			
9	R1: Depo-32-29-21-5-25-24-16-19-Depo R2: Depo-28-12-10-2-7-11-1-30-18-8-4-Depo R3: Depo-15-14-26-20-13-3-6-27-23-22-Depo R4: Depo-9-31-17-Depo	R1: 170 R2: 131,2 R3: 199,6 R4: 40,4 Total: 541,2	R1: 194 R2: 473 R3: 158 R4: 280 Total: 1105	R1: 10.02 R2: 08.39 R3: 09.53 R4: 12.51	R1: 16.46 R2: 16.14 R3: 16.50 R4: 16.00	1070600	8652	41,2334
10	R1: Depo-7-1-2-30-13-15-26-27-24-23-22-25-Depo R2: Depo-17-9-Depo R3: Depo-11-14-16-20-28-29-32-21-Depo R4: Depo-4-31-18-5-12-6-10-3-8-19-Depo	R1: 187,3 R2: 25,6 R3: 145,5 R4: 99 Total: 457,4	R1: 228 R2: 159 R3: 336 R4: 382 Total: 1105	R1: 09.36 R2: 14.36 R3: 09.55 R4: 10.35	R1: 16.39 R2: 16.28 R3: 16.27 R4: 16.29	1028700	9102	39,4669
Rata-rata						1026425	8279,6	41,2943

B6. Data Uji Percobaan Algoritma *Cat Swarm Optimization* (CSO) pada 32 Toko

Tabel 4.89 Data Uji Percobaan Algoritma *Cat Swarm Optimization* (CSO) pada 32 Toko

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
1	R1: Depo-4-7-1-2-9-18-20-13-11-10-8-5-3-Depo R2: Depo-14-15-16-17-19-Depo R3: Depo-12-21-29-28-32-6-Depo R4: Depo-31-30-23-26-27-22-24-25-Depo	R1: 64,6 R2: 57 R3: 92,1 R4: 169 Total: 383,7	R1: 452 R2: 159 R3: 296 R4: 198 Total: 1105	R1: 10.31 R2: 13.14 R3: 10.53 R4: 10.08	R1: 15.45 R2: 16.18 R3: 16.02 R4: 16.21	991350	8571	1694,9263
2	R1: Depo-28-29-21-5-3-8-1-Depo R2: Depo-31-30-15-14-11-9-6-12-32-7-4-Depo R3: Depo-20-13-16-26-25-27-22-24-23-10-Depo R4: Depo-18-17-19-2-Depo	R1: 80,4 R2: 119,3 R3: 149 R4: 38,7 Total: 387,4	R1: 457 R2: 325 R3: 180 R4: 143 Total: 1105	R1: 09.36 R2: 09.40 R3: 11.02 R4: 14.15	R1: 15.10 R2: 15.50 R3: 16.52 R4: 16.36	993700	1113	1711,9354
3	R1: Depo-2-1-21-12-6-3-5-8-4-19-17-Depo R2: Depo-30-31-18-20-13-24-22-23-9-Depo R3: Depo-28-29-32-7-10-11-26-27-25-14-15-16-Depo	R1: 92,5 R2: 93,3 R3: 207,4 Total: 393,2	R1: 524 R2: 234 R3: 347 Total: 1105	R1: 10.20 R2: 11.26 R3: 08.01	R1: 16.45 R2: 16.14 R3: 16.33	796600	441	1677,3417
4	R1: Depo-20-18-31-30-15-14-13-17-19-Depo R2: Depo-2-12-32-28-29-21-6-3-5-7-Depo	R1: 82,2 R2: 100,6	R1: 318 R2: 370	R1: 11.56 R2: 09.55	R1: 16.52 R2: 15.44	991250	943	1720,7602

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R3: Depo-1-4-8-24-22-23-10-9-Depo R4: Depo-11-16-27-26-25-Depo	R3: 71,3 R4: 128,4 Total: 382,5	R3: 337 R4: 80 Total: 1105	R3: 12.14 R4: 11.37	R3: 16.37 R4: 15.41			
5	R1: Depo-31-30-18-20-23-25-17-19-8-5-Depo R2: Depo-1-7-4-10-9-11-13-16-14-15-24-22-27-26-Depo R3: Depo-2-3-12-32-28-29-21-6-Depo	R1: 140,5 R2: 129,3 R3: 98,1 Total: 367,9	R1: 313 R2: 443 R3: 349 Total: 1105	R1: 10.32 R2: 10.17 R3: 11.04	R1: 16.59 R2: 16.57 R3: 16.39	783950	1863	1722,9405
6	R1: Depo-1-20-13-16-14-15-26-27-25-23-22-24-8-5-2-Depo R2: Depo-11-7-12-32-28-29-21-6-3-19-17-Depo R3: Depo-31-30-18-9-10-4-Depo	R1: 167,7 R2: 132,1 R3: 55,2 Total: 355	R1: 273 R2: 481 R3: 351 Total: 1105	R1: 09.37 R2: 08.58 R3: 10.32	R1: 16.40 R2: 16.31 R3: 14.36	777500	1536	1698,3022
7	R1: Depo-1-20-31-30-18-11-7-4-8-10-12-6-3-Depo R2: Depo-14-15-16-13-2-17-19-9-Depo R3: Depo-32-29-28-21-22-27-24-23-5-Depo R4: Depo-25-26-Depo	R1: 89,4 R2: 80,8 R3: 173,9 R4: 110,1 Total: 454,2	R1: 495 R2: 239 R3: 310 R4: 61 Total: 1105	R1: 09.33 R2: 12.31 R3: 09.30 R4: 13.09	R1: 15.50 R2: 16.45 R3: 16.53 R4: 16.03	1027100	9688	1685,0266
8	R1: Depo-26-27-22-23-10-Depo R2: Depo-28-29-32-6-3-5-8-4-19-17-Depo R3: Depo-1-7-12-21-25-24-16-18-Depo R4: Depo-31-30-15-14-13-20-11-9-2-Depo	R1: 107,3 R2: 106,4 R3: 127,6 R4: 74,2 Total: 415,5	R1: 116 R2: 457 R3: 289 R4: 243 Total: 1105	R1: 12.20 R2: 10.08 R3: 10.22 R4: 09.00	R1: 15.55 R2: 16.41 R3: 16.06 R4: 13.15	1007750	6288	1701,6723
9	R1: Depo-20-15-14-26-27-25-23-22-24-16-13-18-Depo R2: Depo-2-1-6-4-10-17-19-9-Depo R3: Depo-11-31-30-8-7-3-12-21-29-28-32-5-Depo	R1: 157,4 R2: 51,2 R3: 151,5 Total: 360,1	R1: 150 R2: 461 R3: 494 Total: 1105	R1: 10.38 R2: 12.22 R3: 08.43	R1: 16.50 R2: 16.53 R3: 16.57	780050	8200	1721,3114
10	R1: Depo-23-22-24-16-4-Depo R2: Depo-18-20-13-26-27-25-10-Depo R3: Depo-2-1-7-12-21-29-28-32-6-3-5-8-Depo R4: Depo-14-15-30-31-11-9-17-19-Depo	R1: 66,1 R2: 132,2 R3: 104,8 R4: 91,5 Total: 394,6	R1: 110 R2: 174 R3: 490 R4: 331 Total: 1105	R1: 13.24 R2: 12.07 R3: 09.55 R4: 10.52	R1: 16.30 R2: 16.57 R3: 16.48 R4: 16.18	997300	5210	1658,9558
Rata-rata						914655	4385,3	1699,3172

B7. Data Uji Percobaan Algoritma Genetika pada 57 Toko

Tabel 4.90 Data Uji Percobaan Algoritma Genetika pada 57 Toko

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
1	R1: Depo-7-Depo R2: Depo-1-Depo R3: Depo-20-54-57-55-31-33-56-52-6-Depo R4: Depo-44-22-23-25-24-21-29-12-2-Depo R5: Depo-46-41-43-30-5-19-Depo R6: Depo-8-10-51-42-47-4-Depo R7: Depo-18-15-9-28-26-49-39-37-48-50-Depo R8: Depo-14-16-27-45-3-17-Depo R9: Depo-32-Depo R10: Depo-13-35-36-34-40-38-53-11-Depo	R1: 7,4 R2: 2,4 R3: 142,9 R4: 82 R5: 106,5 R6: 87,9 R7: 127,1 R8: 59,9 R9: 94 R10: 116,5 Total: 826,6	R1: 600 R2: 600 R3: 415 R4: 400 R5: 340 R6: 564 R7: 325 R8: 583 R9: 600 R10: 271 Total: 4698	R1: 11.52 R2: 08.57 R3: 09.33 R4: 11.40 R5: 11.36 R6: 09.21 R7: 09.26 R8: 11.13 R9: 07.02 R10: 11.17	R1: 15.29 R2: 12.24 R3: 16.38 R4: 16.57 R5: 16.55 R6: 15.17 R7: 15.31 R8: 16.43 R9: 12.19 R10: 16.58	2413300	9778	62,7288
2	R1: Depo-47-43-30-56-Depo R2: Depo-32-Depo R3: Depo-53-38-35-23-22-40-34-Depo R4: Depo-1-Depo R5: Depo-3-45-10-11-2-Depo R6: Depo-8-18-44-27-21-13-Depo R7: Depo-4-19-20-46-9-6-52-5-14-48-50-Depo R8: Depo-7-Depo R9: Depo-28-26-29-36-24-49-25-16-Depo R10: Depo-54-33-57-31-17-Depo R11: Depo-41-42-55-51-15-12-37-39-Depo	R1: 88,2 R2: 94 R3: 136,3 R4: 2,4 R5: 48,7 R6: 59,8 R7: 99,6 R8: 7,4 R9: 77,3 R10: 116 R11: 149,5 Total: 879,2	R1: 89 R2: 600 R3: 164 R4: 600 R5: 520 R6: 592 R7: 330 R8: 600 R9: 363 R10: 545 R11: 295 Total: 4698	R1: 12.54 R2: 07.02 R3: 11.25 R4: 08.57 R5: 11.23 R6: 09.21 R7: 10.28 R8: 11.52 R9: 11.11 R10: 10.16 R11: 09.41	R1: 16.06 R2: 12.09 R3: 16.37 R4: 12.24 R5: 15.55 R6: 14.57 R7: 15.54 R8: 15.29 R9: 16.06 R10: 16.53 R11: 16.08	2639600	8005	61,9564
3	R1: Depo-31-57-5-17-Depo R2: Depo-35-34-28-2-24-23-25-29-3-Depo R3: Depo-41-42-16-26-18-11-Depo R4: Depo-20-19-51-9-15-54-33-Depo R5: Depo-8-10-13-38-40-48-Depo R6: Depo-4-46-45-44-39-37-36-50-Depo R7: Depo-12-14-49-53-30-43-47-52-6-Depo R8: Depo-1-Depo R9: Depo-32-Depo R10: Depo7-Depo R11: Depo-55-56-22-21-27-Depo	R1: 64,6 R2: 118 R3: 90,8 R4: 143,8 R5: 105,2 R6: 162,7 R7: 109,5 R8: 2,4 R9: 94 R10: 7,4 R11: 72,5 Total: 970,9	R1: 576 R2: 442 R3: 526 R4: 123 R5: 579 R6: 269 R7: 317 R8: 600 R9: 600 R10: 600 R11: 66 Total: 4698	R1: 11.20 R2: 09.20 R3: 09.15 R4: 10.29 R5: 09.21 R6: 10.25 R7: 11.09 R8: 08.57 R9: 07.02 R10: 11.52 R11: 12.20	R1: 16.44 R2: 15.58 R3: 15.12 R4: 15.34 R5: 15.55 R6: 16.27 R7: 16.54 R8: 12.24 R9: 12.19 R10: 15.29 R11: 15.13	2685450	8479	68,7378
4	R1: Depo-50-36-35-45-27-Depo	R1: 64	R1: 142	R1: 13.27	R1: 16.32	2693300	9577	61,9046

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R2: Depo-7-Depo R3: Depo-4-2-10-20-46-5-11-21-3-Depo R4: Depo-41-19-14-39-40-37-34-24-12-Depo R5: Depo-6-55-51-43-42-17-Depo R6: Depo-1-Depo R7: Depo-25-26-56-31-30-54-Depo R8: Depo-13-44-29-15-28-23-53-52-Depo R9: Depo-9-22-47-57-33-49-16-Depo R10: Depo-32-Depo R11: Depo-8-48-18-Depo	R2: 7,4 R3: 89,7 R4: 212,3 R5: 89,3 R6: 2,4 R7: 114,2 R8: 130,5 R9: 157,4 R10: 94 R11: 25,4 Total: 986,6	R2: 600 R3: 539 R4: 391 R5: 403 R6: 600 R7: 446 R8: 176 R9: 271 R10: 600 R11: 530 Total: 4698	R2: 11.52 R3: 10.39 R4: 07.50 R5: 11.17 R6: 08.57 R7: 10.35 R8: 11.17 R9: 10.01 R10: 07.02 R11: 09.21	R2: 15.29 R3: 16.55 R4: 16.27 R5: 16.43 R6: 12.24 R7: 16.54 R8: 16.49 R9: 15.55 R10: 12.19 R11: 13.20			
5	R1: Depo-47-30-35-34-40-37-39-23-Depo R2: Depo-15-33-54-12-18-16-11-Depo R3: Depo-20-51-55-57-56-17-13-Depo R4: Depo-7-Depo R5: Depo-32-Depo R6: Depo-8-49-21-22-48-50-Depo R7: Depo-1-Depo R8: Depo-53-38-36-25-26-29-Depo R9: Depo-9-10-6-41-31-43-52-14-Depo R10: Depo-46-42-5-27-28-3-Depo R11: Depo-19-4-2-45-44-24-Depo	R1: 185,8 R2: 106,9 R3: 76,3 R4: 7,4 R5: 94 R6: 50,2 R7: 2,4 R8: 102,5 R9: 115 R10: 102 R11: 78,5 Total: 921	R1: 222 R2: 307 R3: 419 R4: 600 R5: 600 R6: 575 R7: 600 R8: 297 R9: 466 R10: 166 R11: 446 Total: 4698	R1: 09.54 R2: 10.39 R3: 11.44 R4: 11.52 R5: 07.02 R6: 09.46 R7: 08.57 R8: 11.25 R9: 09.27 R10: 11.54 R11: 10.03	R1: 16.31 R2: 15.43 R3: 16.57 R4: 15.29 R5: 12.19 R6: 14.49 R7: 12.24 R8: 16.33 R9: 15.52 R10: 16.26 R11: 15.21	2660500	9445	61,9771
6	R1: Depo-46-42-41-43-47-31-51-Depo R2: Depo-8-40-39-35-13-50-Depo R3: Depo-18-2-Depo-36-34-37-38 R4: Depo-49-3-10-20-19-9-52-56-55-22-23-Depo R5: Depo-7-Depo R6: Depo-17-Depo R7: Depo-2-54-30-53-12-48-Depo R8: Depo-57-33-6-5-11-Depo R9: Depo-1-Depo R10: Depo-28-26-25-21-44-45-24-29-Depo R11: Depo-32-Depo R12: Depo-4-16-14-15-Depo	R1: 97,3 R2: 109,9 R3: 105 R4: 129,9 R5: 7,4 R6: 9 R7: 98 R8: 105,6 R9: 2,4 R10: 85,9 R11: 94 R12: 24,7 Total: 869,1	R1: 302 R2: 555 R3: 168 R4: 125 R5: 600 R6: 271 R7: 463 R8: 415 R9: 600 R10: 402 R11: 600 R12: 197 Total: 4698	R1: 11.18 R2: 09.44 R3: 11.54 R4: 10.48 R5: 11.52 R6: 14.49 R7: 10.57 R8: 10.01 R9: 08.57 R10: 11.06 R11: 07.02 R12: 11.32	R1: 16.29 R2: 16.21 R3: 16.24 R4: 16.10 R5: 15.29 R6: 16.43 R7: 16.45 R8: 15.28 R9: 12.24 R10: 16.25 R11: 12.19 R12: 13.38	2834550	9922	62,2872
7	R1: Depo-2-11-41-43-Depo R2: Depo-34-38-36-40-39-23-22-13-Depo R3: Depo-7-Depo R4: Depo-3-45-9-54-24-21-25-50-Depo	R1: 59,5 R2: 159,1 R3: 7,4 R4: 87,9	R1: 519 R2: 122 R3: 600 R4: 315	R1: 11.54 R2: 10.27 R3: 11.52 R4: 11.23	R1: 16.41 R2: 16.25 R3: 15.29 R4: 16.10	2636300	9830	62,6446

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R5: Depo-6-10-20-15-14-33-30-56-51-55-31-52-4-Depo R6: Depo-19-47-42-57-53-37-12-Depo R7: Depo-32-Depo R8: Depo-1-Depo R9: Depo-8-Depo R10: Depo-16-18-46-5-Depo R11: Depo-49-35-44-29-27-26-28-48-17-Depo	R5: 146,4 R6: 147,1 R7: 94 R8: 2,4 R9: 7,4 R10: 63,3 R11: 98,1 Total: 872,6	R5: 427 R6: 291 R7: 600 R8: 600 R9: 518 R10: 208 R11: 498 Total: 4698	R5: 09.15 R6: 09.18 R7: 07.02 R8: 08.57 R9: 09.21 R10: 11.51 R11: 11.23	R5: 16.53 R6: 15.42 R7: 12.19 R8: 12.24 R9: 12.32 R10: 15.10 R11: 16.43			
8	R1: Depo-1-Depo R2: Depo-20-10-14-15-6-46-5-19-3-4-Depo R3: Depo-33-30-56-57-31-52-11-Depo R4: Depo-16-18-25-49-48-12-13-17-Depo R5: Depo-32-Depo R6: Depo-7-Depo R7: Depo-8-36-34-35-22-23-50-Depo R8: Depo-9-53-54-51-55-43-47-42-41-Depo R9: Depo-38-39-37-40-24-21-27-Depo R10: Depo-28-26-45-44-29-2-Depo	R1: 2,4 R2: 90,3 R3: 109,6 R4: 34,5 R5: 94 R6: 7,4 R7: 70,6 R8: 114,3 R9: 133,1 R10: 60,4 Total: 716,6	R1: 600 R2: 358 R3: 414 R4: 484 R5: 600 R6: 600 R7: 598 R8: 318 R9: 154 R10: 572 Total: 4698	R1: 08.57 R2: 10.12 R3: 11.04 R4: 12.26 R5: 07.02 R6: 11.52 R7: 08.59 R8: 11.04 R9: 11.42 R10: 10.46	R1: 12.24 R2: 15.49 R3: 16.59 R4: 16.43 R5: 12.19 R6: 15.29 R7: 15.03 R8: 16.55 R9: 16.35 R10: 16.05	2358300	8397	63,4609
9	R1: Depo-51-31-43-55-42-10-Depo R2: Depo-15-40-38-24-29-19-Depo R3: Depo-41-46-33-54-5-2-Depo R4: Depo-16-20-9-6-17-Depo R5: Depo-12-48-8-Depo R6: Depo-49-53-35-34-36-39-37-21-50-Depo R7: Depo-32-Depo R8: Depo-1-Depo R9: Depo-28-23-22-57-56-30-47-52-11-Depo R10: Depo-7-Depo R11: Depo-14-18-25-26-45-44-27-4-13-3-Depo	R1: 99,5 R2: 16,5 R3: 155,2 R4: 39,1 R5: 23,6 R6: 118 R7: 94 R8: 2,4 R9: 94,6 R10: 7,4 R11: 96 Total: 746,3	R1: 136 R2: 89 R3: 560 R4: 529 R5: 538 R6: 240 R7: 600 R8: 600 R9: 343 R10: 600 R11: 463 Total: 4698	R1: 11.40 R2: 12.21 R3: 08.37 R4: 12.09 R5: 11.57 R6: 11.05 R7: 07.00 R8: 08.57 R9: 10.56 R10: 11.52 R11: 09.37	R1: 15.50 R2: 16.27 R3: 16.19 R4: 16.43 R5: 15.56 R6: 16.31 R7: 14.08 R8: 12.24 R9: 16.15 R10: 15.29 R11: 15.57	2573150	9691	65,501
10	R1: Depo-7-Depo R2: Depo-44-23-45-16-17-13-Depo R3: Depo-12-49-14-9-6-52-11-28-Depo R4: Depo-57-42-41-55-31-18-27-3-Depo R5: Depo-32-Depo R6: Depo-1-Depo R7: Depo-8-54-33-56-43-Depo R8: Depo-2-10-51-30-47-5-19-4-Depo R9: Depo-46-20-53-21-24-48-50-Depo	R1: 7,4 R2: 74,7 R3: 75,1 R4: 116,2 R5: 94 R6: 2,4 R7: 127,6 R8: 99,6 R9: 95,6	R1: 600 R2: 546 R3: 321 R4: 436 R5: 600 R6: 600 R7: 584 R8: 464 R9: 128	R1: 11.52 R2: 11.06 R3: 11.14 R4: 08.59 R5: 07.02 R6: 08.57 R7: 09.21 R8: 10.57 R9: 10.57	R1: 15.29 R2: 16.57 R3: 15.52 R4: 15.33 R5: 12.19 R6: 12.24 R7: 16.38 R8: 16.51 R9: 14.46	2652000	8823	62,825

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R10: Depo-15-39-37-36-40-22-Depo R11: Depo-26-29-25-38-34-35-Depo	R10: 124,4 R11: 87 Total: 904	R10: 140 R11: 279 Total: 4698	R10: 12.02 R11: 11.09	R10: 16.49 R11: 15.33			
Rata-rata						2614645	9194,7	63,4023

B8. Data Uji Percobaan Algoritma *Cat Swarm Optimization* (CSO) pada 57 Toko

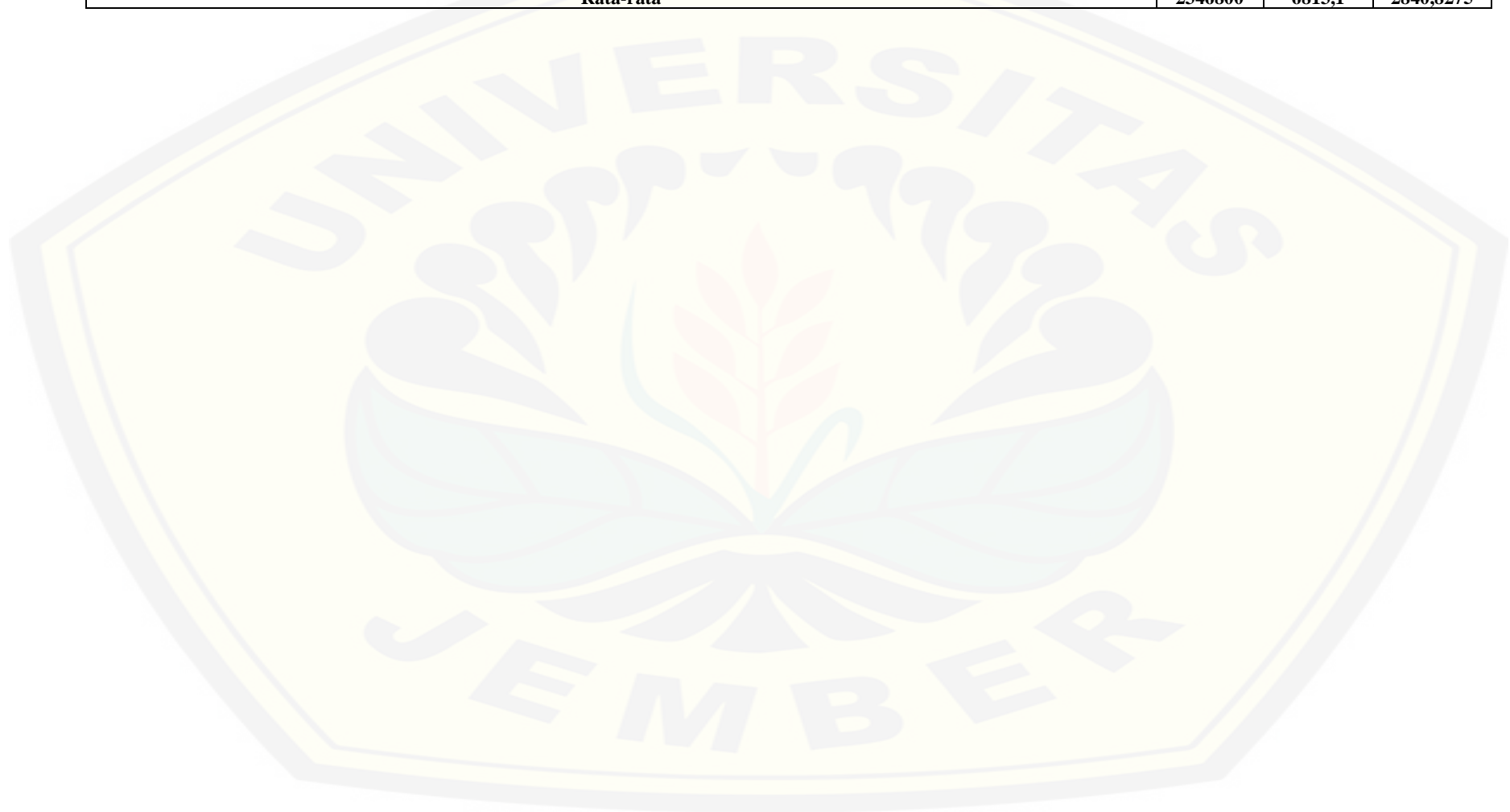
Tabel 4.91 Data Uji Percobaan Algoritma *Cat Swarm Optimization* (CSO) pada 57 Toko

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
1	R1: Depo-32-Depo R2: Depo-6-9-15-18-27-45-44-26-28-Depo R3: Depo-14-35-38-39-37-40-34-36-24-50-Depo R4: Depo-7-Depo R5: Depo-16-13-17-Depo R6: Depo-12-48-49-23-22-56-30-31-55-51-Depo R7: Depo-8-Depo R8: Depo-2-11-52-47-43-42-5-10-Depo R9: Depo-1-Depo R10: Depo-19-20-46-41-57-33-54-53-21-25-29-3-4-Depo	R1: 94 R2: 76 R3: 138,2 R4: 7,4 R5: 12,8 R6: 87,4 R7: 7,4 R8: 66,4 R9: 2,4 R10: 170,6 Total: 662,6	R1: 600 R2: 491 R3: 214 R4: 600 R5: 405 R6: 252 R7: 518 R8: 462 R9: 600 R10: 556 Total: 4698	R1: 07.00 R2: 10.17 R3: 10.42 R4: 11.52 R5: 13.53 R6: 11.57 R7: 09.21 R8: 11.54 R9: 08.57 R10: 08.11	R1: 14.08 R2: 16.02 R3: 16.39 R4: 15.29 R5: 16.43 R6: 16.43 R7: 12.32 R8: 16.59 R9: 12.24 R10: 16.59	2331300	5934	2801,9724
2	R1: Depo-46-42-41-55-51-52-20-5-2-Depo R2: Depo-32-Depo R3: Depo-1-Depo R4: Depo-7-Depo R5: Depo-8-10-19-3-4-Depo R6: Depo-28-26-16-11-43-47-31-Depo R7: Depo-53-54-33-30-56-57-Depo R8: Depo-13-12-38-39-37-40-50-Depo R9: Depo-14-36-34-35-49-48-24-21-44-45-27-Depo R10: Depo-9-6-15-29-23-22-25-18-17-Depo	R1: 87,4 R2: 94 R3: 2,4 R4: 7,4 R5: 32,2 R6: 100,4 R7: 121,4 R8: 121,6 R9: 110 R10: 62,1 Total: 738,9	R1: 523 R2: 600 R3: 600 R4: 600 R5: 599 R6: 505 R7: 309 R8: 145 R9: 287 R10: 530 Total: 4698	R1: 10.23 R2: 07.02 R3: 08.57 R4: 11.52 R5: 09.21 R6: 10.46 R7: 11.25 R8: 11.17 R9: 11.03 R10: 11.18	R1: 16.46 R2: 12.19 R3: 12.24 R4: 15.29 R5: 14.00 R6: 16.52 R7: 16.51 R8: 15.57 R9: 16.38 R10: 16.43	2369450	9932	2769,6885
3	R1: Depo-32-Depo R2: Depo-2-46-42-43-55-Depo R3: Depo-8-49-35-36-23-24-18-Depo R4: Depo-7-Depo R5: Depo-3-28-26-27-29-21-48-12-13-17-Depo	R1: 94 R2: 80,3 R3: 68,2 R4: 7,4 R5: 55	R1: 600 R2: 272 R3: 544 R4: 600 R5: 557	R1: 07.00 R2: 10.58 R3: 09.21 R4: 11.52 R5: 11.23	R1: 14.08 R2: 15.12 R3: 14.56 R4: 15.29 R5: 16.48	2396450	5560	2783,8023

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R6: Depo-1-Depo R7: Depo-20-41-47-22-44-45-25-38-53-50-Depo R8: Depo-9-15-14-54-33-51-52-5-19-4-Depo R9: Depo-34-40-39-37-16-Depo R10: Depo-6-10-57-56-30-31-11-Depo	R6: 2,4 R7: 155,6 R8: 122,4 R9: 106,7 R10: 100,9 Total: 792,9	R6: 600 R7: 467 R8: 265 R9: 255 R10: 517 Total: 4698	R6: 08.57 R7: 08.04 R8: 10.14 R9: 12.05 R10: 10.05	R6: 12.24 R7: 15.54 R8: 15.45 R9: 16.51 R10: 16.58			
4	R1: Depo-4-3-19-8-Depo R2: Depo-32-Depo R3: Depo-18-23-22-54-53-38-39-37-40-48-50-Depo R4: Depo-6-9-15-33-30-47-43-41-5-10-Depo R5: Depo-1-Depo R6: Depo-28-26-25-29-27-45-44-16-11-Depo R7: Depo-20-46-2-42-55-31-56-57-51-52-Depo R8: Depo-14-49-35-34-36-24-21-12-13-17-Depo R9: Depo-7-Depo	R1: 27,2 R2: 94 R3: 161,2 R4: 124 R5: 2,4 R6: 73,7 R7: 143,3 R8: 73,3 R9: 7,4 Total: 706,5	R1: 591 R2: 600 R3: 265 R4: 500 R5: 600 R6: 599 R7: 515 R8: 428 R9: 600 Total: 4698	R1: 11.08 R2: 07.02 R3: 10.00 R4: 09.37 R5: 08.57 R6: 10.46 R7: 09.20 R8: 11.36 R9: 11.52	R1: 15.31 R2: 12.19 R3: 16.42 R4: 16.38 R5: 12.24 R6: 16.56 R7: 16.55 R8: 16.55 R9: 15.29	2153250	9243	2785,3309
5	R1: Depo-1-Depo R2: Depo-20-10-14-15-6-46-5-19-3-4-Depo R3: Depo-33-30-56-57-31-52-11-Depo R4: Depo-16-18-25-49-48-12-13-17-Depo R5: Depo-32-Depo R6: Depo-7-Depo R7: Depo-8-36-34-35-22-23-50-Depo R8: Depo-9-53-54-51-55-43-47-42-41-Depo R9: Depo-38-39-37-40-24-21-27-Depo R10: Depo-28-26-45-44-29-2-Depo	R1: 2,4 R2: 90,3 R3: 109,6 R4: 34,5 R5: 94 R6: 7,4 R7: 70,6 R8: 114,3 R9: 133,1 R10: 60,4 Total: 716,6	R1: 600 R2: 358 R3: 414 R4: 484 R5: 600 R6: 600 R7: 598 R8: 318 R9: 154 R10: 572 Total: 4698	R1: 08.57 R2: 10.12 R3: 11.04 R4: 12.26 R5: 07.02 R6: 11.52 R7: 08.59 R8: 11.04 R9: 11.42 R10: 10.46	R1: 12.24 R2: 15.49 R3: 16.59 R4: 16.43 R5: 12.19 R6: 15.29 R7: 15.03 R8: 16.55 R9: 16.35 R10: 16.05	2358300	1419	2877,9223
6	R1: Depo-32-Depo R2: Depo-6-9-15-18-27-45-44-26-28-Depo R3: Depo-14-35-38-39-37-40-34-36-24-50-Depo R4: Depo-7-Depo R5: Depo-16-13-17-Depo R6: Depo-12-48-49-23-22-56-30-31-55-51-Depo R7: Depo-8-Depo R8: Depo-2-11-52-47-43-42-5-10-Depo R9: Depo-1-Depo R10: Depo-19-20-46-41-57-33-54-53-21-25-29-3-4-Depo	R1: 94 R2: 76 R3: 138,2 R4: 7,4 R5: 12,8 R6: 87,4 R7: 7,4 R8: 66,4 R9: 2,4 R10: 170,6 Total: 662,6	R1: 600 R2: 491 R3: 214 R4: 600 R5: 405 R6: 252 R7: 518 R8: 462 R9: 600 R10: 556 Total: 4698	R1: 07.00 R2: 10.17 R3: 10.42 R4: 11.52 R5: 13.53 R6: 11.57 R7: 09.21 R8: 11.54 R9: 08.57 R10: 08.11	R1: 14.08 R2: 16.02 R3: 16.39 R4: 15.29 R5: 16.43 R6: 16.43 R7: 12.32 R8: 16.59 R9: 12.24 R10: 16.59	2331300	8591	2879,3985
7	R1: Depo-7-Depo R2: Depo-1-Depo	R1: 7,4 R2: 2,4	R1: 600 R2: 600	R1: 11.52 R2: 08.57	R1: 15.29 R2: 12.24	2413300	8505	2819,7578

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R3: Depo-20-54-57-55-31-33-56-52-6-Depo R4: Depo-44-22-23-25-24-21-29-12-2-Depo R5: Depo-46-41-43-30-5-19-Depo R6: Depo-8-10-51-42-47-4-Depo R7: Depo-18-15-9-28-26-49-39-37-48-50-Depo R8: Depo-14-16-27-45-3-17-Depo R9: Depo-32-Depo R10: Depo-13-35-36-34-40-38-53-11-Depo	R3: 142,9 R4: 82 R5: 106,5 R6: 87,9 R7: 127,1 R8: 59,9 R9: 94 R10: 116,5 Total: 826,6	R3: 415 R4: 400 R5: 340 R6: 564 R7: 325 R8: 583 R9: 600 R10: 271 Total: 4698	R3: 09.33 R4: 11.40 R5: 11.36 R6: 09.21 R7: 09.26 R8: 11.13 R9: 07.02 R10: 11.17	R3: 16.38 R4: 16.57 R5: 16.55 R6: 15.17 R7: 15.31 R8: 16.43 R9: 12.19 R10: 16.58			
8	R1: Depo-33-30-31-41-11-Depo R2: Depo-12-49-35-34-36-23-22-56-57-51-6-Depo R3: Depo-32-Depo R4: Depo-16-13-17-Depo R5: Depo-2-15-14-37-39-40-48-50-Depo R6: Depo-3-28-26-45-44-27-29-24-21-Depo R7: Depo-1-Depo R8: Depo-8-Depo R9: Depo-7-Depo R10: Depo-19-20-46-5-52-47-43-42-55-10-Depo R11: Depo-4-9-54-53-38-25-18-Depo	R1: 113,8 R2: 100,4 R3: 108,6 R4: 12,8 R5: 102,4 R6: 75,3 R7: 2,4 R8: 7,4 R9: 7,4 R10: 117,5 R11: 81,5 Total: 729,5	R1: 453 R2: 334 R3: 324 R4: 405 R5: 420 R6: 396 R7: 600 R8: 518 R9: 600 R10: 134 R11: 238 Total: 4698	R1: 11.00 R2: 11.09 R3: 10.35 R4: 13.53 R5: 10.57 R6: 11.23 R7: 08.57 R8: 09.21 R9: 11.50 R10: 12.00 R11: 10.25	R1: 16.56 R2: 16.48 R3: 16.43 R4: 16.43 R5: 16.47 R6: 16.32 R7: 12.24 R8: 12.32 R9: 15.32 R10: 16.59 R11: 14.44	2564750	6189	2868,1667
9	R1: Depo-4-3-19-8-Depo R2: Depo-32-Depo R3: Depo-18-23-22-54-53-38-39-37-40-48-50-Depo R4: Depo-6-9-15-33-30-47-43-41-5-10-Depo R5: Depo-1-Depo R6: Depo-28-26-25-29-27-45-44-16-11-Depo R7: Depo-20-46-2-42-55-31-56-57-51-52-Depo R8: Depo-7-Depo R9: Depo-14-49-35-34-36-24-21-12-13-17-Depo	R1: 27,2 R2: 94 R3: 161,2 R4: 124 R5: 2,4 R6: 73,7 R7: 143,3 R8: 7,4 R9: 73,3 Total: 706,5	R1: 591 R2: 600 R3: 265 R4: 500 R5: 600 R6: 599 R7: 515 R8: 600 R9: 428 Total: 4698	R1: 11.08 R2: 07.02 R3: 10.00 R4: 09.37 R5: 08.57 R6: 10.46 R7: 09.20 R8: 11.52 R9: 11.36	R1: 15.31 R2: 12.19 R3: 16.42 R4: 16.38 R5: 12.24 R6: 16.56 R7: 16.55 R8: 15.29 R9: 16.55	2153250	5362	2982,6881
10	R1: Depo-32-Depo R2: Depo-50-25-38-53-24-21-Depo R3: Depo-6-10-31-30-56-52-20-28-26-Depo R4: Depo-2-46-42-41-5-19-Depo R5: Depo-29-44-45-27-37-17-Depo R6: Depo-18-13-14-54-33-47-43-11-Depo R7: Depo-1-Depo R8: Depo-7-Depo R9: Depo-8-12-48-39-40-34-Depo	R1: 94 R2: 71,7 R3: 94,6 R4: 82,6 R5: 93 R6: 131,2 R7: 2,4 R8: 7,4 R9: 107,9	R1: 600 R2: 174 R3: 500 R4: 527 R5: 519 R6: 219 R7: 600 R8: 600 R9: 598	R1: 07.02 R2: 13.27 R3: 10.16 R4: 10.57 R5: 11.05 R6: 11.02 R7: 08.56 R8: 11.52 R9: 09.21	R1: 12.19 R2: 16.58 R3: 16.28 R4: 16.48 R5: 16.58 R6: 16.14 R7: 12.26 R8: 15.29 R9: 16.08	2396650	7396	2839,5471

No	Rute	Jarak Tempuh (km)	Jumlah Barang (karton)	Waktu Berangkat	Waktu Pulang	Total Biaya (Rupiah)	Iterasi Konvergen	Running Time (detik)
	R10: Depo-49-35-36-23-22-57-55-51-9-15-16-3-4-Depo	R10: 108,5 Total:793,3	R10: 361 Total: 4698	R10: 09.49	R10: 16.08			
Rata-rata						2346800	6813,1	2840,8275



B9. Skrip Program

1. Mengubah bentuk TSP menjadi VRPTW

```
function
[Rute,TotJar,TotBar,Wber,Wpul]=TSP2VRPTW(TSP,Jarak,Kecepatan,BukaTutup,Permi
ntaan,Kapasitas,TService)
if length(TService)==1
    TService=[TService 0];
end
N=length(TSP);a=1;uk=1;
for i=1:N
    temp=[0 TSP(a:i) 0];
    %Perhitungan Waktu
    tw=1;
    timeA=BukaTutup(1,1);
    for j=1:length(temp)-2

timeA=max(timeA+Jarak(temp(j)+1,temp(j+1)+1)/Kecepatan(temp(j)+1,temp(j+1)+1
),BukaTutup(1,temp(j+1)+1));
        timeA=timeA+(Permintaan(temp(j+1)+1)*TService(1)+TService(2))/3600;
    end
    if timeA>BukaTutup(2,temp(j+1)+1) %melebihi waktu tutup toko
        tw=0;
    else
        timeB=timeA;
        for j=length(temp)-1:-1:2
            timeB=timeB-
(Permintaan(temp(j)+1)*TService(1)+TService(2))/3600;
            timeB=timeB-Jarak(temp(j)+1,temp(j-
1)+1)/Kecepatan(temp(j)+1,temp(j-1)+1);
            if timeB>BukaTutup(2,temp(j-1)+1) %tidak memenuhi time windows
                tw=0;
                break;
            end
        end
    end
    timeP=timeA+Jarak(temp(end-1)+1,temp(end)+1)/Kecepatan(temp(end-
1)+1,temp(end)+1);
    if timeP>BukaTutup(2,1) %pulang melebihi jam kerja
        tw=0;
    end
    %Perhitungan Jumlah Barang
    cc=1;
    Jbar=sum(Permintaan(temp+1));
    if Jbar>Kapasitas
        cc=0;
    end
    %Penyimpanan solusi
    if tw==0 || cc==0
        Rute{uk}=temp(2:end-2);
        tjar=Jarak(1,Rute{uk}(1)+1);
        for j=1:length(Rute{uk})-1
            tjar=tjar+Jarak(Rute{uk}(j)+1,Rute{uk}(j+1)+1);
        end
        tjar=tjar+Jarak(Rute{uk}(end)+1,1);
        TotJar(uk)=tjar;
    end
end
end
```

```

TotBar(uk)=sum(Permintaan(Rute{uk}+1));
temp2=[0 Rute{uk} 0];
%Perhitungan Waktu
timeA=BukaTutup(1,1);
for j=1:length(temp2)-2

timeA=max(timeA+Jarak(temp2(j)+1,temp2(j+1)+1)/Kecepatan(temp2(j)+1,temp2(j+1)+1),BukaTutup(1,temp2(j+1)+1));

timeA=timeA+(Permintaan(temp2(j+1)+1)*TService(1)+TService(2))/3600;
end
timeB=timeA;
for j=length(temp2)-1:-1:2
timeB=timeB-
(Permintaan(temp2(j)+1)*TService(1)+TService(2))/3600;
timeB=timeB-Jarak(temp2(j)+1,temp2(j-1)+1)/Kecepatan(temp2(j)+1,temp2(j-1)+1);
end
timeP=timeA+Jarak(temp2(end-1)+1,temp2(end)+1)/Kecepatan(temp2(end-1)+1,temp2(end)+1);
Wber(uk)=timeB; %Waktu Berangkat
Wpul(uk)=timeP; %Waktu Pulang
uk=uk+1;
a=i;
if i==N
Rute{uk}=temp(end-1);
tjar=Jarak(1,Rute{uk}(1)+1);
for j=1:length(Rute{uk})-1
tjar=tjar+Jarak(Rute{uk}(j)+1,Rute{uk}(j+1)+1);
end
tjar=tjar+Jarak(Rute{uk}(end)+1,1);
TotJar(uk)=tjar;
TotBar(uk)=sum(Permintaan(Rute{uk}+1));
temp2=[0 Rute{uk} 0];
%Perhitungan Waktu
timeA=BukaTutup(1,1);
for j=1:length(temp2)-2

timeA=max(timeA+Jarak(temp2(j)+1,temp2(j+1)+1)/Kecepatan(temp2(j)+1,temp2(j+1)+1),BukaTutup(1,temp2(j+1)+1));

timeA=timeA+(Permintaan(temp2(j+1)+1)*TService(1)+TService(2))/3600;
end
timeB=timeA;
for j=length(temp2)-1:-1:2
timeB=timeB-
(Permintaan(temp2(j)+1)*TService(1)+TService(2))/3600;
timeB=timeB-Jarak(temp2(j)+1,temp2(j-1)+1)/Kecepatan(temp2(j)+1,temp2(j-1)+1);
end
timeP=timeA+Jarak(temp2(end-1)+1,temp2(end)+1)/Kecepatan(temp2(end-1)+1,temp2(end)+1);
Wber(uk)=timeB; %Waktu Berangkat
Wpul(uk)=timeP; %Waktu Pulang
end
else
if i==N
Rute{uk}=temp(2:end-1);

```

```

tjar=Jarak(1,Rute{uk}(1)+1);
for j=1:length(Rute{uk})-1
    tjar=tjar+Jarak(Rute{uk}(j)+1,Rute{uk}(j+1)+1);
end
tjar=tjar+Jarak(Rute{uk}(end)+1,1);
TotJar(uk)=tjar;
TotBar(uk)=sum(Permintaan(Rute{uk}+1));
temp2=[0 Rute{uk} 0];
%Perhitungan Waktu
timeA=BukaTutup(1,1);
for j=1:length(temp2)-2
    timeA=max(timeA+Jarak(temp2(j)+1,temp2(j+1)+1)/Kecepatan(temp2(j)+1,temp2(j+1)+1),BukaTutup(1,temp2(j+1)+1));
    timeA=timeA+(Permintaan(temp2(j+1)+1)*TService(1)+TService(2))/3600;
end
timeB=timeA;
for j=length(temp2)-1:-1:2
    timeB=timeB-
    (Permintaan(temp2(j)+1)*TService(1)+TService(2))/3600;
    timeB=timeB-Jarak(temp2(j)+1,temp2(j-1)+1)/Kecepatan(temp2(j)+1,temp2(j-1)+1);
end
timeP=timeA+Jarak(temp2(end-1)+1,temp2(end)+1)/Kecepatan(temp2(end-1)+1,temp2(end)+1);
Wber(uk)=timeB; %Waktu Berangkat
Wpul(uk)=timeP; %Waktu Pulang
end
end
end

```

2. Mengubah jam menjadi waktu

```

function waktu=jam2waktu(jam)
waktu=floor(jam)+(jam-floor(jam))/60*100;

```

3. Menghitung Biaya

```

function Biaya=hitungbiaya(TotalJarak,Upah,BBM)
K=length(TotalJarak);
Biaya=K*Upah+sum(TotalJarak)*BBM;

```

4. Metode *Slide Mutation*

```

function Offspring=SlideMut(Parent,Pm)
N=length(Parent);
rnd=rand(1,N);
ps=find(rnd<=Pm);
Gen=Parent(ps);
Offspring=Parent;
if ~isempty(Gen) && length(Gen)>1
    Gen=[Gen(end) Gen(1:end-1)];
    Offspring(ps)=Gen;
end

```


end

5. Metode Two Point Order Crossover

```
function [Offspring1, Offspring2]=Twocut(Parent1,Parent2)
n=length(Parent1);
i=ceil(rand*n);
j=ceil(rand*n);
while i==j
    j=ceil(rand*n);
end
if i>=j
    temp=i;
    i=j;
    j=temp;
end
Offspring1a=Parent2(i:j);
Offspring2a=Parent1(i:j);
Offspring1b=setdiff(Parent1,Offspring1a,'stable');
Offspring2b=setdiff(Parent2,Offspring2a,'stable');
Offspring1=[Offspring1b(1:i-1) Offspring1a Offspring1b(i:end)];
Offspring2=[Offspring2b(1:i-1) Offspring2a Offspring2b(i:end)];
```