

**OPTIMASI RUTE DISTRIBUSI PRODUK
DENGAN PENERAPAN *VEHICLE ROUTING PROBLEM*
ALGORITMA *TABU SEARCH*
(STUDI KASUS: DISTRIBUSI HARIAN DI SUATU
PERUSAHAAN AGRIBISNIS)**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
sarjana teknik**

**ARIEF RAKHMAT CAHYADI
04 05 07 00 38**



**UNIVERSITAS INDONESIA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JULI 2009**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Arief Rakhmat Cahyadi

NPM : 0405070038

Tanda Tangan : 

Tanggal : Juli 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Arief Rakhmat Cahyadi
NPM : 0405070038
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Rute Distribusi Produk Dengan Penerapan *Vehicle Routing Problem* Algoritma *Tabu Search* (Studi Kasus: Distribusi Harian di Suatu Perusahaan Agribisnis)

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana S1 pada Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Armand Omar Moeis, ST., M.Sc. ()

Penguji : Dr. Ir. T. Yuri M.Eng.Sc. ()

Penguji : Ir. Boy Nurtjahyo M.,MSIE ()

Penguji : Ir. Rahmat Nurcahyo,M.Eng.Sc. ()

Ditetapkan di : Depok

Tanggal : Juli 2009

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, skripsi ini dapat selesai tepat pada waktunya. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk menyelesaikan Program Pendidikan Sarjana di Departemen Teknik Industri Fakultas Teknik Universitas Indonesia.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Armand Omar Moeis, ST., MSc., selaku dosen pembimbing skripsi yang telah memberikan waktu, tenaga, dan pikiran dalam membimbing penulis.
2. Bapak Ir. Amar Rachman, MEIM yang telah memberikan masukan berharga bagi penulis, dan juga seluruh dosen pengajar di Departemen Teknik Industri yang telah memberikan ilmu dan pengetahuan berharga kepada penulis.
3. Bang Komar, atas segala masukan yang sangat berharga serta bantuannya dalam penyusunan program MATLAB.
4. Bapak Liberto, Bapak Ignas, dan Bapak Dadang atas semua bantuan, informasi, dan bimbingannya selama berada di perusahaan.
5. Ayah, Ibu, dan Mbak Novi yang telah memberikan perhatian dan dukungan moril yang sangat berarti bagi penulis.
6. Rekan-rekan bimbingan penulis; Yopi, Warman, Aan, dan Cica atas dukungan, bantuan, dan kebersamaan yang sangat menyenangkan.
7. Arthur Dias dan Aziz Sutrisno, atas bantuannya yang sangat berharga.
8. Seluruh teman-teman TI 2005 atas dukungan dan kebersamaannya.
9. Seluruh pihak yang telah membantu penulis dari awal sampai selesainya penulisan skripsi ini, yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam skripsi ini. Oleh karena itu, segala saran dan kritik sangat diharapkan demi kesempurnaan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu.

Depok, Juli 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama : Arief Rakhmat Cahyadi
NPM : 0405070038
Program Studi : Teknik Industri
Departemen : Teknik Industri
Fakultas : Teknik
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

“Optimasi Rute Distribusi Produk Dengan Penerapan *Vehicle Routing Problem* Algoritma *Tabu Search* (Studi Kasus: Distribusi Harian di Suatu Perusahaan Agribisnis)”

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia / format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilih Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : Juli 2009

Yang menyatakan



(Arief Rakhmat Cahyadi)

ABSTRAK

Nama : Arief Rakhmat Cahyadi
Program Studi : Teknik Industri
Judul : Otimasi Rute Distribusi Produk Dengan Penerapan *Vehicle Routing Problem* Algoritma *Tabu Search*
(Studi Kasus: Distribusi Harian di Suatu Perusahaan Agribisnis)

Penelitian ini membahas mengenai kasus distribusi di suatu perusahaan agribisnis yang memiliki biaya transportasi yang cukup tinggi. Penentuan rute distribusi dengan menggunakan *Vehicle Routing Problem* (VRP) merupakan salah satu metode yang dapat digunakan untuk meningkatkan efisiensi biaya transportasi. Tujuan dari VRP adalah mengatur urutan rute pengiriman sehingga menghasilkan jarak tempuh total seminimal mungkin. Untuk menyelesaikan permasalahan pengiriman produk di perusahaan agribisnis ini, maka dikembangkan model penyelesaian VRP dengan menggunakan algoritma *Tabu Search*. *Tabu Search* merupakan salah satu kelas metaheuristik yang berbasis pada pencarian lokal. Rute usulan hasil penelitian menghasilkan perbaikan, yaitu pengurangan jarak tempuh dan biaya transportasi sebesar 11,13% dan 10,46% untuk satu minggu pengiriman.

Kata Kunci:

Vehicle routing problem, tabu search, optimasi, efisiensi biaya

ABSTRACT

Name : Arief Rakhmat Cahyadi
Study Program : Industrial Engineering
Title : Distribution Route Optimization Using Tabu Search Algorithm for Vehicle Routing Problem (Case Study: Daily Distribution at an Agribusiness Company)

This research studies about a distribution case in an agribusiness company that has high transportation cost. Setting delivering routes using Vehicle Routing Problem (VRP) is one of method that applicable to increase the efficiency of transportation cost. The objective of VRP is to set delivering routes order so that its total distance can be minimize. In order to solve delivery problem in this agribusiness company, VRP model was developed using Tabu Search algorithm. Tabu Search is one of metaheuristics class that works based on local search. The result of this research shows the improvement that reduce total distance traveled and transportation cost by 11,13% and 10,46% for one week of delivery.

Key words:

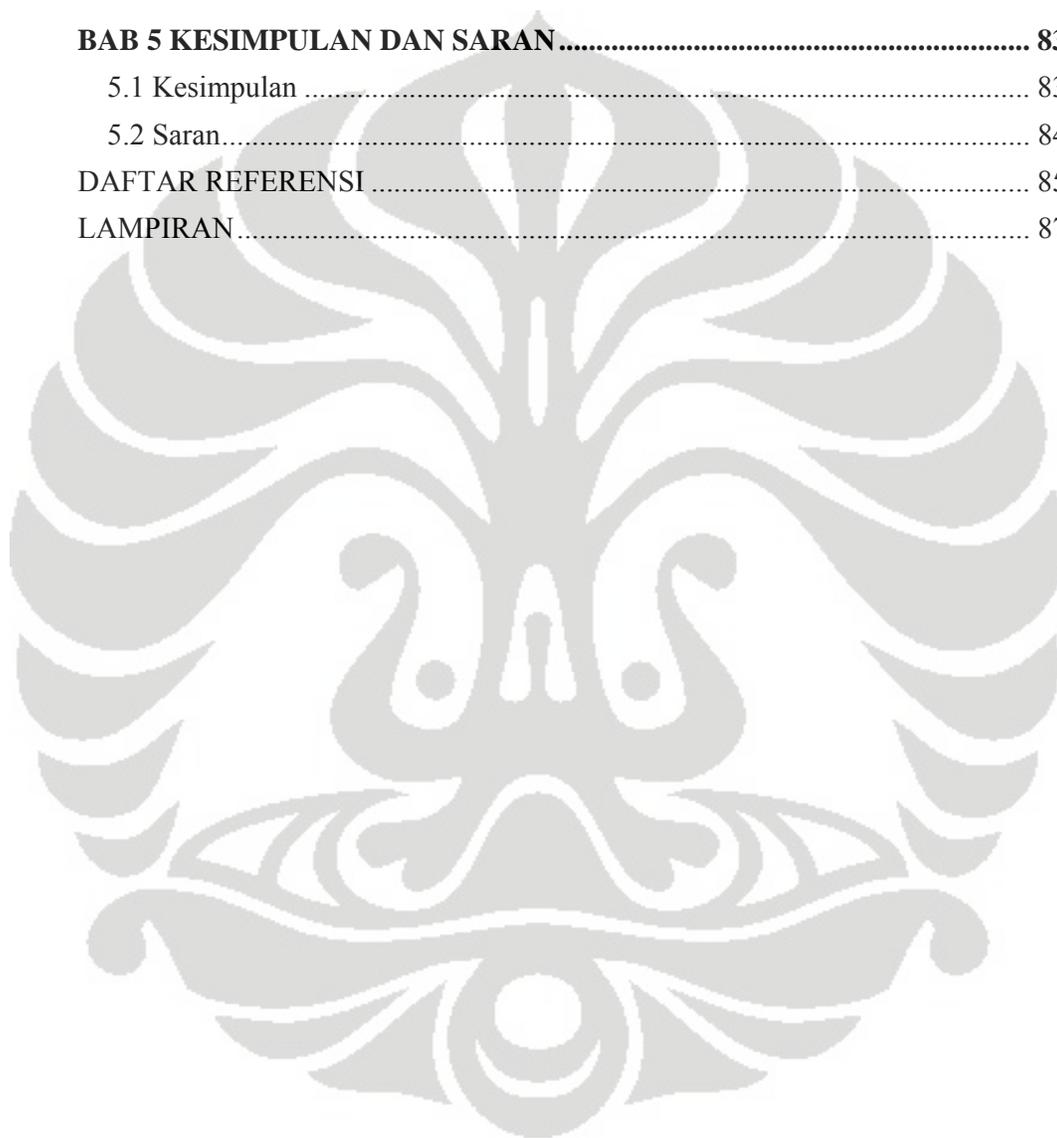
Vehicle routing problem, tabu search, optimization, cost efficiency

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Permasalahan.....	1
1.2 Diagram Keterkaitan masalah.....	3
1.3 Perumusan Permasalahan.....	3
1.4 Tujuan Penelitian.....	3
1.5 Ruang Lingkup Permasalahan.....	5
1.6 Metodologi Penelitian.....	5
1.7 Sistematika Penulisan.....	8
BAB 2 TINJAUAN PUSTAKA.....	10
2.1 <i>Vehicle Routing Problem</i>	10
2.1.1 Definisi dan Karakteristik.....	10
2.1.2 <i>Vehicle Routing Problem with Time Windows</i>	14
2.1.2.1 Model Matematis VRPTW.....	15
2.2 Metode Penyelesaian VRP.....	17
2.2.1 Pendekatan Eksak.....	17
2.2.2 Heuristik Klasik.....	17
2.2.2.1 <i>Sweep Method</i>	19
2.2.2.2 <i>Saving Method</i>	21
2.2.2.3 <i>Nearest Neighbourhood Method</i>	22

2.2.3 Metaheuristik	23
2.3 <i>Tabu Search</i>	24
2.3.1 Pengertian Umum.....	24
2.3.2 Penggunaan Memori	25
2.3.3 Karakteristik dan Prosedur <i>Tabu Search</i>	26
2.4 <i>Tabu Search</i> Pada VRP	30
2.4.1 Komponen <i>Tabu Search</i>	30
2.4.2 Mekanisme Pembentukan Solusi Tetangga	32
2.4.3 Prosedur <i>Tabu Search for VRP</i>	33
BAB 3 PENGUMPULAN DATA.....	35
3.1 Profil Perusahaan	35
3.2 Data yang Dibutuhkan.....	35
3.2.1 Depot.....	36
3.2.2 Armada Pengiriman	36
3.2.3 Biaya Pengiriman.....	37
3.2.4 Konsumen	37
3.2.5 Waktu dan Kecepatan	40
3.2.6 Permintaan	40
3.2.7 Jarak	43
3.2.8 Rute Pengiriman.....	43
BAB 4 PENGOLAHAN DATA DAN ANALISIS	45
4.1 Algoritma Penyelesaian	45
4.2 Verifikasi dan Validasi Program.....	49
4.3 <i>Input Data</i>	51
4.4 Pengolahan Data.....	52
4.4.1 Konfigurasi Parameter	52
4.4.1.1 Studi Parameter	53
4.4.1.2 Hasil Studi Parameter.....	55
4.4.2 Hasil Pengolahan Data	61
4.5 Analisis.....	64
4.5.1 Analisis Metode	65
4.5.1.1 Analisis Algoritma	65
4.5.1.2 Analisis Studi Parameter.....	66

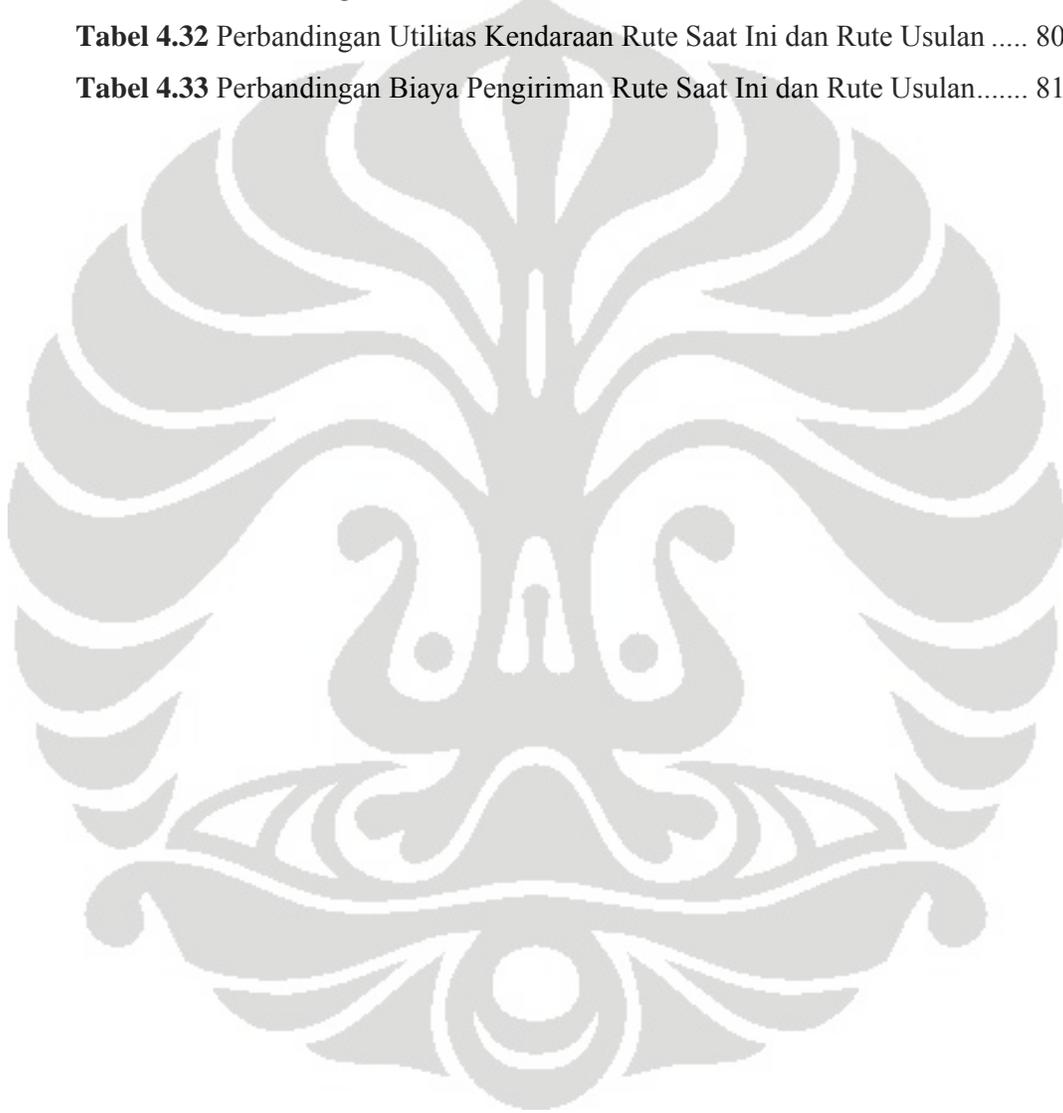
4.5.2 Analisis Program.....	68
4.5.3 Analisis Hasil Optimasi	72
4.5.3.1 Analisis Rute Harian dan Jarak Tempuh.....	73
4.5.3.3 Analisis Utilitas Kendaraan	79
4.5.3.4 Analisis Biaya Pengiriman.....	81
BAB 5 KESIMPULAN DAN SARAN.....	83
5.1 Kesimpulan	83
5.2 Saran.....	84
DAFTAR REFERENSI	85
LAMPIRAN.....	87



DAFTAR TABEL

Tabel 3.1 Daftar Konsumen dan Kodenya	38
Tabel 3.2 Kapasitas Krat untuk Setiap Jenis Sayuran	41
Tabel 3.3 Pengiriman Pada Konsumen Selama 1 Minggu	42
Tabel 3.4 Data Rute Selama 1 Minggu	44
Tabel 4.1 Jarak Antartitik Konsumen <i>Dummy</i>	49
Tabel 4.2 Jumlah Pengiriman Tiap Konsumen <i>Dummy</i>	49
Tabel 4.3 Konfigurasi Paramater yang Digunakan dalam Validasi	50
Tabel 4.4 Konfigurasi I	53
Tabel 4.5 Konfigurasi II	54
Tabel 4.6 Konfigurasi III	54
Tabel 4.7 Konfigurasi IV	54
Tabel 4.8 Hasil Penggunaan Konfigurasi I	55
Tabel 4.9 Hasil Penggunaan Konfigurasi II	56
Tabel 4.10 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 1$	56
Tabel 4.11 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 2$	57
Tabel 4.12 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 3$	57
Tabel 4.13 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 4$	57
Tabel 4.14 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 1$	58
Tabel 4.15 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 2$	59
Tabel 4.16 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 3$	59
Tabel 4.17 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 4$	59
Tabel 4.18 Rekap Hasil Studi Parameter	60
Tabel 4.19 Jarak Tempuh Harian untuk 5 Kali <i>Run</i> Program	61
Tabel 4.20 Rute dan Jarak Tempuh Total Hasil Optimasi	62
Tabel 4.21 Perbandingan Jumlah Kendaraan dan Jarak Tempuh	63
Tabel 4.22 Muatan dan Persentase Utilitas Kendaraan Hasil Optimasi	64
Tabel 4.23 Konfigurasi Terbaik	67
Tabel 4.24 Waktu Run Program untuk Kasus Harian	69
Tabel 4.25 Perbandingan Rute Saat Ini dan Rute Usulan 5 Januari 2009	73

Tabel 4.26 Perbandingan Rute Saat Ini dan Rute Usulan 6 Januari 2009.....	74
Tabel 4.27 Perbandingan Rute Saat Ini dan Rute Usulan 7 Januari 2009.....	75
Tabel 4.28 Perbandingan Rute Saat Ini dan Rute Usulan 8 Januari 2009.....	76
Tabel 4.29 Perbandingan Rute Saat Ini dan Rute Usulan 9 Januari 2009.....	77
Tabel 4.30 Perbandingan Rute Saat Ini dan Rute Usulan 10 Januari 2009.....	78
Tabel 4.31 Perbandingan Rute Saat Ini dan Rute Usulan 11 Januari 2009.....	78
Tabel 4.32 Perbandingan Utilitas Kendaraan Rute Saat Ini dan Rute Usulan	80
Tabel 4.33 Perbandingan Biaya Pengiriman Rute Saat Ini dan Rute Usulan.....	81



DAFTAR GAMBAR

Gambar 1.1 Diagram Keterkaitan Masalah	4
Gambar 1.2 Diagram Alir Metode Penelitian.....	7
Gambar 2.1 Permasalahan Dasar Kelas-Kelas VRP dan Keterkaitannya	14
Gambar 2.2 Contoh Solusi Metode <i>Sweep</i>	21
Gambar 2.3 Ilustrasi Pergerakan <i>Insertion</i>	32
Gambar 2.4 Ilustrasi Pergerakan 2-Opt.....	32
Gambar 2.5 Ilustrasi Pergerakan <i>Interchange</i>	33
Gambar 3.1 Lokasi Depot dan Konsumen	39
Gambar 4.1 Algoritma Penyelesaian Solusi Awal	46
Gambar 4.2 Algoritma <i>Tabu Search</i> pada VRP	47
Gambar 4.3 Tampilan <i>Form Input</i>	71
Gambar 4.4 Tampilan Solusi Akhir.....	72

DAFTAR LAMPIRAN

Lampiran 1 : Matriks Jarak Antar Konsumen

Lampiran 2 : <i>Script M-File</i> Program



BAB 1

PENDAHULUAN

1.1 Latar Belakang Permasalahan

Dalam dunia industri, logistik merupakan salah satu aspek yang penting untuk diperhatikan. Untuk itu, setiap perusahaan dituntut untuk memiliki suatu manajemen logistik yang baik. Manajemen logistik sendiri memiliki tujuan akhir untuk mendapatkan atau menyediakan sejumlah barang atau jasa yang tepat pada tempat dan waktu, serta kondisi yang diinginkan dengan memberikan kontribusi terbesar bagi perusahaan (Ballou, 2004).

Salah satu aspek dari manajemen logistik yang sangat penting adalah manajemen sistem distribusi produk. Untuk mencapai tujuan manajemen logistik, diperlukan suatu sistem distribusi produk yang baik sehingga produk dapat tersedia pada waktu, tempat, dan jumlah yang tepat sesuai dengan permintaan konsumen. Suatu perusahaan harus dapat mengoptimalkan sistem distribusinya agar dapat bersaing dengan para kompetitornya. Hal tersebut dikarenakan distribusi fisik suatu perusahaan memiliki pengaruh yang cukup signifikan terhadap biaya dan juga tingkat pelayanan terhadap konsumen. Salah satu cara untuk mengoptimalkan sistem distribusi tersebut adalah dengan melakukan pengoptimalan pada aktivitas transportasi.

Biaya transportasi merupakan salah satu elemen biaya distribusi yang kontribusinya paling besar, yaitu sekitar $\frac{1}{3}$ sampai $\frac{2}{3}$ dari total biaya aktivitas distribusi (Ballou, 2004). Melihat kontribusi biaya sebesar itu, maka diperlukan suatu usaha untuk dapat meningkatkan efisiensi biaya transportasi. Salah satu cara untuk meningkatkan efisiensi biaya transportasi tersebut adalah dengan membuat rencana rute distribusi yang akan dilalui untuk mengantarkan produk dari depot sampai ke konsumen dengan menggunakan VRP (*Vehicle Routing Problem*).

Secara sederhana, VRP merupakan permasalahan yang meliputi konstruksi rute-rute dari sejumlah kendaraan yang dimulai dari suatu depot utama menuju ke lokasi sejumlah konsumen dengan jumlah permintaan tertentu. Tujuannya adalah untuk meminimumkan jarak tempuh perjalanan dan juga biaya total perjalanan tanpa melebihi kapasitas kendaraan yang ada (Poot, Kant, dan Wagelmans, 2002).

VRP merupakan masalah yang relatif kompleks dan termasuk dalam *nonpolynomial hard (NP-hard)*. Oleh karena itu, pada umumnya VRP dilakukan dengan menggunakan pendekatan heuristik. Dalam VRP sendiri, metode heuristik dapat dibagi menjadi dua, yaitu heuristik klasik dan heuristik modern (metaheuristik). Dibandingkan dengan heuristik klasik, metaheuristik dapat memberikan pencarian solusi yang lebih teliti dan lebih optimal. Dalam dekade terakhir ini, telah banyak penelitian mengenai permasalahan VRP dengan menggunakan berbagai pendekatan metaheuristik. Untuk sebagian besar kasus, *Tabu Search* memberikan hasil pencarian solusi yang lebih baik dibandingkan dengan metode metaheuristik lain seperti *Simulated Annealing* atau *Genetic Algorithm*.

Masalah umum yang seringkali dihadapi perusahaan dalam melakukan pendistribusian produk antara lain adalah jumlah permintaan yang berbeda untuk setiap titik, keterbatasan kapasitas kendaraan, permintaan yang berfluktuatif, batasan waktu pengiriman, jumlah titik kirim yang banyak, dan yang paling penting adalah penentuan rute kendaraan yang optimal untuk dapat mencapai semua titik kirim sehingga dapat menghemat biaya.

Permasalahan tersebut juga dialami oleh suatu perusahaan yang bergerak dalam bidang pertanian. Perusahaan agribisnis ini merupakan perusahaan yang memproduksi berbagai jenis sayuran. Perusahaan ini terletak di daerah Ciawi dan setiap harinya harus melakukan pengantaran produk sayurannya kepada sejumlah konsumen yang ada di Jakarta dan sekitarnya. Untuk memberikan kepuasan kepada pelanggan, perusahaan agribisnis ini harus melakukan pengiriman kepada para konsumennya tepat waktu. Namun, tepat waktu itu sendiri tidak akan cukup jika tidak ditunjang oleh efisiensi rute distribusi. Selama ini, pertimbangan perusahaan dalam menentukan rute distribusi hanyalah berdasarkan keterbatasan kapasitas, lokasi, dan pengalaman, tanpa pernah mempertimbangkan apakah rute tersebut sudah minimal atau belum.

Dalam melakukan distribusi produk sayurannya, perusahaan agribisnis ini menggunakan kendaraan yang dilengkapi dengan kontainer berpendingin. Hal ini dikarenakan sayuran harus berada dalam kondisi masih segar ketika sampai ke tangan konsumen. Akibat dari penggunaan kendaraan berpendingin ini, maka

biaya transportasi pendistribusian produk menjadi relatif mahal dan memberikan kontribusi biaya yang cukup besar. Oleh karena itu masalah transportasi pendistribusian produk harian di perusahaan ini harus benar-benar diperhatikan mengingat biaya transportasi yang begitu besar.

Penjadwalan pengiriman harian dilakukan perusahaan berdasarkan pesanan yang ada untuk setiap harinya. Hal tersebut dikarenakan jumlah pelanggan yang melakukan permintaan dan jumlah permintaannya tidak selalu sama untuk setiap harinya. Oleh karena itu, rute pengiriman harian pun akan berbeda sesuai dengan pesanan dari pelanggan yang ada. Oleh karena jumlah permintaan yang tidak menentu tersebut, maka ada kalanya kapasitas kendaraan tidak mencukupi dan ada kalanya kendaraan tidak terutilisasi secara penuh. Selain itu, jika jumlah permintaan pada suatu hari terlalu banyak, maka tidak jarang terjadi pengiriman kepada pelanggan mengalami keterlambatan. Untuk itu, diperlukan suatu perencanaan sistem distribusi yang lebih baik, khususnya penentuan rute distribusi, sehingga pengiriman akan selalu tepat waktu dan efisiensi biaya distribusi dapat ditingkatkan.

1.2 Diagram Keterkaitan masalah

Pada bagian sebelumnya telah dijelaskan mengenai latar belakang dari penelitian ini. Untuk dapat memberikan gambaran sistemik yang lebih menyeluruh, maka disusun suatu diagram keterkaitan permasalahan seperti pada gambar 1.1. Diagram keterkaitan masalah ini akan membawa kepada bagian berikutnya, yaitu perumusan permasalahan.

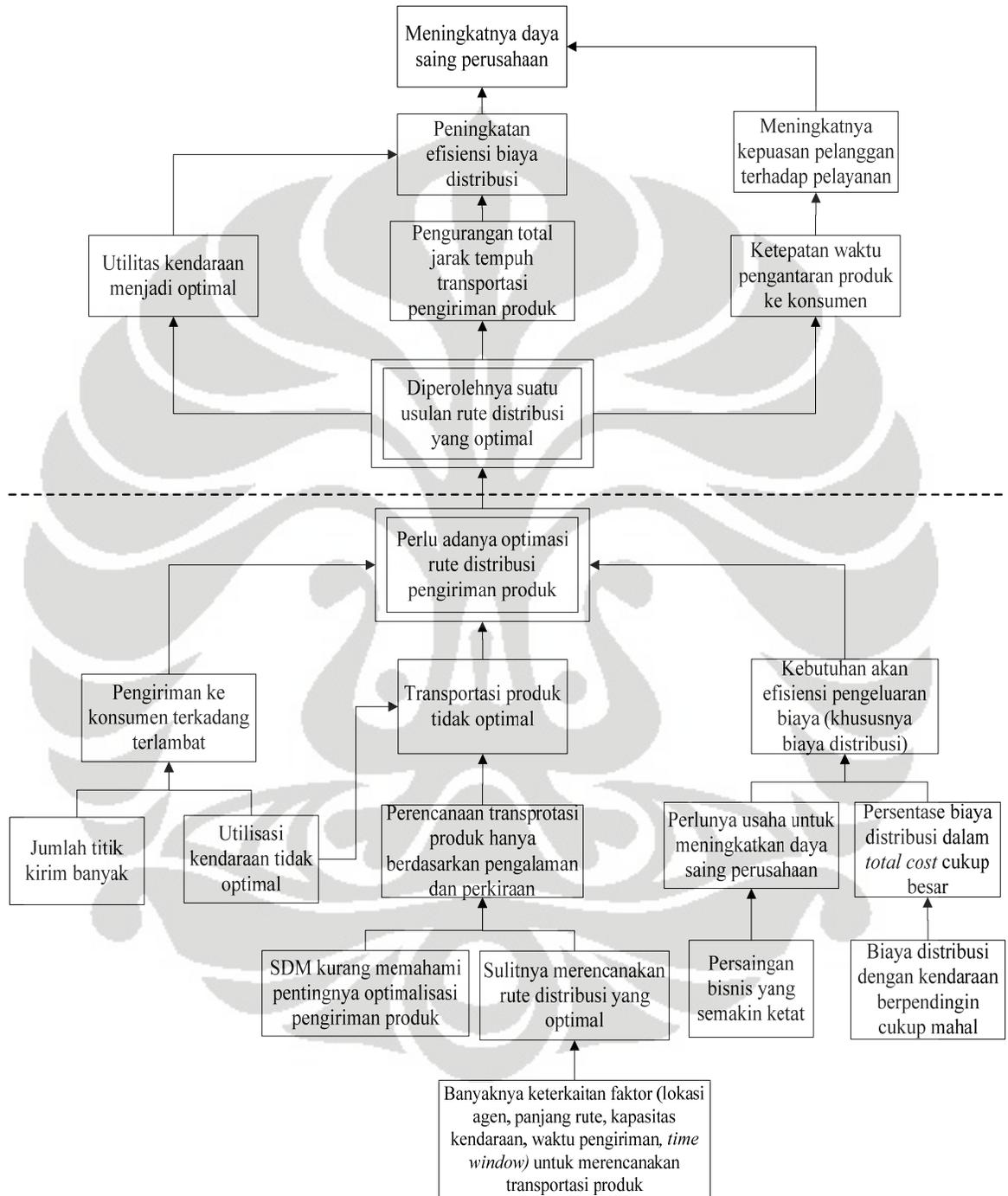
1.3 Perumusan Permasalahan

Pokok permasalahan yang menjadi fokus dalam penelitian ini adalah perlunya suatu usulan rute distribusi produk yang optimal dalam rangka peningkatan efisiensi biaya distribusi.

1.4 Tujuan Penelitian

Tujuan yang diharapkan dari penelitian ini adalah diperolehnya suatu usulan mengenai rute distribusi pengiriman produk yang optimal sehingga dapat

meningkatkan efisiensi biaya transportasi melalui penerapan *Vehicle Routing Problem* menggunakan algoritma *Tabu Search*.



Gambar 1.1 Diagram Keterkaitan Masalah

1.5 Ruang Lingkup Permasalahan

Ruang lingkup dari penelitian ini digunakan agar masalah yang diteliti lebih dapat terarah dan terfokus sehingga penelitian dapat dilakukan sesuai dengan apa yang direncanakan. Ruang lingkup penelitian ini adalah sebagai berikut:

1. Implementasi VRP pada upaya perencanaan rute distribusi harian yang optimal meliputi semua konsumen di wilayah Jabodetabek.
2. Kondisi kendaraan/armada angkut dianggap sama untuk semua armada dan diasumsikan semua kendaraan mampu melakukan pengiriman ke semua pelanggan.
3. Data permintaan yang diambil adalah data permintaan konsumen selama satu pekan, yaitu permintaan pada tanggal 5 Januari 2009 sampai 11 Januari 2009.
4. VRP yang akan diselesaikan adalah VRP dengan *time windows* atau batasan waktu pelayanan.
5. Pengolahan data dilakukan dengan pengembangan program komputer khusus dengan menggunakan perangkat lunak MATLAB.
6. Algoritma yang digunakan untuk mendapatkan solusi awal adalah algoritma *Nearest Neighbourhood* yang kemudian akan dioptimalkan lagi dengan menggunakan algoritma *Tabu Search*

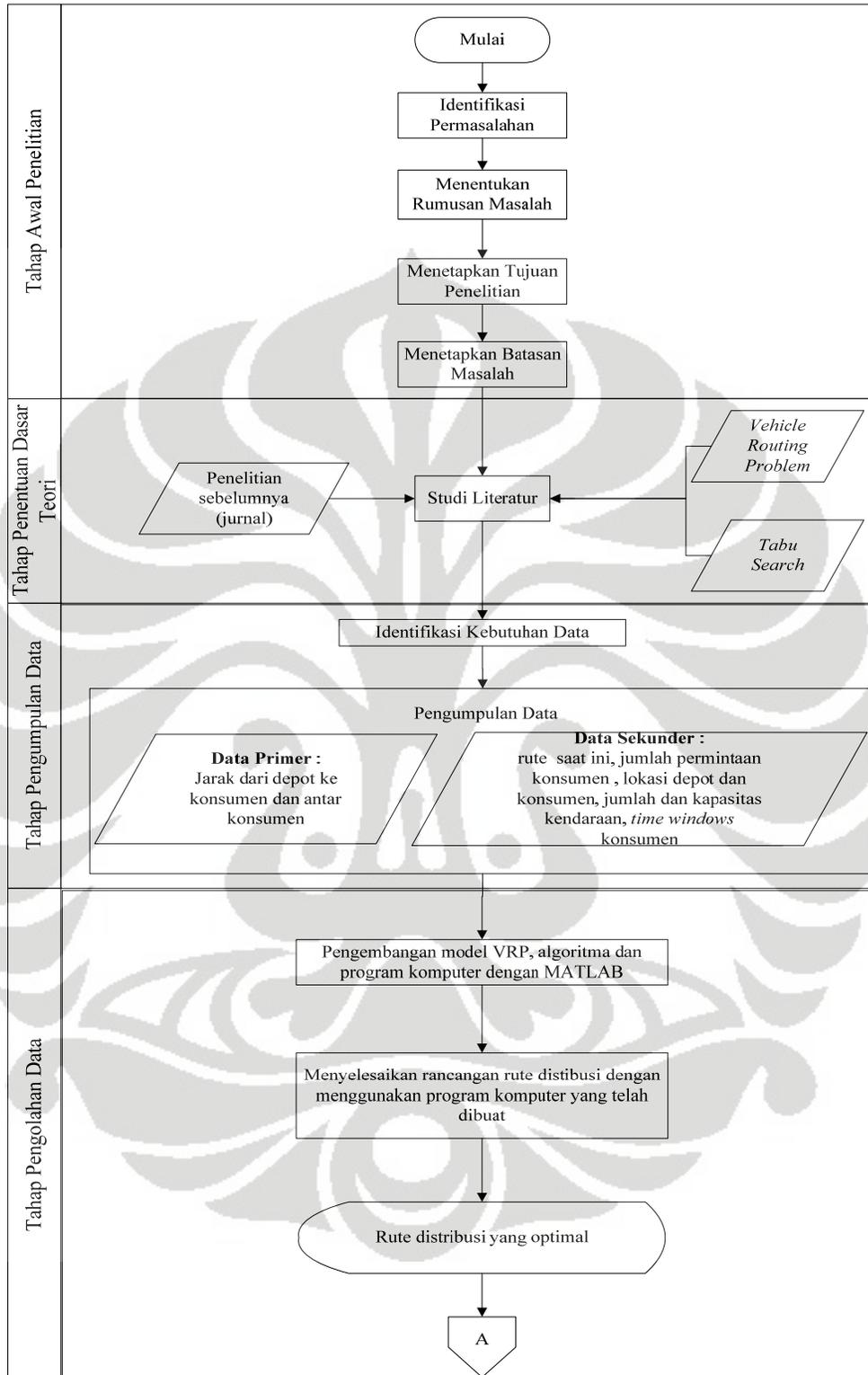
1.6 Metodologi Penelitian

Berikut akan dijelaskan mengenai metodologi atau langkah-langkah yang dilakukan dalam penelitian, sebagaimana tergambar pada diagram alir metodologi penelitian pada gambar 1.2.

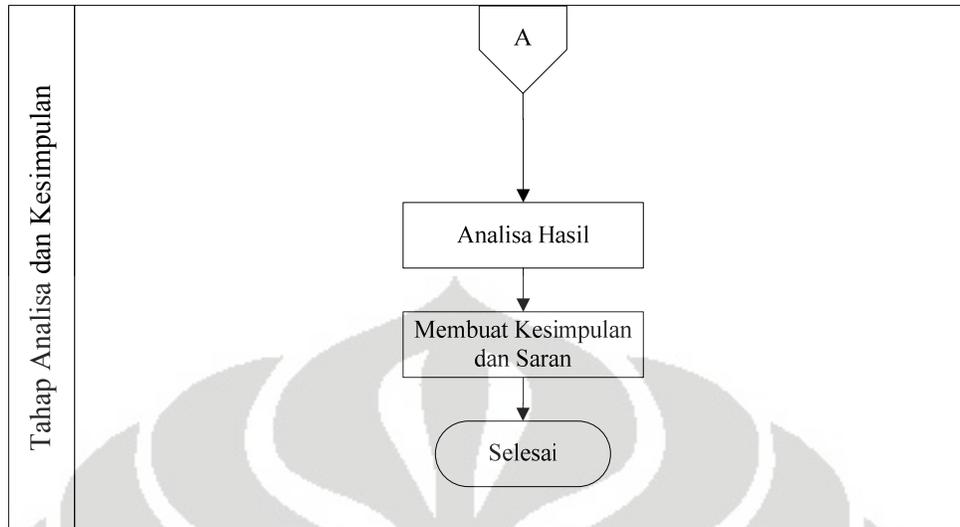
1. Identifikasi permasalahan. Pada tahap ini diperoleh suatu gambaran mengenai latar belakang masalah untuk merumuskan topik yang ingin dikaji lebih lanjut dalam penelitian. Dalam penelitian ini, masalah yang diidentifikasi adalah bahwasanya rute distribusi yang saat ini digunakan oleh suatu perusahaan agribisnis belum optimal dan efisien.
2. Studi literatur. Pada tahap ini dilakukan studi literatur dari berbagai sumber, seperti buku, jurnal, dan artikel. Studi literatur yang dilakukan adalah

mengenai teori dan permasalahan yang berkaitan dengan *Vehicle Routing Problem* dan algoritma *Tabu Search*

3. Perumusan masalah. Pada tahap ini dilakukan perumusan masalah yang menjadi fokus dalam penelitian, yaitu perlunya suatu perencanaan rute distribusi yang efisien dan optimal dengan menggunakan VRP.
4. Menentukan tujuan penelitian. Sejalan dengan perumusan masalah, maka tujuan dari penelitian ini adalah untuk mendapatkan usulan perencanaan rute distribusi yang optimal melalui penerapan VRP.
5. Menetapkan batasan masalah. Batasan masalah perlu ditetapkan agar penelitian lebih terarah, terfokus dan berjalan sesuai dengan rencana
6. Identifikasi dan pengumpulan data yang dibutuhkan. Pada tahap ini dilakukan identifikasi data yang dibutuhkan untuk menyelesaikan permasalahan dan kemudian dilakukan pengumpulan data-data tersebut. Sebagian data merupakan data primer dan sebagian lagi data sekunder. Data primer didapatkan dengan pengukuran jarak dari depot ke semua titik kirim dan antar titik kirim, sedangkan data sekunder diperoleh dari wawancara dan pengumpulan dokumen perusahaan.
7. Pengolahan data. Pada tahap ini data yang telah didapatkan kemudian diolah dan digunakan untuk membuat suatu rancangan rute distribusi baru dengan menggunakan metode *tabu search algorithm for vehicle routing problem with time windows*. Perancangan rute distribusi dengan VRP dilakukan dengan bantuan program komputer khusus menggunakan perangkat lunak MATLAB.
8. Analisa hasil rancangan rute distribusi. Pada tahap ini dilakukan analisis terhadap rute distribusi yang telah dibuat dengan cara membandingkannya dengan rute yang selama ini digunakan.
9. Kesimpulan. Tahap terakhir adalah membuat kesimpulan dari penelitian berdasarkan hasil analisis yang telah dilakukan.



Gambar 1.2 Diagram Alir Metode Penelitian



Gambar 1.2 Diagram Alir Metode Penelitian (sambung)

1.7 Sistematika Penulisan

Sistematika yang digunakan dalam penulisan penelitian ini mengikuti aturan standar baku penulisan tugas akhir mahasiswa. Penulisan tugas akhir ini dibuat dalam lima bab yang memberikan gambaran sistematis sejak awal penelitian hingga tercapainya tujuan penelitian.

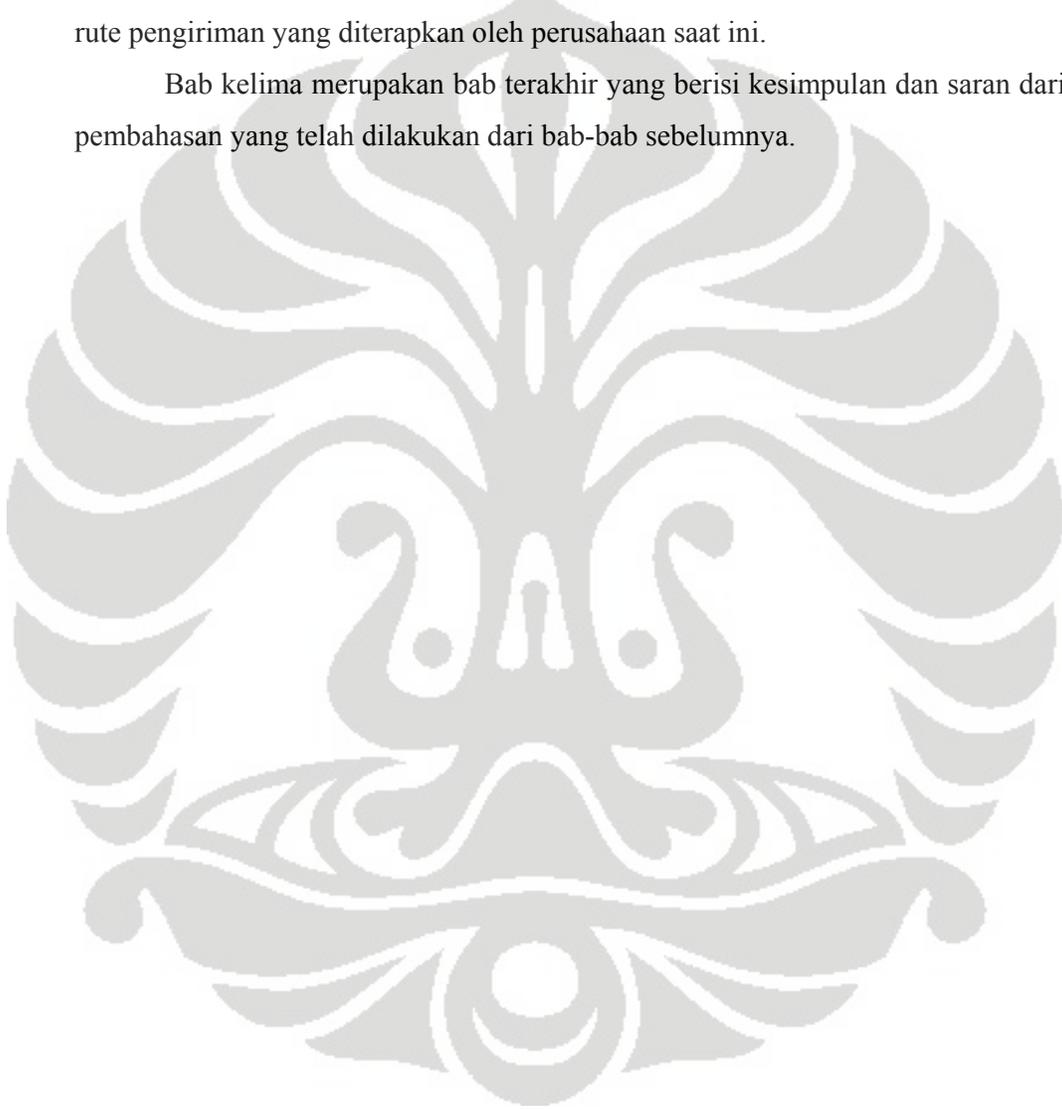
Bab pertama merupakan bab pendahuluan sebagai pengantar untuk menjelaskan isi penelitian secara garis besar. Dalam bab ini terdapat uraian mengenai latar belakang masalah, keterkaitan antar masalah, perumusan masalah, tujuan dan ruang lingkup penelitian, metodologi penelitian, serta sistematika penulisan.

Bab kedua berisi tentang dasar teori yang digunakan dalam penelitian. Secara umum pembahasannya difokuskan pada VRP mulai dari definisi umum dan modelnya, teknik pencarian solusi melalui algoritma eksak, heuristik, dan metaheuristik. Kemudian dilakukan pembahasan yang lebih mendalam mengenai algoritma *Tabu Search* sebagai salah satu pendekatan metaheuristik.

Bab ketiga menjelaskan mengenai pengumpulan data yang dibutuhkan untuk melaksanakan penelitian. Pengumpulan data dilakukan dengan cara observasi, wawancara, dan pengumpulan dokumen di suatu perusahaan agribisnis. Pada bab ini dijelaskan mengenai data-data yang berkenaan dengan aktivitas distribusi pengiriman barang.

Bab keempat berisi tentang pengolahan data dan analisisnya. Pada bab ini, data yang telah diperoleh kemudian diolah dan digunakan untuk membuat suatu rancangan rute distribusi dengan penerapan VRP dan bantuan *software* MATLAB untuk dapat menghasilkan suatu rute pengiriman yang optimal. Kemudian dilakukan analisis terhadap hasil rancangan rute distribusi usulan tersebut. Analisis dilakukan untuk membandingkan rute pengiriman hasil penelitian dengan rute pengiriman yang diterapkan oleh perusahaan saat ini.

Bab kelima merupakan bab terakhir yang berisi kesimpulan dan saran dari pembahasan yang telah dilakukan dari bab-bab sebelumnya.



BAB 2

TINJAUAN PUSTAKA

Bab ini membahas mengenai landasan teori yang digunakan dalam penelitian. Secara umum pembahasannya difokuskan pada VRP mulai dari definisi umum dan modelnya, serta teknik pencarian solusi melalui algoritma eksak, heuristik, dan metaheuristik. Kemudian dijelaskan pembahasan yang lebih mendalam mengenai algoritma *Tabu Search* sebagai salah satu pendekatan metaheuristik dan sebagai salah satu algoritma yang dapat digunakan untuk menyelesaikan permasalahan VRP.

2.1 Vehicle Routing Problem

2.1.1 Definisi dan Karakteristik

Vehicle Routing Problem (VRP), atau dapat juga disebut dengan *Vehicle Scheduling Problem*, berhubungan dengan distribusi produk atau barang jadi antara depot dengan konsumen. VRP pertama kali diperkenalkan oleh Dantzig dan Ramser pada tahun 1959. VRP ini memegang peranan penting pada manajemen distribusi dan telah menjadi salah satu permasalahan dalam optimasi kombinasi yang dipelajari secara luas. Model dan algoritmanya dapat digunakan secara efektif tidak hanya untuk pengiriman dan pengambilan barang, tetapi juga dapat diaplikasikan untuk masalah sistem transportasi sehari-hari, misalnya untuk perencanaan rute bus sekolah, pengumpulan sampah, pembersihan jalan, rute untuk penjual keliling, dan lainnya.

Secara sederhana, VRP merupakan permasalahan yang meliputi konstruksi rute-rute dari sejumlah kendaraan yang dimulai dari suatu depot utama menuju ke lokasi sejumlah konsumen dengan jumlah permintaan tertentu. Tujuannya adalah untuk meminimumkan biaya total tanpa melebihi kapasitas kendaraan (Poot, Kant dan Wagelmans, 2002). VRP merupakan manajemen distribusi barang yang memperhatikan pelayanan, periode waktu tertentu, sekelompok konsumen dengan sejumlah kendaraan yang berlokasi pada satu atau lebih depot yang dijalankan oleh sekelompok pengemudi dengan menggunakan *road network* yang sesuai. VRP dapat didefinisikan sebagai suatu pencarian solusi yang meliputi penentuan

sejumlah rute, dimana masing-masing rute dilalui oleh satu kendaraan yang berawal dan berakhir di depot asalnya, sehingga kebutuhan/permintaan semua pelanggan terpenuhi dengan tetap memenuhi kendala operasional yang ada, juga dengan meminimalisasi biaya transportasi global (Toth dan Vigo, 2002).

Menurut Toth dan Vigo (2002), karakteristik utama VRP berdasarkan komponen-komponennya adalah sebagai berikut.

1. Jaringan jalan, biasanya direpresentasikan dalam sebuah *graph* (diagram) yang terdiri dari *arc* (lengkung atau bagian-bagian jalan) dan *vertex* (titik lokasi konsumen dan depot). Tiap lengkung diasosiasikan dengan biaya (jarak) dan waktu perjalanan (tergantung jenis kendaraan, kondisi/karakteristik jalan, dan periode pelintasan).
2. Konsumen, ditandai dengan *vertex* (titik) dan biasanya memiliki hal-hal seperti berikut.
 - Jumlah permintaan barang (untuk dikirim ataupun diambil), jenis barang dapat berbeda-beda.
 - Periode pelayanan tertentu (*time windows*), dimana di luar rentang waktu tersebut konsumen tidak dapat menerima pengiriman maupun pengambilan.
 - Waktu yang dibutuhkan untuk menurunkan atau memuat barang (*loading/unloading time*) pada lokasi konsumen, biasanya tergantung dari jenis kendaraan.
 - Pengelompokan (*subset*) kendaraan yang tersedia untuk melayani konsumen (sehubungan dengan keterbatasan akses atau persyaratan pemuatan dan penurunan barang).
 - Prioritas atau pinalti sehubungan dengan kemampuan kendaraan untuk melayani permintaan.
3. Depot, juga ditandai dengan suatu titik, merupakan ujung awal dan akhir dari suatu rute kendaraan. Tiap depot memiliki sejumlah kendaraan dengan jenis dan kapasitas tertentu yang dapat digunakan untuk mendistribusikan produk.
4. Kendaraan / armada angkut, memiliki

- Depot asal, dan kemungkinan untuk mengakhiri rutenya di depot lain.
 - Kapasitas (berat, volume atau jumlah palet yang dapat diangkut)
 - Kemungkinan untuk dipisah menjadi beberapa kompartemen untuk mengangkut barang dengan jenis yang berbeda-beda.
 - Alat yang tersedia untuk operasi (pemuatan atau penurunan barang).
 - Pengelompokan (*subset*) lintasan/lengkung dari diagram jaringan jalan.
 - Biaya yang berhubungan dengan penggunaan kendaraan tersebut (unit per jarak, unit per waktu, unit per rute, dan lainnya).
5. Pengemudi, memiliki kendala seperti jam kerja harian, jumlah dan jam istirahat, durasi maksimum perjalanan, serta lembur yang biasanya juga dikenakan pada kendaraan yang digunakan.

Dalam membuat konstruksi rute, terdapat beberapa kendala yang harus dipenuhi, seperti jenis barang yang diangkut, kualitas dari pelayanan, juga karakteristik konsumen dan kendaraan. Beberapa kendala operasional yang sering ditemui misalnya:

- Pada tiap rute, besar muatan yang diangkut oleh kendaraan tidak boleh melebihi kapasitas kendaraan tersebut.
- Konsumen yang dilayani dalam sebuah rute dapat hanya merupakan pengiriman atau pengambilan, atau mungkin keduanya.
- Konsumen mungkin hanya dapat dilayani dalam rentang waktu tertentu (*time windows*) dan jam kerja dari pengemudi kendaraan yang melayaninya.
- Kendala prioritas juga mungkin akan timbul ketika suatu konsumen harus dilayani sebelum konsumen lain. Kendala seperti ini biasanya terdapat pada kasus *pickup and delivery* (pengambilan dan pengiriman dalam satu rute) atau VRP *with backhauls* dimana pengambilan baru dapat dilakukan setelah semua pengiriman selesai dikarenakan kesulitan dalam mengatur peletakan muatan.

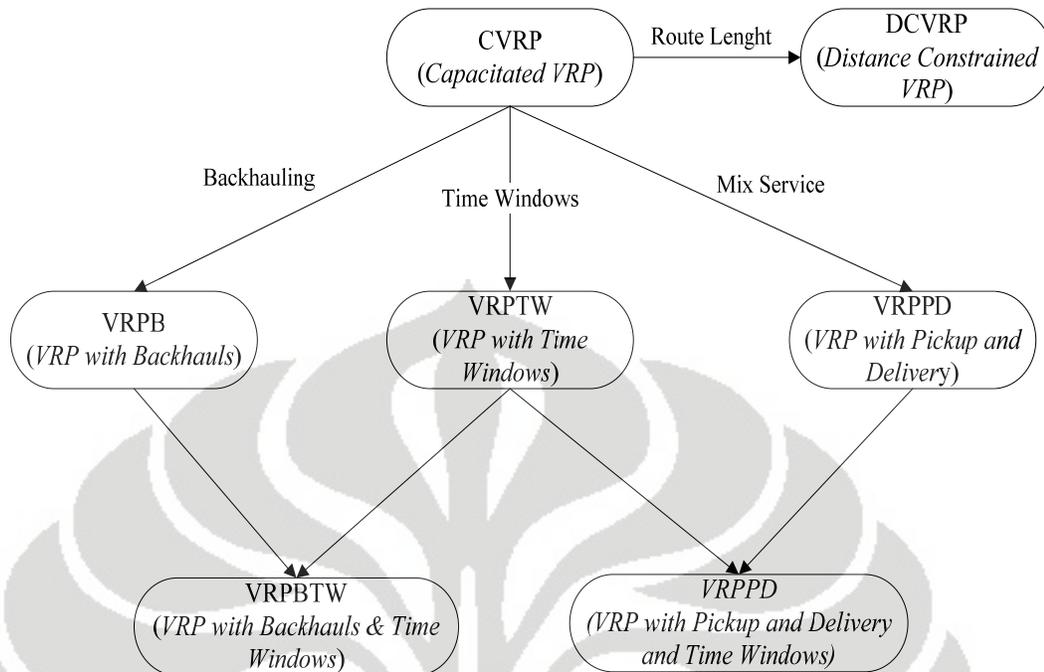
Menurut Toth dan Vigo (2002), terdapat empat tujuan umum dalam VRP, yaitu:

- Meminimumkan biaya transportasi global, terkait dengan jarak dan biaya tetap yang berhubungan dengan kendaraan.
- Meminimumkan jumlah kendaraan yang dibutuhkan untuk melayani semua konsumen.
- Menyeimbangkan rute-rute dalam hal waktu perjalanana dan muatan kendaraan.
- Meminimumkan penalti akibat pelayanan yang kurang memuaskan terhadap konsumen, seperti ketidaksanggupan melayani konsumen secara penuh ataupun keterlambatan pengiriman.

Menurut Toth dan Vigo (2002) ditemukan beberapa kelas atau variasi permasalahan utama dalam VRP, yaitu:

- *Capacitated VRP (CVRP)*, merupakan kelas VRP yang paling sederhana dan yang paling banyak dipelajari dimana kendala yang ada hanya berupa kapasitas kendaraan yang terbatas.
- *Distance Constrained VRP (DCVRP)*, merupakan VRP dengan kendala batasan panjang rute.
- *VRP with time windows (VRPTW)*, yaitu kasus VRP dimana setiap konsumen memiliki batasan rentang waktu pelayanan.
- *VRP with Pick up and Delivery (VRPPD)*, merupakan VRP dengan pelayanan campuran, yaitu pengiriman dan pengambilan barang dalam satu rute.
- *VRP with Backhauls (VRPB)*, dimana pengambilan baru dapat dilakukan setelah semua pengiriman selesai.

Gambar berikut menunjukkan hubungan antara kelas-kelas VRP tersebut.



Gambar 2.1 Permasalahan Dasar Kelas-Kelas VRP dan Keterkaitannya

(Sumber: Toth & Vigo, 2002)

2.1.2 Vehicle Routing Problem with Time Windows

Vehicle routing problem with time windows (VRPTW) merupakan perluasan dari VRP yang paling sering ditemukan dalam pengambilan keputusan mengenai distribusi barang. Setiap kendaraan yang bertugas pada VRP jenis ini hanya dapat keluar dari depot pada jam kerja depot dan melayani konsumen pada jangka waktu tertentu yang ditentukan oleh pihak konsumen. Tiap kendaraan juga harus kembali lagi ke depot sebelum jam kerja depot berakhir. (Braysy dan Gendreau, 2002). Pada sejumlah kasus VRPTW, terdapat *time windows* yang bersifat lunak (*soft*). Pada kasus semacam ini, penalti keterlambatan ikut dimasukkan ke dalam fungsi tujuan (Taillard et al, 1998)

Tujuan dari VRPTW adalah menentukan sejumlah rute untuk melayani seluruh konsumen dengan biaya terkecil (dalam hal ini yang dimaksud dengan biaya adalah jarak tempuh) tanpa melanggar batasan kapasitas dan waktu tempuh kendaraan serta batasan waktu yang diberikan oleh pihak pelanggan. Jumlah rute yang ditentukan tidak boleh melebihi jumlah kendaraan yang ada.

2.1.2.1 Model Matematis VRPTW

Tan, Lee dan Zhu (1999) menyatakan bahwa VRPTW dapat didefinisikan oleh sejumlah kendaraan yang identik (V), sebuah titik khusus yang disebut depot, sejumlah pelanggan yang harus dilayani (C), dan sebuah rute yang menghubungkan depot dengan sejumlah pelanggan. Asumsikan bahwa terdapat K kendaraan sehingga $V=\{0, 1, 2, \dots, K-1\}$ dan terdapat $N+1$ pelanggan sehingga $C=\{0, 1, 2, \dots, N\}$. Depot dianggap sebagai pelanggan ke-0.

Sebuah rute akan dimulai dari depot kemudian menuju sejumlah pelanggan pada rute tersebut dan akan berakhir kembali di depot. Jumlah rute yang tercipta pada VRPTW sama dengan jumlah kendaraan yang digunakan, yaitu K . Jadi, bila diilustrasikan akan ada K garis yang keluar dari depot dan K garis yang masuk ke depot. Setiap rute memiliki biaya sebesar c_{ij} (jarak antara titik i ke titik j) dan waktu tempuh t_i (waktu antara titik i ke titik j). Setiap pelanggan dalam sebuah rute hanya dapat dilayani sebanyak 1 kali oleh 1 kendaraan. Karena tiap kendaraan memiliki kapasitas maksimum sebesar q_k dan masing-masing pelanggan memiliki permintaan yang berbeda-beda, maka q_k harus lebih besar atau sama dengan jumlah permintaan seluruh pelanggan yang berada pada suatu rute. Disamping itu, setiap pelanggan harus dilayani dalam interval waktu tertentu. Kendaraan yang tiba sebelum interval waktu harus menunggu, sedangkan kendaraan yang datang setelah interval waktu akan mendapatkan penalti. Setiap kendaraan harus menyelesaikan tugasnya selama jam kerja depot.

Untuk menyelesaikan masalah VRPTW, kita harus menentukan sejumlah rute (sekaligus urutan pengantaran di tiap rute) untuk melayani seluruh konsumen dengan biaya terkecil tanpa melanggar batasan-batasan yang disebutkan diatas.

Ada dua jenis variable keputusan dalam VRPTW. Variabel keputusan pertama adalah x_{ijk} ($i, j=0, 1, 2, \dots, N; k=0, 1, 2, \dots, k : i \neq j$). Variable ini akan bernilai 1 bila kendaraan k berangkat dari titik i menuju j dan akan bernilai 0 untuk keadaan sebaliknya. Variable keputusan kedua adalah t_i . Variabel ini menunjukkan waktu sebuah kendaraan saat mulai melayani pada titik i .

Model matematis untuk masalah ini adalah sebagai berikut:

$$\text{Fungsi Tujuan} = \text{Min} \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^{K-1} c_{ij} x_{ijk} \quad (2.1)$$

Kendala:

$$\sum_{k=0}^{K-1} \sum_{j=1}^N x_{t_0jk} = K \text{ for } t = 0 \quad (2.2)$$

$$\sum_{k=0}^{K-1} \sum_{j=0, j \neq t}^N x_{tjk} = 1 \text{ for } t = 1, 2, \dots, N \quad (2.3)$$

$$\sum_{i=0, i \neq h}^N x_{ihk} - \sum_{j=0, j \neq h}^N x_{hjk} = 0 \quad \forall h \in [1, N], k \in [0, K-1] \quad (2.4)$$

$$\sum_{i=0}^N m_i \sum_{j=0, j \neq i}^N x_{tjk} \leq q_k \quad \forall k \in [0, K-1] \quad (2.5)$$

$$\sum_{i=0}^N \sum_{j=0, j \neq i}^N x_{tjk} (t_{ij} + f_i + w_i) \leq r_k \quad \forall k \in [0, K-1] \quad (2.6)$$

$$t_0 = 0 \quad (2.7)$$

$$t_i + x_{tjk} (t_{ij} + f_i + w_i) \leq t_j \quad t, j \in [1, N], t \neq j, k \in [0, K-1] \quad (2.8)$$

$$a_i \leq t_i \leq b_i \quad k \in [0, K-1] \quad (2.9)$$

Keterangan:

K = Jumlah kendaraan

N = Jumlah pelanggan

C_i = pelanggan ke- i

C_0 = depot

C_{ij} = jarak dari titik i ke titik j

t_{ij} = waktu tempuh titik i ke titik j

m_i = permintaan di titik i

q_k = kapasitas kendaraan k

t_i = waktu sampai di titik i

e_i = *earliest arrival time* di titik i

f_i = *service time* di titik i

w_i = waktu tunggu di titik i

r_k = waktu kerja maksimum mobil k

Kendala 1 untuk memastikan bahwa ada sebanyak K rute yang keluar dari depot. Kendala 2 dan 3 memastikan bahwa sebuah pelanggan hanya dilayani oleh sebuah kendaraan. Kendala 4 memastikan bahwa muatan yang dibawa tidak melebihi kapasitas maksimum kendaraan. Kendala 5 merupakan kendala batasan *travel time* kendaraan (*time windows* depot). Kendala 6 sampai dengan kendala 8 memastikan bahwa pengantaran dilakukan dalam *time windows* pelanggan.

2.2 Metode Penyelesaian VRP

Menurut Toth dan Vigo (2002), secara umum VRP dapat diselesaikan dengan menggunakan 2 jenis pendekatan, yaitu eksak dan heuristik. Kemudian, kelas penyelesaian heuristik dalam VRP dapat dibagi menjadi dua, yaitu heuristik klasik dan heuristik modern (metaheuristik). Heuristik klasik dikembangkan antara tahun 1960 hingga 1990, dilanjutkan dengan pengembangan algoritma metaheuristik hingga kini.

2.2.1 Pendekatan Eksak

Pada solusi eksak, dilakukan pendekatan dengan menghitung setiap solusi yang mungkin sampai satu terbaik dapat diperoleh. Terdapat beberapa algoritma eksak utama penyelesaian VRP, yaitu:

- *Branch and Bound*
- *Branch and Cut*
- *Set Covering Based*

Secara umum penggunaan metode eksak untuk penyelesaian VRP akan menghabiskan waktu yang lama. Hal tersebut dikarenakan VRP termasuk dalam permasalahan NP-hard (*Non Polynomial-hard*) dimana kompleksitas penyelesaian permasalahan akan meningkat secara eksponensial dengan semakin besarnya permasalahan. Hingga kini, belum ada algoritma eksak yang mampu menyelesaikan kasus-kasus yang terdiri lebih dari 50 konsumen secara konsisten (Toth dan Vigo, 1998). Oleh karena itu, berbagai penelitian terhadap algoritma heuristik telah dilakukan untuk menyederhanakan penyelesaian VRP dan akan dibahas pada bagian selanjutnya.

2.2.2 Heuristik Klasik

Prosedur konstruksi dan perbaikan solusi masalah VRP yang umum digunakan saat ini berasal dari kelas heuristik klasik. Metode-metode tersebut tidak terlalu mengeksplorasi ruang pencarian solusi dan biasanya menghasilkan solusi dengan kualitas yang cukup baik dengan waktu perhitungan yang singkat. Karena penggunaannya dapat diperluas dengan menambahkan kendala-kendala yang

mungkin timbul dalam kehidupan sehari-hari, maka metode-metode tersebut masih dikomersilkan.

Secara umum, terdapat tiga metode heuristik klasik VRP, yaitu:

- Heuristik konstruktif (*constructive heuristics*), yaitu penyusunan solusi yang memungkinkan dengan memperhatikan biaya solusi tersebut, tanpa dilakukan fase perbaikan.
- Heuristik dua fase (*two-phase heuristics*), dimana penyelesaian masalah dibagi menjadi dua bagian, yaitu pengelompokan (*clustering*) rute dan pengkonstruksian rute-rute tersebut. Urutan pengerjaan dapat berupa *Route First - Cluster Second*, dimana pengelompokan dilakukan setelah rute dibuat, atau *Cluster First - Route Second*, dimana rute dibuat setelah pengelompokan dilakukan.
- Metode perbaikan (*improvement method*), yaitu metode yang bertujuan untuk memperbaiki solusi yang mungkin dengan melakukan pertukaran titik dalam satu rute atau antar-rute.

Dalam metode heuristik konstruktif, terdapat dua teknik penyelesaian masalah VRP, yaitu dengan penyatuan dua rute berdasarkan kriteria penghematan dan dengan penugasan secara bertahap dari masing-masing titik ke dalam rute berdasarkan biaya penugasan (*insertion cost*). Untuk teknik yang pertama, biasa digunakan algoritma penghematan (*saving algorithm*). Metode *saving* ini diperkenalkan oleh Clarke dan Wright pada tahun 1964 dan merupakan metode heuristik konstruktif yang paling populer dan sering digunakan. Kemudian, untuk teknik yang kedua, *sequential Insertion heuristics*, dapat digunakan algoritma Mole & Jameson (1976) dan algoritma Christofides, Mingozzi & Toth (1979). Kedua algoritma ini bekerja berdasarkan penugasan secara bertahap (*sequential insertion*).

Metode heuristik dua fase dapat dibagi menjadi dua kelompok, yaitu:

- *Cluster First - Route Second* (pengelompokan kemudian pembuatan rute), terdiri dari:
 - Metode pengelompokan elementer, yaitu dengan melakukan suatu pengelompokan dan kemudian menentukan rute pada tiap kelompok. Contoh algoritmanya adalah algoritma penyapuan (*Sweep Algorithm*),

algoritma Fisher & Jaikumar (1981), dan algoritma Bramel & Simichi-Levi (1995).

- *Truncated Branch-and-Bound* yang merupakan penyederhanaan dari algoritma eksak *Branch-and-Bound*.
- Algoritma kelopak bunga (*Petal Algorithm*) yang menghasilkan sekeluarga besar dari kelompok-kelompok yang saling bertumpuk (*overlapping cluster*) dan rute di dalamnya, kemudian menyeleksi sejumlah rute yang memungkinkan.
- *Route First – Cluster Second* (pembuatan rute kemudian pengelompokan). Dalam fase pertama metode ini, dibuat sebuah tur raksasa berdasarkan TSP (*Traveling salesman Problem*) yang tidak melibatkan kendala-kendala sampingan. Pada fase kedua dilakukan pemecahan tur tersebut menjadi beberapa kelompok rute yang memungkinkan.

Kelompok metode heuristik ketiga adalah metode perbaikan, dibagi menjadi metode perbaikan untuk satu rute dan metode perbaikan untuk banyak rute. Metode Perbaikan untuk satu rute menggunakan prosedur perbaikan TSP, yaitu mekanisme λ -opt. Dalam mekanisme ini, sejumlah λ titik dipindahkan dari tur, kemudian titik-titik tersebut digabungkan kembali untuk perbaikan. Pada algoritma *Lin-Kernighan Algorithm* (Errera, 2001) sebuah tur disebut λ -optimal apabila sudah tidak ada lagi kemungkinan untuk memperoleh tur yang lebih optimal dengan memindahkan sejumlah λ titik manapun dari rantainya dengan set λ titik lain.

Jika dilihat dari kualitas solusi yang dihasilkan, heuristik klasik berdasarkan konstruksi sederhana dan teknik perbaikan lokal tidak dapat menandingi implementasi metode heuristik modern. Namun karena kesederhanaan penerapannya, algoritma heuristik klasik seperti *Saving Algorithm* tetap menjadi metode yang paling populer dan banyak digunakan sebagai dasar dalam perangkat lunak komersil.

2.2.2.1 *Sweep Method*

Sweep method merupakan salah satu metode dalam penyelesaian VRP dengan pengelompokan elementer. Metode ini merupakan metode yang sederhana

dalam perhitungannya, bahkan untuk memecahkan masalah dengan ukuran yang cukup besar. Keakuratan metode ini rata-rata kesalahan perhitungannya adalah sebesar 10% (Ballou dan Agarwal, 1998).

Keakuratan metode ini adalah pada cara pembuatan jalur rutenya. Prosesnya terdiri dari dua tahap, pertama titik pemberhentian ditentukan untuk kendaraan yang ada. Tahap kedua adalah menentukan urutan titik pemberhentian pada rute. Karena melibatkan dua tahapan proses maka total waktu dalam suatu rute dan batasan waktu tidak dapat ditangani dengan baik oleh metode ini.

Metode ini termasuk di dalam jenis metode *cluster* atau pengelompokan, yang mana pengelompokan awal dilakukan dengan menggabungkan perhentian-perhentian yang setiap kelompok mengakomodasi volume masing-masing perhentian. Volume total perhentian dari satu rute mungkin akan melebihi kapasitas kendaraan. Jika terjadi hal tersebut maka beberapa perhentian dipindahkan ke kendaraan yang kapasitasnya belum penuh. Relokasi seperti ini dilakukan dengan menggunakan metode transportasi *linear programming*.

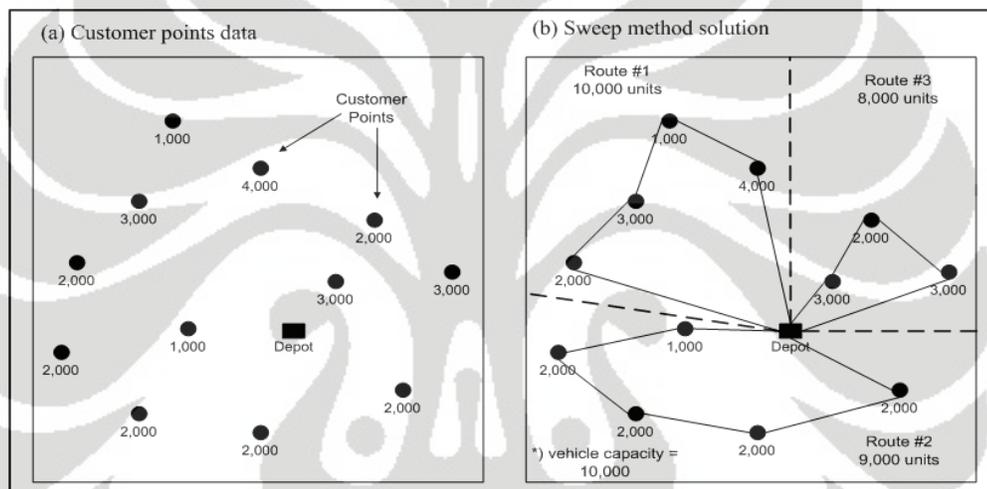
Hal yang menarik dari metode ini adalah perhentian dikelompokkan berdasarkan kedekatan dan logikanya akan menghasilkan jarak total yang rendah. Ketika volume suatu rute melebihi kapasitas kendaraan, relokasi perhentian ke rute lain dilakukan untuk mendapatkan keseimbangan optimum. Karena pengelompokan terpisah dari pengurutan, maka kendala waktu tidak dapat diselesaikan dengan menggunakan metode ini.

Metode *sweep* dapat dilakukan dengan langkah-langkah berikut (Ballou, 2004):

1. Gambarkan lokasi dari semua titik kirim dan juga lokasi depot dalam suatu peta.
2. Buat suatu garis lurus ke arah manapun yang berpangkal dari lokasi depot. Kemudian putar (dengan titik putar adalah lokasi depot) garis tersebut searah atau berlawanan arah jarum jam sampai memotong suatu titik perhentian. Kemudian perhatikan apakah volume permintaan pada titik perhentian tersebut melebihi kapasitas kendaraan. Jika tidak lanjutkan pergerakan rotasi garis sampai memotong titik perhentian selanjutnya. Perhatikan lagi apakah volume permintaan kumulatif melebihi kapasitas

kendaraan. Jika demikian, jangan masukkan titik perhentian terakhir tersebut dalam rute. Kemudian, mulai bentuk rute baru dimulai dari titik perhentian yang terakhir tadi. Lanjutkan pergerakan rotasi penyapuan garis sampai semua titik perhentian telah masuk dalam suatu rute tertentu.

- Setelah dilakukan *clustering* titik-titik perhentian kedalam sejumlah rute. Maka dilanjutkan dengan menentukan urutan perhentian dalam setiap rute untuk meminimalkan jarak. Penentuan urutan perhentian ini dapat dilakukan dengan menggunakan metode *tear-drop* atau algoritma lain yang dapat memecahkan *traveling salesman problem*.



Gambar 2.2 Contoh Solusi Metode *Sweep*

(Sumber: Ballou, 2004)

2.2.2.2 Saving Method

Metode *Saving* ini ditemukan oleh Clarke dan Wright pada tahun 1964. Konsep dasar dari metode ini adalah melakukan penghematan jarak tempuh dengan melakukan penyatuan dua rute. Rute yang dipilih untuk digabung adalah yang memberikan penghematan (*saving*) terbesar dari jarak total yang ditempuh untuk semua rute. Penggabungan rute ini dilakukan terus sehingga tidak ada lagi penghematan yang dapat dilakukan atau sampai kapasitas kendaraan tidak dapat ditambah lagi.

Keunggulan dari metode ini adalah hambatan-hambatan dalam praktek dapat diakomodir, misalnya pengambilan dan pengiriman dalam satu rute, adanya

Universitas Indonesia

waktu khusus untuk pengiriman, dan tipe kendaraan yang beragam. Namun metode ini juga memiliki kelemahan, yaitu waktu perhitungan cenderung meningkat secara geometri seiring dengan bertambahnya jumlah rute atau perhentian.

Metode *saving* ini cukup fleksibel untuk menangani beberapa kendala nyata yang memiliki cakupan luas, relatif cepat dalam perhitunagn untuk persoalan dengan jumlah titik pemberhentian yang banyak dan kemampuan untuk mendapatkan hasil pemecahan masalah yang mendekati optimal (Ballou, 2004). Metode ini dapat membuat rute dan mengurutkan titik-titik pemberhentian dalam rute secara bersamaan.

Tujuan dari metode *saving* adalah untuk meminimalkan total jarak yang harus ditempuh oleh seluruh kendaraan dan secara tidak langsung meminimalkan jumlah kendaraan yang dibutuhkan untuk melayani semua titik pemberhentian.

2.2.2.3 *Nearest Neighbourhood Method*

Prosedur *Nearest Neighbourhood* pada awalnya digunakan untuk menyelesaikan permasalahan TSP (*Traveling Salesman Problem*), yaitu membentuk suatu konstruksi rute dengan menggunakan satu buah kendaraan. Namun demikian, metode ini juga dapat digunakan untuk menyelesaikan masalah VRP. Metode ini termasuk dalam kelas *Route First – Cluster Second* (pembuatan rute kemudian pengelompokan). Dalam fase pertama metode ini, dibuat sebuah tur raksasa berdasarkan TSP (*Traveling salesman Problem*) yang tidak melibatkan kendala-kendala sampingan. Kemudian, pada fase kedua dilakukan pemecahan tur tersebut menjadi beberapa kelompok rute yang memungkinkan berdasarkan batasan kapasitas kendaraan dan juga batasan waktu.

Pada tahap pertama, satu rute besar dibuat berdasarkan jarak perjalanan dari titik yang terakhir dikunjungi ke titik terdekat yang ada pada *network*. Prosedurnya adalah sebagai berikut.

1. Mulai perjalanan dari titik depot.
2. Cari titik konsumen yang paling dekat dengan titik yang terakhir dikunjungi.
3. Kembali ke langkah 2, sampai semua titik konsumen dikunjungi

4. Hubungkan titik awal depot dengan semua titik hingga titik konsumen terakhir untuk membentuk rute besar yang lengkap
5. Pecah rute besar tersebut menjadi beberapa rute kecil berdasarkan kapasitas kendaraan dan batasan waktu yang ada.

2.2.3 Metaheuristik

Heuristik modern atau metaheuristik adalah prosedur pencarian solusi umum untuk melakukan eksplorasi yang lebih dalam pada daerah yang menjanjikan dari ruang solusi yang ada (Dreo, Petrowsky dan Taillard, 2006) Perbedaannya dengan heuristik klasik adalah diperbolehkannya perusakan solusi atau penurunan fungsi tujuan. Kualitas solusi yang dihasilkan dari metode ini jauh lebih baik daripada heuristik klasik. Beberapa contoh metaheuristik adalah *simulated annealing*, *deterministic annealing*, *genetic algorithm*, *neural network*, *ant colony system*, dan *tabu search*.

Prinsip dasar algoritma metaheuristik adalah pencarian lokal dan pencarian populasi. Dalam metode pencarian lokal, eksplorasi yang intensif dilakukan terhadap ruang solusi dengan berpindah dari satu solusi ke solusi tetangga lainnya yang potensial dalam satu lingkungan (*neighbourhood*). Algoritmanya biasa dimulai dengan solusi awal x_1 dan berpindah pada tiap iterasi t dari solusi x_t ke solusi x_{t+1} dalam satu lingkungan $N(x_t)$ dari x_t , hingga persyaratan tertentu dicapai. Jika fungsi tujuan $f(x)$ menandakan biaya dari x , maka $f(x_{t+1})$ tidak selalu lebih kecil dari $f(x_t)$. Oleh karenanya dibutuhkan kehati-hatian untuk menghindari perputaran (*cycling*) pada iterasi.

Pencarian populasi meliputi rekombinasi dari solusi-solusi terbaik untuk menghasilkan solusi-solusi turunan yang baik pula. Contohnya adalah *genetic algorithm* yang memperhatikan populasi solusi pada setiap iterasi. Tiap populasi diperoleh dari populasi sebelumnya dengan mengkombinasikan elemen-elemen terbaiknya dan membuang elemen yang buruk untuk menghasilkan populasi turunan.

Ant colony system merupakan pendekatan konstruktif dimana sejumlah solusi baru dibuat pada setiap iterasi menggunakan sebagian informasi yang didapat dari iterasi-iterasi sebelumnya. Taillard et al. (1998) mengemukakan

bahwa *tabu search*, *genetic algorithm*, dan *ant colony system* merupakan metode yang merekam informasi mengenai solusi-solusi yang ditemui selama proses pencarian berjalan dan menggunakannya untuk memperoleh solusi yang lebih baik.

Metaheuristik secara konsisten mampu menghasilkan solusi yang berkualitas tinggi jika dibandingkan dengan heuristik klasik, meskipun lebih memakan waktu yang lebih lama.

2.3 Tabu Search

2.3.1 Pengertian Umum

Kata tabu atau *taboo* berasal dari bahasa Tongan yang merupakan salah satu bahasa polynesia yang digunakan oleh penduduk pribumi dari pulau Tonga untuk mengungkapkan sesuatu yang tidak boleh disentuh karena merupakan sesuatu yang keramat (Glover dan Laguna, 1997). Dalam kamus Webster, kata *taboo* berarti “suatau larangan yang bersifat sosial sebagai mekanisme protektif” atau dapat berarti “sesuatu yang dilarang karena mengandung resiko”. Resiko yang dihindari dalam hal ini adalah sesuatu yang sifatnya kontra produktif.

Pengertian yang terakhir merupakan pengertian yang sesuai dengan apa yang dimaksud dalam *tabu search*. Resiko yang hendak dihindari adalah resiko dari jalan yang kontra produktif, misalnya jalan yang membawa pada jebakan dan tidak ada harapan untuk kabur. Asosiasi yang paling penting dengan penggunaan kata tabu secara tradisional adalah bahwa pada kenyatannya sasuatu yang tabu biasanya dipahami dengan cara disebarakan atau diturunkan melalui ingatan sosial yang mengalami modifikasi dari waktu ke waktu. Penafsiran ini menciptakan hubungan yang erat dengan kata tabu pada *tabu search*. Elemen yang dilarang dalam *tabu search* juga mendapatkan statusnya dengan kepercayaan pada *memory* atau ingatan yang berkembang, yang memungkinkan perubahan status tergantung pada waktu dan keadaan.

Tabu search sendiri pertama kali diperkenalkan oleh Fred Glover pada tahun 1986 dan dengan cepat menjadi salah satu metode pencarian lokal untuk optimasi kombinatorial yang paling banyak digunakan. *Tabu search* merupakan suatu strategi meta (*meta-strategy*) untuk memandu prosedur pencarian lokal yang

bertujuan untuk mencegah proses pencarian terjebak di optimal lokal (Hertz, Taillard dan Werra, 1997). Untuk mencapai hal tersebut, gerakan-gerakan yang sudah pernah dilakukan tidak diperbolehkan untuk diulang lagi dan disimpan didalam daftar tabu (*tabu list*). Untuk dapat melakukan hal tersebut, *tabu search* memiliki fitur yang disebut *adaptive memory* dan *responsive exploration*.

Fitur *adaptive memory* dan *responsive exploration* dalam *tabu search* memungkinkan dilakukannya implementasi prosedur yang dapat melakukan pencarian berbagai solusi secara cepat dan efektif. Pencarian lokal pada *tabu search* diarahkan oleh informasi yang dikumpulkan selama pencarian. Hal tersebut membuat *tabu search* berbeda dengan desain tanpa memori yang sangat bergantung pada proses semi acak yang mengimplementasikan suatu bentuk *sampling*. Contoh dari metode tanpa memori adalah *heuristic greedy* dan pendekatan *annealing* dan *genetic* yang terinspirasi oleh metafor fisika dan biologi. *Adaptive memory* pada *tabu search* juga berbeda dengan memori yang kaku pada algoritma *branch and bound*.

Upaya untuk melakukan eksplorasi responsif dalam *tabu search*, baik itu implementasi *deterministic* maupun *probabilistic*, didasarkan pada pemahaman bahwa suatu pilihan berdasarkan strategi yang buruk dapat menghasilkan informasi yang lebih banyak daripada suatu pilihan acak yang baik. Dalam suatu sistem yang menggunakan memori, sebuah pilihan buruk yang berdasarkan strategi dapat memberikan petunjuk yang bermanfaat tentang bagaimana strategi tersebut dapat diubah menjadi lebih baik.

2.3.2 Penggunaan Memori

Struktur memori dalam *tabu search* dijalankan dengan dasar pada empat dimensi prinsip, yang terdiri atas *recency*, *frequency*, *quality*, dan *influence* (Glover dan Laguna, 1997). Dimensi kualitas mengacu pada kemampuan untuk mengetahui kelebihan dari solusi-solusi yang dikunjungi selama pencarian. Pada konteks tersebut, memori dapat digunakan untuk mengidentifikasi elemen-elemen umum tentang solusi yang baik atau untuk mengidentifikasi tentang jalan yang membawa pada solusi tersebut. Pada prakteknya, kualitas menjadi landasan untuk pembelajaran berbasis intensif (*intencive-based learning*) dimana penghargaan

diberikan untuk mendukung tindakan-tindakan yang menghasilkan solusi yang baik dan pinalti diberikan untuk menghindari solusi yang buruk. Fleksibilitas struktur memori seperti ini memungkinkan pencarian untuk diarahkan pada lingkungan *multi objective*, dimana kebaikan dari suatu arah pencarian tertentu dapat ditentukan oleh lebih dari satu fungsi. Konsep kualitas dari *tabu search* lebih luas dibandingkan dengan metode optimasi standar.

Memori yang digunakan dalam *tabu search* dapat bersifat eksplisit dan juga bersifat atributif. Memori eksplisit merekam seluruh solusi secara lengkap, biasanya terdiri dari solusi-solusi penting yang dikunjungi selama pencarian. Perluasan dari dapat memori ini merekam solusi penting yang sangat atraktif yang merupakan solusi tetangga yang belum tereksplorasi.

Sebagai alternaif, *tabu search* menggunakan memori atributif untuk tujuan sebagai panduan atau pengarah. Jenis memori ini merekam informasi tentang atribut-atribut solusi yang mengalami perubahan dalam proses perpindahan dari satu solusi ke solusi yang lain. Sebagai contoh, dalam permasalahan grafik dan jaringan (*graph and network setting*), atribut dapat berupa titik/*node* atau arah/*arc* yang ditambahkan, dihilangkan atau direposisi dengan mekanisme perpindahan. Dalam penjadwalan produksi, daftar (*index of jobs*) dapat digunakan sebagai atribut untuk mencegah atau mendorong ke arah pencarian tertentu.

2.3.3 Karakteristik dan Prosedur *Tabu Search*

Sebelum membahas mengenai karakteristik dan prosedur *tabu search*, ada baiknya jika diketahui terlebih dahulu mengenai metode *descent*. Metode *descent* ini merupakan metode pencarian solusi tetangga yang paling populer yang digunakan untuk menemukan aproksimasi dari nilai minimum suatu fungsi tujuan f dalam set S (Hertz, Taillard, dan Werra, 1997). Langkah-langkahnya adalah sebagai berikut:

- Langkah 1.** Pilih solusi awal i dalam S
- Langkah 2.** Temukan solusi terbaik j dalam lingkungan $N(i)$ sehingga $f(j) \leq f(k)$ untuk tiap k dalam $N(i)$.
- Langkah 3.** Jika $f(j) \geq f(k)$ maka berhenti. Lainnya maka $i = j$ dan kembali ke **Langkah 2**.

Metode seperti diatas dapat berhenti pada nilai optimum lokal dan bukan optimum global dari f . Secara umum $N(i)$ tidak dijelaskan secara eksplisit. *Tabu search* dapat dianggap sebagai metode pencarian solusi tetangga yang lebih mendetail dibandingkan dengan metode *descent* (Hertz, Taillard dan Werra, 1997). Agar proses eksplorasi tetap efisien, diperlukan suatu media untuk menyimpan informasi lokal (nilai terkini dari fungsi tujuan) juga informasi lainnya yang terkait dengan proses eksplorasi. Penggunaan memori yang sistematis ini merupakan salah satu karakteristik paling penting dari *tabu search*. Metode eksplorasi pada umumnya hanya menyimpan nilai $f(i^*)$ dari solusi terbaik i^* terkini. Pada *tabu search*, selain menyimpan solusi terbaik terkini, juga menyimpan informasi mengenai seluruh perjalanan hingga solusi yang paling akhir ditemui. Informasi tersebut akan digunakan sebagai panduan pergerakan dari i ke solusi selanjutnya j yang dipilih dari $N(i)$.

Peran memori pada *tabu search* adalah untuk membatasi pilihan hanya pada bagian (*subset*) $N(i)$ tertentu dengan melarang pergerakan ke beberapa solusi tetangga. Dengan kata lain, struktur tetangga $N(i)$ dari solusi i akan berbeda-beda tiap iterasi (*dynamic neighbourhood*).

Apabila diberikan suatu permasalahan optimasi: S adalah set solusi yang mungkin dari fungsi $f: S \rightarrow \mathbb{R}$, temukan solusi i^* dalam S sehingga $f(i^*)$ dapat diterima berdasarkan beberapa kriteria (misalnya apabila $f(i^*) \leq f(i)$ untuk setiap i dalam S), maka reformulasi algoritma *descent* untuk mendekati algoritma *tabu search* adalah sebagai berikut (dengan i^* adalah solusi terbaik terkini dan k adalah iterasi):

- Langkah 1.** Pilih solusi awal i dari S . Ubah $i^* = i$ dan $k = 0$.
- Langkah 2.** Ubah $k = k+1$ dan buat *subset* solusi V^* dari $N(i, k)$.
- Langkah 3.** Pilih solusi terbaik j dari V^* (dengan memperhatikan f yang digunakan) dan ubah $i = j$.
- Langkah 4.** Jika $f(i) < f(i^*)$ maka ubah $i^* = i$
- Langkah 5.** Jika persyaratan penghentian ditemui maka berhenti. Lainnya maka kembali ke **Langkah 2**.

Beberapa persyaratan penghentian pada *tabu search* adalah sebagai berikut:

- Apabila $f(i) \geq f(i^*)$.

- Apabila $N(i, k+1) = \emptyset$.
- Apabila k lebih besar dari umlah iterasi maksimum yang diperbolehkan.
- Jumlah iterasi sejak perbaikan i^* terakhir lebih besar dari angka tertentu.
- Dapat dibuktikan bahwa solusi optimum telah diperoleh.

Aturan penghentian tersebut dapat berpengaruh terhadap prosedur pencarian dan hasilnya, namun definisi $N(i, k)$ pada tiap iterasi k dan pemilihan V^* juga sangat mempengaruhi. Definisi $N(i, k)$ berarti bahwa beberapa solusi yang baru ditemui dipindahkan dari $N(i)$ dan disebut solusi tabu yang harus dihindari pada beberapa iterasi selanjutnya. Daftar solusi tabu (*tabu list*) pada iterasi k yang menyimpan $[T]$ solusi terakhir akan mencegah perputaran sejumlah maksimum $[T]$.

Untuk setiap solusi i di S , $M(i)$ adalah set gerakan yang dapat diaplikasikan pada i untuk memperoleh solusi baru j (dengan notasi: $j = i \oplus m$). Dengan demikian $N(i) = \{j / \exists m \in M(i) \text{ dengan } j = i \oplus m\}$. Untuk setiap m terdapat m^{-1} yang dapat mengembalikan j menjadi i . Jadi, daftar tabu T juga dapat hanya merekam gerakan m yang telah dilakukan, lebih sederhana dibandingkan dengan merekam solusinya.

Beberapa daftar tabu juga dapat digunakan secara bersamaan (T_r). Dengan daftar tersebut, beberapa elemen t_r (berupa i atau m) diberikan status tabu untuk menandakan bahwa elemen tersebut tidak boleh dilibatkan selama beberapa iterasi. Formulasi dari beberapa persyaratan tabu adalah $t_r(i, m) \in T_r$ ($r = 1, \dots, t$). Gerakan m yang diaplikasikan pada solusi i akan menjadi gerakan yang tabu apabila semua persyaratan yang diformulasikan tersebut terpenuhi.

Daftar tabu yang disederhanakan (hanya diisi oleh m) memiliki kelemahan, yaitu status tabu dapat diberikan pada solusi yang belum ditemui. Jika hal tersebut terjadi, padahal solusi tabu tersebut cukup baik dibandingkan dengan solusi-solusi yang telah ditemui, maka status ini dapat dihilangkan oleh persyaratan tingkat aspirasi (*aspiration level conditions*). Gerakan m akan diterima walaupun berstatus tabu apabila memiliki tingkat aspirasi $a(i, m)$ yang lebih baik dari nilai ambang $A(i, m)$. $A(i, m)$ adalah sejumlah nilai yang diharapkan untuk fungsi $a(i, m)$. Formulasi dari persyaratan aspirasi $a_r(i, m) \in A_r(i, m)$ ($r = 1, \dots, a$). Jika setidaknya salah satu persyaratan tersebut dipenuhi oleh gerakan tabu m pada i , maka m akan diterima (apapun statusnya).

Karakteristik lain dari *tabu search* adalah intensifikasi dan diversifikasi. Hal ini berkaitan dengan fakta bahwa dalam proses eksplorasi *tabu search*, fungsi tujuan f dapat diganti oleh fungsi lain yang dimodifikasi f^*

Dalam *tabu search*, memori digunakan secara unik untuk memandu prosedur pencarian. Memori tersebut juga dapat diperdalam untuk memperkuat pencarian di daerah tertentu dari S karena kemungkinan diperolehnya solusi yang dapat diterima. Intensifikasi tersebut dilaksanakan dengan memberikan prioritas tinggi kepada solusi dengan ciri yang mirip dengan solusi terkini. Hal ini dapat dilakukan dengan menambahkan variabel pada fungsi tujuan yang akan menjauhkan solusi dari solusi sebelumnya.

Setelah beberapa iterasi, diversifikasi akan menyebarkan eksplorasi yang lebih dalam ke daerah-daerah lain dari S . Diversifikasi, seperti halnya intensifikasi, dapat dipaksakan dengan menambahkan variabel pada fungsi tujuan. Baik variabel intensifikasi maupun diversifikasi memiliki bobot yang memodifikasi dalam proses sehingga fase intensifikasi dan diversifikasi muncul bergantian selama pencarian. Fungsi tujuan yang dimodifikasi inilah yang dinotasikan dengan f^* dimana $f^* = f + \text{intensifikasi} + \text{diversifikasi}$.

Beberapa karakteristik *tabu search* yang telah dijelaskan diatas dapat diformulasikan pada prosedur *tabu search* yang lebih lengkap, yaitu sebagai berikut:

- Langkah 1.** Pilih solusi awal i dari S . Ubah $i^* = i$ dan $k = 0$.
- Langkah 2.** Ubah $k = k + 1$ dan buat *subset* solusi V^* dari $N(i, k)$ sehingga salah satu persyaratan tabu $t_r(i, m) \in T_r$ dilanggar ($r = 1, \dots, t$), atau setidaknya salah satu persyaratan aspirasi $a_r(i, m) \in A_r(i, m)$ berlaku ($r = 1, \dots, a$).
- Langkah 3.** Pilih solusi terbaik $j = i \oplus m$ dari V^* (dengan memperhatikan f atau f^* yang digunakan) dan ubah $i = j$.
- Langkah 4.** Jika $f(i) < f(i^*)$ maka ubah $i^* = i$.
- Langkah 5.** Perbaharui persyaratan tabu dan aspirasi.
- Langkah 6.** Jika persyaratan penghentian ditemui maka berhenti. Lainnya maka kembali ke **Langkah 2**.

2.4 *Tabu Search* Pada VRP

Tabu search merupakan salah satu metode penyelesaian VRP yang tergabung dalam suatu kelas yang disebut dengan *metaheuristic* (Osman, 1995). Metode *tabu search* ini terbukti sukses dalam memecahkan permasalahan kombinatorial terkait dengan permasalahan optimasi seperti VRP. Dasar dari *tabu search metaheuristic* ini adalah dengan menggunakan strategi pengawalan yang agresif untuk memotong prosedur pencarian lokal untuk membawa keluar eksplorasi dari himpunan solusi dalam rangka menghindari keterjebakan dalam *local optimal* (Cordeau dan Laporte, 2002). Ketika *local optimal* ditemui, strategi agresif bergerak ke solusi terbaik di setiap tetangga walaupun mungkin akan mengakibatkan penurunan dalam nilai tujuan.

Untuk menghindari pencarian ke tempat yang baru saja atau sudah pernah diperoleh, *tabu search* menggunakan struktur memori untuk menyimpan atribut dari solusi yang diterima yang baru saja ditemui dalam suatu daftar yang disebut dengan *tabu list*. Semua atribut solusi yang berada dalam daftar tabu memiliki status tabu (*tabu-active*), dan solusi yang memiliki elemen status tabu tersebut dikatakan sebagai solusi tabu (solusi yang dihindari). Durasi dimana sebuah atribut tetap berstatus tabu disebut sebagai periode tabu atau *tabu tenure* (dinotasikan dengan *tt*). Apabila sebuah atribut telah berada dalam daftar tabu selama *tt* yang ditentukan, maka status tabu untuk atribut tersebut hilang. Algoritma *tabu search* akan terus melakukan pencarian sampai sejumlah iterasi tertentu atau syarat perhentian lainnya sudah terpenuhi.

Algoritma *tabu search* membutuhkan solusi awal untuk diperbaiki. Solusi awal ini dapat diperoleh dari kelas yang lebih rendah, seperti pendekatan eksak ataupun heuristik klasik. Beberapa contoh pendekatan yang dapat digunakan adalah seperti *nearest neighbourhood*, *insertion heuristic*, atau *saving algorithm*.

2.4.1 Komponen *Tabu Search*

Algoritma *tabu search* memiliki beberapa komponen-komponen yang membedakannya dengan algoritma lain, yaitu sebagai berikut:

1. *Tabu list* yaitu memori jangka pendek yang digunakan untuk menyimpan atribut solusi yang telah ditemukan.

2. *Tabu restriction*, yaitu kriteria untuk menentukan status tabu
3. *Aspiration criteria*, yaitu kriteria untuk mengesampingkan atau menghilangkan status tabu agar suatu pergerakan diizinkan karena dapat menghasilkan solusi yang lebih baik.
4. *Stopping rule*, yaitu aturan atau kriteria untuk menghentikan seluruh proses *tabu search*.
5. Skema *tabu tenure*, yaitu penentuan durasi status tabu. Terdapat beberapa skema yang dapat digunakan untuk melakukan kontrol periode tabu, yaitu:

- *Fixed Tabu* (F-Tabu)

Skema ini merupakan metode yang pertama kali dikenal dan digunakan. Periode tabu merupakan nilai yang tetap ($tt = \text{jumlah iterasi tertentu}$) selama proses *tabu search*.

- *Robust Tabu* (Rb-Tabu)

Dalam skema ini, nilai periode tabu tt diambil secara acak pada kisaran tertentu. Selama proses pencarian, nilai tt berubah setelah x iterasi. Nilai x ini juga diambil secara acak.

- *Periodic Tabu* (P-Tabu)

Nilai tt yang digunakan dalam skema ini berubah secara periodik dari nilai yang terkecil, sedang, hingga besar. Perubahan nilai tt terjadi setelah x iterasi.

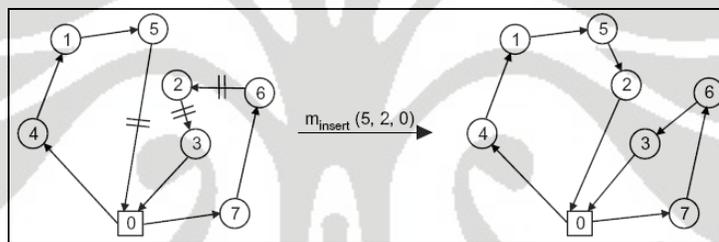
- *Reversed Deterministic Tabu* (Rd-Tabu)

Skema ini merupakan metode baru dengan mekanisme mengubah dan mengembalikan (*reverse*) nilai tt selama proses pencarian. Skema diawali dengan penggunaan beberapa nilai tt yang telah ditentukan ($tt = n/p$, dengan n adalah jumlah konsumen dan p diambil dari kisaran 1-9). Skema ini secara dinamis mengubah nilai nilai tt selama proses pencarian. Nilai tt tersebut dihitung setelah x iterasi, dimana $x = x\text{-max}/n \times p$. Nilai p pada awal pencarian adalah 9, lalu diturunkan satu setelah x iterasi. Setelah mencapai 1, nilai p diulang kembali dari 9 dan seterusnya hingga pencarian dihentikan.

2.4.2 Mekanisme Pembentukan Solusi Tetangga

Setelah semua komponen-komponen *tabu search* diatas ditentukan, maka diperlukan adanya mekanisme pembentukan solusi tetangga. Untuk *tabu search*, terdapat metode untuk memperoleh solusi tetangga. Untuk implementasinya, dapat diadopsi mekanisme himpunan struktur solusi tetangga yang digunakan Homberger dan Gehring (2004), dimana $N = \{N_{insert}, N_{2-opt}, N_{change}\}$ dengan memilih secara acak salah satu struktur solusi tetangga tersebut pada setiap iterasi.

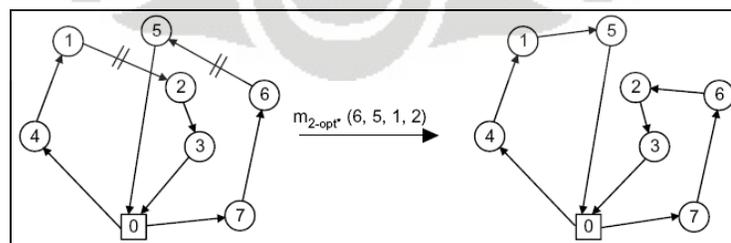
Struktur solusi tetangga N_{insert} merupakan mekanisme pergerakan untuk memindahkan satu titik pelanggan dari satu rute ke rute lain dengan notasi $m_{insert}(i, k, i+)$ yang artinya pemindahan titik k untuk diletakkan antara titik i dan titik $i+$. Metode ini diperkenalkan oleh Saverlsberg (1992). Ilustrasinya dapat dilihat pada gambar berikut.



Gambar 2.3 Ilustrasi Pergerakan *Insertion*

(Sumber: Homberger & Gehring, 2004)

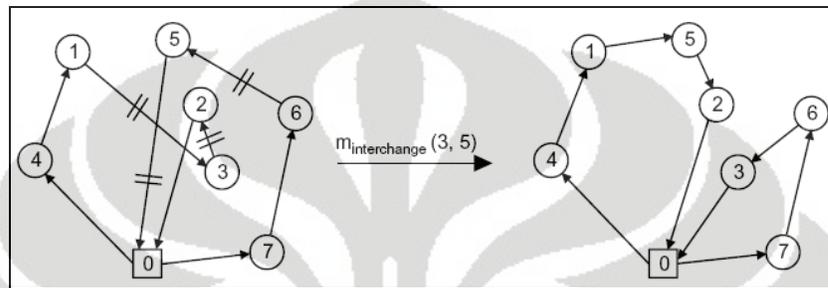
Struktur solusi tetangga N_{2-opt} merupakan mekanisme pergerakan untuk menukarkan dua titik pelanggan dari satu rute ke rute lain dengan notasi $m_{2-opt}(k-, k, i-, i)$ yang artinya menukarkan urutan $k-$, k dengan titik urutan $i-$, i . Metode ini berdasarkan 2-Opt yang diperkenalkan Lin (1965) dan dikembangkan oleh Potvin & Rousseau (1995). Ilustrasinya dapat dilihat pada gambar berikut.



Gambar 2.4 Ilustrasi Pergerakan 2-Opt

(Sumber: Homberger & Gehring, 2004)

Struktur solusi tetangga N_{change} merupakan mekanisme pergerakan untuk menukarkan satu titik pelanggan dari satu rute ke rute lain dengan notasi $m_{change}(k, i)$ yang artinya menukarkan k dengan i . Metode ini berdasarkan 1 -interchange yang diperkenalkan oleh Osman (1993). Ilustrasinya dapat dilihat pada gambar berikut.



Gambar 2.5 Ilustrasi Pergerakan *Interchange*

(Sumber: Homberger & Gehring, 2004)

2.4.3 Prosedur *Tabu Search* for VRP

Algoritma *tabu search* pada VRP bertujuan untuk meminimalkan total jarak tempuh yang dilalui oleh setiap kendaraan. Beberapa pergerakan dipilih secara acak dari set pergerakan yang dapat dilakukan dalam satu iterasi dan diaplikasikan kepada solusi saat ini. Dari solusi tetangga yang dihasilkan, dipilih solusi memungkinkan yang terbaik.

Tahap awal untuk menyelesaikan permasalahan VRP dengan *tabu search* adalah menentukan solusi awal. Solusi awal ini diperoleh dari hasil pengolahan data dengan menggunakan metode heuristik klasik seperti *nearest neighbourhood*. Setelah ditentukan solusi awal, maka langkah selanjutnya adalah mempersiapkan daftar tabu (*Tlist*) yang merupakan daftar kosong. Kemudian solusi saat ini (x_{it}) diperoleh dari solusi awal yang telah ditentukan tadi ($x_{it} = x_{best}$).

Kemudian iterasi dilakukan untuk memperoleh solusi tetangga. Pada tiap iterasi, dibentuk sebanyak $c.n$ (c merupakan bilangan konstan lebih besar dari nol dan n merupakan jumlah *customer*) solusi tetangga yang diperoleh dari mekanisme salah satu struktur N yang dipilih secara acak, terhadap x_{it} , untuk kemudian dievaluasi. Kriteria evaluasi adalah jumlah kendaraan dan total jarak perjalanan. Dari set solusi tetangga $c.n$, dipilih satu solusi terbaik. Kemudian, jika

Universitas Indonesia

solusi tersebut tidak termasuk dalam daftar tabu dan juga memenuhi kriteria aspirasi, maka solusi tersebut digunakan sebagai solusi terbaik (x_{sel}). Perbaharui solusi terbaik saat ini ($x_{sel} = x_{it}$).

Solusi yang terpilih tidak termasuk tabu apabila memenuhi persyaratan bahwa tidak satu pun lengkung antarkonsumen atau lengkung antara konsumen dengan depot dari suatu jadwal rute berstatus tabu dalam daftar tabu. Namun, berdasarkan kriteria aspirasi, status tabu dari satu atau lebih lengkung akan diabaikan jika solusi tetangga tersebut menghasilkan solusi baru yang lebih baik.

Pada akhir tiap iterasi, daftar tabu ($Tlist$) dan solusi terbaik saat ini (x_{best}) diperbaharui ($x_{best} = x_{it}$). Fokus dari pembaruan daftar tabu tersebut adalah eliminasi lengkung-lengkung (lengkung antarkonsumen atau lengkung antara konsumen dengan depot) ketika terjadi transisi dari x_{it} yang lama ke solusi tetangga terbaik yang baru.

Proses pencarian atau iterasi dihentikan apabila sejumlah maksimum iterasi ($maxit$) telah dilakukan dan/atau solusi terbaik x_{best} tidak dapat diperbaiki lagi selama sejumlah $stopit$ iterasi ($stopit > 0$) terakhir.

Dalam melakukan pergerakan (N) pada tiap iterasi, perlu dilakukan pengujian apakah solusi tetangga yang dihasilkan memungkinkan (*feasible*) atau tidak. Uji fisibilitas ini meliputi dua aspek, yaitu:

- Aspek batasan waktu (*time windows*) dari tiap konsumen. Perlu diperhatikan apakah solusi yang dihasilkan masih memenuhi persyaratan waktu pelayanan masing-masing konsumen, dan apakah konsumen berikutnya masih sempat dilayani.
- Aspek kapasitas kendaraan. Pastikan bahwa pengiriman yang dilakukan berdasarkan solusi baru tidak menyebabkan kelebihan kapasitas dari tiap kendaraan.

BAB 3

PENGUMPULAN DATA

Pada bab ini dijelaskan mengenai pengumpulan data yang dibutuhkan untuk melaksanakan penelitian. Pengumpulan data dilakukan dengan cara observasi, wawancara, dan pengumpulan dokumen di suatu perusahaan agribisnis. Pada bab ini dijelaskan mengenai data-data yang berkenaan dengan aktivitas distribusi pengiriman barang.

3.1 Profil Perusahaan

PT. Saung Mirwan didirikan pada tahun 1983 terletak di Desa Sukamanah, Kec. Megamendung - Bogor, dengan ketinggian 670 m dpl. Pada awalnya, T. Hadinata selaku pemilik perusahaan memulai usahanya dengan menanam melon di atas lahan terbuka. Tahun 1985 mulai dikembangkan usahanya dengan menanam bawang putih seluas 7 Ha di daerah Cipanas, Kab. Cianjur dan memperkerjakan karyawan sebanyak 100 orang.

Pada awalnya T. Hadinata mengembangkan usahanya di sekitar desa Sukamanah dengan mencoba usaha tanaman di dalam *green house*, menggunakan sistem tetes irigasi. Hasil percobaan awal yang menunjukkan hasil sangat memuaskan menjadikan T.Hadinata untuk memperbesar usaha ini dengan jenis tanaman melon, paprika, tomat, kyuuri dan shisito. Hingga tahun 1991 luas areal *green house* telah mencapai 1.5 Ha.

Saat ini pelanggan PT Saung Mirwan tersebar di daerah Jabodetabek, Jawa Tengah dan Yogyakarta. Beberapa perusahaan yang menjadi pelanggan perusahaan ini antara lain Mcdonald, Burger King, Carrefour, Ranch Market, Hypermart, dan Hari-hari.

3.2 Data yang Dibutuhkan

Untuk dapat menyelesaikan permasalahan VRP di perusahaan agribisnis ini, maka dibutuhkan beberapa data yang akan digunakan untuk mendapatkan solusi rute transportasi yang optimal dan untuk melakukan analisis perbandingan. Data yang dibutuhkan antara lain adalah data mengenai depot, armada

pengiriman, biaya pengiriman, konsumen, waktu dan kecepatan, permintaan, jarak, dan rute pengiriman.

3.2.1 Depot

Depot merupakan titik awal dimana pengiriman produk dimulai. Pada kasus ini, depot terletak di perusahaan agribisnis itu sendiri, yaitu di daerah Ciawi, Bogor. Komoditi sayuran yang ada di depot ini berasal dari lahan perkebunan yang ada di perusahaan itu sendiri dan juga dari mitra tani yang menjalin kerjasama dengan pihak perusahaan. Biasanya pengambilan komoditi sayuran dari mitra tani diambil pada malam hari dan sampai di depot pada pukul 02.00.

Selanjutnya sayuran yang ada di depot diproses untuk kemudian dikemas dan diangkut ke dalam kendaraan (armada pengirim). Proses pengangkutan sayuran (*loading*) tersebut dilakukan dari pukul 02.00 – 04.00. Kemudian, pada pukul 04.00, seluruh armada kendaraan mulai berangkat dari depot untuk melakukan pengiriman sayuran ke para konsumen.

3.2.2 Armada Pengiriman

Dalam melakukan pengiriman produk sayuran ke konsumen, perusahaan agribisnis ini menggunakan jenis kendaraan mobil *box* yang dilengkapi dengan pendingin. Jumlah kendaraan yang tersedia adalah 6 buah. Keenam kendaraan ini memiliki kapasitas angkut yang sama. Berikut ini adalah data-data mengenai armada angkut yang digunakan:

- Kapasitas muatan : 75 krat
- Bahan Bakar : solar
- *Ratio* bahan bakar : 1 : 7
- Harga Solar : Rp. 4.500,-/ liter

Kapasitas muatan kendaraan dinyatakan dalam satuan krat karena pengiriman produk sayuran dilakukan dengan menggunakan wadah berupa krat. Krat-krat tersebut memiliki ukuran dimensi yang sama. Kapasitas kendaraan tidak dapat dinyatakan dalam satuan berat ataupun satuan volume dikarenakan komoditas yang diangkut (sayuran) memiliki dimensi, volume, dan berat yang berbeda untuk setiap jenisnya.

Kemudian *ratio* bahan bakar kendaraan ini adalah 1 : 7, dimana kebutuhan bahan bakar per kilometer relatif tinggi. Hal ini dikarenakan kendaraan yang dipakai memiliki kontainer yang dilengkapi dengan pendingin sehingga kebutuhan energi atau bahan bakar menjadi lebih besar.

3.2.3 Biaya Pengiriman

Biaya pengiriman diperoleh dengan melakukan perhitungan terhadap biaya bahan bakar, biaya pemeliharaan, biaya ban, dan biaya supir. Besarnya diketahui melalui wawancara dengan pihak supir perusahaan. Berikut ini adalah rincian biaya pengiriman per kilometer per kendaraan.

- Biaya bahan bakar

Kebutuhan bahan bakar	=	0.15 liter/km
Harga bahan bakar	=	Rp. 4.500,-/liter
Biaya bahan bakar	=	Rp. 4.500,- × 0.15 = Rp. 675,-/km
- Biaya pemeliharaan

Perusahaan melakukan *service* berkala setiap 4 bulan. Biaya satu kali *service* adalah Rp. 7.000.000. Biaya ini sudah termasuk ganti oli, filter oli, minyak rem, dan *service* alat pendingin. Rata-rata selama 1 bulan, satu kendaraan akan menempuh jarak 4.500 km, maka untuk 4 bulan akan menempuh jarak 18.000. Jadi, biaya pemeliharaan/km = $7.000.000/18.000 = 388.88$ /km
- Biaya Ban

Satu ban kendaraan dapat digunakan untuk 20.000 km. Harga 1 ban untuk ukuran kendaraan ini adalah Rp. 870.000. Jadi, biaya ban/km untuk 1 kendaraan = $4 * (Rp. 880.000/20.000) = Rp. 176/km$
- Biaya supir dan pendamping = Rp. 75.000,- per pengantaran

3.2.4 Konsumen

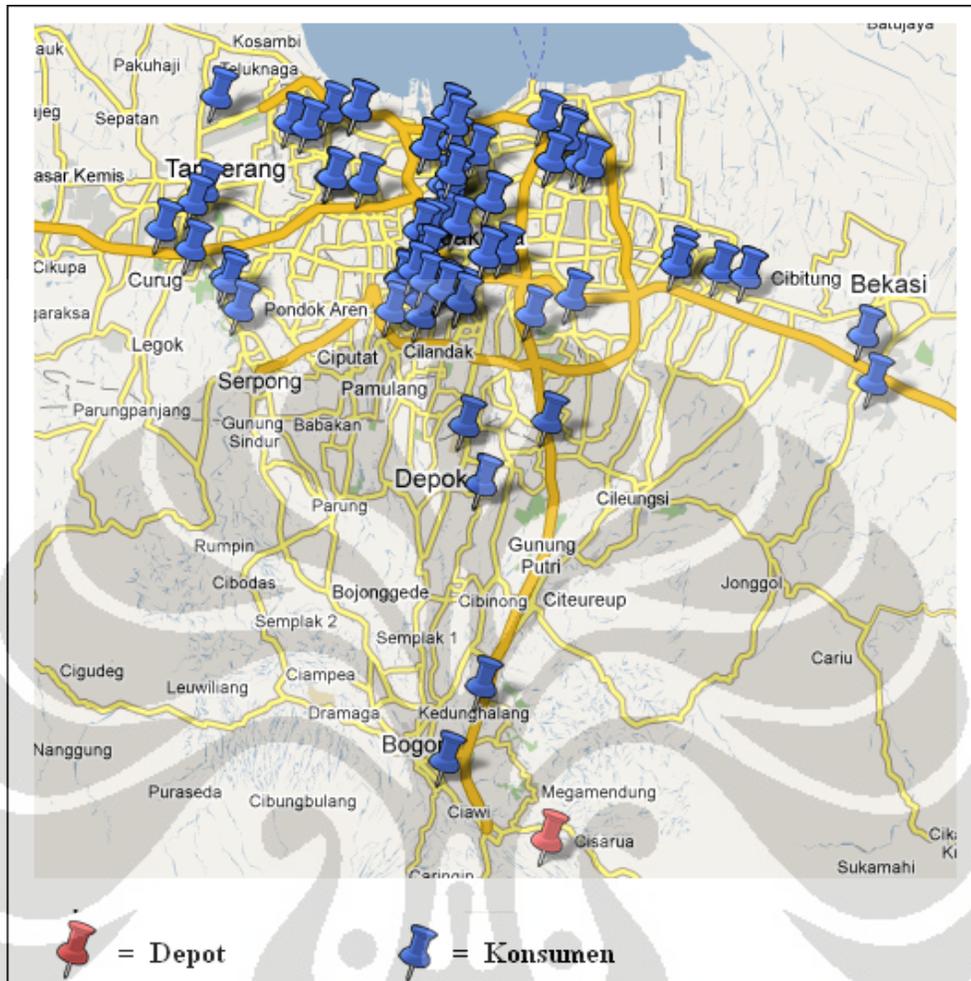
Jumlah konsumen yang dilayani adalah sebanyak 78 konsumen. Konsumen tersebut tersebar di daerah Jabodetabek. Setiap konsumen memiliki kode nama tersendiri yang digunakan untuk memudahkan pengidentifikasian untuk beberapa kepentingan. Berikut ini adalah nama konsumen (titik kirim) beserta dengan kode namanya.

Tabel 3.1 Daftar Konsumen dan Kodenya

No.	Nama Customer	Kode	No.	Nama Customer	Kode
1	Market City	CMS	40	Mc. Donald Makasar	MDMK
2	Carrefour Ambassador	CRA	41	Mc. Donald Palembang	MDR
3	Carrefour Bumi Serpong D	CRB	42	Mc. Donald Samarinda	MDS
4	Carrefour Cikokol	CRC	43	Matahari Cibubur	MPB
5	Carrefour Cikokol	CRE	44	Matahari Cilandak	MPC
6	Carrefour Mt Haryono	CRH	45	Matahari Ekalokasari	MPE
7	Carrefour Lebak Bulus	CRL	46	Matahari Serpong	MPG
8	Carrefour Tm Plm	CRN	47	Matahari Metropolis	MPH
9	Carrefour Puri	CRP	48	Matahari Dc	MPI
10	Carrefour Squer	CRQ	49	Matahari Karawaci	MPK
11	Diamond Artha Gading	DMA	50	Matahari Klp Gading	MPL
12	Diamond Fatmawati	DMF	51	Matahari Hpy Pejaten	MPM
13	Domino Pizza K1 Gading	DOG	52	Matahari Daan Mogot	MPN
14	Domino Pizza	DOK	53	Matahari Depok	MPO
15	Domino Pizza PI	DOP	54	Matahari Hpy Glodog	MPQ
16	Farmers market Kpl Gdg	FMG	55	Matahari Hyper Cikarang	MPR
17	Farmers market Serpong	FMS	56	Matahari Fatmwt Karawaci	MPS
18	Grand Lucky	GNL	57	Matahari Hypermart- Puri	MPU
19	PT. Burger King -Cilandak	GRC	58	Matahari Belanova	MPV
20	PT. Burger King -Grand	GRG	59	Matahari Hyper Jacc	MPX
21	PT. Burger King - Senayan	GRK	60	Puspa Cattering	PCS
22	PT. Burger King- Klp Gading	GRL	61	Purantara	PRN
23	PT. Burger King -Thamrin	GRM	62	Papaya Fresh G	PY
24	PT. Burger King -PI	GRP	63	Papaya Bali	PYB
25	PT. Burger King- Semanggi	GRS	64	Papaya Surabaya	PYS
26	Hari2 Bekasi	HHB	65	Sari Pizza City Walk	PZC
27	Hari2 Cyber Park	HHC	66	Sari Pizza Grand	PZG
28	Hari2 Dhi	HHD	67	Sari Pizza -Kemang	PZK
29	Hari2 Fatmawati	HHF	68	Sari Pizza Kemang	PZM
30	Hari2 Kalideres	HHK	69	Sari Pizza -Marzano	PZO
31	Hari2 Lokasari	HHL	70	Sari Pizza Senayan Citty	PZS
32	Hari2 Roximas	HHR	71	Ranch Market Drmawangsa	RCD
33	PT. Jaddi Pastrisindo	JPG	72	Ranch Market Pejaten	RCJ
34	Klenger Burger	KB	73	Ranch Market 99	RCM
35	Koko Sp Market	KS	74	Ranch Market PI	RCP
36	Mos Burger Plaza	MBP	75	PT. San Miguel	SMP
37	Mc. Donald Jakarta	MD	76	PT. Lion Superindo	SPI
38	Mc. Donald Batam	MDB	77	Sari Kuring	SRK
39	Mc. Donald Batam Formosa	MDBC	78	PT.Trias Tanjung R	TTR

(Sumber: PT. Saung Mirwan)

Berikut ini adalah peta yang memberikan ilustrasi mengenai letak lokasi depot dan persebaran lokasi konsumen.



Gambar 3.1 Lokasi Depot dan Konsumen

Jika dilihat pada gambar diatas, jumlah titik konsumen kurang dari 78 titik. Hal tersebut dikarenakan terdapat beberapa konsumen yang memiliki lokasi yang sama, contohnya adalah konsumen-konsumen dengan kode MDB, MDBC, MDMK, MDR, dan MDSD. Kelima konsumen tersebut memiliki lokasi titik kirim yang sama, yaitu di bandara Soekarno-Hatta. Untuk selanjutnya kelima konsumen tersebut dilambangkan dengan satu kode nama yaitu MDLJ (Mc Donald Luar Jawa). Kemudian konsumen dengan kode nama PY, PYB dan PYS juga memiliki titik lokasi kirim yang sama dan ketiga konsumen tersebut selanjutnya akan dilambangkan dengan kode PY.

3.2.5 Waktu dan Kecepatan

Data waktu yang perlu diketahui adalah *time windows*, dan *service time*. *Time windows* merupakan rentang waktu dimana konsumen dapat dilayani atau masih dapat menerima kiriman produk. Melalui wawancara dengan pihak perusahaan agribisnis ini, diketahui bahwasanya setiap konsumen dapat menerima kiriman produk mulai dari pukul 05.00. Kemudian, konsumen menginginkan produk sudah harus sampai atau terkirim maksimal pukul 09.00, meskipun konsumen-konsumen tersebut masih buka hingga sore atau malam. Dari hasil wawancara tersebut, maka ditentukan *time windows* untuk setiap konsumen adalah dari pukul 05.00 – 09.00.

Data selanjutnya adalah data *service time* atau waktu pelayanan di setiap lokasi konsumen. *Service time* ini dapat dibagi menjadi dua bagian, yaitu waktu penurunan barang (*unloading*) dan waktu untuk pengurusan administrasi. Data ini juga diperoleh dari hasil wawancara pihak perusahaan agribisnis yang mengatakan bahwa waktu rata-rata untuk melakukan penurunan barang pesanan adalah sekitar 5 menit dan waktu untuk mengurus masalah administrasi sekitar 5 menit, sehingga total *service time* di setiap lokasi konsumen adalah 10 menit.

Data kecepatan adalah kecepatan rata-rata kendaraan dalam melakukan pengantaran barang. Dari hasil wawancara dengan supir dengan mempertimbangkan karakteristik dan kondisi jalan, disimpulkan bahwa kecepatan rata-rata setiap kendaraan adalah 45 km/jam.

3.2.6 Permintaan

Seperti yang telah dijelaskan sebelumnya, jumlah permintaan konsumen yang harus dipenuhi berbeda-beda setiap harinya. Konsumen yang melakukan pemesanan juga berbeda-beda untuk setiap harinya. Kemudian, permintaan yang dilakukan konsumen akan berbeda-beda untuk setiap konsumen, baik jumlah permintaannya maupun jumlah jenis sayurannya.

Satuan yang biasa digunakan untuk jumlah permintaan ini adalah satuan berat (kg). Untuk itu, perlu dilakukan pengkonversian satuan permintaan ini ke dalam satuan krat. Jumlah kilogram sayuran yang dapat dimuat ke dalam sebuah

krat berbeda-beda untuk setiap jenis sayuran tersebut. Berikut ini adalah data kapasitas krat (dalam kg) untuk setiap jenis sayuran.

Tabel 3.2 Kapasitas Krat untuk Setiap Jenis Sayuran

No.	Nama Produk	ID Produk	kg/krat	No.	Nama Produk	ID Produk	kg/krat
1	ANGGUR HIJAU	AGH	15	35	LETUCE HEAD	LTD	5
2	ANGGUR MERAH	AGM	15	36	LETTUCE ROMANCE	LTM	6
3	ASPARAGUS	ASG	10	37	NASUBHI	NAS	10
4	BAWANG BOMBAY	BBY	15	38	ENDIVE	NDV	5
5	BUNCIS MINI	BCM	10	39	OKRA	OKA	8
6	BUNCIS TAIWAN	BCT	15	40	PAKCOY BABY	PCB	5
7	BIT	BIT	10	41	PAKCOY HIJAU	PCH	5
8	BROCOLLY	BRC	6	42	PAKCOY PUTIH	PCP	6
9	BROCOLLY BS	BRCS	6	43	PEAR XIANGLI	PEX	5
10	BASIL URA	BSL	3	44	APEL GRANNY SMI	PLS	5
11	BAWANG PUTIH	BWP	12	45	PAPRIKA HIJAU	PPH	8
12	BAWANG MERAH	BWR	12	46	PAPRIKA KUNING	PPK	8
13	CABE MERAH	CBR	10	47	PAPRIKA ORANGE	PPO	8
14	CAYSIN	CYS	5	48	PAPRIKA MERAH	PPR	8
15	DAUN BAWANG LEE	DBL	7	49	PETERSELY	PRL	5
16	DAUN BAWANG	DBW	7	50	PISANG AMBON	PSA	5
17	DAIKON LARGE	DKN	15	51	RADICHIO	RDC	5
18	EDAMAME	EDA	15	52	SELEDRI STIK	SDT	5
19	EDAMAME FROZEN	EDF	15	53	SALAD GARDEN	SLG	5
20	HORINSO	HRN	5	54	SELEDRI	SLI	7
21	JAGUNG ACAR	JGC	10	55	SELADA MERAH	SLR	7
22	JAMUR CAMPIGNON	JMC	10	56	SELADA KERITING	SLT	5
23	JERUK LEMON	JRL	10	57	SHISITO	SST	7
24	JERUK SUNKIST	JRS	10	58	SAWI PUTIH	SWP	10
25	KOL BULAT PUTIH	KBP	8	59	TOMAT CHERRY	TMC	15
26	KEMBANG KOL	KKL	6	60	TOMAT RIANTO	TMT	15
27	KAILAN BABY	KLB	5	61	TOMAT TW	TMW	15
28	KAILAN	KLN	5	62	TIMUN JEPANG	TNJ	15
29	KOL MERAH	KLR	8	63	TIMUN MINI	TNM	15
30	KOL PUTIH BABY	KPB	7	64	TIMUN SAYUR	TNS	15
31	SNOW PEA	KPI	8	65	TERONG SAYUR	TRS	15
32	SUGAR SNAP	KPW	8	66	WORTEL	WRL	15
33	KOL MERAH BABY	KRB	7	67	ZUKINI BABY	ZKB	15
34	KENTANG BESAR	KTS	5	68	ZUKINI HIJAU	ZKN	15

(Sumber: PT. Saung Mirwan)

Setelah dilakukan konversi satuan kg ke krat untuk setiap jenis sayuran, maka dapat diperoleh jumlah permintaan harian setiap konsumen dalam satuan krat. Berikut ini adalah data jumlah permintaan konsumen atau pengiriman kepada konsumen dari tanggal 5 Januari 2009 sampai dengan 11 Januari 2009.

Tabel 3.3 Pengiriman Pada Konsumen Selama 1 Minggu

Senin, 5/1/2009																																								
Customer	CRL	DOP	GRG	GRK	GRL	GRM	GRP	GRS	JPG	MBP	MD	PRN	PZK																											
Jumlah Krat	10	5	6	2	12	12	1	12	2	13	7	69	17	5																										
Selasa, 6/1/2009																																								
Customer	CRH	CRL	DMA	DMF	DOG	FMG	FMS	GRC	GRG	GRM	GRS	JPG	MD	MDLJ	MPB	MPC	MPE	MPG	MPH	MPK	MPL	MPN	MPR	MPS	PZC	PZK	PZO	PZS	SMP											
Jumlah Krat	18	14	6	8	6	16	4	4	10	4	5	16	30	18	16	9	12	8	12	17	5	16	30	5	6	6	5	8	21											
Rabu, 7/1/2009																																								
Customer	CRB	CRH	CRL	DMA	DMF	FMG	FMS	GNL	GRC	GRG	GRL	GRM	GRS	JPG	MBP	MD	MPE	MPK	MPL	MPO	MPQ	MPR	MPS	PRN	PY(S,B)	PZK	PZM	RCJ	SPI											
Jumlah Krat	4	5	28	7	7	22	12	7	1	2	3	3	1	31	4	28	3	23	5	7	15	16	7	12	9	4	1	3	65											
Kamis, 8/1/2009																																								
Customer	CRA	CRB	CRH	CRL	CRP	DMA	DMF	DOK	FMG	FMS	GRC	GRG	GRL	GRM	GRP	GRS	JPG	MBP	MD	MPC	MPE	MPK	MPL	MPS	PRN	PZK	RCM													
Jumlah Krat	11	7	13	21	11	7	9	1	8	14	5	2	4	4	4	2	19	12	54	11	18	21	10	8	2	2	4													
Jumat, 9/1/2009																																								
Customer	CRB	CRH	CRL	CRN	CRQ	DMA	DMF	DOP	FMG	GRC	GRG	GRK	GRM	GRP	GRS	HHB	HHC	HHD	HHF	HHK	HHL	HHR	JPG	KS	MBP	MD	MDLJ	MPC	MPE	MPH	MPI	MPK	MPL	MPN	MPO	MPS	PZM	SRK		
Jumlah Krat	3	15	25	12	2	10	8	8	16	4	12	6	7	12	9	8	7	7	11	9	22	13	33	5	4	16	13	10	12	4	23	18	6	6	6	4	1	4		
Sabtu, 10/1/2009																																								
Customer	CRA	CRB	CRE	CRH	CRL	DMA	DMF	DOK	FMG	FMS	GNL	GRC	GRG	GRL	GRM	GRP	GRS	JPG	MBP	MD	MPB	MPC	MPE	MPK	MPL	PZC	PZG	RCD	RCJ	SPI										
Jumlah Krat	13	11	17	12	36	11	13	8	19	12	8	4	8	5	10	8	5	31	5	8	12	8	6	14	7	8	8	7	3	24										
Minggu, 11/1/2009																																								
Customer	CRE	CRL	DMA	DMF	FMG	FMS	GNL	GRC	GRL	GRM	GRS	MBP	MD	MPC	MPE	MPK	MPL	MPN	MPS	MPV	SPI																			
Jumlah Krat	15	14	11	11	21	10	2	4	12	5	3	6	4	9	16	12	18	14	12	10	53																			

(Sumber: PT. Saung Mirwan, telah diolah kembali)

3.2.7 Jarak

Data jarak yang dikumpulkan adalah jarak antara depot dengan titik kirim (konsumen) dan jarak antar titik kirim. Pengambilan data jarak ini dilakukan dengan menggunakan bantuan peta digital. Peta digital yang digunakan merupakan aplikasi yang dikeluarkan oleh *Google*, yaitu *Google Maps* (www.maps.google.com). Aplikasi peta digital tersebut memiliki *tool* atau alat bantu yang bernama *distance measurement tool*. *Tool* ini dapat digunakan untuk mengukur jarak antara dua titik yang berada di peta.

Pengukuran jarak antara dua titik dilakukan dengan mengikuti alur jalan yang ada di peta sehingga data jarak yang diperoleh dapat mendekati jarak aktual yang ditempuh oleh kendaraan. Pemilihan jalan yang menghubungkan dua titik tertentu dilakukan dengan pertimbangan jarak terdekat dan juga kondisi atau karakteristik jalan (tingkat kemacetan). Kemudian, diasumsikan jarak tempuh dari titik A ke titik B sama dengan jarak tempuh dari titik B ke titik A, sehingga matriks jarak yang dihasilkan akan simetris.

Data jarak dari depot menuju titik kirim dan jarak antara titik kirim dituangkan dalam bentuk matriks jarak. Matriks jarak ini dapat dilihat pada bagian lampiran.

3.2.8 Rute Pengiriman

Data selanjutnya adalah data rute pengiriman yang dilakukan oleh perusahaan. Data ini diperoleh dari dokumentasi pihak perusahaan mengenai perencanaan pengiriman barang harian. Dalam melakukan perencanaan pengiriman, pihak perusahaan membagi transportasi pengiriman produk menjadi beberapa rute. Pembagian rute ini dilakukan berdasarkan kedekatan lokasi konsumen dan juga pertimbangan mengenai kapasitas muatan kendaraan.

Pihak perusahaan tidak melakukan perencanaan untuk menentukan titik kirim mana yang dikunjungi terlebih dahulu di setiap rutenya. Penentuan urutan titik kirim yang akan dikunjungi diserahkan sepenuhnya pada supir masing-masing kendaraan. Oleh karena itu, data urutan rute diperoleh dari hasil wawancara dengan para supir kendaraan.

Berikut ini adalah data rute pengiriman produk harian beserta dengan urutan titik kirim yang dituju untuk setiap rute.

Tabel 3.4 Data Rute Selama 1 Minggu

Hari	Tanggal	Rute	Titik Kirim							
Senin	5/1/2009	Rute 1	GRS	GRG	GRM	MD				
		Rute 2	GRC	CRL	GRP	DOP	MBP	GRK	PZK	
		Rute 3	JPG	GRL	PRN					
Selasa	6/1/2009	Rute 1	MPN	MPH	MPK	FMS	MPG	MPS		
		Rute 2	MPE	MPB	CRH	PZS	PZO			
		Rute 3	MD	GRM	GRG	GRS	GRC	MPC		
		Rute 4	CRL	DMF	PZK	PZC	SMP			
		Rute 5	MDLJ	DMA	FMG	DOG	JPG	MPL		
		Rute 6	MPR							
Rabu	7/1/2009	Rute 1	MPE	CRB	MPS	FMS	MPK	PRN		
		Rute 2	MD	MPQ	PZM	GRM	GRG	GRS	MBP	MPR
		Rute 3	MPO	CRL	GRC	DMF	PY	GNL	PZK	RCJ
		Rute 4	MPL	JPG	FMG	GRL	DMA			
		Rute 5	CRH	SPI						
Kamis	8/1/2009	Rute 1	RCM	CRP	PRN	MPK	FMS	MPS	CRB	
		Rute 2	MPE	CRH	CRA	MBP				
		Rute 3	MD	GRM	GRG	GRS	GRP	DOK		
		Rute 4	CRL	GRC	MPC	DMF	PZK			
		Rute 5	MPL	JPG	FMG	GRL	DMA			
Jumat	9/1/2009	Rute 1	HHD	CRN	MPN	HHK	MPH	MPK	MPS	CRB
		Rute 2	MPE	CRH	PZM	SRK	HHL	HHR		
		Rute 3	MD	GRM	GRG	GRS	GRK	MBP	DOP	GRP
		Rute 4	MPO	CRL	GRC	MPC	DMF	KS	HHF	
		Rute 5	JPG	FMG	DMA	MDLJ				
		Rute 6	MPL	CRQ	HHC	HHB	MPI			
Sabtu	10/1/2009	Rute 1	MPE	MPK	FMS	CRB				
		Rute 2	CRH	CRA	PZC	GNL	MBP	DOK		
		Rute 3	MD	GRM	GRG	PZG	GRS	GRP		
		Rute 4	CRL	GRC	MPC	DMF	RCD	RCJ		
		Rute 5	MPL	JPG	FMG	GRL	DMA			
		Rute 6	MPB	CRE	SPI					
Minggu	11/1/2009	Rute 1	MPE	MPV	MPN	MPK	FMS	MPS		
		Rute 2	MD	GRM	GRS	MBP	SPI			
		Rute 3	CRE	CRL	GRC	MPC	DMF	GNL		
		Rute 4	MPL	FMG	GRL	DMA				

(Sumber: PT. Saung Mirwan)

BAB 4

PENGOLAHAN DATA DAN ANALISIS

Bab ini berisi mengenai pengolahan data dan analisisnya. Pada bab ini, data yang telah diperoleh kemudian diolah dan digunakan untuk membuat suatu rancangan rute distribusi dengan penerapan VRP dan bantuan *software* MATLAB untuk dapat menghasilkan suatu rute pengiriman yang optimal. Kemudian dilakukan analisis terhadap hasil rancangan rute distribusi usulan tersebut. Analisis dilakukan untuk membandingkan rute pengiriman hasil penelitian dengan rute pengiriman yang diterapkan oleh perusahaan saat ini.

4.1 Algoritma Penyelesaian

Dalam melakukan optimasi rute distribusi harian pada perusahaan agribisnis ini, digunakan penerapan VRP dengan menggunakan algoritma metaheuristik *Tabu Search* seperti yang telah dijelaskan pada bab 2. Untuk dapat melakukan optimasi rute distribusi dengan algoritma *Tabu Search* tersebut, maka dibuat suatu program komputer dengan bantuan *software* MATLAB R2008a. (*script M-file* program terlampir). Tahapan algoritma *Tabu Search* digunakan sebagai dasar dari langkah kerja yang dilakukan oleh program tersebut.

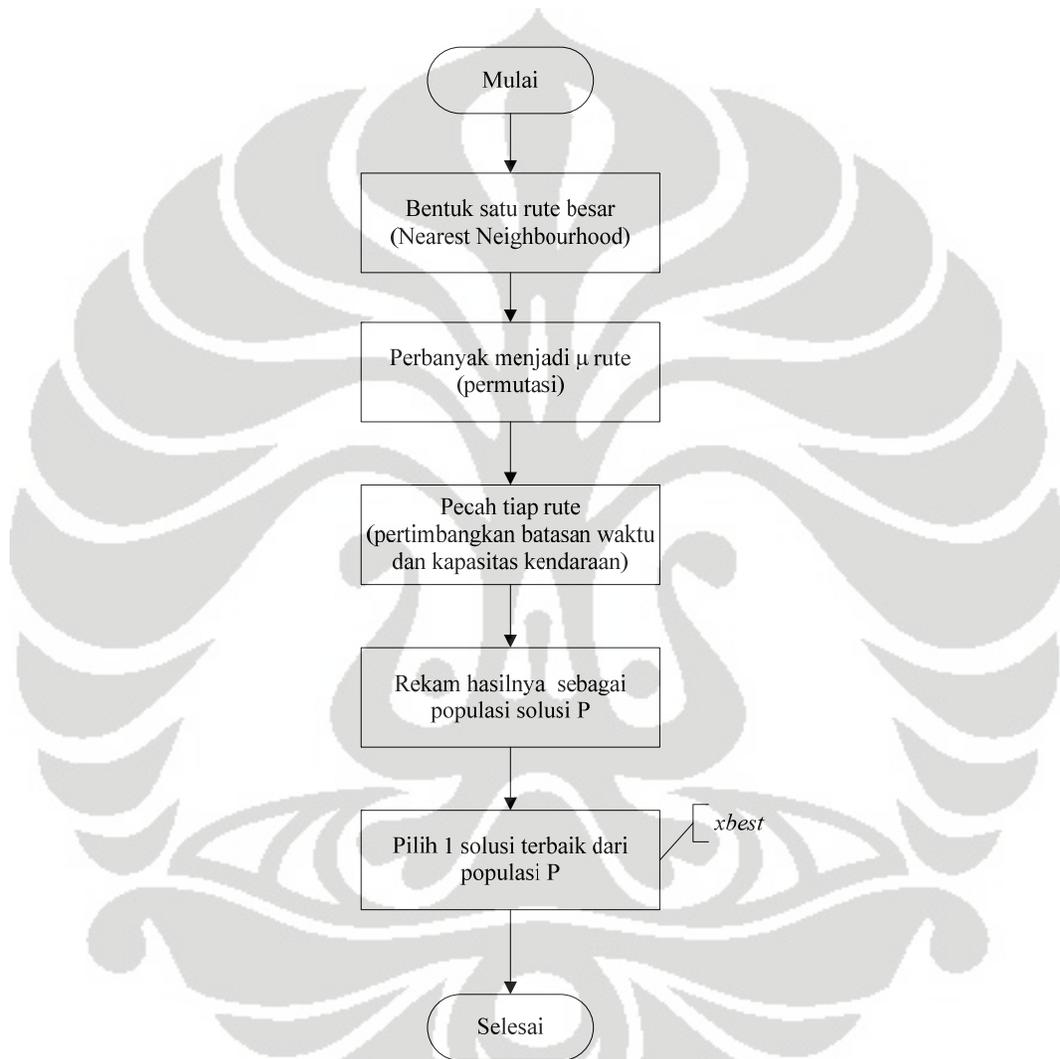
Secara umum, algoritma penyelesaian yang digunakan untuk melakukan optimasi rute distribusi ini terdiri dari dua tahapan, yaitu tahap pembentukan solusi awal dengan menggunakan algoritma *Nearest Neighbourhood* dan tahap meminimumkan jarak total dengan menggunakan *Tabu Search*.

Tahapan yang pertama adalah pembentukan solusi awal. Algoritma yang digunakan untuk memperoleh solusi awal adalah algoritma *Nearest Neighbourhood*. Tahapan pengerjaannya adalah sebagai berikut:

1. Bentuk suatu rute besar dengan ketentuan “mencari lokasi terdekat berikut”, seperti pada masalah TSP (*Traveling Salesman Problem*), dimulai dari depot hingga seluruh lokasi konsumen terlayani.
2. Kemudian perbanyak jumlah rute besar dengan cara permutasi (mengacak urutan lokasi) sehingga akhirnya diperoleh sejumlah μ rute besar.

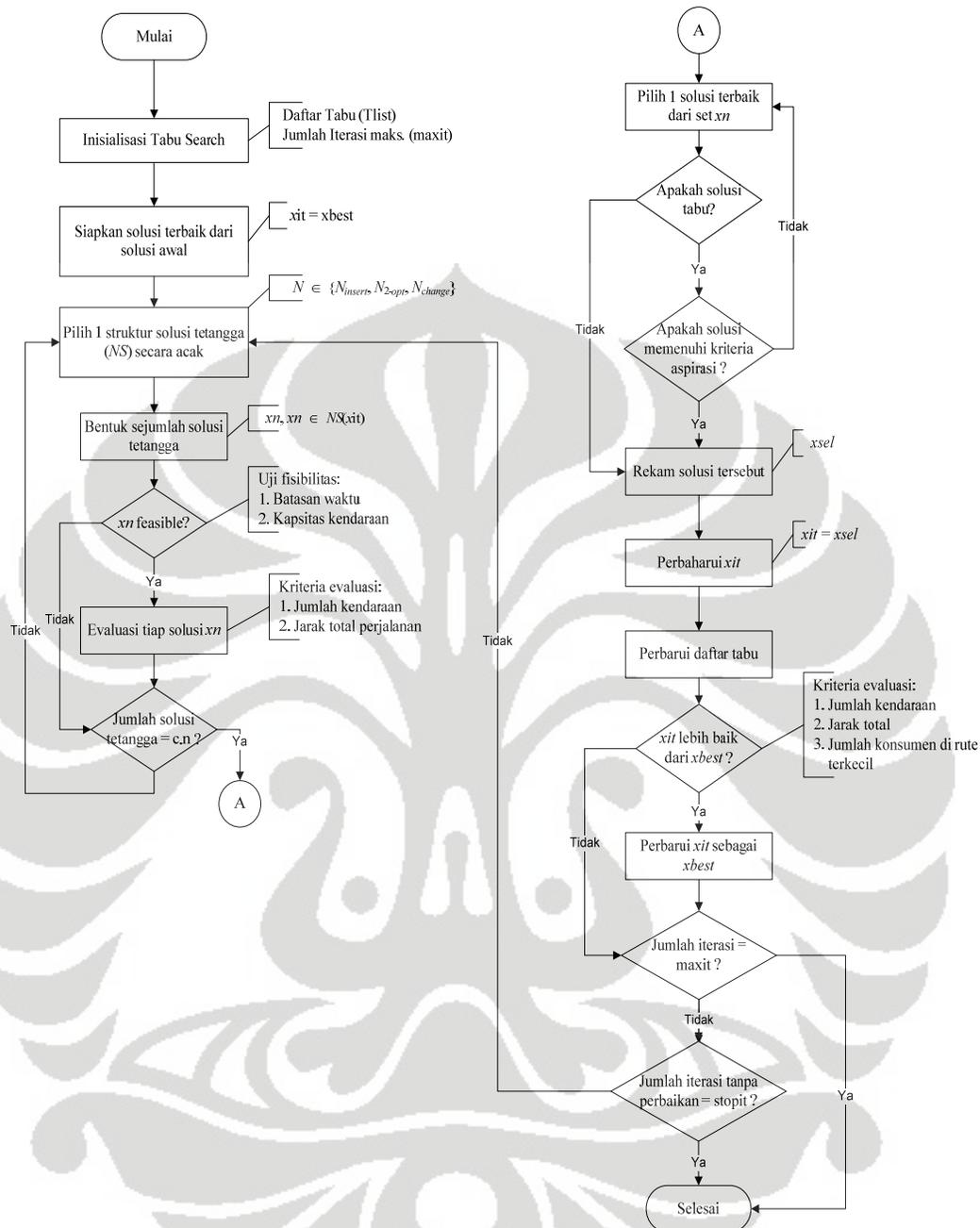
3. Pecah tiap rute besar tadi menjadi beberapa rute dengan menggunakan batasan kapasitas kendaraan dan juga batasan waktu.
4. Kemudian rekam semua hasilnya dan pilih solusi rute terbaik yang akan dijadikan sebagai solusi awal.

Berikut ini merupakan *flowchart* dari algoritma penyelesaian solusi awal.



Gambar 4.1 Algoritma Penyelesaian Solusi Awal

Setelah solusi awal diperoleh, maka dilanjutkan pada tahap berikutnya yaitu melakukan optimasi rute dengan algoritma *Tabu Search* seperti yang telah dijelaskan sebelumnya pada bab 2. Algoritma *Tabu Search* selengkapnya dapat dilihat pada *flowchart* berikut.



Gambar 4.2 Algoritma *Tabu Search* pada VRP

Dapat dilihat pada diagram alir diatas, pada tahap pembentukan solusi tetangga, dilakukan uji fisibilitas untuk setiap solusi tetangga (x_n) yang dihasilkan. Uji fisibilitas tersebut menyangkut dua aspek, yaitu:

- Kapasitas kendaraan. Perlu dipastikan bahwa jumlah pengiriman yang dilakukan berdasarkan solusi yang dihasilkan tidak melebihi kapasitas maksimum kendaraan yang melayani.

- Batasan waktu, yaitu waktu maksimum pelayanan konsumen. Perlu diperhatikan apakah solusi yang dihasilkan dapat memenuhi batasan waktu pelayanan (*time windows*) masing-masing konsumen, dan apakah konsumen berikutnya masih sempat dilayani.

Kemudian, dalam membandingkan solusi baru (*x_{it}*) terhadap solusi terbaik saat ini (*x_{best}*), digunakan tiga kriteria, yaitu jumlah kendaraan, jarak total, dan jumlah konsumen pada rute terkecil. Kriteria-kriteria tersebut bertujuan untuk mengarahkan proses pencarian solusi yang paling optimal, yaitu rute dengan jarak dan jumlah kendaraan terkecil. Kriteria jumlah konsumen pada rute terkecil digunakan untuk memperkirakan kemungkinan rute terkecil tersebut dapat dieliminasi sehingga jumlah kendaraan berkurang.

Dalam penerapan algoritma penyelesaian VRP ini, diperlukan adanya suatu konfigurasi parameter-parameter berikut:

- Ukuran populasi solusi awal (μ)
- Jumlah solusi tetangga yang dihasilkan pada tiap iterasi (*c.n*)
- Panjang *tabu list* (*Tlist*)
- Jumlah maksimum iterasi (*maxit*)
- Jumlah maksimum iterasi tanpa perbaikan (*stopit*), dimana selama *stopit* tidak terjadi perbaikan solusi pada *x_{best}*.

Penentuan nilai-nilai parameter tersebut untuk studi kasus distribusi perusahaan agribisnis ini dilakukan melalui studi parameter yang mengacu pada konfigurasi yang telah diuji Homberger dan Gehring (2004). Penjelasan mengenai studi parameter untuk menentukan konfigurasi ini akan dibahas pada bagian berikutnya.

Fungsi tujuan dari penyelesaian optimasi rute distribusi harian perusahaan agribisnis dengan algoritma *Tabu Search* ini adalah meminimumkan jarak total perjalanan yang harus ditempuh untuk melakukan pengantaran kepada seluruh konsumen tanpa melanggar batasan-batasan yang ada. Batasan-batasan yang dimaksud adalah sebagai berikut:

1. Pada tiap rute, besar muatan yang diangkut oleh kendaraan tidak boleh melebihi kapasitas kendaraan tersebut.
2. Jumlah rute tidak boleh lebih banyak daripada jumlah kendaraan.

3. Konsumen hanya dapat dilayani dalam rentang waktu tertentu (*time windows*).

4.2 Verifikasi dan Validasi Program

Sebelum program digunakan untuk pengolahan data kasus VRP, perlu dilakukan verifikasi dan validasi terhadap program tersebut. Verifikasi merupakan tahapan untuk melihat kesesuaian antara model program yang telah dibuat dengan konsep model yang kita inginkan. Bila program dapat berjalan sesuai dengan keinginan, maka program tersebut telah terverifikasi. Salah satu indikator yang dapat dilihat untuk membuktikannya adalah ketika dilakukan perubahan pada nilai parameter, *output* yang dihasilkan juga akan berubah.

Setelah program terverifikasi, maka selanjutnya perlu dilakukan validasi dengan tujuan untuk memastikan bahwa program tersebut menghasilkan *output* yang benar. Indikatornya adalah ketika program dihadapkan pada suatu masalah, hasil perhitungannya bernilai sama dengan perhitungan manual. Untuk melakukan validasi program ini, digunakan data buatan atau data *dummy*. Jumlah konsumen yang digunakan untuk proses verifikasi program ini adalah sebanyak 5 titik. Berikut ini merupakan tabel data-data *dummy* tersebut.

Tabel 4.1 Jarak Antartitik Konsumen *Dummy*

	0	1	2	3	4	5
0	inf	4	6	3	7	2
1	4	inf	2	6	3	7
2	6	2	inf	7	5	4
3	3	6	7	inf	2	5
4	7	3	5	2	inf	6
5	2	7	4	5	6	inf

Tabel 4.2 Jumlah Pengiriman Tiap Konsumen *Dummy*

Konsumen	Jumlah Pengiriman (krat)
1	30
2	32
3	40
4	26
5	45

Kapasitas kendaraan yang digunakan adalah 75 krat. *Service time* atau waktu *unloading* dan juga batasan waktu (*time windows*) yang digunakan sama seperti pada kasus distribusi harian perusahaan agribisnis, yaitu *service* adalah 10 menit dan *time windows* adalah dari pukul 05.00 hingga 09.00. Konfigurasi parameter yang digunakan untuk validasi program ini adalah sebagai berikut.

Tabel 4.3 Konfigurasi Paramater yang Digunakan dalam Validasi

Parameter	Nilai
Jumlah populasi awal (μ)	1
Jumlah solusi tetangga ($c.n$)	$1.5 = 5$
Panjang <i>tabu list</i> (Tlist)	2
maksimum iterasi ($maxit$)	1
maksimum iterasi tanpa perbaikan (<i>stopit</i>)	1

Hasil *run* program dengan menggunakan data *dummy* dan konfigurasi diatas menunjukkan solusi rute urutan perjalanan **0-5-0-2-1-0-4-3** dan jarak total perjalanan adalah 28 km. Kemudian, dilakukan perhitungan manual dengan mengikuti langkah kerja program, yaitu skenario pemilihan acak untuk mendapatkan solusi tetangga. Iterasi perhitungan manual yang dilakukan adalah sebagai berikut.

1. Pembentukan solusi awal

- Bentuk suatu rute besar (*neirest neighbour*) dengan ketentuan “mencari lokasi terdekat berikut”. Diperoleh urutan rute 0-5-2-1-4-3.
- Pecah rute tersebut berdasarkan kapasitas maksimum kendaraan dan batasan waktu. Diperoleh urutan rute **0-5-0-2-1-0-4-3-0** dengan jarak tempuh 28 km. Solusi ini merupakan solusi awal dan juga merupakan solusi terbaik saat ini (*xbest*).

2. Pencarian solusi optimal

- Solusi terbaik saat ini (*xbest*) adalah solusi awal, yaitu **0-5-0-2-1-0-4-3-0** (jarak tempuh 28 km)
- Bentuk lima solusi tetangga dengan menggunakan struktur solusi tetangga secara acak, kemudian membuat solusi fisibel berdasarkan kapasitas maksimum kendaraan dan kapasitas.

- Solusi tetangga 1 (xn_1)
 $xn_1 = change (5-2) \rightarrow 0-2-0-5-1-0-4-3-0$ (jarak tempuh 37 km)
- Solusi tetangga 2 (xn_2)
 $xn_2 = insert (5-4-0) \rightarrow 0-5-4-0-2-1-0-3-0$ (jarak tempuh 33 km)
- Solusi tetangga 3 (xn_3)
 $xn_3 = insert (5-1-0) \rightarrow 0-5-1-0-2-0-4-3-0$ (jarak tempuh 37 km)
- Solusi tetangga 4 (xn_4)
 $xn_4 = change (2-4) \rightarrow 0-5-0-4-1-0-2-3-0$ (jarak tempuh 34 km)
- Solusi tetangga 5 (xn_5)
 $xn_5 = change (1-4) \rightarrow 0-5-0-2-4-0-1-3-0$ (jarak tempuh 35 km)
- Pilih satu solusi terbaik dari $xn \rightarrow xn_2, xn_2 = xsel$
- $xsel = xit \rightarrow 0-5-4-0-2-1-0-3-0$ (jarak tempuh 33 km)
- Solusi xit tidak lebih baik dari $xbest$, maka solusi akhir adalah $xbest$, yaitu **0-5-0-2-1-0-4-3-0** dengan jarak tempuh 28 km.

Dapat dilihat bahwa hasil perhitungan manual diatas sama dengan hasil dari *run* program. Dengan demikian program telah tervalidasi.

4.3 Input Data

Untuk melakukan optimasi rute distribusi harian dengan algoritma *Tabu Search* ini, diperlukan data-data sebagai berikut.

- Data jarak depot dengan titik-titik konsumen dan juga jarak antarkonsumen. Data ini digunakan untuk mengetahui jarak terpendek antartitik dalam penyelesaian solusi awal (*Nearest Neighbourhood*). Data jarak ini juga dibutuhkan untuk menghitung jarak total perjalanan semua rute, baik rute yang digunakan oleh perusahaan, maupun rute hasil optimasi.
- Data permintaan harian tiap konsumen dan kapasitas kendaraan yang digunakan untuk melakukan pengantaran. Data tersebut digunakan untuk menghitung permintaan yang harus dipenuhi pada tiap rute dan membandingkannya dengan batasan kapasitas kendaraan.

- Data mengenai kecepatan rata-rata kendaraan waktu *unloading* (*service time*) dan batasan waktu (*time windows*). Data tersebut digunakan untuk menghitung waktu tempuh perjalanan tiap rute dan membandingkannya dengan batasan waktu yang ada. Untuk kecepatan rata-rata kendaraan digunakan kecepatan 45 km/ jam. Hal tersebut didasarkan dari hasil wawancara dengan para supir kendaraan dengan mempertimbangkan karakteristik jalan dan tingkat kemacetan yang terjadi.

Semua data tersebut telah dijelaskan pada bab sebelumnya dan dalam pengolahannya, data tersebut harus dimasukkan dalam *form input* yang telah dibuat pada *spreadsheet excel* guna mempercepat proses pengerjaan optimasi rute harian.

Seperti yang telah dijelaskan sebelumnya, jumlah konsumen dan juga jumlah permintaan selalu berubah-ubah setiap harinya. Oleh karena itu, rute distribusi harian juga akan berbeda-beda setiap harinya. Optimasi rute distribusi juga dilakukan untuk rute harian yang akan menghasilkan rute optimal untuk suatu hari tertentu. Pada pengolahan data kali ini, akan dilakukan optimasi pada 7 hari rute distribusi pengiriman produk. Hal ini dilakukan karena pola jumlah permintaan berulang setiap minggunya, meskipun jumlah konsumen dan permintaannya tidak tepat sama untuk setiap minggunya. Data permintaan yang akan diolah disini adalah data permintaan dari tanggal 5 Januari 2009 sampai dengan tanggal 11 Januari 2009.

4.4 Pengolahan Data

4.4.1 Konfigurasi Parameter

Sebelum melakukan pengolahan data dengan menggunakan program yang telah dibuat untuk menentukan usulan rute distribusi yang optimal, dilakukan penentuan konfigurasi parameter yang akan digunakan. Hal tersebut perlu dilakukan karena penggunaan konfigurasi yang tepat akan dapat menghasilkan solusi yang baik dengan waktu *run* program yang dapat diterima. Untuk menentukan konfigurasi yang tepat, dibutuhkan suatu studi parameter. Konfigurasi parameter yang akan ditentukan meliputi 5 variabel, yaitu ukuran populasi solusi awal (μ), jumlah solusi tetangga yang dihasilkan pada tiap iterasi

($c.n$), panjang *tabu list* ($Tlist$), jumlah maksimum iterasi ($maxit$), dan jumlah maksimum iterasi tanpa perbaikan ($stopit$).

4.4.1.1 Studi Parameter

Untuk melakukan studi parameter, kasus permasalahan yang digunakan adalah pengiriman produk pada tanggal 8 Januari 2009 dimana jumlah konsumen yang harus dilayani adalah sebanyak 27 konsumen. Hal tersebut dilakukan karena jumlah rata-rata konsumen harian adalah sekitar 27 kosumen, sehingga diharapkan dengan menggunakan kasus tersebut, konfigurasi parameter terbaik yang didapat akan sesuai untuk digunakan pada setiap kasus distribusi harian.

Berikut ini adalah beberapa alternatif konfigurasi parameter yang dapat digunakan untuk menyelesaikan kasus distribusi di perusahaan agribisnis ini.

- **Konfigurasi I**

Konfigurasi Homberger dan Gehring (2004) untuk kelompok permasalahan terkecil, yaitu permasalahan Solomom (100 titik). Konfigurasinya adalah sebagai berikut.

Tabel 4.4 Konfigurasi I

Parameter	Nilai
μ	10
$c.n$	$20 \cdot 27 = 540$
$Tlist$	10
$maxit$	1000
$stopit$	100

Pada konfigurasi diatas, jumlah iterasi maksimum ditentukan sebesar 1000 dan nilai iterasi maksimum tanpa perbaikan sebesar 100. Kemudian, nilai variabel c ditentukan sebesar 20, sehingga jumlah solusi tetangga adalah 20 dikalikan dengan jumlah konsumen (54), yaitu 540.

- **Konfigurasi II**

Konfigurasi dengan pola seperti pada konfigurasi I, namun jumlah iterasi menyesuaikan dengan jumlah konsumen, dimana:

- Nilai μ , $Tlist$, dan konstanta c adalah sama, yaitu 10,10, dan 2
- Nilai $maxit$ adalah jumlah konsumen dikalikan dengan 10
- Nilai $stopit$ sesuai dengan jumlah konsumen

Dengan menggunakan pola demikian, maka untuk kasus 27 konsumen, diperoleh konfigurasi sebagai berikut.

Tabel 4.5 Konfigurasi II

Parameter	Nilai
μ	10
$c.n$	$20 \cdot 27 = 540$
Tlist	10
<i>maxit</i>	270
<i>stopit</i>	27

- **Konfigurasi III**

Konfigurasi dengan pola seperti pada konfigurasi II, namun dengan nilai c yang lebih kecil. Karena c adalah bilangan konstan dimana $c > 0$, maka digunakan nilai c mulai dari $c = 1$ kemudian naik secara bertahap. Konfigurasinya adalah sebagai berikut.

Tabel 4.6 Konfigurasi III

Parameter	Nilai ($c = 1$)	Nilai ($c = 2$)	Nilai ($c = 3$)	Nilai ($c = \dots$)
μ	10	10	10	10
$c.n$	$1 \cdot 27 = 27$	$2 \cdot 27 = 54$	$3 \cdot 27 = 81$...
Tlist	10	10	10	10
<i>maxit</i>	270	270	270	270
<i>stopit</i>	27	27	27	27

- **Konfigurasi IV**

Konfigurasi dengan pola seperti pada konfigurasi III, namun dengan nilai *maxit* dan *stopit* seperti pada konfigurasi I, yaitu 1000 dan 100. Konfigurasinya adalah sebagai berikut.

Tabel 4.7 Konfigurasi IV

Parameter	Nilai ($c = 1$)	Nilai ($c = 2$)	Nilai ($c = 3$)	Nilai ($c = \dots$)
μ	10	10	10	10
$c.n$	$1 \cdot 27 = 27$	$2 \cdot 27 = 54$	$3 \cdot 27 = 81$...
Tlist	10	10	10	10
<i>maxit</i>	1000	1000	1000	1000
<i>stopit</i>	100	100	100	100

4.4.1.2 Hasil Studi Parameter

Untuk memperoleh konfigurasi mana yang tepat untuk digunakan, maka keempat alternatif konfigurasi yang ada dicoba satu per satu untuk kemudian dilihat solusi yang dihasilnya dan juga waktu *run* program. Hasilnya adalah sebagai berikut.

- **Konfigurasi I**

Konfigurasi ini menggunakan nilai konstanta c yang besar, yaitu 20, dan juga jumlah iterasi yang cukup besar yaitu 1000. Dilakukan 10 kali *run* program dengan menggunakan konfigurasi ini agar diperoleh rata-rata jarak tempuh total dan waktu yang dibutuhkan untuk satu kali *run* program. Hasilnya adalah sebagai berikut.

Tabel 4.8 Hasil Penggunaan Konfigurasi I

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	326.45	684.13
2	303.78	683.62
3	315.28	662.43
4	292.94	668.39
5	300.73	681.13
6	284.59	690.43
7	346.51	680.14
8	303.65	669.12
9	313.66	686.77
10	300.87	681.13
Rata-rata	308.846	678.729

Jika dilihat dari hasil diatas, penggunaan konfigurasi I ini menghasilkan solusi yang cukup baik. Rata-rata jarak total yang dihasilkan adalah 678,729 km. Solusi yang dihasilkan juga cenderung stabil, yaitu antara 662,43 km hingga 690,43 km. Namun demikian, waktu rata-rata untuk melakukan satu kali *run* program relatif sangat lama, yaitu 308,846 detik.

- **Konfigurasi II**

Perbedaan konfigurasi II dengan konfigurasi I terletak pada jumlah iterasi yang lebih kecil. Berikut ini adalah hasil 10 kali *run* program dengan menggunakan konfigurasi ini.

Tabel 4.9 Hasil Penggunaan Konfigurasi II

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	133.54	689.04
2	127.23	691.17
3	133.43	706.74
4	134.46	668.39
5	137.54	683.44
6	133.95	672.99
7	127.43	671.98
8	136.61	688.99
9	134.11	682.95
10	131.54	684.71
Rata-rata	132.984	684.040

Dapat dilihat dari tabel hasil *run* program diatas, kualitas rata-rat solusi yang dihasilkan pada penggunaan konfigurasi II ini tidak lebih baik dari konfigurasi I. Hal tersebut dikarenakan jumlah iterasinya pada konfigurasi ini lebih sedikit dibandingkan dengan konfigurasi I. Namun demikian, waktu rata-rata untuk satu kali *run* program lebih cepat dibandingkan dengan penggunaan konfigurasi I meskipun secara umum waktu *run* tersebut relatif masih lama.

- **Konfigurasi III**

Pada konfigurasi ini, diuji setiap nilai c mulai dari $c = 1$. Untuk setiap nilai c , dilakukan 10 kali *run* program. Hasilnya adalah sebagai berikut.

- Nilai $c = 1$ ($c.n = 1.27 = 27$)

Tabel 4.10 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 1$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	17.46	706.57
2	16.37	709.74
3	16.91	691.94
4	18.05	710.09
5	17.03	690.7
6	16.45	691.73
7	14.69	701.15
8	17.23	694.7
9	16.67	787.56
10	15.76	702.84
Rata-rata	16.662	708.702

- Nilai $c = 2$ ($c.n = 2.27 = 54$)

Tabel 4.11 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 2$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	23.12	701.15
2	26.37	683.78
3	24.76	684.46
4	28.98	697.23
5	22.36	694.81
6	23.13	668.39
7	22.57	684.71
8	24.35	671.98
9	21.54	688.99
10	27.785	663.94
Rata-rata	24.497	683.944

- Nilai $c = 3$ ($c.n = 3.27 = 81$)

Tabel 4.12 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 3$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	34.84	706.57
2	37.25	698.25
3	33.62	663.94
4	36.06	686.77
5	32.76	674.26
6	32.44	703.26
7	34.13	680.04
8	31.54	696.61
9	34.87	671.98
10	35.65	675.7
Rata-rata	34.316	685.738

- Nilai $c = 4$ ($c.n = 4.27 = 108$)

Tabel 4.13 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 4$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	40.98	676.02
2	43.87	693.79
3	42.65	692.74
4	45.16	702.37
5	39.89	694.54

Tabel 4.13 Hasil Penggunaan Konfigurasi III dengan Nilai $c = 4$ (sambungan)

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
6	43.82	701.59
7	50.28	663.94
8	48.21	682.52
9	36.63	682.95
10	42.89	674.26
Rata-rata	43.438	686.472

Dapat dilihat dari hasil-hasil diatas, semakin besar nilai konstanta c , maka akan semakin lama waktu *run* program. Kemudian, dapat dilihat bahwa hasil rata-rata jarak total yang paling baik adalah dengan menggunakan nilai $c = 2$, yaitu 683,944 km, selain itu hasil jarak terkecil juga keluar pada saat menggunakan konfigurasi nilai $c = 2$, yaitu 663,94 km. Pada nilai $c = 2$, rata-rata waktu untuk 1 kali *run* program adalah 24,497 detik. Karena tidak ada perbaikan hasil untuk nilai $c > 2$, maka pengujian hanya dilakukan hingga nilai $c = 4$.

- **Konfigurasi IV**

Pola konfigurasi ini mirip seperti pola pada konfigurasi III. Hanya saja jumlah iterasinya diperbesar menjadi 1000 dan jumlah iterasi tanpa perbaikan ditetapkan sejumlah 100. Sama seperti konfigurasi III, pada konfigurasi ini, diuji setiap nilai c mulai dari $c = 1$. Untuk setiap nilai c , juga dilakukan 10 kali *run*. Hasilnya adalah sebagai berikut.

- Nilai $c = 1$ ($c.n = 1.27 = 27$)

Tabel 4.14 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 1$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	31.82	695.57
2	28.32	700.23
3	31.14	714.61
4	33.14	729.47
5	27.65	699.93
6	33.54	676.49
7	29.18	691.67
8	32.43	689.66
9	30.28	688.99
10	32.45	701.59
Rata-rata	30.995	698.821

- Nilai $c = 2$ ($c.n = 2.27 = 54$)

Tabel 4.15 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 2$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	41.26	674.26
2	39.19	691.34
3	39.08	691.67
4	41.82	662.43
5	37.53	667.51
6	36.61	688.05
7	48.12	662.43
8	40.12	694.81
9	38.43	671.98
10	41.65	680.25
Rata-rata	40.381	678.276

- Nilai $c = 3$ ($c.n = 3.27 = 81$)

Tabel 4.16 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 3$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	58.37	694.91
2	60.62	680.04
3	54.71	684.72
4	58.96	662.43
5	53.32	695.56
6	59.22	675.82
7	54.43	675.7
8	56.34	679.74
9	58.85	694.91
10	57.91	667.51
Rata-rata	57.273	681.134

- Nilai $c = 4$ ($c.n = 4.27 = 108$)

Tabel 4.17 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 4$

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
1	66.94	699.84
2	64.41	689.79
3	79.47	699.48
4	66.34	680.14
5	73.43	665.83

Tabel 4.17 Hasil Penggunaan Konfigurasi IV dengan Nilai $c = 4$ (sambungan)

<i>Run</i>	Waktu <i>Run</i> (detik)	Jarak Total (km)
6	63.32	691.67
7	59.53	682.95
8	68.41	667.51
9	70.53	663.94
10	72.98	671.98
Rata-rata	68.536	681.313

Dapat dilihat dari hasil *run* program diatas bahwasanya rata-rata jarak tempuh terbaik dihasilkan oleh pada nilai $c = 2$, yaitu 678,276 km. Kemudian hasil jarak rute terpendek juga dihasilkan pada nilai $c = 2$, yaitu 662,43 km. Pada nilai $c = 2$ ini, rata-rata waktu untuk 1 kali *run* program adalah 40,381 detik. Karena tidak ada perbaikan hasil untuk $c > 2$, maka pengujian hanya dilakukan hingga nilai $c = 4$.

Berikut ini adalah tabel rekap yang menunjukkan hasil dari penggunaan keempat alternatif konfigurasi untuk melakukan *run* program.

Tabel 4.18 Rekap Hasil Studi Parameter

Konfigurasi	Rata-Rata Waktu <i>Run</i> (detik)	Rata-Rata Jarak Tempuh (km)
I	308.846	678.729
II	132.984	684.04
III ($c=2$)	24.497	683.944
IV ($c=2$)	40.381	678.276

Kemudian dapat dilihat dari tabel diatas, rata-rata jarak tempuh total yang paling baik dihasilkan oleh penggunaan konfigurasi I dan konfigurasi IV ($c = 2$), yaitu 678,729 km dan 678,276 km. Selain itu pada penggunaan dua konfigurasi tersebut, nilai solusi jarak paling minimum dihasilkan, yaitu 662.43 km. Dengan demikian, kedua konfigurasi ini dipilih menjadi 2 kandidat konfigurasi terbaik.

Jika dilihat dari lamanya waktu *run* program yang dihasilkan, penggunaan konfigurasi IV akan menghasilkan waktu *run* program yang jauh lebih kecil dibandingkan dengan jika menggunakan konfigurasi I. Rata-rata waktu *run* program pada konfigurasi IV adalah 40,381 detik, sedangkan pada konfigurasi I, rata-rata waktu *run* program adalah 308,846 detik. Oleh karena hal tersebut, maka

konfigurasi terbaik yang akan digunakan untuk melakukan optimasi rute distribusi harian adalah konfigurasi IV ($c = 2$), dengan pertimbangan kualitas solusi yang dihasilkan cukup stabil dan baik dan waktu *run* program yang relatif lebih singkat.

4.4.2 Hasil Pengolahan Data

Setelah menentukan konfigurasi parameter yang paling tepat, maka dilakukan pengolahan data, yaitu melakukan optimasi rute distribusi harian. Data yang diinputkan ke dalam program ini merupakan data harian, sehingga hasilnya pun masih bersifat harian.

Output yang dihasilkan oleh program MATLAB yang telah dibuat adalah total jarak perjalanan beserta dengan rute-rute perjalanan. Selain itu, dari hasil *output* program, dapat juga diketahui mengenai jumlah muatan yang diangkut oleh tiap-tiap kendaraan. Data yang diolah adalah data pengiriman selama tujuh hari dari tanggal 5 januari 2009 sampai tanggal 11 januari 2009. Untuk masing-masing hari, dilakukan lima kali *run* program dengan menggunakan konfigurasi parameter yang telah ditentukan sebelumnya. Kemudian dari kelima hasil *run* program yang diperoleh, dipilih satu solusi yang paling baik yaitu yang memiliki jarak tempuh total paling kecil. Berikut ini adalah *output* jarak tempuh yang dihasilkan dari pengolahan pada tujuh hari pengiriman produk.

Tabel 4.19 Jarak Tempuh Harian untuk 5 Kali *Run* Program

Hari / Tanggal	Jarak Tempuh Total (km)				
	<i>Run</i> ke-1	<i>Run</i> ke-2	<i>Run</i> ke-3	<i>Run</i> ke-4	<i>Run</i> ke-5
Senin / 5-Jan-09	469.44	469.53	469.44	469.53	470.70
Selasa / 6-Jan-09	859.19	865.21	854.08	868.48	855.91
Rabu / 7-Jan-09	899.52	894.61	898.11	893.79	893.43
Kamis / 8-Jan-09	662.43	667.51	671.98	662.43	674.26
Jumat / 9-Jan-09	976.68	986.66	984.87	984.21	973.97
Sabtu / 10-Jan-09	784.62	783.27	784.99	787.77	785.75
Minggu / 11-Jan-09	662.66	662.41	662.66	662.41	662.82

Angka jarak tempuh total yang dicetak tebal pada tabel diatas merupakan jarak tempuh total yang paling kecil untuk tiap-tiap harinya. Solusi yang menghasilkan jarak terkecil tersebut merupakan solusi terbaik yang akan digunakan untuk menentukan rute distribusi pengantaran produk harian. Berikut

ini merupakan rute-rute distribusi harian beserta dengan jarak tempuh tiap rute yang merupakan hasil solusi terbaik yang diperoleh.

Tabel 4.20 Rute dan Jarak Tempuh Total Hasil Optimasi

Hari / Tanggal	Rute	Titik Kirim										Jarak Rute (km)	Jarak Total (km)
Senin 5-Jan-09	Rute 1	GRS	GRK	MBP	DOP	GRP	PZK					136.82	469.44
	Rute 2	JPG	GRL	GRG	GRM	PRN	CRL	GRC				190.92	
	Rute 3	MD										141.7	
Selasa 6-Jan-09	Rute 1	PZK	PZS	PZO	PZC	GRM	GRG	GRS	CRH	MPL		157.89	854.08
	Rute 2	MPS	MPG	CRL	GRC	MPC	DMF	SMP				170.78	
	Rute 3	MPB	MPR	MPE								179.75	
	Rute 4	FMG	JPG	MD	DOG	DMA						141.34	
	Rute 5	MPN	MDLJ	MPH	FMS	MPK						204.32	
Rabu 7-Jan-09	Rute 1	PZK	MBP	PRN	FMS	MPK	MPS	CRB	DMF	GRC		208.16	893.43
	Rute 2	FMG	MPQ	GRM	PZM	GRG	GNL	GRS	RCJ			153.38	
	Rute 3	MD	JPG	GRL	DMA	MPE						148.79	
	Rute 4	MPO	CRL	PY	CRH	MPL	MPR					221.86	
	Rute 5	SPI										161.24	
Kamis 8-Jan-09	Rute 1	CRB	MPS	FMS	PRN	MPK	CRP	RCM	GRP			213.84	662.43
	Rute 2	MD	JPG									141.7	
	Rute 3	CRL	MPC	GRC	DMF	DOK	PZK	MPE				129.98	
	Rute 4	CRA	MBP	GRS	GRG	GRM	DMA	GRL	FMG	MPL	CRH	176.91	
Jumat 9-Jan-09	Rute 1	MD	JPG	FMG	DMA							142.41	973.97
	Rute 2	CRB	MPS	MPH	MPK	MDLJ	HHK	DOP	GRP			205.23	
	Rute 3	HHR	HHD	CRN	MPN	CRH	MPL					186.34	
	Rute 4	MPE	MPO	HHF	KS	MPC	GRC	CRL				140.07	
	Rute 5	DMF	GRK	MBP	GRS	GRG	GRM	PZM	SRK	HHL		151.40	
	Rute 6	HHB	MPI	CRQ	HHC							148.52	
Sabtu 10-Jan-09	Rute 1	FMG	JPG	MD	GRL	DMA						142.42	783.27
	Rute 2	CRB	FMS	MPK	GRM	PZG	GRG	PZC				188.82	
	Rute 3	MPB	DMF	GRP	RCD	MBP	GNL	GRS	CRA			135.81	
	Rute 4	CRE	MPL	CRH	SPI							186.54	
	Rute 5	MPE	MPC	GRC	CRL	DOK	RCJ					129.68	
Minggu 11-Jan-09	Rute 1	MPE	MPV	DMF	GRC	MPC	CRL					123.84	662.41
	Rute 2	MBP	GNL	GRS	GRM	MPN	FMS	MPK	MPS			208.52	
	Rute 3	MPL	FMG	GRL	MD	DMA						164.89	
	Rute 4	SPI	CRE									165.16	

Tabel diatas merupakan hasil dari optimasi rute pengiriman produk harian. Dapat dilihat bahwasanya jumlah rute pengiriman atau jumlah kendaraan yang digunakan untuk pengantaran berbeda-beda untuk setiap harinya. Hal tersebut

tergantung pada jumlah permintaan harian yang harus diantarkan kepada konsumen.

Kemudian, jarak tempuh total yang dihasilkan pada rute hasil optimasi ini lebih pendek dibandingkan dengan jarak tempuh pada rute pengantaran yang dipakai oleh perusahaan saat ini. Hal ini dikarenakan rute usulan merupakan rute yang dihasilkan dari optimasi dengan menggunakan algoritma *Tabu Search*. Berikut ini adalah perbandingan jarak tempuh dan juga jumlah kendaraan yang digunakan antara rute yang saat ini digunakan oleh pihak perusahaan dan rute usulan hasil optimasi.

Tabel 4.21 Perbandingan Jumlah Kendaraan dan Jarak Tempuh Antara Rute Saat Ini dan Rute Usulan

Hari / Tanggal	Rute Saat Ini		Rute Usulan	
	Jumlah Kendaraan	Jarak Tempuh (km)	Jumlah Kendaraan	Jarak Tempuh (km)
Senin / 5-Jan-09	3	485.68	3	469.44
Selasa / 6-Jan-09	6	1037.81	5	854.08
Rabu / 7-Jan-09	5	919.70	5	893.43
Kamis / 8-Jan-09	5	810.15	4	662.43
Jumat / 9-Jan-09	6	1045.05	6	973.97
Sabtu / 10-Jan-09	6	945.42	5	783.27
Minggu / 11-Jan-09	4	719.11	4	662.41
Total	35	5962.92	32	5299.03

Dapat dilihat dari tabel diatas, total jarak tempuh selama satu minggu pada penggunaan rute saat ini adalah sebesar 5.962,92 km, sedangkan untuk rute usulan adalah sebesar 5.299,03 km. Terjadi penurunan jarak tempuh, yaitu sebesar 663,89 km atau sebesar 11,13 %. Selain itu dapat dilihat juga, untuk beberapa hari, terjadi pengurangan jumlah kendaraan yang digunakan untuk melakukan distribusi produk, yaitu pada hari Selasa, Kamis, dan Sabtu.

Hasil berikutnya adalah mengenai utilitas tiap-tiap kendaraan. Utilitas kendaraan disini maksudnya adalah seberapa besar kapasitas dari satu kendaraan yang digunakan untuk melakukan satu kali pengiriman. Berikut ini adalah rekap hasil mengenai muatan atau jumlah pengiriman yang dilakukan tiap kendaraan dan juga persentase utilitas tiap kendaraan.

Tabel 4.22 Muatan dan Persentase Utilitas Kendaraan Hasil Optimasi

Hari / Tanggal	Rute	Muatan Tiap Kendaraan (krat)	Persentase Utilitas Kendaraan
Senin 5-Jan-09	Rute 1	43	57.33%
	Rute 2	61	81.33%
	Rute 3	69	92.00%
Selasa 6-Jan-09	Rute 1	67	89.33%
	Rute 2	69	92.00%
	Rute 3	58	77.33%
	Rute 4	74	98.67%
	Rute 5	67	89.33%
Rabu 7-Jan-09	Rute 1	75	100.00%
	Rute 2	54	72.00%
	Rute 3	72	96.00%
	Rute 4	70	93.33%
	Rute 5	65	86.67%
Kamis 8-Jan-09	Rute 1	71	94.67%
	Rute 2	73	97.33%
	Rute 3	67	89.33%
	Rute 4	73	97.33%
Jumat 9-Jan-09	Rute 1	75	100.00%
	Rute 2	71	94.67%
	Rute 3	59	78.67%
	Rute 4	73	97.33%
	Rute 5	73	97.33%
	Rute 6	40	53.33%
Sabtu 10-Jan-09	Rute 1	74	98.67%
	Rute 2	71	94.67%
	Rute 3	71	94.67%
	Rute 4	60	80.00%
	Rute 5	65	86.67%
Minggu 11-Jan-09	Rute 1	64	85.33%
	Rute 2	64	85.33%
	Rute 3	66	88.00%
	Rute 4	68	90.67%

4.5 Analisis

Kasus distribusi pengiriman produk di perusahaan agribisnis ini termasuk dalam kelas VRP *with Time Windows* (VRPTW) dikarenakan konsumen hanya dapat dilayani pada rentang waktu tertentu. Usulan penyelesaian kasus tersebut adalah dengan menggunakan algoritma *Tabu Search*. Algoritma tersebut

Universitas Indonesia

dituangkan dalam suatu bahasa pemrograman MATLAB, yang selanjutnya digunakan untuk melakukan pengolahan data untuk membentuk solusi rute distribusi yang optimal. Fungsi tujuannya adalah untuk meminimalkan jarak tempuh total.

Analisis terhadap hasil pengolahan data ini dibagi menjadi tiga bagian, yaitu analisis metode, analisis program, dan analisis hasil optimasi.

4.5.1 Analisis Metode

4.5.1.1 Analisis Algoritma

Metode yang digunakan untuk menyelesaikan permasalahan optimasi rute distribusi harian di perusahaan agribisnis ini adalah dengan penerapan VRP dengan menggunakan algoritma *Tabu Search*. Penggunaan algoritma ini terbukti berhasil untuk mendapatkan solusi rute distribusi optimal yang lebih baik dari rute yang selama ini digunakan oleh pihak perusahaan.

Proses pencarian solusi dilakukan dengan mengiterasi solusi awal sampai diperoleh hasil yang paling optimal. Pada setiap iterasi, dibentuk sejumlah solusi tetangga dengan memilih secara acak diantara tiga mekanisme pembentukan solusi tetangga yang ada (*insertion*, *2-opt*, dan *interchange*). Hal inilah yang menyebabkan adanya perbedaan *output* rute optimal yang dihasilkan pada setiap *run* program, meskipun data yang diinput dan konfigurasi yang digunakan sama.

Pada pencarian solusi awal, digunakan algoritma *Nearest Neighbourhood*. Pemilihan algoritma ini dikarenakan mekanismenya yang sederhana dan cepat untuk mendapatkan sebuah solusi. Kekurangan dari algoritma ini adalah kualitas solusi yang dihasilkan kurang baik jika dibandingkan dengan algoritma heuristik lain seperti metode *saving*. Namun demikian, kualitas solusi awal disini kurang begitu penting karena solusi tersebut nantinya akan diperbaiki dengan menggunakan algoritma *Tabu Search*. Untuk mendapatkan kualitas solusi awal yang lebih baik, maka dilakukan permutasi secara acak pada solusi yang dihasilkan dari algoritma *Nearest Neighbourhood* untuk mendapatkan sejumlah μ solusi. Kemudian, dari sejumlah μ solusi tersebut, dipilih solusi terbaik yang akan digunakan sebagai solusi awal.

Untuk algoritma *Tabu Search* sendiri, skema tabu tenure yang digunakan adalah *fixed tabu tenure*. Tabu tenure merupakan durasi status tabu atau lamanya waktu suatu solusi atau atribut solusi dianggap tabu. Tabu tenure ini juga menunjukkan berapa panjang *tabu list* (*Tlist*). Skema *fixed tabu tenure* berarti menggunakan panjang *tabu list* yang tetap sepanjang algoritma *tabu search* dijalankan. Pada penelitian ini digunakan panjang *tabu list* sebesar sepuluh, sehingga suatu solusi akan dianggap tabu selama sepuluh iterasi. Penentuan panjang *tabu list* ini didasarkan pada studi konfigurasi yang telah dilakukan, yaitu berdasarkan konfigurasi yang digunakan oleh Homberger dan Gehring.

4.5.1.2 Analisis Studi Parameter

Pada studi parameter yang telah dilakukan, hanya dilakukan percobaan untuk menentukan tiga parameter yang paling berpengaruh pada waktu *run* program dan juga solusi yang dihasilkan. Studi parameter ini didasarkan pada konfigurasi yang digunakan oleh Homberger dan Gehring dalam menyelesaikan kasus VRP dengan algoritma *Tabu Search*. Untuk nilai panjang *tabu list* (*Tlist*) dan juga ukuran populasi solusi awal (μ), digunakan nilai yang sama pada konfigurasi Homberger dan Gehring, yaitu masing-masing 10. Kemudian untuk variabel lainnya, yaitu jumlah solusi (*c.n*), jumlah maksimum iterasi (*maxit*), dan jumlah maksimum iterasi tanpa perbaikan (*stopit*) dilakukan percobaan untuk beberapa nilai. Hal tersebut dilakukan karena ketiga parameter ini memiliki pengaruh yang besar, khususnya pada lama waktu *run* program.

Terdapat tiga hal yang mendasari pemilihan konfigurasi parameter yang tepat untuk kasus distribusi harian, yaitu kestabilan rute yang dihasilkan, kesesuaian dengan fungsi tujuan, dan waktu *run* program yang relatif masih dapat diterima. Dari beberapa alternatif konfigurasi yang diuji, jumlah iterasi yang kecil akan menyebabkan ketidakstabilan rute yang dihasilkan dan juga kualitas rute yang kurang baik, seperti pada konfigurasi II dan III. Namun, jumlah iterasi yang besar juga akan mengakibatkan waktu *run* program yang relatif lama.

Konfigurasi yang digunakan Homberger dan Gehring (2004), yaitu konfigurasi I menghasilkan solusi yang cukup baik dan stabil. Namun, waktu untuk satu kali *run* program relatif cukup lama, yaitu sekitar 308,846 detik atau

sekitar 5 menit. Sementara itu, konfigurasi IV pada nilai $c = 2$ juga menghasilkan solusi yang baik dengan waktu rata-rata *run* program jauh lebih cepat yaitu 40.381 detik. Sedangkan untuk konfigurasi II dan III kualitas solusi yang dihasilkan tidak cukup baik dikarenakan jumlah iterasi yang relatif sedikit.

Jika dilihat lebih lanjut, perbedaan antara konfigurasi I dan konfigurasi IV hanya terletak pada nilai jumlah solusi tetangga ($c.n$). Hal ini menunjukkan, selain jumlah iterasi, nilai jumlah solusi tetangga ($c.n$) ini juga sangat mempengaruhi waktu *run* program. Pada konfigurasi I, digunakan nilai $c = 20$, sedangkan pada konfigurasi IV, hasil terbaik adalah pada nilai $c = 2$. Dari penggunaan nilai c yang berbeda cukup jauh ini, didapatkan hasil solusi yang relatif sama baiknya, namun dengan perbedaan waktu *run* program yang sangat jauh, yaitu sekitar 284.35 detik. Oleh karena hal tersebut, maka konfigurasi IV dianggap sebagai konfigurasi terbaik untuk digunakan dalam menyelesaikan masalah distribusi harian di perusahaan agribisnis ini. Konfigurasi tersebut adalah sebagai berikut.

Tabel 4.23 Konfigurasi Terbaik

Parameter	Nilai
μ	10
$c.n$	$2 \cdot n = 2n$
T_{list}	10
$maxit$	1000
$stopit$	100

Dengan demikian, dapat dikatakan bahwa dalam menyelesaikan kasus distribusi dengan menggunakan algoritma *Tabu Search* ini, penggunaan konfigurasi parameter yang tepat merupakan kunci untuk memperoleh hasil yang terbaik (baik solusi maupun waktu *run* program). Acuan konfigurasi parameter dari Homberger dan Gehring (konfigurasi I), terbukti cukup baik, namun perlu dilakukan pengujian kembali pada nilai $c.n$, karena selain mempengaruhi kualitas solusi, nilai $c.n$ juga sangat berpengaruh pada rata-rata waktu *run* program.

Studi parameter yang dilakukan penulis menggunakan kasus distribusi pada hari Kamis tanggal 8 Januari 2009 dengan jumlah titik kirim sebanyak 27 konsumen. Hal tersebut dilakukan dengan pertimbangan bahwa rata-rata jumlah konsumen harian adalah sekitar 27 konsumen. Konfigurasi terbaik dari hasil studi

parameter kemudian digunakan untuk mengolah kasus distribusi pada hari-hari lain. Hal tersebut dapat dilakukan karena konfigurasi yang digunakan (konfigurasi IV) dapat menyesuaikan nilai $c.n$ dengan jumlah konsumen (n) yang ada pada setiap harinya, sehingga jika jumlah konsumennya berbeda, maka nilai $c.n$ juga akan berbeda.

Namun demikian, studi parameter yang lebih mendalam diperlukan untuk dapat menguji konfigurasi parameter seperti apa yang paling tepat untuk digunakan dalam menyelesaikan kasus distribusi harian di perusahaan ini. Mungkin saja konfigurasi parameter terbaik untuk setiap kasus distribusi harian di perusahaan ini berbeda-beda antara hari satu dengan hari lainnya, karena tiap kasus membutuhkan konfigurasi yang unik untuk memperoleh rute yang stabil dan waktu *run* yang relatif singkat. Selain itu, juga perlu diteliti apakah ada faktor lain yang mempengaruhi kualitas solusi dan waktu *run* program selain parameter $c.n$ dan jumlah iterasi. Hal ini dapat dilakukan dengan mengubah-ubah nilai tiap parameter.

4.5.2 Analisis Program

Bahasa pemrograman yang digunakan untuk menginterpretasikan algoritma *Tabu Search* pada penelitian ini adalah MATLAB. Secara umum program yang telah dibuat untuk menyelesaikan permasalahan VRP ini dapat berjalan dengan baik. Solusi akhir yang dihasilkan selalu lebih baik daripada solusi awal. Selain itu solusi rute optimal yang dihasilkan juga memberikan perbaikan pada rute yang selama ini digunakan oleh pihak perusahaan. Hal tersebut ditandai dengan hasil jarak tempuh perjalanan yang lebih kecil dan juga penurunan jumlah kendaraan pada beberapa hari pengiriman.

Kelemahan dari program ini adalah waktu *run* program yang relatif lama. Seperti yang telah dijelaskan sebelumnya, waktu *run* program ini dipengaruhi oleh konfigurasi parameter yang digunakan. Namun, perbedaan jumlah konsumen juga akan mempengaruhi waktu *run* program. Untuk lebih jelasnya dapat dilihat pada tabel berikut.

Tabel 4.24 Waktu Run Program untuk Kasus Harian

Hari / Tanggal	Jumlah Konsumen	Waktu Run Program (detik)					Rata-Rata
		Run ke-1	Run ke-2	Run ke-3	Run ke-4	Run ke-5	
Senin / 5-Jan-09	14	9.72	9.35	9.29	9.35	9.08	9.36
Selasa / 6-Jan-09	29	48.76	47.2	52.36	46.89	48.63	48.77
Rabu / 7-Jan-09	29	48.22	52.09	49.2	48.94	45.87	48.86
Kamis / 8-Jan-09	27	41.26	37.53	38.43	41.82	43.43	40.49
Jumat / 9-Jan-09	38	82.45	85.78	72.93	75.23	80.47	79.37
Sabtu / 10-Jan-09	30	44.43	49.12	51.87	48.51	52.08	49.20
Minggu / 11-Jan-09	21	24.61	18.92	21.93	17.19	17.4	20.01

Tabel diatas menunjukkan waktu *run* program untuk lima kali percobaan setiap harinya. Dapat dilihat bahwa semakin banyak jumlah konsumen, maka akan semakin lama waktu *run* program. Hal ini disebabkan program memerlukan waktu yang lebih lama untuk mencari kombinasi yang paling optimal. Permasalahan VRP merupakan permasalahan *nonpolynomial hard (NP-hard)*, dimana semakin besarnya permasalahan (semakin banyak jumlah konsumen), maka usaha perhitungan untuk menyelesaikan masalah akan semakin besar.

Hal lain yang menyebabkan adanya perbedaan waktu *run* program untuk jumlah konsumen yang berbeda adalah nilai dari parameter $c.n$ yang digunakan. Dengan adanya perbedaan jumlah konsumen maka nilai parameter $c.n$ ini juga akan berbeda, karena n itu sendiri melambangkan jumlah konsumen yang melakukan pemesanan. Seperti yang telah diketahui bahwa semakin tinggi nilai $c.n$, maka waktu *run* program juga akan meningkat, sehingga semakin banyak jumlah konsumen, maka waktu *run* program jelas akan menjadi lebih lama.

Secara umum, untuk menyelesaikan kasus VRP pada penelitian ini, program yang digunakan masih memerlukan waktu *run* yang relatif lama. Hal yang dapat dilakukan untuk mengurangi waktu *run* program secara umum adalah dengan membuat algoritma pemrograman yang lebih efisien.

Asumsi-asumsi dasar yang digunakan program ataupun input program dalam melakukan penentuan rute distribusi optimal untuk kasus di perusahaan agribisnis ini adalah sebagai berikut:

- Jarak antartitik yang diukur merupakan jarak sesuai dengan alur jalan dengan menggunakan bantuan peta digital *Google Maps*. Kemudian, diasumsikan jarak tempuh dari titik A ke titik B sama dengan jarak tempuh dari titik B ke titik A.

- Data kecepatan yang digunakan adalah kecepatan rata-rata yang berlaku konstan, yang digunakan untuk menghitung waktu tempuh.
- *Service time* dan batasan waktu (*time windows*) pada setiap titik sama.
- Hanya digunakan satu jenis kendaraan yang berangkat pada waktu yang sama.

Apabila program ini ingin digunakan pada kasus lain, maka perlu ada penyesuaian asumsi-asumsi tersebut, seperti:

- Jarak antartitik dapat diukur sesuai dengan keadaan aktual (jalur aktual yang dilalui) berdasarkan odometer kendaraan.
- Apabila diinginkan waktu tempuh antartitik yang lebih mendekati kondisi aktual, dapat dikumpulkan data waktu tempuh antartitik dengan menggunakan asumsi kecepatan yang berbead beda sesuai dengan kondisi aktual yang ada.
- Jika terdapat perbedaan batasan waktu dan *service time* disetiap titik, maka tidak menjadi masalah. Program ini dapat mengakomodir hal tersebut, karena *input* data batasan waktu dan *service time* berbentuk matriks yang disesuaikan dengan tiap-tiap konsumen.
- Penggunaan lebih dari satu jenis kendaraan yang berbeda kapasitasnya akan membutuhkan penyesuaian lagi untuk algoritma pada program, dikarenakan hal ini merupakan kelas VRP yang berbeda. Program ini hanya dapat mengakomodir adanya perubahan kapasitas maksimum kendaraan yang digunakan.

Untuk kedepannya, program ini dapat digunakan oleh pihak perusahaan untuk melakukan perencanaan distribusi harian. Untuk itu, semua data antartitik disimpan dalam suatu *spreadsheet excel* yang dapat dipanggil sesuai dengan konsumen yang melakukan pemesanan pada hari tertentu. Untuk memudahkan penggunaan program, maka dibuat suatu *form input* yang berbentuk *spreadsheet excel* yang dapat dengan mudah diisi oleh pengguna program (*user*). Program MATLAB dapat secara otomatis mengambil atau menarik data yang diinputkan pada form tersebut. Data yang dimasukkan kedalam *form* ini hanyalah kode nama konsumen dan jumlah permintaannya. Selain data tersebut, nilai untuk setiap variabel parameter juga dapat diisi pada *form input* lain (*sheet* lain namun masih

pada *file excel* yang sama). Hal ini akan memudahkan pengguna jika ingin mengubah konfigurasi parameter. Untuk variabel atau data-data lain seperti data kapasitas kendaraan, *time windows*, *service time*, kecepatan kendaraan, dan waktu keberangkatan dari depot tidak perlu lagi disertakan dalam *form input* karena data-data tersebut diasumsikan sama untuk setiap harinya. Data-data tersebut sudah tercakup dalam *source code* program. Berikut ini adalah gambaran dari tampilan *form input* tersebut.

FORMINPUT KONSUMEN DAN JUMLAH PERMINTAAN HARIAN		
Hari/Tanggal :	Senin, 5-Jan-2009	
Jumlah Customer :	14	
No.	Kode Customer	Jumlah Permintaan (krat)
1	CRL	10
2	DOP	5
3	GRC	6
4	GRG	2
5	GRK	12
6	GRL	12
7	GRM	1
8	GRP	12
9	GRS	2
10	JPG	13
11	MBP	7
12	MD	69
13	PRN	17
14	PZK	5

OK

Clear

FORMINPUT KONFIGURASI PARAMETER	
Parameter	Nilai
Jumlah Populasi Awal	10
Jumlah Solusi Tetangga	28
Panjang Tabu List	10
Iterasi Maksimum	1000
Stop Iterasi Maksimum	100

Gambar 4.3 Tampilan *Form Input*

Kelemahan lain dari program ini adalah tampilan solusi yang kurang menarik dan solusi rute yang harus diterjemahkan lebih lanjut. Solusi akhir yang

dapat dihasilkan oleh program ini meliputi urutan rute distribusi, jumlah kendaraan yang digunakan, kapasitas yang diangkut masing-masing kendaraan, jumlah konsumen di setiap rute, jarak tempuh tiap kendaraan dan jarak tempuh total, serta waktu *run* program. Berikut ini merupakan tampilan dari solusi akhir yang dihasilkan oleh program MATLAB.

```

-----Solusi Akhir-----
Solusi Rute :
  0  9  5  11  2  8  14  0  12  0  10  6  4  7  13  1  3  0
Jumlah Kendaraan yang Digunakan :
  3
Muatan Tiap Kendaraan (krat) :
  43  69  61
Jumlah Konsumen yang Dilayani Tiap Kendaraan :
  6  1  7
Jarak tempuh Tiap Kendaraan (km) :
  136.8200  141.7000  190.9200
Jarak Tempuh Total
  469.44
Elapsed time is 9.717203 seconds.

```

Gambar 4.4 Tampilan Solusi Akhir

Gambar diatas merupakan tampilan solusi untuk rute distribusi pada hari Senin tanggal 5 Januari 2009. Representasi solusi rute yang dihasilkan berupa deretan angka-angka. Angka 0 merepresentasikan depot dan angka lainnya merepresentasikan nomor urut dari konsumen pada *form input*.

Penulis menyadari bahwa masih banyak kekurangan dari program yang telah dibuat ini. Untuk penggunaannya di perusahaan, masih banyak yang harus diperbaiki, misalkan membuat tampilan solusi yang lebih menarik dan hasil solusi yang langsung menunjukkan kode konsumen pada urutan rute distribusi. Untuk melakukan hal tersebut diperlukan tambahan kode pemrograman yang lebih lanjut.

4.5.3 Analisis Hasil Optimasi

Pada bagian ini akan dianalisis mengenai hasil optimasi rute distribusi harian pada perusahaan agribisnis. Pada bagian ini akan dibandingkan rute

distribusi yang digunakan oleh perusahaan saat ini dengan rute distribusi usulan hasil optimasi. Analisis hasil yang akan dibahas meliputi analisis rute dan jarak tempuh, analisis utilitas kendaraan, dan analisis biaya pengiriman.

4.5.3.1 Analisis Rute Harian dan Jarak Tempuh

Secara umum, rute distribusi yang dihasilkan dari hasil optimasi dengan algoritma *Tabu Search* ini lebih baik dibandingkan dengan rute distribusi yang digunakan oleh pihak perusahaan saat ini. Tidak optimalnya rute distribusi yang digunakan oleh pihak perusahaan saat ini disebabkan karena penentuan rute distribusi tersebut hanya didasarkan pada pengalaman dari pengemudi kendaraan, meskipun pihak perusahaan telah melakukan perencanaan berupa pembagian rute-rute distribusi. Pembagian rute-rute tersebut juga masih kurang optimal, karena hanya terpaku pada pembagian wilayah saja tanpa memperhatikan optimalitas dari utilisasi kendaraan.

Berikut ini akan dibahas mengenai perbandingan rute antara rute usulan hasil optimasi dengan rute yang saat ini digunakan oleh perusahaan dari tanggal 5 Januari 2009 hingga tanggal 11 Januari 2009. Pada pengiriman hari Senin tanggal 5 Januari 2009, terjadi pengurangan jarak tempuh yang dihasilkan oleh rute usulan hasil optimasi. Berikut ini dapat dilihat perbandingan antara rute saat ini yang digunakan oleh perusahaan dan rute usulan.

Tabel 4.25 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 5 Januari 2009

Rute	Titik Kirim								Jarak Rute (km)	Jarak Total (km)
	Rute Usulan	Rute 1	GRS	GRK	MBP	DOP	GRP	PZK		136.82
Rute 2		JPG	GRL	GRG	GRM	PRN	CRL	GRC	190.92	
Rute 3		MD							141.70	
Rute Saat Ini	Rute 1	GRS	GRG	GRM	MD				153.00	485.68
	Rute 2	GRC	CRL	DOP	GRP	GRK	MBP	PZK	136.48	
	Rute 3	JPG	GRL	PRN					196.20	

Rute distribusi usulan memiliki jarak tempuh total 469,44 km. Sementara itu rute saat ini yang digunakan perusahaan memiliki jarak tempuh total 485,68 km. Pengurangan jarak yang terjadi adalah sebesar 16,24 km atau sebesar 3,34%. Pengurangan jarak sebesar ini dapat dikatakan relatif kecil. Kecilnya pengurangan

jarak ini disebabkan oleh jumlah konsumen yang relatif sedikit, yaitu 14 konsumen, sehingga kombinasi alternatif solusi yang ada juga sedikit dan besarnya perbaikan jarak juga kecil. Selain itu, tidak ada pengurangan rute yang dihasilkan oleh rute usulan. Jumlah rute dan jumlah kendaraan yang digunakan pada rute saat ini dan rute usulan sama, yaitu adalah tiga buah.

Pada pengiriman hari Selasa, rute distribusi usulan memiliki jarak tempuh total 854,08 km dan rute saat ini memiliki jarak tempuh total 1037,81 km. Berikut ini merupakan perbandingan antara rute saat ini dan rute usulan pada hari Selasa tanggal 6 Januari 2009.

Tabel 4.26 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 6 Januari 2009

Rute		Titik Kirim									Jarak Rute (km)	Jarak Total (km)
Rute Usulan	Rute 1	PZK	PZS	PZO	PZC	GRM	GRG	GRS	CRH	MPL	157.89	854.08
	Rute 2	MPS	MPG	CRL	GRC	MPC	DMF	SMP			170.78	
	Rute 3	MPB	MPR	MPE							179.75	
	Rute 4	FMG	JPG	MD	DOG	DMA					141.34	
	Rute 5	MPN	MDLJ	MPH	FMS	MPK					204.32	
Rute Saat Ini	Rute 1	MPN	MPH	MPK	FMS	MPG	MPS				200.40	1037.81
	Rute 2	MPE	MPB	CRH	PZS	PZO					135.71	
	Rute 3	MD	GRM	GRG	GRS	GRC	MPC				157.12	
	Rute 4	CRL	DMF	PZK	PZC	SMP					150.86	
	Rute 5	MDLJ	DMA	FMG	DOG	JPG	MPL				220.76	
	Rute 6	MPR									172.96	

Dapat dilihat pada tabel diatas, terjadi penurunan jumlah rute yang digunakan pada rute usulan, yaitu dari 6 rute menjadi 5 rute. Dengan adanya pengurangan jumlah rute tersebut, maka besarnya pengurangan jarak tempuh yang dihasilkan oleh rute usulan menjadi relatif besar. Pengurangan jarak yang dihasilkan adalah sebesar 183,73 km atau sebesar 17,7%.

Dapat dilihat pada rute saat ini, rute 6 hanya mengantarkan pesanan ke satu titik konsumen saja, yaitu MPR. Hal tersebut dikarenakan jumlah permintaan MPR relatif besar, yaitu 30 krat dan tidak bisa dimasukkan ke rute lain karena akan mengakibatkan kelebihan kapasitas kendaraan. Namun, pada rute usulan, kombinasi alokasi rute konsumen diubah sehingga konsumen MPR dapat digabung dengan konsumen lain, yaitu MPB dan MPE. Kombinasi konsumen pada rute-rute lain juga berubah sedemikian rupa tanpa melebihi kapasitas maksimal kendaraan, sehingga dapat hanya menggunakan 5 rute saja untuk

melayani semua konsumen yang ada. Untuk kasus seperti ini, optimalisasi kapasitas kendaraan berperan sangat penting.

Kemudian, berikut ini adalah perbandingan antara rute usulan dan rute saat ini pada hari Rabu tanggal 7 Januari 2009.

Tabel 4.27 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 7 Januari 2009

Rute		Titik Kirim									Jarak Rute (km)	Jarak Total (km)
Rute Usulan	Rute 1	PZK	MBP	PRN	FMS	MPK	MPS	CRB	DMF	GRC	208.16	893.43
	Rute 2	FMG	MPQ	GRM	PZM	GRG	GNL	GRS	RCJ	153.38		
	Rute 3	MD	JPG	GRL	DMA	MPE				148.79		
	Rute 4	MPO	CRL	PY	CRH	MPL	MPR			221.86		
	Rute 5	SPI								161.24		
Rute Saat Ini	Rute 1	MPE	CRB	MPS	FMS	MPK	PRN				208.13	919.70
	Rute 2	MD	MPQ	PZM	GRM	GRG	GRS	MBP	MPR		234.14	
	Rute 3	MPO	CRL	GRC	DMF	PY	GNL	PZK	RCJ		144.85	
	Rute 4	MPL	JPG	FMG	GRL	DMA					162.87	
	Rute 5	CRH	SPI								169.71	

Dapat dilihat pada tabel diatas, rute saat ini menghasilkan jarak tempuh perjalanan 919,7 km, sedangkan rute usulan menghasilkan jarak 892,43 km. Penurunan jarak yang dihasilkan adalah sebesar 26,27 km atau sekitar 2,86%. Jika dilihat dari persentase penurunan jarak, penurunan jarak yang dihasilkan tersebut relatif kecil. Banyaknya rute yang digunakan pada rute pengiriman lama dan rute usulan sama, yang berbeda adalah kombinasi alokasi konsumen di setiap rutenya.

Dapat dilihat rute 5 pada rute usulan hanya melakukan pengiriman ke satu titik konsumen, yaitu SPI. Hal tersebut dikarenakan SPI memiliki jumlah permintaan yang sangat besar, yaitu 65 krat sehingga konsumen ini tidak dapat dimasukkan ke rute-rute lain. Pengurangan jumlah rute tidak dapat dilakukan disini, dikarenakan jumlah total permintaan yang harus dikirim, yaitu 335 krat, melebihi jumlah kapasitas maksimum dari 4 buah kendaraan (300 krat).

Jika diperhatikan lebih lanjut, untuk rute saat ini dan rute usulan, jumlah konsumen yang harus dilayani jumlahnya berbeda-beda pada setiap rute, ada yang sangat sedikit dan ada yang sangat banyak. Dapat dilihat misalnya pada rute 1 pada rute usulan, jumlah konsumen yang harus dilayani oleh kendaraan di rute 1 sangat banyak, yaitu mencapai 10 konsumen. Namun, pada rute 5, jumlah konsumen yang dilayani hanya satu titik. Ketidakseimbangan jumlah konsumen per rute ini dapat terjadi dikarenakan jumlah permintaan konsumen yang berbeda-

beda. Konsumen-konsumen pada rute 1 memiliki jumlah permintaan yang kecil, sehingga kesepuluh konsumen tersebut dapat disatukan dalam satu rute. Total jumlah permintaan kesepuluh konsumen tersebut adalah 75 krat. Jumlah tersebut sama dengan kapasitas maksimum satu kendaraan, sehingga utilitas kendaraan pada rute 1 mencapai 100%. Selain itu, waktu pelayanan pada setiap pelanggan masih dalam rentang waktu (*time windows*) yang ada. Hal yang dapat diambil dari kondisi ini adalah, bahwasanya ketidakseimbangan jumlah konsumen pada rute-rute tiap harinya bukan merupakan suatu masalah, selama kendala kapasitas kendaraan dan batasan waktu masih dapat dipenuhi.

Berikutnya, masuk pada hari Kamis tanggal 8 Januari 2009. berikut ini adalah perbandingan antara rute usulan dan rute saat ini pada hari Kamis tanggal 8 Januari 2009.

Tabel 4.28 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 8 Januari 2009

Rute		Titik Kirim											Jarak Rute (km)	Jarak Total (km)	
Rute Usulan	Rute 1	CRB	MPS	FMS	PRN	MPK	CRP	RCM	GRP					213.84	662.43
	Rute 2	MD	JPG											141.70	
	Rute 3	CRL	MPC	GRC	DMF	DOK	PZK	MPE						129.98	
	Rute 4	CRA	MBP	GRS	GRG	GRM	DMA	GRL	FMG	MPL	CRH			176.91	
Rute Saat Ini	Rute 1	RCM	CRP	PRN	MPK	FMS	MPS	CRB						222.05	810.15
	Rute 2	MPE	CRH	CRA	MBP									136.40	
	Rute 3	MD	GRM	GRG	GRS	GRP	DOK							164.11	
	Rute 4	CRL	MPC	GRC	DMF	PZK								124.72	
	Rute 5	MPL	JPG	FMG	GRL	DMA								162.87	

Dapat dilihat pada tabel diatas, rute saat ini menghasilkan total jarak tempuh perjalanan 810,15 km, sedangkan rute usulan menghasilkan total jarak tempuh 662,43 km. Penurunan jarak yang dihasilkan cukup besar, yaitu 147,72 km atau sekitar 18,23%. Hal ini disebabkan adanya pengurangan jumlah rute yang digunakan untuk melakukan pengantaran, yaitu dari 5 rute, pada rute saat ini, menjadi 4 rute, pada rute usulan.

Jika dilihat sepintas pada rute saat ini, sangatlah kecil untuk melakukan pengurangan jumlah rute, mengingat jumlah kosumen pada rute terkecil (rute 2) adalah 4 konsumen. Namun, demikian dengan adanya perubahan kombinasi alokasi konsumen pada rute-rute distribusi, maka pengurangan jumlah rute dapat dilakukan. Perubahan alokasi konsumen pada rute-rute distribusi ini dapat

dilakukan oleh program, yaitu dengan menggunakan mekanisme pembentukan solusi tetangga (*insertion, 2-opt, dan interchange*).

Kemudian, berikut ini adalah perbandingan antara rute usulan dan rute saat ini pada hari Jumat tanggal 9 Januari 2009.

Tabel 4.29 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 9 Januari 2009

Rute		Titik Kirim									Jarak Rute (km)	Jarak Total (km)
Rute Usulan	Rute 1	MD	JPG	FMG	DMA						142.41	973.97
	Rute 2	CRB	MPS	MPH	MPK	MDLJ	HHK	DOP	GRP		205.23	
	Rute 3	HHR	HHD	CRN	MPN	CRH	MPL				186.34	
	Rute 4	MPE	MPO	HHF	KS	MPC	GRC	CRL			140.07	
	Rute 5	DMF	GRK	MBP	GRS	GRG	GRM	PZM	SRK	HHL	151.40	
	Rute 6	HHB	MPI	CRQ	HHC						148.52	
Rute Saat Ini	Rute 1	HHD	CRN	MPN	HHK	MPH	MPK	MPS	CRB		208.20	1045.05
	Rute 2	MPE	CRH	PZM	SRK	HHL	HHR				153.64	
	Rute 3	MD	GRM	GRG	GRS	GRK	MBP	DOP	GRP		160.90	
	Rute 4	MPO	CRL	GRC	MPC	DMF	KS	HHF			149.25	
	Rute 5	JPG	FMG	DMA	MDLJ						198.64	
	Rute 6	MPL	CRQ	HHC	HHB	MPI					174.42	

Untuk kasus distribusi pada hari Jumat ini, rute saat ini menghasilkan total jarak tempuh perjalanan 1045,05 km, sedangkan rute usulan menghasilkan total jarak tempuh 973,97 km. Penurunan jarak yang dihasilkan adalah sebesar 71,08 km atau sekitar 6,8%. Penurunan jarak tempuh sebesar 71,08 km ini relatif besar dibandingkan penurunan jarak tempuh pada hari-hari sebelumnya yang tidak terjadi pengurangan rute. Hal tersebut dikarenakan jumlah konsumen yang harus dilayani pada hari Jumat ini sangat banyak, yaitu 38 konsumen. Dengan demikian, alternatif solusi optimalnya akan bertambah besar dan kemungkinan untuk mendapatkan solusi rute yang jauh lebih baik dari rute saat ini juga besar.

Pada hari ini keenam armada kendaraan yang tersedia harus digunakan, karena rute usulan tidak dapat menghasilkan pengurangan jumlah rute. Jika dilihat dari total jumlah permintaan yang dipesan oleh semua konsumen, yaitu sebanyak 391 krat, memang tidak memungkinkan untuk melakukan pengiriman dengan menggunakan 5 kendaraan karena jumlah permintaan tersebut melebihi jumlah kapasitas dari 5 mobil (375 krat).

Kemudian, berikut ini adalah perbandingan antara rute usulan dan rute saat ini pada hari Sabtu tanggal 10 Januari 2009.

Tabel 4.30 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 10 Januari 2009

Rute		Titik Kirim								Jarak Rute (km)	Jarak Total (km)
Rute Usulan	Rute 1	FMG	JPG	MD	GRL	DMA				142.42	783.27
	Rute 2	CRB	FMS	MPK	GRM	PZG	GRG	PZC		188.82	
	Rute 3	MPB	DMF	GRP	RCD	MBP	GNL	GRS	CRA	135.81	
	Rute 4	CRE	MPL	CRH	SPI					186.54	
	Rute 5	MPE	MPC	GRC	CRL	DOK	RCJ			129.68	
Rute Saat Ini	Rute 1	MPE	MPK	FMS	CRB					190.13	945.42
	Rute 2	CRH	CRA	PZC	GNL	MBP	DOK			135.83	
	Rute 3	MD	GRM	PZG	GRG	GRS	GRP			160.63	
	Rute 4	CRL	MPC	GRC	DMF	RCD	RCJ			129.56	
	Rute 5	MPL	JPG	FMG	GRL	DMA				162.87	
	Rute 6	MPB	CRE	SPI						166.40	

Dapat dilihat pada tabel diatas, rute saat ini menghasilkan total jarak tempuh 945,42 km, sedangkan rute usulan menghasilkan total jarak tempuh 783,27 km. Penurunan jarak yang dihasilkan adalah sebesar 162,15 km atau sekitar 17,15%. Kasus distribusi pada hari Sabtu ini sama seperti pada kasus distribusi hari Selasa dan Kamis, dimana terjadi pengurangan jumlah rute yang mengakibatkan penurunan jarak yang dihasilkan menjadi relatif besar.

Selanjutnya, berikut ini merupakan perbandingan antara rute distribusi usulan dan rute distribusi yang lama pada hari Minggu tanggal 11 Januari 2009.

Tabel 4.31 Perbandingan Rute Saat Ini dan Rute Usulan Tanggal 11 Januari 2009

Rute		Titik Kirim								Jarak Rute (km)	Jarak Total (km)
Rute Usulan	Rute 1	MPE	MPV	DMF	GRC	MPC	CRL			123.84	662.41
	Rute 2	MBP	GNL	GRS	GRM	MPN	FMS	MPK	MPS	208.52	
	Rute 3	MPL	FMG	GRL	MD	DMA				164.89	
	Rute 4	SPI	CRE							165.16	
Rute Saat Ini	Rute 1	MPE	MPV	MPN	MPK	FMS	MPS			203.18	719.11
	Rute 2	MD	GRM	GRS	MBP	SPI				217.34	
	Rute 3	CRE	CRL	GRC	MPC	DMF	GNL			139.72	
	Rute 4	MPL	FMG	GRL	DMA					158.87	

Untuk kasus distribusi hari Minggu ini, rute saat ini menghasilkan total jarak tempuh 719,11 km, sedangkan rute usulan menghasilkan total jarak tempuh 662,41 km. Penurunan jarak yang dihasilkan adalah sebesar 56,7 km atau sekitar 7,88%. Rute usulan tidak mengurangi jumlah rute yang digunakan pada rute saat ini. Perbedaan antara rute usulan dengan rute saat ini hanya kombinasi alokasi titik konsumen pada rute saja yang berubah. Kombinasi alokasi titik konsumen

pada rute usulan lebih baik atau lebih optimal daripada rute saat ini, sehingga rute usulan dapat memberikan jarak tempuh total yang lebih rendah.

Jika diamati lebih lanjut, adanya pengurangan jumlah rute pengiriman, seperti pada hari Selasa, Kamis, dan Sabtu, akan mengakibatkan pengurangan total jarak tempuh yang besar, yaitu lebih dari 140 km. Hal ini dikarenakan, jarak antara lokasi depot dan rata-rata lokasi konsumen sangat jauh, yaitu sekitar 60-70 km. Oleh karenanya pengurangan satu rute atau satu kendaraan saja akan dapat mengurangi total jarak tempuh kira-kira sebesar dua kali jarak tempuh dari depot ke rata-rata lokasi konsumen. Dari hal tersebut dapat disimpulkan bahwa pada kasus distribusi produk di perusahaan agribisnis ini, penggunaan rute atau jumlah kendaraan yang paling minimal sangatlah penting karena akan dapat meminimalkan jarak tempuh total.

Jika dilihat secara umum, total jarak tempuh selama satu minggu pada penggunaan rute saat ini adalah sebesar 5.962,92 km, sedangkan untuk rute usulan adalah sebesar 5.299,03 km. Terjadi penurunan jarak tempuh, yaitu sebesar 663,89 km atau sebesar 11,13 %. Perbaikan jarak sebesar 11,13% ini masih dapat ditingkatkan dengan melakukan lebih banyak eksperimen dalam menggunakan mekanisme pencarian solusi tetangga. Selain itu, penggunaan konfigurasi parameter yang lebih baik akan dapat meningkatkan kualitas solusi yang dihasilkan.

4.5.3.3 Analisis Utilitas Kendaraan

Seperti yang telah diketahui, optimasi utilitas kendaraan sangat penting peranannya dalam merencanakan rute distribusi pengiriman produk. Dengan adanya utilisasi kendaraan yang optimal, maka akan didapatkan suatu penentuan rute distribusi yang lebih baik, yaitu akan dapat menghasilkan jumlah rute yang seminimal mungkin.

Untuk lebih jelasnya, berikut ini merupakan perbandingan utilitas kendaraan antara rute yang digunakan perusahaan saat ini dengan rute usulan yang dihasilkan dari optimasi dengan algoritma *Tabu Search*.

Tabel 4.32 Perbandingan Utilitas Kendaraan Rute Saat Ini dan Rute Usulan

Hari / Tanggal	Rute	Rute Saat Ini		Rute Usulan	
		Jumlah Pengiriman (krat)	Utilitas Kendaraan	Jumlah Pengiriman (krat)	Utilitas Kendaraan
Senin 5-Jan-09	Rute 1	74	98.67%	43	57.33%
	Rute 2	57	76.00%	61	81.33%
	Rute 3	42	56.00%	69	92.00%
Selasa 6-Jan-09	Rute 1	62	82.67%	67	89.33%
	Rute 2	59	78.67%	69	92.00%
	Rute 3	62	82.67%	58	77.33%
	Rute 4	55	73.33%	74	98.67%
	Rute 5	67	89.33%	67	89.33%
	Rute 6	30	40.00%		
Rabu 7-Jan-09	Rute 1	61	81.33%	75	100.00%
	Rute 2	70	93.33%	54	72.00%
	Rute 3	66	88.00%	72	96.00%
	Rute 4	68	90.67%	70	93.33%
	Rute 5	70	93.33%	65	86.67%
Kamis 8-Jan-09	Rute 1	67	89.33%	71	94.67%
	Rute 2	54	72.00%	73	97.33%
	Rute 3	67	89.33%	67	89.33%
	Rute 4	48	64.00%	73	97.33%
	Rute 5	48	64.00%		
Jumat 9-Jan-09	Rute 1	63	84.00%	75	100.00%
	Rute 2	67	89.33%	71	94.67%
	Rute 3	74	98.67%	59	78.67%
	Rute 4	69	92.00%	73	97.33%
	Rute 5	72	96.00%	73	97.33%
	Rute 6	46	61.33%	40	53.33%
Sabtu 10-Jan-09	Rute 1	43	57.33%	74	98.67%
	Rute 2	54	72.00%	71	94.67%
	Rute 3	47	62.67%	71	94.67%
	Rute 4	71	94.67%	60	80.00%
	Rute 5	73	97.33%	65	86.67%
	Rute 6	53	70.67%		
Minggu 11-Jan-09	Rute 1	74	98.67%	64	85.33%
	Rute 2	71	94.67%	64	85.33%
	Rute 3	55	73.33%	66	88.00%
	Rute 4	62	82.67%	68	90.67%

Dapat dilihat pada tabel tersebut, secara umum utilitas kendaraan pada rute usulan lebih optimal dan seimbang dibandingkan dengan utilitas kendaraan pada rute saat ini. Bahkan, pada sejumlah rute usulan terdapat kendaraan yang utilitasnya 100% atau dapat dikatakan memuat pesanan sejumlah 75 krat. Jika dihitung, utilitas rata-rata untuk satu kendaraan selama satu minggu pada penggunaan rute usulan adalah sebesar 88,42%, sedangkan pada rute yang digunakan perusahaan saat ini sebesar 80,8%. Dengan demikian, rute usulan dapat meningkatkan rata-rata utilitas kendaraan sebesar 7,62%. Hal tersebut sangatlah

baik, karena dengan adanya utilitas kendaraan yang optimal, maka peluang untuk melakukan pengurangan rute akan semakin besar.

Jika melihat kondisi saat ini, pihak perusahaan dalam melakukan pembagian rute-rute distribusi harian hanya terpaku pada pembagian wilayah konsumen yang telah mereka buat. Pihak perusahaan kurang mempertimbangkan optimasi utilitas kendaraan dan peluang-peluang untuk dapat membuat rute distribusi yang lebih baik. Untuk itu, perusahaan dapat menggunakan program penyelesaian masalah VRP dengan algoritma *Tabu Search* yang telah dibuat untuk melakukan perencanaan rute distribusi harian.

4.5.3.4 Analisis Biaya Pengiriman

Biaya pengiriman ini merupakan biaya yang harus dikeluarkan untuk melakukan setiap distribusi pengiriman produk kepada konsumen. Biaya pengiriman ini terdiri dari biaya bahan bakar kendaraan, biaya perawatan kendaraan, biaya ban dan biaya supir. Rincian biaya per kilometer untuk setiap elemen biaya telah dijelaskan pada bab tiga. Tiga dari empat elemen biaya, yaitu biaya bahan bakar, biaya perawatan kendaraan, dan biaya ban merupakan *variable cost* yang berbanding lurus dengan jarak tempuh kendaraan, dimana semakin besar jarak tempuh kendaraan, maka biayanya juga akan semakin besar. Sedangkan biaya supir dihitung berdasarkan jumlah pengantaran yang dilakukan

Berikut ini merupakan perbandingan antara biaya pengiriman pada rute yang saat ini digunakan oleh pihak perusahaan dan rute usulan hasil optimasi.

Tabel 4.33 Perbandingan Biaya Pengiriman Rute Saat Ini dan Rute Usulan

Hari/ Tanggal	Biaya Pengiriman	
	Rute Saat Ini	Rute Usulan
Senin / 5-Jan-09	Rp827,184.92	Rp807,049.27
Selasa / 6-Jan-09	Rp1,736,759.86	Rp1,433,956.71
Rabu / 7-Jan-09	Rp1,515,317.64	Rp1,482,745.99
Kamis / 8-Jan-09	Rp1,379,488.78	Rp1,121,333.71
Jumat / 9-Jan-09	Rp1,745,736.59	Rp1,657,605.92
Sabtu / 10-Jan-09	Rp1,622,207.35	Rp1,346,160.81
Minggu / 11-Jan-09	Rp1,191,610.11	Rp1,121,308.91
Total	Rp10,018,305.25	Rp8,970,161.32

Dapat dilihat pada tabel 4.33, untuk aktivitas distribusi produk selama satu minggu, biaya transportasi pengiriman yang dikeluarkan pada penggunaan rute saat ini adalah sebesar Rp.10.018.305,25, sedangkan untuk rute usulan sebesar Rp.8.970.161,32. Dengan demikian, rute usulan ini berhasil mengurangi biaya pengiriman sebesar Rp.1.048.143,93 untuk satu minggu atau sebesar 10,46%.



BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil studi kasus optimasi rute distribusi pengiriman produk harian di perusahaan agribisnis ini dengan menggunakan algoritma *Tabu Search* dan bantuan bahasa pemrograman MATLAB, diperoleh kesimpulan sebagai berikut.

- Penentuan rute distribusi harian dengan menggunakan algoritma *Tabu Search* ini telah menghasilkan rute distribusi usulan yang optimal dan lebih baik daripada rute yang selama ini digunakan oleh perusahaan. Total jarak tempuh selama satu minggu pada penggunaan rute usulan ini adalah 5.299,03 km dengan penurunan jarak tempuh sebesar 663,89 km atau sebesar 11,13 %.
- Biaya pengiriman pada rute usulan selama satu minggu adalah sebesar Rp.8.970.161,32. Penurunan biaya transportasi pengiriman yang dihasilkan adalah sebesar Rp.1.048.143,93 atau sebesar 10,46%.
- Utilitas rata-rata untuk satu kendaraan selama satu minggu pada penggunaan rute usulan adalah sebesar 88,42%, sedangkan pada rute yang digunakan perusahaan sebesar 80,8%. Dengan demikian, rute usulan dapat meningkatkan rata-rata utilitas kendaraan sebesar 7,62%.
- Penentuan konfigurasi parameter yang tepat akan menghasilkan solusi dengan kualitas yang baik dan waktu *run* program yang cepat. Studi parameter merupakan hal yang penting dalam pengujian metode ini, karena setiap kasus distribusi membutuhkan konfigurasi parameter yang unik untuk memperoleh hasil rute yang stabil.
- Selain penentuan parameter, waktu *run* program juga bergantung pada banyaknya titik konsumen yang harus dilayani. Semakin banyak jumlahnya, maka akan semakin lama waktu *run* program untuk menghasilkan solusi rute. Waktu *run* program untuk menyelesaikan masalah distribusi harian selama satu minggu pada penelitian ini berkisar antara 9 detik sampai 80 detik.

5.2 Saran

Berikut ini adalah saran untuk penelitian selanjutnya.

- Untuk penelitian yang akan datang akan lebih baik jika menggunakan bahasa pemrograman yang lebih maju dari yang sekarang.
- Untuk penggunaan algoritma *Tabu Search*, perlu dilakukan studi parameter yang lebih mendalam dan juga penggunaan mekanisme pembentukan solusi tetangga yang lebih variatif.
- Program yang dibuat hendaknya dapat mengakomodir kepentingan pengguna, seperti *user interface* yang lebih baik dan cara penggunaan yang lebih sederhana. Disamping itu, akan lebih baik jika program yang dibuat dapat terintegrasi pada sistem *database* yang berkaitan, seperti sistem pemesanan dan sistem keuangan.

Kemudian, berikut ini adalah saran dan usulan bagi pihak perusahaan dalam merencanakan rute distribusi produk harian.

- Dalam merencanakan rute distribusi harian, hendaknya pihak perusahaan memperhatikan utilitas dari tiap kendaraan yang digunakan, sehingga nantinya akan dapat menghasilkan rute distribusi yang lebih efisien.
- Untuk melakukan perencanaan rute distribusi harian, pihak perusahaan dapat menggunakan program yang telah dibuat pada penelitian ini.

DAFTAR REFERENSI

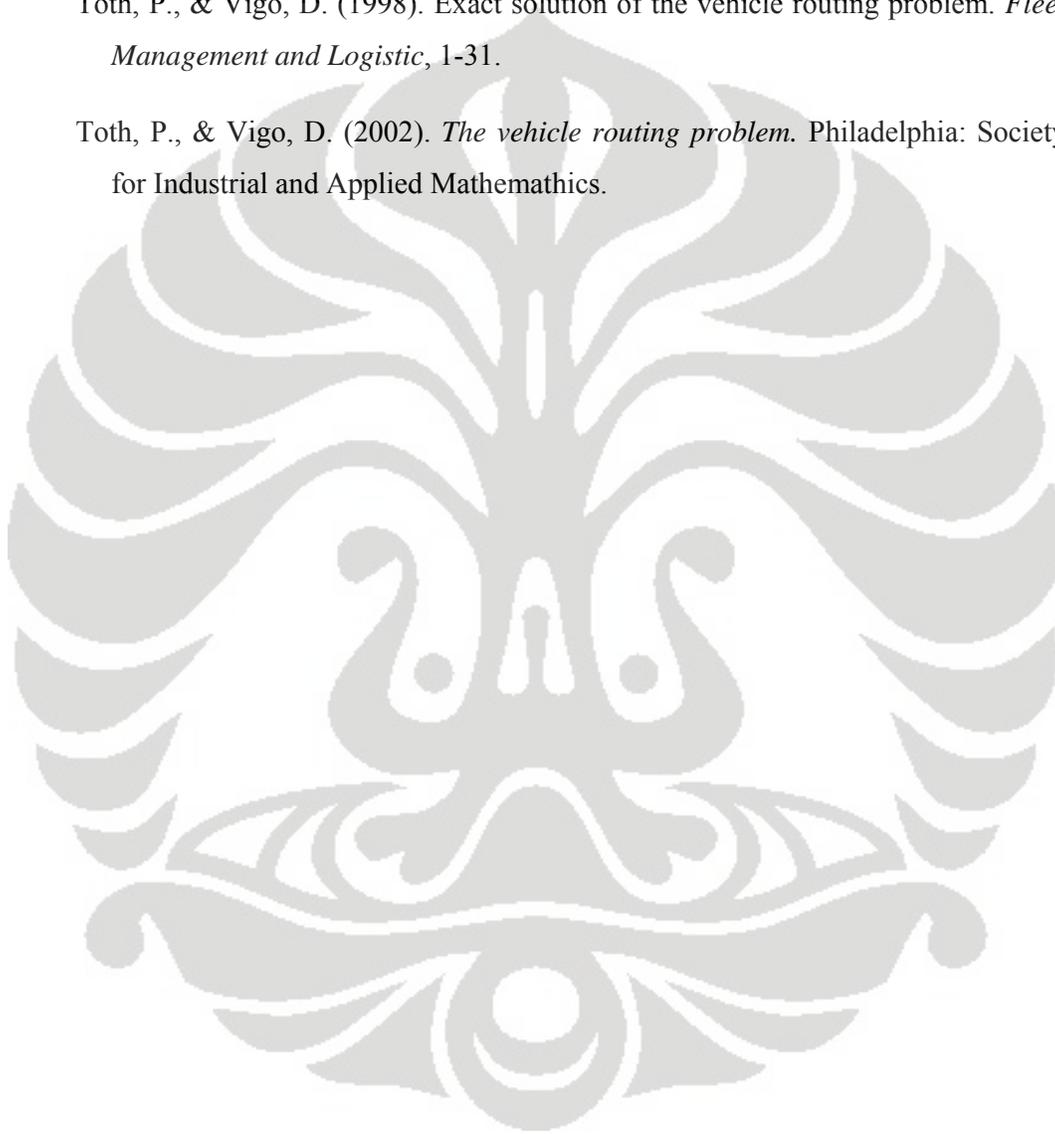
- Ballou, R.H. (2004). *Business logistics management* (5th ed). New Jersey: Prentice-Hall Inc.
- Ballou, R.H., & Agarwal, Y.K. (1998). A performance comparison of several popular algorithms for vehicle routing and scheduling. *Journal of Business Logistics*, Vol. 9 No.1, 51 - 65.
- Braysy, O., & Gendreau, M. (2002). Tabu search heuristics for the vehicle routing problem with time windows. *Journal of SINTEF Applied Mathematics*, 211-237.
- Cordeau, J.F., & Laporte, G. (2002). Tabu search heuristic for vehicle routing problem. *Canada Research Chair in Distribution Management*, 1-26
- Dreo, J., Petworsky, A., & Taillard, E.D. (2006). *Metaheuristics for hard optimization*. Berlin: Springer-Verlag Berlin Heidelberg.
- Glover, F., & Laguna, M. (1997). *Tabu search*. Kluwer, Boston
www.geocities.com/francorbusetti/laguna.pdf, (last updated January 6,2002)
- Hertz, A., Taillard, E.D., De Werra, D. (1997). A tutorial on tabu search. *EPFL, Department of Mathematics*, 1-13.
- Homberger, J., & Gehring, H. (2004). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operation Research*, 220 - 238.
- Osman, I.H. (1995). An introduction to meta-heuristic. In: Lawrence M and Wildson C (eds). *Journal of Operational Research Society*, 92 -122
- Poot, A., Kant, G., Wagelmans, A.P.M. (2002). A saving based method for real life vehicle routing problem. *Journal of The Operational Research Society*, hal. 57 – 68.

Tan, K.C., Lee, L.H., Zhu, K.Q. (1999). Heuristik methods for vehicle routing problem with time windows. *National University of Singapore*, 1-34

Taillard, et al. (1998) Tabu search heuristic for the vehicle routing problem with time windows. *Universite de Motreal*, 1-36

Toth, P., & Vigo, D. (1998). Exact solution of the vehicle routing problem. *Fleet Management and Logistic*, 1-31.

Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia: Society for Industrial and Applied Mathematics.



LAMPIRAN



Lampiran 1 Matriks Jarak Antar Konsumen

Matriks Jarak (km)	O	CMS	CRA	CRB	CRC	CRE	CRH	CRL	CRN	CRP	CRQ	DMA	DMF	DOG	DOK	DOP	FMG	FMS	GNL
O		62.98	61.94	78.50	88.20	48.68	56.15	59.20	79.16	73.82	63.83	65.55	58.59	64.99	57.80	61.85	67.29	87.07	64.73
CMS	62.98		13.10	38.72	39.30	14.00	6.83	20.51	28.86	28.13	22.87	20.86	17.72	19.88	16.05	17.47	22.18	39.46	14.76
CRA	61.94	13.10		30.04	28.22	13.39	6.26	12.34	19.21	14.76	22.19	14.75	9.22	14.01	7.56	9.33	16.31	27.88	3.73
CRB	78.50	38.72	30.04		10.26	34.64	31.89	17.88	27.37	20.28	47.11	45.54	18.49	45.31	26.56	20.75	47.61	10.13	22.68
CRC	88.20	39.30	28.22	10.26		39.76	32.47	26.55	18.11	14.57	48.37	35.03	30.66	34.25	33.64	26.64	36.55	3.57	22.74
CRE	48.68	14.00	13.39	34.64	39.76		7.16	14.87	31.73	26.25	18.99	17.42	10.44	16.54	12.75	17.61	18.84	39.29	14.96
CRH	56.15	6.83	6.26	31.89	32.47	7.16		13.68	22.02	21.29	16.04	14.03	10.88	13.05	9.21	10.64	15.35	32.63	7.92
CRL	59.20	20.51	12.34	17.88	26.55	14.87	13.68		28.14	40.26	33.55	23.64	4.59	22.69	6.53	3.03	24.99	30.01	9.26
CRN	79.16	28.86	19.21	27.37	18.11	31.73	22.02	28.14		9.67	40.46	22.25	20.57	23.18	25.89	21.54	25.48	19.45	19.46
CRP	73.82	28.13	14.76	20.28	14.57	26.25	21.29	40.26	9.67		34.53	27.27	35.74	27.78	20.38	19.57	30.08	13.97	13.50
CRQ	63.83	22.87	22.19	47.11	48.37	18.99	16.04	33.55	40.46	34.53		25.83	29.33	16.19	24.55	29.39	18.49	47.84	23.27
DMA	65.55	20.86	14.75	45.54	35.03	17.42	14.03	23.64	22.25	27.27	25.83		27.47	3.96	22.95	21.32	3.56	41.60	21.60
DMF	58.59	17.72	9.22	18.49	30.66	10.44	10.88	4.59	20.57	35.74	29.33	27.47		26.05	3.12	3.56	28.35	38.35	6.85
DOG	64.99	19.88	14.01	45.31	34.25	16.54	13.05	22.69	23.18	27.78	16.19	3.96	26.05		22.24	23.57	2.30	41.37	21.00
DOK	57.80	16.05	7.56	26.56	33.64	12.75	9.21	6.53	25.89	20.38	24.55	22.95	3.12	22.24		8.11	24.54	33.52	7.91
DOP	61.85	17.47	9.33	20.75	26.64	17.61	10.64	3.03	21.54	19.57	29.39	21.32	3.56	23.57	8.11		25.87	26.53	6.40
FMG	67.29	22.18	16.31	47.61	36.55	18.84	15.35	24.99	25.48	30.08	18.49	3.56	28.35	2.30	24.54	25.87		43.67	23.30
FMS	87.07	39.46	27.88	10.13	3.57	39.29	32.63	30.01	19.45	13.97	47.84	41.60	38.35	41.37	33.52	26.53	43.67		22.57
GNL	64.73	14.76	3.73	22.68	22.74	14.96	7.92	9.26	19.46	13.50	23.27	21.60	6.85	21.00	7.91	6.40	23.30	22.57	
GRC	56.69	19.62	11.13	20.39	32.56	12.34	12.79	2.69	22.47	24.77	31.23	29.37	1.90	27.95	5.02	5.47	30.25	32.01	8.75
GRG	65.40	16.68	3.76	30.95	27.39	17.11	9.85	13.27	17.34	12.79	25.38	11.34	10.83	10.18	11.34	10.70	12.48	26.12	4.53
GRK	64.48	15.77	4.25	28.43	26.46	15.83	8.94	8.89	19.33	13.79	24.54	16.88	6.24	21.91	8.63	5.90	24.21	25.98	2.17
GRL	67.85	22.74	16.87	48.17	37.11	19.40	15.91	25.55	26.04	30.64	19.05	4.12	28.91	2.30	25.10	26.43	0.56	44.23	23.86
GRM	66.96	18.29	5.64	31.93	26.41	18.81	11.46	14.15	16.34	12.41	26.40	10.24	11.80	9.27	12.93	11.62	11.57	23.99	6.16

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	GRC	GRG	GRK	GRL	GRM	GRP	GRS	HHB	HHC	HHD	HHF	HHK	HHL	HHR	JPG	KB	KS	MBP
O	56.69	65.40	64.48	67.85	66.96	61.85	62.43	67.64	64.47	82.23	63.17	79.77	71.84	71.70	69.84	63.07	61.46	64.48
CMS	19.62	16.68	15.77	22.74	18.29	17.47	13.44	26.68	23.15	31.93	22.30	31.03	23.12	22.98	24.74	26.00	20.59	15.77
CRA	11.13	3.76	4.25	16.87	5.64	9.33	2.74	26.01	19.98	22.28	13.81	19.91	10.20	10.06	18.87	17.50	12.10	4.25
CRB	20.39	30.95	28.43	48.17	31.93	20.75	28.15	50.92	48.13	30.44	23.07	24.08	37.39	37.25	50.17	26.77	21.37	28.43
CRC	32.56	27.39	26.46	37.11	26.41	26.64	26.20	52.18	49.04	21.18	35.24	13.61	33.83	33.69	39.11	38.94	33.54	26.46
CRE	12.34	17.11	15.83	19.40	18.81	17.61	14.66	22.80	18.65	34.80	15.02	31.37	23.55	23.41	21.40	18.71	13.31	15.83
CRH	12.79	9.85	8.94	15.91	11.46	10.64	6.60	19.85	16.31	25.09	15.47	24.20	16.29	16.15	17.91	19.16	13.76	8.94
CRL	2.69	13.27	8.89	25.55	14.15	3.03	10.52	37.36	29.78	31.21	9.17	21.58	19.71	19.57	27.55	9.07	7.47	8.89
CRN	22.47	17.34	19.33	26.04	16.34	21.54	16.69	44.27	38.11	3.07	25.15	4.64	15.78	13.94	28.04	28.85	23.45	19.33
CRP	24.77	12.79	13.79	30.64	12.41	19.57	12.73	38.35	34.86	12.74	40.32	7.86	19.23	19.09	32.64	44.02	38.62	13.79
CRQ	31.23	25.38	24.54	19.05	26.40	29.39	22.06	3.81	1.04	43.53	33.92	42.48	31.82	31.68	21.05	37.61	32.21	24.54
DMA	29.37	11.34	16.88	4.12	10.24	21.32	15.08	29.64	26.17	25.32	32.06	27.52	17.78	17.64	6.12	35.75	30.35	16.88
DMF	1.90	10.83	6.24	28.91	11.80	3.56	7.70	33.15	25.21	23.64	4.58	25.24	17.27	17.13	30.91	8.28	2.88	6.54
DOG	27.95	10.18	21.91	2.30	9.27	23.57	13.16	20.01	17.06	26.25	30.63	25.29	16.62	16.48	2.30	34.32	28.92	21.91
DOK	5.02	11.34	8.63	25.10	12.93	8.11	9.94	28.37	25.44	28.96	7.70	27.85	17.78	17.64	27.10	11.40	6.00	8.63
DOP	5.47	10.70	5.90	26.43	11.62	0.00	7.63	33.20	26.34	24.61	8.15	20.20	17.14	17.00	28.43	11.84	6.44	5.90
FMG	30.25	12.48	24.21	0.56	11.57	25.87	15.46	22.31	19.36	28.55	32.93	27.59	18.92	18.78	1.44	36.62	31.22	24.21
FMS	32.01	26.12	25.98	44.23	23.99	26.53	26.24	51.65	48.07	22.52	42.93	15.79	32.56	32.42	46.23	38.39	41.23	25.98
GNL	8.75	4.53	2.17	23.86	6.16	6.40	1.78	27.08	24.12	22.53	11.43	21.22	10.97	10.83	25.86	15.13	9.73	2.17
GRC		12.73	8.14	30.81	13.70	5.47	9.86	35.05	27.11	25.54	6.48	27.14	19.17	19.03	32.81	6.38	4.78	8.14
GRG	12.73		5.81	13.04	1.13	10.70	3.45	29.19	24.80	20.41	15.41	17.92	6.44	6.30	15.04	19.11	13.70	5.81
GRK	8.14	5.81		24.77	7.59	5.90	2.58	28.35	24.23	22.40	10.82	18.03	12.25	12.11	26.77	14.52	9.12	0.00
GRL	30.81	13.04	24.77		12.13	26.43	16.02	22.87	19.92	29.11	33.49	28.15	19.48	19.34	2.00	37.18	31.78	24.77
GRM	13.70	1.13	7.59	12.13		11.62	5.07	30.22	23.92	19.41	16.39	16.66	7.57	7.43	14.13	20.08	14.68	7.59

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	MD	MDLJ	MPB	MPC	MPE	MPG	MPH	MPI	MPK	MPL	MPM	MPN	MPO	MPQ	MPR	MPS	MPU	MPV
O	70.85	88.68	39.69	56.69	12.66	82.21	88.92	73.42	88.83	64.72	52.88	78.40	49.25	63.83	86.48	81.27	71.02	16.61
CMS	25.75	42.31	23.40	19.62	55.58	42.36	41.40	32.46	41.22	15.41	14.20	29.50	23.84	16.65	45.63	46.94	25.90	46.37
CRA	19.87	31.16	22.75	11.13	54.54	30.97	30.32	31.78	29.64	14.84	6.02	19.13	28.78	8.16	44.86	34.39	14.95	45.33
CRB	51.18	21.17	37.63	20.39	71.10	3.33	12.36	56.70	11.89	40.46	11.56	24.73	33.22	32.33	71.90	2.81	20.50	61.89
CRC	40.12	10.90	48.86	32.56	80.80	7.03	2.10	57.96	5.32	41.04	20.24	15.03	54.28	27.49	71.57	7.56	14.51	71.59
CRE	22.41	42.66	10.23	12.34	41.28	42.59	41.86	28.58	41.05	15.73	8.56	29.58	15.45	15.39	41.72	43.59	26.93	32.08
CRH	18.92	35.47	16.56	12.79	48.75	35.53	34.57	25.63	34.38	8.57	7.36	22.66	17.00	9.82	38.80	40.11	19.07	39.54
CRL	28.56	19.99	19.71	2.69	51.80	23.57	28.65	43.14	31.77	22.25	6.32	19.99	15.82	17.48	52.57	23.46	16.25	42.59
CRN	29.05	14.42	38.67	22.47	71.76	21.83	20.21	50.05	21.21	30.59	21.82	3.40	37.49	19.19	67.15	27.34	5.98	62.55
CRP	33.65	31.28	36.29	37.64	66.42	17.40	16.67	44.12	15.72	29.86	33.95	6.55	41.10	13.18	57.12	17.89	0.33	57.21
CRQ	22.06	50.80	24.70	31.23	56.43	51.06	50.47	9.59	49.60	24.61	27.23	40.60	30.06	24.47	25.68	52.06	34.33	47.22
DMA	7.13	32.72	27.15	29.37	58.15	44.29	37.13	35.42	43.36	22.60	17.32	26.07	32.48	7.93	48.82	45.41	32.86	48.94
DMF	31.91	34.95	15.72	1.90	51.19	23.50	32.76	38.92	31.87	19.46	10.27	22.24	11.35	28.32	49.02	23.94	16.77	38.18
DOG	2.09	32.82	25.89	27.95	57.59	45.26	36.35	25.78	43.13	21.62	16.38	27.84	31.37	7.06	45.34	45.76	21.16	48.38
DOK	28.11	36.68	18.55	5.02	50.40	29.00	35.74	34.14	35.28	17.78	12.21	26.49	13.89	13.98	50.46	29.03	20.15	41.19
DOP	29.44	28.92	22.75	5.47	54.45	21.77	28.74	38.98	28.29	19.21	9.28	18.98	18.34	34.31	55.32	26.06	13.28	45.24
FMG	4.39	35.12	28.19	30.25	59.89	47.56	38.65	28.08	45.43	23.92	18.68	30.14	33.67	9.36	47.64	48.06	23.46	50.68
FMS	47.24	13.21	47.26	32.01	79.67	8.26	5.66	57.43	7.41	41.20	23.69	18.58	53.84	26.88	71.05	10.50	14.40	70.46
GNL	26.86	30.32	24.18	8.75	57.33	23.62	24.84	32.86	24.33	16.49	2.94	19.93	20.79	8.92	45.82	24.91	13.22	48.12
GRC	33.82	36.86	17.62	0.00	49.29	25.40	34.66	40.82	33.77	21.36	3.63	24.14	13.25	30.22	50.93	25.84	18.67	40.08
GRG	16.05	29.15	25.41	12.73	58.00	29.08	29.49	34.97	27.88	18.42	6.95	16.45	22.52	4.39	48.41	33.33	12.51	48.79
GRK	27.78	28.96	25.09	8.14	57.08	27.21	28.56	34.13	27.74	17.51	2.57	16.40	20.85	9.86	48.94	30.72	13.01	47.87
GRL	3.01	35.68	28.75	30.81	60.45	48.12	39.21	28.64	45.99	24.48	19.24	30.70	34.23	9.92	48.20	48.62	24.02	51.24
GRM	15.14	25.54	26.71	13.70	59.56	26.37	28.51	35.99	25.75	20.03	7.83	15.50	23.87	3.49	50.05	34.95	11.68	50.35

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	MPX	PCS	PRN	PY	PZC	PZG	PZK	PZM	PZO	PZS	RCD	RCJ	RCM	RCP	SMP	SPI	SRK	TTR
O	66.10	62.86	88.68	62.08	63.52	65.40	56.93	66.53	64.48	64.29	62.33	56.06	71.02	62.08	42.37	80.62	69.57	73.37
CMS	17.39	13.55	42.31	14.74	14.86	16.68	16.92	17.81	15.77	15.65	14.57	15.59	23.15	17.78	30.96	39.77	20.85	24.65
CRA	4.46	12.98	31.16	6.47	2.16	3.76	8.44	4.89	4.25	4.70	7.41	7.09	12.16	9.42	35.91	39.00	7.93	11.73
CRB	31.66	38.61	21.17	24.12	29.21	30.95	27.44	32.08	28.43	26.16	22.40	24.98	22.87	21.04	40.34	66.04	35.12	38.92
CRC	28.09	39.19	10.90	29.74	27.50	27.39	34.52	28.52	26.46	28.43	27.73	33.60	16.57	26.39	61.40	65.71	31.56	35.36
CRE	17.81	13.88	42.66	17.26	15.28	17.11	13.63	18.24	15.83	15.63	14.81	15.79	23.70	17.80	22.58	35.86	21.28	25.08
CRH	10.55	6.72	35.47	7.90	8.02	9.85	10.09	10.98	8.94	8.81	7.73	8.75	16.32	10.94	24.12	32.94	14.02	17.82
CRL	13.97	20.40	19.99	6.47	11.56	13.27	7.41	14.40	8.89	8.63	6.79	7.42	12.13	3.25	22.94	46.71	17.44	21.24
CRN	18.04	28.74	14.42	18.71	16.38	17.34	26.76	18.47	19.33	16.70	21.33	25.42	8.72	18.69	44.61	61.29	14.51	17.31
CRP	13.49	28.01	31.28	15.38	13.63	12.79	21.25	13.92	13.79	13.95	14.38	19.78	5.56	13.42	48.22	51.26	16.96	20.76
CRQ	26.08	22.76	50.80	25.73	23.65	25.38	25.43	26.51	24.54	24.17	23.14	23.93	31.72	28.95	37.18	19.82	29.55	33.35
DMA	12.04	20.75	32.72	21.95	22.68	11.34	23.83	12.47	16.88	23.29	21.29	22.37	17.53	27.00	39.60	42.96	15.51	19.31
DMF	11.53	17.60	34.95	3.98	8.94	10.83	4.00	11.96	6.54	5.99	3.59	2.97	16.12	3.78	18.47	43.16	15.00	18.80
DOG	10.88	19.77	32.82	20.77	20.54	10.18	23.12	11.31	21.91	21.59	20.75	26.93	25.68	33.10	38.50	39.48	14.35	18.15
DOK	12.04	15.93	36.68	5.69	9.52	11.34	0.88	12.47	8.63	7.83	3.91	2.45	17.37	8.05	21.01	44.59	15.51	19.31
DOP	11.40	17.36	28.92	3.63	8.95	10.70	8.98	11.83	5.90	5.82	3.96	10.00	10.72	0.57	25.47	49.46	14.87	18.67
FMG	13.18	22.07	35.12	23.07	22.84	12.48	25.42	13.61	24.21	23.89	23.05	29.23	27.98	35.40	40.80	41.78	16.65	20.45
FMS	26.83	39.34	13.21	28.62	27.15	26.12	34.40	27.25	25.98	26.15	27.56	32.86	16.31	33.13	60.97	65.19	30.29	34.09
GNL	5.23	14.64	30.32	3.04	2.66	4.53	8.78	5.66	2.17	1.47	4.27	8.61	10.62	6.71	27.92	39.96	8.70	12.50
GRC	13.43	19.50	36.86	5.88	10.85	12.73	5.90	13.86	8.14	7.89	5.50	4.87	18.02	5.68	20.37	45.06	16.90	20.70
GRG	0.70	16.57	29.15	7.02	2.38	0.00	12.21	1.13	5.81	5.34	8.22	10.57	8.87	9.96	29.65	42.55	4.17	7.97
GRK	6.52	15.66	28.96	2.48	3.61	5.81	9.50	6.94	0.00	0.91	4.25	8.28	10.27	5.63	27.97	43.08	9.98	13.78
GRL	13.74	22.63	35.68	23.63	23.40	13.04	25.98	14.17	24.77	24.45	23.61	29.79	28.54	35.96	41.36	42.34	17.21	21.01
GRM	1.84	18.18	25.54	7.94	3.26	1.13	13.81	2.26	7.59	7.02	9.83	11.98	8.03	11.23	30.99	44.19	5.30	9.10

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	O	CMS	CRA	CRB	CRC	CRE	CRH	CRL	CRN	CRP	CRQ	DMA	DMF	DOG	DOK	DOP	FMG	FMS	GNL
GRP	61.85	17.47	9.33	20.75	26.64	17.61	10.64	3.03	21.54	19.57	29.39	21.32	3.56	23.57	8.11	0.00	25.87	26.53	6.40
GRS	62.43	13.44	2.74	28.15	26.20	14.66	6.60	10.52	16.69	12.73	22.06	15.08	7.70	13.16	9.94	7.63	15.46	26.24	1.78
HHB	67.64	26.68	26.01	50.92	52.18	22.80	19.85	37.36	44.27	38.35	3.81	29.64	33.15	20.01	28.37	33.20	22.31	51.65	27.08
HHC	64.47	23.15	19.98	48.13	49.04	18.65	16.31	29.78	38.11	34.86	1.04	26.17	25.21	17.06	25.44	26.34	19.36	48.07	24.12
HHD	82.23	31.93	22.28	30.44	21.18	34.80	25.09	31.21	3.07	12.74	43.53	25.32	23.64	26.25	28.96	24.61	28.55	22.52	22.53
HHF	63.17	22.30	13.81	23.07	35.24	15.02	15.47	9.17	25.15	40.32	33.92	32.06	4.58	30.63	7.70	8.15	32.93	42.93	11.43
HHK	79.77	31.03	19.91	24.08	13.61	31.37	24.20	21.58	4.64	7.86	42.48	27.52	25.24	25.29	27.85	20.20	27.59	15.79	21.22
HHL	71.84	23.12	10.20	37.39	33.83	23.55	16.29	19.71	15.78	19.23	31.82	17.78	17.27	16.62	17.78	17.14	18.92	32.56	10.97
HHR	71.70	22.98	10.06	37.25	33.69	23.41	16.15	19.57	13.94	19.09	31.68	17.64	17.13	16.48	17.64	17.00	18.78	32.42	10.83
JPG	69.84	24.74	18.87	50.17	39.11	21.40	17.91	27.55	28.04	32.64	21.05	6.12	30.91	2.30	27.10	28.43	1.44	46.23	25.86
KB	63.07	26.00	17.50	26.77	38.94	18.71	19.16	9.07	28.85	44.02	37.61	35.75	8.28	34.32	11.40	11.84	36.62	38.39	15.13
KS	61.46	20.59	12.10	21.37	33.54	13.31	13.76	7.47	23.45	38.62	32.21	30.35	2.88	28.92	6.00	6.44	31.22	41.23	9.73
MBP	64.48	15.77	4.25	28.43	26.46	15.83	8.94	8.89	19.33	13.79	24.54	16.88	6.54	21.91	8.63	5.90	24.21	25.98	2.17
MD	70.85	25.75	19.87	51.18	40.12	22.41	18.92	28.56	29.05	33.65	22.06	7.13	31.91	2.09	28.11	29.44	4.39	47.24	26.86
MDLJ	88.68	42.31	31.16	21.17	10.90	42.66	35.47	19.99	14.42	31.28	50.80	32.72	34.95	32.82	36.68	28.92	35.12	13.21	30.32
MPB	39.69	23.40	22.75	37.63	48.86	10.23	16.56	19.71	38.67	36.29	24.70	27.15	15.72	25.89	18.55	22.75	28.19	47.26	24.18
MPC	56.69	19.62	11.13	20.39	32.56	12.34	12.79	2.69	22.47	37.64	31.23	29.37	1.90	27.95	5.02	5.47	30.25	32.01	8.75
MPE	12.66	55.58	54.54	71.10	80.80	41.28	48.75	51.80	71.76	66.42	56.43	58.15	51.19	57.59	50.40	54.45	59.89	79.67	57.33
MPG	82.21	42.36	30.97	3.33	7.03	42.59	35.53	23.57	21.83	17.40	51.06	44.29	23.50	45.26	29.00	21.77	47.56	8.26	23.62
MPH	88.92	41.40	30.32	12.36	2.10	41.86	34.57	28.65	20.21	16.67	50.47	37.13	32.76	36.35	35.74	28.74	38.65	5.66	24.84
MPI	73.42	32.46	31.78	56.70	57.96	28.58	25.63	43.14	50.05	44.12	9.59	35.42	38.92	25.78	34.14	38.98	28.08	57.43	32.86
MPK	88.83	41.22	29.64	11.89	5.32	41.05	34.38	31.77	21.21	15.72	49.60	43.36	31.87	43.13	35.28	28.29	45.43	7.41	24.33
MPL	64.72	15.41	14.84	40.46	41.04	15.73	8.57	22.25	30.59	29.86	24.61	22.60	19.46	21.62	17.78	19.21	23.92	41.20	16.49
MPM	52.88	14.20	6.02	11.56	20.24	8.56	7.36	6.32	21.82	33.95	27.23	17.32	10.27	16.38	12.21	9.28	18.68	23.69	2.94

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	GRC	GRG	GRK	GRL	GRM	GRP	GRS	HHB	HHC	HHD	HHF	HHK	HHL	HHR	JPG	KB	KS	MBP
GRP	5.47	10.70	5.90	26.43	11.62		7.63	33.20	26.34	24.61	8.15	20.20	17.14	17.00	28.43	11.84	6.44	5.90
GRS	9.86	3.45	2.58	16.02	5.07	7.63		25.88	22.03	19.76	12.29	18.05	9.89	9.75	18.02	15.98	10.58	2.58
HHB	35.05	29.19	28.35	22.87	30.22	33.20	25.88		4.85	47.34	37.73	46.30	35.63	35.49	24.87	41.42	36.02	28.35
HHC	27.11	24.80	24.23	19.92	23.92	26.34	22.03	4.85		41.18	29.79	40.26	31.24	31.10	21.92	33.49	28.09	24.23
HHD	25.54	20.41	22.40	29.11	19.41	24.61	19.76	47.34	41.18		28.22	7.71	13.36	12.22	31.11	31.92	26.52	22.40
HHF	6.48	15.41	10.82	33.49	16.39	8.15	12.29	37.73	29.79	28.22		29.82	21.85	21.71	35.49	12.86	2.96	11.12
HHK	27.14	17.92	18.03	28.15	16.66	20.20	18.05	46.30	40.26	7.71	29.82		24.36	24.22	30.15	33.52	28.11	18.03
HHL	19.17	6.44	12.25	19.48	7.57	17.14	9.89	35.63	31.24	13.36	21.85	24.36		4.24	21.48	25.55	20.14	12.25
HHR	19.03	6.30	12.11	19.34	7.43	17.00	9.75	35.49	31.10	12.22	21.71	24.22	4.24		21.34	25.41	20.00	12.11
JPG	32.81	15.04	26.77	2.00	14.13	28.43	18.02	24.87	21.92	31.11	35.49	30.15	21.48	21.34		39.18	33.78	26.77
KB	6.38	19.11	14.52	37.18	20.08	11.84	15.98	41.42	33.49	31.92	12.86	33.52	25.55	25.41	39.18		11.15	14.52
KS	4.78	13.70	9.12	31.78	14.68	6.44	10.58	36.02	28.09	26.52	2.96	28.11	20.14	20.00	33.78	11.15		9.42
MBP	8.14	5.81	0.00	24.77	7.59	5.90	2.58	28.35	24.23	22.40	11.12	18.03	12.25	12.11	26.77	14.52	9.42	
MD	33.82	16.05	27.78	3.01	15.14	29.44	19.03	25.87	22.93	32.12	36.50	31.15	22.49	22.35	1.01	40.19	34.79	27.78
MDLJ	36.86	29.15	28.96	35.68	25.54	28.92	29.27	54.61	51.28	17.49	39.54	9.82	35.59	35.45	37.68	43.23	37.83	28.96
MPB	17.62	25.41	25.09	28.75	26.71	22.75	22.53	28.52	24.91	41.74	20.30	45.71	31.85	31.71	30.74	24.00	18.59	25.09
MPC	0.00	12.73	8.14	30.81	13.70	5.47	9.61	35.05	27.11	25.54	6.48	27.14	19.17	19.03	32.81	6.38	4.78	8.14
MPE	49.29	58.00	57.08	60.45	59.56	54.45	55.03	60.24	57.07	74.83	55.77	72.37	64.44	64.30	62.44	55.67	54.06	57.08
MPG	25.40	29.08	27.21	48.12	26.37	21.77	28.49	54.88	51.08	24.90	28.08	20.70	35.52	35.38	50.12	31.77	26.37	27.21
MPH	34.66	29.49	28.56	39.21	28.51	28.74	28.30	54.28	51.14	23.28	37.34	15.71	35.93	35.79	41.21	41.04	35.64	28.56
MPI	40.82	34.97	34.13	28.64	35.99	38.98	31.65	5.78	10.63	53.12	43.51	52.07	41.41	41.27	30.64	47.20	41.80	34.13
MPK	33.77	27.88	27.74	45.99	25.75	28.29	28.00	53.41	49.83	24.28	36.45	17.55	34.32	34.18	47.99	40.15	34.74	27.74
MPL	21.36	18.42	17.51	24.48	20.03	19.21	15.18	28.42	24.88	33.66	24.04	32.77	24.86	24.72	26.48	27.73	22.33	17.51
MPM	3.63	6.95	2.57	19.24	7.83	3.28	4.21	31.05	23.46	24.89	2.86	15.26	13.39	13.25	21.24	2.75	1.15	2.57

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	MD	MDLJ	MPB	MPC	MPE	MPG	MPH	MPI	MPK	MPL	MPM	MPN	MPO	MPQ	MPR	MPS	MPU	MPV
GRP	29.44	28.92	22.75	5.47	54.45	21.77	28.74	38.98	28.29	19.21	3.28	18.98	18.34	34.31	55.32	26.06	13.28	45.24
GRS	19.03	29.27	22.53	9.61	55.03	28.49	28.30	31.65	28.00	15.18	4.21	16.76	19.42	7.43	46.18	30.57	12.68	45.82
HHB	25.87	54.61	28.52	35.05	60.24	54.88	54.28	5.78	53.41	28.42	31.05	44.41	33.87	28.28	23.05	55.88	38.15	51.03
HHC	22.93	51.28	24.91	27.11	57.07	51.08	51.14	10.63	49.83	24.88	23.46	38.80	30.75	23.99	25.32	52.74	35.19	47.86
HHD	32.12	17.49	41.74	25.54	74.83	24.90	23.28	53.12	24.28	33.66	24.89	6.47	40.56	22.26	70.22	30.41	9.05	65.62
HHF	36.50	39.54	20.30	6.48	55.77	28.08	37.34	43.51	36.45	24.04	2.86	26.83	15.93	32.90	53.61	28.52	21.35	42.76
HHK	31.15	9.82	45.71	27.14	72.37	20.70	15.71	52.07	17.55	32.77	15.26	2.80	41.41	21.18	65.38	25.51	7.72	63.16
HHL	22.49	35.59	31.85	19.17	64.44	35.52	35.93	41.41	34.32	24.86	13.39	22.89	28.96	10.83	54.85	39.77	18.95	55.23
HHR	22.35	35.45	31.71	19.03	64.30	35.38	35.79	41.27	34.18	24.72	13.25	22.75	28.82	10.69	54.71	39.63	18.81	55.09
JPG	1.01	37.68	30.74	32.81	62.44	50.12	41.21	30.64	47.99	26.48	21.24	32.70	36.23	11.92	50.20	50.62	26.02	53.23
KB	40.19	43.23	24.00	6.38	55.67	31.77	41.04	47.20	40.15	27.73	2.75	30.52	19.63	36.60	57.30	32.22	25.05	46.46
KS	34.79	37.83	18.59	4.78	54.06	26.37	35.64	41.80	34.74	22.33	1.15	25.12	14.23	31.20	51.90	26.82	19.64	41.05
MBP	27.78	28.96	25.09	8.14	57.08	27.21	28.56	34.13	27.74	17.51	2.57	16.40	20.85	9.86	48.94	30.72	13.01	47.87
MD		38.68	31.75	33.82	63.45	51.13	42.22	31.65	49.00	29.43	22.24	33.71	37.24	12.93	51.21	51.63	27.03	54.24
MDLJ	38.68		49.29	36.86	81.28	48.90	13.00	60.39	14.97	44.04	13.67	11.02	57.40	27.76	77.38	49.32	17.13	72.07
MPB	31.75	49.29		17.62	32.29	40.34	50.96	34.29	49.02	25.14	13.40	39.24	9.56	25.69	48.32	41.34	35.06	23.08
MPC	33.82	36.86	17.62		49.29	25.40	34.66	40.82	33.77	21.36	-3.63	24.14	13.25	30.22	50.93	25.84	18.67	40.08
MPE	63.45	81.28	32.29	49.29		74.81	81.52	66.02	81.43	57.32	45.48	71.00	41.85	56.43	79.08	73.87	63.62	9.21
MPG	51.13	48.90	40.34	25.40	74.81		9.13	60.65	10.01	44.10	17.25	22.12	39.92	30.09	75.35	0.51	16.82	65.60
MPH	42.22	13.00	50.96	34.66	81.52	9.13		60.06	7.42	43.14	22.34	17.13	56.38	29.59	73.67	9.66	16.61	73.69
MPI	31.65	60.39	34.29	40.82	66.02	60.65	60.06		59.19	34.20	36.82	50.19	39.65	34.06	28.82	61.65	43.92	56.81
MPK	49.00	14.97	49.02	33.77	81.43	10.01	7.42	59.19		42.96	25.45	20.34	55.60	28.64	72.80	12.26	16.16	72.22
MPL	29.43	44.04	25.14	21.36	57.32	44.10	43.14	34.20	42.96		15.93	31.24	25.57	18.39	47.37	48.68	27.64	48.11
MPM	22.24	13.67	13.40	-3.63	45.48	17.25	22.34	36.82	25.45	15.93		13.67	9.50	11.16	46.25	17.14	9.94	36.27

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	MPX	PCS	PRN	PY	PZC	PZG	PZK	PZM	PZO	PZS	RCD	RCJ	RCM	RCP	SMP	SPI	SRK	TTR
GRP	11.40	17.36	28.92	3.63	8.95	10.70	8.98	11.83	5.90	5.82	3.96	10.00	10.72	0.57	25.47	49.46	14.87	18.67
GRS	4.15	13.32	29.27	4.01	1.40	3.45	10.82	4.58	2.58	2.35	5.51	7.48	10.03	7.31	26.54	40.32	7.62	11.42
HHB	29.90	26.57	54.61	29.55	27.47	29.19	29.24	30.32	28.35	27.99	26.95	27.74	35.54	32.76	40.99	17.19	33.36	37.16
HHC	25.51	23.03	51.28	23.77	23.95	24.80	26.31	25.93	24.23	23.77	23.97	24.60	32.37	27.04	37.87	19.46	28.97	32.77
HHD	21.11	31.81	17.49	21.78	19.45	20.41	29.83	21.54	22.40	19.77	24.40	28.49	11.79	21.76	47.68	64.36	12.69	14.89
HHF	16.11	22.19	39.54	8.56	13.53	15.41	8.58	16.54	11.12	10.57	8.18	7.55	20.70	8.37	23.06	47.75	19.58	23.38
HHK	18.62	30.92	9.82	20.55	18.36	17.92	28.72	19.05	18.03	18.28	21.83	25.03	10.59	23.71	48.53	59.52	22.09	25.89
HHL	7.14	23.01	35.59	13.46	8.82	6.44	18.65	5.31	12.25	11.78	14.66	17.01	15.31	16.40	36.09	48.99	2.27	1.53
HHR	7.00	22.87	35.45	13.32	8.68	6.30	18.51	5.17	12.11	11.64	14.52	16.87	15.17	16.26	35.95	48.85	2.13	5.77
JPG	15.74	24.63	37.68	25.62	25.40	15.04	27.98	16.17	26.77	26.45	25.61	31.79	30.54	37.96	43.35	44.34	19.21	23.01
KB	19.81	25.88	43.23	12.26	17.22	19.11	12.27	20.24	14.52	14.26	11.87	11.24	24.39	12.06	26.75	51.44	23.28	27.08
KS	14.41	20.48	37.83	6.85	11.82	13.70	6.87	14.83	9.42	8.86	6.47	5.84	18.99	6.66	21.35	46.04	17.87	21.67
MBP	6.52	15.66	28.96	2.48	3.61	5.81	9.50	6.94	0.00	0.91	4.25	8.28	10.27	5.63	27.97	43.08	9.98	13.78
MD	16.75	25.64	38.68	26.63	26.40	16.05	28.99	17.18	27.78	27.45	26.62	32.80	31.55	38.97	44.36	45.35	20.22	24.02
MDLJ	29.86	42.19	0.00	36.77	30.64	29.15	37.55	30.28	28.96	29.56	30.92	36.81	29.74	36.16	64.52	71.52	33.32	37.12
MPB	26.12	23.28	49.29	23.33	24.62	25.41	19.42	26.54	25.09	25.09	22.79	17.09	31.76	23.08	16.68	42.46	29.58	33.38
MPC	13.43	19.50	36.86	5.88	10.85	12.73	5.90	13.86	8.14	7.89	5.50	4.87	18.02	5.68	20.37	45.06	16.90	20.70
MPE	58.70	55.46	81.28	54.68	56.12	58.00	49.53	59.13	57.08	56.89	54.93	48.66	63.62	54.68	34.97	73.22	62.17	65.97
MPG	29.78	42.25	48.90	33.68	31.22	29.08	29.87	30.21	27.21	32.52	25.02	37.22	20.66	37.16	47.05	69.49	33.25	37.05
MPH	30.19	41.29	13.00	31.84	29.60	29.49	36.61	30.62	28.56	30.53	29.83	35.70	18.67	28.49	63.50	67.81	33.66	37.46
MPI	35.67	32.35	60.39	35.32	33.24	34.97	35.02	36.10	34.13	33.76	32.73	33.52	41.31	38.54	46.77	22.96	39.14	42.94
MPK	28.58	41.10	14.97	30.38	28.90	27.88	36.15	29.01	27.74	27.90	29.32	34.62	18.06	34.89	62.73	66.94	32.05	35.85
MPL	19.13	15.29	44.04	16.48	16.60	18.42	18.66	19.55	17.51	17.39	16.31	17.32	24.89	19.51	32.70	41.51	22.59	26.39
MPM	7.66	14.08	13.67	3.16	5.24	6.95	4.09	8.08	2.57	2.31	3.48	1.11	5.82	3.07	16.63	40.39	11.12	14.92

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	O	CMS	CRA	CRB	CRC	CRE	CRH	CRL	CRN	CRP	CRQ	DMA	DMF	DOG	DOK	DOP	FMG	FMS	GNL
MPN	78.40	29.50	19.13	24.73	15.03	29.58	22.66	19.99	3.40	6.55	40.60	26.07	22.24	27.84	26.49	18.98	30.14	18.58	19.93
MPO	49.25	23.84	28.78	33.22	54.28	15.45	17.00	15.82	37.49	41.10	30.06	32.48	11.35	31.37	13.89	18.34	33.67	53.84	20.79
MPQ	63.83	16.65	8.16	32.33	27.49	15.39	9.82	17.48	19.19	13.18	24.47	7.93	28.32	7.06	13.98	34.31	9.36	26.88	8.92
MPR	86.48	45.63	44.86	71.90	71.57	41.72	38.80	52.57	67.15	57.12	25.68	48.82	49.02	45.34	50.46	55.32	47.64	71.05	45.82
MPS	81.27	46.94	34.39	2.81	7.56	43.59	40.11	23.46	27.34	17.89	52.06	45.41	23.94	45.76	29.03	26.06	48.06	10.50	24.91
MPU	71.02	25.90	14.95	20.50	14.51	26.93	19.07	16.25	5.98	0.33	34.33	32.86	16.77	21.16	20.15	13.28	23.46	14.40	13.22
MPV	16.61	46.37	45.33	61.89	71.59	32.08	39.54	42.59	62.55	57.21	47.22	48.94	38.18	48.38	41.19	45.24	50.68	70.46	48.12
MPX	66.10	17.39	4.46	31.66	28.09	17.81	10.55	13.97	18.04	13.49	26.08	12.04	11.53	10.88	12.04	11.40	13.18	26.83	5.23
PCS	62.86	13.55	12.98	38.61	39.19	13.88	6.72	20.40	28.74	28.01	22.76	20.75	17.60	19.77	15.93	17.36	22.07	39.34	14.64
PRN	88.68	42.31	31.16	21.17	10.90	42.66	35.47	19.99	14.42	31.28	50.80	32.72	34.95	32.82	36.68	28.92	35.12	13.21	30.32
PY	62.08	14.74	6.47	24.12	29.74	17.26	7.90	6.47	18.71	15.38	25.73	21.95	3.98	20.77	5.69	3.63	23.07	28.62	3.04
PZC	63.52	14.86	2.16	29.21	27.50	15.28	8.02	11.56	16.38	13.63	23.65	22.68	8.94	20.54	9.52	8.95	22.84	27.15	2.66
PZG	65.40	16.68	3.76	30.95	27.39	17.11	9.85	13.27	17.34	12.79	25.38	11.34	10.83	10.18	11.34	10.70	12.48	26.12	4.53
PZK	56.93	16.92	8.44	27.44	34.52	13.63	10.09	7.41	26.76	21.25	25.43	23.83	4.00	23.12	0.88	8.98	25.42	34.40	8.78
PZM	66.53	17.81	4.89	32.08	28.52	18.24	10.98	14.40	18.47	13.92	26.51	12.47	11.96	11.31	12.47	11.83	13.61	27.25	5.66
PZO	64.48	15.77	4.25	28.43	26.46	15.83	8.94	8.89	19.33	13.79	24.54	16.88	6.54	21.91	8.63	5.90	24.21	25.98	2.17
PZS	64.29	15.65	4.70	26.16	28.43	15.63	8.81	8.63	16.70	13.95	24.17	23.29	5.99	21.59	7.83	5.82	23.89	26.15	1.47
RCD	62.33	14.57	7.41	22.40	27.73	14.81	7.73	6.79	21.33	14.38	23.14	21.29	3.59	20.75	3.91	3.96	23.05	27.56	4.27
RCJ	56.06	15.59	7.09	24.98	33.60	15.79	8.75	7.42	25.42	19.78	23.93	22.37	2.97	26.93	2.45	10.00	29.23	32.86	8.61
RCM	71.02	23.15	12.16	22.87	16.57	23.70	16.32	12.13	8.72	5.56	31.72	17.53	16.12	25.68	17.37	10.72	27.98	16.31	10.62
RCP	62.08	17.78	9.42	21.04	26.39	17.80	10.94	3.25	18.69	13.42	28.95	27.00	3.78	33.10	8.05	0.57	35.40	33.13	6.71
SMP	42.37	30.96	35.91	40.34	61.40	22.58	24.12	22.94	44.61	48.22	37.18	39.60	18.47	38.50	21.01	25.47	40.80	60.97	27.92
SPI	80.62	39.77	39.00	66.04	65.71	35.86	32.94	46.71	61.29	51.26	19.82	42.96	43.16	39.48	44.59	49.46	41.78	65.19	39.96
SRK	69.57	20.85	7.93	35.12	31.56	21.28	14.02	17.44	14.51	16.96	29.55	15.51	15.00	14.35	15.51	14.87	16.65	30.29	8.70
TTR	73.37	24.65	11.73	38.92	35.36	25.08	17.82	21.24	17.31	20.76	33.35	19.31	18.80	18.15	19.31	18.67	20.45	34.09	12.50

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	GRC	GRG	GRK	GRL	GRM	GRP	GRS	HHB	HHC	HHD	HHF	HHK	HHL	HHR	JPG	KB	KS	MBP
MPN	24.14	16.45	16.40	30.70	15.50	18.98	16.76	44.41	38.80	6.47	26.83	2.80	22.89	22.75	32.70	30.52	25.12	16.40
MPO	13.25	22.52	20.85	34.23	23.87	18.34	19.42	33.87	30.75	40.56	15.93	41.41	28.96	28.82	36.23	19.63	14.23	20.85
MPQ	30.22	4.39	9.86	9.92	3.49	34.31	7.43	28.28	23.99	22.26	32.90	21.18	10.83	10.69	11.92	36.60	31.20	9.86
MPR	50.93	48.41	48.94	48.20	50.05	55.32	46.18	23.05	25.32	70.22	53.61	65.38	54.85	54.71	50.20	57.30	51.90	48.94
MPS	25.84	33.33	30.72	48.62	34.95	26.06	30.57	55.88	52.74	30.41	28.52	25.51	39.77	39.63	50.62	32.22	26.82	30.72
MPU	18.67	12.51	13.01	24.02	11.68	13.28	12.68	38.15	35.19	9.05	21.35	7.72	18.95	18.81	26.02	25.05	19.64	13.01
MPV	40.08	48.79	47.87	51.24	50.35	45.24	45.82	51.03	47.86	65.62	42.76	63.16	55.23	55.09	53.23	46.46	41.05	47.87
MPX	13.43	0.70	6.52	13.74	1.84	11.40	4.15	29.90	25.51	21.11	16.11	18.62	7.14	7.00	15.74	19.81	14.41	6.52
PCS	19.50	16.57	15.66	22.63	18.18	17.36	13.32	26.57	23.03	31.81	22.19	30.92	23.01	22.87	24.63	25.88	20.48	15.66
PRN	36.86	29.15	28.96	35.68	25.54	28.92	29.27	54.61	51.28	17.49	39.54	9.82	35.59	35.45	37.68	43.23	37.83	28.96
PY	5.88	7.02	2.48	23.63	7.94	3.63	4.01	29.55	23.77	21.78	8.56	20.55	13.46	13.32	25.62	12.26	6.85	2.48
PZC	10.85	2.38	3.61	23.40	3.26	8.95	1.40	27.47	23.95	19.45	13.53	18.36	8.82	8.68	25.40	17.22	11.82	3.61
PZG	12.73	0.00	5.81	13.04	1.13	10.70	3.45	29.19	24.80	20.41	15.41	17.92	6.44	6.30	15.04	19.11	13.70	5.81
PZK	5.90	12.21	9.50	25.98	13.81	8.98	10.82	29.24	26.31	29.83	8.58	28.72	18.65	18.51	27.98	12.27	6.87	9.50
PZM	13.86	1.13	6.94	14.17	2.26	11.83	4.58	30.32	25.93	21.54	16.54	19.05	5.31	5.17	16.17	20.24	14.83	6.94
PZO	8.14	5.81	0.00	24.77	7.59	5.90	2.58	28.35	24.23	22.40	11.12	18.03	12.25	12.11	26.77	14.52	9.42	0.00
PZS	7.89	5.34	0.91	24.45	7.02	5.82	2.35	27.99	23.77	19.77	10.57	18.28	11.78	11.64	26.45	14.26	8.86	0.91
RCD	5.50	8.22	4.25	23.61	9.83	3.96	5.51	26.95	23.97	24.40	8.18	21.83	14.66	14.52	25.61	11.87	6.47	4.25
RCJ	4.87	10.57	8.28	29.79	11.98	10.00	7.48	27.74	24.60	28.49	7.55	25.03	17.01	16.87	31.79	11.24	5.84	8.28
RCM	18.02	8.87	10.27	28.54	8.03	10.72	10.03	35.54	32.37	11.79	20.70	10.59	15.31	15.17	30.54	24.39	18.99	10.27
RCP	5.68	9.96	5.63	35.96	11.23	0.57	7.31	32.76	27.04	21.76	8.37	23.71	16.40	16.26	37.96	12.06	6.66	5.63
SMP	20.37	29.65	27.97	41.36	30.99	25.47	26.54	40.99	37.87	47.68	23.06	48.53	36.09	35.95	43.35	26.75	21.35	27.97
SPI	45.06	42.55	43.08	42.34	44.19	49.46	40.32	17.19	19.46	64.36	47.75	59.52	48.99	48.85	44.34	51.44	46.04	43.08
SRK	16.90	4.17	9.98	17.21	5.30	14.87	7.62	33.36	28.97	12.69	19.58	22.09	2.27	2.13	19.21	23.28	17.87	9.98
TTR	20.70	7.97	13.78	21.01	9.10	18.67	11.42	37.16	32.77	14.89	23.38	25.89	1.53	5.77	23.01	27.08	21.67	13.78

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	MD	MDLJ	MPB	MPC	MPE	MPG	MPH	MPI	MPK	MPL	MPM	MPN	MPO	MPQ	MPR	MPS	MPU	MPV
MPN	33.71	11.02	39.24	24.14	71.00	22.12	17.13	50.19	20.34	31.24	13.67		42.89	19.58	64.16	24.23	6.07	61.79
MPO	37.24	57.40	9.56	13.25	41.85	39.92	56.38	39.65	55.60	25.57	9.50	42.89		32.47	53.19	40.92	32.18	32.64
MPQ	12.93	27.76	25.69	30.22	56.43	30.09	29.59	34.06	28.64	18.39	11.16	19.58	32.47		46.73	30.53	13.99	47.22
MPR	51.21	77.38	48.32	50.93	79.08	75.35	73.67	28.82	72.80	47.37	46.25	64.16	53.19	46.73		75.76	57.69	69.87
MPS	51.63	49.32	41.34	25.84	73.87	0.51	9.66	61.65	12.26	48.68	17.14	24.23	40.92	30.53	75.76		19.75	64.66
MPU	27.03	17.13	35.06	18.67	63.62	16.82	16.61	43.92	16.16	27.64	9.94	6.07	32.18	13.99	57.69	19.75		54.41
MPV	54.24	72.07	23.08	40.08	9.21	65.60	73.69	56.81	72.22	48.11	36.27	61.79	32.64	47.22	69.87	64.66	54.41	
MPX	16.75	29.86	26.12	13.43	58.70	29.78	30.19	35.67	28.58	19.13	7.66	17.15	23.23	5.09	49.11	34.03	13.22	49.49
PCS	25.64	42.19	23.28	19.50	55.46	42.25	41.29	32.35	41.10	15.29	14.08	29.38	23.72	16.54	45.52	46.83	25.79	46.26
PRN	38.68	0.00	49.29	36.86	81.28	48.90	13.00	60.39	14.97	44.04	13.67	11.02	57.40	27.76	77.38	49.32	17.13	72.07
PY	26.63	36.77	23.33	5.88	54.68	33.68	31.84	35.32	30.38	16.48	3.16	22.24	19.53	11.46	49.33	34.68	16.18	45.47
PZC	26.40	30.64	24.62	10.85	56.12	31.22	29.60	33.24	28.90	16.60	5.24	16.51	23.19	6.42	47.45	32.36	13.93	46.91
PZG	16.05	29.15	25.41	12.73	58.00	29.08	29.49	34.97	27.88	18.42	6.95	16.45	22.52	4.39	48.41	33.33	12.51	48.79
PZK	28.99	37.55	19.42	5.90	49.53	29.87	36.61	35.02	36.15	18.66	4.09	27.37	14.76	14.85	51.33	29.90	21.02	42.06
PZM	17.18	30.28	26.54	13.86	59.13	30.21	30.62	36.10	29.01	19.55	8.08	17.58	23.65	5.52	49.54	34.46	13.64	49.92
PZO	27.78	28.96	25.09	8.14	57.08	27.21	28.56	34.13	27.74	17.51	2.57	16.40	20.85	9.86	48.94	30.72	13.01	47.87
PZS	27.45	29.56	25.09	7.89	56.89	32.52	30.53	33.76	27.90	17.39	2.31	17.43	22.74	9.57	47.33	30.86	14.66	47.68
RCD	26.62	30.92	22.79	5.50	54.93	25.02	29.83	32.73	29.32	16.31	3.48	20.52	18.34	12.82	45.72	24.94	14.19	45.72
RCJ	32.80	36.81	17.09	4.87	48.66	37.22	35.70	33.52	34.62	17.32	1.11	23.27	13.90	28.62	49.44	38.22	19.79	39.45
RCM	31.55	29.74	31.76	18.02	63.62	20.66	18.67	41.31	18.06	24.89	5.82	9.95	30.72	10.84	54.98	23.59	2.82	54.41
RCP	38.97	36.16	23.08	5.68	54.68	37.16	28.49	38.54	34.89	19.51	3.07	23.87	18.84	34.66	55.55	26.30	19.33	45.47
SMP	44.36	64.52	16.68	20.37	34.97	47.05	63.50	46.77	62.73	32.70	16.63	50.01	7.12	39.59	60.32	48.05	39.30	39.76
SPI	45.35	71.52	42.46	45.06	73.22	69.49	67.81	22.96	66.94	41.51	40.39	58.29	47.33	40.87	5.86	69.90	51.83	64.01
SRK	20.22	33.32	29.58	16.90	62.17	33.25	33.66	39.14	32.05	22.59	11.12	20.62	26.69	8.56	52.58	37.50	16.68	52.96
TTR	24.02	37.12	33.38	20.70	65.97	37.05	37.46	42.94	35.85	26.39	14.92	24.42	30.49	12.36	56.38	41.30	20.48	56.76

Lampiran 1 Matriks Jarak Antar Konsumen (lanjutan)

Matriks Jarak (km)	MPX	PCS	PRN	PY	PZC	PZG	PZK	PZM	PZO	PZS	RCD	RCJ	RCM	RCP	SMP	SPI	SRK	TTR
MPN	17.15	29.38	11.02	22.24	16.51	16.45	27.37	17.58	16.40	17.43	20.52	23.27	9.95	23.87	50.01	58.29	20.62	24.42
MPO	23.23	23.72	57.40	19.53	23.19	22.52	14.76	23.65	20.85	22.74	18.34	13.90	30.72	18.84	7.12	47.33	26.69	30.49
MPQ	5.09	16.54	27.76	11.46	6.42	4.39	14.85	5.52	9.86	9.57	12.82	28.62	10.84	34.66	39.59	40.87	8.56	12.36
MPR	49.11	45.52	77.38	49.33	47.45	48.41	51.33	49.54	48.94	47.33	45.72	49.44	54.98	55.55	60.32	5.86	52.58	56.38
MPS	34.03	46.83	49.32	34.68	32.36	33.33	29.90	34.46	30.72	30.86	24.94	38.22	23.59	26.30	48.05	69.90	37.50	41.30
MPU	13.22	25.79	17.13	16.18	13.93	12.51	21.02	13.64	13.01	14.66	14.19	19.79	2.82	19.33	39.30	51.83	16.68	20.48
MPV	49.49	46.26	72.07	45.47	46.91	48.79	42.06	49.92	47.87	47.68	45.72	39.45	54.41	45.47	39.76	64.01	52.96	56.76
MPX		17.27	29.86	7.72	3.08	0.70	12.92	1.83	6.52	6.04	8.92	11.28	9.58	10.66	30.35	43.25	4.87	8.67
PCS	17.27		42.19	14.62	14.74	16.57	16.81	17.70	15.66	15.53	14.45	15.47	23.04	17.66	30.84	39.65	20.74	24.54
PRN	29.86	42.19		36.77	30.64	29.15	37.55	30.28	28.96	29.56	30.92	36.81	29.74	36.16	64.52	71.52	33.32	37.12
PY	7.72	14.62	36.77		5.53	7.02	6.56	8.15	2.48	3.23	1.44	7.41	13.39	9.46	26.66	43.47	11.19	14.99
PZC	3.08	14.74	30.64	5.53		2.38	10.39	3.51	3.61	3.70	6.40	8.76	11.31	8.70	30.31	41.59	6.55	10.35
PZG	0.70	16.57	29.15	7.02	2.38		12.21	1.13	5.81	5.34	8.22	10.57	8.87	9.96	29.65	42.55	4.17	7.97
PZK	12.92	16.81	37.55	6.56	10.39	12.21		13.34	9.50	8.71	4.79	3.33	18.24	8.93	21.88	45.47	16.38	20.18
PZM	1.83	17.70	30.28	8.15	3.51	1.13	13.34		6.94	6.47	9.35	11.70	10.00	11.09	30.78	43.68	3.04	6.84
PZO	6.52	15.66	28.96	2.48	3.61	5.81	9.50	6.94		0.91	4.25	8.28	10.27	5.63	27.97	43.08	9.98	13.78
PZS	6.04	15.53	29.56	3.23	3.70	5.34	8.71	6.47	0.91		3.49	8.23	8.37	5.77	29.87	41.47	9.51	13.31
RCD	8.92	14.45	30.92	1.44	6.40	8.22	4.79	9.35	4.25	3.49		6.12	11.55	3.83	25.46	39.86	12.39	16.19
RCJ	11.28	15.47	36.81	7.41	8.76	10.57	3.33	11.70	8.28	8.23	6.12		17.29	10.07	21.02	43.58	14.74	18.54
RCM	9.58	23.04	29.74	13.39	11.31	8.87	18.24	10.00	10.27	8.37	11.55	17.29		10.36	37.85	49.12	13.04	16.84
RCP	10.66	17.66	36.16	9.46	8.70	9.96	8.93	11.09	5.63	5.77	3.83	10.07	10.36		25.96	49.69	14.13	17.93
SMP	30.35	30.84	64.52	26.66	30.31	29.65	21.88	30.78	27.97	29.87	25.46	21.02	37.85	25.96		54.46	33.82	37.62
SPI	43.25	39.65	71.52	43.47	41.59	42.55	45.47	43.68	43.08	41.47	39.86	43.58	49.12	49.69	54.46		46.72	50.52
SRK	4.87	20.74	33.32	11.19	6.55	4.17	16.38	3.04	9.98	9.51	12.39	14.74	13.04	14.13	33.82	46.72		3.80
TTR	8.67	24.54	37.12	14.99	10.35	7.97	20.18	6.84	13.78	13.31	16.19	18.54	16.84	17.93	37.62	50.52	3.80	

Lampiran 2 Script M-File Program

```
tic;
clc;
clear;
%-----Data Umum-----
jumlah_kendaraan = 6;
jumlah_customer = xlsread('input.xlsm',1,'B3');
kapasitas_maks = 75;
demand = xlsread('input.xlsm',1, 'E6:E70');
jarak = xlsread('input.xlsm',2, 'C3:AX50');
jarak_maks = inf;

%-----Time Windows-----
waktu_berangkat = 'Jan-0-0000 04:00';
batas_awal = ['Jan-0-0000 05:00';'Jan-0-0000 05:00'];
batas_akhir = ['Jan-0-0000 09:00';'Jan-0-0000 09:00'];
service_time = ['Jan-0-0000 00:10';'Jan-0-0000 00:10'];
```

Lampiran 2 Script M-File Program (lanjutan)

```
10'; 'Jan-0-0000 00:10'; 'Jan-0-0000 00:10'; 'Jan-0-0000 00:  ↵
10'; 'Jan-0-0000 00:10'; 'Jan-0-0000 00:10'];
kecepatan_kendaraan = 45; %dalam km/jam

%-----Ubah Data Time Window jadi Num-----
waktu_berangkat_num = datenum(waktu_berangkat);
batas_awal_num = datenum(batas_awal);
batas_akhir_num = datenum(batas_akhir);
service_time_num = datenum(service_time);

%-----Variabel-----
ukuran_populasi = xlsread('input.xlsx',3,'C1');
jumlah_solusi_tetangga_maks = xlsread('input.xlsx',3,'C2');
panjang_tabu_maks = xlsread('input.xlsx',3,'C3');
iterasi_maks = xlsread('input.xlsx',3,'C4');
stop_iterasi_maks = xlsread('input.xlsx',3,'C5');

%-----Pencarian Solusi Awal-----
%-----Bentuk Satu Rute Besar(Nearest Neighbourhood)-----
nomor_rute = 1;
konsumen(1:jumlah_customer+1)=0;
konsumen(1) = 1;
awal = 1;
for i = 1:jumlah_customer
    nilai_min = inf;
    for j = 1:jumlah_customer + 1
        if jarak(awal,j) < nilai_min & konsumen(j)~= 1 & jarak ↵
(awal,j) > 0
            nilai_min = jarak(awal,j);
            urutan_min = j;
        end
    end
    konsumen(urutan_min) = 1;
    solusi_rute(nomor_rute,i) = urutan_min - 1;
    awal = urutan_min;
end

%-----Permutasi Menjadi Jumlah Populasi-----
for nomor_rute = 2:ukuran_populasi
    solusi_rute(nomor_rute, 1:jumlah_customer)= ↵ randperm
```

Lampiran 2 Script M-File Program (lanjutan)

```
(jumlah_customer); %generate random
end

%-----Pecah tiap rute dan hitung jarak serta
kapasitas-----
for nomor_rute = 1:ukuran_populasi
    solusi_vrp(nomor_rute,1) = 0;
    jarak_total = 0;
    kapasitas_total = 0;
    urutan = 2;
    nomor_kendaraan = 1;
    konsumen_terlayani = 0;
    jam_sekarang = waktu_berangkat_num;
    for i = 1:jumlah_customer
        jam_sekarang = jam_sekarang + service_time_num
(solusi_rute(nomor_rute, i)); %lama service
        jarak_total = jarak_total+jarak(solusi_vrp(nomor_rute,
urutan - 1) + 1, solusi_rute(nomor_rute,i)+1);
        kapasitas_total = kapasitas_total + demand(solusi_rute
(nomor_rute,i));
        jam_sekarang = jam_sekarang + datenum([0 0 0 (jarak
(solusi_vrp(nomor_rute, urutan - 1) + 1, solusi_rute(nomor_rute,
i) + 1)/kecepatan_kendaraan) 0 0]);
        if jam_sekarang < batas_awal_num(solusi_rute(nomor_rute,
i)) %jika sampai sebelum batas awal TW
            jam_sekarang = batas_awal_num(solusi_rute(nomor_rute,
i));
        end
        if jarak_total > jarak_maks |kapasitas_total >
kapasitas_maks|jam_sekarang > batas_akhir_num(solusi_rute
(nomor_rute, i));
            jumlah_konsumen(nomor_rute, nomor_kendaraan) =
konsumen_terlayani;
            jarak_kendaraan(nomor_rute, nomor_kendaraan) =
jarak_total - jarak(solusi_vrp(nomor_rute, urutan - 1) + 1,
solusi_rute(nomor_rute, i) + 1) + jarak(solusi_vrp(nomor_rute,
urutan - 1) + 1, 1);
            kapasitas_kendaraan(nomor_rute, nomor_kendaraan) =
kapasitas_total - demand(solusi_rute(nomor_rute, i));
            nomor_kendaraan = nomor_kendaraan + 1;
            solusi_vrp(nomor_rute, urutan) = 0;
            konsumen_terlayani = 0;
            jam_sekarang = waktu_berangkat_num + datenum([0 0 0
(jarak(solusi_vrp(nomor_rute, urutan - 1) + 1, solusi_rute
(nomor_rute, i) + 1)/kecepatan_kendaraan) 0 0]);
            jarak_total = jarak(solusi_vrp(nomor_rute, urutan) +
1, solusi_rute(nomor_rute, i) + 1);
            kapasitas_total = demand(solusi_rute(nomor_rute, i));
```

Lampiran 2 Script M-File Program (lanjutan)

```
        urutan = urutan + 1;
    end
    konsumen_terlayani = konsumen_terlayani + 1;
    solusi_vrp(nomor_rute, urutan) = solusi_rute(nomor_rute,
i);
    urutan = urutan + 1;
    if i == jumlah_customer
        jarak_total = jarak_total+jarak(solusi_rute
(nomor_rute,i)+1, 1);
    end
    end
    jumlah_konsumen(nomor_rute, nomor_kendaraan) =
konsumen_terlayani;
    jarak_kendaraan(nomor_rute, nomor_kendaraan) = jarak_total;
    kapasitas_kendaraan(nomor_rute, nomor_kendaraan) =
kapasitas_total;
    jumlah_kendaraan(nomor_rute) = nomor_kendaraan;
    jarak_vrp(nomor_rute) = sum(jarak_kendaraan(nomor_rute,1:
jumlah_kendaraan(nomor_rute)));
end

%-----Catat rute terbaik-----
[,jumlah_kendaraan_terbaik] = find(jumlah_kendaraan == min
(jumlah_kendaraan));
[,jarak_vrp_terbaik] = jumlah_kendaraan_terbaik(find(jarak_vrp
(jumlah_kendaraan_terbaik)==min(jarak_vrp
(jumlah_kendaraan_terbaik))));
var_dummy = jumlah_konsumen';
var_dummy(var_dummy == 0) = inf;
var_dummy_2 = min(var_dummy);
[,nomor_rute_terbaik] = jarak_vrp_terbaik(find(var_dummy_2
(jarak_vrp_terbaik) == min(var_dummy_2(jarak_vrp_terbaik))));
nomor_rute_terbaik = min(nomor_rute_terbaik);

%-----Jadikan Solusi Terbaik Menjadi Solusi
Awal-----
solusi_vrp_terbaik_akhir(1,:) = solusi_vrp(nomor_rute_terbaik,:);
jarak_kendaraan_terbaik_akhir(1,:) = jarak_kendaraan
(nomor_rute_terbaik,:);
kapasitas_kendaraan_terbaik_akhir(1,:) = kapasitas_kendaraan
(nomor_rute_terbaik,:);
jumlah_konsumen_terbaik_akhir(1,:) = jumlah_konsumen
(nomor_rute_terbaik,:);
jumlah_kendaraan_terbaik_akhir(1) = jumlah_kendaraan
(nomor_rute_terbaik);
jarak_vrp_terbaik_akhir(1) = jarak_vrp(nomor_rute_terbaik);

%-----Daftar            Tabu            Kosong-----
```

Lampiran 2 Script M-File Program (lanjutan)

```
[i,ukuran_solusi_vrp] = size(solusi_vrp);
tabu_list(1:panjang_tabu_maks, ukuran_solusi_vrp) = 0;

%-----ALGORITMA TABU ↙
SEARCH-----
solusi_vrp_tetangga = [];
jumlah_konsumen_tetangga = [];
jumlah_kendaraan_tetangga = [];
kapasitas_kendaraan_tetangga = [];
iterasi = 0;
stop_iterasi = 0;
urutan_tabu = 0;
while iterasi < iterasi_maks & stop_iterasi < stop_iterasi_maks
    iterasi = iterasi + 1;
    nomor_solusi_tetangga = 0;
    while nomor_solusi_tetangga < jumlah_solusi_tetangga_maks
        nomor_rute = 1;
        solusi_vrp(nomor_rute, :) = solusi_vrp_terbaik_akhir ↙
(1,:);
        nomor_NS = fix(rand*3) + 1; %pilih Neighbourhood ↙
        Structure Random
        [i,ukuran_solusi_vrp] = size(solusi_vrp_terbaik_akhir ↙
(1,:));
        awal = 1;
        akhir = 1;
        switch nomor_NS
            case 1 %NS insert
                while akhir == awal |awal == ↙
ukuran_solusi_vrp|akhir == ukuran_solusi_vrp|solusi_vrp ↙
(nomor_rute, awal) == 0|solusi_vrp(nomor_rute,akhir)==0
                    awal = fix(rand*ukuran_solusi_vrp)+ 1;
                    akhir = fix(rand*ukuran_solusi_vrp)+ 1;
                end
                if awal < akhir
                    solusi_vrp_sementara = solusi_vrp(nomor_rute, ↙
([1:awal-1,awal+1 : akhir,awal,akhir + 1:ukuran_solusi_vrp]));
                else
                    solusi_vrp_sementara = solusi_vrp(nomor_rute, ↙
([1:akhir,awal,(akhir+1):(awal-1),(awal+1) : ↙
ukuran_solusi_vrp]));
                end
            case 2 %N-2opt
                nol_1 = 1;
                nol_2 = 1;
                while nol_1 == nol_2|akhir<=awal|solusi_vrp ↙
(nomor_rute, awal)==0|solusi_vrp(nomor_rute, akhir) == 0
                    awal = fix(rand*ukuran_solusi_vrp) + 1;
```

Lampiran 2 Script M-File Program (lanjutan)

```
        akhir = fix (rand*ukuran_solusi_vrp) + 1;
        nol_1 = awal + min(find(solusi_vrp
(nomor_rute, awal:ukuran_solusi_vrp) == 0)) - 1;
        if isempty(nol_1)
            nol_1 = ukuran_solusi_vrp;
        end
        nol_2 = akhir + min(find(solusi_vrp
(nomor_rute, akhir:ukuran_solusi_vrp) == 0)) - 1;

        if isempty(nol_2)
            nol_2 = ukuran_solusi_vrp;
        end
    end
    solusi_vrp_sementara = solusi_vrp(nomor_rute,
([1:awal-1, akhir:nol_2,nol_1+1:akhir-1, awal:nol_1, nol_2+1:
ukuran_solusi_vrp]));
    case 3 %N-change
        while akhir<=awal|solusi_vrp(nomor_rute, awal) ==
0 |solusi_vrp(nomor_rute, akhir) == 0;
            awal = fix(rand*ukuran_solusi_vrp) + 1;
            akhir = fix (rand*ukuran_solusi_vrp) + 1;
        end
        solusi_vrp_sementara = solusi_vrp(nomor_rute,([1:
awal-1,akhir,awal+1:akhir-1,awal,akhir+1:ukuran_solusi_vrp]));
    end
    %cek feasible atau tidak?
    [i,ukuran_solusi_vrp]= size(solusi_vrp_sementara);
    letak_nol = find(solusi_vrp_sementara == 0);
    [i, jumlah_nol] = size(letak_nol);
    solusi_tetangga_feasible = 0;
    nol_terakhir = 0;
    if letak_nol(jumlah_nol) == ukuran_solusi_vrp;
        jumlah_nol = jumlah_nol - 1;
        nol_terakhir = 1;
    end
    for i = 1:jumlah_nol
        if i ~=jumlah_nol
            akhir_rute = letak_nol(i+1)-1;
        else
            akhir_rute = ukuran_solusi_vrp - nol_terakhir;
        end
        jarak_total = 0;
        kapasitas_total = 0;
        nomor_kendaraan = i;
        jam_sekarang = waktu_berangkat_num;
        for j = letak_nol(i):(akhir_rute - 1)
            jarak_total = jarak_total+jarak
(solusi_vrp_sementara(j)+1, solusi_vrp_sementara(j+1)+1);
```

Lampiran 2 Script M-File Program (lanjutan)

```
        kapasitas_total = kapasitas_total + demand
(solusi_vrp_sementara(j+1));
        jam_sekarang = jam_sekarang + datenum([000 (jarak
(solusi_vrp_sementara(j)+1, solusi_vrp_sementara(j+1)+1)
/kecepatan_kendaraan) 00]);
        if jam_sekarang < batas_awal_num
(solusi_vrp_sementara(j+1))
            jam_sekarang = batas_awal_num
(solusi_vrp_sementara(j+1));
        end
        jam_sekarang = jam_sekarang + service_time_num
(solusi_vrp_sementara(j+1));
        if jam_sekarang > batas_akhir_num
(solusi_vrp_sementara(j+1))
            solusi_tetangga_feasible =
solusi_tetangga_feasible+1;
        end
    end
    jarak_total = jarak_total + jarak
(solusi_vrp_sementara(akhir_rute) + 1, 1);
    if jarak_total > jarak_maks | kapasitas_total >
kapasitas_maks
        solusi_tetangga_feasible =
solusi_tetangga_feasible+1;
    else
        jumlah_konsumen_sementara(nomor_kendaraan) =
akhir_rute - letak_nol(i);
        jarak_kendaraan_sementara(nomor_kendaraan) =
jarak_total;
        kapasitas_kendaraan_sementara(nomor_kendaraan) =
kapasitas_total;
        jumlah_kendaraan_sementara = nomor_kendaraan;
        jarak_vrp_sementara = sum
(jarak_kendaraan_sementara);
    end
    end
    if solusi_tetangga_feasible == 0 %solusi feasible,
masukkan solusi tetangga
        nomor_solusi_tetangga = nomor_solusi_tetangga + 1;
        solusi_vrp_tetangga(nomor_solusi_tetangga,1:
ukuran_solusi_vrp) = solusi_vrp_sementara(1:ukuran_solusi_vrp);
        jumlah_konsumen_tetangga(nomor_solusi_tetangga,1:
jumlah_nol) = jumlah_konsumen_sementara(1:jumlah_nol);
        jarak_kendaraan_tetangga(nomor_solusi_tetangga, 1:
jumlah_nol) = jarak_kendaraan_sementara(1:jumlah_nol);
        kapasitas_kendaraan_tetangga(nomor_solusi_tetangga,
1:jumlah_nol) = kapasitas_kendaraan_sementara(1:jumlah_nol);
        jumlah_kendaraan_tetangga(nomor_solusi_tetangga) =
```

Lampiran 2 Script M-File Program (lanjutan)

```
jumlah_kendaraan_sementara;
        jarak_vrp_tetangga(nomor_solusi_tetangga) = sum
(jarak_kendaraan_tetangga(nomor_solusi_tetangga,:));
    end
end

tabu_ok = 0;
iterasi_tabu = 0;
%pilih solusi terbaik
while tabu_ok == 0 & iterasi_tabu <
jumlah_solusi_tetangga_maks
    iterasi_tabu = iterasi_tabu + 1;
    [,jumlah_kendaraan_terbaik_tetangga] = find
(jumlah_kendaraan_tetangga == min(jumlah_kendaraan_tetangga));
    [,jarak_vrp_terbaik_tetangga] =
jumlah_kendaraan_terbaik_tetangga(find(jarak_vrp_tetangga
(jumlah_kendaraan_terbaik_tetangga) == min(jarak_vrp_tetangga
(jumlah_kendaraan_terbaik_tetangga))));
    var_dummy = jumlah_konsumen_tetangga';
    var_dummy(var_dummy == 0) = inf;
    var_dummy_2 = min(var_dummy);
    [,nomor_rute_terbaik_tetangga] =
jarak_vrp_terbaik_tetangga(find(var_dummy_2
(jarak_vrp_terbaik_tetangga) == min(var_dummy_2
(jarak_vrp_terbaik_tetangga))));
    nomor_rute_terbaik_tetangga = min
(nomor_rute_terbaik_tetangga);
    %cek tabu atau tidak?
    hasil_test_tabu = 0;
    for test_tabu = 1 :panjang_tabu_maks
        hasil_test_tabu = hasil_test_tabu + isequal
(solusi_vrp_tetangga(nomor_rute_terbaik_tetangga), tabu_list
(test_tabu,:));
    end
    if hasil_test_tabu == 0
        tabu_ok = 1;
    else
        if hasil_test_tabu > 0
            %copy ke konsumen terbaik
            solusi_vrp_terbaik_akhir(2,:) =
solusi_vrp_tetangga(nomor_rute_terbaik_tetangga,:);
            jarak_kendaraan_terbaik_akhir(2,:) =
jarak_kendaraan_tetangga(nomor_rute_terbaik_tetangga,:);
            kapasitas_kendaraan_terbaik_akhir(2,:) =
kapasitas_kendaraan_tetangga(nomor_rute_terbaik_tetangga,:);
            jumlah_konsumen_terbaik_akhir(2,:) =
jumlah_konsumen_tetangga(nomor_rute_terbaik_tetangga,:);
            jumlah_kendaraan_terbaik_akhir(2) =
```

Lampiran 2 Script M-File Program (lanjutan)

```
jumlah_kendaraan_tetangga(nomor_rute_terbaik_tetangga)
    jarak_vrp_terbaik_akhir(2) = jarak_vrp_tetangga
(nomor_rute_terbaik_tetangga);
    %lihat lebih baik mana
    [,jumlah_kendaraan_terbaik_tabu] = find
(jumlah_kendaraan_terbaik_akhir == min
(jumlah_kendaraan_terbaik_akhir));
    [,jarak_vrp_terbaik_tabu] =
jumlah_kendaraan_terbaik_tabu(find(jarak_vrp_terbaik_akhir
(jumlah_kendaraan_terbaik_tabu) == min(jarak_vrp_terbaik_akhir
(jumlah_kendaraan_terbaik_tabu)));
    var_dummy = jumlah_konsumen_terbaik_akhir';
    var_dummy(var_dummy == 0) = inf;
    var_dummy_2 = min(var_dummy);
    [,nomor_rute_terbaik_tabu] =
jarak_vrp_terbaik_tabu(find(var_dummy_2(jarak_vrp_terbaik_tabu)==
min(var_dummy_2(jarak_vrp_terbaik_tabu)));
    nomor_rute_terbaik_tabu = min
(nomor_rute_terbaik_tabu);
    if nomor_rute_terbaik_tabu == 2 %memenuhi
kriteria aspirasi
        tabu_ok = 1;
    else
        jumlah_kendaraan_tetangga
(nomor_rute_terbaik_tetangga) = inf;
    end
end
end
    %rekam sebagai solusi vrp 2
    solusi_vrp_2 = solusi_vrp_tetangga
(nomor_rute_terbaik_tetangga,:);
    jarak_kendaraan_2 = jarak_kendaraan_tetangga
(nomor_rute_terbaik_tetangga,:);
    kapasitas_kendaraan_2 = kapasitas_kendaraan_tetangga
(nomor_rute_terbaik_tetangga,:);
    jumlah_konsumen_2 = jumlah_konsumen_tetangga
(nomor_rute_terbaik_tetangga,:);
    jumlah_kendaraan_2 = jumlah_kendaraan_tetangga
(nomor_rute_terbaik_tetangga);
    jarak_vrp_2 = jarak_vrp_tetangga
(nomor_rute_terbaik_tetangga);
    %rekam sebagai tabu
    urutan_tabu = urutan_tabu + 1;
    if urutan_tabu > panjang_tabu_maks
        urutan_tabu = urutan_tabu - panjang_tabu_maks;
    end
    tabu_list(urutan_tabu,:) = solusi_vrp_2;
```

Lampiran 2 Script M-File Program (lanjutan)

```
%cek lebih baik dari solusi terbaik?
solusi_vrp_terbaik_akhir(2,:) = solusi_vrp_2;
jarak_kendaraan_terbaik_akhir(2,:) = 0;
kapasitas_kendaraan_terbaik_akhir(2,:) = 0;
jumlah_konsumen_terbaik_akhir(2,:) = 0;
jarak_kendaraan_terbaik_akhir(2, 1:jumlah_kendaraan_2) =
jarak_kendaraan_2(1:jumlah_kendaraan_2);
kapasitas_kendaraan_terbaik_akhir(2, 1:jumlah_kendaraan_2) =
kapasitas_kendaraan_2(1:jumlah_kendaraan_2);
jumlah_konsumen_terbaik_akhir(2, 1:jumlah_kendaraan_2) =
jumlah_konsumen_2(1:jumlah_kendaraan_2);
jumlah_kendaraan_terbaik_akhir(2) = jumlah_kendaraan_2;
jarak_vrp_terbaik_akhir(2) = jarak_vrp_2;

[,jumlah_kendaraan_terbaik_2] = find
(jumlah_kendaraan_terbaik_akhir == min
(jumlah_kendaraan_terbaik_akhir));
[,jarak_vrp_terbaik_2] = jarak_vrp_terbaik_2(find
(jarak_vrp_terbaik_akhir(jumlah_kendaraan_terbaik_2)==min
(jarak_vrp_terbaik_akhir(jumlah_kendaraan_terbaik_2))));
var_dummy = jumlah_konsumen_terbaik_akhir';
var_dummy(var_dummy == 0) = inf;
var_dummy_2 = min(var_dummy);
[,nomor_rute_terbaik_2] = jarak_vrp_terbaik_2(find
(var_dummy_2(jarak_vrp_terbaik_2) == min(var_dummy_2
(jarak_vrp_terbaik_2))));
nomor_rute_terbaik_2 = min(nomor_rute_terbaik_2);

if nomor_rute_terbaik_2 == 2 %lebih baik hasil iterasi,
update solusi terbaik
stop_iterasi = 0;
solusi_vrp_terbaik_akhir(1,:) = solusi_vrp_terbaik_akhir
(2,:);
jarak_kendaraan_terbaik_akhir(1,:) =
jarak_kendaraan_terbaik_akhir(2,:);
kapasitas_kendaraan_terbaik_akhir(1,:) =
kapasitas_kendaraan_terbaik_akhir(2,:);
jumlah_konsumen_terbaik_akhir(1,:) =
jumlah_konsumen_terbaik_akhir(2,:);
jumlah_kendaraan_terbaik_akhir(1) =
jumlah_kendaraan_terbaik_akhir(2);
jarak_vrp_terbaik_akhir(1) = jarak_vrp_terbaik_akhir(2);
else stop_iterasi = stop_iterasi + 1;
end
end
```

Lampiran 2 Script M-File Program (lanjutan)

```
-----Tampilkan Solusi-----  
disp('-----Solusi Akhir-----')  
disp('Solusi Rute :');  
disp(solusi_vrp_terbaik_akhir(1,:));  
disp('Jumlah Kendaraan yang Digunakan :');  
disp(jumlah_kendaraan_terbaik_akhir(1));  
disp('Muatan Tiap Kendaraan (krat) :');  
disp(kapasitas_kendaraan_terbaik_akhir(1,:));  
disp('Jumlah Konsumen yang dilayani Tiap Kendaraan :');  
disp(jumlah_konsumen_terbaik_akhir(1,:));  
disp('Jarak Tempuh Tiap Kendaraan (km) :');  
disp(jarak_kendaraan_terbaik_akhir(1,:));  
disp('Jarak Tempuh Total (km) :');  
disp(jarak_vrp_terbaik_akhir(1));  
toc
```