



**PENERAPAN ALGORITMA *IMPROVED ANT COLONY SYSTEM* DAN
HYBRID ANT COLONY SYSTEM PADA *CAPACITATED VEHICLE*
*ROUTING PROBLEM***

SKRIPSI

Oleh:

**Riska Ayu Oktaviana
NIM 131810101045**

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



**PENERAPAN ALGORITMA *IMPROVED ANT COLONY SYSTEM* DAN
HYBRID ANT COLONY SYSTEM PADA *CAPACITATED VEHICLE*
*ROUTING PROBLEM***

SKRIPSI

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

Oleh:

Riska Ayu Oktaviana
NIM 131810101045

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PERSEMBAHAN

Puji syukur kehadiran Allah SWT atas segala limpahan rahmat dan karunia-Nya serta sholawat salam kepada Nabi Muhammad SAW sehingga saya dapat menyelesaikan skripsi ini dengan lancar. Skripsi ini saya persembahkan untuk:

1. Ibu Puji Rahayu, Bapak Mulyadi, dan adik Vito Dwi Prasetyo terima kasih banyak atas kasih sayang, dukungan dan doa yang diberikan kepada saya.
2. Semua guru-guru sejak taman kanak-kanak hingga Perguruan Tinggi yang terhormat, yang telah memberikan ilmu dan membimbing dengan penuh kesabaran.
3. Sahabatku Bunga, Yulia dan Iris (Maman) yang telah memotivasi dan memberikan persahabatan yang indah.
4. Teman-teman seperjuangan 2013 (Atlas) yang telah menemani selama kurang lebih 4 tahun ini.
5. Almamater Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

“Ambil waktu untuk merencanakan tetapi jika tiba waktunya untuk bertindak berhenti berpikir dan maju terus.”

(Merry Riana^{*)})



* Twitter Merry Riana@MerryRiana

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Riska Ayu Oktaviana

NIM : 131810101045

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, Juni 2017

Yang menyatakan,

Riska Ayu Oktaviana

NIM 131810101045

SKRIPSI

**PENERAPAN ALGORITMA *IMPROVED ANT COLONY SYSTEM* DAN
HYBRID ANT COLONY SYSTEM PADA *CAPACITATED VEHICLE*
*ROUTING PROBLEM***

Oleh

Riska Ayu Oktaviana
NIM 131810101045

Pembimbing:

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

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

PENGESAHAN

Skripsi berjudul “Penerapan Algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

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

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

Anggota II,

Anggota III,

Dr. Firdaus Ubaidillah, S.Si., M.Si.
NIP. 197006061998031003

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

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Penerapan Algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*; Riska Ayu Oktaviana, 131810101045; 2017; 63 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Capacitated Vehicle Routing Problem (CVRP) merupakan bentuk paling dasar dari *Vehicle Routing Problem*. Permasalahan optimasi ini yaitu untuk menemukan rute minimal pendistribusian barang dengan sejumlah kendaraan yang berkapasitas tertentu. Dalam CVRP ini, setiap pelanggan hanya dapat dilayani oleh satu kendaraan, total permintaan dari pelanggan tidak boleh melebihi dari kapasitas kendaraan yang telah ditentukan, serta setiap kendaraan tersebut melayani sejumlah pelanggan dengan memulai rutenya dari depot kemudian kembali ke depot semula.

Data yang digunakan dalam penelitian ini adalah data sekunder dari skripsi Putri (2014) pada PT. Sumber Bening Lestari. Dalam skripsi tersebut dijelaskan bahwa data yang digunakan yaitu data jarak yang meliputi depot dan 22 pelanggan, serta data permintaan dari 22 pelanggan tersebut. Kendaraan yang digunakan yaitu berjumlah 4 dengan masing-masing kendaraan berkapasitas 210 karton.

Algoritma yang digunakan untuk menyelesaikan permasalahan ini yaitu algoritma *Improved Ant Colony System* (IACS) dan *Hybrid Ant Colony System* (HACS) juga dengan bantuan program MATLAB untuk meminimalkan rute jarak tempuh dari kendaraan. Berdasarkan penelitian yang telah dilakukan, menunjukkan bahwa algoritma IACS memperoleh total jarak tempuh yang lebih minimum dibandingkan dengan algoritma HACS. Hal tersebut menunjukkan bahwa pada penelitian ini, algoritma IACS lebih baik daripada algoritma HACS.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan kuasa-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Penerapan Algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*”. Penulisan tugas akhir ini disusun guna memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Pada kesempatan ini, dengan segala hormat penulis mengucapkan terima kasih atas bantuan dan bimbingan dalam penyusunan skripsi ini, kepada:

1. Bapak M. Ziaul Arif, S.Si., M.Sc. selaku dosen pembimbing utama dan Bapak Ahmad Kamsyakawuni, S.Si., M.Kom. selaku dosen pembimbing anggota.
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.
3. Seluruh dosen dan karyawan Jurusan Matematika FMIPA Universitas Jember.
4. Keluarga yang telah memberikan semangat dan doa yang tulus penuh kasih sayangnya.
5. Teman-teman dan semua pihak yang telah membantu terselesaikannya skripsi ini.

Penulis menyadari bahwa penulisan ini masih terdapat kekurangan. Oleh karena itu, penulis mengharap kritik dan saran demi kesempurnaan skripsi ini, semoga bermanfaat.

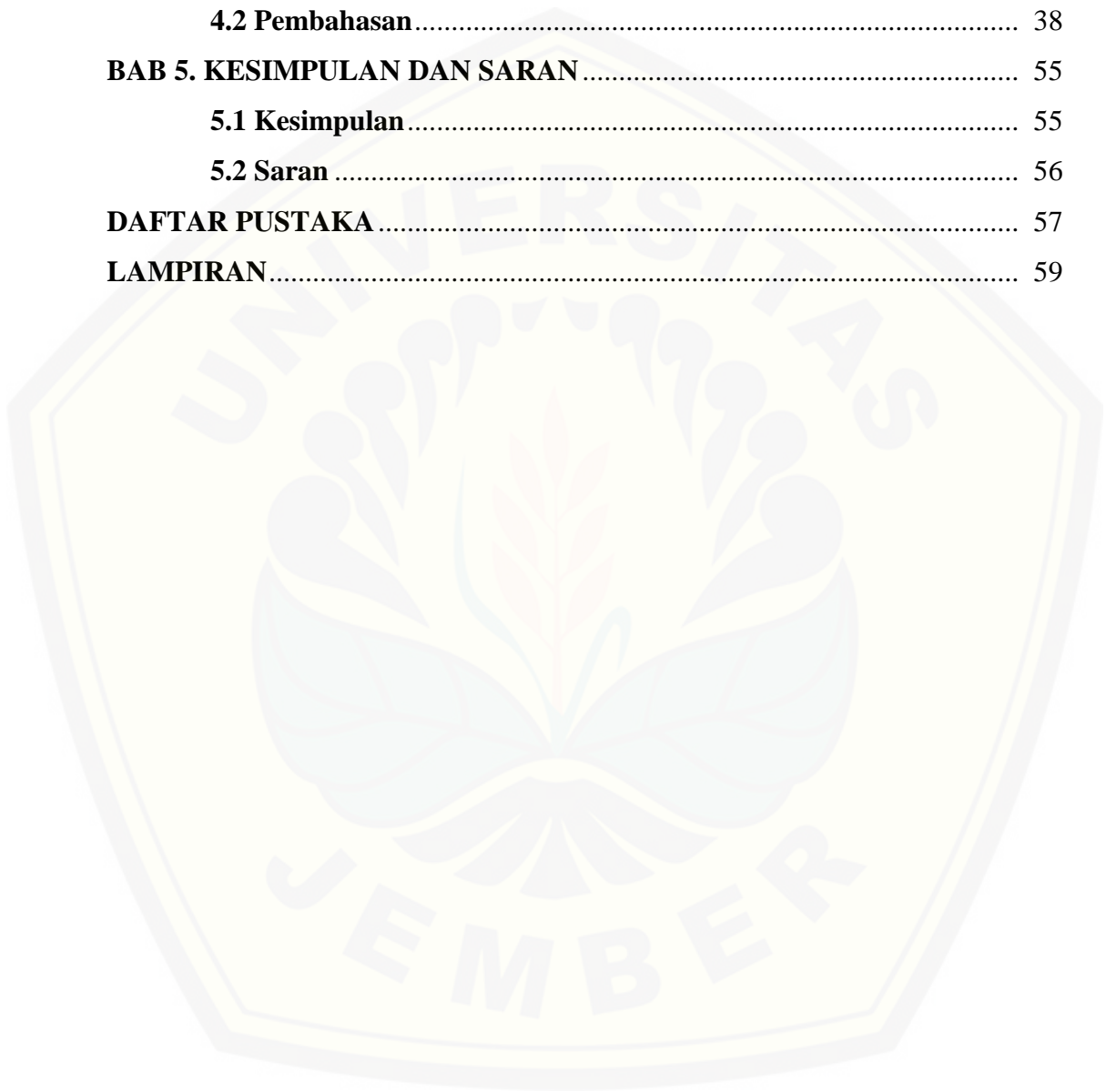
Jember, Juni 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Teori Graf	4
2.2 <i>Vehicle Routing Problem (VRP)</i>	5
2.3 <i>Capacited Vehicle Routing Problem (CVRP)</i>	6
2.4 Algoritma <i>Ant Colony System (ACS)</i>	7
2.4.1 Algoritma <i>Improved Ant Colony System (IACS)</i>	8
2.4.2 Algoritma <i>Hybrid Ant Colony System (HACS)</i>	12
BAB 3. METODE PENELITIAN	15
3.1 Data Penelitian	15
3.2 Langkah-langkah Penelitian	15

BAB 4. HASIL DAN PEMBAHASAN	17
4.1 Hasil	17
4.1.1 Langkah Perhitungan	17
4.1.2 CVRP dengan Program MATLAB.....	35
4.2 Pembahasan	38
BAB 5. KESIMPULAN DAN SARAN	55
5.1 Kesimpulan	55
5.2 Saran	56
DAFTAR PUSTAKA	57
LAMPIRAN	59

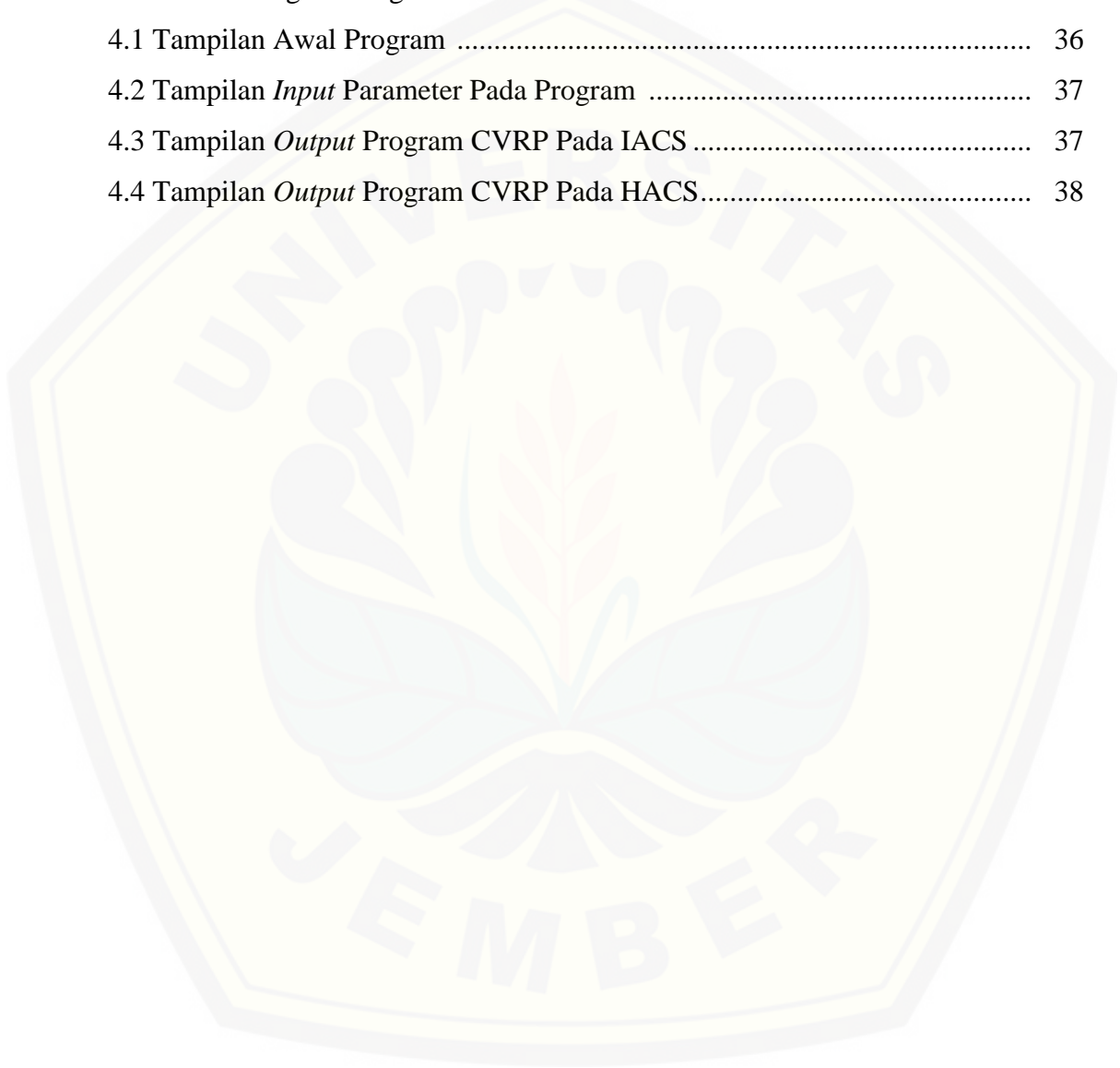


DAFTAR TABEL

	Halaman
4.1 Jarak Pelanggan.....	17
4.2 Permintaan Pelanggan	18
4.3 Pembaruan Feromon Lokal IACS Pertama	23
4.4 Pembaruan Feromon Lokal IACS Kedua	25
4.5 Pembaruan Feromon Global Solusi Pertama	28
4.6 Pembaruan Feromon Global Solusi Kedua	28
4.7 Pembaruan Feromon Rute dari L_{lokal}	34
4.8 Hasil Percobaan dengan Intensitas Feromon (τ_{ij}) = 0,01	39
4.9 Hasil Percobaan dengan Intensitas Feromon (τ_{ij}) = 0,05	40
4.10 Hasil Percobaan dengan Intensitas Feromon (τ_{ij}) = 0,1	40
4.11 Hasil Percobaan dengan Pengendali Visibilitas (β) = 1	41
4.12 Hasil Percobaan dengan Pengendali Visibilitas (β) = 4	42
4.13 Hasil Percobaan dengan Pengendali Visibilitas (β) = 5	43
4.14 Hasil Percobaan dengan Penguapan Feromon (ρ) = 0,1	44
4.15 Hasil Percobaan dengan Penguapan Feromon (ρ) = 0,5	44
4.16 Hasil Percobaan dengan Penguapan Feromon (ρ) = 0,9	45
4.17 Hasil Percobaan dengan Pengendali Nilai <i>Saving</i> (γ) = 0,1	46
4.18 Hasil Percobaan dengan Pengendali Nilai <i>Saving</i> (γ) = 0,5	47
4.19 Hasil Percobaan dengan Pengendali Nilai <i>Saving</i> (γ) = 0,9	48
4.20 Hasil Percobaan dengan Pengendali Feromon (α) = 0,1	49
4.21 Hasil Percobaan dengan Pengendali Feromon (α) = 0,5	49
4.22 Hasil Percobaan dengan Pengendali Feromon (α) = 1	50
4.23 Hasil Percobaan dengan Jumlah Semut (m) = 5	51
4.24 Hasil Percobaan dengan Jumlah Semut (m) = 14	52
4.25 Hasil Percobaan dengan Jumlah Semut (m) = 23	52
4.26 Rute Pendistribusian IACS	53
4.27 Rute Pendistribusian HACS	54

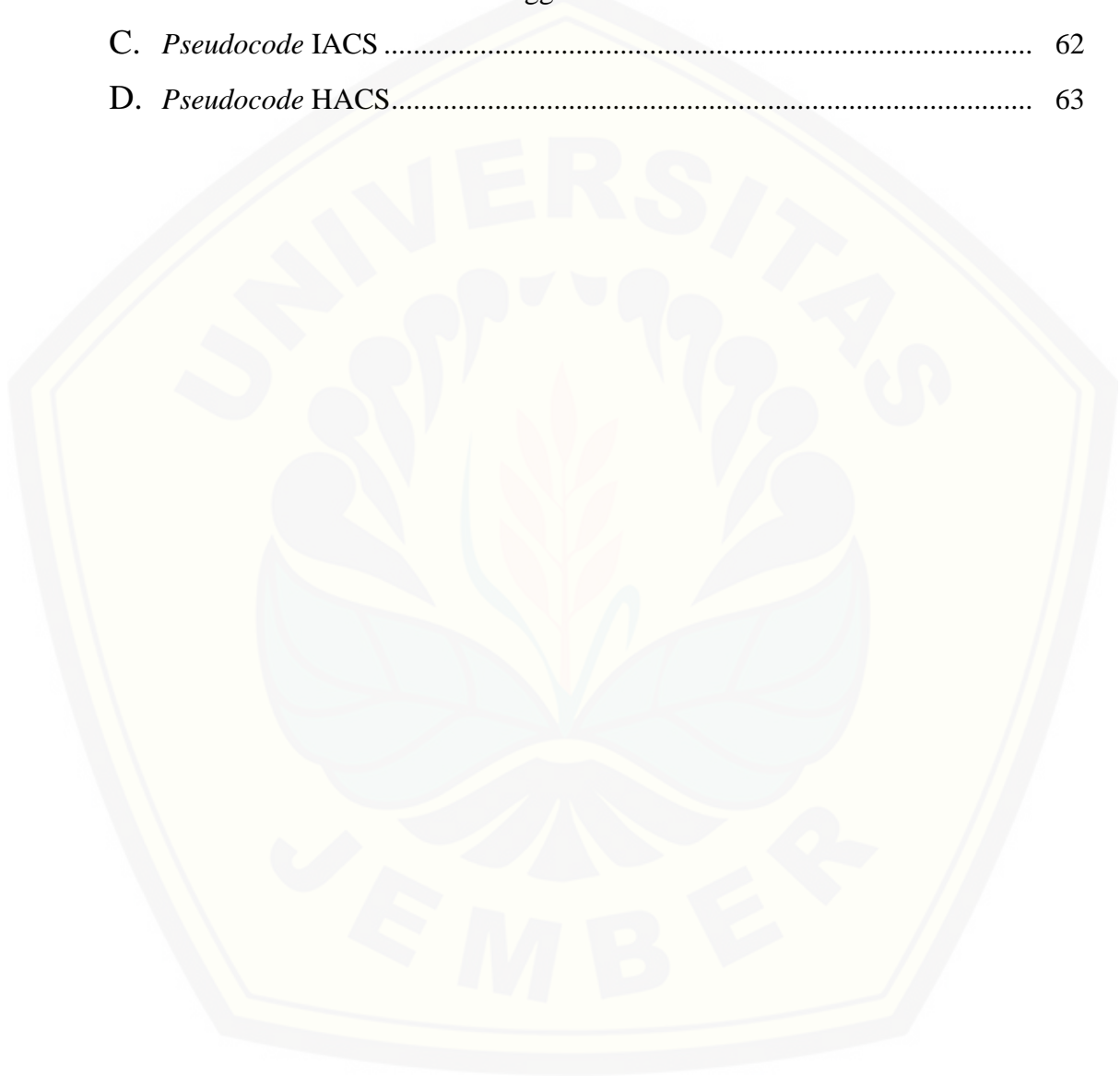
DAFTAR GAMBAR

	Halaman
2.1 Representasi Graf	4
3.1 Skema Langkah-langkah Penelitian	16
4.1 Tampilan Awal Program	36
4.2 Tampilan <i>Input</i> Parameter Pada Program	37
4.3 Tampilan <i>Output</i> Program CVRP Pada IACS	37
4.4 Tampilan <i>Output</i> Program CVRP Pada HACS.....	38



DAFTAR LAMPIRAN

	Halaman
A. Data Jarak 22 Pelanggan di Setiap Kecamatan Kabupaten Jember	59
B. Data Jumlah Permintaan Pelanggan.....	60
C. <i>Pseudocode</i> IACS	62
D. <i>Pseudocode</i> HACS.....	63



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Industri merupakan suatu usaha dalam mengolah bahan atau barang mentah menjadi barang siap pakai, guna menambah nilai jual dari barang tersebut. Dalam dunia industri terdapat permasalahan distribusi. Distribusi yaitu proses pengiriman barang dari produsen (depot) ke konsumen. Permasalahan distribusi dianggap penting karena menyangkut keuntungan yang diperoleh produsen dan cepat lambatnya barang sampai ke tangan konsumen. Jika barang lebih cepat sampai ke tangan konsumen, maka tingkat kepuasan konsumen akan meningkat. Untuk menangani permasalahan tersebut, salah satu solusi yang dapat digunakan yaitu dengan meminimalkan jarak tempuh kendaraan dari produsen ke konsumen.

Vehicle Routing Problem (VRP) merupakan jenis permasalahan dalam distribusi untuk menemukan rute dengan jarak minimum. Dalam VRP, kendaraan berangkat dari suatu depot untuk melayani konsumen dan kembali lagi ke depot asal. VRP mempunyai banyak varian tergantung dari karakteristik permasalahannya. *Capacitated Vehicle Routing Problem* (CVRP) adalah salah satu varian dari VRP dengan kendala setiap kendaraan memiliki kapasitas terbatas. Tujuan dari CVRP yaitu meminimalisasi jumlah kendaraan yang digunakan dalam proses distribusi, jarak tempuh kendaraan dengan batasan jumlah muatan angkut tidak melebihi kapasitas kendaraan yang digunakan.

Permasalahan CVRP pernah diselesaikan oleh Gunawan *et al.* (2012) dengan menggunakan algoritma *Brute-Force*, *Nearest Neighbour Insertion* (NNI) dan *Ant Colony Optimization* (ACO). Dari ketiga algoritma tersebut disimpulkan bahwa algoritma ACO pada penerapan *Ant Colony System* (ACS) merupakan algoritma dengan penyelesaian terbaik, karena mampu menyelesaikan CVRP dengan skala besar dan mampu menghasilkan solusi mendekati optimal. Namun menurut Xiang-pei *et al.* (2006), algoritma ACS mempunyai kelemahan yaitu solusi yang dihasilkan dapat terjebak dalam optimum lokal dan kurang efisien, sehingga untuk mengatasi hal tersebut dilakukan perbaikan pada algoritma ACS

yang disebut *Hybrid Ant Colony System* (HACS). HACS disajikan dengan memperkenalkan perbaikan pada aturan pengkonstruksian rute dan penyesuaian feromon, serta penggabungan algoritma ACS dengan algoritma *Saving* dan *Interchange*. Pada penelitian tersebut, algoritma HACS diterapkan pada permasalahan *Vehicle Routing Problem with Time Window*, serta disimpulkan bahwa algoritma HACS merupakan cara yang efektif dan efisien untuk mengatasi masalah optimasi kombinatorial. Selain itu, algoritma ACS juga telah dikembangkan oleh Chen & Ting pada tahun 2006. Algoritma ACS yang telah mengalami pengembangan ini dikenal sebagai *Improved Ant Colony System* (IACS). IACS merupakan algoritma ACS yang mengalami perbaikan pada aturan pengkonstruksian rute dan pembaruan feromon, serta penggabungan algoritma ACS dengan algoritma *Saving* dan *Insertion*. Pada penelitian tersebut, algoritma IACS diterapkan pada permasalahan VRP dan disimpulkan bahwa algoritma IACS dapat menemukan solusi yang baik.

Mengacu pada permasalahan di atas, penulis tertarik meneliti lebih lanjut tentang CVRP. Dalam penyelesaian kasus ini, penulis menerapkan algoritma IACS dan HACS. Hasil penerapan dari kedua algoritma ke dalam permasalahan akan diuji untuk mengetahui algoritma yang lebih unggul berdasarkan total jarak tempuh.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini adalah:

- a. Bagaimana menerapkan algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*?
- b. Bagaimana perbandingan hasil penerapan algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*?

1.3 Batasan Masalah

Agar tidak memperluas permasalahan, peneliti memberikan asumsi produk yang didistribusikan hanya satu macam (ukuran dan berat sama).

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

- a. Menerapkan algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*.
- b. Mengetahui hasil perbandingan penerapan algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah:

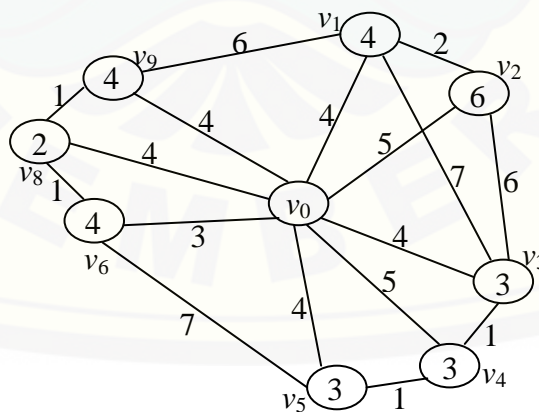
- a. Mendapat solusi yang optimal.
- b. Dapat dijadikan referensi dalam penelitian selanjutnya yang berhubungan dengan algoritma *Ant Colony System*, *Improved Ant Colony System*, *Hybrid Ant Colony System* dan *Capacitated Vehicle Routing Problem*.

BAB 2. TINJAUAN PUSTAKA

2.1 Teori Graf

Menurut Munir (2005), teori graf merupakan pokok bahasan yang memiliki banyak terapan. Graf digunakan untuk merepresentasikan objek dan hubungan antar objek tersebut. Secara sederhana, graf didefinisikan sebagai kumpulan titik yang dihubungkan oleh garis. Graf G didefinisikan sebagai pasangan himpunan $G=(V,E)$ dengan V himpunan tidak kosong dari simpul (*vertex* atau *node*) dan E (*edge*) himpunan sisi yang menghubungkan sepasang simpul.

Terdapat beberapa jenis graf, salah satunya yaitu graf berbobot. Graf berbobot merupakan graf yang sisinya mempunyai bobot tertentu. Bobot pada sisi graf dapat menyatakan jarak, biaya perjalanan, waktu tempuh, dan sebagainya. Gambar 2.1 merupakan contoh graf untuk solusi CVRP. Graf tersebut terdiri dari 10 simpul yang terhubung dan mempunyai bobot. Pada graf tersebut, v_0 menyatakan depot. Sedangkan simpul yang lain menyatakan pelanggan dan angka yang berada pada simpul melambangkan jumlah permintaan pelanggan. Angka pada masing-masing sisi menyatakan jarak yang harus ditempuh untuk sampai pada lokasi yang bersangkutan.



Gambar 2.1 Representasi Graf
(Sumber: Gunawan *et al.*, 2012).

2.2 *Vehicle Routing Problem (VRP)*

Vehicle Routing Problem merupakan suatu permasalahan dalam menentukan rute tempuh kendaraan selama proses distribusi barang untuk melayani sejumlah pelanggan dari depot pusat. Permasalahan VRP memiliki tujuan untuk melayani pelanggan sesuai dengan permintaan dengan meminimalkan jumlah serta biaya angkut kendaraan yang diawali dari depot (pusat) dan kembali lagi ke depot asal (Hasanah & Wahyuningsih, 2013).

Menurut Tarigan (2008), terdapat beberapa faktor yang mempengaruhi VRP sehingga muncul beberapa variasi dalam VRP, antara lain yaitu:

- a. *Capacitated Vehicle Routing Problem (CVRP)* yaitu terdapat beberapa kendaraan dari sebuah depot yang mempunyai kapasitas tertentu harus melayani sejumlah permintaan pelanggan yang telah diketahui jumlahnya dengan biaya minimum. CVRP bertujuan meminimalisasi jarak pada rute perjalanan dengan syarat total permintaan barang tiap rute tidak melebihi kapasitas kendaraan yang melewati rute tersebut.
- b. *Vehicle Routing Problem with Time Windows (VRPTW)* yaitu sejumlah kendaraan harus melayani setiap pelanggan dalam selang waktu yang tertentu. Jika kendaraan sampai sebelum atau sesudah interval waktu yang telah ditentukan, maka akan menimbulkan kerugian baik bagi konsumen maupun penyuplai.
- c. *Multiple Depot Vehicle Routing Problem (MDVRP)* yaitu perusahaan memiliki lebih dari satu depot dengan pelanggan dan depot-depot tidak terkumpul secara teratur, sehingga diperlukan pengaturan para pelanggan ke depot-depot yang ada. Setiap kendaraan berangkat dari suatu depot untuk melayani pelanggan dan kembali lagi ke depot asal.
- d. *Vehicle Routing Problem with Pick-Up and Delivering (VRPPD)* yaitu permasalahan dengan kemungkinan pelanggan mengembalikan barang yang sudah diantarkan, sehingga dimungkinkan pelanggan dapat menerima dan mengembalikan barang secara bersamaan.
- e. *Stochastic Vehicle Routing Problem (SVRP)* yaitu permasalahan pada saat terdapat faktor sampingan acak seperti jumlah pelanggan, jumlah permintaan,

waktu pelayanan. Solusi dari permasalahan ini dibagi menjadi dua tahap dengan solusi pada tahap pertama ditentukan sebelum variabel acak diketahui dan pada tahap kedua dilakukan pengoreksian dengan syarat variabel acak sudah diketahui.

- f. *Periodic Vehicle Routing Problem (PVRP)* yaitu perluasan rentang perencanaan pengiriman menjadi M hari, dari semula hanya dalam rentang sehari.

2.3 Capacitated Vehicle Routing Problem (CVRP)

CVRP merupakan bentuk paling dasar dari VRP yang diperkenalkan melalui makalah yang berjudul *The Truck Dispatching Problem*. CVRP adalah masalah optimasi untuk menemukan rute dengan sejumlah kendaraan berkapasitas tertentu (homogen) yang melayani sejumlah permintaan pelanggan, serta kuantitas permintaannya telah diketahui sebelum proses distribusi berlangsung. Kendaraan berangkat dari depot untuk melakukan pengiriman kemudian kembali lagi ke depot asal. Jarak antara dua lokasi adalah simetris yang berarti jarak dari lokasi A ke B sama dengan jarak dari B ke A (Gunawan *et al.*, 2012).

CVRP didefinisikan sebagai graf $G = (V, E)$ dengan $V = \{v_0, v_1, \dots, v_n\}$ dan himpunan sisi E . Depot dinyatakan sebagai v_0 dengan sejumlah kendaraan yang berkapasitas homogen Q . Setiap pelanggan dengan $i > 0$ memiliki sejumlah permintaan non negatif q_i . Sisi (i, j) memiliki jarak tempuh c_{ij} dan diasumsikan simetrik dengan $c_{ij} = c_{ji}$ dan $c_{ii} = 0$.

Variabel keputusan untuk meminimumkan total jarak tempuh kendaraan adalah

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

Fungsi tujuan untuk meminimumkan total jarak tempuh. Perjalanan dari depot kemudian mengunjungi semua pelanggan dan kembali ke depot asal,

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

dengan kendala berikut:

- a. Setiap pelanggan hanya dapat dilayani oleh satu kendaraan

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

- b. Total permintaan tidak boleh melebihi kapasitas kendaraan

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

- c. Setiap kendaraan berangkat dari depot ($N = 0$) dan setelah melayani pelanggan maka kendaraan akan meninggalkan tempat kemudian kendaraan kembali ke depot asal ($N + 1$)

$$\sum_{j=0}^{N+1} x_{0j}^k = 1 ; k = 1, 2, \dots, K \quad (2.4)$$

$$\sum_{i=0}^{N+1} x_{ih}^k - \sum_{j=0}^{N+1} x_{hj}^k = 0 ; h = 1, 2, \dots, N ; k = 1, 2, \dots, K \quad (2.5)$$

$$\sum_{i=0}^{N+1} x_{i,N+1}^k = 1 ; k = 1, 2, \dots, K \quad (2.6)$$

- d. Variabel keputusan *integer biner*.

$$x_{ij}^k \in \{0,1\} ; i, j = 0, 1, \dots, N + 1 ; k = 1, 2, \dots, K$$

dimana,

K = jumlah kendaraan

N = jumlah pelanggan

c_{ij} = jarak pelanggan i ke pelanggan j

q_j = jumlah permintaan pelanggan j

Q = kapasitas maksimum kendaraan (Kallehauge *et al.*, 2001).

2.4 Algoritma *Ant Colony System* (ACS)

Algoritma ACS merupakan kumpulan algoritma yang terinspirasi dari perilaku serangga berupa koloni semut. Semut selama proses pencarian jalur menuju sumber makanan akan mengeluarkan substansi aromatik yang disebut feromon. Proses peninggalan feromon ini dikenal dengan *stigmergy*. Substansi aromatik ini digunakan untuk memberikan tanda pada jalur yang dilalui sehingga semut dapat kembali ke sarangnya. Semakin sering suatu jalur dilewati maka intensitas feromon suatu jalur akan bertambah sedangkan jalur yang jarang dilewati kadar feromonnya akan mengalami *evaporasi* (penguapan). Proses pembentukan jalur menuju sumber makanan inilah yang akan menentukan rute terbaik. Algoritma ACS ini digunakan untuk menemukan rute terpendek. Terdapat

sejumlah agen yang digunakan untuk membentuk rute. Secara acak, setiap agen ditugaskan untuk membentuk rute dari depot hingga semua konsumen dikunjungi dan menghasilkan solusi. Hal tersebut dilakukan secara iteratif dengan menambahkan konsumen baru hingga semua konsumen dikunjungi (Afrianita, 2011).

Menurut Dorigo & Gambardela (1997), cara kerja ACS secara informal yaitu sejumlah m semut ditempatkan pada n titik ($m \leq n$) berdasarkan aturan inisialisasi. Setiap semut membangun tur dengan aturan transisi status. Selagi membangun turnya, semut juga memodifikasi feromon pada ruas-ruas yang dikunjungi dengan aturan pembaruan feromon lokal. Setelah semua semut menyelesaikan turnya, jumlah feromon dimodifikasi kembali dengan aturan pembaruan feromon global. Dalam membangun turnya, semut menggunakan informasi feromon dan informasi mengenai pemilihan ruas-ruas yang pendek.

Algoritma ACS merupakan algoritma yang mampu mendapatkan solusi yang optimal. Namun, solusi dari algoritma ACS dapat terjebak dalam lokal optimum dan kurang efisien (Xiang-pei *et al.*, 2006). Lokal optimum merupakan nilai fungsi terkecil pada beberapa solusi ketetanggaan yang dimungkinkan. Lokal optimum $f^*=f(x^*)$ dengan x^* merupakan lokal optimum dari persekitaran solusi atau $x^* \in \Omega$. Terdapat $\delta > 0$ sedemikian hingga $f^* \leq f(x)$ untuk semua x pada $\{x \in \Omega : |x-x^*| \leq \delta\}$. Global optimum merupakan nilai fungsi paling optimum dari semua solusi yang dimungkinkan. Global optimum $f^*=f(x^*)$ dengan x^* merupakan global optimum dengan $f^* \leq f(x)$ untuk semua x pada Ω (Pinter, 1995). Mengacu dari hal tersebut maka dilakukan pembaruan pada algoritma ACS (Xiang-pei *et al.*, 2006). Pembaruan algoritma ini disebut dengan algoritma IACS dan HACS.

2.4.1 Algoritma *Improved Ant Colony System* (IACS)

Chen & Ting (2006) mendefinisikan algoritma IACS merupakan algoritma ACS yang sudah mengalami perbaikan pada aturan mengkonstruksi rute dan aturan pembaruan feromon serta dilakukan penggabungan algoritma ACS dengan algoritma *Saving* dan *Insertion*. Berikut merupakan langkah-langkah dalam penyelesaian algoritma IACS:

a. Menentukan parameter

Parameter yang digunakan pada algoritma IACS untuk memodelkan sikap semut dalam mencapai nilai optimal yaitu:

- 1) Intensitas feromon (τ_{ij})
- 2) Tetapan pengendali visibilitas (β)
- 3) Tetapan penguapan feromon pada feromon global (γ)
- 4) Jumlah semut (n)
- 5) Tetapan penguapan feromon pada feromon lokal (ρ).

b. Membangkitkan inisial solusi menggunakan *Nearest Neighbor Heuristic*

Menurut Pop *et al.* (2011), bahwa algoritma ini digunakan untuk membangkitkan inisial solusi. Langkah-langkahnya yaitu:

- 1) Perhitungan dimulai dari depot dengan mencari pelanggan yang memiliki jarak terdekat dari depot sebagai lokasi pertama.
- 2) Dilanjutkan dengan mencari pelanggan dengan jarak terdekat dari pelanggan terpilih dan jumlah pengiriman tidak melebihi kapasitas kendaraan.
 - a) Jika terdapat sisa kapasitas kendaraan maka kembali ke langkah 2
 - b) Jika tidak memiliki sisa kapasitas kembali ke langkah 1
 - c) Jika tidak ada lokasi yang terpilih karena jumlah pengiriman melebihi kapasitas maka kembali ke langkah 1.

c. Menerapkan *local search (insertion)* ke dalam inisial solusi, simpan sebagai solusi pertama dengan cara:

- 1) Random dua posisi yang akan dilakukan *insertion*
- 2) Pindahkan elemen pada posisi 1 ke posisi 2
- 3) Geser elemen mulai posisi 1 + 1 hingga posisi 2.

d. Membentuk solusi berdasarkan aturan mengkonstruksi rute dan lakukan pembaruan feromon lokal pada setiap busur yang dilalui, semut = semut + 1.

1) Aturan konstruksi rute

Setiap semut k bergerak dari simpul i ke simpul j atau busur (i,j) hingga semua simpul dikunjungi. Dengan batasan jika kendala kapasitas kendaraan terpenuhi, semut akan kembali ke depot sebelum memilih simpul

berikutnya. Proses ini berlanjut sampai setiap simpul dikunjungi dan tur selesai. Aturan pemilihan rute sebagai berikut (Bauer *et al.*, 1999):

- a) Eksploitasi yaitu semut memilih rute yang mempunyai nilai fungsi feromon dan informasi heuristik yang paling tinggi diantara seluruh kandidat calon rute yang ada dengan memperdalam pencarian diruang solusi yang sama. Hal ini dilakukan untuk menghindari pencarian yang terlalu random dan jauh dari solusi yang optimal.
- b) Eksplorasi yaitu semut memilih rute dengan probabilitas pemilihan berdasarkan nilai fungsi feromon dan informasi heuristik. Proses ini dapat menghasilkan alteratif solusi yang lebih banyak karena dapat memperluas ruang solusi, sehingga mengurangi kemungkinan terjebak pada optimum lokal.

Berikut merupakan rumus untuk pemilihan rute (Chen & Ting, 2006):

Eksploitasi jika $q \leq q_0$ yaitu

$$P_{ij}^k = \max_{j \in U_k} [(\tau_{ij})(\eta_{ij})^\beta] \quad (2.7)$$

Eksplorasi jika $q > q_0$ yaitu

$$P_{ij}^k = \frac{(\tau_{ij})(\eta_{ij})^\beta}{\sum_{j \in U_k} (\tau_{ij})(\eta_{ij})^\beta} \quad (2.8)$$

$$\eta_{ij} = d_{0i} + d_{0j} - d_{ij} \quad (2.9)$$

dimana,

P_{ij}^k = himpunan kota yang akan dikunjungi semut k

U_k = himpunan kota yang belum dikunjungi ketika berada pada kota i

τ_{ij} = jumlah feromon pada sisi (i,j)

η_{ij} = nilai visibilitas kota i dan kota j , dengan metode *saving*

β = parameter pengendali visibilitas ($\beta > 0$)

q = bilangan random dalam eksploitasi dan eksplorasi

q_0 = parameter untuk kepentingan relatif dari eksploitasi dan eksplorasi

d_{ij} = jarak antara kota i dan kota j

d_{0i} = jarak antara depot dan kota i

d_{0j} = jarak antara depot dan kota j .

2) Pembaruan feromon lokal

Feromon setiap busur (i,j) akan diperbaharui oleh semut berdasarkan *local update*. Pembaruan feromon digunakan untuk membangun tur ketika melewati sisi dan mengubah tingkat feromon. Aturan pembaruan feromon sebagai berikut:

$$\tau_{ij}^{new} = \tau_{ij}^{old} + \rho \Delta \tau_{ij} \quad (2.10)$$

$$\Delta \tau_{ij} = \frac{1}{L_{nn}.n} \quad (2.11)$$

dimana,

τ_{ij}^{new} = jumlah feromon pada sisi (i,j) baru

τ_{ij}^{old} = jumlah feromon pada sisi (i,j) awal

ρ = ketetapan penguapan feromon lokal dengan $0 \leq \rho \leq 1$

$\Delta \tau_{ij}$ = perubahan feromon pada sisi (i,j)

n = jumlah lokasi

L_{nn} = panjang tur yang diperoleh.

- e. Jika semut $>$ maks(semut), maka semut = 2 dan dilanjutkan ke langkah f. Jika tidak maka kembali ke langkah d.
- f. Mengurutkan solusi kedua sampai dengan jumlah maksimum (semut) dan menerapkan *local search* pada solusi terbaik dan disimpan sebagai solusi kedua.
- g. Lakukan pembaruan feromon global pada busur yang membentuk solusi pertama dan kedua. Semut terbaik meletakkan feromon pada busur yang dilalui. Setelah semua semut membentuk iterasinya, maka busur yang membentuk solusi pertama dan kedua jumlah feromonnya akan berubah. Tujuannya yaitu untuk memberikan feromon yang lebih banyak pada tur yang lebih pendek. Pembaruan dilakukan dengan persamaan berikut:

$$\tau_{ij} = (1 - \gamma) \cdot \tau_{ij} + \gamma \cdot \Delta \tau_{ij} \quad (2.12)$$

$$\Delta \tau_{ij} = \frac{(L_3 - L_g) + (L_3 - L_1)}{L_3} \quad (2.13)$$

dimana,

τ_{ij} = jumlah feromon pada sisi (i,j)

γ = parameter penguapan feromon global dengan $0 \leq \gamma \leq 1$

$\Delta\tau_{ij}$ = perubahan feromon pada sisi (i,j)

L_g = panjang tur terbaik secara keseluruhan

L_1 = solusi terbaik setiap iterasi

L_3 = solusi terbaik ke-3 setiap iterasi.

- h. Pilih solusi terbaik dan disimpan sebagai solusi pertama pada generasi berikutnya, iterasi = iterasi + 1.
- i. Jika iterasi \geq max(iterasi) maka berhenti. Jika iterasi $<$ max(iterasi) maka kembali ke langkah d.

2.4.2 Algoritma *Hybrid Ant Colony System* (HACS)

Menurut Xiang-pei *et al.* (2006), ACS merupakan algoritma yang terinspirasi oleh koloni semut alami alam. ACS telah diusulkan untuk memecahkan berbagai jenis masalah optimasi kombinatorial. Namun, ACS mempunyai kelemahan yaitu solusi yang didapatkan dapat terjebak dalam lokal optimum dan kurang efisien. Untuk mencegah proses pencarian agar tidak terjebak secara lokal optimum dan memperbaiki efisiensi dari ACS, *Hybrid Ant Colony System* (HACS) disajikan dengan memperkenalkan pendekatan penyesuaian feromon, menggabungkan ACS dengan algoritma *saving* dan *interchange*. Berikut merupakan langkah-langkah dalam penyelesaian algoritma HACS yaitu:

- a. Mencari solusi optimal awal yang didapatkan dari algoritma *Nearest Neighbour Heuristic* (L_{global}) seperti pada algoritma IACS dan menentukan parameter. Parameter yang digunakan yaitu:
 - 1) Jumlah semut (m)
 - 2) Intensitas feromon (τ_{ij})
 - 3) Tetapan pengendali feromon (α)
 - 4) Tetapan pengendali visibilitas (β)
 - 5) Tetapan pengendali nilai *saving* (γ)
 - 6) Tetapan penguapan feromon (ρ).

b. Tentukan titik selanjutnya dengan aturan berikut:

$$J = \begin{cases} \max_{j \in U_k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\mu_{ij}]^\gamma & , q \leq p_t \\ \text{acak} & , \text{lainnya} \end{cases} \quad (2.14)$$

dimana,

J = himpunan kota yang akan dikunjungi

U_k = himpunan kota yang belum dikunjungi ketika berada pada kota i

τ_{ij} = kepadatan jumlah feromon pada sisi (i, j)

η_{ij} = nilai visibilitas yang merupakan jarak timbal balik antara dua titik (d_{ij})

μ_{ij} = nilai dari algoritma *saving*

α = tetapan pengendali feromon

β = tetapan pengendali visibilitas

γ = tetapan pengendali nilai *saving*

q = bilangan acak

p_t = nilai yang mempengaruhi pemilihan titik.

c. Jika semut $>$ maks(semut) maka dilanjutkan ke langkah d. Jika tidak maka kembali ke langkah b

d. Hitung solusi L_k dari setiap semut, simpan sebagai solusi lokal optimum (L_{lokal})

e. Metode *interchange* dilakukan pada L_{lokal} selama z kali. Perbarui rute dan simpan $L_{lokal} = L_{opt}$ jika $L_{opt} < L_{lokal}$. Berikut merupakan langkah-langkah dari metode *interchange*:

1) Random dua pasang posisi untuk dilakukan *interchange*

2) Pindahkan elemen berpasangan tersebut dari posisi 1 ke posisi 2

f. Dilakukan pembaruan feromon dengan aturan sebagai berikut:

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + \Delta \tau_{ij} \quad (2.15)$$

dengan,

$$\Delta \tau_{ij} = \begin{cases} \frac{1}{L_{lokal}} & , ij \text{ berdasarkan } L_{lokal} \\ 0 & , \text{lainnya} \end{cases} \quad (2.16)$$

dimana,

τ_{ij}^{new} = nilai feromon setelah diperbarui

τ_{ij}^{old} = nilai feromon awal

ρ = ketetapan penguapan feromon dengan nilai 0 sampai 1

$\Delta\tau_{ij}$ = perubahan feromon

Setelah feromon diperbarui, maka τ_{ij} diubah menjadi τ_{max} jika $\tau_{ij} > \tau_{max}$ atau

$\frac{(\tau_{min} + \tau_{max})}{2}$ jika $\tau_{ij} < \tau_{min}$

- g. Bandingkan L_{lokal} dengan L_{global} , jika didapatkan $L_{lokal} < L_{global}$ maka $L_{global} = L_{lokal}$, kemudian perbarui rute global optimum
- h. Adapun untuk melengkapi ruang pencarian, didefinisikan probabilitas seleksi p_t dan ρ dapat disesuaikan. Aturan penyesuaian p_t sebagai berikut:

$$p_t = \begin{cases} 0,95p_{t-1} & , \text{ jika } 0,95p_{t-1} \geq p_{min} \\ p_{min} & , \text{ lainnya} \end{cases} \quad (2.17)$$

dimana, p_{min} merupakan nilai minimum pada proses evolusioner yang digunakan untuk memastikan pemilihan meskipun p_t terlalu kecil. Sedangkan aturan penyesuaian ρ yaitu:

$$\rho_n = \begin{cases} 0,95\rho_{n-1} & , \text{ jika } 0,95\rho_{n-1} \geq \rho_{min} \\ \rho_{min} & , \text{ lainnya} \end{cases} \quad (2.18)$$

dimana, ρ_{min} merupakan nilai minimum pada proses evolusioner yang digunakan untuk mencegah kecepatan konvergensi yang terlalu lama jika ρ terlalu kecil

- i. Jika iterasi > maks(iterasi) maka selesai. Jika tidak maka kembali ke langkah b.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari skripsi Putri (2014) pada PT. Sumber Bening Lestari yang merupakan perusahaan air mineral Flow. Data yang digunakan adalah sebagai berikut:

- a. Jumlah pelanggan berada di 22 kecamatan di Kabupaten Jember, dengan setiap kecamatan hanya terdapat 1 pelanggan.
- b. Jarak dari depot ke pelanggan dan jarak antar pelanggan (Lampiran A).
- c. Banyak permintaan setiap pelanggan (Lampiran B).
- d. Terdapat 4 kendaraan, dengan setiap kendaraan berkapasitas 210 karton.

3.2 Langkah-langkah Penelitian

Langkah yang akan dilakukan dalam penelitian tentang “Penerapan Algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System* pada *Capacitated Vehicle Routing Problem*” secara skematik dapat dilihat pada Gambar 3.1. Berikut langkah-langkah penelitian:

a. Studi Literatur

Penelitian ini diawali dengan mempelajari buku, jurnal dan skripsi yang membahas mengenai *Vehicle Routing Problem*, *Capacitated Vehicle Routing Problem*, *Improved Ant Colony System* dan *Hybrid Ant Colony System*.

b. Pengambilan Data

Langkah kedua yaitu mengambil dan mengumpulkan data yang diperoleh dari skripsi Putri (2014) pada PT. Sumber Bening Lestari yang merupakan perusahaan air mineral Flow.

c. Penerapan IACS dan HACS

Suatu data dipilih untuk mencari rute terpendek. Data tersebut diterapkan menggunakan algoritma *Improved Ant Colony System* seperti pada subsubbab 2.4.1 dan *Hybrid Ant Colony System* pada subsubbab 2.4.2.

d. Pembuatan Program

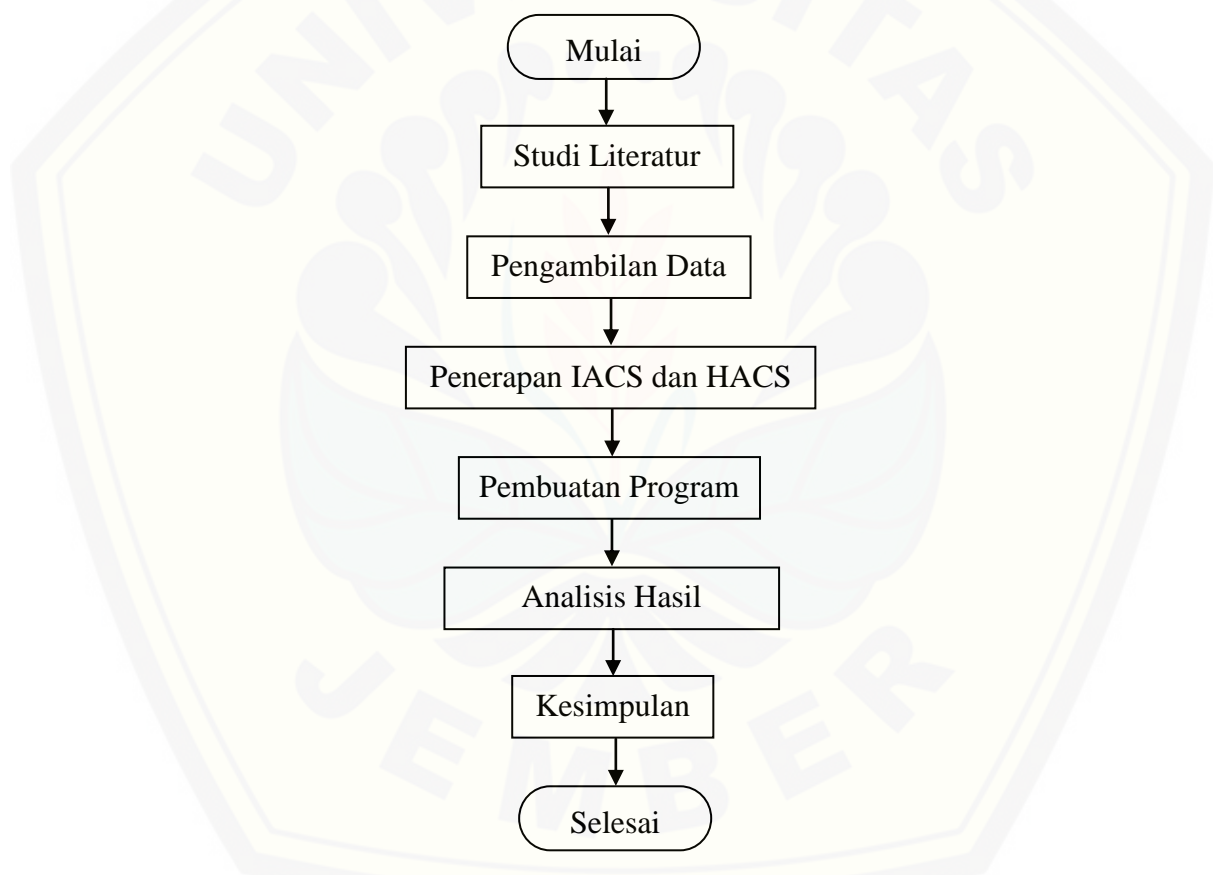
Peneliti akan membuat skrip pemrograman dengan menggunakan Matlab R2009a untuk menyelesaikan *Capacitated Vehicle Routing Problem* dengan algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System*.

e. Analisis Hasil

Percobaan menggunakan program akan dilakukan beberapa kali dengan data. Hasil dari percobaan akan dianalisis untuk mengetahui rute optimal.

e. Kesimpulan

Hasil dari analisis hasil akan ditarik kesimpulan.



Gambar 3.1 Skema Langkah-langkah Penelitian

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini dibahas hasil penyelesaian permasalahan *Capacitated Vehicle Routing Problem* dengan menggunakan algoritma *Improved Ant Colony System* dan *Hybrid Ant Colony System*. Untuk menyelesaikan permasalahan tersebut, digunakan program yang dibuat dengan bahasa pemrograman MATLAB. Kedua algoritma tersebut akan dibandingkan berdasarkan total jarak tempuh minimum.

4.1 Hasil

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari skripsi Putri (2014) pada PT. Sumber Bening Lestari yang merupakan perusahaan air mineral Flow. Data tersebut berupa jarak dari depot ke setiap pelanggan maupun antar pelanggan, jumlah permintaan pelanggan, jumlah kendaraan dan kendala kapasitas dari sejumlah kendaraan.

4.1.1 Langkah Perhitungan

Untuk memberikan gambaran langkah-langkah algoritma IACS dan HACS untuk penyelesaian kasus CVRP maka diberikan contoh penyelesaian manual dengan menggunakan data jarak dari depot ke setiap pelanggan maupun jarak antar pelanggan (dapat dilihat pada Tabel 4.1) dan data berupa jumlah permintaan 5 pelanggan (dapat dilihat pada Tabel 4.2), terdapat 2 truk serta kapasitas setiap kendaraan sebesar 100 karton. Data tersebut merupakan data sekunder yang diambil dari skripsi Ulyawati (2016)

Tabel 4.1 Jarak Pelanggan

Depot	1	2	3	4	5	
Depot	0	50	45	40	34	29
1	50	0	7	10	15	20
2	45	7	0	15	10	15
3	40	10	15	0	8	14
4	34	15	10	8	0	6
5	29	20	15	14	6	0

Tabel 4.2 Permintaan Pelanggan

Pelanggan	Permintaan
1	23
2	26
3	42
4	32
5	37

- a. Langkah perhitungan dengan algoritma *Improved Ant Colony System*.

Berikut ini adalah pencarian CVRP menggunakan algoritma *Improved Ant Colony System* dalam satu iterasi dengan 2 kendaraan berkapasitas 100 karton dan 5 pelanggan. Langkah-langkah penyelesaiannya adalah sebagai berikut:

- 1) Menentukan parameter

Langkah pertama, penulis menentukan parameter diantaranya sebagai berikut:

- Intensitas feromon (τ_{ij}) = 0,01
- Pengendali visibilitas (β) = 2
- Penguapan feromon global (γ) = 0,1
- Jumlah semut (m) = 2
- Penguapan feromon lokal (ρ) = 0,1

- 2) Membangkitkan inisial solusi menggunakan *Nearest Neighbour Heuristic*

Solusi awal dibangkitkan dengan *Nearest Neighbor Heuristic*. Lokasi pertama dimulai dari depot dengan kapasitas daya angkut kendaraan maksimal 100 karton. Perjalanan ini dimulai dari depot kemudian memilih pelanggan dengan jarak yang minimum, sehingga untuk kendaraan 1 memulai perjalanan dari depot ke pelanggan 5 kemudian pelanggan 4 dan kembali ke depot karena untuk pelanggan selanjutnya kapasitas kendaraan tidak dapat menampung permintaan pelanggan, sehingga total kapasitas yang diangkut kendaraan 1 yaitu 69 karton. Untuk perjalanan selanjutnya digunakan kendaraan 2 yang dimulai dari depot ke pelanggan 3, pelanggan 1 dan pelanggan 2 kemudian kembali ke depot asal dengan total kapasitas

yang diangkut kendaraan 2 yaitu 91 karton. Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 5 – pelanggan 4 – depot

K2 : depot – pelanggan 3 – pelanggan 1 – pelanggan 2 – depot

Jarak tempuh yang dilalui kendaraan :

K1 : $0 - 5 - 4 - 0 = 29 + 6 + 34 = 69$ km

K2 : $0 - 3 - 1 - 2 - 0 = 40 + 10 + 7 + 45 = 102$ km

Total jarak tempuh : $69 + 102 = 171$ km

3) Menerapkan *local search (insertion)*

Lakukan *local search* pada solusi awal. Pilih dua elemen yang akan digunakan dalam proses digeser (*insertion*)

Random posisi 1 = 5

Random posisi 2 = 1

Setelah dilakukan random posisi, elemen posisi pertama digeser ke posisi kedua.

Solusi awal	5	4	3	1	2
Solusi baru	4	3	1	5	2

Dari *local search* tersebut didapatkan solusi awal baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 4 dan pelanggan 3, kemudian kembali ke depot dengan kapasitas total 74 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 1, pelanggan 5 dan pelanggan 2 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 86 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6).

Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 4 – pelanggan 3 – depot

K2 : depot – pelanggan 1 – pelanggan 5 – pelanggan 2 – depot

Jarak tempuh yang dilalui kendaraan :

$$K1 : 0 - 4 - 3 - 0 = 82 \text{ km}$$

$$K2 : 0 - 1 - 5 - 2 - 0 = 130 \text{ km}$$

$$\text{Total jarak tempuh : } 82 + 130 = 212 \text{ km}$$

Solusi awal yang telah didapatkan dari perhitungan *Nearest Neighbour Heuristic* dan solusi yang didapatkan dari hasil *local search* (solusi baru) dibandingkan berdasarkan persamaan (2.1) yaitu mengenai jarak tempuh yang minimum. Jika solusi awal lebih optimum dari solusi baru maka solusi awal tetap seperti semula, namun jika solusi baru lebih optimum dibandingkan solusi baru maka solusi baru tersebut dijadikan solusi awal yang kemudian digunakan untuk perhitungan selanjutnya. Dari perhitungan manual ini, didapatkan bahwa solusi awal menghasilkan jarak tempuh 171 km, sedangkan solusi baru sebesar 212 km, sehingga dapat disimpulkan bahwa solusi yang digunakan tetap solusi awal.

- 4) Membentuk solusi berdasarkan aturan mengkonstruksi rute dan lakukan pembaruan feromon lokal. Langkah pertama dalam aturan mengkonstruksi rute yaitu menghitung nilai visibilitas, seperti pada persamaan (2.9). Berikut perhitungannya:

$$\eta_{12} = d_{10} + d_{02} - d_{12} = 50 + 45 - 7 = 88$$

$$\eta_{13} = d_{10} + d_{03} - d_{13} = 50 + 40 - 10 = 80$$

$$\eta_{14} = d_{10} + d_{04} - d_{14} = 50 + 34 - 15 = 69$$

$$\eta_{15} = d_{10} + d_{05} - d_{15} = 50 + 29 - 20 = 59$$

$$\eta_{23} = d_{20} + d_{03} - d_{23} = 45 + 40 - 15 = 70$$

$$\eta_{24} = d_{20} + d_{04} - d_{24} = 45 + 34 - 10 = 69$$

$$\eta_{25} = d_{20} + d_{05} - d_{25} = 45 + 29 - 20 = 54$$

$$\eta_{34} = d_{30} + d_{04} - d_{34} = 40 + 34 - 8 = 66$$

$$\eta_{35} = d_{30} + d_{05} - d_{35} = 40 + 29 - 14 = 55$$

$$\eta_{45} = d_{40} + d_{05} - d_{45} = 34 + 29 - 6 = 57$$

Didapatkan nilai *saving* yang terbesar yaitu 88 terdapat pada visibilitas Π_{12} atau pada kemungkinan rute 0 – 1 – 2 dan rute 0 – 2 – 1. Sesuai dengan persamaan (2.1) mengenai jarak tempuh minimum maka dari kedua rute tersebut dipilih jarak yang terpendek. Rute 0 – 1 – 2 mempunyai jarak tempuh 57 km dan rute 0 – 2 – 1 sebesar 52 km, dapat disimpulkan bahwa rute yang diambil yaitu 0 – 2 – 1.

Kemudian untuk rute selanjutnya menggunakan persamaan (2.7) dan (2.8). Nilai q dibangkitkan secara acak, kemudian dibandingkan dengan parameter q_0 untuk memilih eksploitasi atau eksplorasi. Pada perhitungan manual ini digunakan 2 semut yaitu:

Semut 1

Untuk $q = 0,2$ maka,

$$P_{13}^1 = \frac{(0,01)(80)^2}{(0,01)(80)^2 + (0,01)(69)^2 + (0,01)(59)^2}$$

$$= \frac{64}{64 + 47,61 + 34,81}$$

$$= \frac{64}{146,42}$$

$$= 0,43710$$

$$P_{14}^1 = \frac{47,61}{146,42}$$

$$= 0,32516$$

$$P_{15}^1 = \frac{38,41}{146,42}$$

$$= 0,26233$$

Setelah didapatkan nilai P, maka dihitung nilai probabilitas kumulatif (PK) dari nilai P.

P	PK
0,43710	0,43710
0,32516	0,76226
0,26233	1

Untuk memilih rute diperlukan suatu bilangan random. Pada langkah ini digunakan $\text{rand}=0,7$ sehingga:

$\text{PK}_{13} = 0,43710$ maka $0,43710 - 1 < 0,7 \leq 0,43710$ (tidak memenuhi)

$\text{PK}_{14} = 0,76226$ maka $0,76226 - 1 < 0,7 \leq 0,76226$

$\text{PK}_{15} = 1$ maka $1 - 1 < 0,7 \leq 1$

Rute yang dipilih selanjutnya yaitu $0 - 2 - 1 - 4$

Untuk $q = 0,3$ maka

$$\begin{aligned} P_{43}^1 &= \frac{(0,01)(66)^2}{(0,01)(66)^2 + (0,01)(57)^2} \\ &= \frac{43,56}{43,56 + 32,49} \\ &= \frac{43,56}{76,05} \\ &= 0,57278 \end{aligned}$$

$$\begin{aligned} P_{45}^1 &= \frac{32,49}{76,05} \\ &= 0,42722 \end{aligned}$$

Setelah didapatkan nilai P, maka dihitung nilai probabilitas kumulatif (PK) dari nilai P.

P	PK
0,57278	0,57278
0,42722	1

Untuk memilih rute diperlukan suatu bilangan random. Pada langkah ini digunakan $\text{rand}=0,4$ sehingga:

$\text{PK}_{43} = 0,57278$ maka $0,57278 - 1 < 0,4 \leq 0,57278$

$\text{PK}_{45} = 1$ maka $1 - 1 < 0,4 \leq 1$

Rute yang dipilih selanjutnya yaitu $0 - 2 - 1 - 4 - 3$

Untuk $q = 0,09$

$$P_{45}^1 = [(\tau_{ij})(\eta_{ij})^q] = (0,01)(57)^2 = 32,49$$

Rute yang dipilih selanjutnya yaitu $0 - 2 - 1 - 4 - 3 - 5$

Dari transisi status tersebut didapatkan solusi awal baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke

pelanggan 2, pelanggan 1 dan pelanggan 4, kemudian kembali ke depot dengan kapasitas total 81 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 3 dan pelanggan 5 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 79 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 2 – pelanggan 1 – pelanggan 4 – depot

K2 : depot – pelanggan 3 – pelanggan 5 – depot

Jarak tempuh yang dilalui kendaraan:

K1 : 0 – 2 – 1 – 4 – 0 = 101 km

K2 : 0 – 3 – 5 – 0 = 83 km

Total jarak tempuh: 101 + 83 = 184 km

Setelah didapatkan rute, dilakukan pembaruan feromon dengan persamaan (2.15), dan hasil perhitungan dapat dilihat pada Tabel 4.3.

$$\tau_0 = \frac{1}{L_{m.n}} = \frac{1}{184 \times 6} = 0,00091$$

$$\tau_{ij}^{new} = 0,01 + (0,1 \times 0,00091) = 0,01009$$

Tabel 4.3 Pembaruan Feromon Lokal IACS Pertama

<i>i</i>	<i>j</i>	<i>c_{ij}</i>	<i>η_{ij}</i>	<i>τ_{ij}</i>	P	PK	Rand	Pembaruan feromon lokal
0	2	45	-	0,01	-	-	-	0,01009
2	1	7	88	0,01	-	-	-	0,01009
1	4	15	69	0,01	0,32516	0,76226	0,7	0,01009
4	3	8	66	0,01	0,57278	0,57278	0,4	0,01009
3	5	14	55	0,01	32,49	-	-	0,01009

Semut 2

Untuk $q = 0,06$, maka

$$P_{13}^1 = [(\tau_{ij})(\eta_{ij})^\beta] = (0,01)(80)^2 = 64$$

$$P_{14}^1 = [(\tau_{ij})(\eta_{ij})^\beta] = (0,01009)(69)^2 = 48,03849$$

$$P_{15}^1 = [(\tau_{ij})(\eta_{ij})^\beta] = (0,01)(59)^2 = 34,81$$

Dipilih nilai probabilitas terbesar, sehingga rute selanjutnya yaitu 0 – 2 – 1 – 3

Untuk $q = 0,088$

$$P_{34}^1 = [(\tau_{ij})(\eta_{ij})^\beta] = (0,01009)(66)^2 = 43,95204$$

$$P_{35}^1 = [(\tau_{ij})(\eta_{ij})^\beta] = (0,01009)(55)^2 = 30,52225$$

Dipilih nilai probabilitas terbesar, sehingga rute selanjutnya yaitu 0 – 2 – 1 – 3 – 4

Untuk $q = 0,05$

$$P_{45}^1 = [(\tau_{ij})(\eta_{ij})^\beta] = (0,01)(57)^2 = 32,49$$

Rute selanjutnya yaitu 0 – 2 – 1 – 3 – 4 – 5 – 0

Dari transisi status tersebut didapatkan solusi awal baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 2, pelanggan 1 dan pelanggan 3, kemudian kembali ke depot dengan kapasitas total 91 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 4 dan pelanggan 5 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 69 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 2 – pelanggan 1 – pelanggan 3 – depot

K2 : depot – pelanggan 4 – pelanggan 5 – depot

Jarak tempuh yang dilalui kendaraan:

K1 : 0 – 2 – 1 – 3 – 0 = 102 km

K2 : 0 – 4 – 5 – 0 = 69 km

Total jarak tempuh: 102 + 69 = 171 km

Setelah didapatkan rute, dilakukan pembaruan feromon dengan persamaan (2.15) dan (2.16), hasil perhitungan ditunjukkan pada Tabel 4.4.

$$\tau_0 = \frac{1}{L_{mn} \cdot n} = \frac{1}{171 \times 6} = 0,00098$$

$$\tau_{ij}^{new} = 0,01 + (0,1 \times 0,00098) = 0,01010$$

$$\tau_{ij}^{new} = 0,01009 + (0,1 \times 0,00098) = 0,01019$$

Tabel 4.4 Pembaruan Feromon Lokal IACS Kedua

i	j	c_{ij}	η_{ij}	τ_{ij}	P	PK	rand	Pembaruan feromon lokal
0	2	45	-	0,01009	-	-	-	0,01019
2	1	7	88	0,01009	-	-	-	0,01019
1	3	10	80	0,01000	64	-	-	0,01010
3	4	8	66	0,01009	43,95204	-	-	0,01019
4	5	6	57	0,01009	32,49000	-	-	0,01019

Dari perhitungan semut 1 dan 2 diperoleh hasil, semut 1 dengan rute 0 – 2 – 1 – 4 – 3 – 5 – 0 dan jarak tempuh 184 km. Sedangkan semut 2 dengan rute 0 – 2 – 1 – 3 – 4 – 5 – 0 dan jarak tempuh 171 km. Kedua solusi tersebut kemudian dibandingkan berdasarkan persamaan (2.1) yaitu mengenai jarak tempuh yang minimum. Dengan cara yang sama dengan sebelumnya dapat disimpulkan bahwa solusi yang digunakan yaitu terdapat pada semut 2.

5) Lakukan *local search* pada solusi terbaik dari perhitungan semut

Solusi terbaik yaitu terdapat pada semut ke dua dengan rute 0 – 2 – 1 – 3 – 4 – 5 – 0

Dengan metode *insertion* seperti langkah sebelumnya, maka

Random posisi 1 = 5

Random posisi 2 = 1

Setelah dilakukan random posisi, elemen posisi pertama digeser ke posisi kedua.

Solusi awal	2	1	3	4	5
Solusi baru	2	5	1	3	4

Dari *local search* tersebut didapatkan solusi awal baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 2, pelanggan 5 dan pelanggan 1, kemudian kembali ke depot dengan kapasitas total 86 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 3 dan pelanggan 4 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 74 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 2 – pelanggan 5 – pelanggan 1 – depot

K2 : depot – pelanggan 3 – pelanggan 4 – depot

Jarak tempuh yang dilalui kendaraan:

K1 : $0 - 2 - 5 - 1 - 0 = 130$ km

K2 : $0 - 3 - 4 - 0 = 82$ km

Total jarak tempuh: $130 + 82 = 212$ km

Solusi awal yang telah didapatkan dari perhitungan semut 2 dan solusi yang didapatkan dari hasil *local search* dibandingkan berdasarkan persamaan (2.1) yaitu mengenai jarak tempuh yang minimum. Dari perhitungan didapatkan bahwa solusi awal dengan jarak tempuh 181 km dan solusi setelah *local search* 212 km. Dengan cara yang sama dengan sebelumnya

dapat disimpulkan bahwa solusi yang digunakan yaitu terdapat pada semut 2.

6) Lakukan pembaruan feromon global

Dilakukan pembaruan feromon global pada solusi pertama dan kedua dengan persamaan (2.12) dan (2.13). Hasil rute yang telah diperoleh pada perhitungan sebelumnya diurutkan.

$$0 - 5 - 4 - 3 - 1 - 2 - 0 = 171\text{km (solusi pertama)}$$

$$0 - 4 - 3 - 1 - 5 - 2 - 0 = 212\text{km}$$

$$0 - 2 - 1 - 4 - 3 - 5 - 0 = 184\text{km}$$

$$0 - 2 - 1 - 3 - 4 - 5 - 0 = 171\text{km (solusi kedua)}$$

$$0 - 2 - 5 - 1 - 3 - 4 - 0 = 212\text{km}$$

Hasil perhitungan diurutkan secara *ascending* untuk mendapatkan solusi terbaik ketiga.

$$0 - 5 - 4 - 3 - 1 - 2 - 0 = 171\text{km}$$

$$0 - 2 - 1 - 3 - 4 - 5 - 0 = 171\text{km}$$

$$0 - 2 - 1 - 4 - 3 - 5 - 0 = 184\text{km (solusi ketiga)}$$

$$0 - 2 - 5 - 1 - 3 - 4 - 0 = 212\text{km}$$

$$0 - 4 - 3 - 1 - 5 - 2 - 0 = 212\text{km}$$

Hitung perubahan feromon dengan menggunakan solusi terbaik secara global, iterasi dan solusi terbaik ketiga dari tiap iterasi. Dapat dilihat pada Tabel 4.5 dan Tabel 4.6.

$$\Delta\tau_{ij} = \frac{(184-171)+(184-171)}{184} = 0,14131$$

Solusi pertama

$$\tau_{05} = (1 - 0,1) \cdot 0,01 + 0,1 \cdot 0,14131 = 0,02313$$

$$\tau_{54} = (1 - 0,1) \cdot 0,01 + 0,1 \cdot 0,14131 = 0,02313$$

$$\tau_{43} = (1 - 0,1) \cdot 0,01009 + 0,1 \cdot 0,14131 = 0,02321$$

$$\tau_{31} = (1 - 0,1) \cdot 0,01 + 0,1 \cdot 0,14131 = 0,02313$$

$$\tau_{12} = (1 - 0,1) \cdot 0,01 + 0,1 \cdot 0,14131 = 0,02313$$

Tabel 4.5 Pembaruan Feromon Global Solusi Pertama

i	j	Feromon global
0	5	0,02313
5	4	0,02313
4	3	0,02321
3	1	0,02313
1	2	0,02313

Solusi kedua

$$\tau_{02} = (1 - 0,1) \cdot 0,01019 + 0,1 \cdot 0,14131 = 0,02330$$

$$\tau_{21} = (1 - 0,1) \cdot 0,01019 + 0,1 \cdot 0,14131 = 0,02330$$

$$\tau_{13} = (1 - 0,1) \cdot 0,01010 + 0,1 \cdot 0,14131 = 0,02322$$

$$\tau_{34} = (1 - 0,1) \cdot 0,01019 + 0,1 \cdot 0,14131 = 0,02330$$

$$\tau_{45} = (1 - 0,1) \cdot 0,01019 + 0,1 \cdot 0,14131 = 0,02330$$

Tabel 4.6 Pembaruan Feromon Global Solusi Kedua

i	j	Feromon global
0	1	0,02330
1	2	0,02330
2	3	0,02322
3	4	0,02330
4	5	0,02330

Feromon global dan rute terbaik pada iterasi pertama inilah yang akan digunakan pada iterasi selanjutnya.

b. Langkah perhitungan dengan algoritma *Hybrid Ant Colony System*.

Berikut ini adalah pencarian CVRP menggunakan algoritma *Hybrid Ant Colony System* dalam satu iterasi dengan 2 kendaraan berkapasitas 100 karton dan 5 pelanggan. Langkah-langkah penyelesaiannya adalah sebagai berikut:

1) Menentukan parameter

Solusi optimal awal didapatkan dari algoritma *Nearest Neighbour Heuristic* (L_{global}) seperti pada hasil di IACS yaitu pada rute $0 - 5 - 4 - 3 - 1 - 2 - 0$ dengan jarak tempuh 171 km, kemudian penulis menentukan parameter diantaranya sebagai berikut:

- a. Jumlah semut (m) = 2
- b. Intensitas feromon (τ_{ij}) = 0,01
- c. Tetapan pengendali feromon (α) = 0,1
- d. Tetapan pengendali visibilitas (β) = 2
- e. Tetapan pengendali nilai *saving* (γ) = 0,1
- f. Penguapan feromon (ρ) = 0,1

2) Tentukan titik selanjutnya

Langkah pertama yaitu menghitung nilai visibilitas seperti pada IACS, sehingga didapatkan rute awal $0 - 2 - 1$. Untuk rute selanjutnya menggunakan dengan persamaan (2.14), dimulai dari 1.

Semut 1, $p_0 = 1$

Untuk $q = 0,5$

$$j_{13} = [0,01]^{0,1} [10]^2 [80]^{0,1} = 97,79328$$

$$j_{14} = [0,01]^{0,1} [15]^2 [69]^{0,1} = 216,80407$$

$$j_{15} = [0,01]^{0,1} [20]^2 [59]^{0,1} = 379,44182$$

Karena dipilih rute dengan hasil terbesar, maka $0 - 2 - 1 - 5$

Untuk $q = 0,3$

$$j_{53} = [0,01]^{0,1} [14]^2 [55]^{0,1} = 184,62578$$

$$j_{54} = [0,01]^{0,1} [6]^2 [57]^{0,1} = 34,03219$$

Karena dipilih rute dengan hasil terbesar, maka $0 - 2 - 1 - 5 - 3$

Untuk $q = 0,7$

$$j_{34} = [0,01]^{0,1} [8]^2 [66]^{0,1} = 61,39519$$

Rute yang didapatkan $0 - 2 - 1 - 5 - 3 - 4 - 0$

Dari transisi status tersebut didapatkan solusi awal baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu

maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 2, pelanggan 1 dan pelanggan 5, kemudian kembali ke depot dengan kapasitas total 86 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 3 dan pelanggan 4 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 74 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 2 – pelanggan 1 – pelanggan 5 – depot

K2 : depot – pelanggan 3 – pelanggan 4 – depot

Jarak tempuh yang dilalui kendaraan:

$$K1 : 0 - 2 - 1 - 5 - 0 = 101 \text{ km}$$

$$K2 : 0 - 3 - 4 - 0 = 82 \text{ km}$$

$$\text{Total jarak tempuh: } 101 + 82 = 183 \text{ km}$$

Semut 2

Untuk $q = 0,88$

$$j_{13} = [0,01]^{0,1} [10]^2 [80]^{0,1} = 97,79328$$

$$j_{14} = [0,01]^{0,1} [15]^2 [69]^{0,1} = 216,80407$$

$$j_{15} = [0,01]^{0,1} [20]^2 [59]^{0,1} = 379,44182$$

Karena dipilih rute dengan hasil terbesar, maka $0 - 2 - 1 - 5$

Untuk $q = 0,67$

$$j_{53} = [0,01]^{0,1} [14]^2 [55]^{0,1} = 184,62578$$

$$j_{54} = [0,01]^{0,1} [6]^2 [57]^{0,1} = 34,03219$$

Karena dipilih rute dengan hasil terbesar, maka $0 - 2 - 1 - 5 - 3$

Untuk $q = 0,4$

$$j_{34} = [0,01]^{0,1} [8]^2 [66]^{0,1} = 61,39519$$

Rute yang didapatkan $0 - 2 - 1 - 5 - 3 - 4 - 0$

Dari transisi status tersebut didapatkan solusi awal baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 2, pelanggan 1 dan pelanggan 5, kemudian kembali ke depot dengan kapasitas total 86 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 3 dan pelanggan 4 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 74 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 2 – pelanggan 1 – pelanggan 5 – depot

K2 : depot – pelanggan 3 – pelanggan 4 – depot

Jarak tempuh yang dilalui kendaraan:

$$K1 : 0 - 2 - 1 - 5 - 0 = 101 \text{ km}$$

$$K2 : 0 - 3 - 4 - 0 = 82 \text{ km}$$

$$\text{Total jarak tempuh: } 101 + 82 = 183 \text{ km}$$

3) Hitung solusi dari setiap semut, simpan sebagai solusi lokal optimum (L_{lokal})

$$\text{Semut 1 } 0 - 2 - 1 - 5 - 3 - 4 - 0 = 181 \text{ km}$$

$$\text{Semut 2 } 0 - 2 - 1 - 5 - 3 - 4 - 0 = 181 \text{ km}$$

$$\text{Solusi } L_{lokal} \text{ yaitu } 0 - 2 - 1 - 5 - 3 - 4 - 0 = 181 \text{ km}$$

4) Dilakukan metode *interchange*.

Pada perhitungan ini digunakan $z = 2$. Pilih dua elemen yang saling berdekatan dan dua elemen lain yang juga berdekatan, kemudian tukar posisi kedua pasang elemen tersebut.

Untuk $z = 1$

Random posisi 1 = 2 – 1

Random posisi 2 = 5 – 3

Setelah dilakukan random posisi, elemen posisi pertama digeser ke posisi kedua.

Solusi awal	2	1	5	3	4
Solusi baru	5	3	2	1	4

Berdasarkan metode *interchange* tersebut didapatkan solusi baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 5 dan pelanggan 3, kemudian kembali ke depot dengan kapasitas total 79 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 2, pelanggan 1 dan pelanggan 4 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 81 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 5 – pelanggan 3 – depot

K2 : depot – pelanggan 2 – pelanggan 1 – pelanggan 4 – depot

Jarak tempuh yang dilalui kendaraan:

K1 : $0 - 5 - 3 - 0 = 83$ km

K2 : $0 - 2 - 1 - 4 - 0 = 101$ km

Total jarak tempuh: $83 + 101 = 184$ km

Untuk $z = 2$, dengan langkah yang sama maka

Random posisi 1 = 1 – 5

Random posisi 2 = 3 – 4

Setelah dilakukan random posisi, elemen posisi pertama digeser ke posisi kedua.

Solusi awal	2	1	5	3	4
Solusi baru	2	3	4	1	5

Berdasarkan metode *interchange* tersebut didapatkan solusi baru, kemudian disesuaikan berdasarkan persamaan (2.3). Kapasitas angkut kendaraan yaitu maksimal 100 karton. Kendaraan 1 memulai perjalanan dari depot ke pelanggan 2, pelanggan 3 dan pelanggan 4, kemudian kembali ke depot dengan kapasitas total 100 karton, untuk pelanggan selanjutnya kapasitas kendaraan tidak bisa menampung permintaan pelanggan sehingga kendaraan kembali ke depot. Perjalanan dengan kendaraan 2 dimulai dari depot kemudian ke pelanggan 1 dan pelanggan 5 setelah itu kembali ke depot asal dengan total kapasitas angkut kendaraan 60 karton. Setiap pelanggan hanya dikunjungi tepat satu kali oleh satu kendaraan sesuai persamaan (2.2). Setiap kendaraan memulai perjalanan dari depot mengunjungi titik-titik pelanggan dan kembali ke depot sesuai persamaan (2.4) sampai (2.6). Berikut rute dari setiap kendaraan beserta jarak tempuh dari setiap kendaraan:

K1 : depot – pelanggan 2 – pelanggan 3 – pelanggan 4 – depot

K2 : depot – pelanggan 1 – pelanggan 5 – depot

Jarak tempuh yang dilalui kendaraan:

K1 : $0 - 2 - 3 - 4 - 0 = 102$ km

K2 : $0 - 1 - 5 - 0 = 99$ km

Total jarak tempuh: $102 + 99 = 201$ km

Kemudian pilih rute terbaik dan dilakukan pembaruan rute. Simpan

$L_{lokal} = L_{opt}$ jika $L_{opt} < L_{lokal}$.

L_{lokal} (awal)	<i>Interchange</i>
0 – 2 – 1 – 5 – 3 – 4 – 0	0 – 5 – 3 – 2 – 1 – 4 – 0 = 184 km
0 – 2 – 1 – 5 – 3 – 4 – 0	0 – 2 – 3 – 4 – 1 – 5 – 0 = 201 km

Didapatkan bahwa rute 0 – 5 – 3 – 2 – 1 – 4 – 0 merupakan rute dengan jarak tempuh optimal, simpan sebagai L_{lokal} .

- 5) Dilakukan pembaruan feromon pada L_{lokal} dengan persamaan (2.15) dan (2.16). Sebagai contoh perhitungan yaitu:

$$\Delta\tau_{05} = \frac{1}{L_{lokal}} = \frac{1}{171} = 0,0058479532$$

$$\tau_{05}^{new} = \rho\tau_{05}^{old} + \Delta\tau_{05} = 0,1 \cdot 0,01 + 0,0058479532 = 0,0068479532$$

Untuk perhitungan selanjutnya dilakukan cara yang sama, disajikan dalam Tabel 4.7 berikut.

Tabel 4.7 Pembaruan Feromon Rute dari L_{lokal}

I	j	τ_{ij}^{old}	τ_{ij}^{new}
0	5	0,01	0,0068479532
5	4	0,01	0,0068479532
4	3	0,01	0,0068479532
3	1	0,01	0,0068479532
1	2	0,01	0,0068479532
2	0	0,01	0,0068479532

Sedangkan untuk rute yang tidak termasuk L_{lokal} , diperbarui berdasarkan aturan berikut.

$$\Delta\tau_{ij} = 0$$

$$\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \Delta\tau_{ij} = 0,1 \cdot 0,01 + 0 = 0,001$$

Setelah dilakukan pembaruan feromon, maka τ_{ij} diubah menjadi τ_{max} jika

$\tau_{ij} > \tau_{max}$ atau $\frac{(\tau_{min} + \tau_{max})}{2}$ jika $\tau_{ij} < \tau_{min}$. Pada perhitungan ini diperoleh

bahwa:

$$\tau_{ij} = 0,01$$

$$\tau_{max} = 0,0068479532$$

$$\tau_{min} = 0,001$$

Karena $\tau_{ij} > \tau_{max}$ maka $\tau_{ij} = 0,0068479532$

6) Bandingkan L_{lokal} dengan L_{global} , jika didapatkan $L_{lokal} < L_{global}$ maka

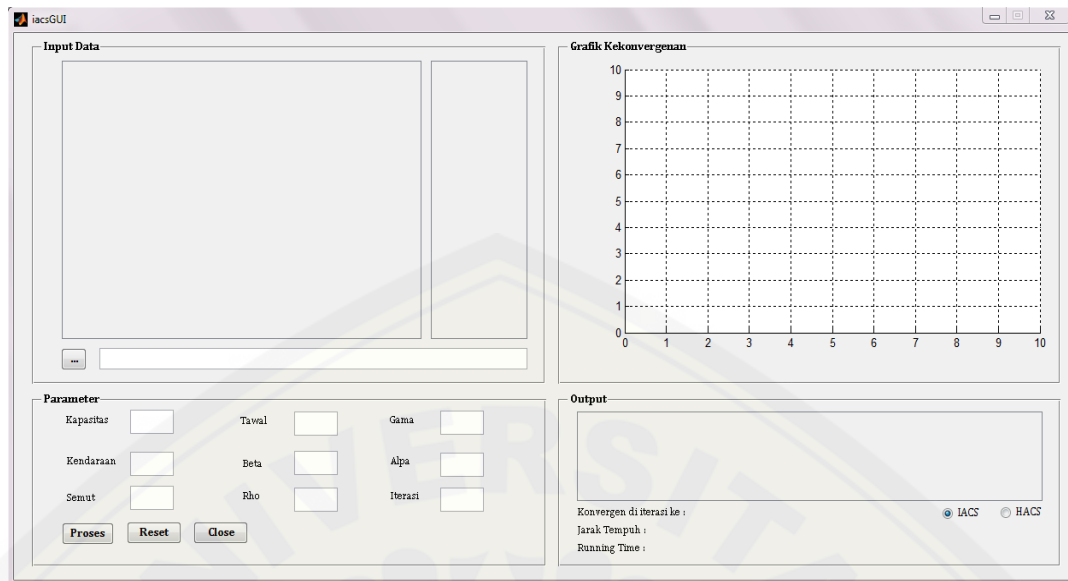
$$L_{global} = L_{lokal}$$

Dari perhitungan *Nearest Neighbour Heuristic* diperoleh bahwa nilai L_{global} terdapat pada rute 0 – 5 – 4 – 3 – 1 – 2 – 0 dengan jarak tempuh 171 km. Sedangkan nilai L_{lokal} dalam perhitungan ini diperoleh dari metode *interchange* yaitu pada rute 0 – 5 – 4 – 3 – 1 – 2 – 0 dengan jarak tempuh 171 km. Karena $L_{global} = L_{lokal}$ maka L_{global} yang akan digunakan pada iterasi berikutnya menggunakan rute dan jarak tempuh dari L_{lokal} .

7) Probabilitas seleksi dengan aturan penyesuaian p_t dan penyesuaian ρ menggunakan persamaan (2.18) dan (2.19). Pada perhitungan ini digunakan $p_t = 1$ dan $p_{min} = 0,2$, sehingga untuk pemilihan titik selanjutnya menggunakan $p_t = 0,95 \times 1 = 0,95$. Sedangkan untuk penguapan feromon ρ pada perhitungan ini digunakan $\rho = 0,1$ dan $p_{min} = 0,1$, sehingga untuk pemilihan titik selanjutnya menggunakan $p_t = 0,95 \times 0,1 = 0,095$.

4.1.2 *Capacitated Vehicle Routing Problem* dengan Program MATLAB.

Pada skripsi ini, dalam penyelesaian CVRP digunakan *software* MATLAB. Tujuan dari program ini yaitu untuk mendapatkan rute pengiriman barang dengan total jarak yang minimum serta membandingkan hasil penerapan dari kedua algoritma HACS dan IACS. Tampilan program dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan Awal Program

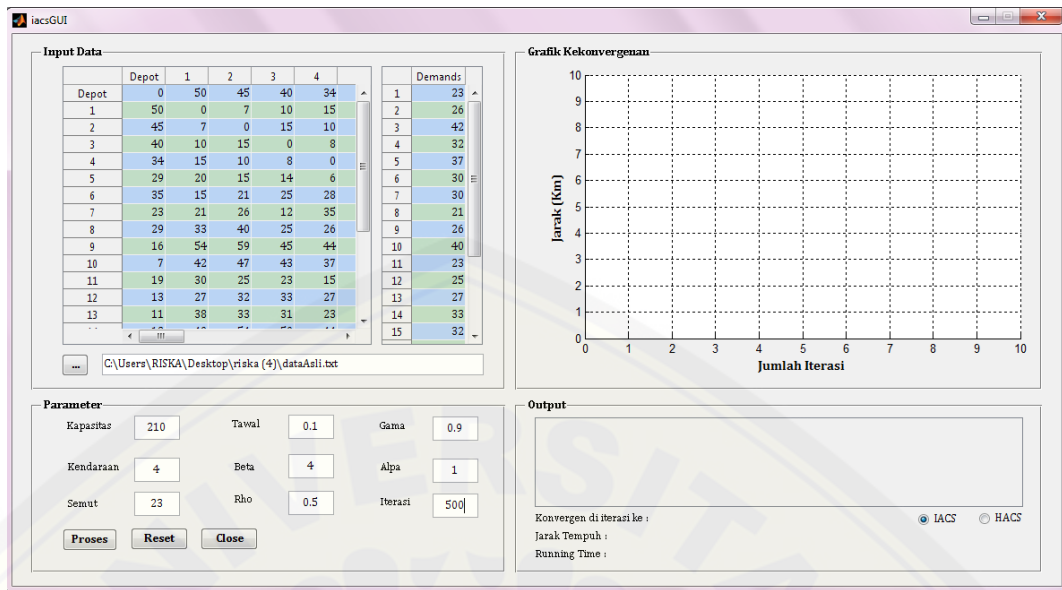
Tampilan program CVRP mempunyai beberapa komponen yaitu:

- Pushbutton* ... untuk memasukkan data. *Pushbutton* proses untuk melakukan proses running. *Pushbutton* reset untuk mengosongkan kembali tampilan GUI. *Pushbutton* close untuk keluar.
- Edit text* Kapasitas untuk memasukkan kapasitas kendaraan dari data yang telah diinputkan. *Edit text* Kendaraan untuk memasukkan jumlah kendaraan yang akan digunakan pada data. *Edit text* Semut, *Edit text* Tawal, *Edit text* Rho, *Edit text* Beta, *Edit text* Gama, *Edit text* Alpa untuk memasukkan berapa jumlah komponen yang akan digunakan pada proses pencarian solusi. *Edit text* Iterasi untuk memasukkan iterasi yang akan digunakan.
- Axis* untuk menampilkan grafik kekonvergenan dari kedua algoritma yaitu HACS dan IACS.
- Radiobutton* untuk melihat hasil kinerja dari masing-masing algoritma.

Langkah-langkah dalam menjalankan program di atas yaitu:

- Input data

Klik ... untuk memasukkan data jarak pelanggan dan jumlah permintaan. Data yang diinput berformat *.txt (*Text Document*).



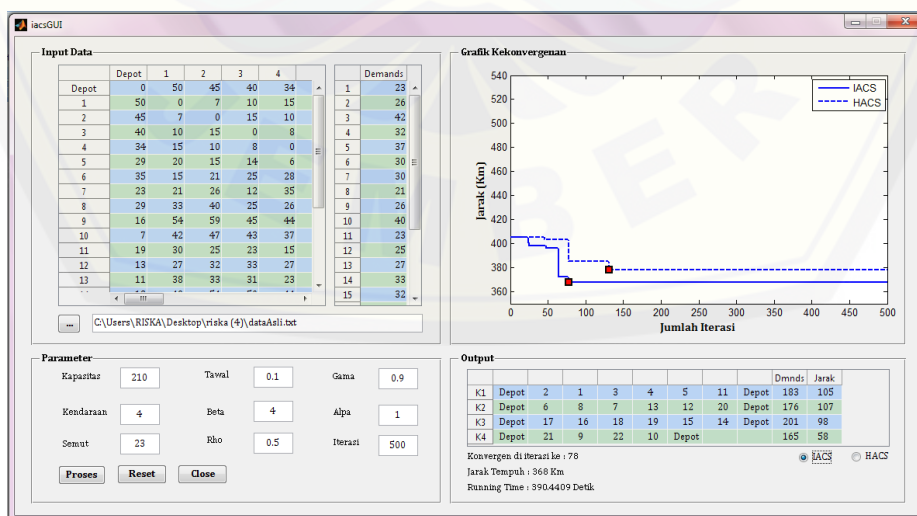
Gambar 4.2 Tampilan Input Parameter pada Program

b. Proses

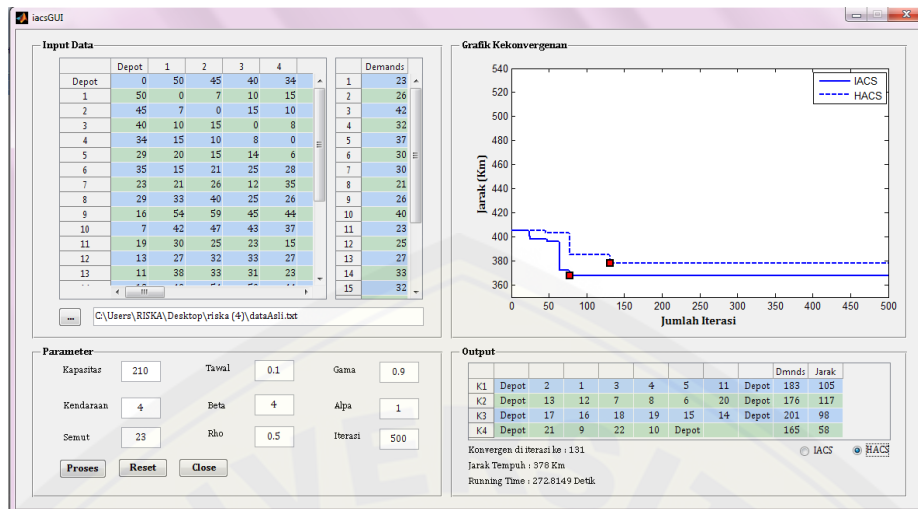
Parameter yang digunakan dalam algoritma harus diisi terlebih dahulu sebelum memulai proses. Kemudian lakukan proses dengan cara klik proses yang terdapat dalam GUI. Jika ingin mengosongkan data parameter maka klik reset. Pseudocode dapat dilihat pada Lampiran C.

c. Output

Setelah diklik proses maka data akan diproses hingga hasil ditampilkan.



Gambar 4.3 Tampilan Output Program CVRP Pada IACS

Gambar 4.4 Tampilan *Output* Program CVRP Pada HACS

d. Analisis Hasil

Untuk menganalisis hasil yang lebih baik, dapat dilihat pada kotak panel *output* yang menampilkan jumlah kendaraan, jarak tempuh tiap kendaraan, jarak tempuh secara total, iterasi kekonvergenan dan *running time*.

4.2 Pembahasan

Berdasarkan hasil kajian di atas, hal yang perlu dibahas yaitu penyelesaian CVRP dengan algoritma IACS dan HACS pada data sekunder yang akan menghasilkan rute optimal yang dilalui kendaraan, iterasi kekonvergenan dan *running time*. Data tersebut diproses dengan menggunakan Matlab dengan masing-masing parameter dilakukan 3 kali percobaan dengan nilai yang berbeda. Masing-masing percobaan dilakukan 5 kali *running* dengan nilai parameter yang sama. Hal tersebut dilakukan untuk mencari hasil yang paling optimal. Berikut hasil percobaannya:

a. Percobaan parameter intensitas feromon (τ_{ij})

1) Ditentukan parameter-parameter pada algoritma IACS dan HACS.

Intensitas feromon (τ_{ij}) = 0,01

Pengendali visibilitas (β) = 4

Penguapan feromon (ρ) = 0,5

Tetapan pengendali penguapan global dan nilai *saving* (γ) = 0,1

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dilakukan pada data terlampir, dapat disimpulkan bahwa algoritma IACS dan HACS dapat diterapkan pada permasalahan CVRP. Pada kasus khusus penelitian ini, jika dilihat dari segi total jarak tempuh maka algoritma IACS lebih baik daripada algoritma HACS karena mampu mendapatkan solusi yang lebih minimum. Berikut merupakan model matematis berdasarkan fungsi tujuan CVRP:

$$z = \sum_{k=1}^4 \sum_{i=0}^{22+1} \sum_{j=0}^{22+1} c_{ij} x_{ij}^k$$

Kendala dalam CVRP:

- a. Setiap pelanggan hanya dapat dilayani oleh satu kendaraan

$$\sum_{k=1}^4 \sum_{j=0}^{22+1} x_{ij}^k = 1 ; i = 1, 2, \dots, 22$$

- b. Total permintaan tidak boleh melebihi kapasitas kendaraan

$$\sum_{i=0}^{22+1} \sum_{j=1}^{22} x_{ij}^k q_j \leq 210 ; k = 1, 2, 3, 4$$

- c. Setiap kendaraan berangkat dari depot ($N = 0$) dan setelah melayani pelanggan maka kendaraan akan meninggalkan tempat kemudian kendaraan kembali ke depot asal ($22 + 1$)

$$\sum_{j=0}^{22+1} x_{0j}^k = 1 ; k = 1, 2, 3, 4$$

$$\sum_{i=0}^{22+1} x_{ih}^k - \sum_{j=0}^{22+1} x_{hj}^k = 0 ; h = 1, 2, \dots, 22 ; k = 1, 2, 3, 4$$

$$\sum_{i=0}^{22+1} x_{i,22+1}^k = 1 ; k = 1, 2, 3, 4$$

- d. Variabel keputusan *integer biner*.

$$x_{ij}^k \in \{0,1\} ; i, j = 0, 1, \dots, 22 + 1 ; k = 1, 2, 3, 4$$

Hasil perhitungan:

- a. Algoritma IACS

$$z = \sum_{k=1}^4 \sum_{i=0}^{22+1} \sum_{j=0}^{22+1} c_{ij} x_{ij}^k = 362 \text{ km}$$

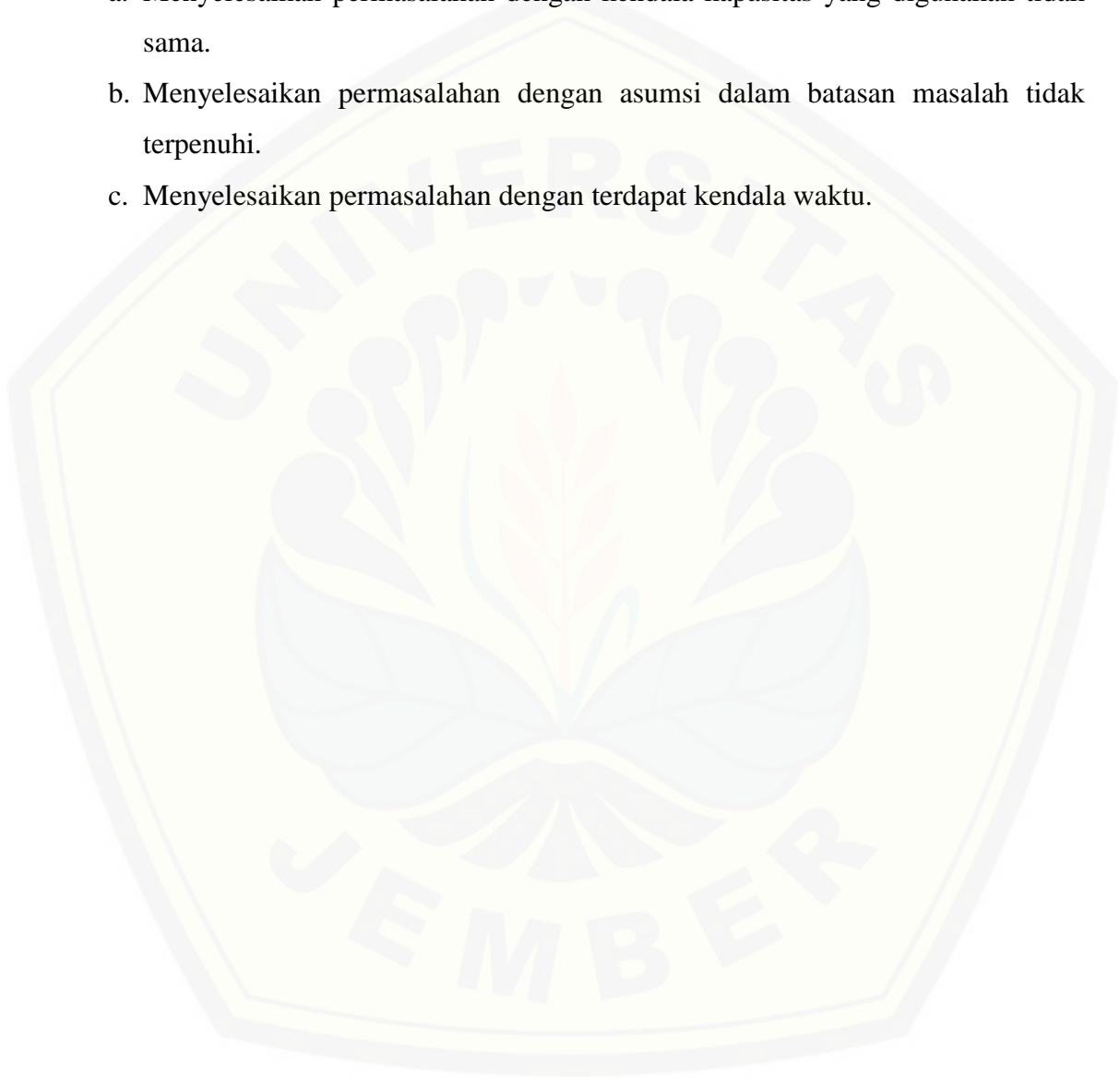
- b. Algoritma HACS

$$z = \sum_{k=1}^4 \sum_{i=0}^{22+1} \sum_{j=0}^{22+1} c_{ij} x_{ij}^k = 369 \text{ km}$$

5.2 Saran

Berdasarkan dari penelitian yang sudah dilakukan, penulis menyarankan agar algoritma IACS diterapkan pada pengembangan permasalahan CVRP seperti berikut:

- a. Menyelesaikan permasalahan dengan kendala kapasitas yang digunakan tidak sama.
- b. Menyelesaikan permasalahan dengan asumsi dalam batasan masalah tidak terpenuhi.
- c. Menyelesaikan permasalahan dengan terdapat kendala waktu.



DAFTAR PUSTAKA

- Afrianita, S. 2011. Algoritma Multiple Ant Colony System pada Vehicle Routing Problem with Time Windows. <http://lib.ui.ac.id>. [Diakses pada 03 Desember 2016].
- Bauer, A., B. Bullnheimer, R. F. Hartl, & C. Strauss. 1999. An Ant Colony Optimization Approach for the Single Machine Total Tardiness Problem. Departement of Management Science University of Vienna. *IEEE*. 1445-1450.
- Chen, C.H., & C. J. Ting. 2006. An Improved Ant Colony Systems Algorithm for The Vehicle Routing Problem. *Journal of the Chinese Institute of Industrial Engineers*. 23(2): 115-126.
- Dorigo, M., dan L. M. Gambardella. 1997. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*. 1(1): 53-66.
- Gunawan., I. Maryati, & H. K. Wibowo. 2012. Optimasi Penentuan Rute Kendaraan pada Sistem Distribusi Barang dengan Ant Colony Optimization. *Seminar Nasional Teknologi Informasi dan Komunikasi Terapan (Seminar)*. 1-6.
- Hasanah, S., & S. Wahyuningsih. 2013. Kinerja Algoritma Improved Ant Colony System (IACS) dalam Menyelesaikan Vehicle Routing Problem with Time Windows (VRPTW). <http://jurnal-online.um.ac.id/data/artikel/artikel6A7D7312745D17ED8FF16268129C76C4.pdf> . [Diakses pada 10 November 2016].
- Kallehauge, B., J. Larsen, & O.B.G. Madsen. 2001. Lagrangean Duality Applied on Vehicle Routing with Time Windows. *Informatics and Mathematical Modelling (IMM)*.
- Munir, R. 2005. Matematika Diskrit, Revisi 5. Bandung: Informatika.
- Pinter, J. D. 1995. Global Optimization. Canada: PCS Inc.

- Pop, P. C., C. P. Sitar, I. Zelina, V. Lupse, & C. Chira. 2011. Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem. *Int. J. of Computers, Communications and Control*. 6(1):158-165.
- Putri, R. A. 2014. Penyelesaian Capacitated Vehicle Routing Problem dengan Metode Savings Heuristic. http://repository.unej.ac.id/bitstream/handle/123456789/59970/Risqie%20Anisa%20Putri%20-%200818101014_1.pdf?sequence=1. [Diakses pada 07 Desember 2016].
- Tarigan, D. 2008. Pemodelan Vehicle Routing Problem Terbuka dengan Keterbatasan Waktu. <http://repository.usu.ac.id/bitstream/123456789/6052/1/067021013.PDF>. [Diakses pada 12 Desember 2016].
- Ulyawati, I. 2016. Penyelesaian Capacitated Vehicle Routing Problem dengan Algoritma Harmony Search dan Tabu Search. <http://repository.unej.ac.id/bitstream/handle/123456789/75929/Ikfi%20Ulyawati%20-%20121810101023%20-1.pdf?sequence=1>. [Diakses pada 27 Januari 2017].
- Xiang-pei, H. U, D. Qiu-lei, L. Yoang-xian, & S. Dan. 2006. An Improved Ant Colony System and Its Application. *IEEE*. 384-389.

LAMPIRAN

A. Data Jarak 22 Pelanggan di Setiap Kecamatan Kabupaten Jember

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	50	45	40	34	29	35	23	29	16	7	19	13	11	18	22	34	27	35	44	14	13	3
1	50	0	7	10	15	20	15	21	33	54	42	30	27	38	49	65	73	55	69	75	46	43	35
2	45	7	0	15	10	15	21	26	40	59	47	25	32	33	54	60	65	60	74	80	50	50	40
3	40	10	15	0	8	14	25	12	25	45	43	23	33	31	50	55	60	57	70	75	46	44	42
4	34	15	10	8	0	6	28	35	26	44	37	15	27	23	44	50	60	50	64	70	40	38	36
5	29	20	15	14	6	0	33	30	31	39	32	10	22	18	39	45	55	44	59	65	35	33	25
6	35	15	21	25	28	33	0	16	17	49	28	36	28	28	51	57	66	57	70	77	42	47	38
7	23	21	26	12	35	30	16	0	13	32	12	20	16	12	33	39	48	39	53	57	29	27	26
8	29	33	40	25	26	31	17	13	0	32	11	21	19	25	34	44	49	40	54	60	30	31	21
9	16	54	59	45	44	39	49	33	32	0	21	35	23	25	28	12	17	17	26	32	20	14	14
10	7	42	47	43	37	32	28	12	11	21	0	22	16	14	21	27	37	27	41	47	17	20	10
11	19	30	25	23	15	10	36	20	21	35	22	0	13	8	29	35	45	35	49	55	26	23	15
12	13	27	32	33	27	22	28	16	19	23	16	13	0	4	23	29	39	29	43	49	12	17	9
13	11	38	33	31	23	18	28	12	25	25	14	8	4	0	21	27	37	27	41	47	16	15	14
14	18	49	54	50	44	39	51	33	34	28	21	29	23	21	0	6	15	10	20	25	14	6	14
15	22	65	60	55	50	45	57	39	44	12	27	35	29	27	6	0	5	6	14	20	20	12	20

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
16	34	73	65	60	60	55	66	48	49	17	37	45	39	37	15	5	0	10	10	25	30	27	30
17	27	55	60	57	50	44	57	39	40	17	27	39	20	27	10	6	10	0	20	25	10	18	20
18	35	69	74	70	64	59	70	53	54	26	41	49	43	41	20	14	10	20	0	7	25	32	34
19	44	75	80	75	70	65	77	57	60	32	47	55	49	47	25	20	25	25	7	0	30	38	40
20	14	45	50	46	40	35	42	29	30	20	17	26	12	16	14	20	30	10	25	30	0	6	10
21	13	43	50	44	38	33	47	27	31	14	20	23	17	15	6	12	27	18	32	38	6	0	10
22	3	35	40	42	36	25	38	26	21	14	10	15	9	14	14	20	30	20	34	40	10	10	0

B. Data Jumlah Permintaan Pelanggan

Pelanggan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Permintaan (karton)	23	26	42	32	37	30	30	21	26	40	23	25	27	33	32	34	38	40	24	43	49	50

Keterangan :

- | | |
|--|---|
| 0 = Depot di Kecamatan Ajung | Pelanggan 5 = Pelanggan di Kecamatan Tanggul |
| Pelanggan 1 = Pelanggan di Kecamatan Gumukmas | Pelanggan 6 = Pelanggan di Kecamatan Puger |
| Pelanggan 2 = Pelanggan di Kecamatan Kencong | Pelanggan 7 = Pelanggan di Kecamatan Balung |
| Pelanggan 3 = Pelanggan di Kecamatan Umbulsari | Pelanggan 8 = Pelanggan di Kecamatan Ambulu |
| Pelanggan 4 = Pelanggan di Kecamatan Semboro | Pelanggan 9 = Pelanggan di Kecamatan Mumbulsari |

Pelanggan 10 = Pelanggan di Kecamatan Jenggawah

Pelanggan 11 = Pelanggan di Kecamatan Bangsalsari

Pelanggan 12 = Pelanggan di Kecamatan Panti

Pelanggan 13 = Pelanggan di Kecamatan Rambipuji

Pelanggan 14 = Pelanggan di Kecamatan Pakusari

Pelanggan 15 = Pelanggan di Kecamatan Mayang

Pelanggan 16 = Pelanggan di Kecamatan Silo

Pelanggan 17 = Pelanggan di Kecamatan Kalisat

Pelanggan 18 = Pelanggan di Kecamatan Ledokombo

Pelanggan 19 = Pelanggan di Kecamatan Sumber Jambe

Pelanggan 20 = Pelanggan di Kecamatan Patrang

Pelanggan 21 = Pelanggan di Kecamatan Sumber Sari

Pelanggan 22 = Pelanggan di Kecamatan Kaliwates

C. Pseudocode IACS

```

Inisialisasi parameter (Semut,  $\tau_{ij}$ ,  $\beta$ ,  $\rho$ ,  $\gamma$ )
Bangkitkan solusi awal menggunakan Nearest Neighbor heuristic (S)
FOR i from 1 to replika do
    |
     $S_1 = \text{localsearch}(S)$ 
    Evaluasi  $S_1$ 
END FOR
Inisialisasi  $\eta_{ij}$ 
FOR i from 1 to Iterasi Max do
    |
    FOR j from 1 to Semut Max do
        |
        IF  $q > q_0$  then
             $P_{ij}^k = \frac{(\tau_{ij})(\eta_{ij})^\beta}{\sum_{j \in \text{UK}} (\tau_{ij})(\eta_{ij})^\beta}$ 
             $S_j = \text{Probabilitas Kumulatif (P)}$ 
        ELSE
             $P_{ij}^k = \max_{j \in \text{UK}} [(\tau_{ij})(\eta_{ij})^\beta]$ 
             $S_j = \text{Maksimum (P)}$ 
        END IF
         $\tau_{ij}^{\text{new}} = \tau_{ij}^{\text{old}} + \rho \tau_0$ 
         $\tau_0 = \frac{1}{L_{mn} \cdot n}$ 
    END FOR
    Pilih Solusi Terbaik  $S_j$  dari tiap semut dan simpan solusi di
    Tabu List  $S_i$ 
    FOR z from 1 to replika do
        |
         $S_j = \text{localsearch}(S_i)$ 
        Evaluasi  $S_j$ 
    END FOR
    Urutkan Solusi  $S_j$ 
     $\tau_{ij}^{\text{new}} = (1-\gamma) \tau_{ij}^{\text{old}} + \gamma \Delta \tau_{ij}$ 
     $\tau_{ij} = \frac{(L_3 - L_g) + (L_3 - L_1)}{L_3}$ 
END FOR

```

D. Pseudocode HACS

```

Inisialisasi parameter (Semut,  $\tau_{ij}$ ,  $\beta$ ,  $\rho$ ,  $\gamma$ ,  $\alpha$ )
Bangkitkan solusi awal menggunakan Nearest Neighbor heuristic (S)
Inisialisasi  $\eta_{ij}$ 
FOR i from 1 to Iterasi Max do
  FOR j from 1 to Semut Max do
     $J = \max_{j \in U_k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\mu_{ij}]^\gamma$ 
    IF  $q \leq pt$  then
       $S_j = \text{Maksimum} ( J )$ 
    ELSE
       $S_j = \text{Random} ( J )$ 
    END IF
  END FOR
  Pilih Solusi Terbaik  $S_j$  dari tiap semut dan simpan solusi di Tabu List  $S_j$ 
  FOR z from 1 to replika do
     $S_j = \text{interchange} ( S_j )$ 
    Evaluasi  $S_j$ 
  END FOR
   $\Delta \tau_{ij} = \rho \tau_{ij}^{old} + \Delta \tau_{ij}$ 
   $\Delta \tau_{ij} = \begin{cases} \frac{1}{L_{lokal}} & , ij \text{ berdasarkan } L_{lokal} \\ 0 & , \text{lainnya} \end{cases}$ 
  IF  $0.95 * pt \geq pmin$  or  $0.95 * \rho \geq pmin$  then
     $pt = 0.95 * pt$ 
     $\rho = 0.95 * \rho$ 
  ELSE
     $pt = pmin$ 
     $\rho = pmin$ 
  END IF
END FOR

```
