

**RANCANG BANGUN *VEHICLE ROUTING PROBLEM*
MENGUNAKAN ALGORITMA *TABU SEARCH***

SKRIPSI

Untuk memenuhi sebagai persyaratan

Mencapai derajat sarjana S-1



Disusun oleh

Sulistiono

11610033

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2015



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi

Lamp :

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Sulistiono

NIM : 11610033

Judul Skripsi : Rancang Bangun *Vehicle Routing Problem* Menggunakan Algoritma *Tabu Search*

sudah dapat diajukan kembali kepada Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Matematika.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 12 Oktober 2015

Pembimbing

Noor Saif Muhammad Mussafi, M.Sc.

NIP. 19820617 200912 1 005



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/3506/2015

Skripsi/Tugas Akhir dengan judul : Rancang Bangun *Vehicle Routing Problem* Menggunakan Algoritma *Tabu Search*

Yang dipersiapkan dan disusun oleh :

Nama : Sulistiono

NIM : 11610033

Telah dimunaqasyahkan pada : 29 Oktober 2015

Nilai Munaqasyah : A

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Noor Saif Muh. Mussafi, M.Sc
NIP. 19820617 200912 1 005

Penguji I

Much. Abrori, S.Si, M.Kom
NIP.19720423 199903 1 003

Penguji II

Moh. Farhan Qudratullah, M.Si
NIP.19790922 200801 1 011

Yogyakarta, 11 November 2015

UIN Sunan Kalijaga

Fakultas Sains dan Teknologi

Dekan

Dr. Maizer Said Nahdi, M.Si
NIP. 19550427 198403 2 001

SURAT KETERANGAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Sulistiono
NIM : 11610033
Program Studi : Matematika
Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi saya yang berjudul “Rancang Bangun *Vehicle Routing Problem* Menggunakan Algoritma *Tabu Search*” merupakan hasil penelitian saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan sepanjang pengetahuan saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 12 Oktober 2015

Penulis,

Sulistiono
NIM.11610033



HALAMAN PERSEMBAHAN

**Atas karunia Allah Subhanahu Wata'ala
Karya ini penulis persembahkan kepada:**

**Kedua Orang Tuaku tercinta
Almarhum Bapak Soedharsono Patria & Ibu Wiwik Ariningsih**

**Kakak-kakaku Bambang Dewanjaya, Heri Dewandaru, Totok Dewanto,
Sulistiyantoro, dan Sulistiawan**

**Teman-Teman PAL
Lukman, Aldi, Syaqui, Fuad, Bang Dayat, Wachid, Eruit, Taufan,
Ridwan, Juni, Dwi (Uthe), Fuji (Fufu), dan Dina**

**dan Almamaterku
Program Studi Matematika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta**

MOTTO

“Ing ngarsa sung tuladha, ing madya mangun karsa, tut wuri handayani”

**(“Di depan memberi contoh, di tengah memberi semangat, di belakang
memberi kekuatan”)**

(Ki Hajar Dewantara)

“Religion without science is blind, science without religion is lame”

(Albert Einstein)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan segala rahmat dan hidayah-Nya, sehingga skripsi yang berjudul “Rancang Bangun *Vehicle Routing Problem* menggunakan Algoritma *Tabu Search*” dapat terselesaikan guna memenuhi syarat memperoleh gelar kesarjanaan di Jurusan Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.

Shalawat serta salam senantiasa tercurahkan kepada Nabi besar Muhammad SAW, yang membawa umat manusia dari zaman kegelapan menuju zaman yang terang seperti saat ini. Penulis menyadari skripsi ini tidak akan selesai tanpa motivasi, bantuan, bimbingan dan arahan dari berbagai pihak. Oleh karena itu, dengan kerendahan hati penulis mengucapkan rasa terimakasih kepada:

1. Ibu Dr. Maizer Said Nahdi, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
2. Bapak Dr. M. Wakhid Musthofa, M.Si, selaku Ketua Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
3. Bapak Noor Saif Muhammad Mussafi, M.Sc, selaku dosen pembimbing skripsi, yang selalu meluangkan waktunya dalam membimbing, memotivasi, serta mengarahkan sehingga skripsi ini dapat terselesaikan.
4. Bapak /Ibu Dosen dan Staf Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta atas ilmu, bimbingan dan pelayanan selama perkuliahan dan penyusunan skripsi.

5. Ayahanda terkasih Alm. Soedharsono Patria dan Ibunda tersayang Wiwik Ariningsih yang senantiasa memberikan kasih sayang, motivasi, doa dan segala pengorbanan untuk memperjuangkan penulis. Karya ini khusus penulis tujukan untuk Ayah dan Ibu tercinta.
6. Kakak-kakakku Bambang Dewanjaya, Heri Dewandaru, Totok Dewanto, Sulistiyantoro, dan Sulistiawan yang memberikan nasehat dan dorongan semangat untuk penulis.
7. Sahabat Trio KPT (Rizqie Adhitrisna Budhi dan R. Wendy Anjar), Wahyu Sanjaya, Rana Yuliawiyata, dan Yopi Setiawan yang telah memberikan hari-hari indah dalam perjalanan hidup penulis semenjak TK hingga sekarang.
8. Sahabat PAL (Lukman, Aldi, Syauqi, Fuad, Bang Dayat, Wachid, Eruit, Taufan, Ridwan, Juni, Dwi (Uthe), Fuji (Fufu), dan Dina) terima kasih atas canda dan tawa yang menghiasi hari-hari indah penulis. Kenangan bersama kalian tidak akan penulis lupakan.
9. Kepada teman-teman matematika 2011 yang selalu memberikan dukungan dan motivasi hingga terselesaikannya skripsi.
10. Mas Dedi selaku pimpinan PT Sinergi Bio Natural atas semua bantuan dalam penelitian.
11. Kepada semua pihak yang tidak dapat penulis sebutkan satu per satu, atas doa dan motivasinya yang telah membantu dalam penyusunan skripsi.

Penulis menyadari masih terdapat kesalahan dan kekurangan dalam penulisan skripsi ini, oleh karena itu penulis mengharapkan adanya kritik dan saran

yang membangun dari semua pihak. Akhir kata, penulis berharap semoga skripsi ini bisa bermanfaat dan membantu bagi berbagai pihak.

Yogyakarta, 12 Oktober 2015

Penulis

Sulistiono
NIM. 11610033



DAFTAR ISI

HALAMAN JUDUL	i
PENGESAHAN SKRIPSI	ii
PERSETUJUAN SKRIPSI	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xv
ABSTRAK	xvi

BAB I PENDAHULUAN

A. Latar Belakang.....	1
B. Rumusan Masalah.....	2
C. Tujuan Penelitian	3
D. Batasan Masalah	3
E. Manfaat Penelitian	4
F. Tinjauan Pustaka.....	4
G. Metode Penelitian	7
H. Sistematika penulisan.....	8

BAB II DASAR TEORI

A. Teori Graf	10
1. Definisi Graf	11
2. Jenis-Jenis Graf	12
3. Keterhubungan	16
4. Graf Berbobot	20
5. Graf Berarah Berbobot	20
6. Graf Hamilton	21
B. <i>Travelling Salesman Problem</i> (TSP)	21
C. <i>Vehicle Routing Problem</i> (VRP).....	23
D. <i>Capacitated Vehicle Routing Problem</i> (CVRP).....	26
E. Algoritma <i>Tabu Search</i>	29
F. Penyelesaian <i>Vehicle Routing Problem</i> menggunakan Algoritma <i>Tabu Search</i>	32
G. MATLAB	36
1. GUI(<i>Graphical User Interface</i>) pada MATLAB.....	37

2. Operator Relasi dan Logika	40
3. <i>Statement Control</i> pada MATLAB	40

BAB III PEMBAHASAN

A. Konsep dan Langkah Algoritma <i>Tabu Search</i>	45
1. Konsep Algoritma <i>Tabu Search</i>	45
2. Langkah Algoritma <i>Tabu Search</i>	46
a. Pembentukan Inisial Solusi (Solusi Awal)	47
b. <i>Neighborhood</i>	48
c. <i>Tabu List</i>	48
d. <i>Aspiration Criteria</i> (Kriteria Aspirasi)	49
e. <i>Termination Criteria</i> (Kriteria Pemberhentian)	49
B. Penerapan Algoritma <i>Tabu Search</i> pada <i>Vehicle Routing Problem</i>	51
1. Penerapan Algoritma <i>Tabu Search</i> secara Manual	54
a. Langkah 1.....	55
b. Langkah 2.....	56
c. Langkah 3.....	61
d. Langkah 4.....	62
e. Langkah 5.....	62
2. Rancang Bangun Algoritma <i>Tabu Search</i> dalam Menyelesaikan VRP.....	64

BAB IV PENUTUP

A. Kesimpulan.....	97
B. Saran.....	98

DAFTAR PUSTAKA	100
-----------------------------	-----

LAMPIRAN-LAMPIRAN	103
--------------------------------	-----

DAFTAR TABEL

Tabel 1.1 Perbedaan penelitian.....	6
Tabel 2.1 Tabel nilai IPK rata-rata mahasiswa matematika.....	43
Tabel 3.1 Daftar depot dan pelanggan beserta permintaan	52
Tabel 3.2 Jarak depot ke pelanggan dan antar pelanggan dalam satuan kilometer	54
Tabel 3.3 Solusi awal VRP menggunakan metode <i>Nearest Neighbor</i>	56
Tabel 3.4 Solusi <i>Neighborhood TSP</i> iterasi 1	57
Tabel 3.5 Cara 1 transformasi solusi <i>Neighborhood</i>	58
Tabel 3.6 Cara 2 transformasi solusi <i>Neighborhood</i>	58
Tabel 3.7 Cara 3 transformasi solusi <i>Neighborhood</i>	59
Tabel 3.8 Solusi <i>Neighborhood VRP</i> iterasi 1	60
Tabel 3.9 Solusi <i>Neighborhood VRP</i> terbaik iterasi 1.....	61
Tabel 3.10 <i>Tabu List</i>	62
Tabel 3.11 Solusi optimal VRP	62
Tabel 3.12 Spesifikasi perangkat keras (<i>Hardware</i>)	64
Tabel 3.13 Spesifikasi perangkat lunak (<i>Software</i>)	64
Tabel 3.14 <i>String property static</i> Text18	66
Tabel 3.15 Solusi optimal VRP menggunakan program.....	96

DAFTAR GAMBAR

Gambar 1.1 <i>Flow chart</i> langkah penelitian	8
Gambar 2.1 Jembatan <i>Könisberg</i>	10
Gambar 2.2 Model graf jembatan <i>Könisberg</i>	11
Gambar 2.3 Contoh graf.....	12
Gambar 2.4 Graf nol	12
Gambar 2.5 Contoh graf lengkap K_1, K_2, K_3 dan K_4	13
Gambar 2.6 Contoh graf ganda (<i>Multigraph</i>)	14
Gambar 2.7 Contoh graf semu (<i>Pseudograph</i>).....	15
Gambar 2.8 Contoh graf berarah (<i>Directed Graph</i> atau <i>Digraph</i>).....	15
Gambar 2.9 Graf G_1 terhubung dan G_2 tidak terhubung.....	16
Gambar 2.10 Graf H	17
Gambar 2.11 Graf sederhana H_2	19
Gambar 2.12 Contoh graf berbobot	20
Gambar 2.13 Contoh graf berarah berbobot.....	21
Gambar 2.14 Contoh TSP	23
Gambar 2.15 Metode <i>Relocated</i>	33
Gambar 2.16 Metode <i>Exchange</i>	33

Gambar 2.17 Metode 2-Opt.....	34
Gambar 2.18 GUIMATLAB	38
Gambar 2.19 Tampilan Fig-file	38
Gambar 2.20 Tampilan M.file	39
Gambar 2.21 <i>Output</i>	44
Gambar 3.1 Langkah Algoritma <i>Tabu Search</i>	46
Gambar 3.2 <i>Flowchart</i> langkah-langkah Algoritma <i>Tabu Search</i>	50
Gambar 3.3 Peta pendistribusian bioseptik Kota Yogyakarta.....	53
Gambar 3.4 Solusi optimal pendistribusian bioseptik Kota Yogyakarta.....	63
Gambar 3.5 <i>Figure</i> program.....	69
Gambar 3.6 Pembuatan <i>background</i> program.....	69
Gambar 3.7 BG_GUI.jpg	71
Gambar 3.8 mapyogyakarta.jpg.....	72
Gambar 3.9 Tampilan <i>help</i>	88
Gambar 3.10 Tampilan awal program	89
Gambar 3.11 Form <i>input</i> jarak	91
Gambar 3.12 <i>Input</i> matriks jarak	92
Gambar 3.13 Form <i>input</i> data.....	93
Gambar 3.14 Tampilan hasil proses perhitungan	94
Gambar 3.15 <i>Output</i> jumlah permintaan pelanggan per rute	95
Gambar 3.16 <i>Output</i> total jarak kendaraan per rute	96

DAFTAR LAMPIRAN

Lampiran 1. Iterasi.....	103
1. Iterasi 1	103
2. Iterasi 2	105
3. Iterasi 3	107
4. Iterasi 4	109
5. Iterasi 5	111
6. Iterasi 6	113
7. Iterasi 7	115
8. Iterasi 8	117
9. Iterasi 9	119
10. Iterasi 10	121
Lampiran 2. <i>Source Code</i> M.Files.....	123

ABSTRAK

RANCANG BANGUN *VEHICLE ROUTING PROBLEM* MENGGUNAKAN ALGORITMA *TABU SEARCH*

Oleh:
Sulistiono
NIM.11610033

Logistik berperan penting dalam dunia industri. Pendistribusian produk merupakan salah satu kegiatan logistik. Kegiatan pendistribusian produk ini memiliki berbagai kendala, seperti keterbatasan jumlah dan kapasitas kendaraan milik perusahaan, perbedaan jumlah permintaan konsumen, dan tersebarnya lokasi konsumen. Salah satu usaha yang dapat dilakukan perusahaan untuk mengoptimalkan pendistribusian produk adalah meminimalkan biaya transportasi melalui penentuan rute optimal kendaraan.

Permasalahan penentuan rute optimal kendaraan dapat direpresentasikan menggunakan graf berarah yang memiliki bobot dan disebut dengan VRP (*Vehicle Routing Problem*). Depot dan pelanggan dinyatakan sebagai simpul, jalan dinyatakan sebagai busur, dan bobot dinyatakan sebagai jarak antar simpul. Salah satu variasi VRP adalah *Capacitated Vehicle Routing Problem* (CVRP), yaitu VRP dengan kendala kapasitas kendaraan. Kasus CVRP tersebut dapat diselesaikan dengan menggunakan Algoritma *Tabu Search*. Cara kerja Algoritma *Tabu Search* dimulai dengan penentuan *initial solution* menggunakan *Nearest Neighbor*, evaluasi *move* menggunakan metode 2-Opt, *Relocated*, dan *Exchange*, *update Tabu List*, kemudian apabila kriteria pemberhentian terpenuhi maka proses Algoritma *Tabu Search* berhenti jika tidak, maka kembali pada evaluasi *move*. Proses perhitungan Algoritma *Tabu Search* dilakukan secara manual dan rancang bangun menggunakan MATLAB pada PT Sinergi Bio Natural.

Berdasarkan proses perhitungan diperoleh dua solusi optimal dengan total jarak optimal sebesar 101,1 km. Solusi optimal pertama diperoleh menggunakan perhitungan manual terdiri dari tiga rute, pertama dari Depot-Chacha Milk Tea 3-Perum Sido Mulyo-Chacha Milk Tea 1-Toko Roti & Katering Asli-Happy Land Medical Centre-Chacha Milk Tea 2-Perum Banteng 3-Perum Banteng 2-Depot (29 km), kedua dari Depot-JIH-Depot (9,2 km), ketiga dari Depot-RSU. Holistika Medika-RS.Panti Rini-Perum Cepoko Indah-RSU. Rajawali Citra-RSU. PKU Muhammadiyah Bantul-Hotel Agung Inn-Depot (62,9 km). Solusi optimal kedua menggunakan perhitungan rancang bangun diperoleh rute kedua sama dengan perhitungan manual, sedangkan rute pertama berawal dari Depot-RSU. Holistika Medika-RS. Panti Rini-Perum Cepoko Indah-RSU. Rajawali Citra-RSU. PKU Muhammadiyah Bantul-Hotel Agung Inn-Perum banteng 2-Depot (63,8 km) dan rute ketiga dari Depot-Perum Banteng 3-Chacha Milk Tea 2-Happy Land Medical Center-Toko Roti & Katering Asli-Chacha Milk Tea 1-Perum Sido Mulyo-Chacha Milk Tea 3-Depot (28,1 km).

Kata Kunci: *Vehicle Routing Problem* (VRP), *Capacitated Vehicle Routing Problem* (CVRP), Algoritma *Tabu Search*

BAB 1

PENDAHULUAN

A. Latar Belakang

Pada dunia industri, logistik memiliki peranan penting dalam meningkatkan kinerja suatu perusahaan. Logistik adalah proses pengelolaan yang strategis mulai dari pengadaan, perpindahan hingga penyimpanan barang, bahan baku, dan produk jadi (yang di dalamnya terkait pula aliran informasi) pada perusahaan dan koneksi pemasaran untuk kepentingan mendapatkan keuntungan maksimal saat ini dan masa depan dengan biaya yang efisien dalam rangka pemenuhan kebutuhan konsumen (Christopher,1992). Kemampuan perusahaan untuk mengelola logistik secara efektif dan efisien dapat mempengaruhi biaya dan tingkat pelayanan terhadap konsumen. Pendistribusian produk yang optimal dalam pengelolaan logistik pada perusahaan berperan penting sehingga dapat bersaing dengan perusahaan sejenis lainnya. Pendistribusian produk memiliki berbagai kendala, seperti keterbatasan jumlah dan kapasitas kendaraan milik perusahaan, perbedaan jumlah permintaan konsumen, dan tersebarnya lokasi konsumen. Salah satu usaha yang dapat dilakukan perusahaan untuk mengoptimalkan pendistribusian produk adalah meminimalkan biaya transportasi melalui penentuan rute optimal kendaraan.

Masalah pencarian rute optimal kendaraan dapat diaplikasikan menggunakan graf dan disebut sebagai *Vehicle Routing Problem* (VRP). VRP digolongkan ke dalam *NP-Hard Problem* karena secara teori ataupun praktik pada dunia nyata memiliki permasalahan yang sangat banyak dan kompleks sehingga sulit untuk

dipecahkan. Kasus *NP-Hard* dapat diselesaikan menggunakan dua metode yaitu metode konvensional dan metode heuristik. Metode konvensional kurang efektif karena semua kemungkinan solusi yang ada dicoba sampai salah satu solusi terbaik tercapai. Kelemahan dari metode ini adalah waktu pencarian solusi yang lama apabila jumlah pelanggan yang dicari pada VRP menjadi lebih banyak (Basuki, 2011). Sedangkan metode heuristik memberikan perkiraan solusi yang mendekati solusi optimal sehingga proses perhitungan menjadi lebih cepat dan efisien (Selim Çetiner, 2003). Metode heuristik dibagi menjadi metode heuristik klasik dan metode metaheuristik. Dibandingkan dengan metode heuristik klasik, metaheuristik menunjukkan pencarian solusi yang lebih teliti sehingga dihasilkan solusi yang lebih baik. Salah satu metode metaheuristik adalah Algoritma Tabu Search.

Algoritma *Tabu Search* merupakan salah satu metode terbaik yang dapat diimplementasikan pada VRP dibandingkan metode lain seperti Algoritma *Simulated Annealing*, *Genetic Search*, *Ant System* dan *Neural Network* karena memiliki *running time* yang cukup cepat dengan hasil mendekati solusi optimal. Algoritma *Tabu Search* dapat menuntun prosedur pencarian lokal heuristik untuk menjelajahi daerah solusi di luar titik optimal lokal (Fred Glover dan Rafael Martí, 2006). *Tabu Search* diperkenalkan oleh Fred Glover pada tahun 1986. Algoritma ini menghasilkan solusi awal (*initial solution*) sebagai basis permulaan untuk mendapatkan solusi baru yang lebih baik dari pencarian solusi tetangga yang berbeda-beda. Setiap proses pada Algoritma *Tabu Search* mencegah terjadinya pengulangan solusi yang sama pada iterasi sebelumnya.

Algoritma *Tabu Search* dapat digunakan untuk mencari solusi optimal VRP. Solusi optimal VRP yaitu rute yang memiliki total jarak tempuh minimum dengan mempertimbangkan kapasitas kendaraan. Pembuatan suatu program (rancang bangun) dapat mempercepat proses pencarian solusi optimal pada VRP. Oleh karena itu, program (rancang bangun) Algoritma *Tabu Search* diharapkan dapat memudahkan pencarian solusi optimal VRP yang lebih efektif dan efisien sehingga dirumuskan judul penelitian yaitu “**Rancang Bangun *Vehicle Routing Problem* menggunakan Algoritma *Tabu Search*”.**

B. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dirumuskan permasalahan sebagai berikut:

1. Bagaimana konsep dan langkah Algoritma *Tabu Search* untuk menyelesaikan VRP?
2. Bagaimana pembuatan program (rancang bangun) VRP menggunakan Algoritma *Tabu Search*?
3. Bagaimana penerapan Algoritma *Tabu Search* dalam kasus pendistribusian produk?

C. Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Menjelaskan konsep dan langkah Algoritma *Tabu Search* untuk menyelesaikan VRP.
2. Membuat program (rancang bangun) yang dapat mengaplikasikan Algoritma *Tabu Search* dalam menyelesaikan VRP.

3. Menerapkan Algoritma *Tabu Search* dalam kasus pendistribusian produk.

D. Batasan Masalah

Pembatasan masalah pada penelitian ini yaitu:

1. Model graf yang digunakan untuk satu rute pada setiap kendaraan adalah graf berarah berbobot (*weighted directed graph*) dan graf Hamilton.
2. *Input* yang diperlukan berupa koordinat letak node, jarak antar node, kapasitas kendaraan, permintaan pelanggan, jumlah solusi *Neighborhood*, dan panjang *tabu list*.
3. Jumlah depot dan jumlah kendaraan adalah 1 unit.
4. *Output* yang dihasilkan berupa solusi urutan kunjungan, total jarak terpendek, jumlah rute, jumlah permintaan per rute, dan visualisasi rute dalam koordinat.
5. Program (rancang bangun) menggunakan Matlab.
6. Permasalahan yang diselesaikan adalah *Capacitated Vehicle Routing Problem* (CVRP).
7. Kepadatan lalu lintas dan kondisi jalan diabaikan.

E. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat bermanfaat sebagai berikut :

1. Bagi Mahasiswa
 - a. Memberikan informasi bagaimana cara menyelesaikan VRP dengan Algoritma *Tabu Search*.
 - b. Menambah ilmu pengetahuan secara teoritis dan aplikasi pendistribusian barang pada VRP menggunakan Algoritma *Tabu Search* dengan disertai program (rancang bangun).

2. Bagi Umum (Instansi/Perusahaan)

Memberikan rekomendasi solusi optimal alternatif VRP dengan menggunakan Algoritma *Tabu Search* disertai program (rancang bangun).

F. Tinjauan Pustaka

Penelitian ini menggunakan beberapa literatur baik yang berasal dari buku, skripsi, jurnal penelitian, dan referensi lainnya. Beberapa sumber yang digunakan sebagai acuan pada penelitian ini diantaranya adalah:

1. Skripsi yang berjudul “Penggunaan Metode Heuristik dalam Permasalahan *Vehicle Routing Problem* dan Implementasinya di PT. Nippon Indosari Corpindo” karangan Aji Raditya pada tahun 2009. Skripsi tersebut menjelaskan penyelesaian VRP dengan menggunakan dua fase. Fase pertama menggunakan metode *Nearest Addition Heuristic (Nearest Neighbor)* sebagai *route construction* untuk mencari solusi *feasible* awal. Fase kedua adalah *route improvement* menggunakan metode *2-Opt, or-Opt, Relocate, Exchange, dan Cross*.
2. Penelitian yang dilakukan oleh Vylda Pavela dan Imam Nurhadi P pada tahun 2013 dengan judul “Penyelesaian *Vehicle Routing Problem* dengan Menggunakan Algoritma *Nearest Neighbor* dan *Tabu Search* (Studi Kasus di PT. Nippon Indosari Corpindo)”. Penelitian ini membahas penyelesaian VRP menggunakan Algoritma *Nearest Neighbor* untuk meminimalkan jumlah kendaraan yang beroperasi dan menggunakan Algoritma *Tabu Search* untuk menghasilkan solusi yang optimal serta membandingkan solusi

optimal menggunakan Algoritma *Tabu Search* dengan metode *2-Opt*, *or-Opt*, *Relocate*, *Exchange*, dan *Cross* oleh Aji Raditya (2009).

3. Jurnal penelitian yang dilakukan oleh Berlian Trifal Mahendra dan Sapti Wahyuningsih pada tahun 2012 dengan judul “Analisis Kerja Algoritma *Tabu Search* pada *Vehicle Routing Problem with Backhaul* (VRPB) dengan Perbaikan *2-Opt*”. Penelitian ini berisi penyelesaian *Vehicle Routing Problem with Backhaul* (VRPB) menggunakan Algoritma *Tabu Search* dengan perbaikan *2-Opt*. VRPB memiliki kendala yaitu pelanggan bisa melakukan permintaan berupa pengiriman atau pengambilan barang dari lokasi tertentu. Pencarian solusi optimal pada penelitian ini menggunakan tahap inisialisasi dan pengembangan. Tahap inisialisasi menggunakan metode *Nearest Neighbor* untuk mencari solusi awal kemudian dilakukan pembentukan rute berjenis VRPB. Tahap pengembangan menggunakan hasil solusi awal dari tahap inisialisasi yang dikembangkan dengan cara pertukaran titik antar rute sekaligus pemeriksaan kendala. Setelah diperoleh solusi akhir menggunakan Algoritma *Tabu Search* maka akan dilakukan perbaikan rute dengan menggunakan Algoritma *2-Opt*.

Perbedaan dengan penelitian ini dapat disajikan dalam Tabel 1.1 sebagai berikut:

Tabel 1.1 Perbedaan penelitian

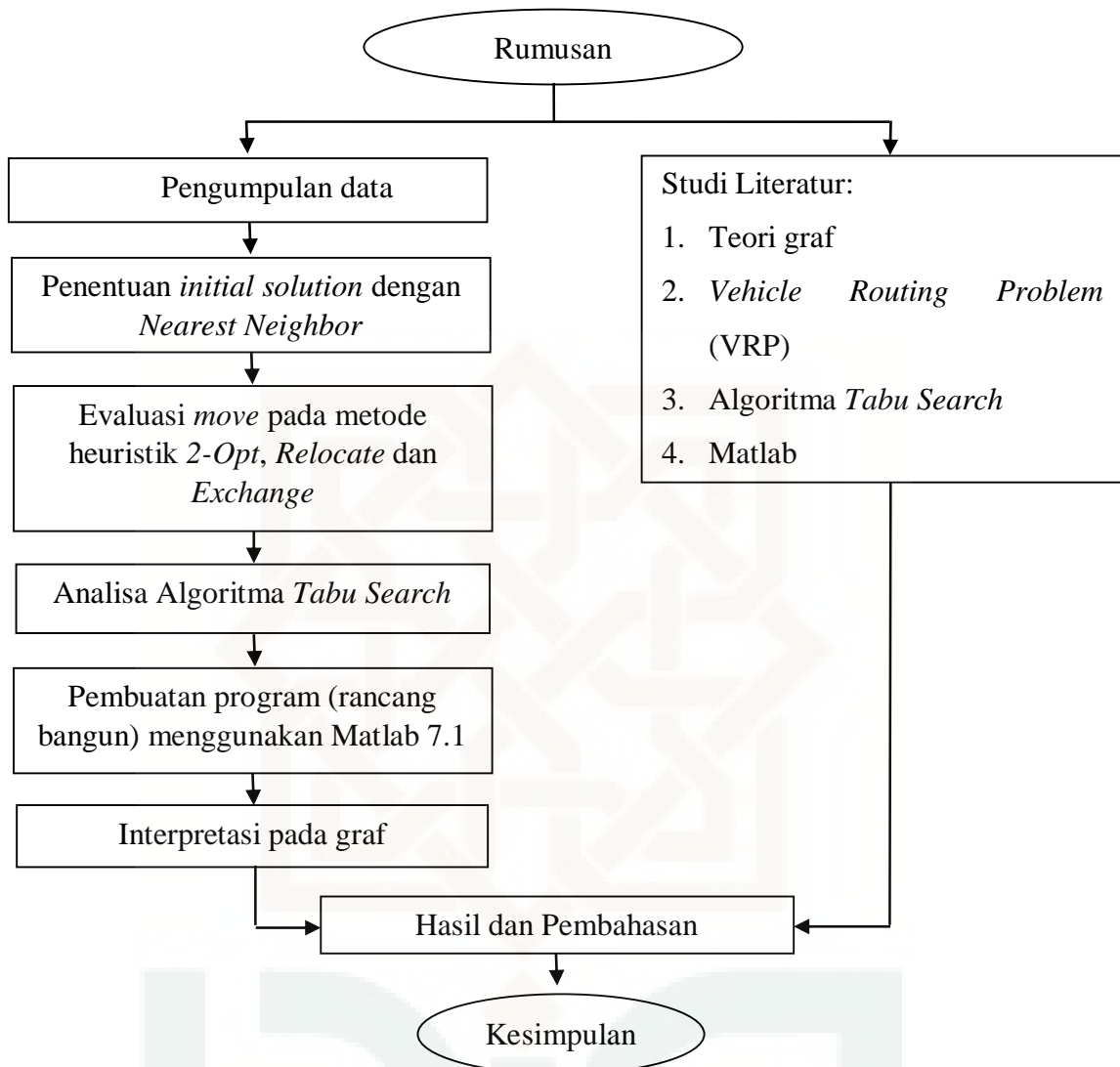
No	Nama	Judul	Perbedaan
1	Aji Raditya	Penggunaan Metode Heuristik dalam Permasalahan <i>Vehicle Routing Problem</i> dan Implementasinya di PT Nippon Indosari Corpindo.	Dalam menyelesaikan VRP Skripsi tersebut menggunakan metode <i>Nearest Neighbor</i> dengan perbaikan <i>2-Opt</i> , <i>or-Opt</i> , <i>Relocate</i> , <i>Exchange</i> , dan <i>Cross</i> (program menggunakan ILOG).
2	Vylda Pavela dan Imam Nurhadi	Penyelesaian <i>Vehicle Routing Problem</i> dengan Menggunakan Algoritma <i>Nearest Neighbor</i> dan <i>Tabu Search</i> (Studi Kasus di PT Nippon Indosari Corpindo).	Penelitian tersebut menyelesaikan VRP menggunakan Algoritma <i>Tabu Search</i> dengan pertukaran posisi antar konsumen yang terdekat (program menggunakan Delphi).
3	Berlian Trifal Mahendra dan Sapti Wahyuningsih	Analisis Kerja Algoritma <i>Tabu Search</i> pada <i>Vehicle Routing Problem with Backhaul</i> (VRPB) dengan Perbaikan <i>2-Opt</i> .	Algoritma <i>Tabu Search</i> yang digunakan pada penelitian tersebut menggunakan penukaran titik antar rute kemudian diperbaiki menggunakan metode <i>2-Opt</i> ,
4	Sulistiono	Rancang Bangun <i>Vehicle Routing Problem</i> menggunakan Algoritma <i>Tabu Search</i>	Skripsi ini menggunakan Algoritma <i>Tabu Search</i> dengan <i>Nearest Neighbor</i> sebagai solusi awal yang diperbaiki dengan metode <i>2-Opt</i> , <i>Relocate</i> , dan <i>Exchange</i> (program menggunakan Matlab 8.1) sehingga lebih efektif dan efisien dalam menemukan solusi VRP.

Skripsi dengan judul “Rancang Bangun *Vehicle Routing Problem* menggunakan Algoritma *Tabu Search*” ini, menggunakan ketiga penelitian tersebut sebagai acuan. Penelitian ini akan menyelesaikan VRP menggunakan Algoritma *Tabu Search* dengan metode perbaikan rute *2-Opt*, *Relocate* dan *Exchange* kemudian dibuat program (rancang bangun) menggunakan Matlab dengan harapan dapat menghasilkan solusi yang optimal dan mempermudah penggunaannya ataupun pengembangannya.

G. Metode Penelitian

Penelitian ini termasuk penelitian deskriptif kualitatif, yaitu suatu penelitian untuk menggambarkan fenomena-fenomena yang ada saat ini atau lampau yang lebih menekankan pada aspek pemahaman secara mendalam terhadap suatu masalah. Proses penelitian yang dilakukan tidak mengadakan manipulasi atau perubahan pada variabel-variabel bebas, tetapi menggambarkan suatu kondisi apa adanya. Hasil penelitian ini tidak membuat generalisasi atau tidak dapat diterapkan dalam kondisi dan tempat yang berbeda dengan tempat penelitian. Tujuan dari penelitian deskriptif kualitatif searah dengan rumusan masalah.

Objek penelitian adalah *Vehicle Routing Problem* dengan menggunakan Algoritma *Tabu Search*. Data yang digunakan untuk penelitian ini adalah data yang terdiri dari jarak antara depot ke konsumen dan antar konsumen, letak konsumen pada koordinat, dan permintaan konsumen. Program Matlab digunakan untuk memudahkan pencarian solusi optimal dari permasalahan tersebut. Buku, skripsi, jurnal penelitian dan referensi lainnya digunakan sebagai referensi teori yang digunakan untuk menyelesaikan kasus atau permasalahan sehingga penelitian ini merupakan studi literatur (Perhatikan Gambar 1.1).



Gambar 1.1 Skema langkah penelitian

H. Sistematika Penulisan

Penulisan skripsi ini dibagi menjadi empat bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas mengenai Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat Penelitian, Tinjauan Pustaka, Metode Penelitian dan Sistematika Penulisan.

BAB II LANDASAN TEORI

Bab ini berisi teori-teori tentang graf, graf berbobot (*weighted graph*), graf berarah berbobot (*weighted directed graph*), graf Hamilton, *Travelling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP), *Capacitated Vehicle Routing Problem* (CVRP), Algoritma *Tabu Search*, penyelesaian VRP menggunakan Algoritma *Tabu Search* dan Matlab yang menjadi dasar pembahasan.

BAB III PEMBAHASAN

Bab ini membahas mengenai konsep, langkah, dan penerapan Algoritma *Tabu Search* dalam menyelesaikan VRP serta pembuatan rancang bangun menggunakan program Matlab.

BAB IV PENUTUP

Bab ini berisi kesimpulan dan saran dari pokok bab–bab sebelumnya.

BAB IV

PENUTUP

A. Kesimpulan

Berdasarkan hasil pembahasan tentang penerapan Algoritma *Tabu Search* pada *Vehicle Routing Problem* dapat ditarik kesimpulan sebagai berikut:

1. Proses perhitungan Algoritma *Tabu Search* terdiri dari lima langkah. Langkah pertama yaitu menentukan solusi awal sebagai iterasi 0 dan menetapkan nilai solusi awal sebagai nilai solusi optimum sementara. Langkah kedua yaitu mencari solusi *Neighborhood* (solusi alternatif) yang tidak melanggar tabu atau memenuhi kriteria aspirasi. Langkah ketiga yaitu memilih solusi terbaik diantara solusi *Neighborhood* pada tiap iterasi yang akan disimpan sebagai solusi optimum. Langkah keempat yaitu memperbarui *Tabu List* dengan memasukkan node yang telah digunakan pada pertukaran node di langkah ketiga. Langkah terakhir yaitu apabila kriteria pemberhentian dipenuhi maka proses perhitungan Algoritma *Tabu Search* berhenti dan diperoleh solusi optimum, jika tidak dipenuhi maka proses kembali berulang dimulai pada langkah kedua.
2. Program (rancang bangun) dibuat menggunakan MATLAB yang dimulai dengan membuat *source code* utama menggunakan *m.file* sebagai kode untuk menjalankan program dan kemudian desain tampilan untuk program dirancang menggunakan *fig-file* sehingga diperoleh program dalam bentuk GUI (*Graphical User Interface*) pada MATLAB.

3. Pada kasus PT Sinergi Bio Natural diperoleh solusi optimum dengan menggunakan Algoritma *Tabu Search*. Jarak terpendek yang didapat menggunakan perhitungan manual dan rancang bangun adalah 101,1 km. Berdasarkan hasil perhitungan dihasilkan dua solusi rute perjalanan optimum alternatif menggunakan perhitungan manual dan rancang bangun sebagai berikut (lihat Tabel 4.1):

Tabel 4.1 Solusi optimum perhitungan manual dan rancang bangun

Rute	Manual		Rancang Bangun	
	Solusi Optimum	Permintaan	Solusi Optimum	Permintaan
1	Depot - Chacha Milk Tea 3 - Perum Sido Mulyo - Chacha Milk Tea 1 - Toko Roti & Katering Asli - Happy Land Medical Centre - Chacha Milk Tea 2 - Perum Banteng 3 - Perum Banteng 2 - Depot	240 liter	Depot - RSU. Holistika Medika - RS. Panti Rini - Perum Cepoko Indah - RSU. Rajawali Citra - RSU. PKU Muhammadiyah Bantul - Hotel Agung Inn - Perum banteng 2 - Depot	260 liter
2	Depot - JIH – Depot	300 liter	Depot - JIH - Depot	300 liter
3	Depot - RSU. Holistika Medika - RS.Panti Rini - Perum Cepoko Indah - RSU. Rajawali Citra - RSU. PKU Muhammadiyah Bantul - Hotel Agung Inn - Depot	240 liter	Depot-Perum Banteng 3 - Chacha Milk Tea 2 - Happy Land Medical Center -Toko Roti & Ketering Asli - Chacha Milk Tea 1 - Perum Sido Mulyo - Chacha Milk Tea 3 - Depot	220 liter

B. Saran

Berdasarkan penelitian yang telah dilakukan, maka terdapat beberapa saran sebagai berikut:

1. Kemampuan program yang telah dibuat masih terbatas pada peta yang statis. Input koordinat masih sulit dipahami oleh user sehingga peneliti selanjutnya dapat mengembangkan program dengan peta yang dinamis dan mudah dipahami oleh user.

2. Bagi Peneliti selanjutnya dapat membandingkan Algoritma *Tabu Search* dengan Algoritma-Algoritma lain seperti Algoritma *Artificial Bee Colony* dan Algoritma *Shuffled Frog-Leaping* untuk menyelesaikan *Vehicle Routing Problem*. Sehingga dapat diperoleh kelebihan dan kekurangan Algoritma *Tabu Search* dibandingkan Algoritma-Algoritma tersebut.



Daftar Pustaka

- Alkallak, I. S & Sha'ban, R. Z. (2008). *Tabu Search Method For Solving The Traveling Salesman Problem. Raf. J. of Comp. & Math's*. 5:141-153
- Balakrishnan, R. & Ranganathan, K. (2012). *A Textbook of Graph Theory Second Edition*. New York: Springer.
- Caric, Tonci & Gold, Hrvoje. (2008). *Vehicle routing problem*. University of Zagreb: In-teh Croatia.
- Çetiner, Selim. (2003). *An Iterative Hub Location and Routing Problem for Postal Delivery Sysems*. Turki: The Middle East Technical University.
- Christopher, Martin. (2011). *Logistics & Supply Chain Management Fourth Edition*. United States of America: Prentice Hall, Inc.
- Cordeau, J.F. , Laporte, G. , Savelsbergh, M.W. , et al. (2007). *Vehicle Routing. Handbook on OR & MS*. 14: 367-370.
- Gendreau, M. (2002). *An Introduction to Tabu Search*. University of Montreal: Montreal
- Glover, F & Kochenberger, G.A. (Eds). (2003). *Handbook of Metaheuristics*. Dordrecht: Kluwer Academic Publisher.
- Glover, F & Laguna, M. (1997). *Tabu Search*. Massachusetts: Kluwer Academic Publisher.
- Glover, F & Marti, R. (2006). *Metaheuristic Produres for Training Neural Networks*. Alba and Marti (Eds.), Springer: 53-70.
- Gooddairrie, Edgar G. & Parmenter, Michael M. (2002). *Discrete Mathematics with Graph Theory Second Edition*. United States of America: Prentice-Hall, Inc.
- Kallehauge, B. , Larsen J. , & Marsen OBG. (2001). *Lagrangean Duality Applied on Vehicle Routing Problem with Time Windows. Technical Report*. IMM. Technical University of Denmark. DK-2800 Kgs. Lyngby – Denmark Knight, Andrew. (2000). *Basic of MATLAB and Beyond*. Boca Raton: CRC Press LLC.
- Kusumadewi, S & Purnomo, H. (2005). *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Graha Ilmu.

- Mahendra, Berlian T & Wahyuningsih, Sapti. (2013). *Analisis Kerja Algoritma Tabu Search pada Vehicle Routing Problem with Backhaul (VRPB) dengan Perbaikan 2-Opt*. Malang: FMIPA Universitas Negeri Malang.
- Mailto US. (10 Juni 2011). *TSP for VRP Solution with Capacity Constraint Using Tabu Search Algorithm*. Diambil pada tanggal 28 Juli 2015, dari <http://www.en.pudn.com>
- Mutakhirroh, Ling. , Saptono, Fajar. , Hasanah, Nur. , Wiryadinata, Romi. (2007). *Pemanfaatan Metode Heuristik Dalam Pencarian Jalur Terpendek Dengan Algoritma Semut Dan Genetika*, Yogyakarta, Seminar Nasional Aplikasi Teknologi Informasi.
- Nugroho, Dwi Satio. (2015). *Penerapan Algoritma Reverse Delete dalam Menentukan Minimum Spanning Tree Obyek Wisata Di Kota Yogyakarta*. Skripsi, tidak diterbitkan, Universitas Islam Negeri Sunan Kalijaga: Yogyakarta.
- Nurhayanti, S. (2013). *Perbandingan Metode Branch and Bound dengan Metode Clarke and Wright Savings untuk Penyelesaian Masalah Distribusi Aqua Galon di PT.Tirta Investama Yogyakarta*. Skripsi, tidak diterbitkan, Universitas Negeri Yogyakarta: Yogyakarta.
- Pavela, Vylida & Nurhadi, Imam. (2012). *Penyelesaian Vehicle Routing Problem dengan Menggunakan Algoritma Nearest Neighbor dan Tabu Search (Studi Kasus di PT Nippon Indosari Corpindo)*. Matematika.1: 1-9.
- Pradhana, Fajar Eska. (2011). *Penerapan Algoritma Tabu Search untuk Menyelesaikan Vehicle Routing Problem*. Skripsi, tidak diterbitkan, Universitas Negeri Semarang: Semarang.
- Raditya, Aji. (2009). *Penggunaan Metode Heuristik dalam Permasalahan Vehicle Routing Problem dan Implementasinya di PT Nippon Indosari Corpindo*. Skripsi, tidak diterbitkan, Institut Pertanian Bogor: Jawa Barat.
- Rahmat, Basuki. 2011. *Perbandingan Genetic Algorithm, Multiple Ant Colony System, dan Tabu Search untuk Penyelesaian Vehicle Routing Problem With Time Windows (VRPTW)*. Jawa Timur.
- Rosen, Kenneth H. 2012. *Discrete Mathematics and Its Application Seventh Edition*. New York: Mc-Graw-Hill.
- Sugiharto, Aris. 2006. *Pemrograman GUI dengan MATLAB*. Yogyakarta: Penerbit Andi.

- Suyanto. 2010. *Algoritma Optimasi Deterministik atau Probabilitik*. Yogyakarta: Graha Ilmu.
- Solomon, M & Desrosiers, J. 1988. *Time window constrained routing and scheduling Problems. Transportation science, vol. 22*
- Taha, H. A. 2003. *Operations Research: An Introduction seventh Edition*. Prentice Hall, Inc.
- Toth, P & Vigo, D. 2002. *The Vehicle Routing Problem*. Philadelphia: Siam.
- Viktaria, Anie. 2015. Efektivitas Algoritma *Simulated Annealing* dan *Large Neighborhood Search* dalam Penyelesaian *Pickup and Delivery Vehicle Routing Problem with Time Windows*. Skripsi, tidak diterbitkan, Universitas Negeri Yogyakarta: Yogyakarta.

LAMPIRAN 1

ITERASI 1

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	<i>Exchange</i>	9	2	1-13-14-10-12-8-16-9-2-4-15-6-11-3-7-5-1
2	2-Opt	10	5	1-13-14-5-7-3-11-6-15-4-9-2-16-8-12-10-1
3	<i>Exchange</i>	9	11	1-13-14-10-12-8-16-2-11-4-15-6-9-3-7-5-1
4	2-Opt	12	16	1-13-14-10-16-8-12-2-9-4-15-6-11-3-7-5-1
5	<i>Exchange</i>	3	5	1-13-14-10-12-8-16-2-9-4-15-6-11-5-7-3-1
6	<i>Relocated</i>	5	15	1-13-14-10-12-8-16-2-9-4-5-15-6-11-3-7-1
7	<i>Relocated</i>	2	14	1-13-2-14-10-12-8-16-9-4-15-6-11-3-7-5-1
8	2-Opt	9	6	1-13-14-10-12-8-16-2-6-15-4-9-11-3-7-5-1
9	2-Opt	10	9	1-13-14-9-2-16-8-12-10-4-15-6-11-3-7-5-1
10	2-Opt	13	5	1-5-7-3-11-6-15-4-9-2-16-8-12-10-14-13-1
11	2-Opt	13	10	1-10-14-13-12-8-16-2-9-4-15-6-11-3-7-5-1
12	<i>Exchange</i>	6	3	1-13-14-10-12-8-16-2-9-4-15-3-11-6-7-5-1
13	<i>Relocated</i>	13	4	1-14-10-12-8-16-2-9-4-13-15-6-11-3-7-5-1
14	<i>Relocated</i>	7	8	1-13-14-10-12-7-8-16-2-9-4-15-6-11-3-5-1
15	<i>Relocated</i>	9	12	1-13-14-10-9-12-8-16-2-4-15-6-11-3-7-5-1
16	<i>Relocated</i>	5	7	1-13-14-10-12-8-16-2-9-4-15-6-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1: 1-13-14-10-12-8-16-9-2-1 Rute 2: 1-4-1 Rute 3: 15-6-11-3-7-5-1	113 km
2	Rute 1: 1-13-14-5-7-3-11-6-15-1 Rute 2: 1-4-1 Rute 3: 1-9-2-16-8-12-10-1	107,2 km
3	Rute 1: 1-13-14-10-12-8-16-2-11-1 Rute 2: 1-4-1 Rute 3: 1-15-6-9-3-7-5-1	126,7 km
4	Rute 1: 1-13-14-10-16-8-12-2-9-1 Rute 2: 1-4-1 Rute 3: 1-15-6-11-3-7-5-1	114,3 km
5	Rute 1: 1-13-14-10-12-8-16-2-9-1 Rute 2: 1-4-1 Rute 3: 1-15-6-11-5-7-3-1	116 km
6	Rute 1: 1-13-14-10-12-8-16-2-9-1	127,5 km

	Rute 2: 1-4-1 Rute 3: 1-5-15-6-11-3-7-1	
7	Rute 1: 1-13-2-14-10-12-8-16-9-1 Rute 2: 1-4-1 Rute 3: 1-15-6-11-3-7-5-1	122,5 km
8	Rute 1 : 1-13-14-10-12-8-16-2-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-9-11-3-7-5-1	111,5 km
9	Rute 1 : 1-13-14-9-2-16-8-12-10-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-7-5-1	104,2 km
10	Rute 1 : 1-5-7-3-11-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-8-12-10-14-13-1	106,5 km
11	Rute 1 : 1-10-14-13-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-7-5-1	109,6 km
12	Rute 1 : 1-13-14-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-3-11-6-7-5-1	126 km
13	Rute 1 : 1-14-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-13-15-6-11-3-7-5-1	107 km
14	Rute 1 : 1-13-14-10-12-7-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-1	110,6 km
15	Rute 1 : 1-13-14-10-9-12-8-16-2-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-7-5-1	115 km
16	Rute 1 : 1-13-14-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	103,4 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-13-14-10-12-8-16-2-9-1	240 liter	31,3 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			103,4 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-13-14-10-12-8-16-2-9-1	240 liter	31,3 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			103,4 km

Solusi *Neighborhood* TSP terbaik iterasi 1: 1-13-14-10-12-8-16-2-9-4-15-6-11-3-5-7-1

ITERASI 2

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	0	0	0	0	0	0
	0	7	0	0	0	0	0	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	<i>Exchange</i>	12	3	1-13-14-10-3-8-16-2-9-4-15-6-11-12-5-7-1
2	2-Opt	16	7	1-13-14-10-12-8-7-5-3-11-6-15-4-9-2-16-1
3	<i>Relocated</i>	4	12	1-13-14-10-4-12-8-16-2-9-15-6-11-3-5-7-1
4	<i>Exchange</i>	8	6	1-13-14-10-12-6-16-2-9-4-15-8-11-3-5-7-1
5	<i>Relocated</i>	14	10	1-13-10-14-12-8-16-2-9-4-15-6-11-3-5-7-1
6	<i>Relocated</i>	10	8	1-13-14-12-8-10-16-2-9-4-15-6-11-3-5-7-1
7	<i>Relocated</i>	13	14	1-14-13-10-12-8-16-2-9-4-15-6-11-3-5-7-1
8	<i>Relocated</i>	15	8	1-13-14-10-12-15-8-16-2-9-4-6-11-3-5-7-1
9	<i>Exchange</i>	10	3	1-13-14-3-12-8-16-2-9-4-15-6-11-10-5-7-1
10	<i>Relocated</i>	9	5	1-13-14-10-12-8-16-2-4-15-6-11-3-5-9-7-1
11	2-Opt	16	7	1-13-14-10-12-8-7-5-3-11-6-15-4-9-2-16-1
12	<i>Exchange</i>	5	11	1-13-14-10-12-8-16-2-9-4-15-6-5-3-11-7-1
13	<i>Exchange</i>	9	16	1-13-14-10-12-8-9-2-16-4-15-6-11-3-5-7-1
14	<i>Relocated</i>	11	9	1-13-14-10-12-8-16-2-11-9-4-15-6-3-5-7-1
15	<i>Relocated</i>	9	5	1-13-14-10-12-8-16-2-4-15-6-11-3-5-9-7-1
16	<i>Exchange</i>	15	2	1-13-14-10-12-8-16-15-9-4-2-6-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-13-14-10-3-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-12-5-7-1	134,9 km
2	Rute 1 : 1-13-14-10-12-8-7-5-3-11-1 Rute 2 : 1-6-15-1 Rute 3 : 1-4-1 Rute 4 : 1-9-2-16-1	126,6 km
3	Rute 1 : 1-13-14-10-1 Rute 2 : 1-4-1 Rute 3 : 1-12-8-16-2-9-15-6-11-1 Rute 4 : 1-3-5-7-1	129,8 km
4	Rute 1 : 1-13-14-10-12-6-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-8-11-3-5-7-1	121,4 km
5	Rute 1 : 1-13-10-14-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	106,9 km

6	Rute 1 : 1-13-14-12-8-10-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	116,4 km
7	Rute 1 : 1-14-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	103,1 km
8	Rute 1 : 1-13-14-10-12-15-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-6-11-3-5-7-1	119,2 km
9	Rute 1 : 1-13-14-3-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-10-5-7-1	146,5 km
10	Rute 1 : 1-13-14-10-12-8-16-2-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-9-7-1	121,3 km
11	Rute 1 : 1-13-14-10-12-8-7-5-3-11-1 Rute 2 : 1-6-15-1 Rute 3 : 1-4-1 Rute 4 : 1-9-2-16-1	126,6 km
12	Rute 1 : 1-13-14-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-5-3-11-7-1	117,8 km
13	Rute 1 : 1-13-14-10-12-8-9-2-16-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	106,6 km
14	Rute 1 : 1-13-14-10-12-8-16-2-11-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-3-5-7-1	116,7 km
15	Rute 1 : 1-13-14-10-12-8-16-2-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-9-7-1	121,2 km
16	Rute 1 : 1-13-14-10-12-8-16-15-9-1 Rute 2 : 1-4-1 Rute 3 : 1-2-6-11-3-5-7-1	116,9 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-14-13-10-12-8-16-2-9-1	240 liter	31 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			103,1km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-14-13-10-12-8-16-2-9-1	240 liter	31 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			103,1km

Solusi *Neighborhood* TSP terbaik iterasi 2: 1-14-13-10-12-8-16-2-9-4-15-6-11-3-5-7-1

ITERASI 3

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	13	0	0	0	0	0
	0	7	14	0	0	0	0	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	<i>Relocated</i>	16	8	1-14-13-10-12-16-8-2-9-4-15-6-11-3-5-7-1
2	<i>Relocated</i>	10	5	1-14-13-12-8-16-2-9-4-15-6-11-3-5-10-7-1
3	2-Opt	10	7	1-14-13-7-5-3-11-6-15-4-9-2-16-8-12-10-1
4	2-Opt	2	6	1-14-13-10-12-8-16-6-15-4-9-2-11-3-5-7-1
5	<i>Exchange</i>	4	7	1-14-13-10-12-8-16-2-9-7-15-6-11-3-5-4-1
6	<i>Exchange</i>	3	11	1-14-13-10-12-8-16-2-9-4-15-6-3-11-5-7-1
7	<i>Relocated</i>	11	14	1-11-14-13-10-12-8-16-2-9-4-15-6-3-5-7-1
8	2-Opt	16	11	1-14-13-10-12-8-11-6-15-4-9-2-16-3-5-7-1
9	2-Opt	10	4	1-14-13-4-9-2-16-8-12-10-15-6-11-3-5-7-1
10	2-Opt	8	5	1-14-13-10-12-5-3-11-6-15-4-9-2-16-8-7-1
11	<i>Relocated</i>	13	7	1-14-10-12-8-16-2-9-4-15-6-11-3-5-7-13-1
12	<i>Exchange</i>	11	10	1-14-13-11-12-8-16-2-9-4-15-6-10-3-5-7-1
13	<i>Relocated</i>	14	4	1-13-10-12-8-16-2-9-4-14-15-6-11-3-5-7-1
14	2-Opt	16	9	1-14-13-10-12-8-9-2-16-4-15-6-11-3-5-7-1
15	<i>Relocated</i>	11	16	1-14-13-10-12-8-11-16-2-9-4-15-6-3-5-7-1
16	<i>Relocated</i>	16	8	1-14-13-10-12-16-8-2-9-4-15-6-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-14-13-10-12-16-8-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	106,3 km
2	Rute 1 : 1-14-13-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-10-7-1	121,4 km
3	Rute 1 : 1-14-13-7-5-3-11-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-8-12-10-1	102,9 km
4	Rute 1 : 1-14-13-10-12-8-16-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-11-3-5-7-1	111,1 km
5	Rute 1 : 1-14-13-10-12-8-16-2-9-7-15-1 Rute 2 : 1-6-11-3-5-1	124,1 km

	Rute 3 : 1-4-1	
6	Rute 1 : 1-14-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-3-11-5-7-1	108,3 km
7	Rute 1 : 1-11-14-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-3-5-7-1	133,8 km
8	Rute 1 : 1-14-13-10-12-8-11-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-3-5-7-1	117,4 km
9	Rute 1 : 1-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-8-12-10-15-6-1 Rute 4 : 1-11-3-5-7-1	124,5 km
10	Rute 1 : 1-14-13-10-12-5-3-11-1 Rute 2 : 1-6-15-1 Rute 3 : 1-4-1 Rute 4 : 1-9-2-16-8-7-1	129,6 km
11	Rute 1 : 1-14-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-13-1	103,4 km
12	Rute 1 : 1-14-13-11-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-10-3-5-7-1	145,1 km
13	Rute 1 : 1-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-3-5-7-1	102,3 km
14	Rute 1 : 1-14-13-10-12-8-9-2-16-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	106,4 km
15	Rute 1 : 1-14-13-10-12-8-11-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-3-5-7-1	119,8 km
16	Rute 1 : 1-14-13-10-12-16-8-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	106,3 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-13-10-12-8-16-2-9-1	215 liter	28,7 km
2	1-4-1	300 liter	9,2 km
3	1-14-15-6-11-3-5-7-1	265 liter	64,4 km
Total jarak tempuh kendaraan			102,3 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-13-10-12-8-16-2-9-1	215 liter	28,7 km
2	1-4-1	300 liter	9,2 km
3	1-14-15-6-11-3-5-7-1	265 liter	64,4 km
Total jarak tempuh kendaraan			102,3 km

Solusi *Neighborhood* TSP terbaik iterasi 3: 1-13-10-12-8-16-2-9-4-14-15-6-11-3-5-7-1

ITERASI 4

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	13	14	0	0	0	0
	0	7	14	4	0	0	0	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	2-Opt	8	6	1-13-10-12-6-15-14-4-9-2-16-8-11-3-5-7-1
2	2-Opt	16	11	1-13-10-12-8-11-6-15-14-4-9-2-16-3-5-7-1
3	2-Opt	10	8	1-13-8-12-10-16-2-9-4-14-15-6-11-3-5-7-1
4	2-Opt	3	5	1-13-10-12-8-16-2-9-4-14-15-6-11-5-3-7-1
5	<i>Exchange</i>	3	12	1-13-10-3-8-16-2-9-4-14-15-6-11-12-5-7-1
6	2-Opt	10	14	1-13-14-4-9-2-16-8-12-10-15-6-11-3-5-7-1
7	<i>Exchange</i>	2	7	1-13-10-12-8-16-7-9-4-14-15-6-11-3-5-2-1
8	2-Opt	8	11	1-13-10-12-11-6-15-14-4-9-2-16-8-3-5-7-1
9	2-Opt	10	14	1-13-14-4-9-2-16-8-12-10-15-6-11-3-5-7-1
10	<i>Exchange</i>	8	4	1-13-10-12-4-16-2-9-8-14-15-6-11-3-5-7-1
11	<i>Exchange</i>	15	16	1-13-10-12-8-15-2-9-4-14-16-6-11-3-5-7-1
12	<i>Exchange</i>	4	14	1-13-10-12-8-16-2-9-14-4-15-6-11-3-5-7-1
13	<i>Exchange</i>	5	6	1-13-10-12-8-16-2-9-4-14-15-5-11-3-6-7-1
14	Relocated	7	13	1-7-13-10-12-8-16-2-9-4-14-15-6-11-3-5-1
15	Relocated	3	11	1-13-10-12-8-16-2-9-4-14-15-6-3-11-5-7-1
16	Relocated	3	5	1-13-10-12-8-16-2-9-4-14-15-6-11-5-3-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-13-10-12-6-15-14-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-8-11-3-5-7-1	111,5 km
2	Rute 1 : 1-13-10-12-8-11-6-15-14-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-3-5-7-1	116,6 km
3	Rute 1 : 1-13-8-12-10-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-3-5-7-1	113,5 km
4	Rute 1 : 1-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-5-3-7-1	111,6 km
5	Rute 1 : 1-13-10-3-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-12-5-7-1	133,8 km
6	Rute 1 : 1-13-14-1 Rute 2 : 1-4-1	124,5 km

	Rute 3 : 1-9-2-16-8-12-10-15-6-1 Rute 4 : 1-11-3-5-7-1	
7	Rute 1 : 1-13-10-12-8-16-7-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-3-5-1 Rute 4 : 1-2-1	126,6 km
8	Rute 1 : 1-13-10-12-11-6-15-14-1 Rute 2 : 1-4-1 Rute 3 : 1-9-2-16-8-3-5-7-1	120,2 km
9	Rute 1 : 1-13-14-1 Rute 2 : 1-4-1- Rute 3 : 1-9-2-16-8-12-10-15-6-1 Rute 4: 1-11-3-5-7-1	124,5 km
10	Rute 1 : 1-13-10-12-1 Rute 2 : 1-4-1 Rute 3 : 1-16-2-9-8-14-15-6-11-1 Rute 4 : 1-3-5-7-1	145 km
11	Rute 1 : 1-13-10-12-8-15-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-16-6-11-3-5-7-1	119,5 km
12	Rute 1 : 1-13-10-12-8-16-2-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	101,3 km
13	Rute 1 : 1-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-5-11-3-6-7-1	134,5 km
14	Rute 1 : 1-7-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-3-5-1	121,5 km
15	Rute 1 : 1-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-3-11-5-7-1	107,5 km
16	Rute 1 : 1-13-10-12-8-16-2-9-1 Rute 2 : 1-4-1 Rute 3 : 1-14-15-6-11-5-3-7-1	111,6 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-13-10-12-8-16-2-9-14-1	240 liter	29,2 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,3 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-13-10-12-8-16-2-9-14-1	240 liter	29,2 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,3 km

Solusi *Neighborhood* TSP terbaik iterasi 4: 1-13-10-12-8-16-2-9-14-4-15-6-11-3-5-7-1

ITERASI 5

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	13	14	4	0	0	0
	0	7	14	4	14	0	0	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	2-Opt	10	16	1-13-16-8-12-10-2-9-14-4-15-6-11-3-5-7-1
2	Relocated	6	11	1-13-10-12-8-16-2-9-14-4-15-11-6-3-5-7-1
3	2-Opt	9	3	1-13-10-12-8-16-2-3-11-6-15-4-14-9-5-7-1
4	Relocated	15	16	1-13-10-12-8-15-16-2-9-14-4-6-11-3-5-7-1
5	2-Opt	9	15	1-13-10-12-8-16-2-15-4-14-9-6-11-3-5-7-1
6	Relocated	14	8	1-13-10-12-14-8-16-2-9-4-15-6-11-3-5-7-1
7	Exchange	12	2	1-13-10-2-8-16-12-9-14-4-15-6-11-3-5-7-1
8	Relocated	4	15	1-13-10-12-8-16-2-9-14-15-4-6-11-3-5-7-1
9	Relocated	13	14	1-10-12-8-16-2-9-14-13-4-15-6-11-3-5-7-1
10	Relocated	5	9	1-13-10-12-8-16-2-5-9-14-4-15-6-11-3-7-1
11	Exchange	12	5	1-13-10-5-8-16-2-9-14-4-15-6-11-3-12-7-1
12	2-Opt	16	6	1-13-10-12-8-6-15-4-14-9-2-16-11-3-5-7-1
13	Relocated	2	5	1-13-10-12-8-16-9-14-4-15-6-11-3-5-2-7-1
14	Relocated	12	7	1-13-10-8-16-2-9-14-4-15-6-11-3-5-7-12-1
15	Exchange	11	10	1-13-11-12-8-16-2-9-14-4-15-6-10-3-5-7-1
16	Relocated	4	16	1-13-10-12-8-4-16-2-9-14-15-6-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-13-16-8-12-10-2-9-14-1 Rute 2 : 1- 4-1 Rute 3 : 1-15-6-11-3-5-7-1	114,7 km
2	Rute 1 : 1-13-10-12-8-16-2-9-14-1 Rute 2 : 1- 4-1 Rute 3 : 1-15-11-6-3-5-7-1	118,1 km
3	Rute 1 : 1-13-10-12-8-16-2-3-1 Rute 2 : 1-11-6-15-1 Rute 3 : 1-4-1 Rute 4 : 1-14-9-5-7-1	145 km
4	Rute 1 : 1-13-10-12-8-15-16-2-9-14-1 Rute 2 : 1- 4-1 Rute 3 : 1-6-11-3-5-7-1	117,5 km
5	Rute 1 : 1-13-10-12-8-16-2-15-1 Rute 2 : 1- 4-1 Rute 3 : 1-14-9-6-11-3-5-7-1	106 ,1km
6	Rute 1 : 1-13-10-12-14-8-16-2-9-1 Rute 2 : 1- 4-1 Rute 3 : 1-15-6-11-3-5-7-1-1	111 km

7	Rute 1 : 1-13-10-2-8-16-12-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	112,4 km
8	Rute 1 : 1-13-10-12-8-16-2-9-14-15-1 Rute 2 : 1-4-1 Rute 3 : 1-6-11-3-5-7-1	112,4 km
9	Rute 1 : 1-10-12-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	101,1 km
10	Rute 1 : 1-13-10-12-8-16-2-5-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-7-1	118,4 km
11	Rute 1 : 1-13-10-5-8-16-2-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-12-7-1	124 km
12	Rute 1 : 1-13-10-12-8-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-14-9-2-16-11-3-5-1 Rute 4 : 1-7-1	128,5 km
13	Rute 1 : 1-13-10-12-8-16-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-2-7-1	112,5 km
14	Rute 1 : 1-13-10-8-16-2-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-12-1	103,5 km
15	Rute 1 : 1-13-11-12-8-16-2-9-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-10-3-5-7-1	143,3 km
16	Rute 1 : 1-13-10-12-8-1 Rute 2 : 1-4-1 Rute 3 : 1-16-2-9-14-15-6-11-1 Rute 4 : 1-3-5-7-1	142,8 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

Solusi *Neighborhood* TSP terbaik iterasi 5 : 1-10-12-8-16-2-9-14-13-4-15-6-11-3-5-7-1

ITERASI 6

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	13	14	4	13	0	0
	0	7	14	4	14	14	0	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	Relocated	4	10	1-4-10-12-8-16-2-9-14-13-15-6-11-3-5-7-1
2	Exchange	13	5	1-10-12-8-16-2-9-14-5-4-15-6-11-3-13-7-1
3	2-Opt	9	7	1-10-12-8-16-2-7-5-3-11-6-15-4-13-14-9-1
4	Relocated	12	5	1-10-8-16-2-9-14-13-4-15-6-11-3-5-12-7-1
5	Relocated	11	6	1-10-12-8-16-2-9-14-13-4-15-11-6-3-5-7-1
6	Relocated	7	16	1-10-12-8-7-16-2-9-14-13-4-15-6-11-3-5-1
7	2-Opt	15	5	1-10-12-8-16-2-9-14-13-4-5-3-11-6-15-7-1
8	2-Opt	2	13	1-10-12-8-16-13-14-9-2-4-15-6-11-3-5-7-1
9	Exchange	2	5	1-10-12-8-16-5-9-14-13-4-15-6-11-3-2-7-1
10	Relocated	3	8	1-10-12-3-8-16-2-9-14-13-4-15-6-11-5-7-1
11	2-Opt	10	9	1-9-2-16-8-12-10-14-13-4-15-6-11-3-5-7-1
12	Exchange	3	5	1-10-12-8-16-2-9-14-13-4-15-6-11-5-3-7-1
13	Relocated	4	2	1-10-12-8-16-4-2-9-14-13-15-6-11-3-5-7-1
14	Exchange	11	6	1-10-12-8-16-2-9-14-13-4-15-11-6-3-5-7-1
15	Relocated	5	6	1-10-12-8-16-2-9-14-13-4-15-5-6-11-3-7-1
16	Exchange	8	14	1-10-12-14-16-2-9-8-13-4-15-6-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-4-1 Rute 2 : 1-10-12-8-16-2-9-14-13-15-1 Rute 3 : 1-6-11-3-5-7-1	113,5 km
2	Rute 1 : 1-10-12-8-16-2-9-14-5-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-13-7-1	144,7 km
3	Rute 1 : 1-10-12-8-16-2-7-5-3-1 Rute 2 : 1-11-6-15-1 Rute 3 : 1-4-1 Rute 4: 1-13-14-9-1	128,7 km
4	Rute 1 : 1-10-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-12-7-1	109,4 km
5	Rute 1 : 1-10-12-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-11-6-3-5-7-1	117,9 km

6	Rute 1 : 1-10-12-8-7-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-1	108 km
7	Rute 1 : 1-10-12-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-5-3-11-6-15-7-1	118,2 km
8	Rute 1 : 1-10-12-8-16-13-14-9-2-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	116,4 km
9	Rute 1 : 1-10-12-8-16-5-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-2-7-1	122,6 km
10	Rute 1 : 1-10-12-3-8-16-2-9-14-1 Rute 2 : 1-13-1 Rute 3 : 1-4-1 Rute 4 : 1-15-6-11-5-7-1	121,5 km
11	Rute 1 : 1-9-2-16-8-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	103,4 km
12	Rute 1 : 1-10-12-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-5-3-7-1-1	110,4 km
13	Rute 1 : 1-10-12-8-16-1 Rute 2 : 1-4-1 Rute 3 : 1-2-9-14-13-15-6-11-3-1 Rute 4 : 1-5-7-1	140,3 km
14	Rute 1 : 1-10-12-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-11-6-3-5-7-1	117,9 km
15	Rute 1 : 1-10-12-8-16-2-9-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-5-6-11-3-7-1	128,8 km
16	Rute 1 : 1-10-12-14-16-2-9-8-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	113,3 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-9-2-16-8-12-10-14-13-1	240 liter	31,3 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			103,4 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

Solusi *Neighborhood* TSP terbaik iterasi 6: 1-9-2-16-8-12-10-14-13-4-15-6-11-3-5-7-1

ITERASI 7

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	13	14	4	13	10	0
	0	7	14	4	14	14	9	0

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	2-Opt	8	4	1-9-2-16-4-13-14-10-12-8-15-6-11-3-5-7-1
2	2-Opt	14	3	1-9-2-16-8-12-10-3-11-6-15-4-13-14-5-7-1
3	<i>Exchange</i>	5	3	1-9-2-16-8-12-10-14-13-4-15-6-11-5-3-7-1
4	2-Opt	6	7	1-9-2-16-8-12-10-14-13-4-15-7-5-3-11-6-1
5	<i>Exchange</i>	14	8	1-9-2-16-14-12-10-8-13-4-15-6-11-3-5-7-1
6	<i>Exchange</i>	6	13	1-9-2-16-8-12-10-14-6-4-15-13-11-3-5-7-1
7	2-Opt	13	11	1-9-2-16-8-12-10-14-11-6-15-4-13-3-5-7-1
8	<i>Exchange</i>	5	14	1-9-2-16-8-12-10-5-13-4-15-6-11-3-14-7-1
9	Relocated	16	10	1-9-2-8-12-10-16-14-13-4-15-6-11-3-5-7-1
10	Relocated	3	5	1-9-2-16-8-12-10-14-13-4-15-6-11-5-3-7-1
11	Relocated	8	2	1-9-8-2-16-12-10-14-13-4-15-6-11-3-5-7-1
12	<i>Exchange</i>	14	2	1-9-14-16-8-12-10-2-13-4-15-6-11-3-5-7-1
13	2-Opt	16	15	1-9-2-15-4-13-14-10-12-8-16-6-11-3-5-7-1
14	2-Opt	10	3	1-9-2-16-8-12-3-11-6-15-4-13-14-10-5-7-1
15	Relocated	2	3	1-9-16-8-12-10-14-13-4-15-6-11-3-2-5-7-1
16	<i>Exchange</i>	5	16	1-9-2-5-8-12-10-14-13-4-15-6-11-3-16-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-9-2-16-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-12-8-15-6-11-1 Rute 4 : 1-3-5-7-1	141,9 km
2	Rute 1 : 1-9-2-16-8-12-10-3-1 Rute 2 : 1-11-6-15-1 Rute 3 : 1-4-1 Rute 4 : 1-13-14-5-7-1	156,4 km
3	Rute 1 : 1-9-2-16-8-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-5-3-7-1	112,7 km
4	Rute 1 : 1-9-2-16-8-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-7-5-3-11-6-1	113,2 km
5	Rute 1 : 1-9-2-16-14-12-10-8-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	121,8 km

6	Rute 1 : 1-9-2-16-8-12-10-14-6-1 Rute 2 : 1-4-1 Rute 3 : 1-15-13-11-3-5-7-1	134,1 km
7	Rute 1 : 1-9-2-16-8-12-10-14-11-6-1 Rute 2 : 1-15-1 Rute 3 : 1-4-1 Rute 4 : 1-13-3-5-7-1	144,3 km
8	Rute 1 : 1-9-2-16-8-12-10-5-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-14-7-1	141,4 km
9	Rute 1 : 1-9-2-8-12-10-16-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	115,9 km
10	Rute 1 : 1-9-2-16-8-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-5-3-7-1	112,7 km
11	Rute 1 : 1-9-8-2-16-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	105,6 km
12	Rute 1 : 1-9-14-16-8-12-10-2-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	124,1 km
13	Rute 1 : 1-9-2-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-12-8-16-6-11-1 Rute 4 : 1-3-5-7-1	143,5 km
14	Rute 1 : 1-9-2-16-8-12-3-11-1 Rute 2 : 1-6-15-1 Rute 3 : 1-4-1 Rute 4 : 1-13-14-10-5-7-1	138,3 km
15	Rute 1 : 1-9-16-8-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-2-5-7-1	116,2 km
16	Rute 1 : 1-9-2-5-8-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-16-7-1	122,8 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-9-8-2-16-12-10-14-13-1	240 liter	33,5 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			105,6 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

Solusi *Neighborhood* TSP terbaik iterasi 7 : 1-9-8-2-16-12-10-14-13-4-15-6-11-3-5-7-1

ITERASI 8

Tabu List								
	1	2	3	4	5	6	7	8
Move	0	5	13	14	4	13	10	8
	0	7	14	4	14	14	9	2

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	2-Opt	8	11	1-9-11-6-15-4-13-14-10-12-16-2-8-3-5-7-1
2	2-Opt	16	7	1-9-8-2-7-5-3-11-6-15-4-13-14-10-12-16-1
3	<i>Relocated</i>	9	4	1-8-2-16-12-10-14-13-4-9-15-6-11-3-5-7-1
4	<i>Relocated</i>	4	14	1-9-8-2-16-12-10-4-14-13-15-6-11-3-5-7-1
5	2-Opt	9	6	1-6-15-4-13-14-10-12-16-2-8-9-11-3-5-7-1
6	<i>Relocated</i>	7	9	1-7-9-8-2-16-12-10-14-13-4-15-6-11-3-5-1
7	Exchange	8	12	1-9-12-2-16-8-10-14-13-4-15-6-11-3-5-7-1
8	2-Opt	16	4	1-9-8-2-4-13-14-10-12-16-15-6-11-3-5-7-1
9	<i>Relocated</i>	12	15	1-9-8-2-16-10-14-13-4-15-12-6-11-3-5-7-1
10	<i>Relocated</i>	6	8	1-9-6-8-2-16-12-10-14-13-4-15-11-3-5-7-1
11	Exchange	8	13	1-9-13-2-16-12-10-14-8-4-15-6-11-3-5-7-1
12	<i>Relocated</i>	2	14	1-9-8-16-12-10-14-2-13-4-15-6-11-3-5-7-1
13	Exchange	12	11	1-9-8-2-16-11-10-14-13-4-15-6-12-3-5-7-1
14	Exchange	9	14	1-14-8-2-16-12-10-9-13-4-15-6-11-3-5-7-1
15	<i>Relocated</i>	10	9	1-10-9-8-2-16-12-14-13-4-15-6-11-3-5-7-1
16	2-Opt	12	14	1-9-8-2-16-14-10-12-13-4-15-6-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1: 1-9-11-6-15-1 Rute 2: 1-4-1 Rute 3: 1-13-14-10-12-16-2-8-3-1 Rute 4: 1-5-7-1	142,8 km
2	Rute 1: 1-9-8-2-7-5-3-11-6-1 Rute 2: 1-15-1 Rute 3: 1-4-1 Rute 4: 1-13-14-10-12-16-1	119,7 km
3	Rute 1: 1-8-2-16-12-10-14-13-1 Rute 2: 1-4-1 Rute 3: 1-9-15-6-11-3-5-7-1	106 km
4	Rute 1: 1-9-8-2-16-12-10-1 Rute 2: 1-4-1 Rute 3: 1-14-13-15-6-11-3-5-7-1	105,6 km

5	Rute 1 : 1-6-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-12-16-2-8-9-11-1 Rute 4 : 1-3-5-7-1	142,4 km
6	Rute 1 : 1-7-9-8-2-16-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-1	116,8 km
7	Rute 1 : 1-9-12-2-16-8-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	111,1 km
8	Rute 1 : 1-9-8-2-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-12-16-15-6-11-1 Rute 4: 1-3-5-7-1	147,5 km
9	Rute 1 : 1-9-8-2-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-5-7-1	120,8 km
10	Rute 1 : 1-9-6-8-2-16-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-11-3-5-7-1	118,2 km
11	Rute 1 : 1-9-13-2-16-12-10-14-8-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	124,5 km
12	Rute 1 : 1-9-8-16-12-10-14-2-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	118,7 km
13	Rute 1 : 1-9-8-2-16-11-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-12-3-5-7-1	136,4 km
14	Rute 1 : 1-14-8-2-16-12-10-9-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	112,1 km
15	Rute 1 : 1-10-9-8-2-16-12-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	107,3 km
16	Rute 1 : 1-9-8-2-16-14-10-12-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	112,7 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-9-8-2-16-10-14-13-1	200 liter	30 km
2	1-4-1	300 liter	9,2 km
3	1-15-12-6-11-3-5-7-1	280 liter	81,6 km
Total jarak tempuh kendaraan			120,8 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

Solusi *Neighborhood* TSP terbaik iterasi 8 : 1-9-8-2-16-10-14-13-4-15-12-6-11-3-5-7-1

ITERASI 9

Tabu List								
	1	2	3	4	5	6	7	8
Move	12	5	13	14	4	13	10	8
	15	7	14	4	14	14	9	2

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	Relocated	13	7	1-9-8-2-16-10-14-4-15-12-6-11-3-5-7-13-1
2	Relocated	10	9	1-10-9-8-2-16-14-13-4-15-12-6-11-3-5-7-1
3	Relocated	8	6	1-9-2-16-10-14-13-4-15-12-6-8-11-3-5-7-1
4	Relocated	4	11	1-9-8-2-16-10-14-13-15-12-6-11-4-3-5-7-1
5	Relocated	12	13	1-9-8-2-16-10-14-12-13-4-15-6-11-3-5-7-1
6	Exchange	12	7	1-9-8-2-16-10-14-13-4-15-7-6-11-3-5-12-1
7	Relocated	2	5	1-9-8-16-10-14-13-4-15-12-6-11-3-5-2-7-1
8	Relocated	8	15	1-9-2-16-10-14-13-4-15-8-12-6-11-3-5-7-1
9	Relocated	8	14	1-9-2-16-10-14-8-13-4-15-12-6-11-3-5-7-1
10	Exchange	16	14	1-9-8-2-14-10-16-13-4-15-12-6-11-3-5-7-1
11	2-Opt	14	3	1-9-8-2-16-10-3-11-6-12-15-4-13-14-5-7-1
12	2-Opt	16	6	1-9-8-2-6-12-15-4-13-14-10-16-11-3-5-7-1
13	Exchange	5	14	1-9-8-2-16-10-5-13-4-15-12-6-11-3-14-7-1
14	Relocated	10	9	1-10-9-8-2-16-14-13-4-15-12-6-11-3-5-7-1
15	Relocated	3	2	1-9-8-3-2-16-10-14-13-4-15-12-6-11-5-7-1
16	Exchange	9	3	1-3-8-2-16-10-14-13-4-15-12-6-11-9-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-9-8-2-16-10-14-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-5-7-13-1	120,8 km
2	Rute 1 : 1-10-9-8-2-16-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-5-7-1	119 km
3	Rute 1 : 1-9-2-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-8-11-3-5-7-1	128,8 km
4	Rute 1 : 1-9-8-2-16-10-14-13-15-12-1 Rute 2 : 1-6-11-1 Rute 3 : 1-4-1 Rute 4 : 1-3-5-7-1	160 km
5	Rute 1 : 1-9-8-2-16-10-14-12-13-1	116,2 km

	Rute 2 : 1-4-1 Rute 3 : 1-15-6-11-3-5-7-1	
6	Rute 1 : 1-9-8-2-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-7-6-11-3-5-12-1	129 km
7	Rute 1 : 1-9-8-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1--15-12-6-11-3-5-1 Rute 4 : 1-2-7-1	141,5 km
8	Rute 1 : 1-9-2-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-8-12-6-11-3-5-7-1	118,1 km
9	Rute 1 : 1-9-2-16-10-14-8-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-5-7-1-1	130,4 km
10	Rute 1 : 1-9-8-2-14-10-16-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-5-7-1	134,8 km
11	Rute 1 : 1-9-8-2-16-10-3-11-6-1 Rute 2 : 1-12-15-1 Rute 3 : 1-4-1 Rute 4 : 1-13-14-5-7-1	152,6 km
12	Rute 1 : 1-9-8-2-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-11-3-5-7-1	128,1 km
13	Rute 1 : 1-9-8-2-16-10-5-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-14-7-1	158,8 km
14	Rute 1 : 1-10-9-8-2-16-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-3-5-7-1	119 km
15	Rute 1 : 1-9-8-3-2-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-5-7-1-1	138,1 km
16	Rute 1 : 1-3-8-2-16-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-12-6-11-9-5-7-1	152,8 km

Solusi *Neighborhood* VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-9-8-2-6-12-15-1	180 liter	52,8 km
2	1-4-1	300 liter	9,2 km
3	1-13-14-10-16-11-3-5-7-1	300 liter	66,1 km
Total jarak tempuh kendaraan			128,1 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

Solusi *Neighborhood* TSP terbaik iterasi 9 : 1-9-8-2-6-12-15-4-13-14-10-16-11-3-5-7-1

ITERASI 10

Tabu List								
	1	2	3	4	5	6	7	8
Move	12	16	13	14	4	13	10	8
	15	6	14	4	14	14	9	2

No	Metode	Move		Solusi <i>Neighborhood</i> TSP
1	Exchange	2	9	1-2-8-9-6-12-15-4-13-14-10-16-11-3-5-7-1
2	Exchange	13	9	1-13-8-2-6-12-15-4-9-14-10-16-11-3-5-7-1
3	Relocated	15	9	1-15-9-8-2-6-12-4-13-14-10-16-11-3-5-7-1
4	2-Opt	4	3	1-9-8-2-6-12-15-3-11-16-10-14-13-4-5-7-1
5	2-Opt	11	7	1-9-8-2-6-12-15-4-13-14-10-16-7-5-3-11-1
6	2-Opt	15	10	1-9-8-2-6-12-10-14-13-4-15-16-11-3-5-7-1
7	Exchange	6	14	1-9-8-2-14-12-15-4-13-6-10-16-11-3-5-7-1
8	2-Opt	9	15	1-15-12-6-2-8-9-4-13-14-10-16-11-3-5-7-1
9	2-Opt	4	7	1-9-8-2-6-12-15-7-5-3-11-16-10-14-13-4-1
10	2-Opt	4	11	1-9-8-2-6-12-15-11-16-10-14-13-4-3-5-7-1
11	Relocated	3	11	1-9-8-2-6-12-15-4-13-14-10-16-3-11-5-7-1
12	Exchange	12	14	1-9-8-2-6-14-15-4-13-12-10-16-11-3-5-7-1
13	Exchange	6	9	1-6-8-2-9-12-15-4-13-14-10-16-11-3-5-7-1
14	Relocated	7	3	1-9-8-2-6-12-15-4-13-14-10-16-11-7-3-5-1
15	Relocated	14	6	1-9-8-2-14-6-12-15-4-13-10-16-11-3-5-7-1
16	Exchange	14	9	1-14-8-2-6-12-15-4-13-9-10-16-11-3-5-7-1

No	Solusi <i>Neighborhood</i> VRP	Jarak Total Rute
1	Rute 1 : 1-2-8-9-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-11-3-5-7-1	128,9 km
2	Rute 1 : 1-13-8-2-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-9-14-10-16-11-3-5-7-1	136 km
3	Rute 1 : 1-15-9-8-2-6-12-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-11-3-5-7-1	121,3 km
4	Rute 1 : 1-9-8-2-6-12-15-3-1 Rute 2 : 1-11-16-10-14-13-1 Rute 3 : 1-4-1 Rute 4 : 1-5-7-1	174,9 km

5	Rute 1 : 1-9-8-2-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-7-5-3-11-1	129,5 km
6	Rute 1 : 1-9-8-2-6-12-10-14-13-1 Rute 2 : 1-4-1 Rute 3 : 1-15-16-11-3-5-7-1	122,7 km
7	Rute 1 : 1-9-8-2-14-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-6-10-16-11-3-5-7-1-1	145,3 km
8	Rute 1 : 1-15-12-6-2-8-9-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-11-3-5-7-1	128,1 km
9	Rute 1 : 1-9-8-2-6-12-15-7-5-3-1 Rute 2 : 1-11-16-10-14-13-1 Rute 3 : 1-4-1	155,7 km
10	Rute 1 : 1-9-8-2-6-12-15-11-16-1 Rute 2 : 1-10-14-13-1 Rute 3 : 1-4-1 Rute 4 : 1-3-5-7-1	152,7 km
11	Rute 1 : 1-9-8-2-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-3-11-5-7-1	130,9 km
12	Rute 1 : 1-9-8-2-6-14-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-12-10-16-11-3-1 Rute 4 : 1-5-7-1	152,1 km
13	Rute 1 : 1-6-8-2-9-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-11-3-5-7-1	131,4 km
14	Rute 1 : 1-9-8-2-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-14-10-16-11-7-3-5-1	140,6 km
15	Rute 1 : 1-9-8-2-14-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-10-16-11-3-5-7-1	137,8 km
16	Rute 1 : 1-14-8-2-6-12-15-1 Rute 2 : 1-4-1 Rute 3 : 1-13-9-10-16-11-3-5-7-1-1	134,6 km

Solusi *Neighborhood VRP* terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-9-8-2-6-12-15-1	180 liter	58,2 km
2	1-4-1	300 liter	9,2 km
3	1-13-14-10-16-3-11-5-7-1	300 liter	62,5 km
Total jarak tempuh kendaraan			130,9 km

Solusi VRP terbaik

Rute	Solusi VRP	Permintaan	Jarak
1	1-10-12-8-16-2-9-14-13-1	240 liter	29 km
2	1-4-1	300 liter	9,2 km
3	1-15-6-11-3-5-7-1	240 liter	62,9 km
Total jarak tempuh kendaraan			101,1 km

LAMPIRAN 2

Source Code M.Files

```
function varargout = GUI_TABUSEARCH(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @GUI_TABUSEARCH_OpeningFcn, ...
'gui_OutputFcn',  @GUI_TABUSEARCH_OutputFcn, ...
'gui_LayoutFcn',  [], ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function GUI_TABUSEARCH_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
hback = axes('unit','normalized','position',[0 0 1 1]);
uistack(hback,'bottom');
[back, map]=imread('BG_GUI.jpg');
image(back)
colormap(map)
set(hback,'handlevisibility','off','visible','off')
guidata(hObject, handles);
set(handles.Jarak,'visible','off');
set(handles.text8,'visible','off');
set(handles.jumlahkapasitas,'visible','off');
set(handles.text11,'visible','off');
axes(handles.map);
img = imread ('mapyogyakarta.jpg');
min_x = 0;
max_x = 100;
min_y = 0;
max_y = 100;
imshow(img);
x_min = min_x;
x_max = max_x;
y_min = min_y;
y_max = max_y;
imagesc ([x_min x_max ], flipud([y_min y_max]), img);
set(gca,'Ydir','normal')
xlabel('sumbu-x'), ylabel('sumbu-y')

function varargout = GUI_TABUSEARCH_OutputFcn(hObject, eventdata,
handles)
```

```
varargout{1} = handles.output;

function coX_Callback(hObject, eventdata, handles)

function coX_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function coY_Callback(hObject, eventdata, handles)

function coY_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Kapasitas_Callback(hObject, eventdata, handles)

function Kapasitas_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Permintaan_Callback(hObject, eventdata, handles)

function Permintaan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function JumlahNeighborhood_Callback(hObject, eventdata, handles)

function JumlahNeighborhood_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Jarak_Callback(hObject, eventdata, handles)

function Jarak_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jaraktotal_Callback(hObject, eventdata, handles)

function jaraktotal_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```

        set(hObject, 'BackgroundColor', 'white');
    end

function Tabulist_Callback(hObject, eventdata, handles)

function Tabulist_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function jumrute_Callback(hObject, eventdata, handles)

function jumrute_CreateFcn(hObject, eventdata, ~)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function jumlahkapasitas_Callback(hObject, eventdata, handles)

function jumlahkapasitas_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function hasilrute_Callback(hObject, eventdata, handles)

function hasilrute_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function InputJarak_Callback(hObject, eventdata, handles)
set(handles.Jarak, 'visible', 'on');
set(handles.text8, 'visible', 'on');

function Proses_Callback(hObject, eventdata, handles)
set(handles.Jarak, 'visible', 'off');
set(handles.text8, 'visible', 'off');
x=str2num(get(handles.coX, 'string'));
y=str2num(get(handles.coY, 'string'));
KK=str2num(get(handles.Kapasitas, 'string'));
D=str2num(get(handles.Permintaan, 'string'));
MI=str2num(get(handles.iterasi, 'string'));
JSN=str2num(get(handles.JumlahNeighborhood, 'string'));
Pnjng_TL = str2num(get(handles.Tabulist, 'string'));;%panjang Tabu
list
TL = ones(Pnjng_TL, 2);
MJ = str2num(get(handles.Jarak, 'string'));
JSM = sum(sum(MJ));
Node = length(x)-1;

```

```

tic
clc
SA = solusiNN(Node,MJ);
SVRPA = RubahSolusiVRP(SA, D, KK);
JSA = HitungTD(SVRPA, MJ);
ST = SA;
SVRPT = SVRPA;
JST = JSA;
STerkini = SA;
SVRPTerkini = SVRPA;
JSTerkini = JSA;
DaftarSTSPN = zeros(JSN, Node+ 2);
DaftarSVRPN = zeros(JSN, Node* 2 + 1);
DaftarJSVRPN = zeros(1, JSN);
DaftarMetode = zeros(JSN, 2);
DaftarPM = zeros(JSN,1);
STSPNTerbaik = zeros(1, Node+ 2);
SVRPNTerbaik = zeros(1, Node* 2 + 1);
JSVRPNTerbaik = 0;
STSPNTabuTerbaik = zeros(1, Node+ 2);
SVRPNTabuTerbaik = zeros(1, Node* 2 + 1);
JSVRPNTabuTerbaik = 0;

for i = 1 : MI
for j = 1 : JSN
    Pilihan = randi(3);
switch (Pilihan)
case 1
    [DaftarSTSPN(j, :) P_1 P_2 ] =
MetodeRelocated(STerkini);
    DaftarSVRPN(j, :) = RubahSolusiVRP(DaftarSTSPN(j,
:), D, KK);
    DaftarJSVRPN(j) = HitungTD(DaftarSVRPN(j, :), MJ);
case 2
    [DaftarSTSPN(j, :) P_1 P_2 ] =
MetodeExchange(STerkini);
    DaftarSVRPN(j, :) = RubahSolusiVRP(DaftarSTSPN(j,
:), D, KK);
    DaftarJSVRPN(j) = HitungTD(DaftarSVRPN(j, :), MJ);
case 3
    [DaftarSTSPN(j, :) P_1 P_2 ] =
Metode2Opt(STerkini);
    DaftarSVRPN(j, :) = RubahSolusiVRP(DaftarSTSPN(j,
:), D, KK);
    DaftarJSVRPN(j) = HitungTD(DaftarSVRPN(j, :), MJ);
end
    DaftarPM(j,:)=[Pilihan];
    DaftarMetode(j, :) = [P_1 P_2];
end
    ApakahTabu = zeros(1, JSN);
    for j = 1 : JSN
        for k = 1 : Pnjng_TL
            if DaftarMetode(j, 1) == TL(k,
1)||DaftarMetode(j, 1) == TL(k,
2)||DaftarMetode(j, 2) == TL(k,
1)||DaftarMetode(j, 2) == TL(k, 2)

```



```

                                ApakahTabu(j) = 1;
                                end
                                end
                                end

                                IndeksTabuTerbaik = 1;
                                IndeksTidakTabuTerbaik = 1;
                                JSVRPNTerbaik = JSM;
                                JSVRPNTabuTerbaik = JSM;
                                for j = 1 : JSN
                                if ApakahTabu(j) == 0
                                if DaftarJSVRPN(j) < JSVRPNTerbaik
                                        STSPNTerbaik = DaftarSTSPN(j, :);
                                        SVRPNTerbaik = DaftarSVRPN(j, :);
                                        JSVRPNTerbaik = DaftarJSVRPN(j);
                                        IndeksTidakTabuTerbaik = j;
                                end
                                else
                                if DaftarJSVRPN(j) < JSVRPNTabuTerbaik
                                        STSPNTabuTerbaik = DaftarSTSPN(j, :);
                                        SVRPNTabuTerbaik = DaftarSVRPN(j, :);
                                        JSVRPNTabuTerbaik = DaftarJSVRPN(j);
                                        IndeksTabuTerbaik = j;
                                end
                                end
                                end

                                if JSVRPNTabuTerbaik < JST
                                        ST = STSPNTabuTerbaik;
                                        SVRPT = SVRPNTabuTerbaik;
                                        JST = JSVRPNTabuTerbaik;
                                        STerkini = STSPNTabuTerbaik;
                                        SVRPTerkini = SVRPNTabuTerbaik;
                                        JSTerkini = JSVRPNTabuTerbaik;
                                        IndeksTL = mod(i, Pnjng_TL) + 1;
                                        TL(IndeksTL, :) = DaftarMetode(IndeksTabuTerbaik, :);
                                else
                                        STerkini = STSPNTerbaik;
                                        SVRPTerkini = SVRPNTerbaik;
                                        JSTerkini = JSVRPNTerbaik;
                                if JSVRPNTerbaik < JST
                                        ST = STSPNTerbaik;
                                        SVRPT = SVRPNTerbaik;
                                        JST = JSVRPNTerbaik;
                                end

                                IndeksTL = mod(i, Pnjng_TL) + 1;
                                TL(IndeksTL, :) = DaftarMetode(IndeksTidakTabuTerbaik, :)

                                end
                                end
                                toc
                                axes(handles.map);
                                TotalDistance=JST
                                img = imread ('mapyogyakarta.jpg');
                                min_x = 0;
                                max_x = 100;
                                min_y = 0;

```

```

max_y = 100;
x=X;
y=Y;

x_min = min_x;
x_max = max_x;
y_min = min_y;
y_max = max_y;
imagesc ([x_min x_max ], flipud([y_min y_max])), img);
set(gca, 'Ydir', 'normal')
xlabel('sumbu-x'), ylabel('sumbu-y')
hold on

    shortestPath=SVRPT;

    xd=[x(shortestPath)];
    yd=[y(shortestPath)];
    text(xd(1),yd(1), [' Depot ']);
for i=2:length(shortestPath)-1
if shortestPath(i)~=shortestPath(1)
    text(xd(i),yd(i), [' ', num2str(shortestPath(i))]);
    rute(i)=shortestPath(i);
end
end
    plot(xd,yd, '-', 'LineWidth', 1, 'MarkerEdgeColor', 'b', ...
'MarkerFaceColor', 'b', ...
'MarkerSize', 10)

    hold on

for k=1:length(rute)-1
if rute(k)==0
    rute(k)=1;
end
end
    rute(length(rute)+1)=1

for ii=1:length(rute)
    rutes(1,ii)=rute(ii);
end
    k=0
for zz=1:length(rute)
    hold on
    colour=mod(k,6);
    ed=[x(rute(zz))]
fd=[y(rute(zz))]
    if colour==1
        plot(ed,fd, '-o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'g', ...
'MarkerSize', 10)
        hold on
    elseif colour==2
        plot(ed,fd, '-o', 'LineWidth', 2, 'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'r', ...

```

```

        'MarkerSize',10)
        hold on
    elseif colour==3
        plot(ed,fd,'-o','LineWidth',2,'MarkerEdgeColor','k',...
        'MarkerFaceColor','y',...
        'MarkerSize',10)
        hold on
    elseif colour==4
        plot(ed,fd,'-o','LineWidth',2,'MarkerEdgeColor','k',...
        'MarkerFaceColor','w',...
        'MarkerSize',10)
        hold on
    end
    if rute(zz)==1
        k=k+1;
        plot(ed,fd,'o','LineWidth',2,'MarkerEdgeColor','r',...
        'MarkerFaceColor','k',...
        'MarkerSize',13)
        hold on
    end
    end
    d=find(rutes==1);
    for uu=1:length(d)-1
        ab(1,uu)=uu;
    end
    kp=0;
    index=1;
    for ee=2:length(rute)
        kap=kp+D(rute(ee));
        kp=kap;
        total(index)=kp;
        if rute(ee)==1
            index=index+1;
            kp=0;
        end
    end
    TJPR = 0 ;
    index2=1;
    for ii = 1 : length(rute)-1
        totalJPR = TJPR + MJ (rute (ii), rute (ii+1));
        TJPR=totalJPR;
        totalJ(index2)=TJPR;
        if rute(ii+1)==1
            index2=index2+1;
            TJPR=0;
        end
    end
    end

    hold on
    set(handles.jaraktotal,'string',TotalDistance);
    set(handles.hasilrute,'string',rutes);
    set(handles.jumlahkapasitas,'string',total);
    set(handles.jumrute,'string',length(d)-1);

    function Kosongkan_Callback(hObject, eventdata, handles)
    set(handles.hasilrute,'string','');

```

```

set(handles.coX,'string','');
set(handles.coY,'string','');
set(handles.Kapasitas,'string','');
set(handles.Permintaan,'string','');
set(handles.iterasi,'string','');
set(handles.jaraktotal,'string','');
set(handles.JumlahNeighborhood,'string','');
set(handles.Jarak,'string','');

function keluar_Callback(hObject, eventdata, handles)
close

function lihatpermintaan_Callback(hObject, eventdata, handles)
set(handles.jumlahkapasitas,'visible','on');
set(handles.text11,'visible','on');

function tutuppermintaan_Callback(hObject, eventdata, handles)
set(handles.jumlahkapasitas,'visible','off');
set(handles.text11,'visible','off');

function iterasi_Callback(hObject, eventdata, handles)

function iterasi_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function lihathelp_Callback(hObject, eventdata, handles)
set(handles.instruksihelp,'visible','on');

function tutuphelp_Callback(hObject, eventdata, handles)
set(handles.instruksihelp,'visible','off');

function totaljarak = HitungTD (solusi , MatriksJarak)
Node = numel (solusi) - 1;
totaljarak = 0 ;
for i = 1 : Node
    totaljarak = totaljarak + MatriksJarak (solusi (i), solusi
(i+1));
end

function [Solusi2Opt P_1 P_2]= Metode2Opt (SA)
Node = numel(SA) - 2;
Index_1 = randi(Node) + 1;
Index_2 = Index_1;
while Index_1 >= Index_2
    Index_1 = randi(Node) + 1;
    Index_2 = randi(Node) + 1;
end
P_1 = SA(Index_1);
P_2 = SA(Index_2);
Solusi2Opt = SA;

```

```

Solusi2Opt(Index_1:Index_2) = SA(Index_2:-1:Index_1);

function [SolusiEx P_1 P_2] = MetodeExchange(SA)
Node = numel(SA) - 2;
Index_1 = randi(Node) + 1;
Index_2 = Index_1;
while Index_2 == Index_1
    Index_2 = randi(Node) + 1;
end
P_1 = SA(Index_1);
P_2 = SA(Index_2);
SolusiEx = SA;
SolusiEx(Index_1) = SA(Index_2);
SolusiEx(Index_2) = SA(Index_1);
end

function [SolusiRE P_1 P_2] = MetodeRelocated(SA)
Node = numel(SA) - 2;
Index_1 = randi(Node) + 1;
Index_2 = Index_1;
while Index_2 == Index_1
    Index_2 = randi(Node) + 1;
end
P_1 = SA(Index_1);
P_2 = SA(Index_2);
SolusiRE = SA;
if Index_1 > Index_2
    SolusiRE(Index_2) = SA(Index_1);
    SolusiRE(Index_2 + 1 : Index_1) = SA(Index_2 : Index_1 - 1);
else
    SolusiRE(Index_2) = SA(Index_1);
    SolusiRE(Index_1 : Index_2 - 1) = SA(Index_1 + 1 : Index_2);
end

function solusivrp = RubahSolusiVRP (solusitsp, demandkota,
kapasitaskendaraan)
jumlahkota = numel(solusitsp) - 2;
solusivrp = ones (1, jumlahkota * 2 + 1);
demandsekarang = 0;
indexvrpsekarang = 1;

for i =2 : jumlahkota + 1
if demandsekarang + demandkota(solusitsp(i)) <= kapasitaskendaraan
% tidak perlu buat rute baru
    demandsekarang = demandsekarang +
demandkota(solusitsp(i));
    indexvrpsekarang = indexvrpsekarang + 1;
    solusivrp (indexvrpsekarang) = solusitsp (i);
else% perlu buat rute baru
% sisipkan depot kota 1
%demandsekarang = 0;
    indexvrpsekarang = indexvrpsekarang + 1;
    solusivrp (indexvrpsekarang) = 1;
% sisipkan kota berikutnya di rutebaru
    demandsekarang = demandkota(solusitsp(i));
    indexvrpsekarang = indexvrpsekarang + 1;

```

```

        solusivrp (indexvrpsekarang) = solusitsp (i);
end

end

function solusiNN = solusiNN (Node,MatriksJarak)

N_cities = Node+1;
distances = MatriksJarak;
distances(distances==0) = realmax;

shortestPathLength = realmax;

for i = 1:N_cities

    startCity = 1;
    path = startCity;
    distanceTraveled = 0;
    distancesNew = distances;
    currentCity = startCity;

    for j = 1:N_cities-1

        [minDist,nextCity] = min(distancesNew(:,currentCity));
        if (length(nextCity) > 1)
            nextCity = nextCity(1);
        end

        path(1,end+1) = nextCity;
        distanceTraveled = distanceTraveled +...
            distances(currentCity,nextCity);

        distancesNew(currentCity,:) = realmax;

        currentCity = nextCity;

    end

    path(1,end+1) = startCity;
    distanceTraveled = distanceTraveled +...
        distances(currentCity,startCity);

    if (distanceTraveled < shortestPathLength)
        shortestPathLength = distanceTraveled;
        shortestPath = path;
    end

end

solusiNN=shortestPath;

```

TUTORIAL

VEHICLE ROUTING PROBLEM ALGORITMA TABU SEARCH

Oleh :

Sulistiono (11610033)

Dosen Pembimbing : Noor Saif Muhammad Mussafi, M.Sc

Program Studi Matematika

Fakultas Sains dan Teknologi



UIJ
STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

KATA PENGANTAR

Pertama-tama, penulis bersyukur kepada Allah SWT, karena hanya dengan limpahan rahmat dan karunia-Nya penulis bisa menyelesaikan tutorial program *Vehicle Routing Problem Algoritma Tabu Search* ini. Tutorial ini membahas penggunaan program *Vehicle Routing Problem Algoritma Tabu Search* secara praktis. Program ini dibuat menggunakan MATLAB 8.1 dengan berbasis GUI (*Graphic User Interface*). Tutorial ini dimulai dari landasan pendahuluan, landasan teori yang digunakan, kemudian cara kerja dan penggunaan program tersebut. Pembuatan tutorial ini diharapkan dapat mempermudah pemahaman sekaligus bisa digunakan sebagai rujukan yang bermanfaat. Penulis menyampaikan rasa terima kasih dan penghargaan setinggi-tingginya kepada keluarga dan rekan-rekan yang telah mendorong penulis untuk menyelesaikan tutorial ini. Penulis berharap buku ini akan bermanfaat bagi banyak pihak, aamiin.

Yogyakarta, 10 November 2015

Sulistiono

1. PENDAHULUAN

Pada dunia industri, logistik memiliki peranan penting dalam meningkatkan kinerja suatu perusahaan. Kemampuan perusahaan untuk mengelola logistik secara efektif dan efisien dapat mempengaruhi biaya dan tingkat pelayanan terhadap konsumen sehingga dapat bersaing dengan perusahaan sejenis lainnya. Salah satu usaha yang dapat dilakukan perusahaan untuk mengoptimalkan pendistribusian produk adalah meminimalkan biaya transportasi melalui penentuan rute optimal kendaraan yang disebut *Vehicle Routing Problem* (VRP). Kasus VRP merupakan TSP dengan menyertakan kendala satu kendaraan dengan kapasitas sehingga digolongkan ke dalam *NP-Hard Problem* karena secara teori ataupun praktik pada dunia nyata memiliki permasalahan yang sangat banyak dan kompleks sehingga sulit untuk dipecahkan. Kasus *NP-Hard* dapat diselesaikan menggunakan pendekatan solusi optimal dengan metode heuristik yang memberikan perkiraan solusi. Dibandingkan dengan metode heuristik klasik, metaheuristik menunjukkan pencarian solusi yang lebih teliti.

Algoritma *Tabu Search* merupakan salah satu metode metaheuristik yang dapat menuntun prosedur pencarian lokal heuristik untuk menjelajahi daerah solusi di luar titik optimal lokal. Algoritma *Tabu Search* dapat digunakan untuk mencari solusi optimal VRP yaitu rute yang memiliki total jarak tempuh minimum dengan mempertimbangkan kapasitas kendaraan. Langkah Algoritma *Tabu Search* dimulai dengan penentuan *initial solution* menggunakan *Nearest Neighbor*, evaluasi *move* menggunakan metode *2-Opt*, *Relocated*, dan *Exchange*, *update Tabu List*, kemudian apabila kriteria pemberhentian terpenuhi maka proses Algoritma *Tabu Search* berhenti jika tidak, maka kembali pada evaluasi *move*. Pembuatan suatu program (rancang bangun) dapat mempercepat proses pencarian solusi optimal pada VRP. Program (rancang bangun) dibuat menggunakan MATLAB yang dimulai dengan membuat *source code* utama menggunakan *m.file* kemudian desain tampilan dirancang menggunakan *fig-file* sehingga diperoleh program dalam bentuk GUI (*Graphical User Interface*). Program (rancang bangun) Algoritma *Tabu Search* dapat memudahkan pencarian solusi optimal VRP yang lebih efektif dan efisien.

2. TUJUAN

Diharapkan setelah membaca tutorial ini dapat:

- a. Mengetahui program *Vehicle Routing Problem* Algoritma *Tabu Search*.
- b. Mengetahui cara kerja dan penggunaan program *Vehicle Routing Problem* Algoritma *Tabu Search*.
- c. Mampu mengerti bagaimana menyelesaikan permasalahan pendistribusian.

3. LANDASAN TEORI

3.1 *Capacitated Vehicle Routing Problem (CVRP)*

Vehicle Routing Problem (VRP) merupakan bagian dari TSP, artinya VRP merupakan TSP dengan menyertakan kendala satu kendaraan dengan kapasitas [26]. Beberapa komponen beserta karakteristiknya yang terdapat dalam masalah VRP menurut Toth dan Vigo (2002), yaitu depot, jaringan jalan, konsumen, kendaraan dan pengemudi

Capacitated Vehicle Routing Problem (CVRP) merupakan salah satu variasi dari masalah VRP dengan kendala kapasitas kendaraan yang terbatas. CVRP dapat direpresentasikan sebagai suatu graf berarah berbobot (*weighted directed graph*) $D = (V, A)$ dimana $V = \{v_0, v_1, v_2, \dots, v_n\}$ adalah himpunan simpul (*nodes*) dan $A = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ adalah himpunan busur (*arcs*) yang menghubungkan himpunan simpul (*nodes*). Simpul v_0 dinyatakan sebagai depot dan yang lainnya adalah pelanggan. Setiap elemen dari busur (*arcs*) menyatakan jarak. Sedangkan setiap simpul memiliki permintaan (*demand*) yang dinotasikan sebagai q_i , dengan $i = 1, 2, 3, \dots, n$. Himpunan $K = \{k_1, k_2, k_3, \dots, k_n\}$ merupakan kumpulan kendaraan yang homogen. Kapasitas kendaraan yang digunakan dinotasikan dengan Q [3].

Diberikan d_{ij} adalah jarak dari simpul i ke simpul j (d_{ij} merupakan bilangan *nonnegative*). Jarak diasumsikan semetrik ($d_{ij} = d_{ji}$) dan ($d_{ii} = d_{jj} = 0$). Permasalahan tersebut kemudian dapat dibuat menjadi model matematika dengan tujuan meminimumkan total jarak tempuh perjalanan kendaraan:

Didefinisikan variabel keputusan

$$x_{i,j}^k = \begin{cases} 1, & \text{jika kendaraan } k \text{ mengunjungi simpul } v_j \text{ setelah simpul } v_i \\ 0, & \text{selainnya} \end{cases}$$

$$y_i^k = \begin{cases} 1, & \text{jika simpul } v_i \text{ dilayani oleh kendaraan } k \\ 0, & \text{selainnya} \end{cases}$$

Keterangan variabel

- $D = (V,A)$
- $V =$ himpunan simpul $\{v_0, v_1, v_2, \dots, v_n\}$, dimana v_0 adalah depot dan v_1, v_2, \dots, v_n adalah pelanggan
- $A =$ himpunan sisi berarah (arcs) , $\{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$
- $d_{ij} =$ jarak antara simpul v_i ke simpul v_j
- $q_i =$ permintaan pelanggan ke $i, i \in V$
- $K = \{k_1, k_2, k_3, \dots, k_n\}$ kendaraan seragam yang digunakan
- Q adalah kapasitas masing-masing kendaraan $k_i \in K, i = \{1, 2, 3, \dots, n\}$

Fungsi tujuannya meminimumkan total jarak tempuh kendaraan. Jika Z adalah fungsi tujuan, maka

$$\text{Min } Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{i,j}^k \quad (6)$$

Kendala

- a. Setiap simpul hanya boleh dikunjungi tepat satu kali oleh 1 kendaraan.

$$\sum_{k \in K} \sum_{j \in N} x_{i,j}^k = 1, \quad \forall i \in V \quad (7)$$

- b. Kendaraan yang telah mengunjungi simpul i , kendaraan k harus meninggalkan simpul tersebut menuju simpul lain.

$$\sum_{j \in V} x_{i,j}^k - \sum_{j \in V} x_{j,i}^k = 0, \quad \forall i \in V, \forall k \in K \quad (8)$$

- c. Total jumlah permintaan pelanggan dalam satu rute tidak melebihi kapasitas kendaraan.

$$\sum_{i \in V} q_i \sum_{j \in V, j \neq i} x_{ij}^k \leq Q, \forall k \in K \quad (9)$$

- d. Setiap rute perjalanan kendaraan berawal dari depot

$$\sum_{j \in V} x_{0,j}^k = 1, \forall k \in K \quad (10)$$

- e. Setiap rute perjalanan kendaraan berakhir di depot

$$\sum_{j \in V} x_{j,0}^k = 1, \forall k \in K \quad (11)$$

- f. Batasan ini memastikan bahwa tidak terdapat subrute pada setiap rute yang terbentuk.

$$x_{i,j}^k = 1 \Rightarrow y_i - q_j = y_j, \forall i, j \in V, k \in K \quad (12)$$

$$y_0 = Q, 0 \leq y_i, \forall i \in V \quad (13)$$

- g. Variable keputusan $x_{i,j}^k$ merupakan bilangan *biner*.

$$x_{i,j}^k \in \{0,1\}, \forall i, j \in V, k \in K \quad (14)$$

Variabel keputusan hanya akan terdefinisi jika jumlah permintaan simpul v_i dan simpul v_j tidak melebihi kapasitas kendaraan. Apabila kapasitas kendaraan tidak memadai untuk pelanggan berikutnya maka kendaraan harus mengisi muatan di depot sehingga akan terbentuk rute baru.

3.2 Algoritma *Tabu Search*

Algoritma *Tabu Search* memiliki lima elemen utama yang digunakan untuk menyelesaikan VRP yaitu :

- a. Representasi Solusi

Representasi solusi yang digunakan Algoritma *Tabu Search* adalah suatu urutan titik-titik (*nodes*), dimana tiap titik (*node*) hanya terlihat sekali dalam urutan. Titik (*node*) tersebut merepresentasikan depot dan pelanggan.

- b. Pembentukan Solusi Awal (*Initial Solution*)

Solusi awal dibentuk menggunakan metode random atau metode heuristik yang akan diperbaiki pada iterasi berikutnya.

- c. Solusi Neighborhood

Solusi *Neighborhood* merupakan solusi alternatif yang diperoleh dengan melakukan perpindahan *node* (*move*). Setiap perpindahan *node* (*move*) akan menghasilkan satu solusi *Neighborhood*.

d. *Tabu List*

Tabu list berisi atribut *move* yang telah ditemukan sebelumnya. Ukuran *Tabu List* akan bertambah seiring meningkatnya ukuran masalah. Ukuran *Tabu List* yang terlalu panjang tidak akan menghasilkan kualitas solusi yang baik karena dapat menyebabkan terlalu banyak perpindahan *node* (*move*) yang dilarang (Glover dan Kochenberger, 2003).

e. Kriteria Aspirasi

Kriteria aspirasi adalah suatu metode untuk membatalkan status tabu (Glover dan Kochenberger, 2003).

f. Kriteria Pemberhentian.

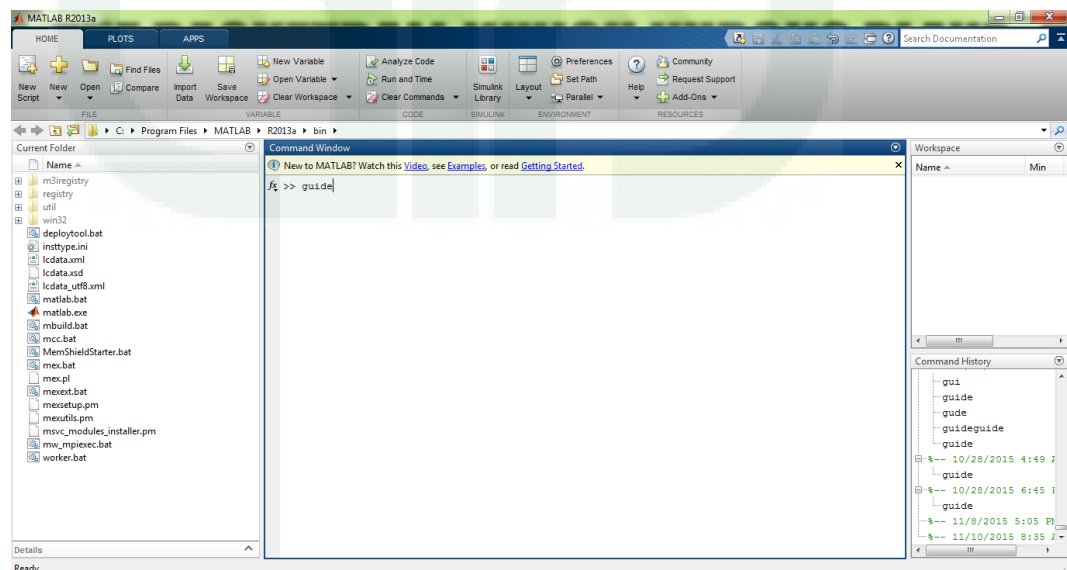
Kriteria pemberhentian (*termination criteria*) yang dipakai yaitu setelah semua iterasi yang telah ditentukan terpenuhi.

4. Tutorial Penggunaan Program *Vehicle Routing Problem* Algoritma *Tabu Search*.

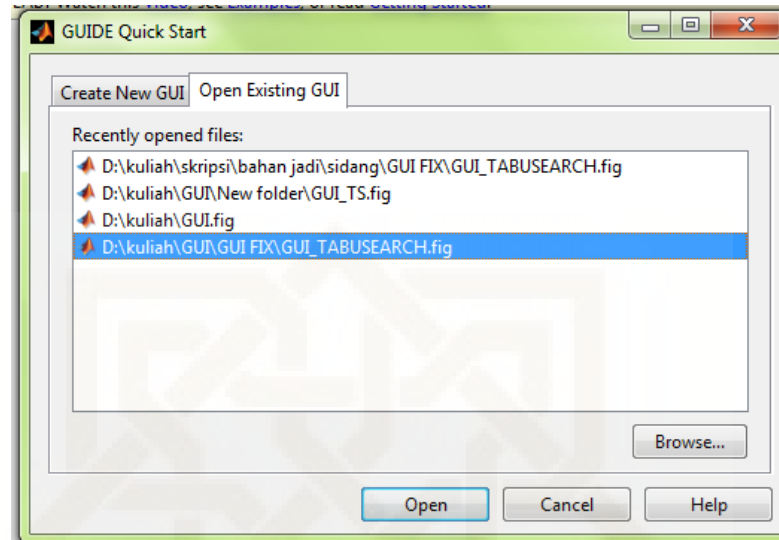
4.1 Input

Program ini dibuka menggunakan MATLAB 8.1 dengan cara sebagai berikut:

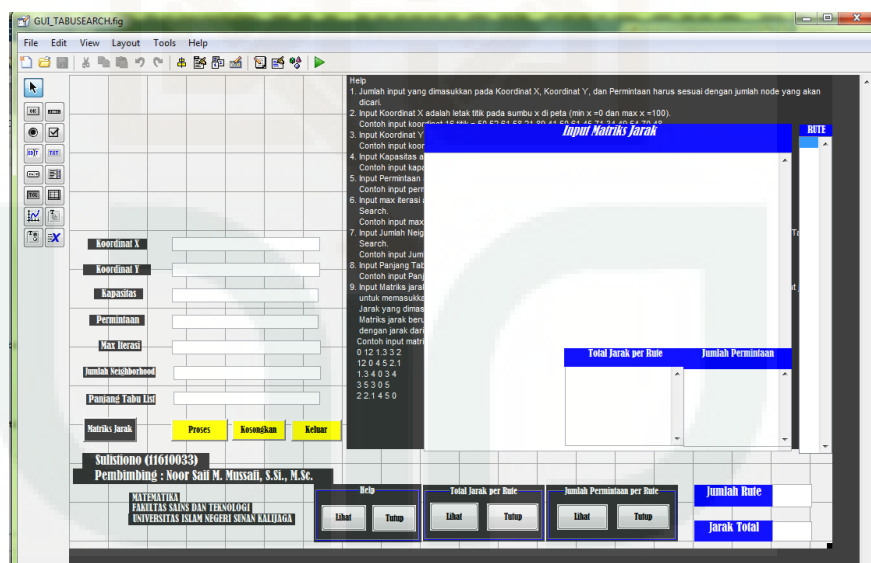
a. Buka MATLAB 8.1




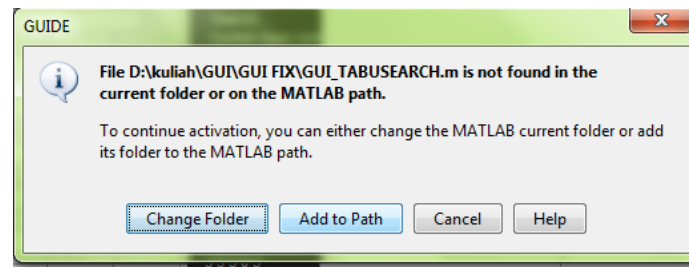
Lalu ketikkan “*guide*” agar MATLAB langsung membuka file bertipe GUI dalam komputer. Setelah itu akan muncul tampilan sebagai berikut:



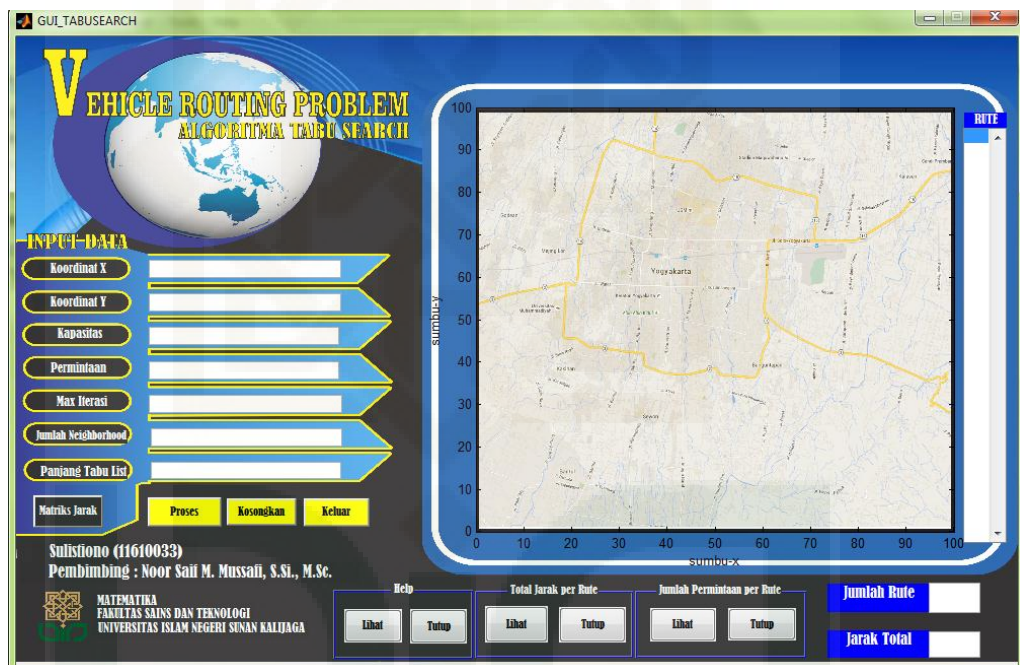
Pada tampilan tersebut pilih file GUI_TABUSEARCH.fig kemudian klik *open*. Setelah itu akan muncul tampilan sebagai berikut:



Tampilan tersebut adalah tampilan *fig file* pada MATLAB yang berfungsi untuk mendesain GUI sehingga dapat dibuat sesuai keinginan. Running program dapat dilakukan dengan cara klik tombol run program (). Setelah itu akan muncul *pop up* dan klik *Add to Path*.

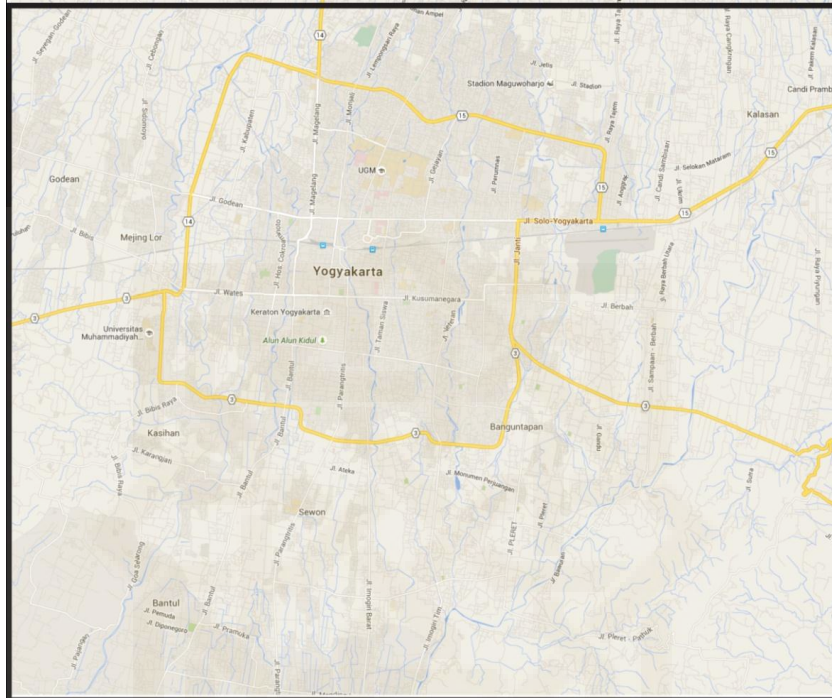


Setelah itu kemudian akan muncul tampilan awal program. Dalam tampilan awal program ini berisi berbagai macam input data yang akan digunakan pada program. Berikut adalah tampilan awal program:



Pada Gambar tersebut koordinat kartesius sumbu x dan y dari 0 sampai 100 pada peta tidak merepresentasikan jarak per kilometer. Koordinat x dan y hanya digunakan untuk memudahkan *user* melakukan *plot* letak *node* agar sesuai data asli pada peta Provinsi DIY sehingga solusi optimum dapat direpresentasikan menjadi graf. Kondisi tersebut dapat dilakukan karena jarak tiap node sudah diinputkan user berdasarkan jarak sebenarnya yang telah didapatkan menggunakan google maps atau sejenisnya sehingga tidak mempengaruhi proses perhitungan solusi. Gambar peta dan rentang batas koordinat kartesius sumbu x

dan y dapat dirubah sesuai kebutuhan *user* dengan mengganti perintah pada m.file. Berikut adalah peta Provinsi DIY yang digunakan dalam program:



Berdasarkan terdapat delapan *input* yang harus dilakukan *user* yaitu koordinat x, koordinat y, kapasitas, permintaan, max iterasi, jumlah *Neighborhood*, panjang *Tabu List*, dan matriks jarak dengan rincian sebagai berikut:

- a) *Input* pada koordinat x yaitu letak titik (*node*) depot dan pelanggan pada sumbu x di Gambar mapyogyakarta.jpg. Input koordinat x dipisah menggunakan spasi.

Contoh input koordinat x :

$x = 12\ 45\ 23$

Contoh tersebut memiliki arti input koordinat $x_1 = 12$, $x_2 = 45$, dan $x_3 = 23$

- b) *Input* pada koordinat y yaitu letak titik (*node*) depot dan pelanggan pada sumbu y di Gambar mapyogyakarta.jpg. Input koordinat y juga dipisah menggunakan spasi.

Contoh input koordinat y :

$y = 10\ 21\ 76$

Contoh tersebut memiliki arti input koordinat $y_1 = 10$, $y_2 = 21$, dan $y_3 = 76$

Pada input koordinat x dan y memiliki pengertian bahwa koordinat kartesius contoh tersebut adalah $(12,10)$, $(45,21)$, dan $(23,76)$.

- c) *Input* kapasitas yaitu besarnya kapasitas maksimum kendaraan yang digunakan.
- d) *Input* permintaan yaitu permintaan tiap *node* (depot dan pelanggan). Depot tidak memiliki permintaan sehingga untuk depot permintaannya adalah 0.
- e) *Input* max iterasi yaitu maksimum iterasi yang akan digunakan pada proses Algoritma *Tabu Search*.
- f) *Input* jumlah *Neighborhood* yaitu jumlah solusi *Neighborhood* yang akan dimasukkan ke dalam daftar solusi *Neighborhood*.
- g) *Input* panjang *Tabu List* yaitu panjang *Tabu List* yang akan digunakan pada proses Algoritma *Tabu Search*.
- h) *Input* matriks jarak dilakukan dengan cara klik tombol “Matriks Jarak” sehingga akan muncul secara otomatis form input jarak untuk memasukkan jarak antar titik (*node*). Untuk tampilan form input jarak dapat dilihat pada Gambar 3.11 berikut ini:



4.2 Contoh perhitungan program

Sebuah perusahaan memiliki 1 buah kendaraan dengan kapasitas 300 liter. Kemudian perusahaan tersebut memiliki data permintaan pelanggan dan jarak depot ke pelanggan dan antar pelanggan sebagai berikut:

Tabel Data permintaan pelanggan Bioseptik

Pelanggan	Permintaan	Pelanggan	Permintaan
1	-	9	10 liter
2	50 liter	10	35 liter
3	70 liter	11	55 liter
4	300 liter	12	40 liter
5	40 liter	13	20 liter
6	20 liter	14	25 liter
7	5 liter	15	50 liter
8	10 liter	16	50 liter

Tabel Jarak depot ke pelanggan dan antar pelanggan dalam satuan kilometer

Pelanggan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	8,9	16,2	4,6	19,4	14,2	10,7	6,3	7	2,4	17,2	8,7	0,7	1,4	6,9	8,3
2		0	8,9	6	15,9	9,6	5,1	3,3	4	9,1	8,8	6,2	8,5	8,5	9,0	1
3			0	13,7	12,6	12,3	7	11,6	10,3	17,7	3,2	13,4	17,1	16,1	16,7	8,6
4				0	25,8	9,5	11,1	3,2	2,4	6,6	12,2	9,2	5	3,9	4,6	6,7
5					0	24,8	10,4	16,9	21,2	19,1	15,9	14,5	19,8	21,5	27,6	14,3
6						0	18,5	10,8	8,5	15,1	10,4	13,9	14,2	13,4	8,7	10,6
7							0	6,6	9,2	11	10,4	7,5	10,9	11,4	15	4
8								0	4	7,9	10,9	5,2	7,4	7,1	8,9	2
9									0	7,8	9,8	8,8	6,6	6,1	5	5,6
10										0	20,1	6	2,8	3,8	9,4	8,6
11											0	14,7	16,9	16,2	16,4	9,1
12												0	7,4	8,3	13,5	6,1
13													0	1,6	7,6	8,1
14														0	7	7,4
15															0	10,7
16																0

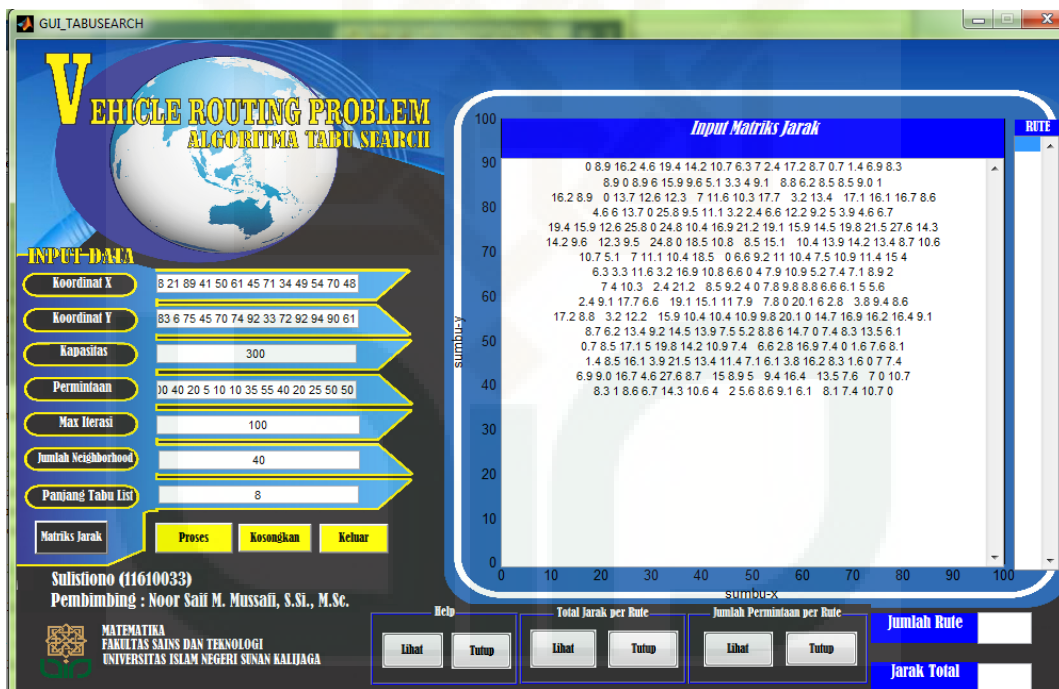
Perusahaan tersebut sudah menandai letak semua pelanggan dalam peta program sehingga dengan data jarak dan data permintaan pelanggan yang sudah diperoleh maka diinputkan data sebagai berikut:

Koordinat X : 50 52 61 58 21 89 41 50 61 45 71 34 49 54 70 48
 Koordinat Y : 96 63 30 83 6 75 45 70 74 92 33 72 92 94 90 61
 Kapasitas : 300
 Permintaan : 0 50 70 300 40 20 5 10 10 35 55 40 20 25 50 50
 Max Iterasi : 100
 Jumlah Neighborhood : 40
 Panjang Tabu List : 8

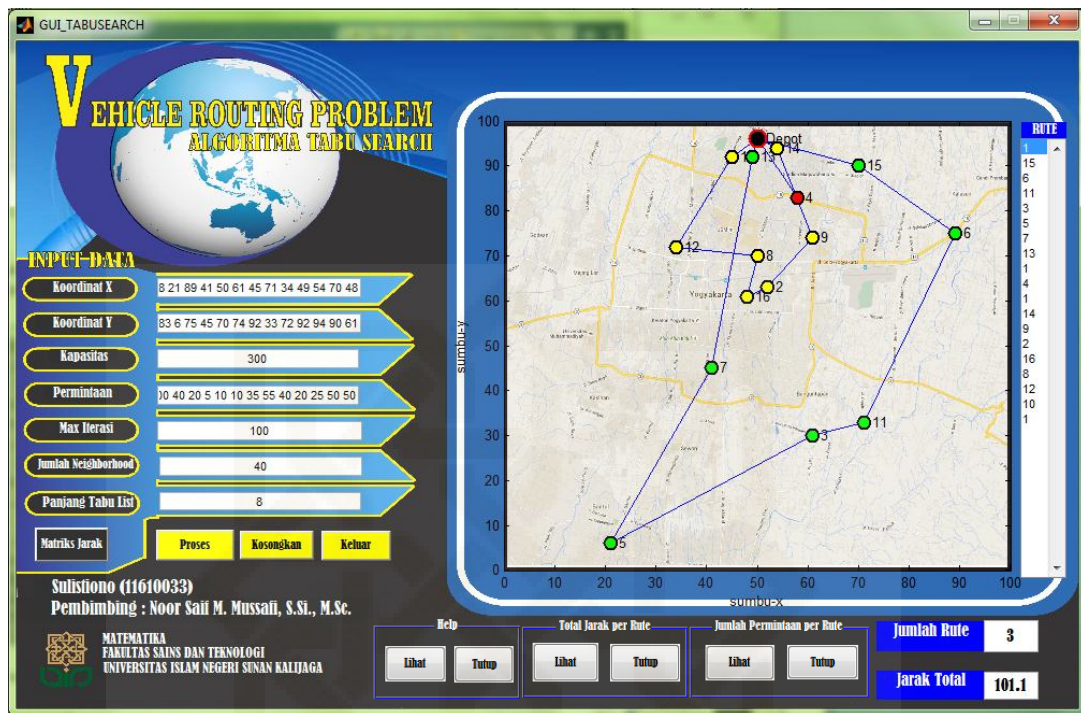
Untuk Input data matriks jarak dapat diisi dengan data matriks jarak antar *node* yang sesuai dengan Table data jarak.



Setelah selesai melakukan proses input data selesai, maka akan diperoleh tampilan sebagai berikut:

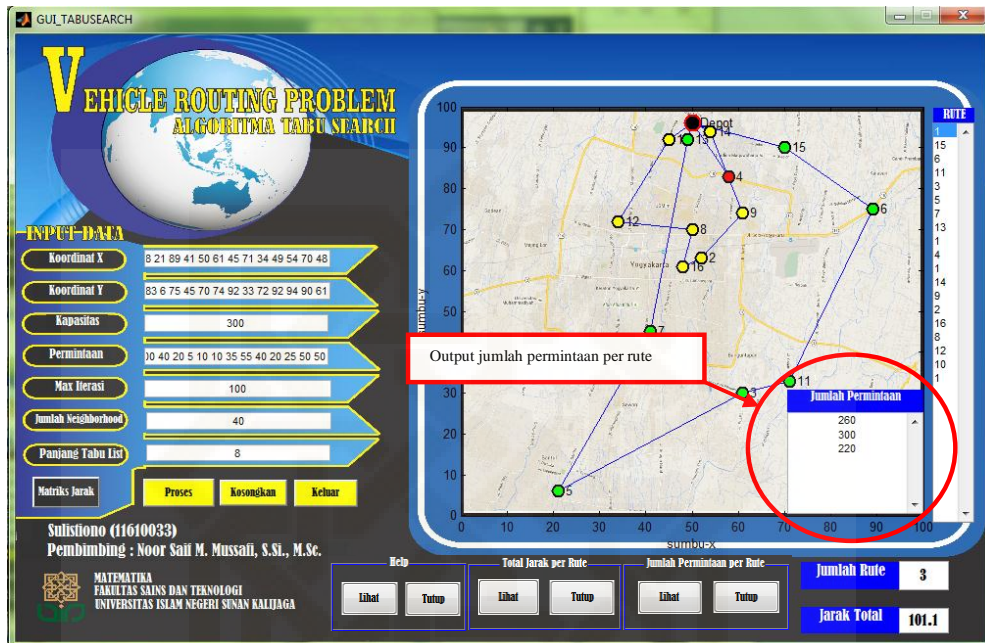


Input data koordinat x, koordinat y, dan matriks jarak harus sesuai dengan jumlah *node* yang diketahui sehingga tidak terjadi error pada saat menjalankan proses perhitungan. Setelah semua data selesai dimasukkan, maka selanjutnya dapat dilihat hasil proses perhitungan Algoritma *Tabu Search* dengan klik pushbutton “Proses”. Berikut adalah tampilan setelah klik tombol pushbutton “Proses”:



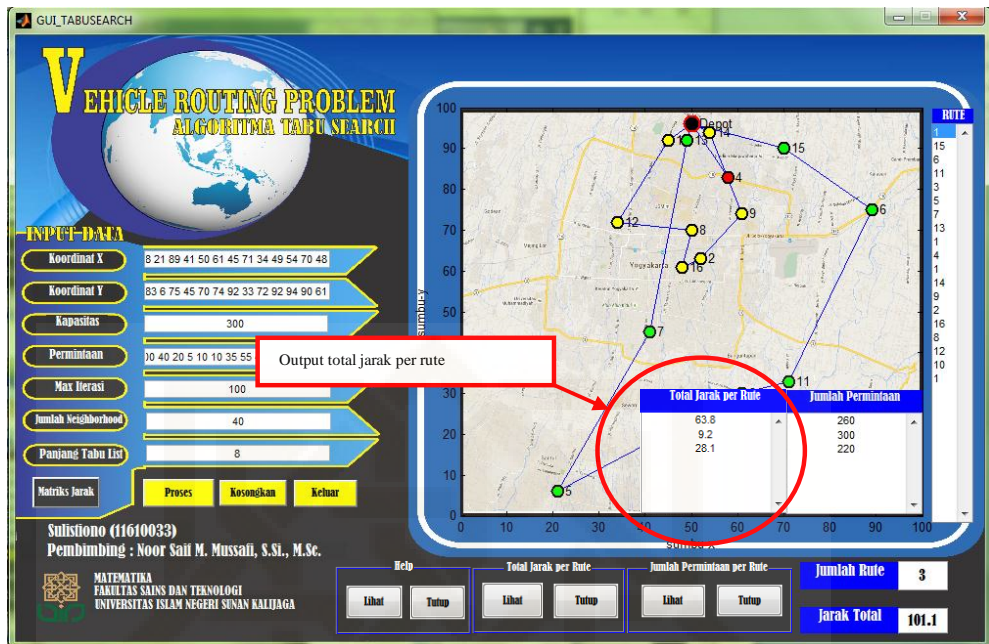
- Output* “Jumlah Rute” adalah 3 yang berarti bahwa rute yang dilalui kendaraan berjumlah 3 rute.
- Output* “Jarak Total” adalah 101,1 yang berarti total jarak tempuh minimum kendaraan pada semua rute adalah 101,1 km.
- Solusi optimal yang diperoleh pada *output* “Rute” adalah 1-15-6-11-3-5-7-13-1-4-1-14-9-2-16-8-12-10-1. Solusi tersebut berarti bahwa kendaraan harus melakukan perjalanan dengan rute sebagai berikut:
 - Rute 1 : 1-15-6-11-3-5-7-13-1
 - Rute 2 : 1-4-1
 - Rute 3 : 1-14-9-2-16-8-12-10-1
- Output* plot pada Gambar mapyogyakarta.jpg adalah representasi graf solusi optimal VRP menggunakan Algoritma *Tabu Search*. Pada output tersebut warna *node* setiap rute berbeda, untuk depot berwarna hitam, pelanggan rute 1 berwarna hijau, pelanggan rute 2 berwarna merah, dan pelanggan rute 3 berwarna kuning. Setiap sisi *node* direpresentasikan dengan garis berwarna biru.

- e) Untuk melihat *output* jumlah permintaan pelanggan per rute dapat dilakukan dengan klik *pushbutton* “Lihat” pada panel “Jumlah Permintaan per Rute” sehingga diperoleh tampilan sebagai berikut:



Pada Gambar tersebut diperoleh jumlah permintaan pelanggan rute 1 sebesar 260 liter, rute 2 sebesar 300 liter, dan rute 3 sebesar 220 liter.

- f) Untuk melihat *output* total jarak kendaraan per rute dapat dilakukan dengan klik *pushbutton* “Lihat” pada panel “Total Jarak per Rute” sehingga diperoleh tampilan sebagai berikut:



Pada tersebut diperoleh total jarak kendaraan pada rute 1 adalah 63,8 km, rute 2 adalah 9,2 km, dan rute 3 adalah 28,1 km. Berdasarkan hasil perhitungan program diperoleh solusi optimal untuk perusahaan tersebut yaitu:

Tabel Solusi Optimal VRP Menggunakan Program

Rute	Solusi VRP	Permintaan	Jarak
1	1-15-6-11-3-5-7-13-1	260 liter	63,8 km
2	1-4-1	300 liter	9,2 km
3	1-14-9-2-16-8-12-10-1	220 liter	28,1 km
Total jarak tempuh kendaraan			101,1 km