



PENERAPAN ALGORITMA *CHEAPEST INSERTION HEURISTIC* (CIH) DENGAN *SWEEP CLUSTERING* PADA *CAPACITATED VEHICLE ROUTING PROBLEM* (CVRP)

SKRIPSI

Oleh

**Nor Laela Ramadhaniyah
NIM 161810101066**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020**



PENERAPAN ALGORITMA *CHEAPEST INSERTION HEURISTIC* (CIH) DENGAN *SWEEP CLUSTERING* PADA *CAPACITATED VEHICLE ROUTING PROBLEM* (CVRP)

SKRIPSI

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

Oleh

Nor Laela Ramadhaniyah
NIM 161810101066

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Khamid dan Ibunda Nadziroh yang memberikan seluruh semangat dan doanya selama perjalanan studi;
2. Bapak Abduh Riski, S.Si., M.Si. selaku Dosen Pembimbing Utama dan Dr. Agustina Pradjaningsih, S.Si., M.Si. selaku Dosen Pembimbing Anggota yang telah membimbing secara intensif dalam menyempurnakan skripsi ini;
3. Seluruh keluarga besar Bani M.Yunus dan Bani M. Nur di Pasuruan yang tercinta;
4. Almamater Jurusan Matematika FMIPA Universitas Jember, SMAN 1 Pasuruan, SMPN 2 Pasuruan dan SDN Karangketug 1 Pasuruan;
5. Seluruh teman-teman Misdirection'16;

MOTTO

“Allah tidak membebani kewajiban kepada seseorang kecuali sesuai dengan kesanggupannya”

(terjemahan Surat *Al-Baqarah*, Ayat 286)*)

atau

“Sesungguhnya dimana ada kesulitan disitu ada kelapangan”

(terjemahan syrat *Al-Insyirah*, Ayat 5)*)



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

PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama : Nor Laela Ramadhaniyah

NIM : 161810101066

menyatakan dengan sebenarnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma *Cheapest Insertion Heuristic* (CIH) dengan *Sweep Clustering* Pada *Capacitated Vehicle Routing Problem* (CVRP)” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggungjawab 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 apabila ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Januari 2020

Yang menyatakan,

Nor Laela Rmadhaniyah

NIM 161810101066

SKRIPSI

**PENERAPAN ALGORITMA *CHEAPEST INSERTION HEURISTIC* (CIH)
DENGAN *SWEEP CLUSTERING* PADA *CAPACITATED VEHICLE
ROUTING PROBLEM* (CVRP)**

Oleh

Nor Laela Ramadhaniyah

NIM 161810101066

Pembimbing:

Dosen Pembimbing Utama : Abduh Riski, S.Si., M.Si.

Dosen Pembimbing Anggota : Dr Agustina Pradjaningsih, S.Si, M.Si

PENGESAHAN

Skripsi berjudul “Penerapan Algoritma *Cheapest Insertion Heuristic* (CIH) dengan *Sweep Clustering* Pada *Capacitated Vehicle Routing Problem* (CVRP)” telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

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

Dr. Agustina Pradjaningsih, S.Si, M.Si.
NIP. 197108022000032009

Anggota II,

Anggota III,

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

Dr. Kristiana Wijaya, S.Si., M.Si.
NIP. 197408132000032004

Mengesahkan

Dekan,

Drs. Achmad Sjaifullah, M.Sc., Ph.D.
NIP. 195910091986021001

RINGKASAN

Penerapan Algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* Pada *Capacitated Vehicle Routing Problem*; Nor Laela Ramadhaniyah, 161810101066; 2020; 60 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Industri pada masa kini berkembang sangat pesat seiring dengan kecanggihan teknologi yang semakin modern. Maka dari itu, persaingan industri juga semakin ketat. Dengan adanya persaingan itulah yang membuat setiap perusahaan harus bisa mengatur perusahaannya, salah satunya dengan proses distribusi dari perusahaan itu. Proses distribusi adalah proses pengiriman barang dari depot ke pelanggan. Penting bagi setiap perusahaan karena berkaitan dengan kepuasan pelanggan yang mempengaruhi keuntungan produksi. Konsep distribusi diantaranya jarak, waktu dan harga minimum itu dipengaruhi oleh kapasitas kendaraan yang akan didistribusikan dari depot ke pelanggan, dengan begitu hal tersebut bisa merupakan kasus *Capacitated Vehicle Routing Problem*.

Penelitian ini bertujuan untuk mendapatkan solusi permasalahan *Capacitated Vehicle Routing Problem* pada pendistribusian produk dengan menerapkan dua jenis algoritma yaitu *Cheapest Insertion Heuristic* dengan *Sweep Clustering* dan algoritma *Cheapest Insertion Heuristic*. Data yang digunakan dalam penelitian ini adalah data distribusi beras. Data ini terdiri atas sebuah depot dan 30 pelanggan. Data yang digunakan terdapat empat kriteria yang meliputi lokasi depot dan pelanggan, jumlah permintaan setiap pelanggan, jarak antar lokasi dan kapasitas kendaraan.

Algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* menggunakan dua metode yaitu *Forward Sweep* dan *Backward Sweep*. Hasil penelitian menunjukkan bahwa dari penerapan kedua algoritma menghasilkan total jarak yang berbeda. Hasil optimal yang di dapatkan terjadi pada algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering*. Hal ini karena secara umum jarak total minimal paling banyak didapatkan dengan menggunakan

algoritma tersebut daripada jarak total minimal dengan menggunakan algoritma *Cheapest Insertion Heuristic*. Demikian bisa disimpulkan bahwa pada permasalahan *Capacitated Vehicle Routing Problem* menggunakan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* lebih baik daripada hanya menggunakan algoritma *Cheapest Insertion Heuristic*.



PRAKATA

Puji syukur penulis panjatkan atas kehardirat Allah SWT, yang telah melimpahkan rahmat, taufiq, dan hidayahnya sehingga skripsi yang berjudul “Penerapan Algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* Pada *Capacitated Vehicle Routing Problem*” dapat selesai. Skripsi ini disusun guna untuk melengkapi tugas akhir dan salah satu syarat untuk menyelesaikan program sarjana (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Orangtua saya yang selalu memberi doa dan dukungan;
2. Bapak Abduh Riski S.Si., M.Si. selaku Dosen Pembimbing Utama dan Dr. Agustina Pradjaningsih, S.Si, M.Si selaku Dosen Pembimbing Anggota yang telah menyisihkan sedikit waktunya untuk membimbing penulis selama menyusun skripsi ini;
3. Bapak Kosala Dwidja Purnomo, S.Si., M.Si. selaku Dosen Penguji I dan Dr. Kristiana Wijaya, S.Si, M.Si. selaku Dosen Penguji II yang telah banyak memberikan kritik dan saran dalam penulisan skripsi ini;
4. teman-teman angkatan 2016 yang telah memberikan motivasinya;

Penulis menyadari bahwa masih banyak kekuarangan dalam penyusunan skripsi, untuk itu diharapkan kritik dan saran guna penyempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat memberikan manfaat terhadap pembaca.

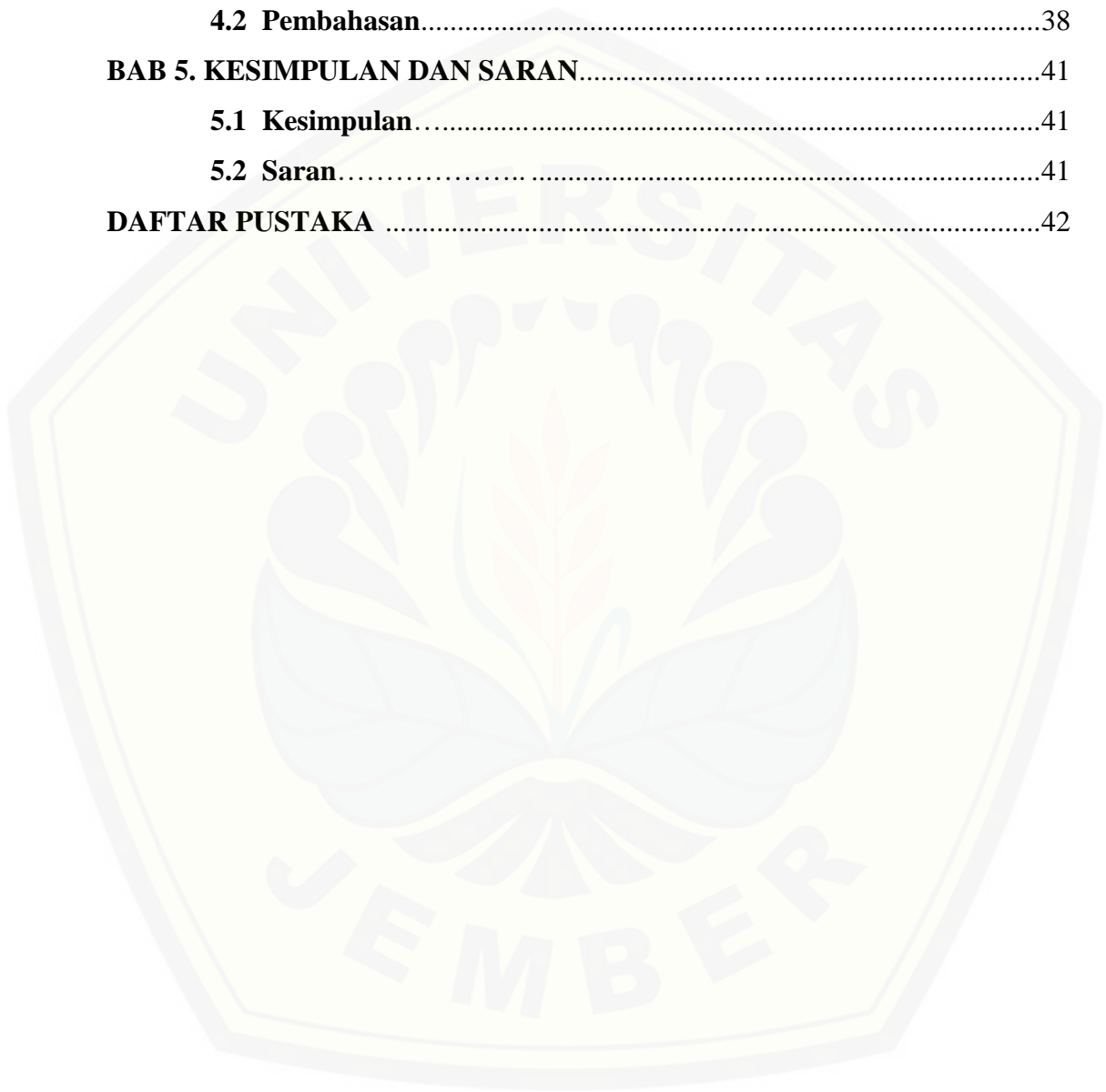
Jember, Januari 2020

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4. Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 <i>Vehicle Routing Problem (VRP)</i>	4
2.2 <i>Capacitated Vehicle Routing Problem</i>	7
2.3 <i>Algoritma Sweep</i>	8
2.4 <i>Insertion Heuristic</i>	10
2.5 <i>Algoritma Cheapest Insertion Heuristic</i>	11
BAB 3. METODE PENELITIAN	14
3.1 Data Penelitian	14
3.2 Langkah-langkah Penelitian	14

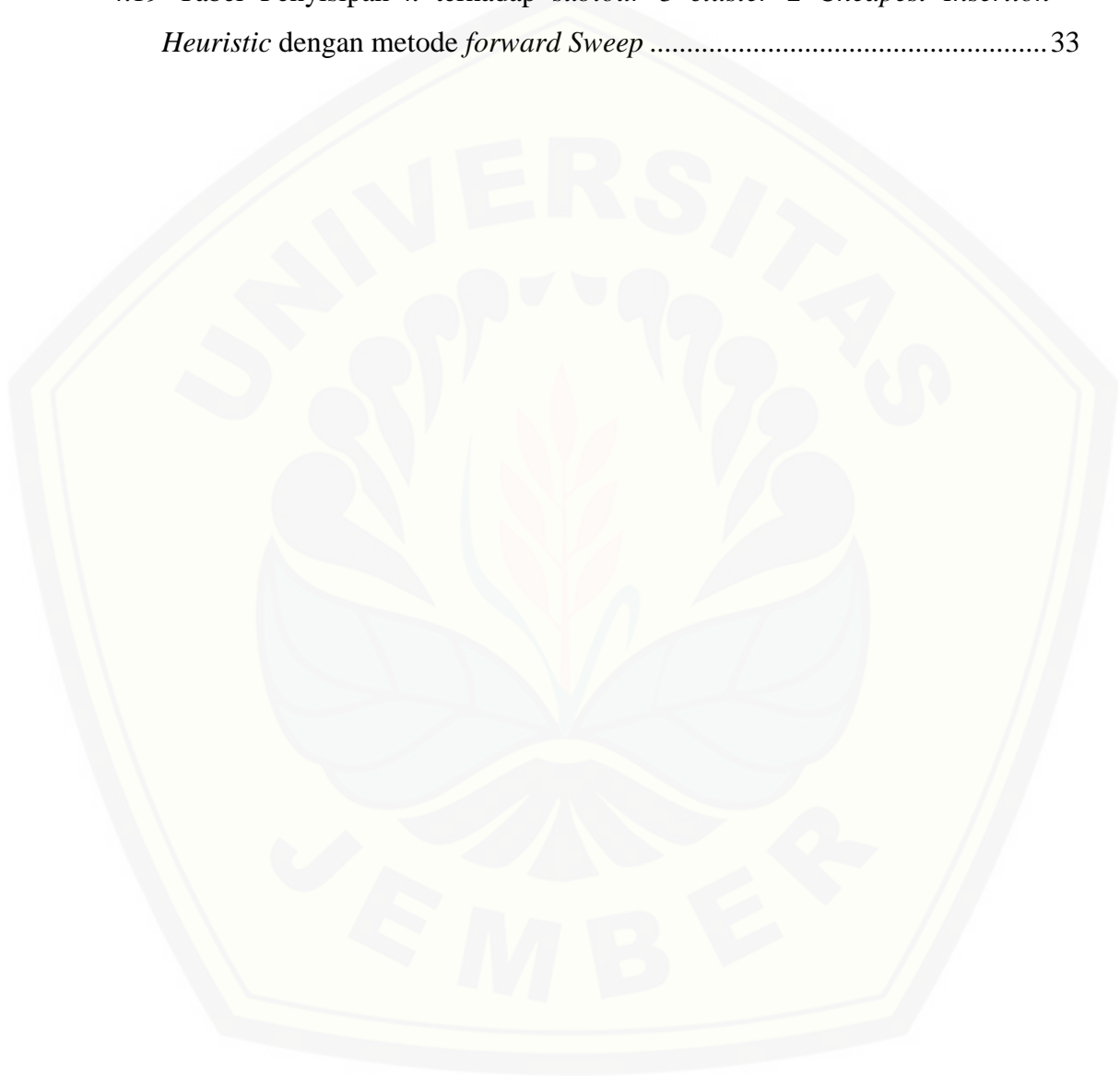
BAB 4. HASIL DAN PEMBAHASAN	17
4.1 Hasil Penelitian	17
4.1.1 Penyelesaian Manual.....	17
4.1.2 Penyelesaian Menggunakan Program.....	34
4.2 Pembahasan	38
BAB 5. KESIMPULAN DAN SARAN	41
5.1 Kesimpulan	41
5.2 Saran	41
DAFTAR PUSTAKA	42



DAFTAR TABEL

	Halaman
2.1 Tabel jarak antar titik	12
2.2 Tabel sisipan titik ke <i>subtour</i> (1,5)	12
2.3 Tabel sisipan titik ke <i>subtour</i> berikutnya	13
2.4 Tabel sisipan titik terakhir.....	13
4.1 Tabel Jarak Pelanggan.....	18
4.2 Tabel Permintaan Pelanggan.....	18
4.3 Tabel Penyisipan k terhadap <i>subtour</i> 2 rute 1 <i>Cheapest Insertion Heuristic</i> ..	19
4.4 Tabel Penyisipan k terhadap <i>subtour</i> 3 rute 1 <i>Cheapest Insertion Heuristic</i> ..	20
4.5 Tabel Penyisipan k terhadap <i>subtour</i> 4 rute 1 <i>Cheapest Insertion Heuristic</i> ..	21
4.6 Tabel Penyisipan k terhadap <i>subtour</i> 2 rute 2 <i>Cheapest Insertion Heuristic</i> ..	22
4.7 Tabel Penyisipan k terhadap <i>subtour</i> 3 rute 2 <i>Cheapest Insertion Heuristic</i> ..	22
4.8 Tabel Data Koordinat Lokasi depot dan pelanggan	24
4.9 Tabel Urutan Pelanggan Berdasarkan nilai θ	26
4.10 Tabel Penyisipan k terhadap <i>subtour</i> 2 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>forward Sweep</i>	29
4.11 Tabel Penyisipan k terhadap <i>subtour</i> 2 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>backward Sweep</i>	29
4.12 Tabel Penyisipan k terhadap <i>subtour</i> 3 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>forward Sweep</i>	29
4.13 Tabel Penyisipan k terhadap <i>subtour</i> 4 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>forward Sweep</i>	30
4.14 Tabel Penyisipan k terhadap <i>subtour</i> 3 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>backward Sweep</i>	30
4.15 Tabel Penyisipan k terhadap <i>subtour</i> 4 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>backward Sweep</i>	31
4.16 Tabel Penyisipan k terhadap <i>subtour</i> 5 cluster 1 <i>Cheapest Insertion Heuristic</i> dengan metode <i>backward Sweep</i>	31

4.17 Tabel Penyisipan k terhadap <i>subtour 2 cluster 2 Cheapest Insertion Heuristic</i> dengan metode <i>forward Sweep</i>	32
4.18 Tabel Penyisipan k terhadap <i>subtour 2 cluster 2 Cheapest Insertion Heuristic</i> dengan metode <i>backward Sweep</i>	32
4.19 Tabel Penyisipan k terhadap <i>subtour 3 cluster 2 Cheapest Insertion Heuristic</i> dengan metode <i>forward Sweep</i>	33



DAFTAR GAMBAR

	Halaman
2.1 <i>Subtour</i>	11
2.2 Lintasan terpendek kelima titik	13
3.1 Skema langkah-langkah penelitian	16
4.1 <i>Forward Sweep</i>	27
4.2 <i>Backward Sweep</i>	27
4.3 Tampilan Awal Program	34
4.4 Tampilan Input pada Program.....	35
4.5 Tampilan Input dan Parameter pada Program.....	36
4.6 Tampilan <i>Output</i> Program CVRP pada CIH dan CIH dengan Sweep Clustering metode <i>Forward</i>	36
4.7 Tampilan <i>Output</i> Program CVRP pada CIH dan CIH dengan Sweep Clustering metode <i>Forward</i>	36
4.8 Tampilan Rute <i>Output</i> Program CVRP pada CIH	37
4.9 Tampilan Rute <i>Output</i> Program CVRP pada CIH dengan <i>Sweep Clustering</i> metode <i>Forward</i>	37
4.10 Tampilan Rute <i>Output</i> Program CVRP pada CIH dengan <i>Sweep</i> <i>Clustering</i> metode <i>Backward</i>	37

DAFTAR LAMPIRAN

	Halaman
A. Lokasi Depot dan Pelanggan.....	43
B. Koordinat Lokasi Pelanggan.....	44
C. Permintaan Pelanggan.....	44
D. Data Jarak Antar Setiap Lokasi.....	45
E. <i>Script Program Capacitated Vehicle Routing Problem</i> menggunakan <i>Algoritma Cheapest Insertion Heuristic</i>	46
F. <i>Script Program Capacitated Vehicle Routing Problem</i> menggunakan <i>Algoritma Cheapest Insertion Heuristic</i> dengan <i>Sweep Clustering</i>	47
G. Total Jarak Tempuh Data dengan Menggunakan Program.....	49
H. Gambar Data dengan Menggunakan Program	52

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Industri pada masa kini berkembang sangat pesat seiring dengan kecanggihan teknologi yang semakin modern. Maka dari itu, persaingan industri juga semakin ketat. Dengan adanya persaingan itulah yang membuat setiap perusahaan harus bisa mengatur perusahaannya, salah satunya dengan proses distribusi dari perusahaan itu. Proses distribusi adalah proses pengiriman barang dari depot ke pelanggan. Penting bagi setiap perusahaan karena berkaitan dengan kepuasan pelanggan yang mempengaruhi keuntungan produksi. Kepuasan pelanggan pada saat distribusi berkaitan dengan cepatnya produk sampai ke pelanggan dengan aman, tepat waktu dan tidak rusaknya produk sesuai dengan permintaan pelanggan, dan murahnya harga penjualan. Murahnya harga penjualan juga salah satu faktornya dari biaya distribusi yang rendah. Dalam konsep distribusi bahwa jarak, waktu bahkan harga minimum juga dipengaruhi oleh kapasitas kendaraan yang akan didistribusikan dari depot ke pelanggan, dengan begitu hal tersebut bisa dikenal dengan istilah *Capacitated Vehicle Routing Problem*.

Capacitated Vehicle Routing Problem merupakan variasi dari *Vehicle Routing problem*. *Vehicle Routing Problem* merupakan suatu permasalahan menemukan rute untuk sekumpulan kendaraan yang harus melayani sejumlah pelanggan dari depot. Sedangkan *Capacitated Vehicle Routing Problem* dengan mencari sejumlah rute minimum dari depot dimana kendaraan berangkat dan kembali lagi ke depot dan pelanggan yang dilayani satu kali oleh tepat satu kendaraan dengan aturan yaitu setiap kendaraan mempunyai kapasitas terbatas dan tidak boleh melebihi dari jumlah muatan yang akan diangkut.

Permasalahan *Capacitated Vehicle Routing Problem* telah diselesaikan menggunakan beberapa algoritma diantaranya adalah menggunakan metode *Saving Heuristic* yang sudah dilakukan oleh Putri (2014). Dari metode *Saving Heuristic* kemudian dikembangkan oleh Yansyah (2016) memadukan dengan algoritma *Cheapest Insertion Heuristic*. tetapi dengan kasus yang berbeda yaitu

Vehicle Routing Problem Time Windows (VRPTW). Kesimpulan penelitiannya adalah algoritma *Cheapest Insertion Heuristic* menghasilkan rute yang lebih baik dibandingkan dengan *Cheapest Insertion Heuristic* dengan *Savings Method*. Kemudian, *Capacitated Vehicle Routing Problem* juga dapat diselesaikan dengan menggunakan algoritma *Sweep* yang sudah dilakukan oleh (Saraswati dkk, 2017).

Berdasarkan uraian sebelumnya, penulis tertarik untuk meneliti lebih lanjut tentang *Capacitated Vehicle Routing Problem* yang akan diselesaikan menggunakan metode *Cheapest Insertion Heuristic* yang diawali dengan pengelompokan titik menggunakan *Sweep Clustering*. Penerapan algoritma *Cheapest Insertion Heuristic* berdasarkan *Sweep Clustering* yang akan diteliti diharapkan dapat menentukan rute yang lebih optimal dibandingkan dengan algoritma *Cheapest Insertion Heuristic*.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka permasalahan yang akan dibahas dalam proposal ini adalah sebagai berikut:

- a. Bagaimana menyelesaikan penerapan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* pada *Capacitated Vehicle Routing Problem* sehingga menghasilkan rute optimal?
- b. Bagaimana perbandingan hasil rute menggunakan algoritma *Cheapest Insertion Heuristic* dan *Cheapest Insertion Heuristic* dengan *Sweep Clustering*?

1.3 Tujuan Penelitian

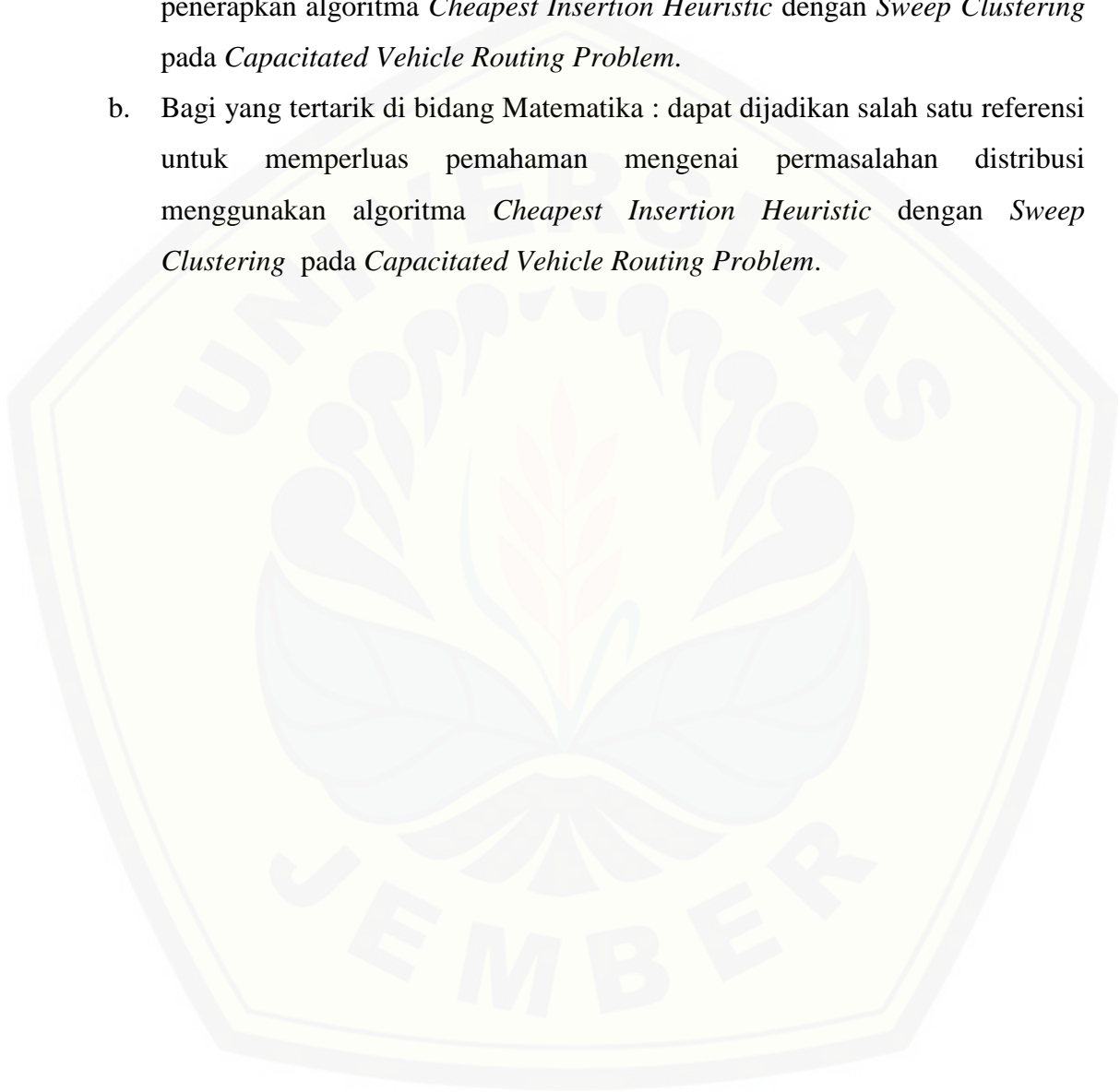
Adapun tujuan dari penulisan proposal ini adalah sebagai berikut:

- a. Menyelesaikan penerapan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* pada *Capacitated Vehicle Routing Problem* sehingga menghasilkan rute yang optimal.
- b. Membandingkan penyelesaian *Capacitated Vehicle Routing Problem* menggunakan algoritma *Cheapest Insertion Heuristic* dan *Cheapest Insertion Heuristic* (CIH) dengan *Sweep Clustering*

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

- a. Bagi Perusahaan : mendapatkan rute pendistribusian optimal mengenai penerapan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* pada *Capacitated Vehicle Routing Problem*.
- b. Bagi yang tertarik di bidang Matematika : dapat dijadikan salah satu referensi untuk memperluas pemahaman mengenai permasalahan distribusi menggunakan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* pada *Capacitated Vehicle Routing Problem*.



BAB 2. TINJAUAN PUSTAKA

2.1 *Vehicle Routing Problem (VRP)*

Vehicle Routing Problem (VRP) adalah salah satu permasalahan dalam mendesain rute dengan biaya minimum, yang berasal dan berakhir pada depot tertentu, yang menggunakan sejumlah kendaraan (*vehicle*) dengan kapasitas muatan yang homogen untuk melayani seorang pelanggan. Setiap pelanggan akan dikunjungi oleh kendaraan pengangkut sebanyak satu kali.

VRP secara sederhana dapat digambarkan sebagai berikut:

- a. Kendaraan akan berangkat dari depot untuk mengunjungi semua pelanggan dan harus kembali lagi ke depot awal.
- b. Setiap pelanggan hanya boleh dikunjungi tepat satu kali oleh satu kendaraan.
- c. Setiap kendaraan yang berangkat harus membawa barang dan banyaknya barang tersebut tidak lebih dari atau sama dengan kapasitas dari kendaraan yang digunakan tersebut.
- d. Menyelesaikan masalah untuk menemukan suatu himpunan rute dalam mendistribusikan barang dari lokasi depot ke lokasi pelanggan dengan biaya seminimum mungkin.

Tujuan VRP adalah untuk menentukan rute dengan total jarak minimum, dimana setiap rute berawal dan berakhir pada depot dan masing-masing pelanggan tepat dilayani sekali dengan total permintaan dalam satu rute tidak melebihi kapasitas kendaraan (Charles dan Sarker, 2008).

Berdasarkan batasan atau kendala yang ada, VRP dibagi beberapa tipe :

a. *Capacitated VRP (CVRP)*

Capacitated Vehicle Routing Problem adalah sebuah VRP pada sejumlah kendaraan dengan kapasitas tersendiri yang harus melayani sejumlah permintaan pelanggan yang telah diketahui untuk satu komoditas dari sebuah depot dengan biaya transit minimum. Tujuan dari CVRP adalah meminimalisasi jumlah kendaraan dan total waktu perjalanan, dan total permintaan barang untuk tiap rute tidak boleh melebihi kapasitas kendaraan yang melewati rute tersebut. Solusi dikatakan 'layak' jika jumlah total barang yang diatur untuk tiap rute tidak

melebihi kapasitas kendaraan yang melewati rute tersebut. Oleh karena itu, *CVRP* sama seperti *VRP* dengan faktor tambahan yaitu tiap kendaraan punya kapasitas tersendiri untuk satu komoditas.

CVRP dapat dijelaskan sebagai berikut :

- 1) Suatu depot harus melayani sejumlah n pelanggan.
- 2) Depot mempunyai satu *vehicle* dengan kapasitas tertentu Q untuk melayani semua titik.
- 3) Tiap pelanggan mempunyai permintaan tertentu sebesar Q_i ($i = 1,2,3,\dots,n$) yang harus dipenuhi dalam sekali pelayanan.
- 4) Karena depot hanya mempunyai satu *vehicle* dengan kapasitas terbatas, maka *vehicle* tersebut harus secara periodik kembali ke depot untuk mengambil barang (*reloading*) untuk memenuhi permintaan pelanggan yang lain.
- 5) Tidak mungkin melayani lebih dari 1 pelanggan dalam waktu yang bersamaan karena *vehicle* hanya satu.

b. *VRP With Time Windows (VRPTW)*

Vehicle Routing Problem with Time Window, hampir sama dengan *VRP*, namun memiliki batas tambahan yaitu sebuah jangka waktu, yang berhubungan dengan setiap pelanggan, yang mendefinisikan sebuah jangka waktu dimana sang pelanggan harus disuplai, interval waktu di depot tersebut sebagai batas penjadwalan. Tujuannya adalah meminimalisasi jumlah kendaraan dan total waktu perjalanan dan waktu menunggu yang dibutuhkan untuk menyuplai semua pelanggan pada jam-jam tertentu.

c. *Multiple Depot VRP (MDVRP)*

Multiple Depot Vehicle Routing Problem adalah meminimalisasi jumlah kendaraan dan total waktu perjalanan dan total permintaan barang yang harus dilakukan dari beberapa depot. *MDVRP* membutuhkan pengaturan para pelanggan ke depot-depot yang ada. Tiap kendaraan pergi dari satu depot melayani pelanggan-pelanggan yang sudah ditentukan akan dilayani oleh depot tersebut, dan kembali lagi ke depot tersebut.

d. *VRP With Pick-Up and Delivering (VRPPD)*

Vehicle Routing Problem with Pick-Up and Delivering adalah sebuah *VRP* dimana ada peluang kejadian pelanggan mengembalikan barang yang sudah diantarkan. Dalam *VRPPD* yang perlu diperhatikan adalah bahwa barang yang dikembalikan dapat dimasukkan ke dalam kendaraan pengantar. Batasan ini membuat perencanaan pengantaran menjadi lebih sulit dan bisa berakibat pada penyalahgunaan kapasitas kendaraan, memperbesar jarak perjalanan atau kendaraan yang diperlukan lebih dari yang seharusnya. Tujuannya adalah meminimalisasi jumlah kendaraan dan total waktu perjalanan dengan batasan bahwa kendaraan yang digunakan harus punya kapasitas yang cukup untuk mengantarkan barang ke pelanggan dan pengembalian barang ke depot.

e. *Split Delivery VRP (SDVRP)*

Split Delivery Vehicle Routing Problem adalah perluasan *VRP* jika tiap pelanggan dapat dilayani dengan kendaraan yang berbeda andaikan biayanya dapat berkurang. Perluasan ini perlu dilakukan jika jumlah permintaan pelanggan sama besar dengan kapasitas dari kendaraan. Tujuannya untuk meminimalisasi jumlah kendaraan dan total waktu perjalanan untuk pelayanan. Solusi dianggap layak jika tiap rute memenuhi batasan standar *VRP* ditambah dengan tiap pelanggan bisa dilayani oleh lebih dari satu kendaraan.

f. *Stochastic VRP (SVRP)*

Stochastic Vehicle Routing Problem atau *SPRV* adalah variasi *VRP* yang terjadi jika faktor sampingnya yang muncul bersifat acak. Dalam *SPRV* untuk mendapatkan solusi, masalah harus dibagi menjadi dua tahap. Solusi pertama ditentukan sebelum variabel random diketahui. Pada tahap kedua, pengoreksian dilakukan jika nilai dari variabel random sudah diketahui. Tujuannya adalah meminimalisasi jumlah kendaraan dan total waktu perjalanan untuk melayani pelanggan dengan nilai random untuk tiap pengantaran (pelanggan, permintaan, waktu).

g. *Periodic VRP*

Dalam *Periodic Vehicle Routing Problem* adalah *VRP* digeneralisasi dengan memperluas rentang perencanaan pengiriman menjadi M hari, dari semula hanya dalam rentang sehari. Tujuannya adalah meminimalisasi jumlah kendaraan dan total waktu perjalanan untuk melayani tiap pelanggan. (Prana, 2007).

2.2 *Capacitated Vehicle Routing Problem*

Capacitated Vehicle Routing Problem merupakan bentuk paling dasar dari *VRP*. *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 lokasi B ke A (Gunawan dkk, 2012).

Variabel keputusan untuk meminimumkan total jarak tempuh kendaraan adalah

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

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

Meminimumkan

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

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 rute berawal dari depot

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

d. Kendaraan selalu datang dan pergi disetiap pelanggan

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

e. Setiap rute berakhir di depot :

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

f. Variabel keputusan *integer biner*.

$$x_{ij}^k \in \{0, 1\}; i, j = 0, 1, \dots, N + 1; k = 1, 2, \dots, K \quad (2.7)$$

dengan,

K = banyak kendaraan

N = banyak pelanggan

c_{ij} = jarak depot i ke pelanggan j

q_j = banyak permintaan pelanggan j

Q = kapasitas maksimum kendaraan k (Kallehauge dkk, 2001)

2.3 Algoritma Sweep

Algoritma *Sweep* termasuk dalam algoritma *heuristic* untuk meminimumkan rute kendaraan dengan menggunakan pengelompokkan. Algoritma *sweep* merupakan algoritma dua tahap, yaitu tahap pertama terdiri dari *clustering* pelanggan yang mana *clustering* awal dilakukan dengan menggabungkan titik-titik dalam satu *cluster* berdasarkan kapasitas maksimal kendaraan, dan tahap kedua adalah membentuk rute-rute untuk masing-masing *cluster*. Berikut ini merupakan langkah-langkah yang harus dilakukan dalam menyelesaikan permasalahan CVRP dengan menggunakan algoritma *sweep* (Boonkleaw dkk, 2009) :

a. Tahap Pengelompokan (Clustering) :

Tahap pertama dalam algoritma *sweep* adalah mengelompokkan masing-masing titik pelanggan ke dalam sebuah *cluster*. Berikut ini adalah langkah-langkah dalam pengelompokan :

- 1) Menggambar masing-masing pelanggan (yang selanjutnya disebut sebagai titik) dalam koordinat kartesius dan menetapkan titik depot sebagai pusat koordinat.
- 2) Menentukan semua koordinat polar dari masing-masing titik yang berhubungan dengan depot. Langkah untuk mengubah koordinat kartesius (x, y) menjadi koordinat polar (r, θ) adalah sebagai berikut :

$$r = \sqrt{x^2 + y^2} \quad (2.8)$$

$$\theta = \tan^{-1} \frac{y}{x} \quad (2.9)$$

Setelah diperoleh masing-masing titik dalam koordinat polar maka selanjutnya menggambarkan titik-titik tersebut dalam bidang dua dimensi.

- 3) Melakukan pengelompokan (*clustering*) dimulai dari titik yang memiliki sudut polar terkecil dan seterusnya berurutan sampai titik yang memiliki sudut polar terbesar dengan memperhatikan kapasitas kendaraan.
- 4) Memastikan semua titik “tersapu” dalam *cluster* saat ini.
- 5) Pengelompokan dihentikan ketika dalam satu *cluster* tidak melebihi kapasitas maksimal kendaraan.
- 6) Membuat *cluster* baru dengan langkah yang sama seperti langkah c dimulai dari titik yang memiliki sudut polar terkecil yang belum termasuk dalam *cluster* sebelumnya (titik yang terakhir ditinggalkan).
- 7) Mengulangi langkah 3 – 6 , sampai semua titik telah dimasukkan dalam sebuah *cluster*.

b. Tahap Pembentukan Rute :

Tahap pembentukan rute distribusi, dari tiap *cluster* akan diselesaikan dengan metode *Nearest Neighbour* sehingga didapatkan diperoleh rute perjalanan dari tiap *cluster*

2.4 Insertion Heuristic

Insertion Heuristics adalah teknik dengan menggunakan metode *heruristic* yang dirancang untuk mencegah masalah dan menghasilkan solusi yang baik atau memecahkan masalah lebih sederhana dengan pemecahan masalah yang lebih kompleks. *Insertion Heuristics* sangatlah lugas, dan ada banyak varian yang bisa dipilih. Dasar-dasar *insertion heuristics* adalah memulai dengan tur *subset* dari semua pelanggan, dan kemudian memasukkan sisanya dengan beberapa heuristik. *Subtour* awal sering berbentuk suatu segitiga atau sebuah *convex hull*. *Insertion Heuristics* juga dapat memulai dengan tepi tunggal sebagai *subtour*.

Pendekatan intuitif adalah memulai dengan sebuah *subtour*, yaitu tur pada *subset* kecil dari titik, dan kemudian memperpanjang tur ini dengan memasukkan simpul yang tersisa satu demi satu sampai semua titik telah dimasukkan. Ada beberapa kemungkinan untuk menerapkan skema seperti penyisipan. Mereka dapat diklasifikasikan menurut fitur ini:

- a. Bagaimana membangun tur awal.
- b. Bagaimana memilih simpul berikutnya yang akan dimasukkan.
- c. Bagaimana menyisipkan simpul yang dipilih.

Tur dimulai biasanya beberapa tur pada tiga titik, misalnya mereka titik yang membentuk segitiga terbesar. Untuk masalah Euclidean, tur awal yang baik adalah tur yang mengikuti *convex hull* dari semua titik. Ini adalah pilihan yang wajar karena urutan titik dari tur *convex hull* dihormati dalam setiap tur yang optimal. Sebuah titik baru biasanya dimasukkan ke dalam tur pada titik yang menyebabkan kenaikan minimum panjang tur.

Perbedaan utama antara skema penyisipan urutan simpul dimasukkan :

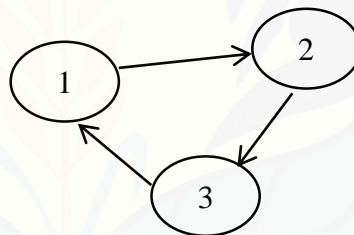
- a. *Farthest Insertion Heuristics*: Masukkan simpul yang memiliki jarak minimal ke simpul tur maksimal. Ide dibalik strategi ini adalah untuk memperbaiki tata letak keseluruhan tur pada awal proses penyisipan.
- b. *Cheapest/Nearest Insertion Heuristics*: Diantara semua simpul tidak dimasukkan sejauh ini, simpul yang dipilih adalah simpul dimana penyisipan menyebabkan kenaikan terendah dalam panjang keseluruhan dari tur.

2.5 Algoritma *Cheapest Insertion Heuristic*

Algoritma *Cheapest Insertion Heuristic* adalah algoritma *Insertion* yang pada setiap penambahan titik batu yang akan disisipkan ke dalam *subtour* mempunyai bobot penyisipan paling minimal. Bobot penyisipan diperoleh dari persamaan $c(i, k, j) = d(i, k) + d(k, j) - d(i, j)$. Algoritma ini memberikan rute perjalanan yang berbeda tergantung dari urutan penyisipan titik-titik pada *subtour* yang bersangkutan (Nilsson, 2003).

Berikut ini adalah algoritma CIH :

- Penelusuran dimulai dari sebuah titik pertama yang dihubungkan dengan sebuah titik terakhir.
- Buat sebuah hubungan *subtour* antara n pelanggan tersebut. Yang dimaksud *subtour* adalah perjalanan dari titik pertama dan berakhir di pelanggan pertama, misal terdapat 2 pelanggan $(1,3) \rightarrow (3,2) \rightarrow (2,1)$ seperti tergambar dalam Gambar 2.1.



Gambar 2.1 *Subtour*

- Ganti salah satu arah hubungan dari dua titik dengan kombinasi, yaitu C_{ij} dengan C_{ik} dan C_{kj} , dengan k diambil dari titik yang belum masuk *subtour* dan dengan tambahan jarak terkecil. Jarak diperoleh dari

$$c_{ik} + c_{kj} - c_{ij} \quad (2.10)$$

c_{ik} adalah jarak dari titiki ke titik k , c_{kj} adalah jarak dari titik k ke titik j dan

c_{ij} adalah jarak dari titik i ke titik j .

- Ulangi langkah 3 sampai seluruh titik masuk dalam *subtour*

Sebagai contoh diberikan 5 titik dengan jarak antar titik seperti berikut :

Tabel 2.1 Tabel jarak antar titik

Titik Asal	Titik Tujuan	Jarak
1	2	132
1	3	217
1	4	164
1	5	58
2	3	290
2	4	201
2	5	79
3	4	113
3	5	303
4	5	196

Untuk mencari jarak terpendek diantara ke 5 titik tersebut, ambil langkah-langkah berikut ini :

Ambil perjalanan terpendek dari tabel, dalam contoh kali ini diambil 1 ke 5

- Buat *subtour* (1,5) \rightarrow (5,1)
- Buat tabel yang menyimpan titik yang bisa disisipkan dalam *subtour* beserta jaraknya, seperti yang ditampilkan dalam Tabel 2.2.

Tabel 2.2 Tabel sisipan titik ke *subtour* (1,5)

Sisi yang akan diganti	Sisi yang ditambahkan ke <i>subtour</i>	Tambahan Jarak
(1,5)	(1,2) – (2,5)	$c_{12} + c_{25} - c_{15} = 153$
(1,5)	(1,3) – (3,5)	$c_{13} + c_{35} - c_{15} = 462$
(1,5)	(1,4) – (4,5)	$c_{14} + c_{45} - c_{15} = 302$
(5,1)	(5,2) – (2,1)	$c_{52} + c_{21} - c_{51} = 153$
(5,1)	(5,3) – (3,1)	$c_{53} + c_{31} - c_{51} = 462$
(5,1)	(5,4) – (4,1)	$c_{54} + c_{41} - c_{51} = 302$

C_{12} dan C_{25} atau dengan kata lain titik nomer 2 disisipkan diantara perjalanan dari titik 1 ke titik 5.

- Kemudian lanjutkan dengan membuat tabel baru yang berisi titik yang bisa disisipkan kedalam *subtour* (1,2) \rightarrow (2,5). Tabel sisipan berikutnya dapat dilihat pada Tabel 2.3

Tabel 2.3 tabel sisipan titik ke *subtour* berikutnya

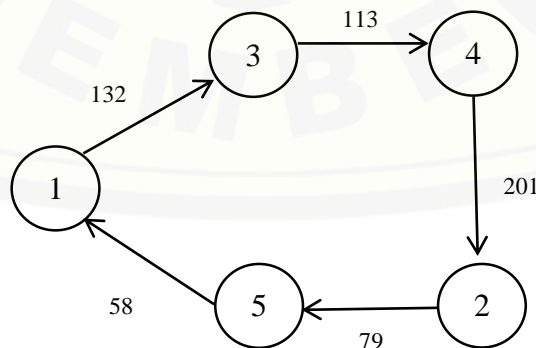
Sisi yang akan diganti	Sisi yang ditambahkan ke <i>subtour</i>	Tambahan Jarak
(1,2)	(1,3) – (3,2)	$c_{13} + c_{32} - c_{12} = 375$
(1,2)	(1,4) – (4,2)	$c_{14} + c_{42} - c_{12} = 233$
(2,5)	(2,3) – (3,5)	$c_{23} + c_{35} - c_{25} = 514$
(2,5)	(2,4) – (4,5)	$c_{24} + c_{45} - c_{25} = 318$
(5,1)	(5,3) – (3,1)	$c_{53} + c_{31} - c_{51} = 462$
(5,1)	(5,4) – (4,1)	$c_{54} + c_{41} - c_{51} = 302$

Dari Tabel 2.3, diperoleh tambahan jarak terkecil adalah 233, yaitu penambahan $C_{14} \rightarrow C_{42}$ ke dalam *subtour* sebelumnya. *Subtour* saat ini adalah (1,4) → (4,2) → (2,5) → (5,1). Karena masih ada titik yang belum masuk dalam perhitungan rute, maka dibuat tabel lagi yang berisi perjalanan dari dan ke titik yang belum disisipkan seperti pada Tabel 2.4 :

Tabel 2.4 Tabel sisipan titik terakhir

Sisi yang akan diganti	Sisi yang ditambahkan ke <i>subtour</i>	Tambahan Jarak
(1,4)	(1,3) – (3,4)	$c_{13} + c_{34} - c_{14} = 166$
(4,2)	(1,3) – (3,2)	$c_{13} + c_{32} - c_{42} = 202$
(2,5)	(2,3) – (3,5)	$c_{23} + c_{35} - c_{25} = 514$
(5,1)	(5,3) – (3,1)	$c_{53} + c_{31} - c_{51} = 462$

Dari Tabel 2.4 dapat diperoleh tambahan jarak terkecil dengan menambahkan $C_{13} \rightarrow C_{34}$. Dari langkah-langkah diatas diperoleh rute dengan jarak terpendek untuk mengunjungi keseluruhan titik dengan *subtour* (1,3) → (3,4) → (4,2) → (2,5) → (5,1) seperti ditunjukkan pada Gambar 2.2 berikut ini :



Gambar 2.2 Lintasan terpendek kelima titik

Dari lintasan terpendek tersebut dapat diperoleh jarak tempuh totalnya adalah

$$c_{13} + c_{34} + c_{42} + c_{25} + c_{51} = 132 + 113 + 201 + 79 + 58 = 668$$

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang akan digunakan dalam penelitian ini adalah data primer yang diambil dari perusahaan yang mendistribusikan beras kemasan “Salsa Muliya” Jember, dengan beberapa kendaraan dimana setiap kendaraan berkapasitas berbeda. Dan memiliki jumlah pelanggan yang mempunyai permintaan berbeda-beda. Dengan jarak antara pelanggan dan depot yang berbeda, serta lokasi depot dan pelanggan yang berbeda. Data lokasi depot dan pelanggan berupa koordinat geografis yaitu *latitude* (y) dan *longitude* (x) pada google maps. Data selengkapnya ada di lampiran.

3.2 Langkah-Langkah Penelitian

Langkah-langkah yang akan dilakukan pada penelitian ini adalah sebagai berikut:

a. Studi Literatur

Studi literatur yang dilakukan pada penelitian ini adalah mengumpulkan literatur untuk mengetahui dan memahami tentang *Capacitated Vehicle Routing Problem*, *Sweep Clustering* dan algoritma *Cheapest Insertion Heuristic* yang didapat dari berbagai sumber berupa buku, artikel dan jurnal dari internet.

b. Pengambilan Data

Pada tahap ini, dilakukan pengambilan dan pengumpulan data-data yang dibutuhkan dalam penerapan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* pada *Capacitated Vehicle Routing Problem* adalah data primer dengan melakukan wawancara, yaitu pada perusahaan yang mendistribusikan beras kemasan “Salsa Muliya” Jember .

c. Penerapan Algoritma

Pada tahap ini, suatu data dipilih kemudian dibuat beberapa *cluster* dengan menggunakan *Sweep Clustering*. Proses dari *Sweep Clustering* adalah sebagai berikut:

- 1) Merepresentasikan data lokasi depot dan pelanggan dalam koordinat polar dengan depot sebagai pusat koordinat;
- 2) Membuat *radial line* dan menentukan arah perputarannya;
- 3) Memperbesar sudut *radial line* hingga mencakup pelanggan pertama;
- 4) Memperbesar sudut *radial line* hingga pelanggan-pelanggan yang tercakup memiliki total permintaan kurang dari atau sama dengan kapasitas kendaraan (terbentuk satu *cluster*);
- 5) Mengulangi langkah 3 dan 4 hingga membentuk beberapa *cluster* yang mencakup semua pelanggan.

Masing-masing *cluster* kemudian dicari rute jarak minimum antar pelanggan dengan menggunakan algoritma *Cheapest Insertion Heuristic*. Proses dari algoritma *cheapest Insertion Heuristic* adalah sebagai berikut :

- 1) Pada satu *cluster* membuat *subtour* awal dari depot ke satu pelanggan kemudian kembali ke depot;
- 2) Menghitung bobot *subtour* awal, pilih bobot terkecil;
- 3) Apabila pada satu *cluster* masih ada pelanggan yang belum terbentuk *subtour*. Maka, pelanggan yang belum terbentuk disisipkan pada *subtour* yang terpilih sebelumnya;
- 4) Menghitung bobot *subtour* yang telah disisipkan, pilih bobot terkecil;
- 5) Mengulangi langkah 3 dan 4 sampai tidak ada pelanggan dalam satu *cluster* yang belum terbentuk *subtour*;
- 6) Menghasilkan rute pada *subtour* yang telah dicari;
- 7) Mengulangi langkah 1 sampai langkah 6 untuk *cluster-cluster* selanjutnya hingga masing-masing *cluster* mempunyai rute terpendek;

d. Pembuatan Program

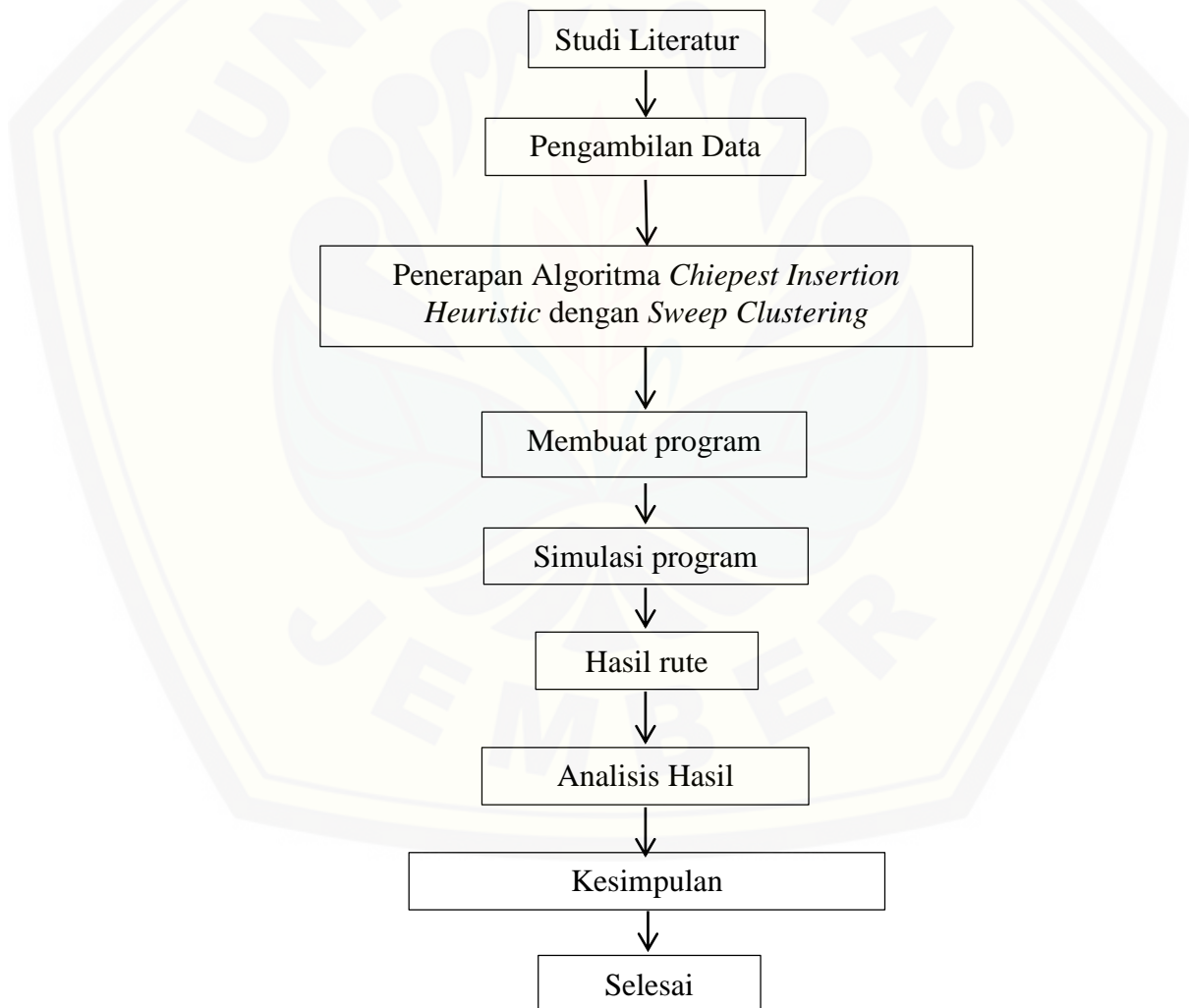
Pembuatan program merupakan langkah pembuatan sarana pembantu yang berupa suatu aplikasi untuk menyelesaikan permasalahan. Pembuatan aplikasi ini dilakukan menggunakan *software* MATLAB.

e. Analisis Hasil

Analisis Hasil, dilakukan perbandingan dan pembahasan hasil yang diperoleh dari penerapan *sweep clustering* dan algoritma *Cheapest Insertion Heuristic* untuk menyelesaikan *Capacitated Vehicle Routing Problem* pada kasus distribusi beras. Hasil yang dibandingkan adalah dengan menentukan rute terpendek dengan hanya menggunakan algoritma *Cheapest Insertion Heuristic*.

f. Kesimpulan

Hasil dari langkah e akan ditarik kesimpulan yang akan menjawab rumusan masalah.



Gambar 3.1 Skema langkah-langkah penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dilakukan pada data terlampir, dapat disimpulkan sebagai berikut:

1. *Capacitated Vehicle Routing Problem* menggunakan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* dan menggunakan algoritma *Cheapest Insertion Heuristic* juga dengan bantuan program MATLAB dapat meminimalkan jarak dari depot ke pelanggan. Dengan menggunakan data 7 titik hingga menggunakan data 30 titik.
2. Berdasarkan hasil dan pembahasan yang sudah dilakukan, rute optimal yang dihasilkan adalah pada algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering*, karena secara umum jarak total minimal paling banyak didapatkan dengan menggunakan algoritma tersebut daripada jarak total minimal dengan menggunakan algoritma *Cheapest Insertion Heuristic*. Demikian bisa disimpulkan bahwa pada permasalahan *Capacitated Vehicle Routing Problem* menggunakan algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* lebih baik daripada hanya menggunakan algoritma *Cheapest Insertion Heuristic*.

5.2 Saran

Berdasarkan dari penelitian yang sudah dilakukan dengan menggunakan data simetris, penulis menyarankan agar pada penelitian selanjutnya menggunakan data yang asimetris. Algoritma *Cheapest Insertion Heuristic* dengan *Sweep Clustering* dan algoritma *Cheapest Insertion Heuristic* belum sepenuhnya optimal karena masih ada titik yang hasilnya berbeda, sehingga untuk peneliti selanjutnya menerapkan *Capacitated Vehicle Routing Problem* dengan memodifikasi *Sweep Clustering* agar pada satu *cluster* tidak terdapat jarak yang jauh antar titik meskipun dengan sudut terkecil.

DAFTAR PUSTAKA

- Boonkleaw, A., S. Suthikannarunai dan R. Srinon. 2009. Strategic Planning and Vehicle Routing Algorithm for Newspaper Delivery Problem : Case Study of Morning Newspaper, Bangkok, Thailand. *Proceeding of the World Congress on Engineering and Computer Science*, USA : San francisco. 2.
- Charles, S. N. dan R. A. Sarker. 2008. *Optimization Modelling : A Practical Introduction*. USA: New York
- Gunawan, I., Maryati dan H. K. Wibowo. 2012. Optimasi Penentuan Rute Kendaraan pada Sistem Distribusi Barang dengan Ant Colony Optimization. *Jurnal Seminar Nasional Teknologi Informasi dan Komunikasi Terapan*.25: 1-6.
- Kallehauge, B., J. Larsen dan O. B. G. Madsen. 2001. Lagrangean Duality Applied to Vehicle Routing Problem with Time Window. *Informatic and Mathematical Modelling*. 33(5):1464-1487
- Nilson C. 2003. Heuristic for the Traveling Salesman Problem. *Computer Science*. Swedia:Linkoping University
- Prana, R. 2007. Aplikasi Kombinatorial pada Vehicle Routing Problem. *Jurnal*. Bandung: Jurusan Teknik Informatika ITB.
- Putri, R. A. 2014. Penyelesaian Capacitated Vehicle Routing Problem dengan metode Saving Heuristic. *Skripsi*. Jember: Universitas Jember.
- Saraswati, R., W. Sutopo, dan M. Hisjam. 2017. Penyelesaian Capacitated Vehicle Routing Problem dengan menggunakan Algoritma Sweep untuk Penentuan Rute Distribusi Koran. *Jurnal Manajemen Pemasaran*. 10:11-2.
- Yansyah, M. D. 2016. Aplikasi Algoritma Cheapest Insertion Heuristic (CIH) with Saving Method dalam kasus Vehicle Routing Problem with Time Window (VRPTW) pada Pengangkutan Sampah. *Skripsi*. Jember: Universitas Jember.

LAMPIRAN

A. Lokasi Depot dan Pelanggan

No	Nama Toko	Alamat
0	Depot	Jl. Jember ambululo no 9, Krajan Selatan , Kertonegoro
1	Wiwik Handayani	Dsn Mandilis wetan Desa Senenrejo rt2 rw2
2	Umbar Wiyono	Dsn Curah Rejo – Curah Takir
3	Badrui Tamam	Jl Cendrawasih Dsn Punco Desa Curah Takir
4	Moh Solikin	Jl glantangan – pondok kemiri kidul
5	agen 46	Jl argopuro – mandiku
6	Eva Muafiqoh	Jl kartini – sidodadi
7	Andi Wawan Anjaya	Pondok rejo – tempurejo
8	Huzaimati	Jl Yos Sudarso No 242 Langsepan. Kranjingan
9	Pasti Jaya	Jl MT Haryono no 12 Gg Demak wirolegi kaangbaru
10	Nur Hasan	Jl Letjen S. Parman Gg 4 kali oktak
11	Sri Hartutik	Jl Tawang Mangu No 14 A Tegal Gede
12	Flamboyant	Pondok Jambuan Antirogo
13	Maria Budi S	Jl Madura 55 – gumuk kerang
14	Zaenuri	Jl Tawan Mangu no 9B lingkaran panji
15	Muhtar	Jl KH Wahid Hasyim B 40 kebondalem
16	Ratna Windari	Jl Basuki Rahmat no 214 muktisari - Tegal Besar
17	Kop.Al Baim Aien	Jl Sultan Agung no. 2 kepatihan
18	Ibnu Sulyanto	Jl Sentot Prawirodirjo 43 kaliwates kidul
19	AM Sulistyoy	Jl Cadika 8/9 Rt3 Rw12 Sempusari
20	Ali Abdillah	Jl KH Agus Salim- tegal besar kulon
21	Vauziah Nur Hikmah	Jl Hayam Wuruk I/44
22	Supini	Jl Arwana 15 - gebang taman
23	Mbak Liss	Dsn Krajan Kidul Desa Sumberrejo
24	Elok Kusumawati	Bregoh - sumberejo
25	Rindang	Krajan Lor rt1 rw 2 sumberejo
26	Ar-rum	Jl mawar rt 3 rw 6 kebonsari sabrang
27	Sumber Alam	jl kota blater no 10 pontang, pontang krajan
28	Glory	jl Glory no. 66 II Glory, sumberan
29	Gentong berkah	jl candradimuka no 45, dusun sentong 2/19 karanganyar
30	p. ngadi	jl raya watu ulo, Dusun watu ulo, sumber rejo

B. Koordinat Lokasi Pelanggan

No	Latitude	Longitude	No	Latitude	Longitude
0	-8,301862	113,623202	16	-8,204981	113,696867
1	-8,368001	113,797920	17	-8,168964	113,700303
2	-8,336503	113,751247	18	-8,181284	113,683689
3	-8,357704	113,768312	19	-8,175537	113,663989
4	-8,355071	113,670548	20	-8,186432	113,690148
5	-8,366054	113,666142	21	-8,183675	113,670092
6	-8,364501	113,660068	22	-8,165165	113,676251
7	-8,33158	113,696347	23	-8,390133	113,577921
8	-8,202754	113,732874	24	-8,413901	113,595258
9	-8,186488	113,729563	25	-8,386974	113,546200
10	-8,183902	113,721992	26	-8,3664467	113,606173
11	-8,15555	113,720901	27	-8,3609736	113,628356
12	-8,157426	113,733927	28	-8,3457835	113,607903
13	-8,178483	113,708671	29	-8,3343229	113,609739
14	-8,152126	113,727888	30	-8,4190774	113,570895
15	-8,178276	113,698012			

C. Permintaan Pelanggan

No	Permintaan	No	Permintaan	No	Permintaan
1	60	11	20	21	200
2	50	12	400	22	200
3	129	13	150	23	250
4	149	14	150	24	323
5	70	15	175	25	443
6	190	16	120	26	485
7	50	17	100	27	300
8	100	18	100	28	100
9	200	19	350	29	350
10	50	20	200	30	100

D. Data Jarak Antara Setiap Lokasi

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

**E. Script Program Capacitated Vehicle Routing Problem menggunakan Algoritma
Cheapest Insertion Heuristic**

```

function [Route, QRoute, DRoute] = CIH(Data_Dist, Data_Demand, Capacity)
% Cheapest Insertion Heuristic
% Capacitated Vehicle Routing Problem
% 1. Data_Dist is data distance between 2 nodes. This input must be a matrix
% (N+1) x (N+1), where N is the number of customers.
% 2. Data_Demand is number of customer demand. This input must be a column
% vector. Demand of Depot must be included which value is zero (0).
% 3. Capacity is the capacity of the vehicle.
N = size(Data_Dist,1)-1; %Number of customers
Node = 1:N; %Set of nodes
%Initial Route
Route = [];
QRoute = [];
DRoute = [];
%Initial sub tour
Cijk = []; Sub_Tour = [];
for i = 1:length(Node)
    Cijk(i) = Data_Dist(1,Node(i)+1) + Data_Dist(Node(i)+1,1);
end
id = find(Cijk==min(Cijk));
Sub_Tour = [0 Node(id(1)) 0];
SumQ = Data_Demand(Node(id(1))+1);
Cij = min(Cijk);
Node = setxor(Node, Node(id(1)));
k=1;
while ~isempty(Node)
    %Insertion
    a = 1; Temp = []; Cijk = [];
    for i = 2:length(Sub_Tour)
        for j = 1:length(Node)
            Temp(a,:) = [Sub_Tour(1:i-1) Node(j) Sub_Tour(i:end)];
            Cijk(a) = Cij - Data_Dist(Sub_Tour(i-1)+1,Sub_Tour(i)+1) + ...
                Data_Dist(Sub_Tour(i-1)+1,Node(j)+1) +
                Data_Dist(Node(j)+1,Sub_Tour(i)+1);
            a = a+1;
        end
    end
    %Find minimum Cij
    id = find(Cijk==min(Cijk));
    NP = setxor(Sub_Tour, Temp(id(1),:));
    %Calculate Demand
    TQ = SumQ + Data_Demand(NP+1);
    if TQ > Capacity %Sum of Demand exceed the Capacity
        Route{k} = Sub_Tour;
        QRoute(k) = SumQ;
        DRoute(k) = Cij;
        k = k+1;
        %Build new sub tour
        Cijk = []; Sub_Tour = [];
        for i = 1:length(Node)
            Cijk(i) = Data_Dist(1,Node(i)+1) + Data_Dist(Node(i)+1,1);
        end
        id = find(Cijk==min(Cijk));
        Sub_Tour = [0 Node(id(1)) 0];
        SumQ = Data_Demand(Node(id(1))+1);
        Cij = min(Cijk);
        Node = setxor(Node, Node(id(1)));
    end
end

```

```

else
    Sub_Tour = Temp(id(1),:);
    Cij = min(Cijk);
    SumQ = TQ;
    Node = setxor(Node, NP);
end
end
end

Num_Node = 0;
Num_Route = size(Route,2);
for k = 1:Num_Route
    Num_Node = Num_Node+length(Route{k})-2;
end
if Num_Node~=N
    Route{Num_Route+1} = Sub_Tour;
    QRoute(Num_Route+1) = SumQ;
    DRoute(Num_Route+1) = Cij;
End

```

F. Script Program Capacitated Vehicle Routing Problem menggunakan Algoritma Cheapest Insertion Heuristic dengan Sweep Clustering

```

function [Route, QRoute, DRoute] = CIH_Sweep(Data_Coor, Data_Dist, Data_Demand, Capacity, Direct)
% Cheapest Insertion Heuristic with Sweep Clustering
% Capacitated Vehicle Routing Problem
% 1. Data_Coor is data Latitude Longitude Coordinates of every node (Depot and Customers). This input must be a matrix with 2 columns (latitude and longitude).
% 2. Data_Dist is data distance between 2 nodes. This input must be a matrix (N+1) x (N+1), where N is the number of customers.
% 3. Data_Demand is number of customer demand. This input must be a column vector. Demand of Depot must be included which value is zero (0).
% 4. Capacity is the capacity of the vehicle.
% 5. Direct is a direction of sweeping clustering. (1) Forward/ counterclockwise, (0) Backward/clockwise
if nargin == 4 %CIH_Sweep(Data_Coor, Data_Dist, Data_Demand, Capacity)
    Direct = 1;
end
N = size(Data_Dist,1)-1; %Number of customers
%Calculate theta
for i = 1:N
    x = Data_Coor(i+1,2) - Data_Coor(1,2);
    y = Data_Coor(i+1,1) - Data_Coor(1,1);
    if x > 0 && y > 0 %Quadrant 1
        Theta(i) = atand(y/x);
    elseif x < 0 && y > 0 %Quadrant 2
        Theta(i) = 180 + atand(y/x);
    elseif x < 0 && y < 0 %Quadrant 3
        Theta(i) = 180 + atand(y/x);
    elseif x > 0 && y < 0 %Quadrant 4
        Theta(i) = 360 + atand(y/x);
    elseif y == 0 %x axis
        if x >= 0
            Theta(i) = 0;
        else
            Theta(i) = 180;
        end
    elseif x == 0 %y axis
        if y > 0
            Theta(i) = 90;

```

```

        else
            Theta(i) = 270;
        end
    end
end
%Sorting theta & clustering
if Direct == 1
    [rTheta, idx] = sort(Theta, 'ascend');
elseif Direct == 0
    [rTheta, idx] = sort(Theta, 'descend');
end
k = 1; a = 1; all = 0;
for i = 1:N
    SumQ = sum(Data_Demand(idx(a:i)+1));
    if SumQ > Capacity
        Cluster{k} = idx(a:i-1);
        QRoute(k) = sum(Data_Demand(idx(a:i-1)+1));
        k = k+1;
        a = i;
        all = all + length(Cluster{k-1});
    end
end
if all < N
    Cluster{k} = idx(a:N);
    QRoute(k) = sum(Data_Demand(idx(a:N)+1));
end
K=size(Cluster,2); %Number of vehicles
for k = 1:K
    Node = Cluster{k};
    %Initial sub tour
    Cijk = []; Sub_Tour = [];
    for i = 1:length(Node)
        Cijk(i) = Data_Dist(1,Node(i)+1) + Data_Dist(Node(i)+1,1);
    end
    id = find(Cijk==min(Cijk));
    Sub_Tour = [0 Node(id(1)) 0];
    Cij = min(Cijk);
    Node = setxor(Node, Node(id(1)));
    while ~isempty(Node)
        %Insertion
        a = 1; Temp = []; Cijk = [];
        for i = 2:length(Sub_Tour)
            for j = 1:length(Node)
                Temp(a,:) = [Sub_Tour(1:i-1) Node(j) Sub_Tour(i:end)];
                Cijk(a) = Cij - Data_Dist(Sub_Tour(i-1)+1,Sub_Tour(i)+1) + ...
                    Data_Dist(Sub_Tour(i-1)+1,Node(j)+1) +
                    Data_Dist(Node(j)+1,Sub_Tour(i)+1);
                a = a+1;
            end
        end
        %Find minimum Cij
        id = find(Cijk==min(Cijk));
        NP = setxor(Sub_Tour, Temp(id(1),:));
        Sub_Tour = Temp(id(1),:);
        Cij = min(Cijk);
        Node = setxor(Node, NP);
    end
    %Route
    Route{k} = Sub_Tour;
    DRoute(k) = Cij;
end
End

```


G. Total jarak tempuh data dengan menggunakan Program

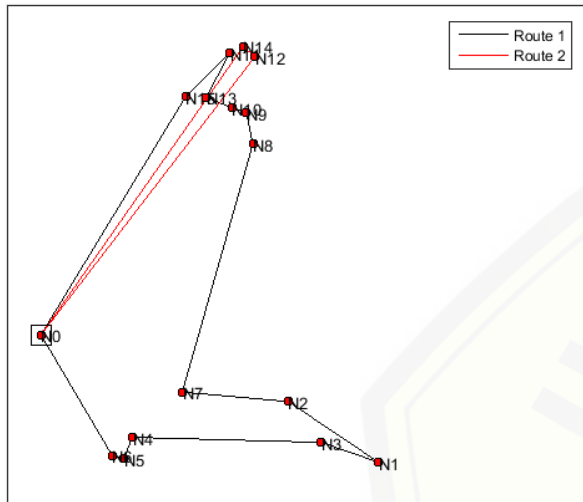
KAPASITAS KENDARAAN 1500 KARUNG

Titik	<i>Cheapest Insertion Heuristic</i>	<i>Cheapest Insertion Heuristic Sweep Clustering forward</i>	<i>Cheapest Insertion Heuristic Sweep Clustering backward</i>
7	Total Jarak Tempuh = 75.5 km • Rute 1 : 0-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 75.5 km • Rute 1 : 0-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 75.5 km • Rute 1 : 0-7-2-1-3-4-5-6-0
8	Total Jarak Tempuh = 99.8 km • Rute 1 : 0-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 99.8 km • Rute 1 : 0-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 99.8 km • Rute 1 : 0-8-7-2-1-3-4-5-6-0
9	Total Jarak Tempuh = 105.4 km • Rute 1 : 0-8-9-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 105.4 km • Rute 1 : 0-8-9-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 105.4 km • Rute 1 : 0-8-9-7-2-1-3-4-5-6-0
10	Total Jarak Tempuh = 105.9 km • Rute 1 : 0-10-9-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 105.9 km • Rute 1 : 0-10-9-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 105.9 km • Rute 1 : 0-10-9-8-7-2-1-3-4-5-6-0
11	Total Jarak Tempuh = 111.9 km • Rute 1 : 0-11-10-9-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 111.9 km • Rute 1 : 0-11-10-9-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 111.9 km • Rute 1 : 0-11-10-9-8-7-2-1-3-4-5-6-0
12	Total Jarak Tempuh = 115.9 km • Rute 1 : 0-11-12-10-9-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 115.9 km • Rute 1 : 0-11-12-10-9-8-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 115.9 km • Rute 1 : 0-11-12-10-9-8-7-2-1-3-4-5-6-0
13	Total Jarak Tempuh = 164.1 km • Rute 1 : 0-13-11-10-9-8-7-2-1-3-4-5-6-0 • Rute 2 : 0-12-0	Total Jarak Tempuh = 146.3 km • Rute 1 : 0-13-11-12-10-9-8-7-4-5-6-0 • Rute 2 : 0-2-1-3-0	Total Jarak Tempuh = 161.6 km • Rute 1 : 0-13-11-12-10-7-2-1-3-4-5-6-0 • Rute 2 : 0-9-8-0
14	Total Jarak Tempuh = 165.7 km • Rute 1 : 0-13-11-14-10-9-8-7-2-1-3-4-5-6-0 • Rute 2 : 0-12-0	Total Jarak Tempuh = 147.4 km • Rute 1 : 0-13-11-14-12-10-9-8-4-5-6-0 • Rute 2 : 0-1-3-2-7-0	Total Jarak Tempuh = 163.2 km • Rute 1 : 0-13-11-14-12-10-7-2-1-3-4-5-6-0 • Rute 2 : 0-9-8-0
15	Total Jarak Tempuh = 168.4 km • Rute 1 : 0-15-11-13-10-9-8-7-2-1-3-4-5-6-0 • Rute 2 : 0-12-14-0	Total Jarak Tempuh = 152.8 km • Rute 1 : 0-15-11-14-12-13-10-9-8-6-0 • Rute 2 : 0-5-4-1-3-2-7-0	Total Jarak Tempuh = 167.8 km • Rute 1 : 0-15-11-14-13-7-2-1-3-4-5-6-0 • Rute 2 : 0-12-10-9-8-0
16	Total Jarak Tempuh = 183.8 km • Rute 1 : 0-16-15-11-14-13-10-9-8-7-4-5-6-0 • Rute 2 : 0-2-1-3-12-0	Total Jarak Tempuh = 138.4 km • Rute 1 : 0-15-11-14-12-9-8-10-13-16-0 • Rute 2 : 0-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 167.9 km • Rute 1 : 0-15-11-14-13-16-7-2-3-1-4-5-6-0 • Rute 2 : 0-12-10-9-8-0
17	Total Jarak Tempuh = 185.3 km • Rute 1 : 0-16-15-17-11-13-10-9-8-7-4-5-6-0 • Rute 2 : 0-2-1-3-12-14-0	Total Jarak Tempuh = 139.9 km • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-7-2-1-3-4-5-6-0	Total Jarak Tempuh = 168.6 km • Rute 1 : 0-15-17-11-14-13-16-7-2-3-1-4-5-6-0 • Rute 2 : 0-12-10-9-8-0

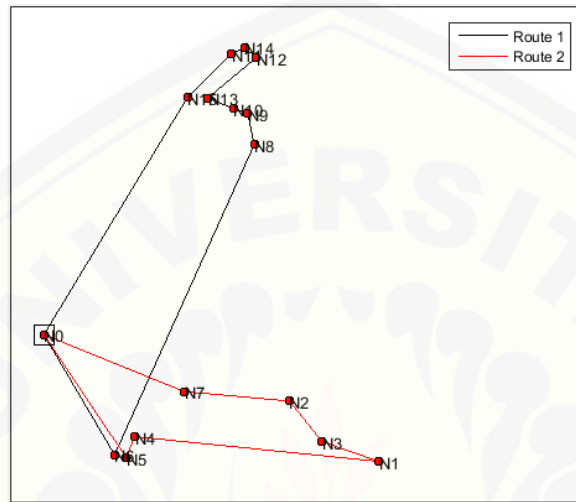
18	Total Jarak Tempuh = 189.3 km <ul style="list-style-type: none"> • Rute 1 : 0-18-15-17-11-13-9-10-8-16-7-4-5-6-0 • Rute 2 : 0-2-1-3-12-14-0 	Total Jarak Tempuh = 164.9 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-18-7-2-1-3-4-5-6-0 	Total Jarak Tempuh = 167.8 km <ul style="list-style-type: none"> • Rute 1 : 0-18-15-17-11-14-13-7-2-1-3-4-5-6-0 • Rute 2 : 0-8-9-12-10-16-0
19	Total Jarak Tempuh = 180.2 km <ul style="list-style-type: none"> • Rute 1 : 0-19-18-15-17-13-16-7-4-5-6-0 • Rute 2 : 0-11-14-12-10-9-8-2-1-3-0 	Total Jarak Tempuh = 167.5 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-19-18-7-2-1-3-4-5-6-0 	Total Jarak Tempuh = 172.7 km <ul style="list-style-type: none"> • Rute 1 : 0-19-18-15-17-11-7-2-1-3-4-5-6-0 • Rute 2 : 0-13-14-12-9-10-8-16-0
20	Total Jarak Tempuh = 179.6 km <ul style="list-style-type: none"> • Rute 1 : 0-19-18-15-20-16-7-4-5-6-0 • Rute 2 : 0-17-11-14-12-13-10-9-8-2-3-1-0 	Total Jarak Tempuh = 167.5 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-19-18-20-7-2-1-3-4-5-6-0 	Total Jarak Tempuh = 170.1 km <ul style="list-style-type: none"> • Rute 1 : 0-19-18-20-17-7-2-1-3-4-5-6-0 • Rute 2 : 0-15-11-14-12-9-8-10-13-16-0
21	Total Jarak Tempuh = 198.7 km <ul style="list-style-type: none"> • Rute 1 : 0-21-18-17-15-20-16-7-4-5-6-0 • Rute 2 : 0-8-13-10-9-12-14-11-19-0 • Rute 3 : 0-2-1-3-0 	Total Jarak Tempuh = 215.1 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-19-21-18-20-7-1-3-4-5-6-0 • Rute 3 : 0-2-0 	Total Jarak Tempuh = 209.8 km <ul style="list-style-type: none"> • Rute 1 : 0-19-21-18-7-2-1-3-4-5-6-0 • Rute 2 : 0-20-15-17-11-14-12-13-16-0 • Rute 3 : 0-9-8-0
22	Total Jarak Tempuh = 230 km <ul style="list-style-type: none"> • Rute 1 : 0-21-18-17-15-20-16-7-4-5-6-0 • Rute 2 : 0-19-22-11-14-9-10-13-0 • Rute 3 : 0-1-3-2-8-12-0 	Total Jarak Tempuh = 200.8 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-19-22-21-18-20-4-5-6-0 • Rute 3 : 0-1-3-2-7-0 	Total Jarak Tempuh = 215.8 km <ul style="list-style-type: none"> • Rute 1 : 0-19-22-21-7-2-1-3-4-5-6-0 • Rute 2 : 0-18-17-11-14-12-13-10-15-20-16-0 • Rute 3 : 0-9-8-0
23	Total Jarak Tempuh = 241 km <ul style="list-style-type: none"> • Rute 1 : 0-21-18-20-16-7-4-5-6-23-0 • Rute 2 : 0-19-22-17-11-14-13-9-10-15-0 • Rute 3 : 0-1-3-2-8-12-0 	Total Jarak Tempuh = 218.4 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-23-6-19-22-20-18-21-0 • Rute 3 : 0-5-4-1-3-2-7-0 	Total Jarak Tempuh = 230.1 km <ul style="list-style-type: none"> • Rute 1 : 0-19-22-21-7-2-1-3-4-5-6-23-0 • Rute 2 : 0-16-20-15-13-14-11-17-18-21-0 • Rute 3 : 0-12-10-9-8-0
24	Total Jarak Tempuh = 249.1 km <ul style="list-style-type: none"> • Rute 1 : 0-18-20-16-7-4-5-6-23-24-0 • Rute 2 : 0-19-22-17-11-13-9-10-15-21-0 • Rute 3 : 0-1-3-2-8-12-14-0 	Total Jarak Tempuh = 224.9 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-21-18-20-22-19-23-0 • Rute 3 : 0-7-2-3-1-4-5-6-24-0 	Total Jarak Tempuh = 206.9 km <ul style="list-style-type: none"> • Rute 1 : 0-7-2-1-3-4-5-6-23-24-0 • Rute 2 : 0-19-22-17-11-13-15-20-18-21-0 • Rute 3 : 0-8-9-10-12-14-16-0
25	Total Jarak Tempuh = 229.8 km <ul style="list-style-type: none"> • Rute 1 : 0-7-4-5-6-25-23-24-0 • Rute 2 : 0-19-17-13-10-15-20-18-21-0 • Rute 3 : 0-1-3-2-8-9-12-14-11-22-16-0 	Total Jarak Tempuh = 234.2 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-7-2-3-1-4-5-6-23-24-0 	Total Jarak Tempuh = 234.2 km <ul style="list-style-type: none"> • Rute 1 : 0-7-2-1-3-4-5-6-23-24-0 • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-15-17-11-14-12-13-9-10-8-16-0
26	Total Jarak Tempuh = 280.8 km <ul style="list-style-type: none"> • Rute 1 : 0-23-24-26-0 	Total Jarak Tempuh = 256.8 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 	Total Jarak Tempuh = 237.3 km <ul style="list-style-type: none"> • Rute 1 : 0-7-2-1-3-4-5-6-24-0

	<ul style="list-style-type: none"> • Rute 3 : 0-18-20-16-7-4-5-6-25-0 • Rute 2 : 0-19-22-17-11-13-9-10-15-21-0 • Rute 3 : 0-1-3-2-8-12-14-0 	<ul style="list-style-type: none"> • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-23-24-26-5-6-4-0 • Rute 4 : 0-1-3-2-7-0 	<ul style="list-style-type: none"> • Rute 2 : 0-25-23-26-0 • Rute 3 : 0-19-22-17-11-13-15-20-18-21-0 • Rute 4 : 0-8-9-10-12-14-16-0
27	Total Jarak Tempuh = 294.4 km <ul style="list-style-type: none"> • Rute 1 : 0-27-5-4-6-26-23-0 • Rute 3 : 0-25-24-7-16-20-18-21-0 • Rute 2 : 0-19-22-17-11-14-13-9-10-15-0 • Rute 3 : 0-1-3-2-8-12-0 	Total Jarak Tempuh = 251.6 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-27-26-24-23-0 • Rute 4 : 0-7-2-1-3-4-5-6-0 	Total Jarak Tempuh = 238.3 km <ul style="list-style-type: none"> • Rute 1 : 0-24-27-5-6-4-1-3-2-7-0 • Rute 2 : 0-25-23-26-0 • Rute 3 : 0-19-22-17-11-13-15-20-18-21-0 • Rute 4 : 0-8-9-10-12-14-16-0
28	Total Jarak Tempuh = 288.8 km <ul style="list-style-type: none"> • Rute 1 : 0-26-6-4-5-27-28-0 • Rute 3 : 0-18-20-16-7-24-23-25-0 • Rute 2 : 0-19-22-17-11-13-9-10-15-21-0 • Rute 3 : 0-1-3-2-8-12-14-0 	Total Jarak Tempuh = 251.5 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-24-23-26-27-28-0 • Rute 4 : 0-7-2-1-3-4-5-6-0 	Total Jarak Tempuh = 238.6 km <ul style="list-style-type: none"> • Rute 1 : 0-24-27-5-6-4-1-3-2-7-0 • Rute 2 : 0-25-23-26-28-0 • Rute 3 : 0-19-22-17-11-13-15-20-18-21-0 • Rute 4 : 0-8-9-10-12-14-16-0
29	Total Jarak Tempuh = 272.8 km <ul style="list-style-type: none"> • Rute 1 : 0-26-6-5-27-28-29-0 • Rute 3 : 0-16-7-4-25-23-24-0 • Rute 2 : 0-19-17-13-10-15-20-18-21-0 • Rute 3 : 0-22-11-14-12-9-8-2-3-1-0 	Total Jarak Tempuh = 258.2 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-26-23-28-29-0 • Rute 4 : 0-24-27-5-6-4-1-3-2-7-0 	Total Jarak Tempuh = 258.2 km <ul style="list-style-type: none"> • Rute 1 : 0-24-27-5-6-4-1-3-2-7-0 • Rute 2 : 0-26-23-28-29-0 • Rute 3 : 0-21-18-20-22-19-25-0 • Rute 4 : 0-15-17-11-14-12-13-9-10-8-16-0
30	Total Jarak Tempuh = 277.5 km <ul style="list-style-type: none"> • Rute 1 : 0-26-6-5-27-28-29-0 • Rute 3 : 0-16-7-4-25-23-30-24-0 • Rute 2 : 0-19-17-13-10-15-20-18-21-0 • Rute 3 : 0-22-11-14-12-9-8-2-3-1-0 	Total Jarak Tempuh = 264.7 km <ul style="list-style-type: none"> • Rute 1 : 0-15-17-11-14-12-13-9-10-8-16-0 • Rute 2 : 0-21-18-20-22-19-25-0 • Rute 3 : 0-26-30-23-28-29-0 • Rute 4 : 0-24-27-5-6-4-1-3-2-7-0 	Total Jarak Tempuh = 264.7 km <ul style="list-style-type: none"> • Rute 1 : 0-24-27-5-6-4-1-3-2-7-0 • Rute 2 : 0-26-30-23-28-29-0 • Rute 3 : 0-21-18-20-22-19-25-0 • Rute 4 : 0-15-17-11-14-12-13-9-10-8-16-0

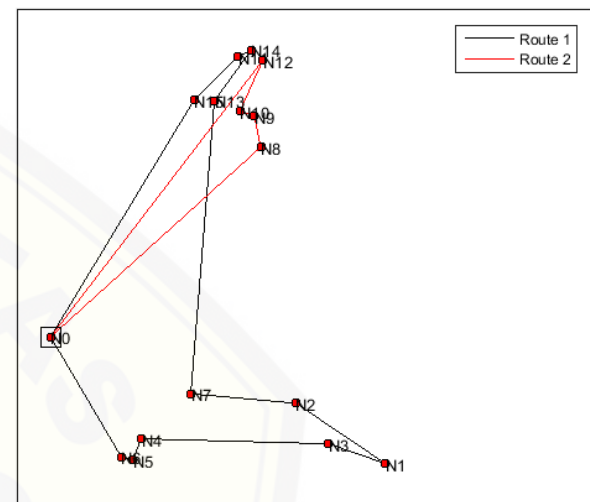
15 titik



CIH

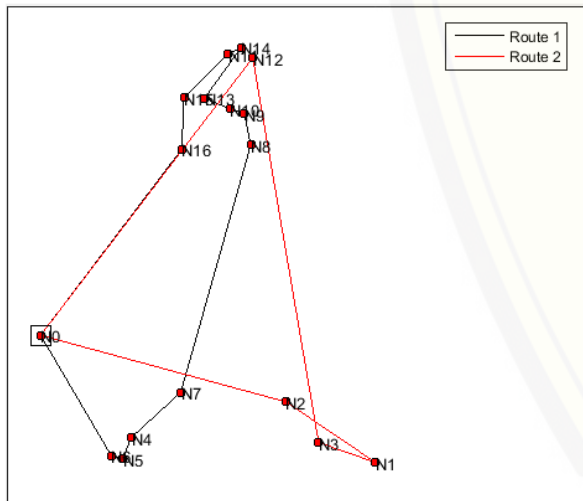


CSF

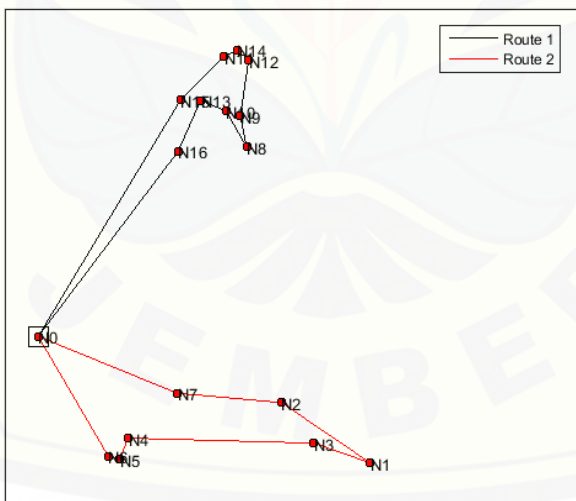


CSB

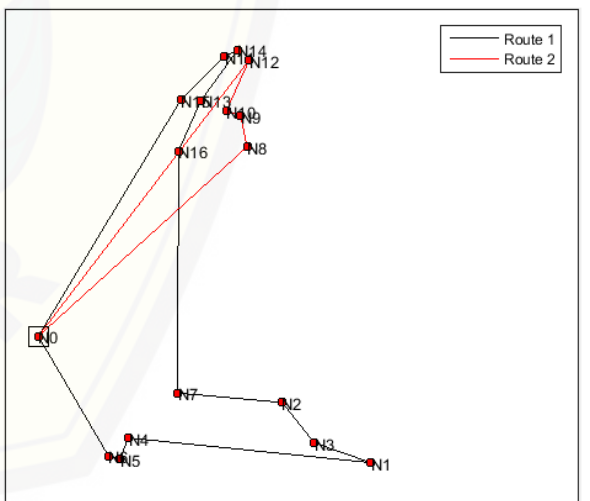
16 titik



CIH

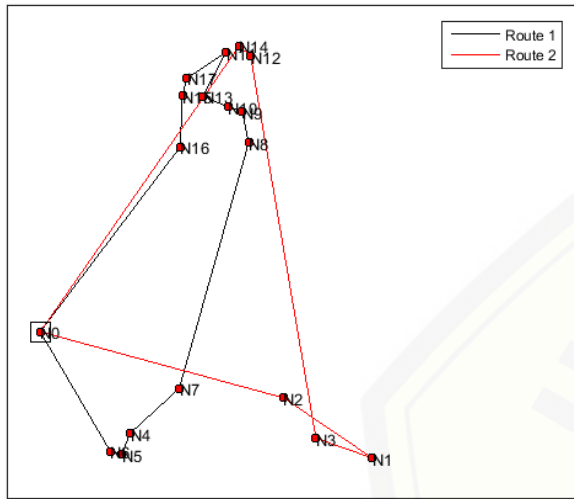


CSF

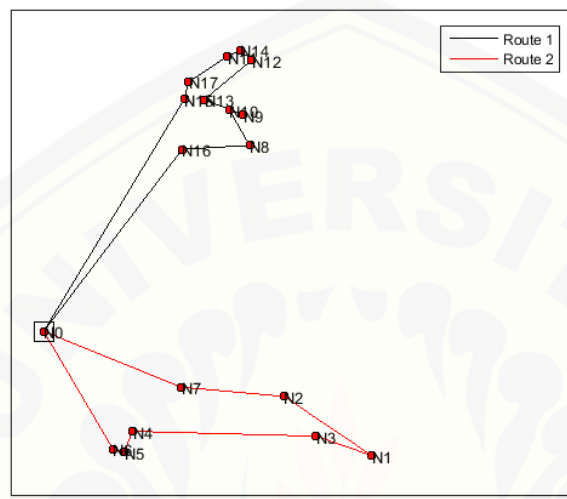


CSB

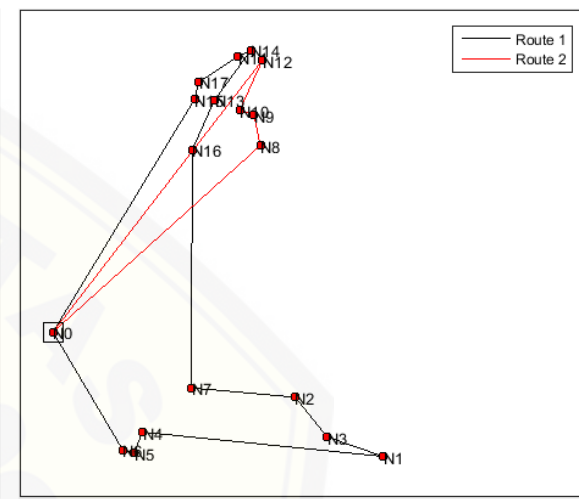
17 titik



CIH

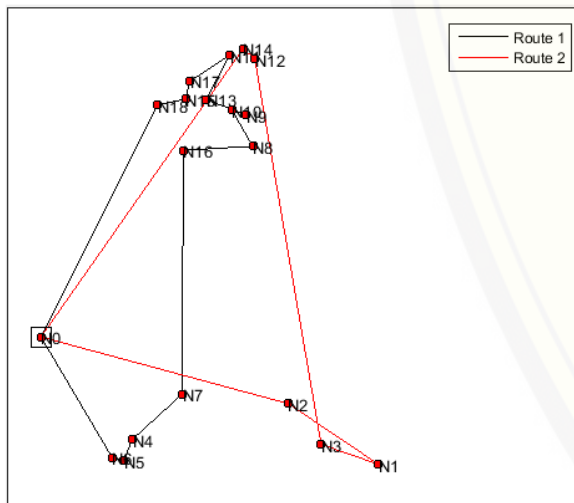


CSF

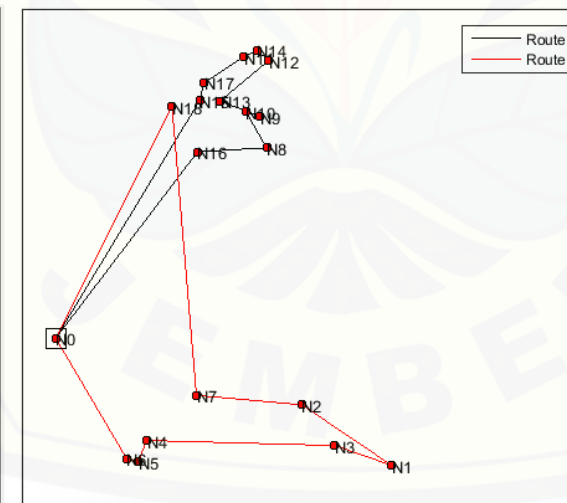


CSB

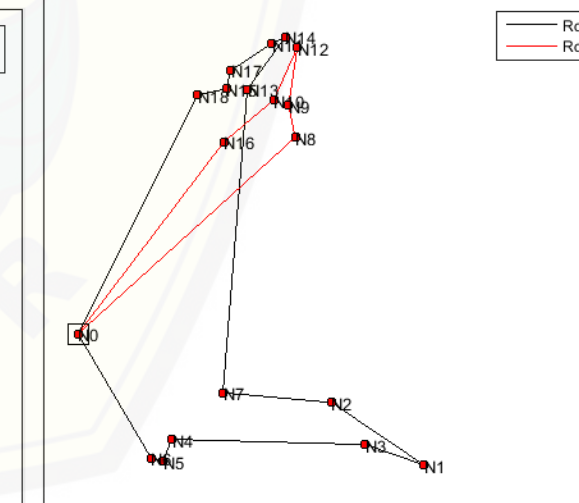
18 titik



CIH

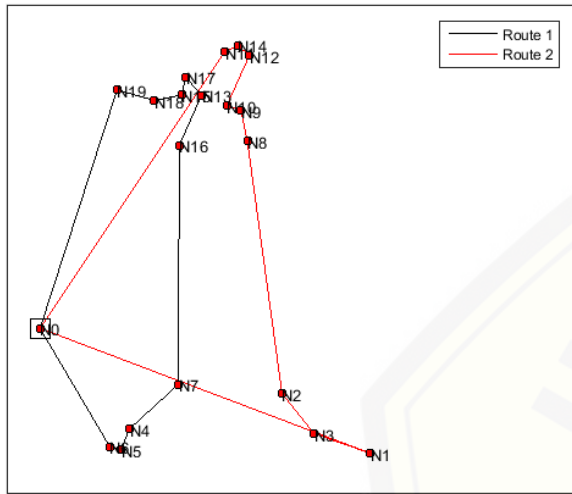


CSF

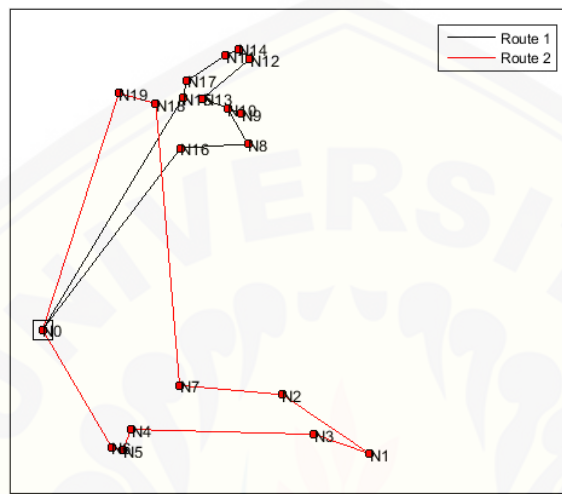


CSB

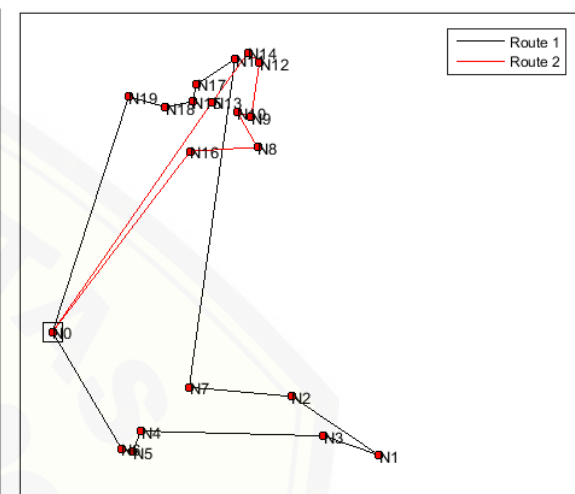
19 titik



CIH

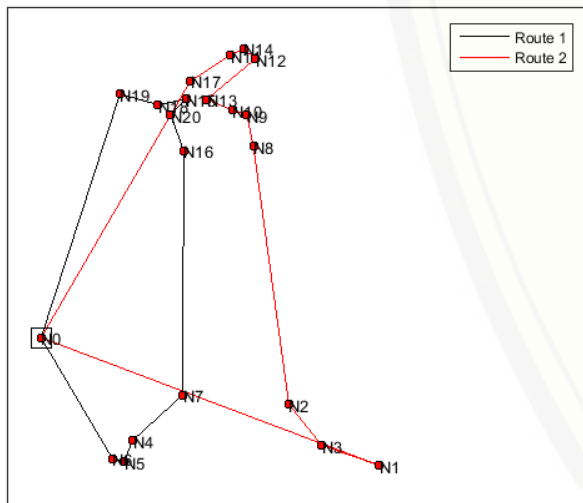


CSF

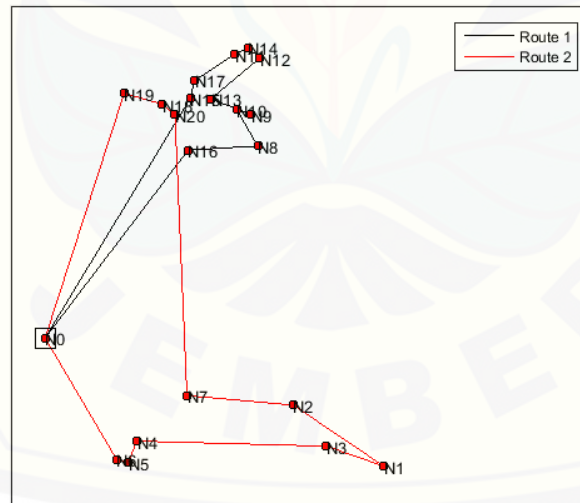


CSB

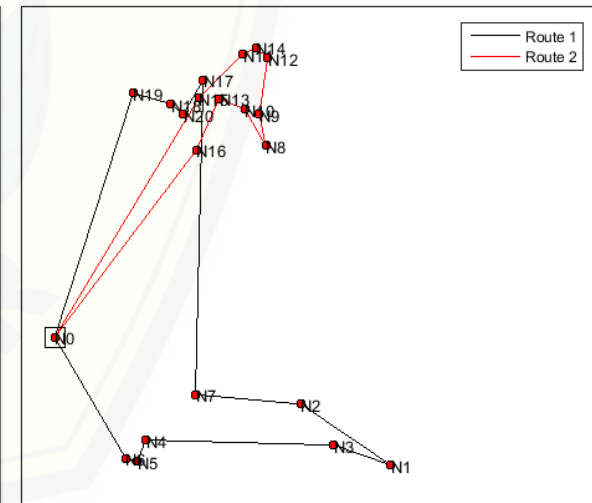
20 titik



CIH

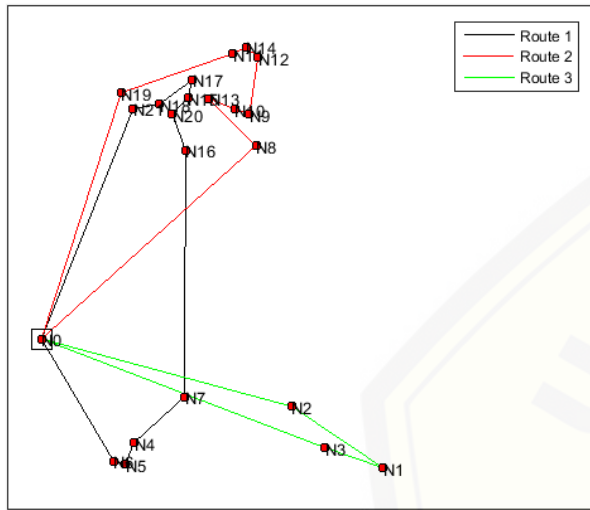


CSF

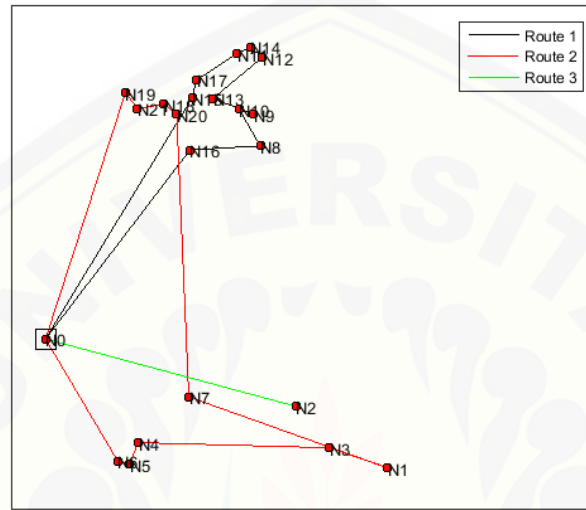


CSB

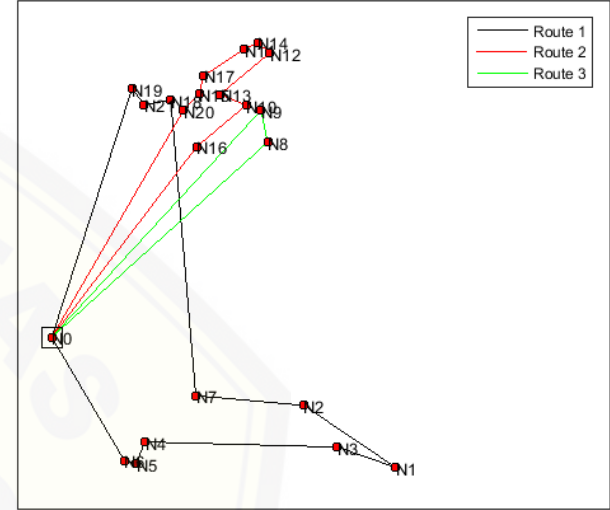
21 titik



CIH

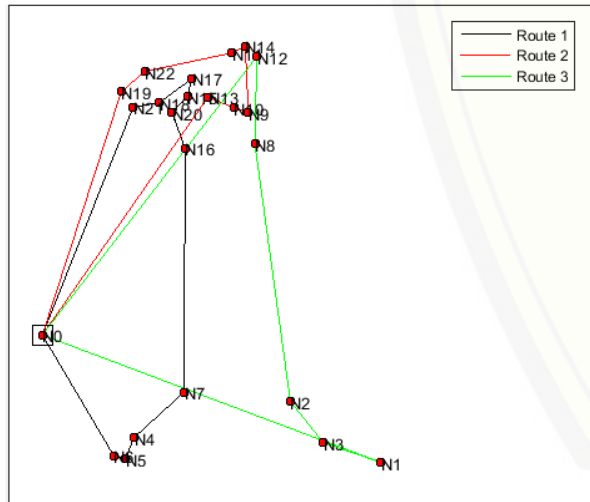


CSF

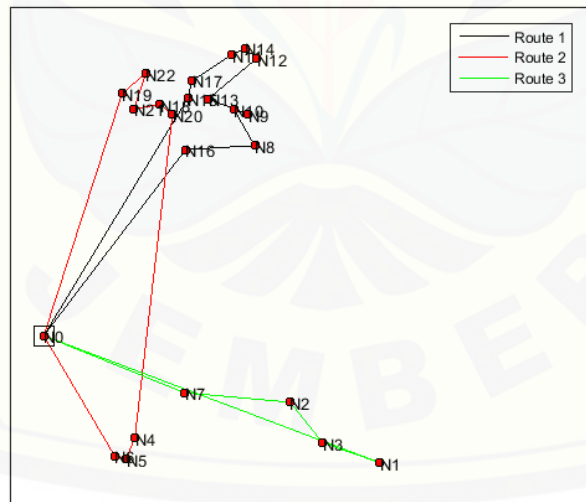


CSB

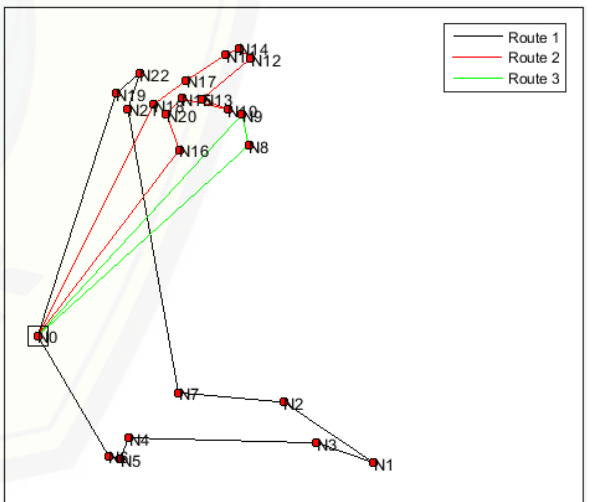
22 titik



CIH

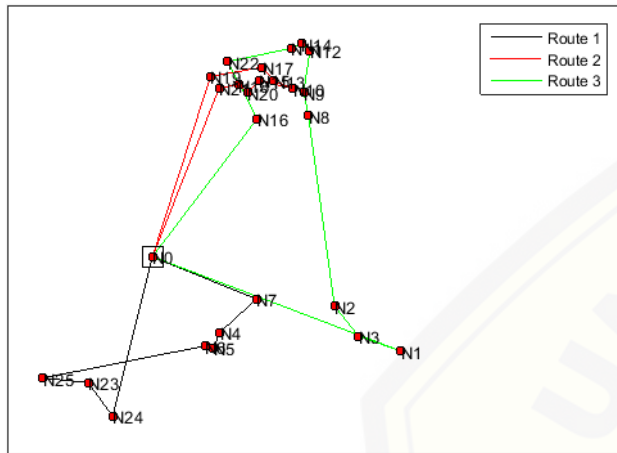


CSF

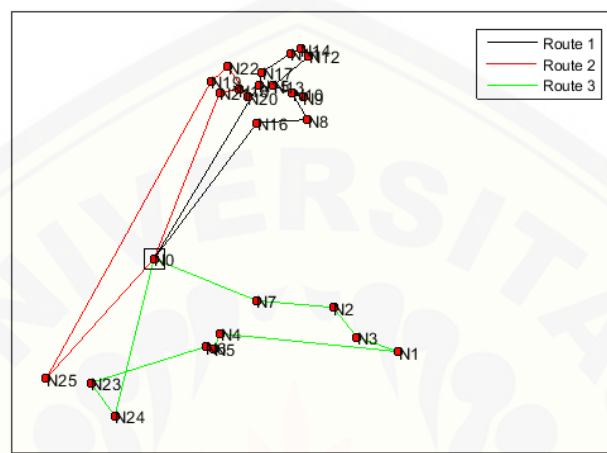


CSB

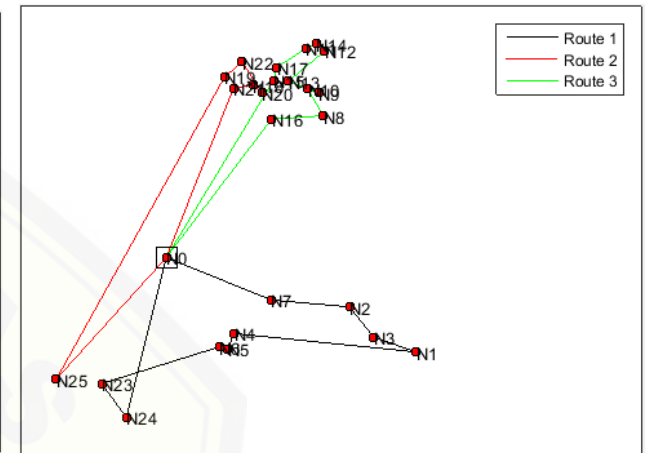
25 titik



CIH

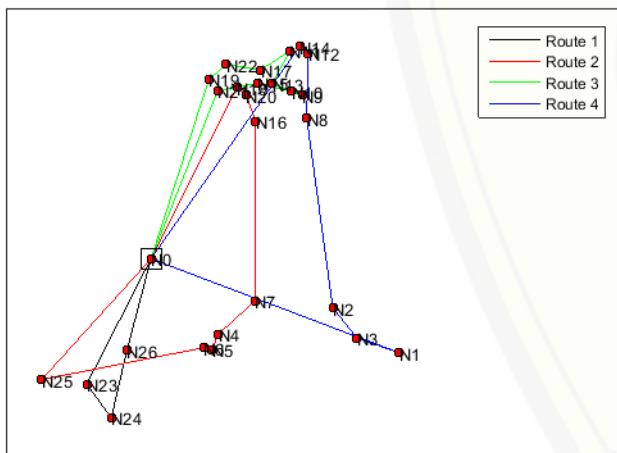


CSF

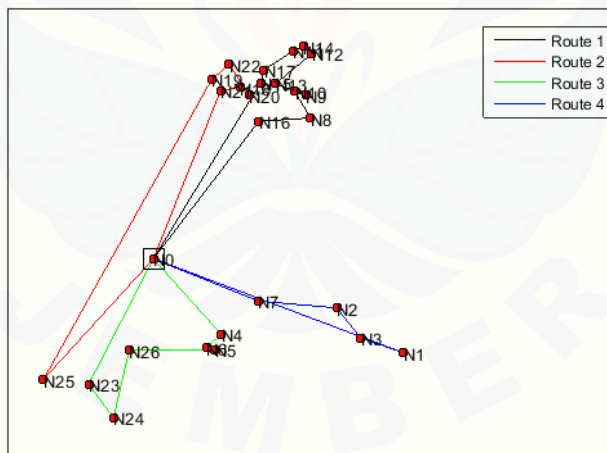


CSB

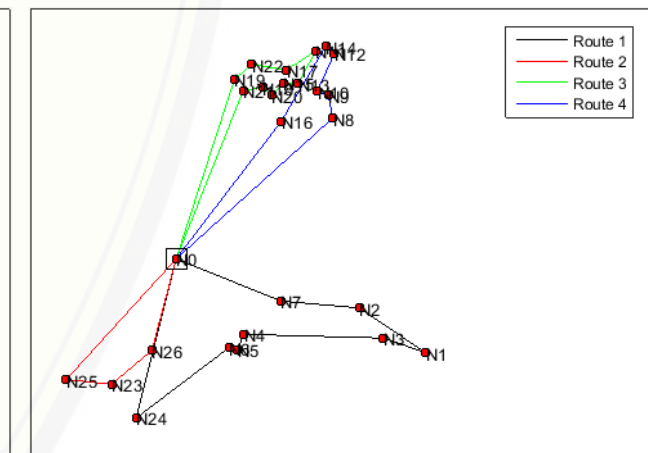
26 titik



CIH



CSF



CSB

