



**APLIKASI ALGORITMA *CHEAPEST INSERTION HEURISTIC* (CIH)
WITH SAVINGS METHOD DALAM KASUS *VEHICLE ROUTING
PROBLEM WITH TIME WINDOWS* (VRPTW)
PADA PENGANGKUTAN SAMPAH**

SKRIPSI

Oleh
Masrifan Dwi Yansyah
NIM 121810101076

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



**APLIKASI ALGORITMA CHEAPEST INSERTION HEURISTIC (CIH)
WITH SAVINGS METHOD DALAM KASUS VEHICLE ROUTING
PROBLEM WITH TIME WINDOWS (VRPTW)
PADA PENGANGKUTAN SAMPAH**

SKRIPSI

diajukan guna melengkapi tugas akhir matematika dan memenuhi salah satu syarat untuk menyelesaikan program studi matematika (S1) dalam gelar sarjana sains

Oleh
Masrifan Dwi Yansyah
NIM 121810101076

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

PERSEMBAHAN

Dengan segala kerendahan hati dan puji syukur yang tak terhingga pada Allah SWT, skripsi ini saya persembahkan untuk:

1. Ibunda Dwi Anjar Bintariringtyas, Ayahanda Zainal Arifin Soefyan, Kakak Fian Fajri Mulaika Tiska, serta adik Firtanti Mustika Harum Sari yang telah memberi dukungan, doa, dan pengorbanan selama ini;
2. Guru-guruku sejak taman kanak-kanak sampai dengan Perguruan Tinggi yang terhormat, yang telah memberikan ilmu dan membimbing dengan penuh keasabaran;
3. Teman-teman seperjuangan angkatan 2012 (Bathics 12) yang telah menemanini selama kurang lebih 4 tahun ini;
4. Almamater Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

Karena itu, ingatlah kepada-Ku niscaya Aku ingat (pula) kepadamu,
dan bersyukurlah kepada-Ku, dan janganlah
kamu mengingkari (nikmat)-Ku
(Q.S Al-Baqarah: 152)*

Harga kebaikan manusia adalah diukur menurut apa yang telah
dilaksanakan/diperbuatnya
(Ali Bin Abi Thalib)

*) Departemen Agama Republik Indonesia 1998. *Al Qur'an dan Terjemahannya*, Semarang: PT Kumudasmoro Grafindo.

PERNYATAAN

Saya yang beranta tangan di bawah ini:

Nama : Marifan Dwi Yansyah

NIM : 1218101010876

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: “Aplikasi Algoritma *Cheapest Insertion Heuristic (CIH) With Savings Method* dalam Kasus *Vehicle Routing Problem With Time Windows (VRPTW)* pada Pengangkutan Sampah” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, Juni 2016

Yang menyatakan,

Masrifan Dwi Yansyah

NIM 121810101076

SKRIPSI

**APLIKASI ALGORITMA *CHEAPEST INSERTION HEURISTIC*
DALAM KASUS *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS*
PADA PENGANGKUTAN SAMPAH**

Oleh
Masrifan Dwi Yansyah
NIM 121810101076

Pembimbing

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

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

PENGESAHAN

Skripsi berjudul “Aplikasi Algoritma *Cheapest Insertion Heuristic (CIH) With Savings Method* dalam Kasus *Vehicle Routing Problem With Time Windows (VRPTW)* pada Pengangkutan Sampah” telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji

Ketua,

Sekretaris,

Ahmad Kamsyakawuni, S.Si., M.kom
NIP. 197211291998021001
Anggota I,

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

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

Dr. Alfian Futuhul Hadi, S.Si., M.Si
NIP. 197407192000121001

Mengesahkan
Dekan,

Drs. Sujito, PhD
NIP. 196102041987111001

RINGKASAN

Aplikasi Algoritma *Cheapest Insertion Heuristic (CIH) With Savings Method* dalam Kasus *Vehicle Routing Problem With Time Windows (VRPTW)* pada Pengangkutan Sampah; Masrifan Dwi Yansyah, 121810101076; 2016; 57 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Permasalahan pengangkutan sampah adalah contoh lain pendistribusian suatu barang dari suatu depot ke pelanggan. Jika masalah pendistribusian biasanya adalah mengirimkan barang pada pelanggan, namun berbeda dengan sistem pengangkutan sampah. Sistem pengangkutan sampah adalah pengambilan barang (sampah) di pelanggan (TPS). Permasalahan pengangkutan sampah termasuk salah satu kasus *Vehicle Routing Problem With Time Windows (VRPTW)*. Untuk menyelesaikan kasus VRPTW pada pengangkutan sampah, dalam skripsi ini menggunakan dua algoritma yaitu algoritma *Cheapest Insertion Heuristic (CIH)* dan *Cheapest Insertion Heuristic with Savings Method*.

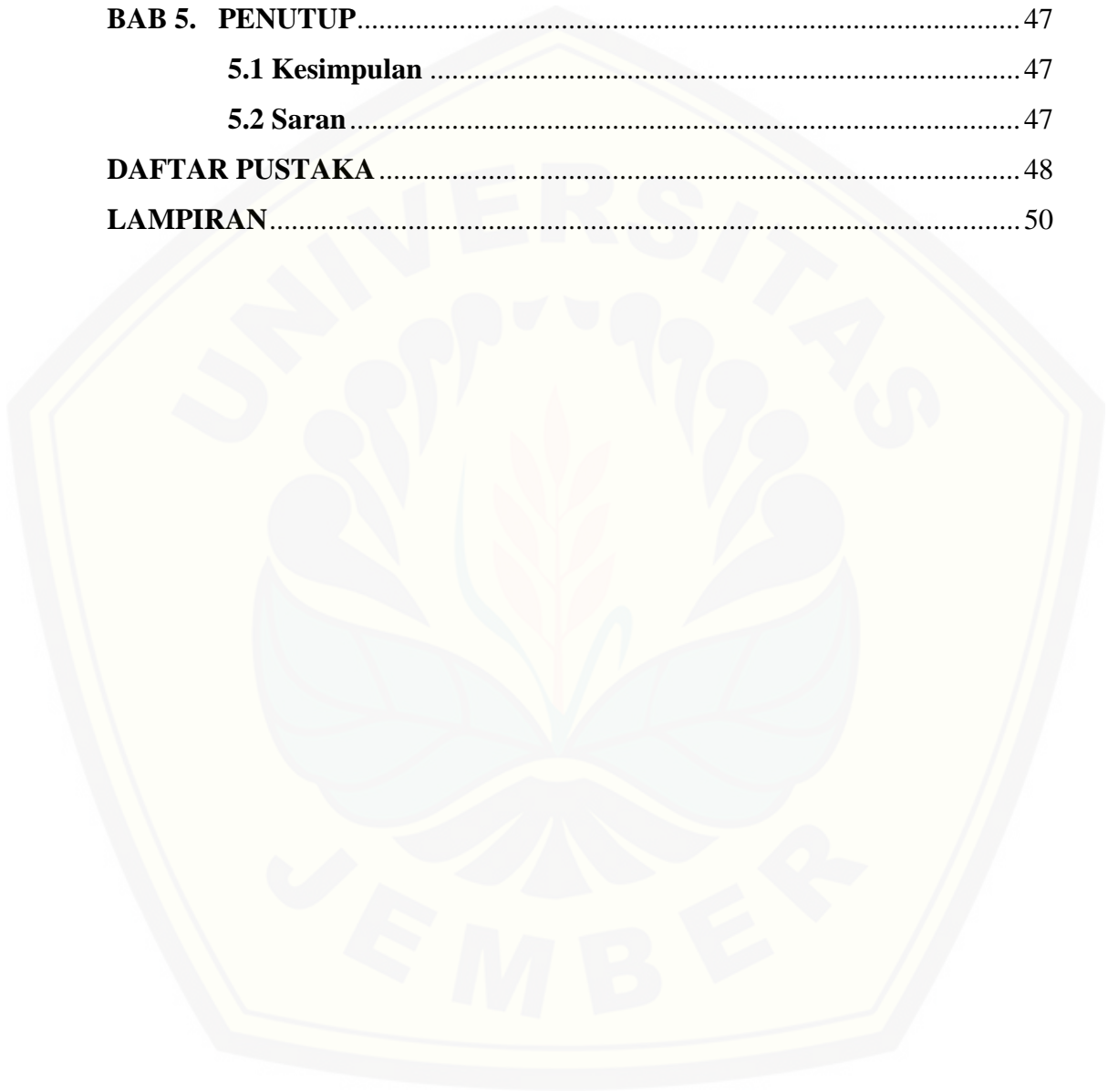
Data yang digunakan dalam penelitian ini adalah data sekunder dari skripsi Sakinah (2012). Dalam skripsi tersebut dijelaskan bahwa proses pengangkutan sampah yang dilakukan Dinas Kebersihan dan Pertamanan Kota Malang selama 8 jam yang dimulai dari pukul 06.00 sampai 14.00. Dengan kapasitas kendaraan sebesar $8.000 m^3$. Untuk kecepatan rata-rata kendaraan 3 km/jam dan waktu pelayanan pengangkutan sampah setiap $1 m^3$ di TPS adalah 3 detik.

Algoritma yang digunakan untuk menyelesaikan permasalahan ini adalah *Cheapest Insertion Heuristic (CIH)* dan *Cheapest Insertion Heuristic (CIH) with savings method* juga dengan bantuan program MATLAB untuk meminimalkan jarak dan waktu pengangkutan sampah. Berdasarkan rute optimal yang dihasilkan oleh program yang dibuat dengan bantuan MATLAB, algoritma CIH mengahailkan rute yang lebih baik dengan total jarak keseluruhan yang lebih minimal dan waktu yang dibutuhkan dalam proses pengangkutan sampah lebih baik dibandingkan dengan algoritma CIH *with savings method*.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBINGAN.....	v
HALAMAN PENGESAHAN.....	vi
RINGKASAN	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Teori Graf.....	5
2.2 <i>Vehicle Routing Problem (VRP)</i>.....	6
2.3 <i>Vehicle Routing Problem With Time Windows (VRPTW)</i>.....	8
2.4 <i>Cheapest Insertion Heuristic (CIH)</i>.....	12
2.5 Metode Penghematan (<i>Savings Method</i>).....	13
BAB 3. METODOLOGI	16
3.1 Data Penelitian	16
3.2 Langkah-langkah Penelitian	17
3.3 Flowchart Penelitian	20
BAB 4. HASIL DAN PEMBAHSAN.....	21

4.1 Hasil	21
4.1.1 Penyelesaian Manual.....	21
4.1.2 Penyelesain Dengan Program	39
4.2 Pembahasan	45
BAB 5. PENUTUP	47
5.1 Kesimpulan	47
5.2 Saran	47
DAFTAR PUSTAKA	48
LAMPIRAN	50



DAFTAR GAMBAR

	Halaman
2.1 Contoh graf tertutup	5
2.2 Contoh graf tertutup dengan bobot sisi	8
2.3 Contoh graf kasus VRPTW	9
2.4 <i>Time windows</i>	10
2.5 Solusi kasus VRPTW	11
2.7 Ilustrasi konsep penghematan	14
4.1 Tampilan program utama	39
4.2 Tampilan panel inputan.....	40
4.3 Tampilan input data pada tabel	40
4.4 Tampilan panel proses.....	41
4.5 Tampilan panel output	42
4.6 Tampilan rute optimal (a).....	42
4.7 Tampilan rute optimal (b)	43
4.8 Tampilan total jarak	43
4.9 Tampilan total waktu.....	43
4.10 Tampilan total kapasitas.....	44
4.11 Tampilan rute optimal CIH dalam graf	44
4.12 Tampilan rute optimal CIH <i>with savings method</i> dalam graf	44

DAFTAR TABEL

	Halaman
2.1 Matriks ketetanggaan	6
2.2 Matriks bersisian	6
3.1 TPS-TPS yang dilalui untuk pengangkutan sampah.....	20
3.2 Volume sampah yang harus diangkat disetiap TPS	21
3.3 Jarak TPA ke TPS dan antar TPS	21
4.1 Penyisipan k terhadap <i>subtour</i> 2 (CIH).....	23
4.2 Penyisipan k terhadap <i>subtour</i> 3 (CIH).....	24
4.3 Penyisipan k terhadap <i>subtour</i> 4 (CIH).....	25
4.4 Penyisipan k terhadap <i>subtour</i> 5 (CIH).....	26
4.5 Penyisipan k terhadap <i>subtour</i> 6 (CIH).....	27
4.6 Nilai <i>saving</i>	31
4.7 Penyisipan k terhadap <i>subtour</i> 2 (CIH with <i>savings method</i>).....	33
4.8 Penyisipan k terhadap <i>subtour</i> 3 (CIH with <i>savings method</i>).....	34
4.9 Penyisipan k terhadap <i>subtour</i> 4 (CIH with <i>savings method</i>).....	35
4.10 Penyisipan k terhadap <i>subtour</i> 5 (CIH with <i>savings method</i>).....	37

DAFTAR LAMPIRAN

	Halaman
A. Tampilan skrip program fungsi tujuan	50
B. Tampilan skrip program fungsi kendala kapasitas dan kendala waktu	50
C. Tampilan skrip program fungsi kendala kendaraan berawal dan berakhir pada TPA	51
D. Data jarak 33 titik	52
E. Data kapasitas 33 titik	55
F. Output dari algoritma CIH dengan data 8 titik	56
G. Output dari algoritma CIH <i>with savings method</i> dengan data 8 titik	56
H. Output dari algoritma <i>savings method</i> dengan data 8 titik	56
I. Output dari algoritma CIH dengan data 33 titik	57
J. Output dari algoritma CIH <i>savings method</i> dengan data33 titik	57

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pesatnya pembangunan di wilayah perkotaan di Indonesia dengan diikuti pula pesatnya perkembangan jumlah penduduk di wilayah kota. Hal ini merupakan salah satu pemicu banyaknya limbah yang dihasilkan oleh penduduk kota. Penumpukan sampah di beberapa titik di kota tidak disertai dengan penyediaan sarana dan prasarana yang sebanding oleh pemerintah, akibatnya pelayanan tidak maksimal dan terjadi penurunan kualitas lingkungan. Untuk menanggulangi permasalahan ini, peranan pemerintah dan kepedulian masyarakat setempat sangat dibutuhkan.

Pengelolaan sampah harus semakin diperhatikan karena selain berhubungan dengan pengendalian kualitas lingkungan yang bersih juga berhubungan dengan efisiensi biaya. Transportasi sampah adalah sub-sistem Dinas Kebersihan yang bertugas membawa sampah dari lokasi Tempat Pembuangan Sampah (TPS) ke Tempat Pembuangan Akhir (TPA). Dengan optimasi sub-sistem ini diharapkan pengangkutan sampah menjadi lebih efektif dan efisien dengan tujuan akhir meminimalkan penumpukan sampah yang akan memberi dampak langsung bagi kesehatan lingkungan masyarakat dan keindahan kota. Hal ini merupakan salah satu permasalahan dalam kehidupan sehari-hari yang dapat dipecahkan dengan menggunakan matematika.

Permasalahan pengangkutan sampah ini adalah contoh lain pendistribusian suatu barang atau jasa dari suatu depot ke pelanggan. Jika masalah pendistribusian biasanya adalah mengirimkan barang pada pelanggan, namun berbeda dengan sistem pengangkutan sampah. Sistem pengangkutan sampah adalah pengambilan barang (sampah) di pelanggan (TPS). Menurut Hasanah (2013), persoalan pendistribusian barang juga terdapat diberbagai bidang pelayanan umum, misalnya persoalan pengiriman barang, penjemputan penumpang bus, penentuan jalur pembersihan jalan dan penentuan trayek angkutan umum.

Pada suatu kasus pendistribusian barang, depot harus bisa melayani *customer* yang tersebar di seluruh wilayah, seringkali kendaraan harus menempuh perjalanan yang jauh dan tidak efisien. Pada saat mendistribusikan barang tersebut, kendaraan yang digunakan harus menempuh jarak yang minimum dan memenuhi waktu yang telah ditentukan. Maka dari itu minimasi jarak dan waktu tempuh merupakan solusi utama dari perencanaan rute pendistribusian barang maupun rute pengangkutan sampah, sehingga didapat rute pengangkutan yang paling optimal.

Dari gambaran permasalahan ini, sangat penting melakukan kajian lebih lanjut tentang upaya untuk mengoptimalkan proses pengangkutan sampah dengan satu kali putaran rute. Permasalahan ini merupakan salah satu model *Vehicle Routing Problem With Time Windows* (VRPTW) yang nantinya diaplikasikan dengan bantuan algoritma *Cheapest Insertion Heuristic*.

Merujuk dari skripsi Sakinah (2012) yang berkesimpulan Penggunaan pada kasus VRPTW Algoritma *Clark and Wright* dan *Nearest Insertion Heuristic* mampu mengoptimalkan proses pengangkutan sampah. Dengan satu kali putaran rute, dimana penumpukan sampah pada beberapa wilayah pelayanan dapat diminimalisir dengan menggunakan algoritma tersebut. Menurut Indah (2011) yang berkesimpulan bahwa rute optimal yang dihasilkan oleh program Delphi 7.0, algoritma *Cheapest Insertion Heuristic* (CIH) menghasilkan rute yang lebih optimal dibandingkan dengan algoritma semut untuk jumlah titik kurang dari 7. Sedangkan untuk titik lebih dari 7, algoritma semut menghasilkan rute yang lebih optimal dibandingkan dengan algoritma *Cheapest Insertion Heuristic* (CIH).

Merujuk dari skripsi di atas, penulis tertarik dengan perbandingan hasil proses dari algoritma yang penulis pakai yaitu *Cheapest Insertion Heuristic* dan penggabungan algoritma *Cheapest Insertion Heuristic* dengan *Savings Method* pada kasus *Vehicle Routing Problem With Time Windows* Pada Pengangkutan Sampah. Penggabungan antara algoritma CIH dan *savings method* dilakukan karena pemilihan subtour awal pada algoritma CIH tidak ada ketentuan khusus. Maka dari itu dengan menggabungkan dua algoritma di atas diharapkan dapat menghasilkan hasil yang lebih baik dibandingkan dengan algoritma CIH. Atas

dasar inilah penulis memilih judul: Aplikasi Algoritma *Cheapest Insertion Heuristic (CIH) With Savings Method* dalam Kasus *Vehicle Routing Problem With Time Windows (VRPTW)* pada Pengangkutan Sampah.

1.2 Perumusan Masalah

Adapun permasalahan yang timbul dari latar belakang terkait masalah pengangkutan sampah adalah sebagai berikut:

- a. Bagaimana membuat rute pengangkutan sampah yang efektif dan efisien pada setiap pelayanan yang dengan meminimalkan jarak dan waktu pengangkutan sampah?
- b. Bagaimana perbandingan hasil proses pengangkutan sampah menggunakan algoritma CIH dengan CIH *with Savings Method*?

1.3 Batasan Masalah

Agar pembahasan tidak menyimpang, maka perlu dibuat suatu batasan masalah sebagai berikut:

- a. Hanya menentukan rute yang optimal berdasarkan jarak dan waktu.
- b. Kecepatan rata-rata truk sampah disetiap tempat 30 km/jam.
- c. Keberangkatan truk sampah berawal dan berakhir di titik yang sama yaitu di TPA.
- d. Rata-rata volume sampah diasumsikan sama tiap harinya di setiap TPS.
- e. Jarak antara TPA ke TPS dan jarak antar TPS dianggap simetris (jarak antara tempat satu dengan yang lain dan sebaliknya adalah sama).

1.4 Tujuan

Berdasarkan perumusan masalah di atas, maka tujuan yang ingin dicapai pada pemodelan ini adalah:

- a. Mengoptimalkan proses pengangkutan sampah dengan satu kali putaran rute agar waktu kerja lebih efisien.
- b. Membuat rute pengangkutan sampah yang efektif dan efisien pada setiap wilayah pelayanan yang ada.

- c. Meminimalisasi jarak dan waktu pengangkutan sampah pada setiap wilayah pelayanan yang ada.

1.5 Manfaat

Berdasarkan perumusan masalah di atas, maka manfaat yang dapat diambil pada skripsi ini adalah:

- a. Bagi pemerintah kota, khususnya dinas kebersihan kota adalah sebagai alternatif solusi mengenai pengoptimalan rute pengangkutan sampah agar lebih efektif dan efisien.
- b. Bagi kalangan akademik, khususnya Jurusan Matematika dapat dijadikan salah satu referensi untuk memperluas pemahaman mengenai permasalahan distribusi atau pengangkutan sampah dengan menggunakan metode ini.
- c. Bagi penulis, untuk mengetahui lebih dalam mengenai sistem pengangkutan dan pengoptimalan penjadwalan serta rute yang efektif dan efisien dengan menggunakan metode penyelesaian *Vehicle Routing Problem Time Window* (VRPTW).

BAB 2. TINJAUAN PUSTAKA

2.1 Teori Graf

Teori graf adalah salah satu pokok bahasan dari ilmu matematika yang memiliki banyak penerapannya dan digunakan dalam kehidupan sehari-hari sampai saat ini. Menurut catatan sejarah konsep teori graf ini muncul pada saat meneliti dan memecahkan permasalahan pertama graf yang dihadapi ketika membangun jembatan Königsberg oleh karya seorang matematikawan dari Swiss yang bernama Leonhard Euler pada tahun 1736. Graf adalah pasangan antara himpunan tak kosong disebut dengan titik (*vertex*) dan himpunan sisi yang menghubungkan titik-titik tersebut dimana setelah dihubungkan akan membentuk sebuah pola tertentu. Kini graf semakin berkembang dan sangat berperan dalam berbagai sektor terutama sektor jaringan komunikasi, peta, optimasi jalur terpendek dan transportasi.

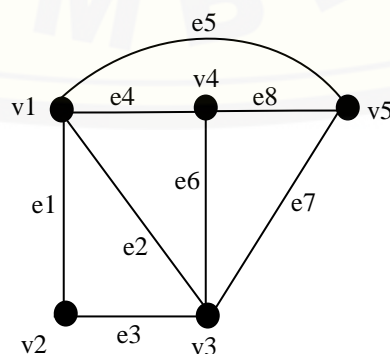
Terdapat beberapa cara merepresentasikan graf, dua diantaranya yang sering digunakan adalah ketetanggaan dan matriks bersisian.

a. Matriks Ketetanggaan

Matriks ketetanggaan didefinisikan sebagai berikut, misalkan $A = (a_{ij})$ adalah matriks $m \times m$ yang didefinisikan oleh,

$$a_{ij} = \begin{cases} 1, & \text{simpul } v_i \text{ bertetangga dengan simpul } v_j \\ 0, & \text{lainnya} \end{cases}$$

Perhatikan Gambar 2.1 menunjukkan graf tertutup yang terdiri dari 5 simpul dan 8 sisi serta Tabel 2.1 yang menunjukkan matriks ketetanggaannya.



Gambar 2.1 Contoh graf tertutup

Tabel 2.1 Matriks ketetanggan

	$v1$	$v2$	$v3$	$v4$	$v5$
$v1$	0	1	1	1	1
$v2$	1	0	1	0	0
$v3$	1	1	0	1	1
$v4$	1	0	1	0	1
$v5$	1	0	1	1	0

b. Matriks Bersisian

Matriks bersisian didefinisikan sebagai berikut, misalkan $B = (b_{ij})$ adalah matriks $m \times n$ yang didefinisikan oleh,

$$b_{ij} = \begin{cases} 1, & \text{simpul } v_i \text{ bersisian dengan sisi } e_j \\ 0, & \text{lainnya} \end{cases}$$

Perhatikan Gambar 2.1 menunjukkan graf berarah yang terdiri dari 5 simpul dan 8 sisi serta Tabel 2.1.2 yang menunjukkan matriks bersisiannya.

Tabel 2.2 Matriks bersisian

	$e1$	$e2$	$e3$	$e4$	$e5$	$e6$	$e7$	$e8$
$v1$	1	1	0	1	1	0	0	0
$v2$	1	0	1	0	0	0	0	0
$v3$	0	1	1	0	0	1	1	0
$v4$	0	0	0	1	0	1	0	1
$v5$	0	0	0	0	1	0	1	1

Teori graf dapat memecahkan masalah meminimalisasi jarak dan jumlah kendaraan dalam suatu pendistribusian barang. Teori graf merupakan salah satu cabang matematika yang penting dan banyak manfaat karena teori-teorinya dapat diterapkan dalam kehidupan sehari-hari. Suatu permasalahan dapat dilihat dan diamati dengan jelas dengan menggunakan rumusan atau teori graf yang tepat, sehingga mudah untuk dianalisis.

2.2 Vehicle Routing Problem (VRP)

Pendeskripsian VRP merupakan permasalahan dalam menentukan sejumlah rute untuk sekumpulan kendaraan identik yang harus melayani sejumlah *customer* dari depot pusat. Tujuan dari permasalahan VRP adalah untuk melayani

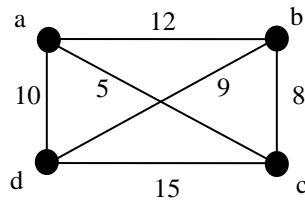
customer sesuai dengan permintaan dengan meminimalkan biaya angkut dan jumlah kendaraan yang dimulai dan berakhir di depot pusat.

Vehicle Routing Problem (VRP) adalah suatu model yang memiliki banyak varian yang menggambarkan masalah transportasi sebagai model graf. Tujuan dari VRP adalah untuk menemukan rute dengan biaya minimum untuk pengiriman suatu produk kepada sejumlah *customer* di beberapa lokasi yang berbeda dengan menggunakan beberapa kendaraan. Setelah diperkenalkan pada tahun 1959 oleh Dantzig dan Ramser melalui makalah mereka yang berjudul "*The Truck Dispatching Problem*", telah banyak metode atau algoritma yang dipakai atau diadaptasi untuk memecahkan VRP dan varian-variannya. Yang membuat VRP menarik untuk dibahas adalah model masalah yang metodenya dapat diimplementasikan ke aplikasi yang sangat fungsional di dunia nyata.

Vehicle routing problem (VRP) termasuk persoalan yang diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota. Deskripsi persoalannya adalah sebagai berikut: diberikan sejumlah dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seseorang pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan.

Kota dapat dinyatakan sebagai titik (*vertex*), sedangkan sisi (*edge*) menyatakan jalan yang menghubungkan antar dua kota. Bobot pada sisi menyatakan jarak antara dua kota. Dalam kasus ini jarak antar kota dianggap simetris, yaitu jarak dari kota 1 ke kota 2 adalah sama dengan jarak dari kota 2 ke kota 1 dan graf yang direpresentasikan sebagai permasalahannya merupakan graf yang terhubung secara penuh artinya pada setiap simpul yang ada pasti terhubung dengan simpul yang lain.

Persoalan perjalanan pedagang tidak lain menentukan sirkuit Hamilton yang memiliki bobot minimum pada sebuah graf terhubung. Pada persoalan ini, jika setiap simpul mempunyai sisi ke simpul yang lain. Pada sebarang graf lengkap dengan n buah simpul ($n > 2$), jumlah sirkuit Hamilton yang berbeda adalah $(n-1)!/2$. Contoh kasus sebagai berikut :



Gambar 2.2 Contoh graf tertutup dengan bobot sisi

Graf lengkap dengan $n = 4$ simpul seperti yang ditunjukkan pada Gambar 2.2.

Graf tersebut memiliki $(4-1)!/2 = 3$ sirkuit Hamilton, yaitu:

$$I_1 = (a, b, c, d, a) \text{ atau } (a, d, c, b, a) \implies \text{panjang rute} \\ = 10 + 12 + 8 + 15 = 45$$

$$I_2 = (a, c, d, b, a) \text{ atau } (a, b, d, c, a) \implies \text{panjang rute} \\ = 12 + 5 + 9 + 15 = 41$$

$$I_3 = (a, c, b, d, a) \text{ atau } (a, d, b, c, a) \implies \text{panjang rute} \\ = 10 + 5 + 9 + 8 = 32$$

Jadi sirkuit Hamilton terpendek adalah $I_3 = (a, c, b, d, a)$ atau (a, d, b, c, a) dengan panjang sirkuit $= 10 + 5 + 9 + 8 = 32$.

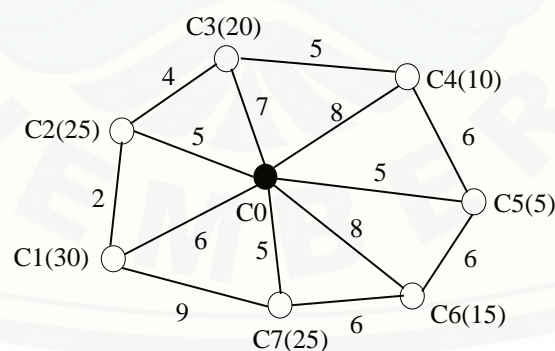
2.3 Vehicle Routing Problem With Time Windows (VRPTW)

Vehicle Routing Problem With Time Windows (VRPTW) merupakan salah satu variasi dari *Vehicle Routing Problem* (VRP) dengan mempertimbangkan batasan waktu yang ada, baik dari *customer* maupun dari perusahaan sendiri. Selain batasan waktu, batasan yang perlu diperhatikan adalah batasan kapasitas armada. Model VRPTW kemudian disesuaikan dengan permasalahan pada perusahaan sehingga menjadi metode optimasi yang sesuai dengan kondisi perusahaan. Armada juga tidak boleh melanggar batasan waktu yang telah ditetapkan. Rute yang terbentuk pada persoalan VRPTW berawal dan berakhir pada depot yang sama dengan jalur sub rute yang berbeda untuk masing-masing kendaraan.

VRPTW masuk dalam kategori persoalan optimasi, teknik lain selain metode eksak untuk pencarian solusinya bisa menggunakan prosedur pencarian *heuristic*. Metode *heuristic* memberikan suatu cara untuk menyelesaikan

permasalahan optimasi yang lebih sulit dan dengan kualitas dan waktu penyelesaian yang lebih cepat daripada solusi eksak. Pada skripsi ini juga dikembangkan prosedur algoritma *heuristic* algoritma *Cheapest Insertion Heuristic* (CIH) untuk mendapatkan solusi yang mendekati optimum dan waktu pengambilan keputusan (*run time*) yang cepat.

Vehicle Routing Problem with Time Windows adalah turunan dari *Capacitated Vehicle Routing Problem* dengan tambahan properti *time windows* yang diasosiasikan pada tiap-tiap *customer* dan *central depot*. *Time windows* yang diasosiasikan pada tiap *customer* adalah *range* waktu dimana *customer* dapat menerima barang kiriman. *Time windows* yang diasosiasikan pada *central depot* adalah *range* waktu antara semua kendaraan memulai rute dan waktu paling lambat untuk mengakhiri rute. *Time windows* terdiri dari waktu buka dan waktu tutup *customer* dan *central depot*. Jika suatu *customer* buka pukul 7-10 pagi, maka kendaraan harus sampai ke *customer* tersebut sebelum jam 10 pagi. Jika kendaraan tersebut sampai sebelum pukul 7 pagi, maka kendaraan tersebut harus menunggu sampai waktu itu untuk mulai melayani *customer*. Pada varian ini juga diperkenalkan istilah *service time* (waktu pelayanan), yang merepresentasikan waktu yang dibutuhkan untuk melayani suatu *customer*. *Service time* dapat berupa waktu yang dibutuhkan untuk menurunkan barang dari suatu kendaraan.

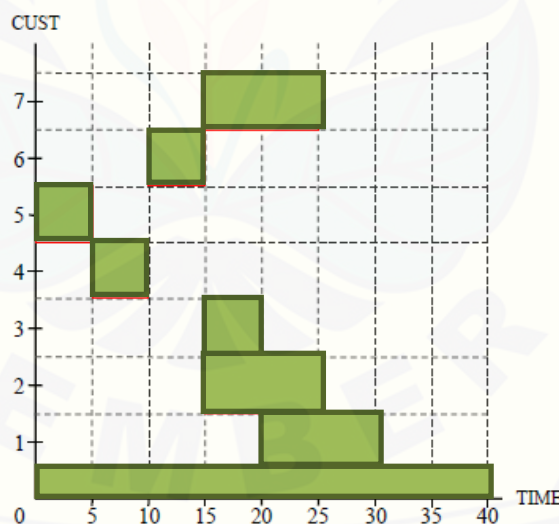


Gambar 2.3 Contoh graf kasus VRPTW

Time windows yang dijabarkan diatas merupakan *single time windows* (satu interval). Pada kasus lain, terdapat kemungkinan bahwa interval lebih dari satu, yang biasa disebut *multiple time windows*. Contoh dari kasus ini adalah sebuah toko yang buka pada jam 7-11 pagi (interval pertama) dan jam 3-6 sore

(interval kedua). Jika setiap kendaraan “harus” sampai di suatu *customer* sebelum *customer* itu tutup, maka *time windows* yang dipakai tergolong sebagai *hard*. Jika kendaraan “boleh” sampai setelah *customer* tutup, tetapi harus membayar sejumlah denda/penalti, maka *time windows* tergolong sebagai *soft*.

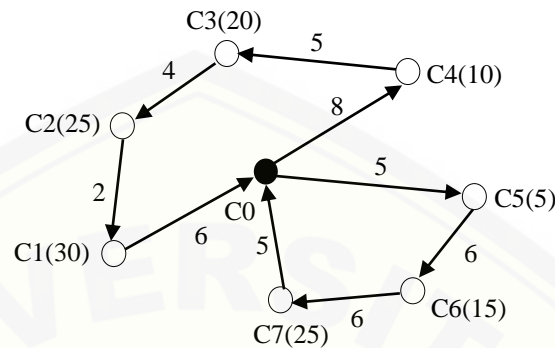
Diberikan contoh kasus sederhana yang representasi grafnya dapat dilihat pada gambar 2.3. Terdapat 7 *customer* yang harus dilayani. Kendaraan yang tersedia di *central depot* adalah 2. Kapasitas maksimum tiap kendaraan adalah 10 satuan. Titik-titik (*vertices*) merepresentasikan *central depot* dan semua *customer*. C0 merepresentasikan *central depot*, sedangkan C1-C7 merepresentasikan *customer* yang dilayani. Angka yang berada di *vertex* merepresentasikan akhir dari *time windows* (*end*). *Edge weight* (angka yang berada di tiap-tiap *edge*) merepresentasikan biaya perjalanan yang berupa waktu perjalanan antara 2 lokasi yang dihubungkan oleh *edge* tersebut dalam suatu satuan waktu. Berikut ini diberikan grafik rentang *time windows* masing-masing *customer* dan *central depot*.



Gambar 2.4 Time windows

Garis horizontal pada grafik di gambar 2.4 menunjukkan garis waktu. Sedangkan garis vertikal menunjukkan indeks *customer*, mulai dari *central depot* (0) sampai dengan *customer* ke 7. Batang horizontal yang berwarna gelap muda pada grafik tersebut menunjukkan rentang waktu antara waktu buka dan tutup para *customer* dan *central depot*. Waktu awal keberangkatan kendaraan adalah

waktu awal *time windows* dari *central depot*, yaitu 0. Untuk representasi graf dari contoh kasus yang diberikan dapat dilihat pada gambar 2.5



Gambar 2.5 Solusi kasus VRPTW

Dari permasalahan yang telah dijabarkan, solusi yang dicari adalah rute optimal atau rute dengan total waktu perjalanan minimal untuk dua kendaraan yang berangkat dari *central depot* untuk melayani 7 *customer* yang masing-masing memiliki *time windows*. Rute dari kendaraan pertama direpresentasikan oleh *edges* yang menuju *vertex* C4-C3-C2-C1, sedangkan rute dari kendaraan kedua direpresentasikan oleh *edges* yang menuju *vertex* C5-C6-C7. Untuk detail rute dapat dilihat dibawah ini.

Rute #1 : C4 – C3 – C2 – C1

Rute #2 : C5 – C6 – C7

Cost 47

Cara menghitung total biaya adalah dengan menjumlahkan semua *edge weight* (biaya perjalanan) yang merupakan bagian dari rute kendaraan. Kendaraan pertama menempuh $8+5+4+2+6=25$, dan kendaraan kedua menempuh $5+6+6+5=22$. Total biaya adalah $25+22=47$ satuan.

Vehicle Routing Problem with Time Windows (VRPTW) memiliki formulasi model yang digunakan untuk merumuskan fungsi tujuan dan fungsi kendala pada pendistribusian barang. Formulasi VRPTW disini memiliki bentuk yang berbeda-beda sesuai dengan kasus yang ditangani. Secara matematis formulasi VRPTW dapat dijelaskan dibawah ini.

Fungsi tujuan:

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

Fungsi kendala:

$$\sum_{i=0}^n q_i x_{ij} \leq Q \quad (2.2)$$

$$\sum_{i=0}^{n-1} t_{i,i+1} + t_{n,0} + \sum_{i=0}^n ts \leq T_w \quad (2.3)$$

$$\sum_{i=0}^n x_{i0} = 1, \sum_{j=0}^n x_{0j} = 1 \quad (2.4)$$

Keterangan:

Z = total jarak

n = jumlah titik

c_{ij} = jarak antar titik

x_{ijk} = variabel keputusan, yaitu :

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

q_i = kapasitas sampah

Q = kapasitas maksimal truk

$t_{i,i+1}$ = waktu perjalanan antar titik

ts = waktu pelayanan pengangkutan sampah di setiap TPS

T_w = batas waktu kegiatan

Untuk persamaan (2.1) adalah fungsi tujuan yang bertujuan untuk meminimasi jarak, untuk persamaan (2.2) adalah fungsi kendala kapasitas yang menunjukkan bahwa total kapasitas sampah yang diangkut dalam truk tidak boleh melebihi kapasitas maksimal truk, untuk persamaan (2.3) adalah fungsi kendala waktu yang menunjukkan bahwa total waktu perjalanan dan waktu pelayanan tidak boleh melebihi batas waktu kegiatan pengangkutan sampah yang telah ditentukan, dan untuk persamaan (2.4) fungsi kendala yang menunjukkan bahwa setiap kendaraan yang meninggalkan TPA harus kembali ke TPA.

2.4 Cheapest Insertion Heuristic (CIH)

Algoritma *Cheapest Insertion Heuristic* yaitu algoritma mencari suatu siklus *Hamilton* yang memiliki total beban minimum. Dalam langkah-langkah

pemilihan titik berikutnya pada algoritma ini akan disisipkan dan dalam siklus yang akan disisipkan oleh titik tersebut dengan nilai penyisipan minimum dilakukan bersama.

Algoritma CIH ini diterapkan pada graf terhubung berbobot tidak berarah, berawal dari penentuan *tour* awal yang terdiri dari titik awal kemudian mencari titik baru dimana sisi terkaitnya memiliki beban minimum. Kemudian melakukan langkah pemilihan yaitu titik baru yang tidak dalam *tour* dan juga sisi dalam *tour* sehingga nilai penyisipannya minimum. Lalu titik yang telah terpilih tersebut disisipkan pada sisi yang juga telah terpilih tadi. Proses pada algoritma ini berhenti jika semua titik telah terpilih sehingga terbentuk rute yang merupakan siklus *Hamilton*.

Berikut langkah-langkahnya:

2. Penelusuran dimulai dari titik awal pemberangkatan dalam satu perjalanan. Titik awal tersebut dibuat satu perjalanan ketitik akhir.
3. Dibuat sebuah hubungan *subtour* antara 2 titik tersebut. Yang dimaksud *subtour* adalah perjalanan dari kota pertama dan berakhir di titik awal.
4. Ganti salah satu arah hubungan (*arc*) dari dua titik dengan kombinasi dua *arc*, yaitu *arc* (i, j) dengan *arc* (i, k) dan *arc* (k, j), dengan k diambil dari kota yang belum masuk *subtour* dan dengan tambahan jarak terkecil.

Jarak diperoleh dari:

$$c_{ik} + c_{kj} - c_{ij}$$

c_{ik} adalah jarak dari titik i ke titik k , c_{kj} adalah jarak dari titik k ke titik j dan c_{ij} adalah jarak dari titik i ke titik j .

5. Ulangi langkah 3 sampai seluruh titik masuk dalam *subtour*.

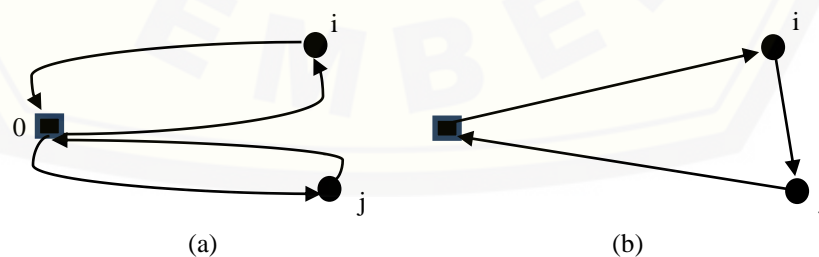
2.5 Metode Penghematan (*Savings Method*)

Pada tahun 1964, Clarke dan Wright mempublikasikan sebuah algoritma sebagai solusi permasalahan dari berbagai rute kendaraan, yang sering disebut sebagai permasalahan klasik dari rute kendaraan (*the classical vehicle routing problem*). Algoritma ini didasari pada suatu konsep yang disebut konsep *savings*. Algoritma ini dirancang untuk menyelesaikan masalah rute kendaraan dengan

karakteristik sebagai berikut. Dari suatu depot barang harus diantarkan kepada pelanggan yang telah memesan. Untuk sarana transportasi dari barang-barang ini, sejumlah kendaraan telah disediakan, dimana masing-masing kendaraan dengan kapasitas tertentu sesuai dengan barang yang diangkut. Setiap kendaraan yang digunakan untuk memecahkan permasalahan ini harus menempuh rute yang telah ditentukan, memulai dan mengakhiri di depot, dimana barang-barang yang diantarkan kepada satu atau lebih pelanggan. Permasalahannya adalah untuk menetapkan lokasi untuk pelanggan di antara rute-rute yang ada, urutan rute yang dapat mengunjungi semua pelanggan dari rute yang ditetapkan dari kendaraan yang dapat melalui semua rute.

Tujuannya adalah untuk menentukan suatu solusi yang meminimalkan total pembiayaan kendaraan. Lebih dari itu, solusi ini harus memuaskan batasan bahwa setiap pelanggan dikunjungi sekali, di mana jumlah yang diminta diantarkan, dan total permintaan pada setiap rute harus sesuai dengan kapasitas kendaraan. Biaya-biaya kendaraan ditetapkan oleh biaya pengangkutan dari beberapa titik ke titik-titik yang lain. Pembiayaan tidak harus sama pada dua jalur di antar dua titik.

Metode *savings* (*savings heuristic*) adalah sebuah algoritma *heuristic*. Bagaimanapun juga metode ini sering menghasilkan solusi yang baik. Dasar dari konsep penghematan ini untuk mendapatkan penghematan biaya dengan menggabungkan rute menjadi satu rute yang digambarkan pada Gambar 2.5, titik 0 adalah depot.



Gambar 2.6 Ilustrasi konsep penghematan

Sumber : Jens Lysgaard (2007)

Berdasarkan Gambar 2.6 (a) pelanggan i dan j dikunjungi dengan rute yang terpisah. Sebuah alternatif untuk masalah ini adalah mengunjungi dua pelanggan

pada rute yang sama, sebagai contoh pada urutan i - j seperti yang diperlihatkan pada Gambar 2.6 (b). karena biaya transportasi diberikan, penghematan yang terjadi dari pengangkutan pada rute Gambar 2.6 (b) dibanding rute pada Gambar 2.6 (a) dapat dihitung. Biaya kendaraan yang ditunjukkan di antara titik i dan j oleh c_{ij} , total biaya kendaraan oleh D_a pada Gambar 2.6 (a) adalah :

$$D_a = c_{oi} + c_{io} + c_{oj} + c_{jo}$$

Ekivalen dengan dengan biaya kendaraan D_b pada Gambar 2.7 (b) adalah :

$$D_b = c_{oi} + c_{ij} + c_{jo}$$

Dengan menggunakan kedua rute memperoleh penghematan S_{ij}

$$S_{ij} = D_a - D_b = c_{oi} + c_{oj} - c_{ij}$$

Besarnya nilai S_{ij} mengindikasikan suatu hal yang menarik, dengan biaya yang telah ditentukan, untuk mengunjungi titik i dan j pada rute yang sama dimana titik j dikunjungi setelah mengunjungi titik i . Ada 2 versi pada algoritma penghematan, versi berurutan (rentetan) dan versi paralel. Pada versi rentetan secara tepat, 1 rute dibuat/dijalani pada suatu waktu (tidak termasuk rute yang hanya dengan 1 pelanggan), sementara versi paralel lebih dari 1 rute dapat dijalani pada suatu waktu.

Pada tahap pertama dari algoritma ini penghematan dari semua pasang pelanggan dihitung, dan semua titik disortir dengan urutan yang menurun dari penghematan. Kemudian dari urutan teratas di daftar sortiran pada pasangan titik, satu pasang titik dikerjakan bersamaan pada satu waktu. Ketika pasangan titik i - j dikerjakan, 2 rute mengunjungi i dan j digabung (misalnya j dikunjungi setelah i pada rute yang dihasilkan), jika hal ini dapat dilakukan tanpa menghapus rute sebelumnya yang telah ditetapkan antara 2 titik, dan jika total permintaan pada rute yang dihasilkan tidak melebihi kapasitas kendaraan. Lakukan pengecekan apakah semua permintaan pelanggan sudah dilayani, jika sudah maka rute terbentuk lalu selesai, namun jika belum maka pilih pelanggan selanjutnya berdasarkan pasangan pelanggan terakhir yang terpilih dengan nilai savings terbesar.

BAB 3. METODOLOGI

3.1 Data Penelitian

Data yang digunakan untuk penelitian ini adalah data-data yang dibutuhkan untuk menyelesaikan permasalahan VRPTW, dalam skripsi ini, data penelitian diambil dari skripsi Sakinah (2012). Dalam skripsi tersebut dijelaskan bahwa proses pengangkutan sampah yang dilakukan Dinas Kebersihan dan Pertamanan Kota Malang selama 8 jam yang dimulai dari pukul 06.00 sampai 14.00. Dengan kapasitas kendaraan sebesar $8.000 m^3$. Untuk kecepatan rata-rata kendaraan $30^{km}/jam$ dan waktu pelayanan pengangkutan sampah setiap $1 m^3$ di TPS adalah 3 detik ($0,00083 jam$). Berikut data rute, jarak dan kapasitas sampah pada TPA dan TPS yang dilalui untuk pengangkutan sampah:

Tabel 3.1 TPS-TPS Yang Dilalui Untuk Pengangkutan Sampah

Sumber : Sakinah. (2012)

TPS	Rute	Total Jarak (km)
TPS Perum PBI	TPA-Blimbing-	134 km
TPS Kanjuruhan	Merjosari-Mergan-	
TPS Tlogomas	Klayatan-kebonsari-	
TPS Raya Langsep	Mulyorejo-Supiturang-	
TPS Keben	TPA	
TPS Terminal Gadang		
TPS Bakalan Krajan		

Tabel 3.2 Volume Sampah Yang Harus Diangkut Di Setiap TPS

Sumber : Sakinah. (2012)

TPS	Volume sampah yang diangkut (m ³)	Total sampah (m ³)
1	998	8053
2	813	
3	876	
4	1367	
5	1450	
6	819	
7	1730	

Tabel 3.3 Jarak TPA ke TPS dan Antar TPS

Sumber : Sakinah. (2012)

v	0	1	2	3	4	5	6	7
0	0	12,6	9,4	13,2	6,8	6,8	13,2	5
1	12,6	0	5,2	8	7,6	9,2	12,2	10,6
2	9,4	5,2	0	4,8	4,2	6,4	8,8	7,2
3	13,2	8	4,8	0	8,2	11,8	14	11,2
4	6,8	7,6	4,2	8,2	0	2,8	15,2	4,8
5	6,8	9,2	6,4	11,8	2,8	0	2,4	2,6
6	13,2	12,2	8,8	14	15,2	2,4	0	5,6
7	5	10,6	7,2	11,2	4,8	2,6	5,6	0

3.2 Langkah – langkah Penelitian

Prosedur yang dilakukan dalam penelitian ini adalah sebagai berikut:

a. Studi Literatur

Langkah pertama yang dilakukan dalam penelitian ini adalah melakukan studi literatur yang diperoleh dari sebagai literatur tentang algoritma *Cheapest Insertion Heuristic* (CIH) yaitu artikel-artikel dan sumber lain yang berhubungan dengan algoritma tersebut.

b. Pengambilan Data

Langkah selanjutnya pengambilan data yang diperoleh dari skripsi Sakinah (2012). Data yang diperoleh yaitu lokasi TPA dan TPS, volume sampah setiap TPS per hari, jarak antara TPA dengan TPS dan antar TPS, waktu proses pengangkutan, ketetapan waktu pelayanan dan kapasitas kendaraan.

c. Penyelesaian VRPTW dengan algoritma CIH dan CIH *with Savings Method*

A. Langkah – langkah Algoritma CIH

Penelusuran dimulai dari titik awal pemberangkatan dalam satu perjalanan. Titik awal tersebut dibuat satu perjalanan ketitik akhir.

- 1) Menentukan titik awal dan titik akhir
- 2) Dibuat sebuah hubungan *subtour* antara 2 titik, dimana satu titik adalah TPA dan satu titik adalah TPS, dengan pemilihan TPS acak. Yang dimaksud *subtour* adalah perjalanan dari titik awal dan berakhir di titik awal.
- 3) Cek apakah *subtour* tersebut memenuhi kendala kapasitas dan kendala waktu. Jika ya, maka i dan j terhubung sehingga menghasilkan *subtour*. Jika tidak maka i dan j tidak terhubung.
- 4) Ganti salah satu arah hubungan (*arc*) dari dua titik dengan kombinasi dua *arc*, yaitu *arc* (i, j) dengan *arc* (i, k) dan *arc* (k, j), dengan k diambil dari kota yang belum masuk *subtour* dan dengan tambahan jarak terkecil. Jarak diperoleh dari:

$$c_{ik} + c_{kj} - c_{ij}$$
 c_{ik} adalah jarak dari titik i ke titik k , c_{kj} adalah jarak dari titik k ke titik j dan c_{ij} adalah jarak dari titik i ke titik j .
- 5) Ulangi langkah 2 sampai seluruh titik masuk dalam *subtour*.

B. Langkah-langkah Algoritma CIH *with Savings Method*

Penelusuran dimulai dari titik awal pemberangkatan dalam satu perjalanan. Titik awal tersebut dibuat satu perjalanan ketitik akhir.

- 1) Menghitung nilai *saving* dengan rumus

$$S_{ij} = c_{oi} + c_{oj} - c_{ij}, \text{ dengan } i = 1, 2, \dots, n - 1 \text{ dan } j = i + 1, \dots, n$$

$$i \neq j, i, j \in N \setminus \{0\}$$

Untuk kasus $c_{i0} = 0$ (tidak terhubung langsung) atau $c_{0j} = 0$ atau $c_{ij} = 0$ maka S_{ij} tidak dihitung.

- 2) Urutkan *saving* dari yang terbesar ke terkecil. Pilih S_{ij} terbesar
- 3) Bangun *subtour* awal yang berawal dan berakhir di TPA dari s_{ij} yang telah dipilih tersebut. Sehingga diperoleh rute $[0, i, j, 0]$. Yang dimaksud *subtour* adalah perjalanan dari kota pertama dan berakhir di titik awal.
- 4) Cek apakah *subtour* tersebut memenuhi kendala kapasitas dan kendala waktu. Jika ya, maka i dan j terhubung sehingga menghasilkan *subtour*. Jika tidak maka i dan j tidak terhubung.
- 5) Ganti salah satu arah hubungan (*arc*) dari dua titik dengan kombinasi dua *arc*, yaitu *arc* (i, j) dengan *arc* (i, k) dan *arc* (k, j) , dengan k diambil dari kota yang belum masuk *subtour* dan dengan tambahan jarak terkecil.

Jarak diperoleh dari :

$$c_{ik} + c_{kj} - c_{ij}$$

c_{ik} adalah jarak dari titik i ke titik k , c_{kj} adalah jarak dari titik k ke titik j dan c_{ij} adalah jarak dari titik i ke titik j .

- 6) Ulangi langkah 4 sampai seluruh titik masuk dalam *subtour*.

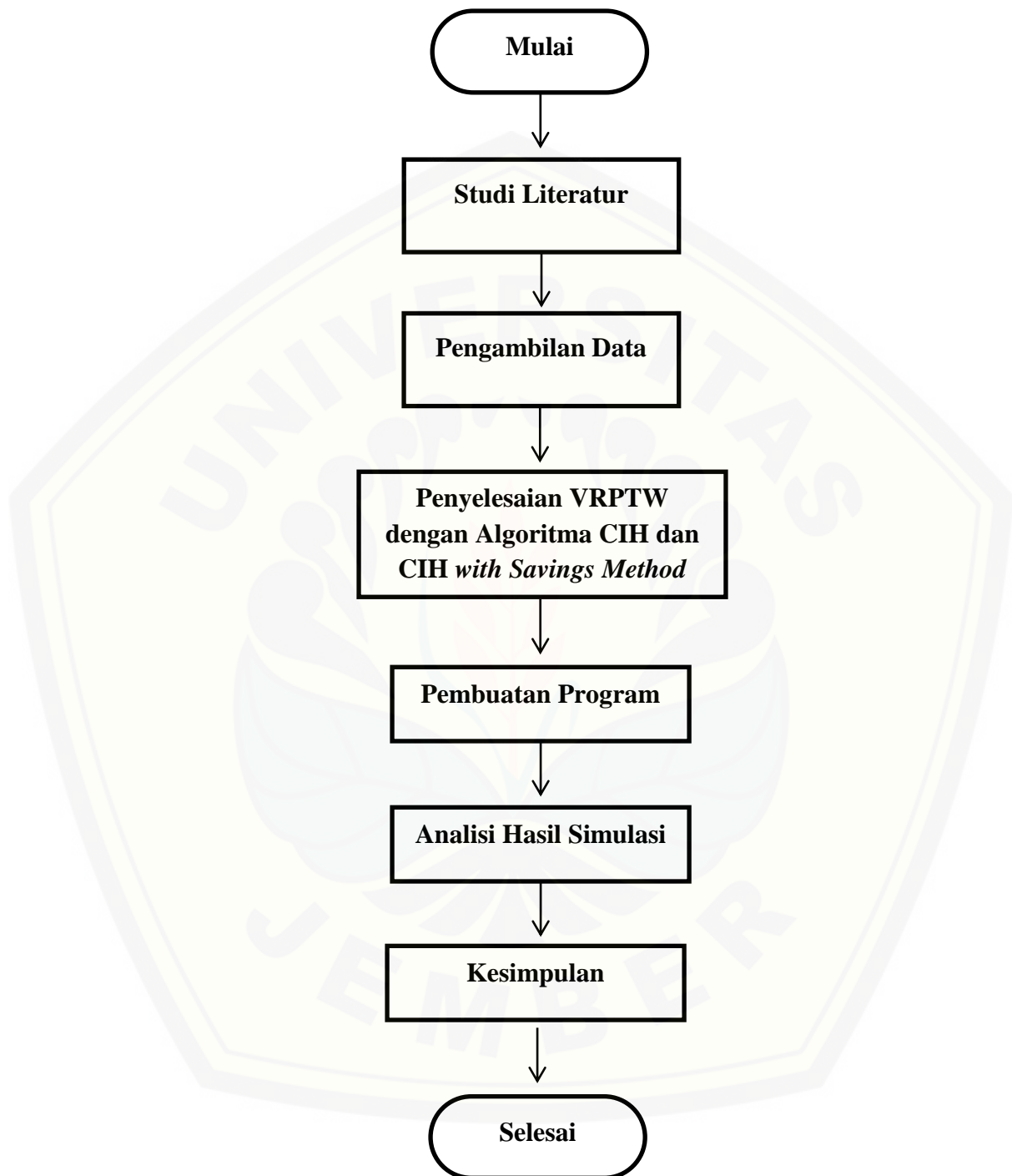
d. Pembuatan Program

Langkah penelitian selanjutnya adalah membuat program dengan menggunakan *software* MATLAB. Pada langkah ini penulis akan membuat skrip program berupa tampilan GUI untuk menyelesaikan VRPTW dengan Algoritma CIH dan CIH *with Savings Method*.

e. Analisis Hasil

Dalam penelitian ini, *variable* yang dapat diubah-ubah yaitu *variable* jumlah titik TPS. Hasil dari langkah ini berupa rute perjalanan, total jarak yang ditempuh dan total kapasitas sampah yang diangkut dan waktu yang ditempuh dari setiap rute yang dihasilkan untuk penyelesaian VRPTW dengan Algoritma CIH dan CIH *with Savings Method*.

3.3 Flowchart Penelitian



BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pembahasan dalam penelitian dapat diambil kesimpulan sebagai berikut:

1. Dalam kasus VRPTW pada pengangkutan sampah dengan menggunakan algoritma *Cheapest Insertion Heuristic* (CIH) dan *Cheapest Insertion Heuristic* (CIH) *with savings method* juga dengan bantuan program MATLAB dapat meminimalkan jarak dan waktu pengangkutan sampah. Dengan menggunakan data 8 titik dan dengan menggunakan data 33 titik.
2. Berdasarkan rute optimal yang dihasilkan, algoritma CIH menghasilkan rute yang lebih baik dengan total jarak keseluruhan dan waktu yang dibutuhkan dalam proses pengangkutan sampah lebih minimal dibandingkan dengan algoritma CIH *with savings method*. Total jarak algoritma CIH *with savings method* lebih besar dibandingkan dengan CIH, jarak subtour awal $[0, i, j, 0]$ yang dipilih dari indeks (titik-titik) S_{ij} dengan nilai saving terbesar menghasilkan jarak yang belum minimal, berbeda dengan penyisipan titik ke *subtour* dengan menggunakan algoritma CIH yang menghasilkan total jarak yang minimal. Total jarak yang belum minimal ini dikarenakan pencarian nilai *saving* hanya mencari nilai penghematan dari 2 titik, yang berawal dan berakhir di titik v_0 , bukan mencari jarak terpendek dari titik awal yang telah terpilih.

5.2 Saran

Dalam penelitian ini memiliki salah satu batasan masalah yaitu jarak antar titik simetris, sehingga untuk penelitian selanjutnya dapat menggunakan jarak antar titik yang tidak simetris. Algoritma CIH tidak terbatas hanya dapat digunakan pada kasus VRPTW, sehingga untuk peneliti selanjutnya dapat menerapkan algoritma CIH pada kasus selain VRPTW, misalnya CVRP, MDVRP dan lain-lain.

DAFTAR PUSTAKA

- Maryanti, Gunawan, Pickerling, Wibowo. 2012. *Vehicle Routing Problem Berbasis Ant Colony System Untuk Optimasi Penentuan Rute Kendaraan Pada Sistem Distribusi Barang Dan Jasa*. Tidak Diterbitkan. Skripsi. Jurusan Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya.
- Hasanah dan Wahyuningsih. 2013. *Kinerja Algoritma Improved Ant Colony System (IACS) Dalam Menyelesaikan Vehicle Routing Problem With Time Windows (VRPTW)*. Tidak Diterbitkan. Skripsi. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Malang, Malang.
- Kusrini, K., dan Istiyanto, J. E. 2008. Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data. *Jurnal Informatika*, **8**(2):109.
- Sakinah. 2012. *Analisis Sistem Pengangkutan Sampah Kota Malang Dengan Metode Penyelesaian Vehicle Routing Problem with Time Windows (VRPTW) untuk Meminimalisasi Jarak dan Waktu*. Tidak Diterbitkan. Skripsi. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Malang, Malang.
- Apriliani. 2011. *Penyelesaian Traveling Salesman Problem (TSP) Menggunakan Algoritma Semut Dan Algoritma Cheapest Insertion Heuristic (CIH)*. Tidak Diterbitkan. Skripsi. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jember, Jember.
- Mulia. 2011. *Aplikasi Algoritma Ant System (AS) Dalam Kasus Traveling Salesman Problem (TSP)*. Tidak Diterbitkan. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah, Jakarta.
- Putri, R. A. 2014. *Penyelesaian Capacitated Vehicle Routing Problem dengan Metode Saving Heuristic*. Tidak Diterbitkan. Skripsi. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jember, Jember.
- Sari, I. N., dan Widodo, A. 2014. Penentuan Rute Yang Optimal Pada Distribusi Kacang Menggunakan Ant Colony System (Studi Kasus Di Pt Qlaworks Indonesia). *Jurnal Mahasiswa Matematika*, **2**(4), 271.
- Pavela, V., dan Purwanto, I. N. 2013. Penyelesaian Vehicle Routing Problem dengan Menggunakan Algoritma Nearest Neighbor dan Tabu Search. *Jurnal Mahasiswa Matematika*, **1**(4), 244.

Rohandi, S. M., Imran, A., dan Prassetiyo, H. 2014. Penentuan Rute Distribusi Produk Obat Menggunakan Metode Sequential Insertion Dan Clarke & Wright Savings (Studi Kasus di PT X Bandung). *Reka Integra*, 2(2).



LAMPIRAN

Lampiran A. Tampilan skrip program fungsi tujuan

```

rute=get(handles.figure1,'userdata');
jarak=get(handles.uipanel1,'userdata');
for i=1:size(rute,2)
    Jartot(i)=0;
    for j=1:length(rute{i})-1
Jartot(i)=Jartot(i)+jarak(rute{i}(j),rute{i}(j+1));
        njt=length(Jartot);
        koljt(i)={'Rute ' num2str(i)};
        for z=1:njt
            barjt(i,z)={num2str(Jartot(z))};
        end
    end
end
a=barjt';
set(handles.CIH,'data',a(:,i),'rowname',koljt);

rutee=get(handles.radiobuttonRT,'userdata');
jarakk=get(handles.radiobuttonTJ,'userdata');
for i=1:size(rutee,2)
    Jartott(i)=0;
    for j=1:length(rutee{i})-1
Jartott(i)=Jartott(i)+jarakk(rutee{i}(j),rutee{i}(
j+1));
        njt=length(Jartott);
        koljtt(i)={'Rute ' num2str(i)};
        for z=1:njt
            barjtt(i,z)={num2str(Jartott(z))};
        end
    end
end
b=barjtt';
set(handles.CIHSV,'data',b(:,i),'rowname',koljtt);

```

Lampiran B. Tampilan skrip program fungsi kendala kapasitas dan kendala waktu

```

while Qtotal<=Qw && Ttotal<=Tw
    Qtotal=sum(q(jalur{ii,iter}));
    Ttotal=0;
        for i=1:length(jalur{ii,iter})-1
Ttotal=Ttotal+jarak(jalur{ii,iter}(i),jalur{ii,ite
r}(i+1))/kec;
        end
end

```

```
St=sum(q(jalur{ii,iter})*waktupel);  
Ttotal=Ttotal+St;
```

```
Qtot(ii)=sum(q(rute{ii}));  
Qtotal=0;  
Ttotal=0;  
for i=1:length(rute{ii})-1  
  
Ttotal=Ttotal+jarak(rute{ii}(i),rute{ii}(i+1))/kec  
;  
end  
St=sum(q(rute{ii})*waktupel);  
Ttot(ii)=Ttotal+St;
```

Lampiran C. Tampilan skrip program fungsi kendala kendaraan berawal dan berakhir pada TPA

```
jalur{1,1}=[ta tb ta];  
  
if length(d)>1  
    jalur{ii+1,1}=[ta d(2) ta];  
    iter=1;  
    ii=ii+1;  
else  
    break;  
end
```

Lampiran D. Data jarak 33 titik

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16
V0	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	15,8
V1	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	17
V2	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	17,9
V3	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	19,4
V4	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	15,8
V5	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	18,6
V6	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	20,3
V7	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	16,8
V8	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	15,4
V9	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	13,9
V10	4,5	5,5	6,6	7,1	3,4	7,1	7,5	4	4,1	4,1	0	4,2	12,5	13,3	18,1	19,1	13,7
V11	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	15,2	10,5
V12	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	22,9
V13	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	2,8
V14	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	4,9
V15	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	6,1
V16	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	0
V17	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	12,5
V18	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	8,6
V19	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	13,7
V20	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	6,4

	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32
v0	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
v1	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
v2	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
v3	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
v4	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
v5	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
v6	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
v7	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
v8	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
v9	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
v10	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
v11	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
v12	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
v13	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
v14	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
v15	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
v16	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
v17	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
v18	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
v19	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
v20	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

	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16
V21	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	36,9
V22	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	12,9
V23	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	14,9
V24	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	10,6
V25	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	23,9
V26	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	29,4
V27	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	20,5
V28	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	38,2
V29	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	42,4
V30	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	17,7	16	16,3
V31	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	17,3	15,6	14,7
V32	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	37,3

	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32
V21	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
V22	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
V23	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
V24	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
V25	34,5	31,6	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
V26	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
V27	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
V28	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
V29	39,8	38,6	39,3	37,9	16,6	52,8	47,7	50,4	61	66,9	58	4,3	0	44,1	42,5	17,6
V30	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
V31	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
V32	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

Lampiran E. Data kapasitas 33 titik

V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21
0	183	11	55	338	31	4	31	11	48	22	13	27	24	29	109	13	57	27	22	88	117

V22	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32
44	2	2	68	102	4	91	16	6	38	44

Lampiran F. Output dari algoritma CIH dengan data 8 titik

No	Rute	Jarak (km)	Waktu (jam)	Kapasitas (m ³)
1	0, 1, 2, 4, 5, 6, 7, 0	37,8	7,2	7.177
2	0, 3, 0	26,4	1,6	876
	Total	64,2	8,8	8.053

Lampiran G. Output dari algoritma CIH *with savings method* dengan data 8 titik

No	Rute	Jarak (km)	Waktu (jam)	Kapasitas (m ³)
1	0, 1, 3, 2, 4, 5, 7, 0	40	7,36	7.234
2	0, 6, 0	26,4	1,56	819
	Total	66,4	8,92	8.053

Lampiran H. Output dari algoritma *savings method* dengan data 8 titik

No	Rute	Jarak (km)	Waktu (jam)	Kapasitas (m ³)
1	0, 1, 3, 2, 6, 5, 4, 0	46,2	6,8	6.323
2	0, 7, 0	10	1,8	1730
	Total	56,2	8,6	8.053

Lampiran I. Output dari algoritma CIH dengan data 33 titik

No	Rute	Jarak (km)	Waktu (jam)	Kapasitas (m ³)
1	0, 2, 12, 6, 3, 5, 7, 1, 0	28,7	1,24	342
2	0, 4, 8, 10, 11, 18, 20, 9, 0	38	1,72	547
3	0, 13, 14, 15, 24, 26, 27, 22, 23, 25, 16, 31, 30, 17, 19, 0	195	6,93	520
4	0, 21, 29, 28, 32, 0	87,6	3,14	268
Total		349,3	13,03	1677

Lampiran J. Output dari algoritma CIH *with savings method* dengan data 33 titik

No	Rute	Jarak (km)	Waktu (jam)	Kapasitas (m ³)
1	0, 4, 26, 27, 10, 0	100,4	3,73	466
2	0, 1, 7, 9, 11, 18, 20, 17, 8, 5, 3, 6, 12, 2, 0	69,1	2,79	586
3	0, 31, 30, 13, 14, 15, 24, 22, 23, 25, 16, 19, 32, 0	185	6,5	401
4	0, 21, 29, 28, 0	70,8	2,54	224
Total		425,3	15,56	1677