



**IMPLEMENTASI ALGORITMA CODEQ DAN MODIFIKASI
ALGORITMA CODEQ PADA *CAPACITATED VEHICLE ROUTING
PROBLEM WITH TIME WINDOWS (CVRPTW)***

SKRIPSI

Oleh:

**Yulia Nurul Faridah
NIM 131810101047**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017**



**IMPLEMENTASI ALGORITMA CODEQ DAN MODIFIKASI
ALGORITMA CODEQ PADA *CAPACITATED VEHICLE ROUTING
PROBLEM WITH TIME WINDOWS (CVRPTW)***

SKRIPSI

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

Oleh:

**Yulia Nurul Faridah
NIM 131810101047**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. ALLAH SWT yang telah memberikan semua kemudahan dan kesempurnaan dalam kehidupan ini;
2. Kedua orang tua saya, Ayahanda Syamsul Arifin dan Ibunda Nurul Husnah yang senantiasa memberikan semangat, dorongan moril maupun materi, kasih sayang serta do'a tiada henti;
3. Kakak saya, Khusnul Hasanah dan Muchlis Santoso yang memberikan semangat serta dukungan moril;
4. Seluruh guru dan dosen, sejak taman kanak-kanak hingga perguruan tinggi yang telah memberi banyak ilmu dan membimbing dengan tulus;
5. Sahabat saya, Mufty Akbar Rosyid yang memberikan motivasi, dukungan moril serta kasih sayang;
6. Sahabat saya, Risky Damayanti, Dewinta Bunga P, Riska Ayu yang telah menjadi sahabat terbaik selama kuliah, terima kasih atas motivasi-motivasinya, semangat kalian dan candaan kalian;
7. teman-teman "ATLAS", dan semua pihak yang telah memberikan dukungan serta motivasi, sehingga skripsi ini bisa terselesaikan;
8. Almamater tercinta Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

Maka sesungguhnya bersama kesulitan ada kemudahan, sesungguhnya bersama kesulitan dan kemudahan. Maka apabila engkau telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan yang lain), dan hanya kepada Tuhanmulah engkau berharap
(terjemahan Surat Al – Insyirah Ayat 5 - 8)^{*)}

Musuh yang paling berbahaya di atas dunia ini adalah penakut dan bimbang.
Teman yang paling setia, hanyalah keberanian dan keyakinan yang teguh.^{**)}

^{*)} Al-Qur'an. 2006. Al-Qur'an terjemahan Bahasa Indonesia. Kudus: Menara Kudus.

^{**)} Andrew Jackson

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Yulia Nurul Faridah

NIM : 131810101047

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Implementasi Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada *Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan dalam institusi manapun dan juga 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, Juni 2017

Yang menyatakan,

Yulia Nurul Faridah

NIM 131810101047

SKRIPSI

**IMPLEMENTASI ALGORITMA CODEQ DAN MODIFIKASI
ALGORITMA CODEQ PADA *CAPACITATED VEHICLE ROUTING
PROBLEM WITH TIME WINDOWS (CVRPTW)***

Oleh

Yulia Nurul Faridah
NIM 131810101047

Pembimbing:

Dosen Pembimbing Utama : Kusbudiono, S.Si., M.Si.

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

PENGESAHAN

Skripsi berjudul “Implementasi Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW)” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

Kusbudiono, S.Si., M.Si.

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

NIP. 197704302005011001

NIP. 197211291998021001

Anggota II,

Anggota III,

Dr. Firdaus Ubaidillah, S.Si., M.Si.

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

NIP. 197006061998031003

NIP. 196908281998021001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph.D.

NIP. 196102041987111001

RINGKASAN

Implementasi Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada *Capacitated Vehicle Routing Problem with Time Windows*; Yulia Nurul Faridah, 131810101047; 2017; 96 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Vehicle Routing Problem merupakan permasalahan distribusi yang menyatakan bahwa sebuah kendaraan harus berawal dari depo untuk mengirimkan sejumlah barang kepada n pelanggan dan kembali ke depo dengan memenuhi kendala bahwa semua pelanggan hanya dapat dilayani sekali oleh sebuah kendaraan. Tujuan dari permasalahan tersebut untuk meminimalkan biaya distribusi secara keseluruhan dengan mereduksi total jarak yang ditempuh, total waktu perjalanan, dan mengurangi jumlah kendaraan yang dibutuhkan. VRP telah dikembangkan sesuai dengan kendala distribusi yang ada, seperti *Capacitated Vehicle Routing Problem with Time Windows*. CVRPTW merupakan permasalahan distribusi yang mempunyai kendala kapasitas kendaraan yang homogen dan *time windows* untuk meminimalkan total waktu perjalanan.

Pada penelitian ini, menggunakan algoritma CODEQ dan modifikasi algoritma CODEQ untuk menyelesaikan kasus CVRPTW. Hasil dari kedua algoritma tersebut akan dibandingkan dengan tujuan untuk melihat hasil dari solusi yang terbaik berdasarkan nilai minimal total waktu perjalanan

Tujuan dari penelitian ini yaitu dapat mengetahui hasil dari perbandingan algoritma CODEQ dan modifikasi algoritma CODEQ pada kasus CVRPTW dan dapat mengetahui pengaruh populasi terhadap kinerja algoritma CODEQ dan modifikasi algoritma CODEQ. Data distribusi yang digunakan dalam penelitian ini merupakan data sekunder yang terdiri dari 32 toko dan 57 toko. Data distribusi tersebut berupa data jumlah permintaan setiap toko, jarak antara depo dengan toko dan jarak antar toko, kecepatan rata-rata kendaraan, waktu operasional dan waktu pelayanan (*service*).

Hasil yang diperoleh setelah melakukan beberapa percobaan menunjukkan bahwa modifikasi algoritma CODEQ menghasilkan solusi yang lebih baik dari algoritma CODEQ dengan menggunakan data 32 toko maupun 57 toko. Namun, algoritma CODEQ memerlukan *running time* yang lebih cepat dibandingkan dengan modifikasi algoritma CODEQ. Semakin besar nilai populasi dan iterasi maksimal maka hasil yang didapat akan semakin baik (total waktu perjalanan lebih minimum). Hasil solusi mendekati optimal yang didapatkan setelah melakukan beberapa percobaan dengan menggunakan 32 toko, 80 populasi dan 3000 iterasi yaitu 19 jam 22 menit 13 detik dengan *running time* sebesar 308,1974 detik untuk modifikasi algoritma CODEQ dan 22 jam 17 menit 29 detik dengan *running time* sebesar 169,9125 detik untuk algoritma CODEQ.

PRAKATA

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik dan karuniaNya sehingga skripsi yang berjudul “Implementasi Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada *Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)*” dapat terselesaikan. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata 1 (S1) di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

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

1. Bapak Kusbudiono S.Si., M.Si. selaku Dosen Pembimbing Utama dan Bapak Ahmad Kamsyakawuni, S.Si., M.Kom. selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan dan bantuan dalam penyusunan skripsi ini;
2. Bapak Dr. Firdaus Ubaidillah, S.Si., M.Si. selaku dosen penguji I dan Bapak Kosala Dwidja Purnomo, S.Si., M.Si. selaku dosen penguji II yang telah memberikan kritik dan saran yang membangun untuk penyempurnaan skripsi ini;
3. Ibu Ika Hesti Agustin, S.Si., M.Si. selaku Dosen Pembimbing Akademik yang telah membimbing dalam pemilihan mata kuliah;
4. Seluruh dosen dan karyawan Jurusan Matematika Fakultas MIPA Universitas Jember yang telah memberikan ilmu serta membantu selama proses perkuliahan berlangsung;
5. Keluarga yang selalu memberikan doa tiada henti dan dukungan baik lahir maupun batin;
6. Semua pihak yang membantu terselesainya skripsi ini.

Penulis menyadari bahwa dalam menyusun skripsi ini masih terdapat kekurangan baik isi maupun susunannya. Oleh sebab itu, penulis mengharapkan saran dan kritik demi penyempurnaan skripsi ini. Akhirnya penulis berharap

semoga skripsi ini dapat memberi manfaat dan sumbangan pengetahuan bagi pembaca.

Jember, Juni 2017

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSEMBAHAN	ii
HALAMAN MOTTO	iii
PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Graf	5
2.2 <i>Vehicle Routing Problem</i>	6
2.3 <i>Capacitated Vehicle Routing Problem with Time Windows</i>	8
2.4 Algoritma CODEQ	11
2.5 Algoritma <i>Defferential Evolution Plus</i>	17
BAB 3. METODELOGI PENELITIAN	20
3.1 Data Penelitian	20
3.2 Langkah-langkah Penelitian	20
BAB 4. HASIL DAN PEMBAHASAN	25

4.1 Hasil	25
4.1.1 Perhitungan Manual	25
4.1.2 Program Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada CVRPTW Menggunakan MATLAB	38
4.2 Pembahasan	45
4.2.1 Penerapan Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada CVRPTW	45
4.2.2 Pengaruh Populasi Terhadap Kinerja Algoritma CODEQ dan Modifikasi Algoritma CODEQ pada CVRPTW	58
BAB 5. KESIMPULAN DAN SARAN	60
5.1 Kesimpulan	60
5.2 Saran	61
DAFTAR PUSTAKA	62
LAMPIRAN	64

DAFTAR TABEL

	Halaman
4.1. Data Jarak 8 Toko (Km)	25
4.2. Jumlah Permintaan dan Waktu Operasional Toko	26
4.3. Data Kecepatan Rata-rata Kendaraan	26
4.4. Hasil percobaan pertama algoritma CODEQ pada 32 toko	46
4.5. Hasil percobaan pertama modifikasi algoritma CODEQ pada 32 toko	46
4.6. Hasil percobaan ke-2 algoritma CODEQ pada 32 toko	47
4.7. Hasil percobaan ke-2 modifikasi algoritma CODEQ pada 32 toko	47
4.8. Hasil percobaan ke-3 algoritma CODEQ pada 32 toko	48
4.9. Hasil percobaan ke-3 modifikasi algoritma CODEQ pada 32 toko	48
4.10. Hasil percobaan ke-4 algoritma CODEQ pada 32 toko	49
4.11. Hasil percobaan ke-4 modifikasi algoritma CODEQ pada 32 toko	49
4.12. Hasil percobaan ke-5 algoritma CODEQ pada 32 toko	50
4.13. Hasil percobaan ke-5 modifikasi algoritma CODEQ pada 32 toko	50
4.14. Hasil percobaan ke-6 algoritma CODEQ pada 32 toko	51
4.15. Hasil percobaan ke-6 modifikasi algoritma CODEQ pada 32 toko	52
4.16. Hasil percobaan ke-7 algoritma CODEQ pada 32 toko	52
4.17. Hasil percobaan ke-7 modifikasi algoritma CODEQ pada 32 toko	53
4.18. Hasil percobaan ke-8 algoritma CODEQ pada 32 toko	54
4.19. Hasil percobaan ke-8 modifikasi algoritma CODEQ pada 32 toko	54
4.20. Hasil percobaan ke-9 algoritma CODEQ pada 32 toko	55
4.21. Hasil percobaan ke-9 modifikasi algoritma CODEQ pada 32 toko	55
4.22. Hasil dari sembilan percobaan pada 32 toko	56
4.23. Hasil percobaan ke-10 algoritma CODEQ pada 57 toko	56
4.24. Hasil percobaan ke-10 modifikasi algoritma CODEQ pada 57 toko	57

DAFTAR GAMBAR

	Halaman
2.1 Graf Berarah dan Berbobot	6
2.2 Contoh Penyelesaian CVRPTW	11
3.1 Skema Langkah-langkah Penelitian	24
4.1 Tampilan Awal Program Algoritma CODEQ	39
4.2 Tampilan Awal Program Modifikasi Algoritma CODEQ	40
4.3 Tampilan Hasil Program Algoritma CODEQ	40
4.4 Tampilan Hasil Program Modifikasi Algoritma CODEQ	41
4.5 Tampilan Tabel Data Algoritma CODEQ	42
4.6 Tampilan Tabel Data Modifikasi Algoritma CODEQ	42
4.7 Tampilan <i>Input</i> Parameter Algoritma CODEQ	43
4.8 Tampilan <i>Input</i> Parameter Modifikasi Algoritma CODEQ	43
4.9 Tampilan <i>Output</i> Program Algoritma CODEQ	44
4.10 Tampilan <i>Output</i> Program Modifikasi Algoritma CODEQ	44
4.11 Representasi Rute Kendaraan Modifikasi Algoritma CODEQ	45

DAFTAR LAMPIRAN

	Halaman
A1. Jumlah Permintaan dan Waktu Operasional untuk 32 Toko.....	64
A2. Jarak 32 Toko	65
A3. Kecepatan Rata-rata Kendaraan untuk 32 Toko	67
A4. Jumlah Permintaan dan Waktu Operasional untuk 57 Toko	70
A5. Jarak 57 Toko	71
A6. Kecepatan Rata-rata Kendaraan untuk 57 Toko	79
B. Hasil Percobaan algoritma CODEQ dengan 32 Toko	87
C. Hasil Percobaan modifikasi algoritma CODEQ dengan 32 Toko	92
D. Hasil Percobaan algoritma dengan 57 Toko	95

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kegiatan distribusi barang selalu menjadi bagian dalam produsen untuk mengirimkan barang terhadap konsumen. Dalam melakukan pendistribusian barang, perusahaan harus mampu menentukan rute yang akan dilewati sehingga membuat pengiriman barang akan lebih cepat sampai dengan jarak rute yang terpendek dan mengeluarkan biaya paling minimum. Terdapat banyak masalah dalam melakukan pengiriman barang yakni masalah armada kendaraan yang kapasitasnya harus sesuai dengan permintaan konsumen agar tidak membuang banyak biaya saat melakukan pengiriman barang, rute yang akan ditempuh harus memiliki biaya yang paling minimal dan juga mempertimbangkan waktu perjalanan agar pengiriman barang menjadi cepat dan memperoleh biaya paling minimal.

Kegiatan distribusi sangat penting bagi perusahaan besar maupun perusahaan kecil sehingga terbentuklah permasalahan *Vehicle Routing Problem* (VRP) untuk menyelesaikan masalah distribusi dengan biaya pengiriman seminimal mungkin, dengan permasalahan ini dapat menentukan rute terpendek dalam pengiriman barang dari distributor ke pelanggan. Masalah distribusi pada kehidupan nyata seringkali jauh lebih kompleks dari masalah VRP, sehingga masalah VRP dikembangkan menurut kendala atau batasan distribusi yang ada. Berbagai macam jenis VRP yang telah dikembangkan, seperti *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW). CVRPTW merupakan kombinasi antara *Vehicle Routing Problem with Time Windows* (VRPTW) dan *Capacitated Vehicle Routing Problem* (CVRP).

Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) adalah permasalahan distribusi barang yang mempunyai kendala kapasitas kendaraan yang homogen dan *time windows* untuk meminimalkan total waktu perjalanan. *Time windows* pada CVRPTW yaitu waktu perjalanan yang ditentukan oleh pelanggan untuk menerima barang. Permasalahan ini dapat diselesaikan dengan menggunakan metode heuristik dan metode metaheuristik.

Tujuan dari metode heuristik yaitu untuk mendapatkan performa komputasi atau penyederhanaan konseptual dan berpotensi pada biaya keakuratan atau presisi, sedangkan metode metaheuristik merupakan metode lanjut berbasis heuristik untuk menyelesaikan persoalan optimisasi secara efisien, misalnya menggunakan algoritma *Genetika*, algoritma *Particle Swarm Optimization*, algoritma *Simulated Annealing* dan lain-lain.

CVRPTW telah dibahas oleh Reviladi (2016) menggunakan algoritma *Floyd Warshall* (metode eksak) dan *Nearest neighbour* (metode heuristik). Hasil dari penelitian tersebut menyatakan bahwa algoritma *floyd warshall* dapat membentuk rute optimal dibandingkan dengan algoritma *nearest neighbor* namun, proses penyelesaiannya dengan algoritma *nearest neighbor* lebih cepat dan praktis dibanding dengan algoritma *floyd warshall*. Agyei, *et al* (2015) membahas CVRPTW menggunakan algoritma *Clarke and Wright savings with local search*, dalam penelitiannya menghasilkan bahwa algoritma tersebut dapat menyelesaikan permasalahan CVRPTW dengan baik. Gunarso (2014) melakukan penelitian mengenai algoritma CODEQ yang diaplikasikan pada permasalahan *Jobshop Scheduling* dengan studi kasus untuk memproduksi spring bed, yang menghasilkan semakin besar jumlah populasi, kemungkinan untuk mendapatkan nilai *makespan* yang minimum tanpa harus melakukan banyak iterasi dan *runing* yang banyak akan semakin besar. Algoritma CODEQ dapat menyelesaikan VRP yang dilakukan oleh Garside dan Satya (2014) dengan menggunakan pendekatan *route first-cluster second* dalam menyelesaikan permasalahan penentuan rute.

Algoritma CODEQ merupakan metode metaheuristik penggabungan dari algoritma *Chaotic Search*, *Opposition-based Learning*, *Differential Evolution*, dan *Quantum Mechanics* yang tidak membutuhkan parameter kecuali untuk ukuran populasi. Algoritma CODEQ telah berhasil diterapkan untuk menyelesaikan permasalahan *constrained optimization* (Omran dan Ayed, 2009). Performansi algoritma CODEQ terbukti menghasilkan solusi yang baik pada permasalahan optimasi kontinu (Omran, 2010). Performansi algoritma CODEQ lebih baik jika dibandingkan dengan algoritma *Differential Evolution* dan algoritma *Simulated Annealing* dalam permasalahan diskrit yaitu penempatan kapasitor dalam sistem

distribusi dengan *integer programming* (Chio dan Chung, 2013). Algoritma CODEQ didasarkan pada algoritma DE, sedangkan telah ada penelitian tentang memodifikasi algoritma DE menjadi algoritma DE *Plus*. Pada tahun 2012 Wiratno *et al*, mengembangkan algoritma DE untuk menyelesaikan penjadwalan *flow shop* banyak mesin dengan multi obyektif. Berdasarkan penelitian tersebut hasil solusi algoritma DE *Plus* lebih baik dari algoritma DE.

Berdasarkan keunggulan algoritma CODEQ dalam menyelesaikan permasalahan kontinu dan diskrit, maka penelitian ini menggunakan algoritma CODEQ untuk menerapkan pada salah satu permasalahan optimasi diskrit, yaitu *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW). Selain itu peneliti memodifikasi algoritma CODEQ yaitu dengan menggantikan algoritma DE menjadi algoritma DE *Plus*. Modifikasi algoritma CODEQ diharapkan mampu mendapatkan hasil yang jauh lebih baik lagi dari algoritma CODEQ. Penelitian ini bertujuan untuk membentuk rute perjalanan dengan jarak yang seminimal mungkin dengan kendala kapasitas dan meminimalkan waktu pengiriman dengan kendala batasan waktu *time windows*.

Data yang akan digunakan merupakan data sekunder yang diambil dari penelitian Ulyawati (2016) yang membahas tentang penyelesaian *Capacitated Vehicle Routing Problem* dengan algoritma *harmony search* dan algoritma *tabu search*. Data tersebut merupakan data distribusi dari perusahaan air minum kemasan. Data distribusi tersebut tidak mencantumkan data-data mengenai waktu yang digunakan untuk menyelesaikan permasalahan *Capacitated Vehicle Routing Problem with Time Windows*. Sehingga peneliti akan menambahkan data waktu perjalanan, waktu operasional, waktu pelayanan (*service*) dan mengansumsikan jenis dan kapasitas kendaraan dianggap sama.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas berdasarkan latar belakang, yaitu:

- a. bagaimana penerapan algoritma CODEQ dan modifikasi algoritma CODEQ pada CVRPTW?

- b. bagaimana perbandingan hasil solusi (total waktu minimal) dari algoritma CODEQ dan modifikasi algoritma CODEQ pada kasus CVRPTW?
- c. bagaimana pengaruh populasi terhadap kinerja dan hasil dari algoritma CODEQ dan modifikasi algoritma CODEQ pada kasus CVRPTW?

1.3 Batasan Masalah

Agar tidak memperluas permasalahan, peneliti memberikan asumsi-asumsi sebagai berikut:

- a. jenis barang distribusi hanya satu macam (ukuran dan berat sama),
- b. waktu servis dianggap sama untuk setiap barang.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

- a. untuk menerapkan algoritma CODEQ dan modifikasi algoritma CODEQ pada CVRPTW,
- b. untuk mengetahui hasil solusi (total waktu minimal) dari perbandingan algoritma CODEQ dan modifikasi algoritma CODEQ pada kasus CVRPTW,
- c. mengetahui pengaruh populasi terhadap kinerja algoritma CODEQ dan modifikasi algoritma CODEQ pada kasus CVRPTW.

1.5 Manfaat Penelitian

Berdasarkan rumusan masalah, manfaat yang dapat diambil dari penelitian ini, yaitu:

- a. diharapkan mendapatkan solusi optimal dari penyelesaian *Capacitated Vehicle Routing Problem with Time Windows* menggunakan algoritma CODEQ dan modifikasi algoritma CODEQ,
- b. menambah wawasan menggunakan algoritma CODEQ dan modifikasi algoritma CODEQ untuk penyelesaian *Capacitated Vehicle Routing Problem with Time Windows*.

BAB 2. TINJAUAN PUSTAKA

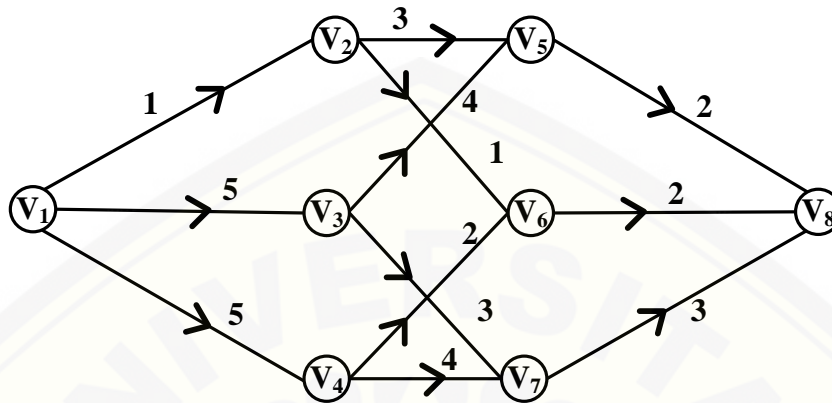
2.1 Graf

Graf G adalah suatu himpunan (V,E) dimana $V = \{v_1, v_2, v_3, \dots, v_n\}$ merupakan himpunan tak kosong dengan elemen-elemennya disebut titik (*vertex*), sedangkan $E = \{e_1, e_2, e_3, \dots, e_n\}$ merupakan himpunan yang boleh kosong dan himpunan tak terurut dari elemen-elemen di V yang anggotanya disebut sisi. Sehingga setiap sisi e_k diidentifikasi dengan pasangan (v_i, v_j) dari titik. Suatu *walk* dapat didefinisikan dengan urutan bergantian yang terbatas dari titik dan sisi yang dimulai dan diakhiri dengan titik yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$. Tidak ada sisi yang muncul (dilalui) lebih dari sekali dalam melewati *walk* sedangkan titik (*vertex*) dapat muncul lebih dari sekali. Lintasan tertutup (*cycle*) terjadi jika diawali dan diakhiri dengan titik yang sama. Namun, jika diawali dan diakhiri dengan titik yang berbeda maka lintasan tersebut merupakan lintasan terbuka. Graf juga dapat dihubungkan antara graf yang satu dengan yang lainnya jika ada setidaknya satu lintasan antara setiap pasangan titik di graf G .

Graf berbobot merupakan graf yang setiap sisinya diberikan sebuah harga (bobot). Suatu graf dapat dikelompokkan berdasarkan orientasi arah atau panah, yaitu graf berarah dan graf tak berarah. Graf berarah yaitu graf yang setiap sisinya memiliki orientasi arah atau panah sehingga (v_i, v_j) dan (v_j, v_i) mempunyai dua buah busur yang berbeda $(v_i, v_j) \neq (v_j, v_i)$. Sedangkan graf tak berarah yaitu graf yang sisinya tidak mempunyai orientasi arah atau panah sehingga merupakan busur yang sama $(v_i, v_j) = (v_j, v_i)$ (Munir, 2010).

Untuk mempresentasikan pada permasalahan VRP, graf berarah akan menunjukkan jalur yang akan dilewati dari depo ke pelanggan dan lintasan tertutup menggambarkan rute kendaraan yang akan ditempuh yang diawali dan diakhiri dari depo. Graf terhubung menggambarkan banyaknya rute yang akan ditempuh. Sedangkan graf berbobot dapat diasumsikan sebagai jarak antara

pelanggan, biaya perjalanan, waktu tempuh, ongkos produksi. Graf berarah dan berbobot dapat dilihat pada Gambar 2.1.



Gambar 2.1 Graf Berarah dan Berbobot

2.2 Vehicle Routing Problem (VRP)

Penentuan rute kendaraan merupakan salah satu masalah dari kegiatan distribusi. Untuk meminimumkan biaya, suatu perusahaan harus mengetahui rute terpendek dengan jarak minimum yang akan ditempuh, waktu yang minimum, dan juga memperhatikan kapasitas kendaraan yang akan digunakan untuk melakukan pengiriman tersebut. Salah satu karakteristik permasalahan tersebut dapat diselesaikan dengan *Vehicle Routing Problem* (VRP). VRP pertama kali diteliti oleh Dantzig dan Ramser untuk menentukan formulasi pemrograman matematika dengan pendekatan algoritma untuk memecahkan suatu masalah pada pengiriman *gasoline* ke pusat pelayanan (Caric dan Hrvoje, 2008).

Menurut Caric dan Hrvoje (2008), definisi dari VRP adalah sebuah kendaraan harus berawal dari depo untuk mengirimkan sejumlah barang kepada n pelanggan dan kembali ke depo dengan memenuhi kendala bahwa semua pelanggan hanya dapat dilayani sekali oleh sebuah kendaraan. Tujuan dari permasalahan tersebut untuk meminimalkan biaya distribusi secara keseluruhan. Biaya distribusi dapat diperbaiki dengan mereduksi total jarak yang ditempuh dan mengurangi jumlah kendaraan yang dibutuhkan. Selanjutnya penelitian VRP terus

berkembang dengan meliputi pendekatan pemecahan masalah dan munculnya kendala-kendala baru yang ada pada masalah distribusi.

Menurut Toth dan Daniele (2002), variasi VRP yang telah dikembangkan dengan sesuai kendala atau batasan masalah yang ada, sebagai berikut:

- a. *Vehicle Routing Problem with Time Windows* (VRPTW), merupakan kasus VRP dimana setiap konsumen memiliki batasan rentang waktu pelayanan;
- b. *Capacitated Vehicle Routing Problem* (CVRP), merupakan kendala yang ada hanya berupa kapasitas kendaraan yang homogen;
- c. *Capacitated Vehicle Routing Problem with Time Windows* (CVRPTW), mempunyai kendala kapasitas kendaraan yang homogen dan waktu pelayanan;
- d. *Distance Constrained Vehicle Routing Problem* (DCVRP), dengan kendala batasan panjang rute;
- e. *Vehicle Routing Problem with Pick Up And Delivery* (VRPPD), dengan kendala pelayanan campuran, yaitu pengiriman dan pengambilan barang dalam satu rute;
- f. *Vehicle Routing Problem with Backhaul* (VRPB), dengan kendala pengambilan barang dapat dilakukan setelah semua pengiriman selesai;
- g. *Multiple Depot Vehicle Routing Problem* (MDVRP), menggunakan banyak depo untuk mengirimi pelanggan;
- h. *Split Delivery Vehicle Routing Problem* (SDVRP), pelanggan dapat dilayani oleh kendaraan yang berbeda-beda;
- i. gabungan MDVRP dan VRPB, kondisi depo sebagai pusat distribusi barang lebih dari satu dan pelanggan dapat melakukan permintaan berupa pengiriman dan pengambilan barang (VRPB);

Menurut Toth dan Daniele (2002), solusi yang dihasilkan dari permasalahan VRP yaitu meminimalkan total jarak tempuh kendaraan dari depo ke pelanggan dengan memperhatikan kendala-kendala yang harus terpenuhi, seperti:

- a. rute kendaraan dimulai dari depo dan kembali ke depo,
- b. masing-masing pelanggan hanya dapat dilayani oleh satu kendaraan,
- c. kapasitas kendaraan tidak boleh melebihi daya kapasitasnya,

- d. permintaan pelanggan tidak boleh melebihi kapasitas kendaraan. Apabila kapasitas kendaraan sudah mencapai batas, maka pelanggan berikutnya akan dilayani oleh kendaraan selanjutnya,
- e. tidak terdapat subrute untuk setiap kendaraan.

Tujuan yang ingin dicapai dalam menyelesaikan masalah *Vehicle Routing Problem* (VRP), yaitu:

- a. meminimasi ongkos (biaya) perjalanan secara keseluruhan yang dipengaruhi oleh jarak yang ditempuh dan jumlah kendaraan yang digunakan,
- b. meminimasi jumlah kendaraan yang digunakan untuk melayani semua pelanggan,
- c. mencari rute terpendek sehingga dapat meminimasi ongkos pengiriman,
- d. meminimasi keluhan pelanggan. Toth dan Daniele (2002)

VRP diklasifikasikan dalam *NP-hard problem* sehingga metode eksak sangat sulit untuk diterapkan dalam permasalahan kasus VRP. Sehingga cara yang tepat untuk mendapatkan solusi dengan kondisi real dan sangat mendekati nilai optimal, digunakanlah metode heuristik atau metode metaheuristik (Caric dan Hrvoje, 2008).

2.3 Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)

CVRPTW merupakan salah satu varian yang dipelajari dari masalah VRP, dapat dikatakan bahwa CVRPTW adalah generalisasi dari *Capacitated Vehicle Routing Problem* (CVRP). Bahkan, di samping keterbatasan kapasitas yang diperkenalkan di CVRP, kendaraan juga harus mempunyai batasan *time windows* yang terkait dengan waktu pengiriman yang telah ditentukan oleh pelanggan (Caric dan Hrvoje, 2008). Menurut Sousa, *et al* (2011), CVRPTW sangat penting dalam memperoleh pemecahan masalah atau solusi dari masalah rute optimal. Waktu tunggu yang lama harus dihindari untuk alasan seperti upah *driver*, kepuasan pelanggan atau kondisi produk.

Selain masalah rute yang harus ditempuh saat mengirim barang, pada CVRPTW juga terdapat beberapa kendala seperti kapasitas kendaraan dan waktu pengiriman. Dalam Reviladi (2016), yang dimaksud dari kendala kapasitas yaitu

setiap kendaraan memiliki kapasitas tertentu yang homogen dan jika kapasitas sudah penuh maka kendaraan tersebut tidak bisa melayani pelanggan berikutnya. Waktu pengiriman juga sangat penting dalam melakukan kegiatan distribusi barang, karena akan berpengaruh terhadap kepuasan pelanggan.

CVRPTW dapat direpresentasikan pada graf $G = (V, A)$, dimana $V = \{v_0, v_1, v_2, \dots, v_n\}$ menunjukkan *vertex* yang merupakan titik dimana depo dan pelanggan itu berada, dengan v_0 melambangkan depo dan yang lainnya melambangkan pelanggan. Sedangkan $A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ menunjukkan sisi dan bobotnya bernilai non negatif yang berhubungan dengan waktu perjalanan (t_{ij}) antar pelanggan i dan pelanggan j . Tiap pelanggan mempunyai waktu operasional $[t_{ai}, t_{bi}]$ yang menjelaskan bahwa kendaraan harus tiba diantara selang waktu $[t_{ai}, t_{bi}]$ yang telah ditentukan oleh pelanggan. Sedangkan waktu operasional pada depo $[t_{a0}, t_{b0}]$ merupakan interval waktu awal keberangkatan dari depo dan kembali ke depo, sehingga kendaraan tidak boleh meninggalkan depo (Agyei *et al*, 2015). Berikut beberapa notasi dan definisinya yang digunakan pada perumusan permasalahan CVRPTW adalah sebagai berikut :

C merupakan banyaknya pelanggan,

V merupakan banyaknya titik termasuk depo,

K merupakan banyaknya kendaraan,

T merupakan waktu perjalanan yang telah ditetapkan oleh depo,

x_{ij}^k (variabel keputusan), menghubungkan antara simpul i ke j yang menggunakan kendaraan k . Jika terhubung dapat ditulis 1 dan jika tidak 0,

y_j^k (variabel keputusan), pelayanan saat kendaraan k tiba di titik j ,

t_{ij} , waktu perjalanan dari titik i ke j untuk semua kendaraan,

d_j , permintaan barang untuk pelanggan j ,

Q , kapasitas maksimum untuk kendaraan.

Formulasi matematis permasalahan CVRPTW yang bertujuan untuk meminimalkan total waktu perjalanan, yang meliputi waktu perjalanan antar pelanggan dan waktu pelayanan pelanggan untuk *tour* lengkap dapat disajikan sebagai berikut :

$$\text{Min } \sum_k^K \sum_{i=0}^V \sum_{j=0}^V x_{ij}^k \cdot t_{ij} + \sum_{i=0}^V y_j^k \cdot \varphi_j \quad (2.1)$$

dimana φ merupakan *unloading time* pada setiap pelanggan (waktu pelayanan pelanggan untuk membongkar barang permintaan pelanggan).

Adapun untuk kendala *Capacitated Vehicle Routing Problem with Time Windows*, sebagai berikut:

a. persamaan (2.2) menjamin bahwa setiap pelanggan hanya dapat dilayani oleh satu kendaraan,

$$\sum_{k=1}^K y_j^k = 1 \quad \forall j \in C, k \in K \quad (2.2)$$

b. persamaan (2.3) menyatakan bahwa setiap kendaraan harus memulai perjalanan dari depo dan kembali ke depo,

$$\sum_{j=1}^V x_{0j}^k = 1 \quad \forall k \in K \quad (2.3)$$

$$\sum_{j=1}^V x_{j0}^k = 1 \quad \forall k \in K$$

c. persamaan (2.4) menjelaskan bahwa setelah kendaraan tiba di pelanggan dan melayani pelanggan tersebut dan kemudian dilanjutkan berangkat ke pelanggan berikutnya,

$$\sum_{j=1}^V x_{ij}^k = \sum_{j=1}^V x_{ji}^k \quad \forall i, j \in V, k \in K \quad (2.4)$$

d. pertidaksamaan (2.5) menjelaskan total permintaan pada sebuah rute tidak boleh melebihi kapasitas kendaraan,

$$\sum_{j=1}^V d_j (\sum_{k=1}^K y_j^k) \leq Q \quad \forall k \in K \quad (2.5)$$

e. pertidaksamaan (2.6) memastikan waktu perjalanan kendaraan tidak boleh melebihi *time windows*,

$$\sum_{i=0}^V \sum_{j=0}^V x_{ij}^k \cdot t_{ij} + \sum_{i=0}^V y_j^k \cdot \varphi_j \leq T \quad \forall (i, j) \in V, k \in K \quad (2.6)$$

f. pertidaksamaan (2.7) merupakan kendala eliminasi rute, yang memastikan bahwa tidak ada rute yang keluar dari depo,

$$\sum_{i=0, j=0}^S x_{ij}^k \leq |S| - 1, \quad \forall S \subseteq C, 2 \leq |S| \leq n + 2 \quad (2.7)$$

g. persamaan (2.8) dan persamaan (2.9) menyatakan variabel keputusan,

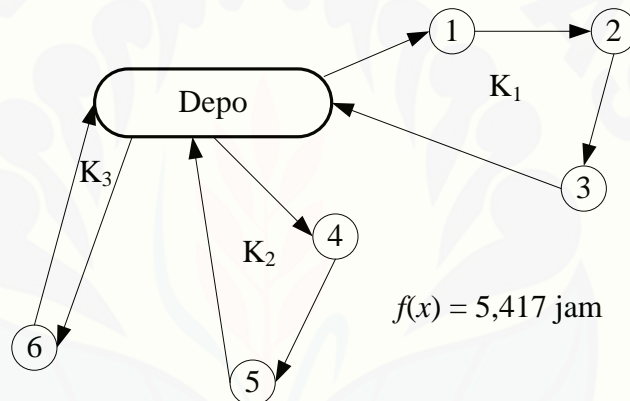
$$y_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in V, k \in K \quad (2.8)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in V, k \in K \quad (2.9)$$

Menurut Agyei, *et al* (2015) faktor yang dapat mempengaruhi waktu distribusi yaitu kecepatan kendaraan, kondisi jalan, produktivitas pengemudi dan kemacetan lalu lintas. Agar dapat dengan mudah menghitung perkiraan waktu distribusi maka peneliti akan menggunakan jarak antar kota dan kecepatan rata-rata pada persamaan (2.10) sebagai berikut:

$$T = \frac{\text{jarak}}{\text{kecepatan rata-rata}} \quad (2.10)$$

Solusi dari penyelesaian CVRPTW menghasilkan rute distribusi yang memiliki total waktu tempuh yang minimal dengan memenuhi kendala-kendala tersebut. Contoh penyelesaian dari CVRPTW dengan total waktu minimal perjalanan 5,417 jam dengan 3 rute dapat dilihat pada Gambar 2.2.



Gambar 2.2 Contoh Penyelesaian CVRPTW

2.4 Algoritma CODEQ

Perkembangan ilmu pengetahuan memberikan alternatif baru dalam penyelesaian permasalahan optimasi. Omran dan Salman pada tahun 2008 mengembangkan algoritma CODEQ yang merupakan metode metaheuristik untuk mencari titik optimal yang efektif dengan menggabungkan konsep dari algoritma *Chaotic Search*, *Opposition-based Learning*, *Differential Evolution* dan *Quantum Mechanics*. Algoritma CODEQ tidak memerlukan parameter kecuali untuk ukuran populasi. Untuk ukuran populasi parameter yang digunakan yaitu jumlah populasi dan iterasi maksimal. Performansi algoritma CODEQ terbukti menghasilkan solusi yang baik dan dapat diterapkan pada dimensi tinggi (Omran, 2010). Ada dua langkah utama dalam algoritma CODEQ yaitu, langkah pertama adalah

mutasi masing-masing individu. Langkah kedua adalah mengganti individu terburuk dengan kebalikannya atau dengan solusi di sekitar individu terbaik.

Berikut penjelasan metode-metode yang digunakan dalam penggabungan algoritma CODEQ adalah sebagai berikut :

a. *Differential Evolution* (DE)

Metode ini merupakan metode optimasi matematis fungsi multidimensional yang ditemukan oleh Price dan Storn pada tahun 1995. DE juga termasuk kelompok *Evolutionary Algorithm* (EA) (Omran, 2010). Populasi awal dari DE dipilih secara acak yang mencakup seluruh bagian ruang parameter. DE dapat menghasilkan populasi baru dengan menambahkan perbedaan antara vektor populasi kedua dan ketiga. Proses tersebut dinamakan operasi mutasi. Langkah berikutnya yaitu *crossover* yang diterapkan untuk menghasilkan keturunan (*trial vector*) dengan menggabungkan vektor mutasi dan vektor lain yang telah ditentukan (vektor target). Jika vektor keturunan menghasilkan nilai fungsi tujuan lebih rendah atau nilai *fitness* lebih baik dari vektor target, maka vektor keturunan menggantikan vektor target di iterasi berikutnya. Operasi terakhir disebut seleksi (Storn & Kenneth, 1997).

Ada tiga tahap dalam menyelesaikan metode *Differential Evolution* yang mengacu pada operator *Evolutionary Algorithm*, yaitu :

1) mutasi

Setiap vektor target ($x_i(t)$) dan vektor keturunan ($v_i(t)$) dapat dimunculkan berdasarkan mutasi dari target vektor. Vektor target ($x_i(t)$, $x_{i1}(t)$, $x_{i2}(t)$) dipilih secara acak dari populasi dimana $i_1 \neq i_2$. Kemudian dihitung dari perbedaan vektor $x_{i1}(t)$ dan $x_{i2}(t)$ yang akan menghasilkan vektor keturunan dengan persamaan sebagai berikut:

$$v_i(t) = x_i(t) + F(x_{i1}(t) - x_{i2}(t)) \quad (2.11)$$

dimana $F \in (0, \infty)$ merupakan faktor skala yang digunakan untuk mengontrol amplifikasi dari variasi diferensial ($x_{i1}(t) - x_{i2}(t)$).

2) *crossover*

Pada tahap *crossover*, DE melakukan pendekatan diskrit dimana parameter dari vektor target ($x_i(t)$) digabung dengan parameter dari vektor keturunan ($v_i(t)$)

yang akan menghasilkan *offspring* $\mu_i(t)$, dengan menggunakan persamaan sebagai berikut :

$$\mu_{ij}(t) = \begin{cases} v_{ij}(t), & U(0,1) < P_r \text{ atau } j = r \\ x_{ij}(t), & \text{yang lain} \end{cases} \quad (2.12)$$

dimana $U(0,1)$ merupakan evaluasi dari membangkitkan bilangan acak. Sedangkan P_r merupakan konstanta *crossover* yang ditentukan oleh pengguna dan r adalah indeks yang dipilih secara acak yang menjamin μ_i mendapat setidaknya satu parameter dari v_i .

3) seleksi

DE mengimplementasikan tahapan seleksi yang sangat sederhana. Vektor keturunan yang dihasilkan ($\mu_i(t)$) dapat menggantikan vektor target ($x_i(t)$), hanya jika nilai *fitness* dari vektor keturunan lebih baik dari vektor target.

b. *Opposition-based Learning* (OBL)

Opposition-based Learning pertama kali diusulkan oleh Tizhoosh pada tahun 2005 dan diterapkan untuk beberapa masalah oleh Rahnamayan dkk. Konsep dari OBL adalah pertimbangan atas estimasi dan berhubungan dengan estimasi berlawanan secara bersamaan untuk mendekati solusi yang akan diuji (Omran, 2010). OBL merupakan pendekatan baru untuk *machine intelligence*, yang telah banyak digunakan dalam permasalahan optimasi, *neural networks training*, dan *reinforcement learning*. Untuk menangani masalah optimasi, OBL didasarkan pada pemanfaatan *opposite number* dari arah pencarian lokasi baru (Goa *et al*, 2012). Misalkan $x = (x_1, x_2, \dots, x_n)$ adalah titik pencarian tunggal pada ruang solusi dimensi- n dan $x_i \in (a_i, b_i), i = 1, 2, \dots, n$. Maka *opposite number* $x' = (x'_1, x'_2, \dots, x'_n)$ dari $x = (x_1, x_2, \dots, x_n)$ dapat didefinisikan sebagai berikut :

$$x'_i = a_i + b_i - x_i, \quad i = 1, 2, \dots, n \quad (2.13)$$

c. *Chaotic Search* (CS)

Chaos merupakan perilaku waktu asimtotik dan aperiodik dalam sistem deterministik yang menunjukkan ketergantungan pada kondisi awal dan termasuk *infinite unstable periodic motions* dalam persamaan nonlinear. Hal yang mendasar

tentang teori *Chaos* adalah perubahan terkecil dalam sistem dapat menyebabkan perbedaan yang sangat besar dalam perilaku sistem.

Dalam matematika, *chaotic map* (persamaan *chaotic*) merupakan peta yang memiliki beberapa jenis perilaku *chaotic*. Persamaan tersebut diparameterisasi dengan waktu diskrit atau parameter waktu kontinu. Persamaan diskrit biasanya mengambil bentuk fungsi iterasi. *Chaotic map* sering ditemukan dalam penelitian sistem dinamik. Contoh *chaotic map* antara lain *Henon Map*, *Tent Map*, *Lorenz attractor*, *Piecewise Linear Chaotic Map* dan *Logistic Map*. *Piecewise linear chaotic map* (PWLCM) merupakan *chaotic map* yang sederhana dan efisien dengan perilaku dinamis yang baik (Omran, 2010). Berikut persamaan dari PWLCM:

$$x(t) = \begin{cases} \frac{x(t-1)}{p}, & x(t-1) \in (0, p) \\ \frac{1-x(t-1)}{1-p}, & x(t-1) \in [p, 1) \end{cases} \quad (2.14)$$

PWLCM berperilaku *chaotic* pada $(0,1)$ ketika $p \in (0, 0,5) \cup (0,5, 1)$. Variabel *chaotic* (x) dapat diinputkan secara acak yaitu $x(0) \sim U(0, 1)$ untuk melakukan *chaotic search* dengan hasil yang baik (Omran, 2010).

d. *Quantum Mechanics* (QM)

Dalam *newton mechanics*, sebuah partikel memiliki posisi dan kecepatan yang menentukan pada lintasannya. Namun, pada *Quantum Mechanics*, posisi partikel dan kecepatan tidak dapat ditentukan secara bersamaan yang sesuai dengan prinsip ketidakpastian. Oleh karena itu, istilah lintasan jarak pendek tidak berarti. Sun, dkk (Omran, 2010) mengusulkan model *quantum* dari *particle swarm optimization* (PSO) yang disebut *quantum particle swarm optimization* (QPSO), dimana partikel dapat bergerak menurut persamaan berikut:

$$x(t) = g \pm \frac{L}{2} \ln\left(\frac{1}{u}\right) \quad (2.15)$$

dengan g adalah penarikan lokal, L adalah parameter yang harus menuju ke nol bilamana $t \rightarrow \infty$ untuk menjamin konvergensi, sedangkan u merupakan bilangan acak seragam $(0,1)$ (Omran, 2010).

Tujuan dari algoritma CODEQ yaitu dapat mengatasi kelemahan dari algoritma *Differential Evolution* dengan menggunakan algoritma *Chaotic Search*,

Opposition-based Learning dan *Quantum Mechanics*. Dalam langkah algoritma CODEQ, proses crossover pada metode DE dikeluarkan untuk menghindari kekurangan pada algoritma CODEQ.

Berikut langkah-langkah algoritma CODEQ untuk menyelesaikan permasalahan CVRPTW, yaitu :

- input data. Data yang digunakan merupakan data distribusi,
- inisialisasi parameter (jumlah populasi), kemudian membangkitkan solusi awal sejumlah populasi yang dibangkitkan secara random. Sedangkan populasi pada kasus CVRPTW merupakan semua urutan rute pengiriman,
- mutasi. Pembangkitan vektor mutan $v_i(t)$ dari vektor target $r_i(t)$ untuk setiap iterasi t . Untuk mencari vektor target dapat dilakukan dengan cara memilih tiga individual (x_i, x_{i1}, x_{i2}) secara random, dimana $x_{i1} \neq x_{i2}$.

$$v_i(t) = x_i + (x_{i1} - x_{i2}) \ln\left(\frac{1}{u}\right) \quad (2.16)$$

Persamaan tersebut diturunkan dari metoda *Quantum Mechanics*,

- perbandingan, membandingkan nilai mutan $v_i(t)$ dan nilai vektor target $r_i(t)$, apabila nilai fungsi tujuan lebih rendah atau nilai *fitness* mutan $v_i(t)$ lebih baik dari vektor target maka mutan $v_i(t)$ menggantikan vektor target $r_i(t)$. Jika nilai fungsi tujuan lebih rendah atau nilai *fitness* vektor target lebih baik dari mutan $v_i(t)$ maka tetap menggunakan vektor target $r_i(t)$ untuk melakukan iterasi selanjutnya. Vektor hasil perbandingan dapat ditulis dengan vektor $x_i(t)$,
- pembangkitan vektor baru $w(t)$ yang digunakan untuk perbandingan selanjutnya agar dapat meningkatkan konvergensi

$$w(t) = \begin{cases} LB + UB - r \times x_b(t), & r \leq 0,5 \\ x_g(t) + |x_{i1}(t) - x_{i2}(t)| \times (2c(t) - 1), & r > 0,5 \end{cases} \quad (2.17)$$

dimana,

LB : batas bawah permasalahan

UB : batas atas permasalahan

$x_b(t)$: nilai *worst (least fittest)* vektor pada iterasi t atau nilai individu yang paling jelek

$x_g(t)$: nilai *best (least fittest)* vektor pada iterasi t atau nilai individu yang paling baik

$x_{i1}(t)$ dan $x_{i2}(t)$: vektor yang dipilih secara random, dimana $x_{i1}(t) \neq x_{i2}(t)$

$c(t)$: *variable chaotic*, yang didapatkan dari persamaan berikut:

$$c(t) = \begin{cases} \frac{c(t-1)}{p}, & c(t-1) \in (0, p) \\ \frac{(1-c(t-1))}{p}, & c(t-1) \in (p, 1) \end{cases} \quad (2.18)$$

dengan $c(0)$ dan p merupakan bilangan random $(0,1)$. Nilai $w(t)$ yang akan diperoleh bergantung pada bilangan random r . Penetapan nilai 0,5 mengacu pada algoritma yang telah dikembangkan oleh Omran dan Ayed (2009),

- f. perbandingan nilai $x_i(t)$ dan nilai $w(t)$, apabila nilai fungsi tujuan lebih rendah atau nilai *fitness* $w(t)$ lebih baik dari nilai x_i maka menggunakan $w(t)$ sebagai solusi iterasi pada tahap ini. Jika sebaliknya maka menjadi solusi tetap dari x_i ,
- g. pemberhentian iterasi. Jika iterasi maksimal belum tercapai maka kembali pada langkah mutasi dengan menggunakan individu hasil perbandingan x_i dan $w(t)$ dan selanjutnya sampai pada kriteria pemberhentian (Omran, 2010),

Solusi awal untuk mengaplikasikan algoritma CODEQ dalam permasalahan CVRPTW dengan membangkitkan data populasi secara random kemudian mencari nilai fungsi tujuan dari populasi tersebut yang sesuai dengan kendala CVRPTW. Sehingga data tersebut dapat dibandingkan dengan nilai fungsi tujuan vektor mutan.

Pada tahap mutasi dan membangkitkan vektor $w(t)$ akan menghasilkan bilangan real, sehingga perlu diubah menjadi bilangan integer dengan menggunakan metode *integer order criterion* agar dapat membentuk rute selanjutnya. Gen dengan bilangan real terbesar akan menjadi pelanggan dengan *ordinal number* terbesar (n), selanjutnya gen dengan bilangan real terbesar kedua akan menjadi pelanggan ($n-1$), dan seterusnya (Garside dan Satya, 2014).

Menurut Chio dan Chung (2013), pemilihan parameter algoritma *Differential Evolution* sangat penting. Namun, parameter tersebut sangat sensitif dalam menyelesaikan sebuah permasalahan. Misalnya pada parameter F (faktor skala) pada tahap mutasi. Jika faktor skala yang dimasukkan dengan bilangan kecil maka solusi yang didapatkan akan semakin baik, namun akan membutuhkan konvergensi yang lama. Jika faktor skala yang dimasukkan dengan bilangan besar

maka solusi yang didapatkan akan cepat konvergen, namun akan mudah terjebak pada solusi lokal optimum. Sehingga pada algoritma CODEQ, parameter F pada tahap mutasi digantikan dengan metode *Quantum Mechanics*. Pada metode *Quantum Mechanics* tersebut menggunakan bilangan random $u(0,1)$. Lin *et al* (2000) menggunakan bilangan random (0,1) untuk menggantikan parameter F , namun tidak menjamin akan menghasilkan solusi konvergen yang cepat dan mendapatkan hasil solusi yang sbaik.

2.5 Algoritma *Differential Evolution Plus*

Algoritma *Differential Evolution Plus* merupakan modifikasi dari algoritma *Differential Evolution*. Perbedaan yang dimiliki algoritma *DE Plus* dengan algoritma *DE*, yaitu dengan melakukan tiga modifikasi. Pertama, memodifikasi pengendalian nilai parameter F dan Cr (*adaptive parameter*) dengan cara menghitung F dan Cr pada tiap generasi berubah-ubah. Kedua, menambahkan prosedur *Local Search* pada *DE* untuk meningkatkan kualitas solusi yang dihasilkan. Ketiga, *Crossover* pada mulanya merupakan langkah untuk menyilangkan populasi target dengan populasi mutan, dimodifikasi menjadi langkah untuk memilih populasi target yang akan dimutasi sehingga dilakukan sebelum langkah mutasi (Wiratno *et al.*, 2012).

Adapun tahapan yang digunakan untuk menyelesaikan algoritma *Differential Evolution Plus*, yaitu:

a. *Crossover*

Langkah *crossover* pada algoritma *DE* dilakukan dengan menyilangkan setiap vektor $x_{i,g}$ dengan vektor mutan $v_{i,g}$ untuk membentuk vektor hasil persilangan $u_{i,g}$ yang menggunakan persamaan (2.19)

$$u_{i,g} = u_{i,i,g} = \begin{cases} v_{j,i,g}, & \text{jika } j(\text{rand}_j(0,1)) \leq Cr \text{ atau } j = j_{rand} \\ x_{j,i,g}, & \text{jika } j \text{ yang lain} \end{cases} \quad (2.19)$$

Pada algoritma *DE Plus*, *crossover* dilakukan sebelum mutasi untuk memilih populasi yang akan dimutasi. Konsep dari *crossover* tersebut yaitu apabila bilangan random lebih kecil atau sama dengan Cr maka populasi tersebut akan dimutasi. Namun, jika bilangan random lebih besar dari Cr maka populasi

tersebut tidak akan dimutasi. Probabilitas *crossover* ($Cr \in (0,1)$), merupakan nilai yang didefinisikan untuk mengendalikan fraksi nilai parameter yang disalin dari mutan. Probabilitas *crossover* untuk tiap generasi akan ditentukan menggunakan persamaan (2.20):

$$Cr = Cr_{min} + G \cdot \frac{Cr_{max} - Cr_{min}}{MAXGEN} \quad (2.20)$$

dimana,

Cr = probabilitas *crossover*

Cr_{min} = nilai terkecil dari probabilitas *crossover*

Cr_{max} = nilai terbesar dari probabilitas *crossover*

G = iterasi pada saat waktu *running time*

MAXGEN = jumlah maksimum iterasi

Tujuan dari penentuan nilai Cr untuk meningkatkan keragaman vektor yang akan mengalami *crossover* dan menghindari dari local optimal.

b. Mutasi

Mutasi pada algoritma DE dan Algoritma DE *Plus* dilakukan dengan cara yang sama. Algoritma DE *Plus* melakukan mutasi dan kombinasi terhadap populasi yang telah dipilih pada tahap *crossover*. Formulasi yang digunakan untuk menyelesaikan proses mutasi dapat dilihat pada persamaan (2.21):

$$v_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (2.21)$$

dimana,

F = parameter mutasi

$x_{r0,g}, x_{r1,g}, x_{r2,g}$ = vektor acak

Faktor skala $F \in (0,1)$ merupakan bilangan real positif yang mengendalikan tingkat populasi. Sehingga nilai parameter F tiap generasi akan berubah-ubah dengan menghitung nilai parameter pada tiap generasi yang menggunakan persamaan (2.22)

$$F = \begin{cases} \max\left(F_{min}, 1 - \left|\frac{f_{max}}{f_{min}}\right|\right), & \text{jika } \left|\frac{f_{max}}{f_{min}}\right| < 1 \\ \max\left(F_{min}, 1 - \left|\frac{f_{min}}{f_{max}}\right|\right), & \text{jika yang lain} \end{cases} \quad (2.22)$$

c. Seleksi

Tahap seleksi akan diperoleh dari solusi yang terbaik. Formulasi tahap seleksi dapat dilihat pada persamaan (2.23):

$$x_{i,g+1} = \begin{cases} v_{i,g}, & \text{jika } f(v_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}, & \text{jika sebaliknya} \end{cases} \quad (2.23)$$

d. *Local Search*

Pada tahap ini, hasil dari proses sebelumnya akan dikenai prosedur *local search*. Prosedur *local search* yang digunakan merupakan *insert-based local search*. *Insert-based local search* cenderung mengarahkan pencarian ke daerah solusi yang menjanjikan dalam waktu yang relatif singkat. Cara kerja prosedur *insert-based local search* yaitu dengan menyisipkan titik u ke posisi titik v (dimana $u \neq v$) (Wiratno *et al.*, 2012).

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan untuk penelitian ini merupakan data sekunder yang diambil dari penelitian Ulyawati (2016) yang berupa data distribusi dari perusahaan air mineral dan data tersebut tidak mencantumkan data mengenai data *time windows*. Sehingga peneliti menambahkan data *time windows* dan mengasumsikan jenis dan kapasitas kendaraan sejenis yang dibutuhkan untuk menyelesaikan permasalahan CVRPTW. Data yang digunakan adalah sebagai berikut:

- a. terdapat data distribusi 32 pelanggan dan 57 pelanggan yang berada di Kabupaten Jember,
- b. jumlah permintaan barang dan waktu operasional pada depo dan setiap toko untuk 32 pelanggan (lihat Lampiran A1) dan untuk 57 pelanggan (lihat Lampiran A4),
- c. jarak antara depo dengan toko dan jarak antar toko untuk 32 pelanggan (lihat Lampiran A2) dan untuk 57 pelanggan (lihat Lampiran A5),
- d. kecepatan rata-rata kendaraan dari depo ke setiap toko dan jarak antar toko dengan toko yang lain untuk 32 pelanggan (lihat Lampiran A3) dan untuk 57 pelanggan (lihat Lampiran A6),
- e. waktu pelayanan setiap satu kali angkut per karton dari kendaraan ke toko yaitu 20 detik,
- f. kendaraan memiliki kapasitas 600 karton,
- g. waktu operasional pada depo dalam proses pengiriman barang selama 10 jam yang dimulai pada pukul 07.00-17.00.

3.2 Langkah-langkah Penelitian

Langkah-langkah yang akan dilakukan untuk menyelesaikan permasalahan CVRPTW dengan menggunakan algoritma CODEQ dan modifikasi algoritma CODEQ, seperti berikut :

- a. Studi Literatur

Langkah awal yang dilakukan dalam penelitian ini adalah melakukan studi literatur yang diperoleh tentang *Vehicle Routing Problem*, *Capacitated Vehicle Routing Problem with Time Windows*, algoritma CODEQ dan algoritma DE Plus dari artikel-artikel di internet dan sumber lain yang berhubungan dengan permasalahan dan algoritma tersebut.

b. Pengambilan dan Pengumpulan Data

Kemudian pengambilan data. Data yang diperoleh antara lain jarak dari depo dan toko, jumlah permintaan barang dari setiap toko, waktu pelayanan pada setiap toko, waktu perjalanan, dan jumlah kendaraan yang digunakan dalam pendistribusi barang.

c. Menerapkan Algoritma CODEQ pada CVRPTW

Pada tahap ini, data yang diperoleh diterapkan pada kasus CVRPTW dengan menerapkan algoritma CODEQ dengan langkah-langkah sebagai berikut:

- 1) input data,
- 2) inisialisasi parameter (populasi), kemudian membangkitkan solusi awal sejumlah populasi awal yang dibangkitkan secara random. Dimana populasi merupakan semua urutan rute pengiriman (kandidat solusi). Dari kandidat solusi tersebut akan dipecah menjadi beberapa rute yang memenuhi kendala CVRPTW untuk menentukan nilai fungsi tujuannya,
- 3) mutasi, dengan membangkitkan vektor mutan $v_i(t)$ yang hasilnya berupa urutan rute yang memenuhi kendala CVRPTW agar memperoleh nilai fungsi tujuan,
- 4) perbandingan, nilai fungsi tujuan mutan dibandingkan dengan nilai fungsi tujuan populasi. Jika nilai fungsi tujuan mutan rendah maka akan menggantikannya yang akan digunakan pada proses selanjutnya ($x_i(t)$),
- 5) pembangkitan vektor $w(t)$ yang hasilnya berupa urutan rute yang memenuhi kendala CVRPTW agar memperoleh nilai fungsi tujuan,
- 6) perbandingan, nilai fungsi tujuan $x_i(t)$ pada proses sebelumnya dibandingkan dengan nilai fungsi tujuan vektor $w(t)$. Jika nilai fungsi tujuan vektor $w(t)$ rendah maka akan menggantikannya yang akan digunakan pada proses selanjutnya,

- 7) kriteria pemberhentian jika iterasi maksimal terpenuhi. Apabila belum terpenuhi maka kembali pada langkah mutasi.
 - 8) seleksi, memilih solusi yang paling optimal (nilai fungsi tujuan paling rendah) dari proses sebelumnya.
- d. Kelemahan dari algoritma CODEQ terdapat pada langkah mutasi. Pada langkah tersebut menggunakan metode *Quantum Mechanics* untuk menggantikan parameter F pada algoritma *Differential Evolution* dengan menggunakan bilangan random $u(0,1)$. Lin *et al* (2000) menggunakan bilangan random $(0,1)$ untuk menggantikan parameter F , namun tidak menjamin akan menghasilkan solusi konvergen yang cepat dan mendapatkan hasil solusi yang baik. Sehingga peneliti memodifikasi algoritma CODEQ tanpa mengubah karakteristik dari metode *Quantum Mechanics*, yaitu dengan menambahkan langkah *local search* pada algoritma CODEQ. Modifikasi ini berdasarkan algoritma *Differential Evolution Plus*.
- e. Menerapkan Modifikasi Algoritma CODEQ pada CVRPTW
- Pada tahap ini, data yang diperoleh diterapkan pada kasus CVRPTW dengan menerapkan modifikasi algoritma CODEQ dengan langkah-langkah sebagai berikut :
- 1) input data,
 - 2) inisialisasi parameter (populasi), kemudian membangkitkan solusi awal sejumlah populasi awal yang dibangkitkan secara random. Dimana populasi merupakan semua urutan rute pengiriman (kandidat solusi). Dari kandidat solusi tersebut akan dipecah menjadi beberapa rute yang memenuhi kendala CVRPTW untuk menentukan nilai fungsi tujuannya,
 - 3) mutasi, dengan membangkitkan vektor mutan $v_i(t)$ yang hasilnya berupa urutan rute yang memenuhi kendala CVRPTW agar memperoleh nilai fungsi tujuan,
 - 4) perbandingan, nilai fungsi tujuan mutan dibandingkan dengan nilai fungsi tujuan populasi. Jika nilai fungsi tujuan vektor mutan rendah maka akan menggantikannya yang akan digunakan pada proses selanjutnya ($x_i(t)$),

- 5) *local search*, hasil dari proses sebelumnya akan dikenai prosedur *insert-based local search*. Dengan cara menyisipkan titik u ke posisi titik v (dimana $u \neq v$) pada sebuah urutan rute. Hasil dari *local search* berupa urutan rute yang memenuhi kendala CVRPTW agar memperoleh nilai fungsi tujuan,
- 6) perbandingan, nilai fungsi tujuan *local search* dibandingkan dengan nilai fungsi tujuan sebelumnya. Jika nilai fungsi tujuan *local search* rendah maka akan menggantikannya yang akan digunakan pada proses selanjutnya ($x_i(t)$),
- 7) pembangkitan vektor $w(t)$ yang hasilnya berupa urutan rute yang memenuhi kendala CVRPTW agar memperoleh nilai fungsi tujuan,
- 8) perbandingan, nilai fungsi tujuan vektor $w(t)$ dibandingkan dengan nilai fungsi tujuan sebelumnya. Jika nilai fungsi tujuan vektor $w(t)$ rendah maka akan menggantikannya yang akan digunakan pada proses selanjutnya,
- 9) kriteria pemberhentian jika iterasi maksimal terpenuhi. Apabila belum terpenuhi maka kembali pada langkah mutasi.
- 10) seleksi, memilih solusi yang paling optimal (nilai fungsi tujuan paling rendah) dari proses sebelumnya.

f. Pembuatan Program Matlab

Selanjutnya membuat program dengan menggunakan *software* matematika, yaitu menggunakan MATLAB R2009a. Pada langkah ini peneliti membuat skrip program berupa tampilan GUI untuk penyelesaian CVRPTW dengan menggunakan algoritma CODEQ dan modifikasi algoritma CODEQ.

g. Simulasi program dilakukan dengan menggunakan data yang didapat pada langkah b.

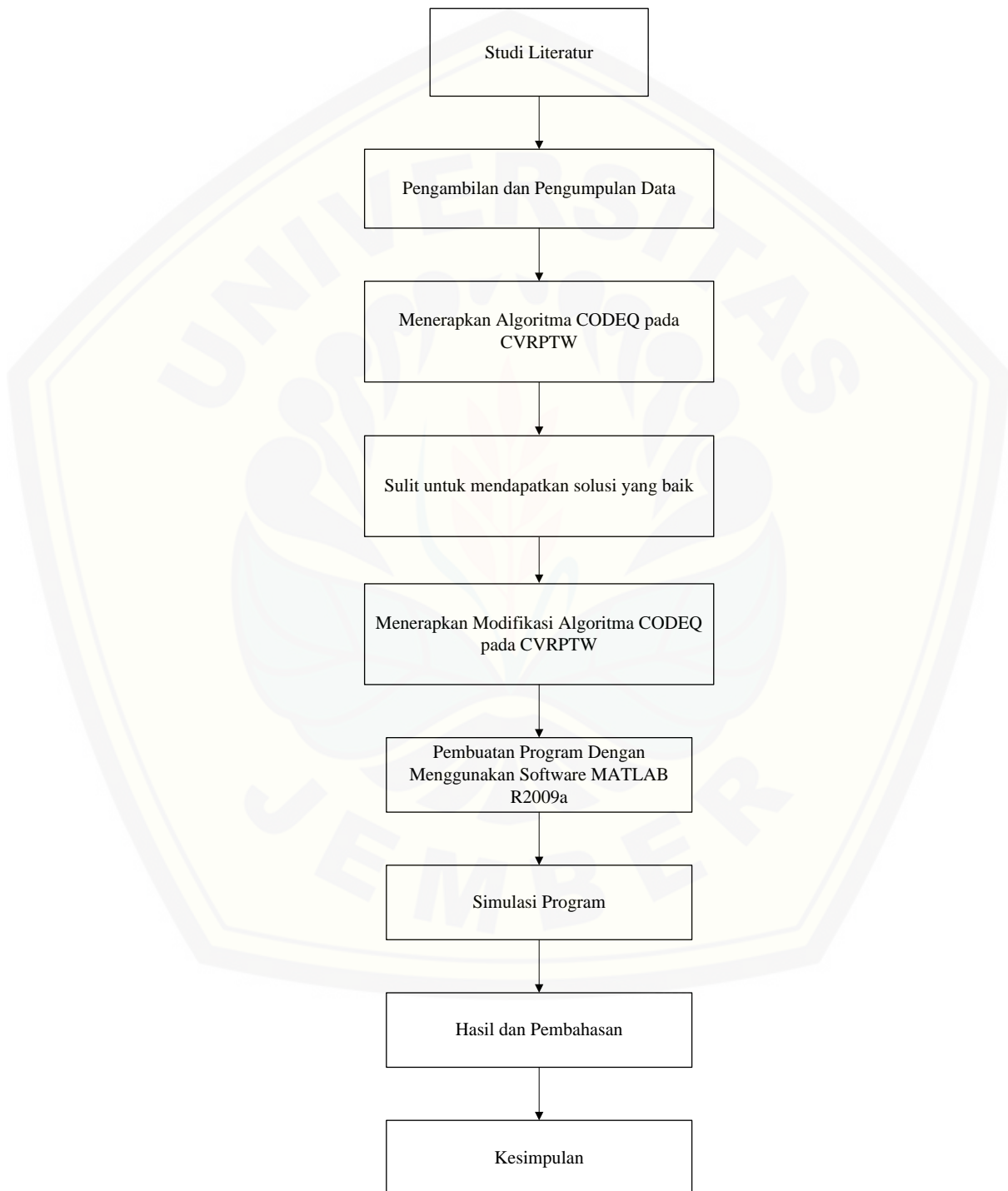
h. Hasil dan Pembahasan

Dalam penelitian ini, parameter yang dapat diubah-ubah yaitu jumlah populasi. Hasil dari penelitian ini berupa rute kendaraan, total waktu perjalanan, total permintaan dan total jarak dari setiap rute kendaraan yang dihasilkan untuk menyelesaikan *Capacitated Vehicle Routing Problem with Time Windows* dengan menggunakan algoritma CODEQ dan modifikasi algoritma CODEQ.

i. Kesimpulan

Hasil dari simulasi program akan ditarik kesimpulan untuk menjawab tujuan masalah.

Skema langkah-langkah penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema Langkah-langkah Penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

- a. Algoritma CODEQ dan modifikasi algoritma CODEQ dapat diterapkan pada kasus CVRPTW. Algoritma tersebut dapat menyelesaikan data berukuran kecil maupun data yang berukuran besar yaitu menggunakan data 32 toko dan 57 toko. Modifikasi algoritma CODEQ dapat menghasilkan solusi yang lebih baik dibandingkan algoritma CODEQ pada kedua data yang digunakan. Namun, algoritma CODEQ memerlukan *running time* lebih cepat dibandingkan modifikasi algoritma CODEQ. Modifikasi algoritma CODEQ mendapatkan solusi terbaik yaitu 19 jam 22 menit 13 detik dengan *running time* sebesar 308,1974 Detik pada 80 populasi 3000 iterasi untuk data 32 toko, dan 54 jam 29 menit 11 detik dengan *running time* sebesar 832,9274 detik pada 100 populasi 5000 iterasi untuk data 57 toko. Sedangkan algoritma CODEQ mendapatkan solusi terbaik yaitu 22 jam 17 menit 29 detik yang memerlukan *running time* sebesar 169,9125 detik pada 80 populasi 3000 iterasi untuk data 32 toko, dan 58 jam 16 menit 32 detik dengan *running time* sebesar 438,661 detik pada 100 populasi 5000 iterasi untuk data 57 toko. Total waktu perjalanan dari semua kendaraan yang dihasilkan modifikasi algoritma CODEQ lebih minimum daripada hasil algoritma CODEQ.
- b. Semakin banyak solusi yang mungkin (jumlah populasi) yang direpresentasikan maka kemungkinan untuk mendapatkan hasil solusi yang mendekati optimal lebih besar. Tetapi, jika jumlah populasi besar namun iterasi maksimal yang dimasukkan sedikit maka hasil yang didapatkan kurang optimal, karena solusi yang diuji memiliki ruang pencarian yang kecil. Ada kemungkinan jika penambahan iterasi semakin besar akan menghasilkan solusi yang mendekati optimal.

5.2 Saran

Pada penelitian ini menggunakan kendaraan yang identik sehingga kapasitas kendaraan dianggap sama dan jumlah kendaraannya tidak ditetapkan. Namun, pada kehidupan nyata jarang sekali perusahaan menggunakan satu jenis kendaraan yang identik. Sehingga pada penelitian selanjutnya diharapkan dapat menyelesaikan dengan kendaraan yang tidak identik (kapasitas berbeda) dan jumlah kendaraannya telah ditetapkan.



DAFTAR PUSTAKA

- Agyei, W., Kwaku, F. D., William, O. D. & Emmanuel, A. A. 2015. Modelling Newspaper Distribution as Capacitated Vehicle Routing Problem with Time Windows : Case Study of Daily Graphic Newspaper, Ashanti Region, Ghana. *Archives of Current Research International*. 2(1): 12-22.
- Caric, T. & Hrvoje, G. 2008. *Vehicle Routing Problem*. Austria : In-The Publishing.
- Chio, J. P. & Chung, F. C. 2013. A Novel Evolutionary Algorithm for Capacitor Placement in Distribution Systems. *GSTF Journal of Engineering Technology (JET)*. 2(3): 1-5.
- Garside, A. K. & Satya, S. 2014. Performasi Algoritma CODEQ dalam Penyelesaian Vehicle Routing Problem. *Jurnal Teknik Industri*. 16(1):51-56.
- Gunarso, T. 2014. Aplikasi Algoritma CODEQ untuk Penyelesaian Masalah Jobshop Scheduling pada Produksi Springbed. *Skripsi*. Jember: Universitas Jember.
- Goa, X. Z., Wang, X., Ovaska, S. J. & Zengker K. 2012. A hybrid optimization method of harmony search and opposition-based learning, *Engineering Optimization*. 44(8): 825-914.
- Lin, Y. C., Kao, S. H. & Feng, S. W. 2000. Plant Scheduling and Planning Using Mixed-integer Hybrid Differential Evolution with Multiplier Updating. *IEEE*. 1: 593-600.
- Munir, R. 2010. *Matematika Diskrit*. Bandung: Informatika Bandung.
- Omran, M. G. H. & Ayed, S. 2009. Constrained optimization using CODEQ. *Chaos, Solutions & Fractals*. 42: 662-668.
- Omran, M. G. H. 2010. CODEQ : An Effective Metaheuristic for Continuous Global Optimization. *International Journal Metaheuristics*. 1(2): 108-131.
- Reviladi, I. 2016. Implementasi Algoritma Floyd Warshall dan Nearest Neighbour dalam Pengoptimalan Rute Capacitated Vehicle Routing Problem with Time Windows. *Artikel Jurnal Skripsi*. Yogyakarta: Universitas Negeri Yogyakarta. <http://journal.student.uny.ac.id/ojs/index.php/math/article/view/4816/4470>. [Diakses pada 05 Oktober 2016].

- Sousa, J. S., Haidar, A. B., Rui, B. & Alexandre, S. 2011. A Multi Objective Approach to Solve Capacitated Vehicle Routing Problem with Time Windows Using Mixed Integer Linear Programming. *International Journal of Advanced Science and Technology*. 28: 1-8.
- Stron, R. & Kenneth, P. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*. 11: 341-359.
- Toth, P. & Daniele, V. 2002. The Vehicle Routing Problem. *Society For Industrial and Applied Mathematic*. Philadelphia, USA.
- Ulyawati, I. 2016. Penyelesaian Capacitated Vehicle Routing Problem dengan Algoritma Harmony Search dan Algoritma Tabu Search. *Skripsi*. Jember: Universitas Jember.
- Wiratno, S. E., Rudi, N. & Budi, S. 2012. Algoritma Differential Evolution Untuk Penjadwalan Flowshop Banyak Mesin Dengan Multy Obyektif. *Jurnal Teknik Industri Institut Teknologi Sepuluh Nopember*. 13(1): 1-6.

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	11	14	23	20	29	27	22	29	22	21	23	34	32	32	33
1	11	0	23	25	26	21	25	20	20	19	17	20	17	23	26	29
2	14	23	0	25	25	22	25	21	23	23	23	25	31	33	33	33
3	23	25	25	0	30	33	24	26	22	25	24	26	31	34	33	34
4	20	26	25	30	0	20	24	21	18	20	17	23	28	33	33	34
5	29	21	22	33	20	0	28	21	22	22	20	23	32	30	31	32
6	27	25	25	24	24	28	0	26	24	25	23	26	24	32	33	34
7	22	20	21	26	21	21	26	0	22	23	20	25	32	34	34	34
8	29	20	23	22	18	22	24	22	0	22	21	23	27	33	32	33
9	22	19	23	25	20	22	25	23	22	0	49	21	28	32	32	32
10	21	17	23	24	17	20	23	20	21	49	0	28	28	34	34	35
11	23	20	25	26	23	23	26	25	23	21	21	0	28	27	29	51
12	34	17	31	31	28	32	24	32	27	28	27	28	0	34	34	34
13	32	23	33	34	33	30	32	34	33	32	30	27	34	0	25	29

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

Kode Toko	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0 (depo)	34	25	31	26	30	32	31	32	33	31	33	32	34	33	34	30	36
1	28	22	31	22	28	29	30	29	29	35	32	31	35	34	35	31	30
2	35	28	32	27	32	32	33	33	33	32	33	33	33	32	35	31	30
3	34	28	33	28	33	35	31	35	32	32	34	34	36	35	35	32	32
4	34	27	32	25	32	33	32	34	31	32	34	34	34	33	35	31	30
5	33	25	29	25	29	38	32	35	33	33	35	34	36	35	33	30	32
6	34	28	31	27	32	35	31	34	32	32	34	33	36	35	35	33	32

Kode Toko	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
7	27	28	33	27	33	36	33	36	32	33	35	34	37	35	36	33	32
8	34	26	30	25	29	34	32	36	38	34	35	35	34	33	33	30	30
9	35	25	31	25	31	32	32	35	37	33	35	34	32	32	35	31	29
10	34	25	32	25	32	32	47	37	39	34	36	35	31	31	35	32	33
11	30	23	26	23	29	31	31	36	31	33	34	33	32	31	35	30	28
12	35	30	33	28	33	35	32	34	35	32	34	34	37	36	33	32	31
13	28	26	31	26	31	37	26	37	29	32	35	35	35	33	33	9	31
14	27	29	31	27	32	37	45	41	45	33	34	35	34	33	32	30	32
15	24	29	34	32	31	35	32	32	26	33	35	38	37	33	38	29	212
16	0	28	34	28	38	38	30	41	32	32	32	33	32	32	43	35	32
17	28	0	24	23	23	30	29	34	28	30	32	31	31	31	25	23	29
18	34	24	0	25	26	33	31	37	32	32	34	33	31	33	31	25	29
19	28	23	25	0	24	30	29	33	30	31	32	33	30	30	30	26	27
20	38	23	26	24	0	34	28	38	33	30	35	34	33	32	33	36	34
21	38	30	33	30	34	0	35	41	37	33	38	36	35	29	35	33	36
22	30	29	31	29	28	35	0	21	23	27	30	22	33	33	31	33	32
23	41	34	37	33	38	41	21	0	31	26	32	30	34	33	34	32	32
24	32	28	32	30	33	37	23	31	0	28	31	27	33	33	32	31	32
25	32	30	33	31	30	33	27	26	28	0	31	29	33	33	32	30	31
26	32	32	34	32	35	38	30	32	31	31	0	31	34	33	33	32	32
27	33	31	33	33	34	36	22	30	27	29	31	0	33	33	35	32	31
28	32	31	31	30	33	35	33	34	33	33	34	33	0	26	35	34	31
29	32	31	33	23	32	29	33	33	33	33	33	33	26	0	34	33	26
30	43	25	31	30	33	35	31	34	32	32	33	33	35	34	0	27	31
31	35	23	25	26	36	33	31	32	31	30	32	32	34	33	27	0	29
32	32	29	29	27	34	36	32	32	32	31	32	31	31	26	31	29	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	29	Biting	13	11.00	20.30
1	Gebang1	600	09.00	16.00	30	Pontang	70	11.00	18.30
2	Gebang2	254	11.00	19.30	31	Karang Anyar	117	12.00	20.30
3	Patrang	33	11.30	21.00	32	Ambulu	600	08.00	19.00
4	Slawu	25	10.30	18.00	33	Tempurejo	2	12.00	15.00
5	Mangli	83	14.00	21.30	34	Ledokombo	51	08.30	19.30
6	Kaliwates	121	09.30	17.30	35	Suren	2	10.00	15.30
7	Kepatihan1	600	12.00	19.00	36	Sumberlesung	2	11.00	17.00
8	Kepatihan2	518	09.30	18.00	37	Garahan	83	14.00	17.0
9	Tegal Besar	23	09.00	16.00	38	Pace	17	12.00	21.00
10	Sempusari	8	10.00	18.30	39	Silo	5	12.00	19.30
11	Jember Kidul	104	13.30	20.00	40	Sumberjati	4	13.30	21.00
12	Wirolegi	10	11.30	18.00	41	Balung Lor	160	08.00	17.00
13	Karangrejo	22	11.30	20.30	42	Tutul	4	11.00	17.30
14	Kranjingan	33	09.00	16.00	43	Balung Kulon	1	14.00	21.00
15	Kebonsari	27	08.00	14.30	44	Suger Kidul	18	08.00	16.0
16	Tegal Gede	112	12.00	19.30	45	Jelbuk	121	12.00	19.00
17	Sumbersari	271	15.00	21.30	46	Bangsalsari	11	07.30	14.00
18	Antirogo	2	08.00	15.00	47	Tamansari	5	10.00	18.30
19	Sukorambi	15	08.30	17.30	48	Kertosari	10	12.30	20.00
20	Dukuh Mencek	2	07.00	14.30	49	Pakusari	5	08.00	15.00
21	Kalisat	19	13.30	20.00	50	Sumberpinang	4	13.30	20.30
22	Ajung Kalisat	19	11.00	18.00	51	Mangaran	4	12.00	18.30
24	Glagahwero	13	14.00	22.00	52	Ajung	3	14.00	21.00
25	Gumuksari	52	10.30	18.30	53	Karang Kedawung	69	12.00	19.30
26	Arjasa	144	11.30	18.30	56	Jatimulyo	13	13.30	21.00
27	Candijati	13	12.30	19.30					
28	Darsono	22	07.30	17.00	57	Jenggawah	105	08.00	17.00

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

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

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

Kode Toko	13	14	15	16	17	18	19	20	21	22	23	24	25
31	38	39	39	36	34	31	33	34	36	35	36	38	35
32	25	25	25	26	25	25	26	24	27	27	28	28	26
33	30	30	31	30	30	29	30	29	32	31	31	32	30
34	33	33	33	33	32	30	31	33	29	30	32	30	32
35	38	39	37	38	36	35	33	34	37	27	32	41	42
36	34	34	34	33	33	30	31	33	31	87	33	32	32
37	37	38	38	37	35	34	33	34	35	31	34	33	35
38	35	34	38	38	34	33	32	33	32	33	33	32	18
39	37	37	38	37	36	34	33	34	33	35	35	34	34
40	37	37	37	36	36	35	33	35	35	34	35	33	35
41	34	40	39	38	34	33	37	36	36	36	35	35	36
42	36	43	38	36	36	32	37	36	35	35	35	35	35
43	36	39	38	37	35	32	37	37	35	35	32	34	35
44	34	33	31	24	34	33	34	35	33	31	32	35	32
45	31	30	29	21	30	28	30	32	29	29	29	30	28
46	30	31	29	32	27	31	30	31	30	31	31	32	31
47	34	37	33	36	32	32	35	36	34	16	35	36	34
48	28	30	30	32	30	31	26	28	31	43	32	36	30
49	27	29	29	30	27	26	26	28	28	55	30	31	27
50	18	31	31	24	18	27	30	31	28	56	31	30	25
51	30	32	29	37	29	25	29	28	33	34	35	35	32
52	31	35	28	27	27	28	29	24	33	34	32	35	33
53	32	33	30	33	32	34	29	31	31	31	32	33	31
54	29	29	29	38	28	28	29	30	31	32	32	33	31
55	33	34	32	39	32	31	32	31	34	9	36	37	34
56	32	33	32	39	32	31	32	32	34	12	35	36	33
57	30	27	29	35	30	28	30	29	30	9	31	32	32

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

Kode Toko	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
31	30	34	30	32	28	0	21	26	33	38	35	35	31	35	35	34
32	25	26	25	26	19	21	0	20	27	29	28	28	26	29	29	25
33	28	30	28	29	26	26	20	0	31	32	31	33	30	33	33	28
34	29	31	29	30	33	33	27	31	0	21	20	27	27	30	22	33
35	33	36	33	37	33	38	29	32	21	0	31	29	29	32	30	35
36	30	31	29	33	35	35	28	31	20	31	0	27	29	31	31	35
37	33	33	31	59	35	35	28	33	27	29	27	0	29	27	28	121
38	31	32	31	33	30	31	26	30	27	29	29	29	0	31	29	33
39	32	34	32	34	35	35	29	33	30	32	31	27	31	0	31	35
40	32	33	31	34	35	35	29	33	22	30	31	28	29	31	0	34
41	29	30	28	30	31	34	25	28	33	35	35	121	33	35	34	0
42	28	30	28	30	29	32	24	28	33	34	34	35	33	34	36	26
43	28	29	28	30	56	28	25	31	33	33	34	35	33	34	36	26
44	27	32	29	30	31	32	27	31	31	34	32	33	32	35	33	35
45	23	26	26	25	30	30	30	29	31	31	34	33	31	34	35	33
46	27	28	26	28	33	31	26	30	31	33	33	33	32	33	34	31
47	30	31	30	32	29	30	23	28	32	36	36	36	34	38	38	35
48	29	34	28	32	33	35	25	31	31	32	33	34	29	34	35	32
49	27	30	29	29	33	34	25	30	28	30	31	32	31	34	34	32
50	27	33	27	30	33	34	26	31	30	31	30	33	31	35	35	33
51	27	29	26	28	28	30	22	28	34	35	35	37	33	36	35	22
52	26	29	25	28	34	32	23	29	35	36	35	37	33	36	37	34
53	31	33	31	34	30	31	25	30	30	29	32	34	29	35	35	34
54	31	33	26	34	30	33	25	29	31	31	33	34	30	35	34	33
55	28	29	28	30	28	33	23	30	35	35	36	36	34	40	38	36
56	28	31	29	31	22	27	21	26	33	34	34	35	32	39	35	29
57	26	27	26	29	29	20	20	26	30	34	31	43	30	33	32	31

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

Kode Toko	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
31	32	28	32	30	31	30	35	34	34	30	32	31	33	33	27	20
32	24	25	27	30	26	23	25	25	26	22	23	25	25	23	21	20
33	28	31	31	29	30	28	31	30	31	28	29	30	29	30	26	26
34	33	33	31	31	31	32	31	28	30	34	35	30	31	35	33	30
35	34	33	34	31	33	36	32	30	31	35	36	29	31	35	34	34
36	34	34	32	34	33	36	33	31	30	35	35	32	33	36	34	31
37	35	35	33	33	33	36	34	32	33	37	37	34	34	36	35	43
38	33	33	32	31	32	34	29	31	31	33	33	29	30	34	32	30
39	34	34	35	34	33	38	34	34	35	36	36	35	35	40	39	33
40	36	36	33	35	34	38	35	34	35	35	37	35	34	38	35	32
41	26	26	35	33	31	35	32	32	33	22	34	34	33	36	29	31
42	0	23	34	33	26	22	35	34	32	29	35	33	33	32	29	28
43	23	0	33	32	26	24	33	33	32	29	35	34	33	29	28	29
44	34	33	0	26	30	34	32	34	37	38	34	34	34	32	32	31
45	33	32	26	0	29	33	32	28	29	29	30	31	32	31	31	28
46	26	26	30	29	0	35	32	32	31	30	26	32	30	32	32	29
47	22	24	34	33	35	0	36	36	35	23	20	35	33	35	28	32
48	35	33	32	32	32	36	0	20	33	35	36	30	34	36	34	31
49	34	33	34	28	32	36	20	0	23	33	33	29	31	34	32	28
50	32	32	37	29	31	35	33	23	0	33	40	26	41	35	33	31
51	29	29	38	29	30	23	35	33	33	0	29	30	30	28	29	23
52	35	35	34	30	26	20	36	33	40	29	0	30	30	36	35	27
53	33	34	34	31	32	35	30	29	26	30	30	0	29	34	33	29
54	33	33	34	32	30	33	34	31	41	30	30	29	0	33	32	28
55	32	29	32	31	32	35	36	34	35	28	36	34	33	0	25	27
56	29	28	32	31	32	28	34	32	33	29	35	33	32	25	0	22
57	28	29	31	28	29	32	31	28	31	23	27	29	28	27	22	0

Lampiran B. Hasil dari Percobaan algoritma CODEQ dengan menggunakan data 32 toko

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan Setiap Kendaraan	Konvergen pada Iterasi ke-	Running Time (detik)
40	1000	K1 = Depo – Toko 1 – 30 – Depo	K1 = 38	K1 = 121	1 Jam 50 Menit 52 Detik	954	32,3371
		K2 = Depo – Toko 10 – Depo	K2 = 9	K2 = 64	0 Jam 47 Menit 3 Detik		
		K3 = Depo – Toko 25 – 18 – 19 – 7 – 4 – Depo	K3 = 96,3	K3 = 153	4 Jam 7 Menit 20 Detik		
		K4 = Depo – Toko 28 – 29 – 32 – Depo	K4 = 71	K4 = 172	3 Jam 12 Menit 9 Detik		
		K5 = Depo – Toko 6 – 5 – Depo	K5 = 17,7	K5 = 16	0 Jam 43 Menit 21 Detik		
		K6 = Depo – Toko 17 – 15 – 13 – Depo	K6 = 49,2	K6 = 124	2 Jam 24 Menit 7 Detik		
		K7 = Depo – Toko 16 – 20 – 14 – Depo	K7 = 50,7	K7 = 45	1 Jam 46 Menit 26 Detik		
		K8 = Depo – Toko 9 – 11 – 26 – 27 – 22 – 24 – 23 – Depo	K8 = 116,8	K8 = 112	4 Jam 33 Menit 2 Detik		
		K9 = Depo – Toko 21 – 12 – 3 – 31 – 8 – 2 – Depo	K9 = 99	K9 = 298	4 Jam 55 Menit 25 Detik		
2000	2000	K1 = Depo – Toko 22 – 15 – 30 – 8 – Depo	K1 = 84,2	K1 = 15	2 Jam 39 Menit 45 Detik	1971	63,0913
		K2 = Depo – Toko 32 – 28 – 29 – 12 – 11 – 14 – 13 – 9 – 18 – 32 – 24 – Depo	K2 = 181,4	K2 = 287	7 Jam 12 Menit 19 Detik		
		K3 = Depo – Toko 1 – Depo	K3 = 2,4	K3 = 115	0 Jam 51 Menit 25 Detik		
		K4 = Depo – Toko 5 – 6 – 31 – 10 – 3 – 2 – Depo	K4 = 64,8	K4 = 254	3 Jam 40 Menit 51 Detik		
		K5 = Depo – Toko 4 – 20 – 26 – 27 – 25 – 16 – 17 – 7 – Depo	K5 = 147,2	K5 = 295	6 Jam 28 Menit 24 Detik		
		K6 = Depo – Toko 19 – Depo	K6 = 23	K6 = 22	1 Jam 0 Menit 25 Detik		

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan Setiap Kendaraan	Konvergen pada Iterasi ke-	Running Time (detik)
	3000	K7 = Depo – Toko 21 – Depo	K7 = 49,8	K7 = 117	2 Jam 12 Menit 22 Detik	571	30,784
		K1 = Depo – Toko 1 – Depo	K1 = 2,4	K1 = 115	0 Jam 51 Menit 25 Detik		
		K2 = Depo – Toko 8 – 2 – Depo	K2 = 15,6	K2 = 10	0 Jam 44 Menit 28 Detik		
		K3 = Depo – Toko 6 – 3 – 5 – 7 – 32 – Depo	K3 = 60,3	K3 = 85	2 Jam 23 Menit 31 Detik		
		K4 = Depo – Toko 30 – 31 – Depo	K4 = 44,5	K4 = 127	2 Jam 9 Menit 29 Detik		
		K5 = Depo – Toko 17 – 19 – Depo	K5 = 28,7	K5 = 125	1 Jam 50 Menit 40 Detik		
		K6 = Depo – Toko 12 – Depo	K6 = 18,4	K6 = 2	0 Jam 33 Menit 8 Detik		
		K7 = Depo – Toko 16 – 4 – Depo	K7 = 35,3	K7 = 104	1 Jam 41 Menit 32 Detik		
		K8 = Depo – Toko 18 – 13 – 15 – 14 – 20 – 11 – 9 – 10 – 22 – 24 – 23 – Depo	K8 = 112,7	K8 = 194	4 Jam 30 Menit 17 Detik		
		K9 = Depo – Toko 21 – 28 – 29 – Depo	K9 = 75,3	K9 = 278	3 Jam 51 Menit 21 Detik		
60	1000	K10 = Depo – Toko 27 – 26 – 25 – Depo	K10 = 124,4	K10 = 65	4 Jam 20 Menit 25 Detik		
		K1 = Depo – Toko 10 – 8 – 4 – 14 – 26 – 27 – 2 – Depo	K1 = 121,3	K1 = 232	5 Jam 18 Menit 9 Detik		
		K2 = Depo – Toko 9 – 30 – 16 – 13 – 15 – 22 – 24 – 31 – Depo	K2 = 104,9	K2 = 221	4 Jam 32 Menit 20 Detik		
		K3 = Depo – Toko 6 – 25 – 23 – Depo	K3 = 88,9	K3 = 24	3 Jam 4 Menit 48 Detik		
		K4 = Depo – Toko 20 – 29 – 28 – 32 – Depo	K4 = 94,1	K4 = 185	3 Jam 57 Menit 17 Detik		
		K5 = Depo – Toko 3 – 12 – 7 – 1 – Depo	K5 = 23,2	K5 = 175	1 Jam 54 Menit 33 Detik		
		K6 = Depo – Toko 5 – Depo	K6 = 12,4	K6 = 11	0 Jam 29 Menit 19 Detik		
		K7 = Depo – Toko 17 – 19 – 18 – Depo	K7 = 36,6	K7 = 138	2 Jam 10 Menit 4 Detik		

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan Setiap Kendaraan	Konvergen pada Iterasi ke-	Running Time (detik)
	2000	K8 = Depo – Toko 21 – 11 – Depo	K8 = 61,3	K8 = 119	2 Jam 41 Menit 11 Detik	1941	105,6639
		K1 = Depo – Toko 2 – 1 – Depo	K1 = 5,6	K1 = 1 20	1 Jam 1 Menit 33 Detik		
		K2 = Depo – Toko 6 – 3 – 5 – 7 – 4 – 8 – 9 – Depo	K2 = 31,3	K2 = 226	2 Jam 37 Menit 2 Detik		
		K3 = Depo – Toko 11– 12 – Depo	K3 = 30,9	K3 = 4	1 Jam 7 Menit 6 Detik		
		K4 = Depo – Toko 10 – 32 – 13 – 14 – 15 – 16 – 18 – 19 – 17 – Depo	K4 = 115,5	K4 = 266	5 Jam 31 Menit 36 Detik		
		K5 = Depo – Toko 21 – 20 – 23 – 25 – 22 – 24 – Depo	K5 = 145,8	K5 = 153	5 Jam 30 Menit 52 Detik		
		K6 = Depo – Toko 27 – 26 – Depo	K6 = 99,1	K6 = 48	3 Jam 20 Menit 48 Detik		
		K7 = Depo – Toko 28 – 29 – 30 – 31 – Depo	K7 = 100,1	K7 = 288	4 Jam 43 Menit 36 Detik		
	3000	K1 = Depo – Toko 5 – 21 – 29 – 32 – Depo	K1 = 84,5	K1 = 140	3 Jam 26 Menit 12 Detik	2537	144,4684
		K2 = Depo – Toko 31 – 26 – 14 – 15 – 30 – 3 – 28 – 6 – Depo	K2 = 175,4	K2 = 405	7 Jam 29 Menit 41 Detik		
		K3 = Depo – Toko 1 – Depo	K3 = 2,4	K3 = 115	0 Jam 51 Menit 25 Detik		
		K4 = Depo – Toko 8– Depo	K4 = 11,5	K4 = 5	0 Jam 25 Menit 15 Detik		
		K5 = Depo – Toko 9 – 27 – 25 – 24 – 23 – 22 – 18 – Depo	K5 = 121,7	K5 = 96	4 Jam 38 Menit 4 Detik		
		K6 = Depo – Toko 2 – 11 – 13 – 20 – 12 – 7 – 4 10 – Depo	K6 = 59,6	K6 = 206	3 Jam 25 Menit 2 Detik		
		K7 = Depo – Toko 19 – 17 – 16 – Depo	K7 = 45,5	K7 = 138	2 Jam 22 Menit 5 Detik		

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan Setiap Kendaraan	Konvergen pada Iterasi ke-	Running Time (detik)
80	1000	K1 = Depo – Toko 3 – 1 – 2 – Depo	K1 = 16,7	K1 = 168	1 Jam 41 Menit 51 Detik	964	45,5020
		K2 = Depo – Toko 5 – 4 – 6 – 7 – 9 – Depo	K2 = 30,3	K2 = 173	2 Jam 13 Menit 52 Detik		
		K3 = Depo – Toko 12 – Depo	K3 = 18,4	K3 = 2	0 Jam 33 Menit 4 Detik		
		K4 = Depo – Toko 8 – Depo	K4 = 11,4	K4 = 5	0 Jam 25 Menit 15 Detik		
		K5 = Depo – Toko 10 – 31 – 11 – 16 – 13 – 15 – 14 – 17 – 18 – 19 – Depo	K5 = 100,4	K5 = 379	5 Jam 41 Menit 37 Detik		
		K6 = Depo – Toko 21 – 20 – 23 – 22 – 24 – 25 – Depo	K6 = 145,1	K6 = 153	5 Jam 30 Menit 4 Detik		
		K7 = Depo – Toko 27 – 26 – Depo	K7 = 99,1	K7 = 48	3 Jam 20 Menit 48 Detik		
		K8 = Depo – Toko 28 – 29 – 32 – 30 – Depo	K8 = 105,1	K8 = 178	4 Jam 23 Menit 36 Detik		
2000	2000	K1 = Depo – Toko 1 – 28 – 29 – 32 – 21 – 18 – Depo	K1 = 125,6	K1 = 417	6 Jam 16 Menit 8 Detik	1591	129,3873
		K2 = Depo – Toko 19 – 17 – Depo	K2 = 26,7	K2 = 125	1 Jam 50 Menit 40 Detik		
		K3 = Depo – Toko 3 – Depo	K3 = 12,8	K3 = 48	0 Jam 49 Menit 23 Detik		
		K4 = Depo – Toko 5 – Depo	K4 = 12,4	K4 = 11	0 Jam 29 Menit 19 Detik		
		K5 = Depo – Toko 22 – 24 – 25 – Depo	K5 = 87,8	K5 = 21	3 Jam 3 Menit 58 Detik		
		K6 = Depo – Toko 4 – Depo	K6 = 7,4	K6 = 91	0 Jam 52 Menit 32 Detik		
		K7 = Depo – Toko 9 – 2 – 20 – 27 – 26 – Depo	K7 = 113,9	K7 = 122	4 Jam 18 Menit 14 Detik		
		K8 = Depo – Toko 30 – 31 – 15 – 14 – 13 – 16 – 23 – 10 – 7 – 11 – 6 – 12 – 8 –	K8 = 131	K8 = 270	5 Jam 53 Menit 12 Detik		

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan Setiap Kendaraan	Konvergen pada Iterasi ke-	Running Time (detik)
	3000	Depo					
		K1 = Depo – Toko 2 – Depo	K1 = 4,2	K1 = 5	0 Jam 19 Menit 40 Detik	2816	169,9125
		K2 = Depo – Toko 4 – 1 – Depo	K2 = 9,2	K2 = 206	1 Jam 36 Menit 4 Detik		
		K3 = Depo – Toko 5 – 3 – 6 – 7 – 8 – 10 – Depo	K3 = 29	K3 = 143	1 Jam 58 Menit 59 Detik		
		K4 = Depo – Toko 12 – Depo	K4 = 18,4	K4 = 2	0 Jam 33 Menit 8 Detik		
		K5 = Depo – Toko 9 – 11 – 30 – 15 – 14 – 13 – 16 – 31 – Depo	K5 = 79,6	K5 = 238	3 Jam 51 Menit 48 Detik		
		K6 = Depo – Toko 20 – Depo	K6 = 20,2	K6 = 13	0 Jam 44 Menit 44 Detik		
		K7 = Depo – Toko 17 – 19 – 18 – Depo	K7 = 36,6	K7 = 138	2 Jam 10 Menit 4 Detik		
		K8 = Depo – Toko 22 – 24 – 23 – 25 – Depo	K8 = 94,5	K8 = 23	3 Jam 20 Menit 20 Detik		
		K9 = Depo – Toko 27 – 26 – Depo	K9 = 99,1	K9 = 48	3 Jam 20 Menit 48 Detik		
		K10 = Depo – Toko 21 – 29 – 28 – 32 – Depo	K10 = 87,6	K10 = 289	4 Jam 21 Menit 43 Detik		

Lampiran C. Hasil dari Percobaan modifikasi algoritma CODEQ dengan menggunakan data 32 toko

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan	Konvergen pada Iterasi ke-	Running Time (detik)
40	1000	K1 = Depo – Toko 18 – 31 – 6 – 7 – 9 – 19 – 17 – Depo	K1 = 74,9	K1 = 330	4 Jam 28 Menit 4 Detik	820	51,312
		K2 = Depo – Toko 10 – 11 – 20 – 26 – 25 – 24 – 22 – 23 – 8 – Depo	K2 = 140,1	K2 = 151	5 Jam 27 Menit 15 Detik		
		K3 = Depo – Toko 2 – 32 – 28 – 29 – 21 – 5 – 3 – 12 – Depo	K3 = 103,4	K3 = 355	5 Jam 19 Menit 49 Detik		
		K4 = Depo – Toko 1 – Depo	K4 = 2,4	K4 = 115	0 Jam 51 Menit 25 Detik		
		K5 = Depo – Toko 4 – Depo	K5 = 7,4	K5 = 91	0 Jam 52 Menit 32 Detik		
		K6 = Depo – Toko 27 – 13 – 15 – 14 – 16 – 30 – Depo	K6 = 106,4	K6 = 63	3 Jam 30 Menit 22 Detik		
	2000	K1 = Depo – Toko 30 – 31 – 14 – 15 – 26 – 27 – 25 – 23 – 22 – 24 – 16 – 8 – 5 – 6 – Depo	K1 = 194,1	K1 = 253	7 Jam 55 Menit 51 Detik		
		K2 = Depo – Toko 2 – Depo	K2 = 4,2	K2 = 5	0 Jam 19 Menit 40 Detik		
		K3 = Depo – Toko 19 – 17 – Depo	K3 = 28,7	K3 = 125	1 Jam 50 Menit 40 Detik		
		K4 = Depo – Toko 1 – Depo	K4 = 2,4	K4 = 115	0 Jam 51 Menit 25 Detik		
		K5 = Depo – Toko 3 – 10 – 11 – 13 – 18 – 20 – 7 – 4 – 9 – Depo	K5 = 58,8	K5 = 316	4 Jam 0 Menit 40 Detik		
		K6 = Depo – Toko 21 – 29 – 28 – 12 – 32 – Depo	K6 = 107,4	K6 = 291	4 Jam 54 Menit 31 Detik		
3000	K1 = Depo – Toko 9 – 10 – 4 – 30 – 18 – 2 – Depo	K1 = 53	K1 = 235	3 Jam 6 Menit 43 Detik	27723	184,9248	
	K2 = Depo – Toko 1 – Depo	K2 = 2,4	K2 = 115	0 Jam 51 Menit 25 Detik			
	K3 = Depo – Toko 21 – 29 – 28 – 12 – 7 – 6 – 3 – 5 – 8 – Depo	K3 = 90,6	K3 = 359	4 Jam 57 Menit 8 Detik			
	K4 = Depo – Toko 32 – Depo	K4 = 48,2	K4 = 11	1 Jam 24 Menit 0 Detik			
	K5 = Depo – Toko 17 – 19 – Depo	K5 = 28,7	K5 = 125	1 Jam 50 Menit 40 Detik			
	K6 = Depo – Toko 31 – 11 – 20 – 14 – 15 – 26 – 27 – 25 – 24 – 22 – 23 – 16 – 13 – Depo	K6 = 186,8	K6 = 260	7 Jam 32 Menit 41 Detik			
60	1000	K1 = Depo – Toko 21 – 29 – 28 – 32 – 6 – 7 – 4 – 5 – Depo	K1 = 99,8	K1 = 406	5 Jam 39 Menit 38 Detik	860	80,0696
		K2 = Depo – Toko 8 – 13 – 16 – 20 – 1 – 2 –	K2 = 107,9	K2 = 478	6 Jam 35 Menit 8 Detik		

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan	Konvergen pada Iterasi ke-	Running Time (detik)
	2000	11 – 9 – 3 – 10 – 18 – 17 – 19 – Depo K3 = Depo – Toko 30 – 31 – 23 – 26 – 27 – 15 – 14 – 25 – 22 – 24 – Depo K4 = Depo – Toko 12 – Depo K1 = Depo – Toko 7 – 11 – 31 – 30 – 20 – 13 – 15 – 14 – 26 – 27 – 22 – 24 – 16 – 18 – Depo K2 = Depo – Toko 9 – 1 – Depo K3 = Depo – Toko 5 – 25 – 23 – 10 – 2 – Depo K4 = Depo – Toko 19 – 17 – 9 – Depo K5 = Depo – Toko 8 – 29 – 28 – 21 – 32 – 12 – 6 – 3 – 4 – Depo	K3 = 195,8 K4 = 18,4 K1 = 157,3	K3 = 219 K4 = 2 K1 = 270	7 Jam 24 Menit 37 Detik 0 Jam 33 Menit 8 Detik 6 Jam 49 Menit 19 Detik	1929	182,0502
	3000	K1 = Depo – Toko 32 – 29 – 28 – 21 – 3 – 6 – 12 – 7 – 5 – 4 – Depo K2 = Depo – Toko 1 – 11 – 20 – 31 – 15 – 14 – 16 – 13 – 26 – 18 – 30 – 17 – 19 – 2 – Depo K3 = Depo – Toko 8 – Depo K4 = Depo – Toko 9 – 10 – 25 – 27 – 23 – 22 – 24 – Depo	K1 = 108,3 K2 = 160,7 K3 = 11,4 K4 = 115,7	K1 = 456 K2 = 497 K3 = 5 K4 = 147	6 Jam 10 Menit 31 Detik 8 Jam 16 Menit 57 Detik 0 Jam 25 Menit 15 Detik 4 Jam 37 Menit 39 Detik	2846	297,6055
80	1000	K1 = Depo – Toko 7 – 32 – 29 – 28 – 21 – 11 – 13 – 18 – 9 – 10 – Depo K2 = Depo – Toko 20 – Depo K3 = Depo – Toko 1 – Depo K4 = Depo – Toko 5 – 12 – 6 – 4 – 2 – 8 – Depo	K1 = 128,2 K2 = 20,2 K3 = 2,4 K4 = 43,1	K1 = 501 K2 = 13 K3 = 115 K4 = 119	7 Jam 4 Menit 31 Detik 0 Jam 44 Menit 44 Detik 0 Jam 51 Menit 25 Detik 2 Jam 17 Menit 30 Detik	933	104,5973

Digital Repository Universitas Jember

Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan	Konvergen pada Iterasi ke-	Running Time (detik)
		K5 = Depo – Toko 30 – 31 – 14 – 15 – 26 – 27 – 25 – 24 – 22 – 23 – 16 – Depo	K5 = 184,8	K5 = 232	7 Jam 15 Menit 53 Detik		
	2000	K6 = Depo – Toko 17 – 19 - Depo	K6 = 28,7	K6 = 125	1 Jam 50 Menit 40 Detik	1297	203,0478
		K1 = Depo – Toko 19 – 7 – 9 – Depo	K1 = 31,3	K1 = 181	2 Jam 16 Menit 53 Detik		
		K2 = Depo – Toko 4 – 1 – 7 – Depo	K2 = 14,6	K2 = 216	1 Jam 51 Menit 50 Detik		
		K3 = Depo – Toko 18 – 31 – 20 – 13 – 16 – 23 – 24 – 22 – 10 – Depo	K3 = 90,1	K3 = 249	4 Jam 3 Menit 8 Detik		
		K4 = Depo – Toko 21 – 29 – 28 – 32 – 3 – 6 – 5 – Depo	K4 = 96,1	K4 = 249	4 Jam 3 Menit 8 Detik		
		K5 = Depo – Toko 2 – Depo	K5 = 18,4	K5 = 2	0 Jam 33 Menit 8 Detik		
		K6 = Depo – Toko 2 – 11 – 30 – 15 – 14 – 26 – 27 – 25 – 8 – Depo	K6 = 158,1	K6 = 104	5 Jam 34 Menit 5 Detik		
	3000	K1 = Depo – Toko 4 – 11 – 26 – 27 – 25 – 23 – 22 – 24 – 10 – Depo	K1 = 149,2	K1 = 226	6 Jam 26 Menit 55 Detik		
		K2 = Depo – Toko 18 – 20 – 13 – 16 – 14 – 15 – 30 – 31 – 9 – 8 – 2 – Depo	K2 = 85,3	K2 = 272	4 Jam 31 Menit 36 Detik		
		K3 = Depo – Toko 17 – 19 – 7 – Depo	K3 = 33,3	K3 = 135	2 Jam 5 Menit 4 Detik		
		K4 = Depo – Toko 1 – 28 – 29 – 32 – 21 – 5 – 3 – 6 – 12 – Depo	K4 = 118,6	K4 = 470	6 Jam 18 Menit 38 Detik		

Lampiran D. Hasil dari percobaan algoritma CODEQ dan modifikasi algoritma CODEQ dengan 57 toko

Algoritma	Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan	Konvergen	Running Time (detik)
CODEQ	100	5000	K1 = Depo – Toko 4 – Depo	K1 = 4,2	K1 = 25	0 Jam 20 Menit 20 Detik	4895	438,661
			K2 = Depo – Toko 1– Depo	K2 = 2,4	K2 = 600	3 Jam 28 Menit 0 Detik		
			K3 = Depo – Toko 7 – Depo	K3 = 7,4	K3 = 600	3 Jam 40 Menit 11 Detik		
			K4 = Depo – Toko 3 – 2 – 20 – 11 – 5 – 10 – Depo	K4 = 41,2	K4 = 484	4 Jam 10 Menit 12 Detik		
			K5 = Depo – Toko 15 – Depo	K5 = 11,4	K5 = 27	0 Jam 38 Menit 44 Detik		
			K6 = Depo – Toko 8 – 9 – Depo	K6 = 14,9	K6 = 541	3 Jam 41 Menit 36 Detik		
			K7 = Depo – Toko 6 – 14 – Depo	K7 = 23	K7 = 154	1 Jam 43 Menit 36 Detik		
			K8 = Depo – Toko 12 – 18 – 16 – 13 – 19 – Depo	K8 = 42,7	K8 = 161	2 Jam 35 Menit 30 Detik		
			K9 = Depo – Toko 17 – Depo	K9 = 9	K9 = 271	1 Jam 56 Menit 3 Detik		
			K10 = Depo – Toko 21 – 22 – 23 – 25 – 24 – 27 – 28 – Depo	K10 = 64,5	K10 = 140	3 Jam 7 Menit 59 Detik		
			K11 = Depo – Toko 26 – 29 – Depo	K11 = 28,4	K11 = 157	1 Jam 47 Menit 4 Detik		
			K12 = Depo – Toko 32– Depo	K12 = 94	K12 = 600	7 Jam 5 Menit 36 Detik		
			K13 = Depo – Toko 31 – 30 – 33 – Depo	K13 = 96,6	K13 = 189	4 Jam 22 Menit 49 Detik		
			K14 = Depo – Toko 38 – 34 – 36 – 35 – 37 – 39 – 40 – Depo	K14 = 151,3	K14 = 164	6 Jam 1 Menit 25 Detik		
			K15 = Depo – Toko 46– Depo	K15 = 48,2	K15 = 11	1 Jam 40 Menit 4 Detik		
			K16 = Depo – Toko 43 – 42 – 47 – 41 – Depo	K16 = 76,6	K16 = 170	3 Jam 23 Menit 43 Detik		

Algoritma	Jumlah Populasi	Jumlah Iterasi Maksimal	Rute Kendaraan Ke-	Total Jarak (km)	Total Kapasitas Kendaraan (pack)	Total Waktu Perjalanan	Konvergen	Running Time (detik)
Modifikasi CODEQ	100	5000	K17 = Depo – Toko 48 – 45 – 44 – 49 – 50 – 52 – 55 – 51 – Depo	K17 = 122	K17 = 167	4 Jam 50 Menit 45 Detik	4444	832,9247
			K18 = Depo – Toko 53 – 54 – 57 – 56 – Depo	K18 = 79,6	K18 = 237	3 Jam 43 Menit 7 Detik		
			K1 = Depo – Toko 6 – 10 – 46 – 20 – 51 – 31 – 55 – 57 – 56 – 52 – Depo	K1 = 117,7	K1 = 366	6 Jam 8 Menit 50 Detik		
			K2 = Depo – Toko 40 – 54 – 5 – Depo	K2 = 95,9	K2 = 137	3 Jam 43 Menit 40 Detik		
			K3 = Depo – Toko 8 – 13 – 12 – 48 – 49 – Depo	K3 = 30,8	K3 = 565	4 Jam 40 Menit 11 Detik		
			K4 = Depo – Toko 3 – 2 – 20 – 11 – 5 – 10 – Depo	K4 = 41,2	K4 = 484	4 Jam 25 Menit 40 Detik		
			K5 = Depo – Toko 2 – Depo	K5 = 2,4	K5 = 254	1 Jam 32 Menit 44 Detik		
			K6 = Depo – Toko 32 – Depo	K6 = 94	K6 = 600	7 Jam 5 Menit 36 Detik		
			K7 = Depo – Toko 33 – 30 – 47 – 43 – 42 – 41 – 11 – Depo	K7 = 116,6	K7 = 346	6 Jam 1 Menit 1 Detik		
			K8 = Depo – Toko 27 – 44 – 45 – 18 – 50 – 26 – 28 – Depo	K8 = 77,3	K8 = 324	4 Jam 38 Menit 14 Detik		
			K9 = Depo – Toko 7 – Depo	K9 = 7,4	K9 = 600	3 Jam 40 Menit 11 Detik		
			K10 = Depo – Toko 9 – 14 – 15 – Depo	K10 = 22,8	K10 = 83	1 Jam 20 Menit 34 Detik		
			K11 = Depo – Toko 17 – Depo	K11 = 9	K11 = 271	1 Jam 56 Menit 3 Detik		
			K12 = Depo – Toko 1 – Depo	K12 = 2,4	K12 = 600	3 Jam 28 Menit 0 Detik		
K13 = Depo – Toko 4 – Depo	K13 = 4,2	K13 = 25	0 Jam 20 Menit 20 Detik					
K14 = Depo – Toko 3 – 19 – 25 – 23 – 39 – 37 – 29 – 16 – Depo	K14 = 120	K14 = 315	5 Jam 18 Menit 18 Detik					