



TESIS - TI 185401

VEHICLE ROUTING PROBLEM WITH SPLIT SERVICE, TIME WINDOW AND INTERMEDIATE FACILITY UNTUK MUNICIPAL SOLID WASTE COLLECTION DI KOTA SURABAYA DENGAN ANT COLONY OPTIMIZATION ALGORITHM

DELLA HASSIBA DAYANARA
02411650032007

DOSEN PEMBIMBING
Niniet Indah Arvitrida, S.T., M.T., Ph.D.
Nurhadi Siswanto, S.T., M.S.I.E., Ph.D.

PROGRAM MAGISTER
BIDANG KEAHLIAN MANAJEMEN LOGISTIK DAN RANTAI PASOK
DEPARTEMEN TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019

VEHICLE ROUTING PROBLEM WITH SPLIT SERVICE, TIME WINDOW AND INTERMEDIATE FACILITY UNTUK MUNICIPAL SOLID WASTE COLLECTION DI KOTA SURABAYA DENGAN ANT COLONY OPTIMIZATION ALGORITHM

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

DELLA HASSIBA DAYANARA

NRP. 024116 5003 2007

Tanggal Ujian : 14 Januari 2019

Periode Wisuda : Maret 2019

Disetujui oleh:

1. **Niniet Indah Arvitrida, S.T., M.T., Ph.D.**

NIP: 19840705 200912 2007


(Pembimbing)

2. **Nurhadi Siswanto, S.T, M.T., Ph.D.**

NIP: 19700523 199601 1001


(Ko-Pembimbing)

3. **Erwin Widodo, S.T., M.Eng., Dr.Eng.**

NIP: 19740517 199903 1002


(Penguji 1)

4. **Prof. Iwan Vanany, S.T., M.T., Ph.D.**

NIP: 19710927 199903 1002


(Penguji 2)



Dekan Fakultas Teknologi Industri,

Dr. Bambang Lelono Widjiantoro, S.T., M.T.

NIP: 19690507 199512 1001

LEMBAR PERNYATAAN KEASLIAN TESIS

Saya yang bertanda tangan di bawah ini:

Nama : Della Hassiba Dayanara

NRP : 02411650032007

Program Studi : Magister Teknik Industri – ITS

Menyatakan bahwa tesis dengan judul:

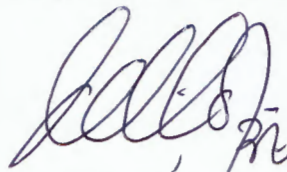
***“VEHICLE ROUTING PROBLEM WITH SPLIT SERVICE, TIME WINDOW
AND INTERMEDIATE FACILITY UNTUK MUNICIPAL SOLID WASTE
COLLECTION DI KOTA SURABAYA DENGAN ANT COLONY
OPTIMIZATION ALGORITHM”***

adalah benar- benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan- bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Seluruh referensi yang dikutip dan dirujuk telah saya tulis secara lengkap di daftar pustaka. Apabila dikemudian hari ternyata pernyataan saya ini tidak benar, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Januari 2019

Yang membuat pernyataan,



Della Hassiba Dayanara

02411650032007

**VEHICLE ROUTING PROBLEM WITH SPLIT SERVICE, TIME WINDOW
AND INTERMEDIATE FACILITY UNTUK MUNICIPAL SOLID WASTE
COLLECTION DI KOTA SURABAYA DENGAN
ANT COLONY OPTIMIZATION ALGORITHM**

Nama : Della Hassiba Dayanara
NRP : 024 1155 003 2007
Dosen Pembimbing : Niniet Indah Arvitrida, S.T., M.T. Ph.D.
Dosen Ko Pembimbing : Nurhadi Siswanto, S.T., M.S.I.E., Ph.D

ABSTRAK

Volume produksi sampah Kota Surabaya dari tahun 2013 sampai dengan 2017 terus meningkat sejalan dengan meningkatnya pertumbuhan penduduk Kota Surabaya. Jumlah kendaraan yang dimiliki oleh Dinas Kebersihan Ruang Terbuka Hijau terbatas sehingga agar dapat memenuhi kebutuhan pengangkutan sampah di setiap LPS yang ada di Surabaya, perlu dilakukan perencanaan yang optimal. Model untuk kasus MSW *collection* Kota Surabaya adalah *Vehicle Routing Problem with Split Service, Time Window and Intermediate Facility* (VRPSSTWIF). Kendaraan melakukan *loading* di setiap LPS dengan permintaan *deterministic* dan melakukan *unloading* pada LPA sebagai *intermediate facilities* dalam model ini. Kendaraan akan berangkat dan kembali ke depot dalam keadaan muatan kosong pada jam operasional atau dalam *time window constraint*. Kendaraan akan melakukan ritase lebih dari satu kali apabila belum melebihi jam operasional. Dalam 1 LPS dapat dilayani lebih dari satu kali dengan kendaraan yang sama ataupun yang berbeda sesuai dengan jumlah *demand* pada LPS tersebut. Dalam pencarian rute optimal yang memiliki nilai total jarak minimal, pendekatan yang digunakan adalah metode metaheuristik dengan algoritma ACO yang diharapkan dapat diselesaikan dengan waktu komputasi yang lebih cepat. Hasilnya terdapat rute rekomendasi baru dengan jumlah jarak total 4256.7 km, selisih 354.4 km dari rute *existing* dengan menggunakan 35 kendaraan. Hasil analisa sensitivitas didapatkan bahwa dengan kendaraan DKRTH saat ini yang berjumlah 44 diperkirakan dapat menangani kenaikan *demand* hingga 20%.

Kata kunci: *vehicle routing problem, municipal solid waste, ant colony optimization*

(halaman ini sengaja dikosongkan)

VEHICLE ROUTING PROBLEM WITH SPLIT SERVICE, TIME WINDOW AND INTERMEDIATE FACILITY FOR MUNICIPAL SOLID WASTE COLLECTION IN SURABAYA CITY WITH ANT COLONY OPTIMIZATION ALGORITHM

Nama : Della Hassiba Dayanara
NRP : 024 1155 003 2007
Dosen Pembimbing : Niniet Indah Arvitrida, S.T., M.T. Ph.D.
Dosen Ko Pembimbing : Nurhadi Siswanto, S.T., M.S.I.E., Ph.D

ABSTRACT

The volume of waste production in the Surabaya City from 2013 to 2017 increase in line with the population growth in the city of Surabaya. The number of vehicles owned by Dinas Kebersihan dan Ruang Terbuka Hijau is limited so that in satisfying the order from every LPS in Surabaya that must be collect by the compactor, the transportation needs optimization planning. The model for the case of MSW collection in Surabaya City is Vehicle Routing Problem with Split Service, Time Window and Intermediate Facility (VRPSSMTTWIF). The vehicles load from every LPS with deterministic demand and unload to LPA as an intermediate facility in this model. The vehicle will depart from the depot and return to the depot in a state of empty load during operating hours or with time window constraint. The vehicle will do the ritase more than once when not exceed the operational hours. In 1 LPS can be served more than once with the same vehicle or different according to the number of demand. In finding the optimal routes that have a minimum total distance value, this research using the approach methodology, metaheuristic with the ACO algorithm which is expected to be solved with faster computation time. The result is a new recommendation route with a total distance of 4256.7 km. The difference between the existing and recommendation route is 354.4 km by using 35 vehicles. The results of the sensitivity analysis show that with the current DKRTH vehicles, the number of 44 units of compactor is estimated to be able to handle the increasing demand until up to 20%.

Kata kunci: *vehicle routing problem, municipal solid waste, ant colony optimization*

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat dan rezekiNya sehingga penulis dapat menyelesaikan penulisan Tesis yang berjudul “*Vehicle Routing Problem With Split Service, Time Window And Intermediate Facility untuk Municipal Solid Waste Collection di Kota Surabaya dengan Ant Colony Optimization Algorithm*” dengan baik dan tepat waktu. Tak lupa juga shalawat serta salam tercurahkan kepada junjungan Nabi Muhammad SAW yang telah menyampaikan petunjuk kepada umatnya.

Laporan tesis ini diajukan sebagai syarat untuk menyelesaikan studi Strata 2 (S2) di Jurusan Teknik Industri. Selama pelaksanaan dan penyusunan tesis ini, penulis telah menerima bantuan dari berbagai pihak. Oleh sebab itu, pada kesempatan ini, penulis tidak lupa mengucapkan terima kasih dan penghargaan kepada:

1. Allah SWT yang telah memberikan kemudahan, kelancaran dan keyakinan kepada penulis, bahwa penulis mampu menyelesaikan Tesis ini;
2. Kedua orang tua penulis yaitu Bapak Ir. H. Burhanudin dan Ibu Hj. Rozana Aliya Hayati, kakak penulis yaitu Isabella Ayu, S.T. dan Moh. Hatta Fahamsyah, M.Sc, serta adik penulis yaitu Hudan Akbar , yang senantiasa mendoakan dan memotivasi penulis;
3. Ibu Niniet Indah Arvitrida, S.T., M.T. Ph.D. dan Bapak Nurhadi Siswanto, S.T., M.S.I.E., Ph.D selaku dosen pembimbing penulis yang telah memberikan pengarahan, bimbingan serta dukungan penuh selama penulis menyelesaikan Tesis ini;
4. Bapak Erwin Widodo, S.T., M. Eng., Dr. Eng dan Bapak Prof. Iwan Vanany, S.T., M.T., Ph.D. selaku penguji Seminar Proposal dan Seminar Hasil penulis untuk penelitian ini yang telah memberikan saran-saran perbaikan serta bimbingan yang mendukung penelitian ini;
5. Dinas Kebersihan dan Ruang Terbuka Hijau yang telah memberikan kesempatan penulis untuk melakukan penelitian ini dan memberikan bimbingan serta arahan yang dibutuhkan oleh penulis;

6. Calon pendamping penulis yang telah senantiasa memberikan semangat dan dukungan penuh demi terselesaikannya penelitian ini dengan baik.
7. Teman – teman Pasca Sarjana Teknik Industri Program Magister dan Doktoral Teknik Industri ITS, yang telah memberikan bantuan dan memotivasi penulis dalam menyelesaikan Tesis ini.

Dalam penulisan Tesis ini, penulis merasa masih banyak kekurangan pada teknis penulisan dan materi laporan. Untuk itu, kritik dan saran dari semua pihak sangat diharapkan demi penyempurnaan pembuatan Tesis ini. Penulis berharap semoga Tesis ini dapat bermanfaat bagi objek amatan dan rekan – rekan di Teknik Industri ITS pada khususnya.

Surabaya, Januari 2019

Penulis

DAFTAR ISI

Halaman cover	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN TESIS	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	9
1.3 Tujuan Penelitian	10
1.4 Manfaat Penelitian	10
1.5 Ruang Lingkup Penelitian	11
1.5.1 Batasan Penelitian	11
1.5.2 Asumsi Penelitian	11
1.6 Sistematika Penulisan	12
BAB II TINJAUAN PUSTAKA	15
2.1 <i>Municipal Solid Waste</i>	15
2.1.1 <i>Municipal Solid Waste Collection</i>	16
2.1.2 Metode Pengangkutan Sampah	18
2.2 <i>Vehicle Routing Problem</i>	20

2.2.1	<i>Vehicle Routing Problem with Time Window</i>	21
2.2.2	<i>Vehicle Routing Problem with Intermediate Facility</i>	22
2.2.3	<i>Vehicle Routing Problem with Split Service</i>	23
2.3	VRP untuk MSW <i>collection</i>	24
2.4	Optimasi	26
2.4.1	Metaheuristik	26
2.4.2	<i>Ant Colony Optimization</i>	28
2.5	Penelitian Terdahulu	32
BAB III METODOLOGI PENELITIAN		45
3.1	Diagram Alir Penelitian	45
3.2	Tahap Pendahuluan	45
3.3	Tahap Pengumpulan Data	45
3.4	Tahap pengembangan Model	47
3.4.1	Pengembangan Model Matematis.....	47
3.4.2	Pengembangan Model dengan Algoritma <i>Ant Colony Optimization</i> .	48
3.4.3	Verifikasi dan Validasi Model.....	50
3.5	Tahap Pengerjaan Model.....	50
3.6	Tahap Pembahasan dan Analisis.....	50
3.7	Tahap Kesimpulan Saran	51
BAB IV PENGUMPULAN DATA		53
4.1	<i>Municipal Solid Waste Collection</i> di Kota Surabaya.....	53
4.2	Pengumpulan Data	54
4.2.1	Data Kendaraan Truk <i>Compactor</i>	54
4.2.2	Data Lokasi <i>Node</i> yang Dkunjungi Truk <i>Compactor</i>	55
4.2.3	Data <i>Demand LPS</i>	56
4.2.4	Data Jarak Tempuh Antar Lokasi	57

4.2.5	Data Waktu Tempuh Antar Lokasi	58
4.2.6	Data Rute dan Penugasan Kendaraan Truk <i>Compactor Existing</i>	59
BAB V PENGEMBANGAN MODEL.....		61
5.1	Pengembangan Model Konseptual VRPSSTWIF untuk MSW <i>Collection</i> di Kota Surabaya.....	61
5.2	Pengembangan Model Matematis VRPSSTWIF untuk MSW <i>Collection</i> di Kota Surabaya.....	63
5.2.1	Skenario Pengembangan Model Matematis	63
5.2.2	Notasi Model Matematis VRPSSTWIF	64
5.2.3	Model Matematis VRPSSTWIF.....	65
5.3	Pengembangan Algoritma ACO model VRPSSTWIF untuk MSW <i>Collection</i> di Kota Surabaya.....	68
5.3.1	Parameter ACO untuk VRPSSTWIF	68
5.3.2	Algoritma ACO untuk VRPSSTWIF	70
5.4	Verifikasi Validasi Model dan Algoritma	73
5.4.1	Verifikasi dan Validasi Model	75
5.4.2	Verifikasi dan Validasi Algoritma	81
BAB VI HASIL DAN ANALISIS.....		85
6.1	Percobaan Model Algoritma.....	85
6.1.1	Parameter Model Algoritma untuk MSW <i>Collection</i> di Kota Surabaya.....	85
6.1.2	Hasil <i>Running</i> Model Algoritma untuk MSW <i>Collection</i> di Kota Surabaya.....	89
6.2	Perbandingan Rute <i>Existing</i> dan Rute Rekomendasi	91
6.3	Penugasan Truk <i>Compactor</i> untuk MSW <i>Collection</i> di Kota Surabaya	92
6.4	Analisa Sensitivitas.....	94
6.5	Analisis Hasil Penelitian.....	95

BAB VII KESIMPULAN DAN SARAN	97
7.1 Kesimpulan	97
7.2 Saran.....	98
DAFTAR PUSTAKA.....	99
LAMPIRAN 1	103
LAMPIRAN 2	107
LAMPIRAN 3	108
LAMPIRAN 4	109
LAMPIRAN 5	112
LAMPIRAN 6	114
LAMPIRAN 7	117
LAMPIRAN 8	119
LAMPIRAN 9	121
LAMPIRAN 10	124
LAMPIRAN 11	125

DAFTAR GAMBAR

Gambar 1.1 TPA Benowo	6
Gambar 1.2 Proses <i>Loading</i> Kontainer pada Truk <i>Compactor</i>	6
Gambar 1.3 <i>Stationary Container System</i> Mekanis	6
Gambar 2.1 Ilustrasi Kedatangan Berdasarkan <i>Time Window</i>	22
Gambar 2.2 VRP <i>with Intermediate Facility</i>	23
Gambar 2.3 4 <i>Split Cycle</i>	24
Gambar 2.4 Klasifikasi Metode Optimasi.....	27
Gambar 2.5 Optimasi dengan Metaheuristik	27
Gambar 2.6 Metode Metaheuristik	28
Gambar 2.7 Ilustrasi Perjalanan Semut Pada ACO.....	29
Gambar 3.1 Diagram Alir Penelitian	46
Gambar 3.2 Algoritma ACO	49
Gambar 4.1 Contoh Pengambilan Data Melalui Google Maps.....	58
Gambar 5.1 Model Konseptual MSW <i>Collection</i> di Kota Surabaya	62
Gambar 5.2 Tahapan Algoritma ACO untuk Model VRPSSTWIF.....	69
Gambar 5.3 CVRP <i>exact 10 node</i>	74
Gambar 5.4 CVRP ACO <i>10 node</i>	75
Gambar 5.5 TSP <i>Exact</i> dengan Lingo.....	757
Gambar 5.6 TSP ACO dengan Matlab.....	69
Gambar 5.7 Hasil <i>Running Clustering</i> TSP <i>Exact</i> dengan Matlab	82
Gambar 5.4 Hasil <i>Running Clustering</i> TSP ACO dengan Matlab.....	753

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 1.1 Proyeksi Produksi Sampah Tahun 2025 Berdasarkan Wilayah.....	2
Tabel 1.2 Volume Produksi Sampah Surabaya 2013 – 2017.....	5
Tabel 2.1 Sistem Pengangkutan Sampah	18
Tabel 2.2 Posisi Model Penelitian.....	36
Tabel 2.3 Posisi Metode Penelitian	37
Tabel 2.4 Penelitian Sebelumnya	38
Tabel 4.1 Data Kendaraan Truk <i>Compactor</i>	54
Tabel 4.2 Data Lokasi <i>Node</i> Kunjungan	55
Tabel 4.3 Data <i>Demand</i> LPS.....	57
Tabel 4.4 Data Rute dan Penugasan Kendaraan Truk <i>Compactor Existing</i>	59
Tabel 4.5 Data Kendaraan Truk <i>Compactor Existing</i> yang Melebihi Waktu Akhir Pelayanan Depot.....	59
Tabel 5.1 Simbol Model Konseptual	61
Tabel 5.2 Perbandingan <i>Exact</i> dan ACO CVRP	75
Tabel 5.3 Jarak Tempuh Sampel Kecil	76
Tabel 5.4 Waktu Tempuh Sampel Kecil	77
Tabel 5.5 <i>Demand</i> Sampel Kecil	77
Tabel 5.6 Sub Rute 1 Validasi dan Verifikasi Model	78
Tabel 5.7 Hasil <i>Clustering</i> Manual Metode <i>Exact</i> TSP.....	78
Tabel 5.8 Evaluasi <i>Clustering</i> Manual Rute <i>Exact</i>	79
Tabel 5.9 Hasil <i>Clustering</i> Manual Metode ACO TSP.....	80
Tabel 5.10 Evaluasi <i>Clustering</i> Manual Rute ACO	81
Tabel 6.1 Uji Coba Parameter	86
Tabel 6.2 Parameter terhadap Waktu Komputasi	88
Tabel 6.3 Nilai Minimum Hasil Uji Coba.....	88
Tabel 6.4 Hasil <i>Running</i> 10 Replikasi Model Algoritma.....	89
Tabel 6.5 <i>Gantt Chart Cycle Time</i> Rute <i>Existing</i>	91
Tabel 6.6 Penugasan Kendaraan Rekomendasi	93
Tabel 6.7 Analisa Sensitivitas terhadap Perubahan <i>Demand</i>	95

(Halaman ini sengaja dikosongkan)

. BAB I

PENDAHULUAN

Pada bab ini akan menjelaskan pendahuluan yang berisikan latar belakang penelitian, perumusan masalah dalam penelitian, tujuan penelitian, manfaat penelitian, dan sistematika penelitian yang digunakan dalam penelitian ini.

1.1 Latar Belakang

Sampah adalah sesuatu yang tidak digunakan, dianggap sudah tidak memiliki manfaat lagi dan harus dikelola dengan baik sehingga tidak membahayakan lingkungan. Menurut Ramdhani, *et al.* (2016) jumlah volume sampah sebanding dengan tingkat konsumsi manusia terhadap barang atau material yang digunakan sehari-hari, sehingga pengelolaan sampah tidak terlepas dari pengelolaan gaya hidup masyarakat, pengangkutan sampah yang tidak responsif terhadap pertumbuhan jumlah timbunan di masing-masing lahan pembuangan akan menimbulkan potensi bahaya pencemaran lingkungan.

Municipal Solid Waste (MSW) adalah sampah padat masyarakat. MSW umumnya dikenal sebagai limbah atau sampah yang terdiri dari barang-barang sehari-hari yang sudah tidak digunakan dan kemudian dibuang, seperti kemasan produk, klipng rumput, perabotan, pakaian, botol, sisa makanan, koran, peralatan, cat, dan baterai. Bahan-bahan ini berasal dari rumah, sekolah, rumah sakit, dan industri (Sipra, *et al.*, 2018).

Sejalan dengan perkembangan suatu kota, jumlah sampah perkotaan menjadi sesuatu yang penting untuk diperhatikan dalam gaya hidup perkotaan, MSW bahkan lebih cepat pertumbuhannya dibandingkan dengan *rate of urbanization*. Sepuluh tahun yang lalu ada 2,9 miliar penduduk perkotaan yang menghasilkan sekitar 0,64 kg MSW per orang per hari (0,68 miliar ton per tahun) dan diperkirakan pada tahun 2012 jumlah sebelumnya telah meningkat menjadi sekitar 3 miliar penduduk yang menghasilkan 1,2 kg per orang per hari (1,3 miliar ton per tahun). Pada tahun 2025, kemungkinan akan meningkat menjadi 4,3 miliar penduduk perkotaan yang menghasilkan sekitar 1,42 kg / kapita / hari sampah kota

Tabel 1.1 Proyeksi Produksi Sampah Tahun 2025 Berdasarkan Wilayah
(Sumber: Hoornweg & Tata, 2012)

Regional	Data yang Tersedia			Proyeksi Tahun 2025			
	Total Populasi Urban (Juta)	Urban Waste Generation		Proyeksi Populasi		Proyeksi Urban Waste	
		Per Kapita (Kg/Kapita/Hari)	Total (Tons/Hari)	Total Populasi (juta)	Populasi Urban	Per Kapita (Kg/Kapita / Hari)	Total (Tons/Hari)
AFR	260	0.65	169119	1152	518	0.85	441840
EAP	777	0.95	738958	2124	1229	1.5	1865379
ECA	227	1.1	254389	339	239	1.1	354810
LCR	399	1.1	437545	681	466	1.6	728392
MENA	162	1.1	173545	379	257	1.43	369320
OECD	729	2.2	1566286	1031	842	2.1	1742417
SAR	426	0.45	19241	1938	734	0.77	567545
Total	2980	1.2	3359083	7644	4285	1.4	6069703

(2,2 miliar ton per tahun). Berdasarkan data yang ada pada Tabel 1.1, Pembuang sampah tahunan di Asia Timur dan Wilayah Pasifik dimana Indonesia termasuk didalamnya adalah sekitar 270 juta ton per tahun. Produksi limbah per kapita berkisar antara 0,44 hingga 4,3 kg per orang per hari untuk wilayah tersebut, dengan rata-rata 0,95 kg / kapita / hari (Hoornweg & Tata, 2012).

MSW *managemant* adalah aktivitas multidisiplin yang mencakup beberapa aktivitas seperti pemisahan sumber, penyimpanan, pengumpulan, transfer dan transportasi, pengelolaan dan pemulihan, dan pembuangan (Das & Bhattacharyya, 2015). Sampah merupakan konsekuensi kehidupan yang sering menimbulkan masalah dan jumlahnya akan semakin meningkat seiring dengan peningkatan jumlah penduduk dan beragam aktivitasnya. Sehingga banyak penelitian-penelitian yang berfokus pada isu tentang penanganan sampah untuk pencarian solusi praktis dan taktis yang mampu mengurangi masalah yang bisa ditimbulkan oleh tata kelola sampah yang tidak baik. Salah satu yang menjadi topik yang sering dibahas adalah MSW *collection* yaitu bagaimana cara pengumpulan sampah disuatu daerah dengan beberapa titik pengumpulan sementara yang kemudian dijadikan satu dan diantarkan ketempat pengelolaan sampah akhir.

Banyak penelitian terkait MSW *collection* di dalam maupun luar negeri. Beberapa penelitian yang dilakukan di Indonesia adalah dari Fitria, *et al.* (2009)

yang melakukan penelitian terkait *MSW collection* di Kota Bandung. Tujuan penelitiannya adalah untuk menentukan rute pengumpulan dan pengangkutan sampah di Kota Bandung. Rute yang dihasilkan diusahakan agar memiliki jumlah kendaraan minimum sebagai tujuan utamanya dan total waktu penyelesaian minimum sebagai tujuan berikutnya. Model yang digunakan adalah *vehicle routing problem multiple trip dan intermediate facility* (VRPMTIF). *Intermediate facility* sebagai titik kunjungan terakhir sebelum kembali ke depot, dimana setiap kendaraan harus mengunjungi TPA pada akhir rute. Serta Ramdhani, *et al.* (2016) yang melakukan penelitian di Kota Surabaya. Tujuan penelitiannya adalah untuk menentukan berapa jumlah kendaraan yang dibutuhkan dan kapan jadwal keberangkatan pengangkutan sampah dengan kendaraan minimum untuk semua rute kendaraan pada kasus pengambilan dan pengumpulan sampah Dinas Kebersihan dan Ruang Terbuka Hijau (DKRTH). Model permasalahan *vehicle assignment problem* dikembangkan dengan *integer linear programming* dengan pendekatan *assignment problem*.

Di luar Indonesia penelitian yang serupa juga dilakukan oleh Buhrkal, *et al.* (2012) yang mengemukakan bagaimana cara mengumpulkan sampah secara efisien. Mengangkat 2 kasus nyata pada perusahaan yang menangani *MCW collection* di Danish Company, Nort America dan di Greater Copenhagen, Denmark dengan kebutuhan yang berbeda terutama terkait kebijakan waktu istirahat dan makan siang. Keduanya menggunakan model matematis *Waste Collection Vehicle Routing Problem with Time Windows* (WCVRPTW). Fokus penelitian adalah untuk menemukan biaya dengan rute yang optimal untuk truk sampah sehingga semua sampah dapat dikosongkan diangkut menuju lokasi pembuangan dengan memenuhi waktu pelayanan masing-masing *customer* dan mempertimbangkan durasi waktu berhenti kendaraan untuk beristirahat diantara *arc* (i j) dalam satu kali operasional truk sampah.

Pada penelitian Das & Bhattacharyya (2015) *MSW collection* dan rute transportasi di Kolkota India memiliki tujuan untuk mengoptimalkan pengumpulan sampah kota dan skema transportasi yang fokus pada permasalahan meminimasi jarak pada masing-masing pengumpulan sampah dan rute transportasi dengan menggunakan metode heuristik untuk menentukan langkah optimal. Langkah

pengoptimalan dibagi menjadi beberapa langkah dengan karakteristik pengumpulan sampah yang berbeda untuk setiap langkahnya. Model pendekatan yang digunakan adalah *covering salesman problem* yang disesuaikan pada setiap langkahnya.

Dalam penelitian di Kota Danang, Vietnam oleh Son & Louati (2016), MSW *collection* berhubungan dengan pembangunan model *vehicle routing*. Penelitian ini bertujuan untuk memaksimalkan jumlah sampah yang dikumpulkan dari *multiple transfer station, gather sites* dan dengan menggunakan beberapa jenis kendaraan yang berbeda dalam *time window* serta meminimalkan emisi lingkungan yang ditimbulkan oleh kendaraan. Model yang digunakan merupakan pengembangan dari *vehicle routing* untuk MSW dengan membuka beberapa keterbatasan sebelumnya. Kemudian untuk membandingkan rute sebelum dan skenario rute yang telah dibuat digunakan metode *Multicriteria Decision Analysis* (MCDA) dengan menggunakan *Analytical Hierarchy Process* (AHP) (Son & Louati, 2016).

Kota Surabaya merupakan salah satu kota Maju di Indonesia dengan jumlah penduduk terbesar di wilayah Jawa Timur. Selanjutnya jumlah penduduk tersebut akan mempengaruhi tingkat konsumsi masyarakat yang memicu jumlah sampah meningkat. Berdasarkan sensus tahun 2010 dan 2015 oleh Badan Pusat Statistik Provinsi Jawa Timur, jumlah penduduk Kota Surabaya laju pertumbuhannya adalah 0.55%. Angka laju pertumbuhan yang cukup besar maka akan berdampak tingginya laju pertumbuhan produksi sampah kota. Pada Tabel 1.2 ditampilkan volume produksi sampah Surabaya dari tahun 2013 sampai dengan 2017. Penduduk yang terus bertambah juga menimbulkan volume produksi sampah yang terus bertambah juga. Tidak semua sampah perharinya 100% dapat diangkut ke Lahan Pembuangan Akhir (LPA) karena besarnya volume sampah dan terus bertambah dengan keterbatasan kendaraan angkut yang melayani. Oleh karena itu, perlu adanya manajemen yang baik untuk mengangkut sampah kota menuju LPA.

Volume sampah Kota Surabaya yang selalu bertambah maka, menjadikan perlu adanya sistem yang baik untuk mengatur termasuk rute dan penjadwalan pengambilan sampah pada setiap Lahan Pembuangan Sementara (LPS) untuk dibawa ke LPA Benowo. Kondisi TPA Benowo dapat dilihat pada Gambar 1.1.

Tabel 1.2 Volume Produksi Sampah Surabaya 2013 – 2017
(Sumber: Dinas Ruang Terbuka Hijau, 2017)

Uraian	2013	2014	2015	2016	2017
Jumlah Penduduk (Jiwa)	3,200,460	2,853,661	2,943,528	3,307,300	3,316,131
Volume Produksi Sampah (m3/hari)	9,601.38	9,185.94	9,475.21	9,710.61	10,674.63
Jumlah sampah yang ditangani di TPA (m3/hari)	4,647.33	4,852	4,926	5,237.7	5,539.15

Salah satu kendaraan yang digunakan sebagai alat angkut sampah kota adalah truk *compactor*. Truk *compactor* merupakan kendaraan terbaru dari Dinas Ruang Terbuka Hijau (DKRTH) yang paling efisien karena dengan perlengkapan mesin *compactor* dapat membawa sampah dengan maksimal disetiap kendarannya. Truk *compactor* melayani 81 LPS/Depo yang memiliki permintaan pengangkutan sampah yang bersifat *deterministic*, proses pengangkutannya dapat dilihat pada Gambar 1.2.

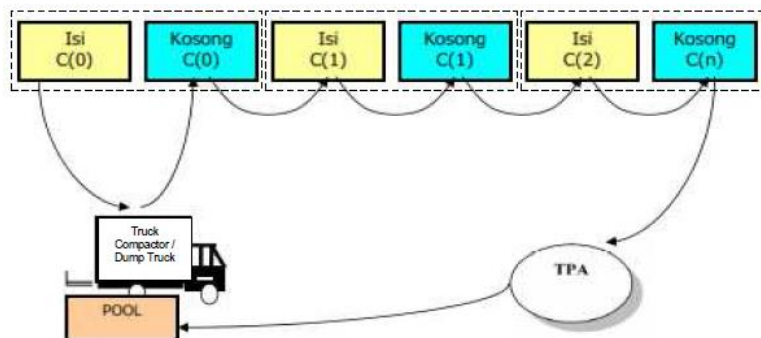
Pola pengangkutan sampah Kota Surabaya oleh kendaraan truk *compactor* berdasarkan Peraturan Menteri Pekerjaan Umum tentang Penyelenggaraan Prasarana dan Sarana Persampahan dalam Penanganan Sampah Rumah Tangga dan Sampah Sejenis, menjelaskan bahwa pola pengangkutannya adalah tidak langsung menggunakan *stationary container system*. Sistem ini menggunakan kendaraan dengan sistem pengumpulan sampah wadah atau kontainer tidak dibawa berpindah-pindah. Ada 2 cara untuk *stationary container system* yaitu dengan cara mekanis dan manual. Pada kegiatan operasional pengangkutan sampah Kota Surabaya dengan truk *compactor* menggunakan *stationary container system* mekanis terlihat pada Gambar 1.3, dimana menggunakan bantuan mesin pres elektrik untuk memasukkan sampah pada kontainer kendaraan. Setiap *container* yang dilakukan proses *loading* memiliki waktu per unit per kontainer.



Gambar 1.1 TPA Benowo



Gambar 1.2 Proses *Loading* Kontainer pada Truk *Compactor*



Gambar 1.3 *Stationary Container System* Mekanis
(Sumber: Peraturan Menteri Pekerjaan Umum Nomor 3 Tahun 2013)

DKRTH pada saat ini sudah mempunyai rute *existing* kendaraan pada semua truk *compactor* yang dimiliki untuk melayani permintaan LPS. DKRTH menggunakan satuan ritase untuk 1 kali periode dalam pengambilan sampah disuatu atau sejumlah LPS dengan kapasitas kendaraan dan diantarkan ke LPA. Rute *existing* dimulai dari kendaraan berangkat rata-rata mulai pukul 03.00-05.00 WIB dari depot, kemudian dapat melayani 1 atau lebih ritase dan kembali ke depot pukul 15.00 WIB. Sebelum kembali ke depot kendaraan akan mengisi BBM di Pom Bensin yang sama. Lokasi Pom bensin berada diantara Lokasi LPA dan depot. Tantangan yang ada adalah keterbatasannya kendaraan truk *compactor* yang

berjumlah 44 kendaraan harus melayani 80 LPS/Depo dengan jumlah permintaan yang berbeda-beda. Selain itu, LPA Benowo terletak di pinggir Kota Surabaya sehingga jarak yang ditempuh untuk kendaraan cukup jauh terlebih bagi LPS-LPS yang letaknya ada di sisi pinggir kota. DKRTH belum memiliki sistem yang menunjang untuk pengambilan keputusan terkait rute yang harus dilakukan dalam pengambilan sampah oleh truk *compactor* yang dipengaruhi perubahan kondisi di lapangan untuk mendapatkan solusi optimal berdasarkan perubahan tersebut. Oleh karena itu, perlu adanya suatu metode dalam penentuan jumlah kendaraan dan rute perjalanan agar secara optimal dapat memenuhi kebutuhan dengan minimal jarak yang ditempuh dalam beberapa aturan yang sudah ditentukan.

Adanya korelasi antara permasalahan transportasi kendaraan dengan MSW *collection* telah menjadi isu terkini di beberapa negara khususnya kota-kota besar. Pada penelitian MSW *collection* Kota Surabaya VRP yang ada harus disesuaikan dengan beberapa hal, yaitu: kapasitas kendaraan (*capacited*), jam operasional kerja (*time window*) dan adanya LPA sebagai *intermediate facility* yang harus dikunjungi sebelum kendaraan kembali ke depot. Jumlah permintaan pada beberapa LPS lebih tinggi dari pada kapasitas kendaraan sehingga, 1 LPS dapat dikunjungi lebih dari 1 kali dengan atau tidak menggunakan kendaraan yang sama (*split service*). Pada LPS dilakukan proses *loading* kontainer kedalam truk *compactor* sedangkan pada LPA dilakukan proses *unloading* muatan. Model untuk kasus MSW *collection* Kota Surabaya adalah *Vehicle Routing Problem with Split service, Time Window, and Intermediate Facility* (VRPSSTWIF).

Dalam pencarian nilai optimal untuk model VRP MSW *collection* dapat didekati dengan beberapa metode. Ramdhani, *et al.* (2016) yang melakukan penelitian di Kota Surabaya menggunakan model permasalahan *vehicle assignment problem* yang dikembangkan dengan *integer linear programming* dan menggunakan *software* Lingo untuk penyelesaiannya. Hasil yang didapatkan belum optimal karena lamanya waktu komputasi sehingga dalam sekian waktu *running* model dihentikan ketika belum mencapai nilai optimal. Markov, *et al.* (2015) membandingkan model *waste collection VRP with intermediate facilities* dengan metode *Mix Integer Linear Programming* (MILP) dengan heuristik *multiple neighborhood search*. Dalam kondisi *small size sample* keduanya bisa digunakan,

hasilnya dengan metode heuristik dapat mendapatkan solusi yang lebih optimal. Penelitian Das & Bhattacharyya (2015) terkait MSW *collection* pada awalnya didekati dengan MILP namun penyelesaiannya juga menggunakan metode heuristik dalam pencarian nilai optimal sampah yang dikumpulkan. Hasilnya dapat mengurangi 30% total jarak yang ditempuh dalam pengumpulan sampah. Benjamin & Beasley (2010) melakukan penelitian terkait VRP untuk pengumpulan sampah dengan adanya *time windows*, periode waktu istirahat supir dan beberapa fasilitas pembuangan. Membandingkan beberapa metode metaheuristik yaitu *Tabu Search* (TS), *Variable Neighbourhood Search* (VNS) dan kombinasi antara keduanya, *Variable Neighbourhood Tabu Search* (VNNTS). Hasil yang didapatkan VNS adalah algoritma yang memberikan nilai paling efektif karena memiliki waktu komputasi tercepat. Tujuan penelitian oleh Karadimas, *et al.* (2007) adalah untuk mengidentifikasi skenario yang paling hemat biaya untuk MSW *collection* Athena dengan memperkirakan biaya operasionalnya dan mensimulasikan aplikasinya. Algoritma yang diusulkan adalah *Hybridized Ant Colony* untuk menyelesaikan model VRP tersebut.

VRP dan VRPTW tergolong dalam permasalahan NP-hard combinatorial. Menggunakan metode *exact* dengan permasalahan tersebut akan menjadi sulit terselesaikan, membutuhkan waktu yang lama dalam penentuan solusi yang layak, apalagi untuk solusi yang optimal bahkan untuk contoh yang cukup kecil. Metode heuristik adalah metode pencarian solusi yang lebih baik dari pada *exact* dengan biaya komputasi yang wajar namun tidak dapat menjamin akan ditemukannya solusi optimal tetapi solusi yang mendekati optimal dan layak. Seringkali heuristik hanya dapat digunakan masalah spesifik, sehingga metode yang bekerja untuk satu masalah tidak dapat digunakan untuk memecahkan masalah yang berbeda. Metode metaheuristik dalam pengambilan keputusan dapat memberikan solusi berkualitas tinggi untuk aplikasi penting dalam berbagai macam masalah dan dalam ukuran yang besar. Metaheuristik dapat menyelesaikan beberapa masalah kombinatorial yang sulit misalnya VRPTW (El-Sherbeny, 2010).

Banyak pendekatan yang digunakan dalam metode metaheuristik, baik *single based solution metaheuristic* atau *population based metaheuristic*. Keduanya dapat memberikan solusi bagi permasalahan kombinatorial NP - hard.

Namun, *Population based metaheuristic* relatif dapat memberikan waktu komputasi lebih cepat dibandingkan *single based solution metaheuristic* karena pencarian solusi langsung berdasarkan populasi tertentu. Dalam kategori *swarm intelligence*, ACO yang pertama kali dirancang untuk kasus TSP dan untuk beberapa kombinasi untuk VRP ACO mampu untuk menyelesaikannya.

Penentuan keputusan semut dalam perjalanan pencarian makanan adalah sebuah *artificial intelligence algorithm* yang baik untuk kasus VRP. Pendekatan ini relevan karena menghasilkan solusi dalam permasalahan penting didalam transportasi dan hasil yang dihasilkan dari beberapa eksperimen menunjukkan teknik ini kompetitif terhadap beberapa Teknik lainnya dalam pencarian solusi untuk VRP (Bell & McMullen, 2004). Berdasarkan hasil dari studi kelayakan sederhana oleh Reed, *et al.* (2014) cukup menunjukkan keefektifan sistem ACO dalam menentukan rute optimal dengan berbagai pertimbangan didalamnya. Pada kasus ini terdapat *node* sebagai tempat untuk *unload* sampah sebelum kembali ke depot. ACO kompetitif dengan algoritma metaheuristik lain untuk masalah CVRP dengan untuk masalah yang lebih rumit maka jumlah iterasi perlu ditambah.

Penelitian ini berfokus pada pengembangan model VRPSSMTTWIF untuk MSW *collection* truk *compactor* di Kota Surabaya. Kendaraan melakukan *loading* disetiap LPS dengan permintaan *deterministic* dan melakukan *unloading* pada LPA sebagai *intermediate facilities* dalam model ini. Kendaraan akan berangkat dan kembali ke depot dalam keadaan muatan kosong. Dalam pencarian rute optimal yang memiliki nilai jarak minimal, pendekatan yang digunakan adalah metode metaheuristik dengan algoritma ACO dengan mempertimbangkan banyaknya LPS yang harus dikunjungi serta adanya LPA yang harus dikunjungi sebelum kembali ke depot. Algoritma ACO, diharapkan dapat membantu pencarian nilai optimal dengan baik dengan waktu komputasi yang lebih cepat.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, terdapat permasalahan yang penting untuk dibahas yaitu: bagaimana mengembangkan model VRPSSTWIF untuk MSW *collection* di Kota Surabaya dengan pendekatan algoritma ACO sehingga didapatkan rute kendaraan dengan jarak optimal untuk

DKRTH Kota Surabaya. Selain itu, karena pertumbuhan produksi sampah masyarakat selalu meningkat maka pada penelitian ini juga perlu melakukan analisa sensitivitas model dalam menghadapi perubahan jumlah permintaan.

1.3 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai adalah:

1. Menghasilkan model yang dapat membantu dalam keputusan yang terkait dengan:
 - a. Rute kendaraan truk *compactor* dalam *MSW collection* di Kota Surabaya.
 - b. Jumlah minimum kendaraan yang dibutuhkan untuk dapat melayani semua *demand* LPS dalam satu kali operasional harian.
 - c. Jadwal keberangkatan kendaraan dalam *MSW collection* di Kota Surabaya.

Model meliputi model konseptual, model matematis dan model algoritma *Ant Colony Optimization* (ACO).

2. Membandingkan jarak total rute *existing* dengan rute yang direkomendasikan.
3. Melakukan penugasan kendaraan dari beberapa pertimbangan yang dihasilkan dari tahapan-tahapan sebelumnya.
4. Melakukan analisa sensitivitas model yang dipengaruhi jumlah *demand* yang tersedia di LPS terhadap total jarak tempuh, jumlah kendaraan dan waktu komputasi untuk pemenuhan permintaan pengangkutan sampah.

1.4 Manfaat Penelitian

Manfaat yang akan diperoleh dari dilakukannya penelitian ini adalah:

1. Memberikan kontribusi terhadap pengembangan model VRP yang disesuaikan untuk kasus *MSW collection* di Kota Surabaya.
2. Memberikan gambaran kepada DKRTH Kota Surabaya mengenai rute optimal dengan jarak minimum dan dengan jumlah kendaraan minimal dalam melayani semua permintaan pengangkutan sampah dalam waktu operasional harian kendaraan.

3. Memberikan gambaran mengenai kendaraan yang dibutuhkan yang terdapat di depot sehingga Dinas Kebersihan dan Ruang Terbuka Hijau Kota Surabaya dapat melakukan pembaharuan kendaraan.

1.5 Ruang Lingkup Penelitian

Ruang lingkup penelitian ini akan dibahas mengenai batasan dan asumsi yang digunakan pada penelitian ini. Tujuan dari ditentukannya ruang lingkup ini adalah untuk menjaga agar penelitian ini memiliki *boundary* yang jelas.

1.5.1 Batasan Penelitian

Batasan yang digunakan dalam penelitian ini adalah:

1. Penelitian dilakukan di DKRTH Kota Surabaya dengan amatan satu jenis kendaraan dinas yaitu truk *compactor*.
2. Daerah fokus penelitian adalah Kota Surabaya dengan wilayah yang dilayani oleh kendaraan *compactor*.
3. Waktu akhir pelayanan kendaraan pada Dinas Kebersihan dan Ruang Terbuka Hijau Kota Surabaya yaitu pada pukul 15.00.
4. Data kondisi jalan sebagai waktu tempuh kendaraan berdasarkan Google Maps, dengan jarak tempuh terdekat dari rute yang terbentuk.

1.5.2 Asumsi Penelitian

Asumsi yang digunakan dalam penelitian ini adalah:

1. Permintaan ditentukan sebelumnya secara *deterministic*.
2. Kapasitas maksimum pengangkutan oleh kendaraan truk *compactor* adalah 35 kontainer dengan tanpa mempertimbangkan rasio pemadatan sampah.
3. Waktu awal pelayanan di setiap TPS adalah pukul 05.00 WIB.
4. Model VRP yang digunakan dalam penelitian ini bersifat *symetric* VRP.
5. Waktu *loading* diasumsikan secara konstan membutuhkan 1 menit untuk setiap kontainer yang diangkut oleh truk *compactor* sehingga durasi pelayanan pada setiap TPS berbeda berdasarkan jumlah kontainer yang harus dikosongkan. Sedangkan durasi *unloading* semua *compactor* diasumsikan

secara konstan 60 menit. Waktu asumsi berdasarkan hasil pengamatan dengan dihitung rata-rata waktu yang dibutuhkan.

6. Setiap kendaraan melakukan pengisian BBM 1 kali 1 hari tepat setelah melakukan proses *unloading* terakhir dan sesaat sebelum kembali ke depot pada tempat yang telah ditentukan.
7. Jumlah sampah yang masuk di kontainer pada masing-masing TPS dianggap penuh dan harus diangkut setiap hari.
8. Kondisi semua kendaraan yang digunakan tidak mengalami kerusakan.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini dibagi menjadi beberapa bab. Adapun penjelasan dari masing-masing bab adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang dilakukannya penelitian, perumusan masalah yang dibahas pada penelitian, ruang lingkup yang digunakan dalam penelitian, tujuan dan manfaat yang bisa diambil dari penelitian yang dilakukan, serta sistematika yang diterapkan dalam penelitian ini.

BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan berbagai teori yang digunakan pada pelaksanaan penelitian ini. Teori yang digunakan bersumber dari berbagai referensi seperti jurnal dan buku. Penelitian-penelitian yang sudah dilakukan di area yang sama dengan penelitian ini menjadi landasan teori dalam penulisan penelitian ini.

BAB III METODE PENELITIAN

Bab ini membahas tentang tahapan yang dilakukan dalam penelitian untuk mengembangkan model *MSW collection* dengan pendekatan ACO. Metodologi penelitian ini berguna sebagai panduan dalam melakukan penelitian, sehingga penelitian berjalan secara efektif dan sistematis.

BAB IV PENGUMPULAN DATA

Bab ini membahas tahap pengumpulan data dan data apa saja yang digunakan dalam penelitian ini.

BAB V PENGEMBANGAN MODEL

Bab ini membahas mengenai pengembangan model konseptual, model matematis dan model algoritma ACO yang digunakan pada penelitian ini serta dilakukan verifikasi dan validasi untuk model matematis dan algoritma ACO.

BAB VI ANALISIS DAN INTERPRETASI DATA

Bab ini memperlihatkan hasil yang didapatkan dari model yang dibangun serta melakukan analisis sensitivitas terhadap parameter yang dianggap penting.

BAB VII KESIMPULAN DAN SARAN

Bab ini membahas tentang penarikan kesimpulan dari penelitian yang dilakukan serta memberikan saran berdasarkan penelitian yang dilakukan.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang ada dalam penelitian ini. Tinjauan pustaka penting dilakukan secara mendalam untuk mengetahui teori-teori yang terkait dengan penelitian. Berikut adalah tinjauan pustaka dalam penelitian ini.

2.1 *Municipal Solid Waste*

Waste Managemant adalah semua kegiatan dan tindakan yang diperlukan untuk mengelola sampah sejak permulaan hingga pembuangan akhir. Aktifitas yang termasuk didalamnya antara lain: pengumpulan, transportasi, perawatan, pembuangan dan pengelolaan sampah. Sampah bisa berasal dari industri, aktivitas biologis, rumah tangga, dan kasus-kasus khusus di mana itu dapat menimbulkan ancaman bagi kesehatan manusia jika tidak dikelola dengan baik. Sampah dapat berupa zat padat, cair, atau gas dan masing -masing memiliki metode pembuangan dan pengelolaan yang berbeda.

Sampah zat padat adalah jenis sampah yang secara intensitasnya membutuhkan pengelolaan paling tinggi, terlebih *Municipal Solid Waste (MSW)* yaitu sampah padat masyarakat. MSW umumnya dikenal sebagai limbah atau sampah yang terdiri dari barang-barang sehari-hari yang sudah tidak digunakan dan kemudian dibuang, seperti kemasan produk, klipng rumput, perabotan, pakaian, botol, sisa makanan, koran, peralatan, cat, dan baterai. Bahan-bahan ini berasal dari rumah, sekolah, rumah sakit, dan industri (Sipra, *et al.*, 2018).

MSW management adalah satu hal yang hampir setiap pemerintah kota sediakan bagi warganya. Meskipun tingkat layanan, dampak lingkungan dan biayanya sangat bervariasi, pengelolaan MSW dapat dikatakan sebagai layanan kota yang paling penting dan berfungsi sebagai prasyarat untuk tindakan kota lainnya (Hoornweg & Tata, 2012).

Menurut Das & Bhattacharyya (2008), dibutuhkan strategi dan kebijakan tentang manajemen sampah kota. Manajemen sampah kota adalah aktivitas

multidisiplin yang mencakup generasi, pemisahan sumber sampah, penyimpanan, pengumpulan, pengiriman dan transportasi, pengolahan dan pemulihan, dan pembuangan akhir. Selama beberapa dekade terakhir, perhatian terhadap *waste managemant* meningkat karena adanya kepedulian masyarakat terhadap lingkungan. Li, *et al.* (2008) mengungkapkan satu tantangan besar dari banyak hal yang ada didalamnya adalah bagaimana mendistribusikan kendaraan dan kru yang bertugas dalam *waste management collection* secara efektif di wilayah metropolitan yang sedang berkembang.

Waste Managemant menurut Belien *et al.* (2014) merupakan aktivitas yang lebih dari sekadar mengumpulkan sampah. Didalamnya terdapat pengumpulan, transportasi, pemrosesan, daur ulang, pembuangan, dan pemantauan bahan limbah. Banyak faktor, termasuk masalah lingkungan, ekonomi, teknis, legislatif, institusional, dan politik, yang harus dipertimbangkan. Beberapa keputusan penting harus dibuat, diantaranya adalah pembukaan fasilitas baru, karena lokasi yang tersedia menjadi semakin langka, atau perluasan fasilitas saat ini. Kedua, pengalokasian truk ke fasilitas pembuangan tertentu. Ketiga, pengembangan rute yang efisien. Secara alami, pengumpulan sampah adalah aspek siklus yang paling penting dan mahal, karena intensitas kerjanya serta perlunya truk dalam jumlah yang tidak sedikit dalam proses pengumpulan.

2.1.1 *Municipal Solid Waste Collection*

Aktivitas pengumpulan sampah menyumbang sekitar 80% dari semua biaya yang terkait dengan pembuangan sampah. Di masa lalu, pengumpulan limbah padat dilakukan tanpa menganalisa permintaan dan pembangunan rute diserahkan kepada pengemudi. Di sisi lain kota terus berkembang, sehingga menjadi penting adanya upaya peningkatan sistem pengumpulan yang efisien. Secara optimal, harus ada metode yang mencoba memaksimalkan permasalahan tersebut. Hal ini mendorong banyak penelitian yang fokus membahas tentang *MSW collection* yang dikembangkan pada panjang rute, biaya, jumlah kendaraan pengumpul, dsb. (Belien, *et al.*, 2014).

Pengumpulan dan operasi pembuangan dimulai ketika limbah ditempatkan untuk diambil dan berakhir ketika limbah dibuang di fasilitas daur ulang (Li, *et al.*,

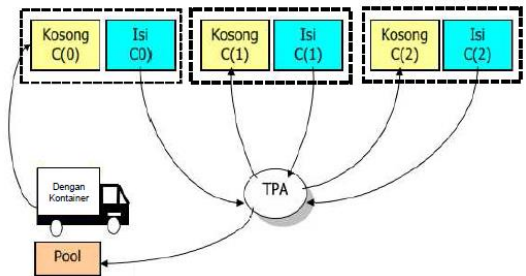
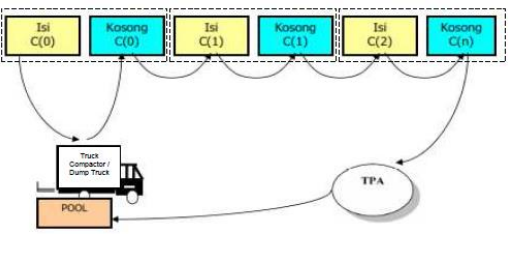
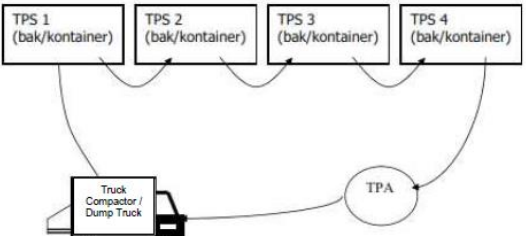
2008). *MSW collection* penelitian ini mengambil sampah dari produsen sampah yang sudah dikumpulkan pada titik-titik tertentu yang sering disebut dengan Tempat Penampungan Sementara (TPS) dan diantarkan ke Tempat Pemrosesan Akhir (TPA). Menurut Peraturan Menteri Pekerjaan Umum Republik Indonesia Nomor: 03/PRT/M/2013 tentang Penyelenggaraan Prasarana Dan Sarana Persampahan Dalam Penanganan Sampah Rumah Tangga Dan Sampah Sejenis Sampah Rumah Tangga, TPS adalah tempat sebelum sampah diangkut ke tempat pendauran ulang, pengolahan, dan/atau tempat pengolahan sampah terpadu dan TPA adalah tempat untuk memroses dan mengembalikan sampah ke media lingkungan secara aman bagi manusia dan lingkungan.

MSW collection dapat dikumpulkan dengan beberapa cara (Hoornweg & Tata, 2012):

1. *House-to-house*: pengumpul sampah mengunjungi setiap rumah untuk mengumpulkan sampah, pengguna umumnya membayar biaya untuk layanan ini.
2. *Community Bins*: pengguna membawa sampah mereka ke tempat sampah komunitas yang ditempatkan pada titik-titik tetap di lingkungan, MSW diambil oleh pemerintah kota atau yang ditunjuk sesuai dengan jadwal yang ditetapkan.
3. *Curbside Pick-Up*: pengguna meninggalkan sampah mereka langsung di luar rumah mereka sesuai dengan jadwal penjemputan sampah yang ditetapkan dengan pihak berwenang setempat.
4. *Self Delivered*: generator mengirimkan sampah langsung ke tempat pembuangan atau stasiun transfer, atau menyewa operator pihak ketiga.
5. *Contracted* atau *Delegated Service*: pemerintah mempekerjakan perusahaan yang mengatur jadwal dan biaya penagihan kepada pelanggan, memberi lisensi kepada operator swasta dan mungkin menunjuk area pengumpulan untuk mendorong efisiensi pengumpulan.

Tabel 2.1 Sistem Pengangkutan Sampah

(Sumber: Peraturan Menteri Pekerjaan Umum Republik Indonesia Nomor: 03/PRT/M/2013)

Ilustrasi Pengangkutan	Proses Pengangkutan
<p>HCS</p> 	<p>Kendaraan dari pool dengan membawa kontainer kosong menuju lokasi kontainer isi untuk mengganti atau mengambil dan langsung membawanya ke TPA, kendaraan dengan membawa kontainer kosong dari TPA menuju kontainer isi berikutnya dan demikian seterusnya sampai rit terakhir.</p>
<p>SCS Mekanis</p> 	<p>Kendaraan dari pool menuju kontainer pertama, sampah dituangkan kedalam truk kompaktor dan meletakkan kembali kontainer yang kosong kemudian kendaraan menuju kontainer berikutnya sampai truk penuh untuk kemudian menuju TPA dan demikian seterusnya sampai rit terakhir.</p>
<p>SCS Manual</p> 	<p>Kendaraan dari poll menuju TPS pertama, sampah dimuat ke dalam truk kompaktor atau truk biasa kemudian kendaraan menuju TPS berikutnya sampai truk penuh untuk kemudian menuju TPA dan demikian seterusnya sampai rit terakhir.</p>

2.1.2 Metode Pengangkutan Sampah

Terdapat pada lampiran Peraturan Menteri Pekerjaan Umum Republik Indonesia Nomor: 03/PRT/M/2013, bahwa kegiatan pengangkutan sampah harus mempertimbangkan: pola pengangkutan, jenis peralatan atau sarana pengangkutan, rute pengangkutan dan aspek pembiayaan.

1. Pola Pengangkutan

Pola pengangkutan sampah dapat dilakukan berdasarkan sistem pengumpulan sampah. Jika pengumpulan dan pengangkutan sampah menggunakan sistem pemindahan (TPS/TPS 3R) atau sistem tidak langsung, proses pengangkutannya dapat menggunakan sistem kontainer angkat (*Hauled Container System* = HCS) ataupun sistem kontainer tetap (*Stationary Container System* = SCS). SCS dapat

dilakukan secara mekanis maupun manual. Sistem mekanis menggunakan *compactor truck* dan kontainer yang kompetibel dengan jenis truknya. Sedangkan sistem manual menggunakan tenaga kerja dan kontainer dapat berupa bak sampah atau jenis penampungan lainnya. Perbedaan dari ketiga system tersebut dapat dilihat pada Tabel 2.1.

Fokus penelitian ini adalah sistem pengangkutan dengan menggunakan SCS mekanis seperti yang ada pada Gambar 1.3. *Stationary Container System* (SCS) merupakan sistem pengumpulan sampah yang wadah pengumpulannya tidak dibawa berpindah-pindah (tetap), dimana wadah tersebut diangkat maupun yang tidak dapat diangkat. Sistem ini biasanya digunakan untuk *container* kecil dengan alat angkut truk *compactor*. Keuntungan dalam pemakaian SCS adalah kendaraan tidak melakukan perjalanan ke daerah pembuangan akhir sebelum kontainer terisi penuh sehingga tingkat utilitas yang tinggi. Sedangkan kerugiannya adalah sistem ini tidak fleksibel dalam pengambilan barang besar dan tidak memiliki volume besar dalam penyimpanannya.

2. Sarana Pengangkutan

Pemilihan jenis peralatan atau sarana yang digunakan dalam proses pengangkutan sampah antara dengan mempertimbangkan beberapa factor sebagai berikut: Umur teknis peralatan (5 – 7) tahun, kondisi jalan daerah operasi, jarak tempuh, karakteristik sampah, tingkat persyaratan sanitasi yang dibutuhkan dan daya dukung pemeliharaan. Jenis sarana dapat berupa beberapa alat angkut seperti truk *compactor*, Penelitian ini menggunakan sarana pengangkutan truk *compactor* yang merupakan kendaraan angkut yang dilengkapi sistem hidrolis untuk memadatkan dan membongkar muatannya. Pengisian muatan masih tetap secara manual dengan tenaga kerja.

3. Rute Pengangkutan

Rute pengangkutan dibuat agar pekerja dan peralatan dapat digunakan secara efektif. Pada umumnya rute pengumpulan dicoba berulang kali, karena rute tidak dapat digunakan pada semua kondisi.

Pada langkah awal pembuatan rute maka ada beberapa langkah yang harus diikuti agar rute yang direncanakan menjadi lebih efisien, yaitu :

- a. Penyiapan peta yang menunjukkan lokasi-lokasi dengan jumlah timbulan sampah.
 - b. Analisis data kemudian diplot ke peta daerah pemukiman, perdagangan, industri dan untuk masing-masing area, diplot lokasi, frekuensi pengumpulan dan jumlah kontainer.
 - c. Layout rute awal.
 - d. Evaluasi layout rute awal dan membuat rute lebih seimbang dengan cara dicoba berulang kali.
4. Aspek Pembiayaan

Biaya pemindahan dan pengangkutan sampah terdiri atas: biaya investasi (sarana yang dibutuhkan untuk pengangkutan seperti truk sampah yang digunakan) dan biaya operasional (operasi dan pemeliharaan pengangkutan sampah).

2.2 *Vehicle Routing Problem*

Salah satu masalah *routing* yang paling terkenal dan paling sederhana adalah *Travelling Salesman Problem* (TSP). TSP adalah permasalahan dimana sejumlah kota harus dikunjungi oleh seorang penjual dan harus kembali ke kota yang sama pada saat mulai. Rute dibangun untuk meminimalkan jarak yang harus ditempuh. Masalah *Vehicle Routing Problem* (VRP) adalah m-TSP dengan adanya permintaan di setiap kota dan kapasitas kendaraan (El-Sherbeny, 2010). Perbedaannya terletak pada kapasitas kendaraan, jika TSP kapasitas dianggap selalu dapat memenuhi pada VRP tidak bisa sehingga memunculkan beberapa solusi misalnya dengan sistem *multitrip*, kendaraan yang bersifat heterogen, dsb.

Menurut Toth & Vigo (2002) VRP adalah jika adanya satu set permintaan transportasi dan suatu armada, dimana selanjutnya akan ditemukan rute kendaraan untuk memenuhi permintaan transportasi dengan armada kendaraan yang diberikan dengan biaya minimum. VRP bertugas menentukan kendaraan mana yang menangani permintaan mana, bagaimana urutannya, sehingga semua rute kendaraan dapat dilaksanakan secara layak. Terdapat banyak model VRP yang

memiliki karakteristik spesifik untuk beberapa jenis model sehingga dapat diselesaikan dengan baik.

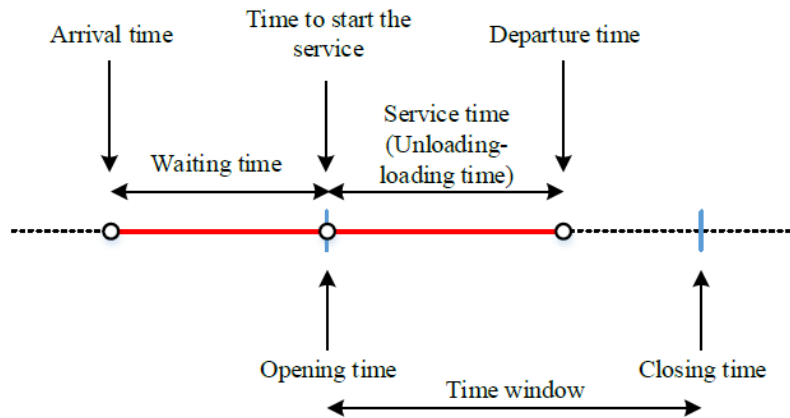
Beberapa diantaranya diungkapkan oleh Solomon (1987), variasi dari VRP adalah sebagai berikut:

1. *Capacitated VRP (CVRP)*, yaitu setiap kendaraan punya kapasitas yang terbatas.
2. *VRP with Time Windows (VRPTW)*, yaitu setiap pelanggan harus dilayani dalam jangka waktu tertentu.
3. *Multiple Depot VRP (MDVRP)*, yaitu distributor memiliki banyak depot untuk menyuplai pelanggan.
4. *VRP with Pick-Up and Delivering (VRPPD)*, yaitu pelanggan mungkin mengembalikan barang pada depot asal.
5. *Split Delivery VRP (SDVRP)*, yaitu pelanggan dilayani dengan kendaraan berbeda.
6. *Stochastic VRP (SVRP)*, yaitu munculnya 'random values' (seperti jumlah pelanggan, jumlah permintaan, waktu pelayanan atau waktu perjalanan).
7. *Periodic VRP*, yaitu pengantar hanya dilakukan dihari tertentu.

2.2.1 Vehicle Routing Problem with Time Window

Setiap pelanggan memiliki *time window* yang menunjukkan waktu buka dan tutup ketika diasumsikan bahwa *time window* konstan dan lebih besar dari atau sama dengan waktu layanan. Jika kendaraan tiba di pelanggan tertentu sebelum waktu bukanya, maka kendaraan harus menunggu dan layanan (*loading unloading*) dimulai pada waktu bukanya. Waktu keberangkatan kendaraan pada pelanggan harus kurang dari atau sama dengan waktu tutup. Gambar 2.1 menunjukkan ilustrasi ketika kendaraan tiba di pelanggan sebelum waktu pembukaannya (Suprayogi & Priyandari, 2017).

Pada VRPTW pelayanan setiap pelanggan harus dilayani dalam suatu interval waktu yang disebut dengan *time window*. *Time window* bisa menjadi *hard* dan *soft*, *hard* apabila pelanggan datang terlalu awal maka pelanggan perlu menunggu hingga jam pelayanan buka, waktu menunggu tidak dikenakan biaya. Sedangkan untuk *soft* setiap pelanggaran akan mendapatkan biaya pinalti.

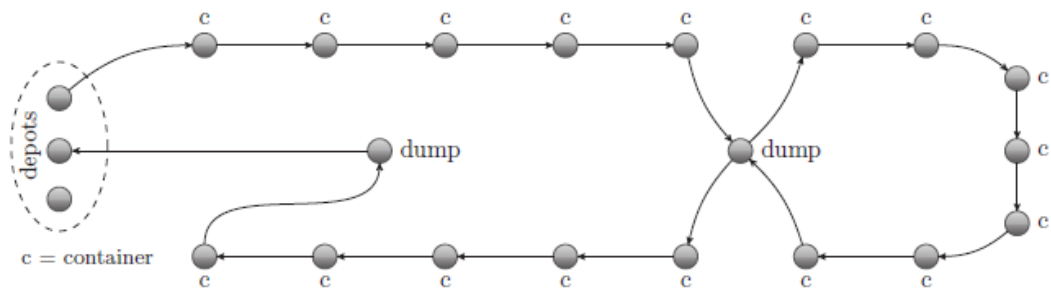


Gambar 2.1 Ilustrasi Kedatangan Berdasarkan *Time Window*
(Sumber: Suprayogi & Priyandari, 2017)

Sebuah *time window* $[a_0, b_0] = [a_{n+1}, b_{n+1}]$ dimana a_0 dan b_0 waktu paling awal berangkat dari depot dan paling akhir kembali ke depot. Pada sebuah *arc node* $(i, j) \in A$ dapat dihilangkan dengan pertimbangan *time window* $a_i + s_i + t_{ij} > b_j$ atau dengan batasan kapasitas $q_i + q_j > Q$ atau dengan faktor lainnya. Dalam pertimbangan *time window* waktu awal pelayanan pada suatu *node* ditambahkan *service time* pada *node* tersebut ditambahkan dengan waktu perjalanan dari *node* tersebut menuju ke *node* selanjutnya harus lebih besar dari b_j diluar itu maka *node* tidak dapat dikunjungi (Toth & Vigo, 2002).

2.2.2 *Vehicle Routing Problem with Intermediate Facility*

Intermediate facility merupakan fasilitas tambahan yang digunakan dalam pembentukan rute. *Intermediate facility* dapat pula diartikan sebagai tempat atau fasilitas untuk mengangkut atau membongkar muatan. Dalam permasalahan pengangkutan *intermediate facility* menunjukkan tempat dimana kendaraan dapat membongkar muatannya (*unloading*). Untuk kasus pengangkutan sampah, kendaraan yang masih belum bermuatan meninggalkan depo dan mulai mengangkut barang dari sejumlah TPS. Ketika muatan dalam kendaraan telah mencapai batas dari kapasitas kendaraan, kendaraan akan menuju ke *intermediate facility* untuk melakukan operasi pembongkaran. Setelah muatan dibongkar



Gambar 2.2 VRP with Intermediate Facility
(Sumber: Markov, *et al.*, 2015)

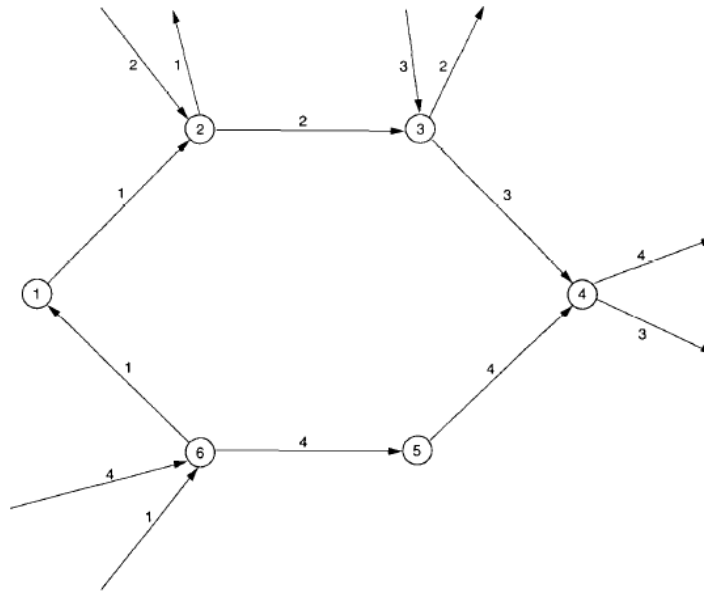
kendaraan dapat memulai pengangkutan kembali dan kembali lagi ke depo ketika jam kerja telah selesai. Masalah pengangkutan ini berbeda dengan pengiriman sehingga memerlukan model khusus sebagai teknik pemecahan (Fitria, *et al.*, 2009).

Pada Gambar 2.3 setiap tur kendaraan dimulai dan berakhir di salah satu dari beberapa depot, tidak harus sama dan merupakan urutan pengumpulan sampah yang diikuti oleh pembuangan yang tersedia ketika kendaraan mencapai kapasitas tertentu. Pada saat kembali ke depot keadaan truk harus kosong. Masalah pengumpulan sampah tersebut dimodelkan sebagai VRP dengan *intermediate facility* (VRP-IF), *dumping* disinilah yang disebut dengan *intermediate facility* (Markov, *et al.*, 2015).

2.2.3 Vehicle Routing Problem with Split Service

Menurut Toth & Vigo (2002) hingga saat ini, kebanyakan VRP diasumsikan bahwa satu *node* dilayani oleh satu kendaraan dalam satu operasi layanan, dimana layanan tersebut tidak dibagi dengan kendaraan lain. Namun kenyataannya, ada dua alasan untuk membagi beberapa layanan: Di satu sisi, jika permintaan melebihi kapasitas kendaraan, lebih dari satu kunjungan tidak dapat dihindari. Di sisi lain, membagi layanan menjadi beberapa permintaan layanan yang lebih kecil dapat menghasilkan penghematan biaya yang signifikan. Sehingga, beberapa penelitian berusaha menghapuskan asumsi tersebut dan menjadi *Vehicle Routing Split Delivery* (VRPSD) yang terangkum dalam Tabel 2.2.

Gambar 2.3 merupakan salah satu contoh ilustrasi dari *split service* VRP. Pada gambar tersebut terdapat 4 *split cycles* yang artinya terdapat 4 *node* dimana *service* pada *node* tersebut tidak hanya dilayani dalam 1 kali kunjungan bahkan dengan kendaraan yang berbeda.



Gambar 2.3 4 *Split Cycle*
(Sumber: Dror, *et al.*, 1994)

2.3 VRP untuk MSW collection

Penelitian ini menggunakan penelitian Buhrkal, *et al.* (2012) sebagai acuan dalam konsep pengembangan model matematis VRP untuk dikembangkan menjadi VRPSDMTTWIF yang sesuai dengan sistem penelitian. Sebelumnya model ini disebut dengan *waste collection vehicle routing problem with time window* (WCVRPTW) yang bertujuan untuk menemukan rute kendaraan, meminimalkan total biaya perjalanan dan memenuhi kapasitas kendaraan dengan semua pelanggan hanya dikunjungi sekali dengan sesuai *time window* nya.

Permasalahan didefinisikan pada sebuah grafik $G = (V, A)$, dimana kumpulan *nodes* $V = V^d \cup V^f \cup V^c$ yang terdiri dari $V^d = \{0\}$, m merupakan tempat pembuangan $V^f = \{1, \dots, m\}$, dan n merupakan konsumen $V^c = \{m + 1, \dots, m + n\}$ dan sekumpulan *arc* adalah $A = \{(i, j) | i, j \in V, i \neq j\}$. Pada $K = \{1, \dots, k\}$ menjadi kumpulan kendaraan pada t_{ij} dan C_{ij} merupakan waktu perjalanan dan biaya yang terkait $arc(i, j)$. Masing-masing *node* $i \in V$ dikaitkan dengan waktu pelayanan s_i dan *time window* $[[a]_i, b_i]$ dan menetapkan q_i sebagai jumlah pengangkutan pada konsumen $i \in V^c$. Diasumsikan bahwa semua kendaraan mempunyai kapasitas C .

Pada model dari permasalahan depot dibagi menjadi 2 yaitu depot berangkat dan depot kembali $[\{0,0\}]'$. Permasalahan ini dapat dimodelkan dengan menggunakan tiga tipe variabel yaitu: $X_{ijl} \in \{0,1\}$ bernilai 1 jika dan hanya jika kendaraan $l \in K$ digunakan pada $arc(i,j) \in A$, d_{il} menunjukkan akumulasi dari permintaan di $node i \in K$ untuk kendaraan $l \in K$ dan w_{il} menunjukkan waktu untuk memulai pelayanan di $node i \in V$ untuk kendaraan $l \in K$.

Berikut merupakan formulasi matematis WCVRPTW:

$$Min = \sum_{(i,j) \in A} C_{ij} \sum_{l \in K} X_{ijl} \quad (2.1)$$

Subject to:

$$\sum_{j \in V} X_{0jl} = 1 \quad \forall l \in K \quad (2.2)$$

$$\sum_{i \in V} X'_{i0l} = 1 \quad \forall l \in K \quad (2.3)$$

$$\sum_{i \in V} \sum_{l \in K} X_{ijl} = 1 \quad \forall j \in V_c \quad (2.4)$$

$$\sum_{i \in V} X_{ijl} = \sum_{i \in K} X_{jil} \quad \forall j \in V_c \cup V_f, l \in K \quad (2.5)$$

$$a_i \leq w_{il} \leq b_i \quad \forall i \in V, l \in K \quad (2.6)$$

$$w_{il} + s_i + t_{ij} \leq w_{jl} + (1 - x_{ijl})M \quad \forall (i,j) \in A, l \in K \quad (2.7)$$

$$\sum_{i \in (0,0')} d_{il} = 0 \quad \forall l \in K \quad (2.8)$$

$$d_{il} + q_i \leq d_{jl} + (1 - x_{ijl})M \quad \forall (i,j) \in V/V_f, j \in l \in V, l \in K \quad (2.9)$$

$$d_{il} \leq C \quad \forall i \in V, l \in K \quad (2.10)$$

$$d_{il} \geq 0 \quad \forall i \in V, l \in K \quad (2.11)$$

$$X_{ijl} \in \{0,1\} \quad \forall (i,j) \in A, l \in K \quad (2.12)$$

Fungsi tujuan pada (2.1) untuk meminimasi biaya perjalanan dengan memenuhi beberapa *constraints*. Pada (2.2) dan (2.3) semua kendaraan harus meninggalkan dan kembali ke depot. *Constraint* (2.4) memastikan bahwa semua konsumen harus dilayani oleh kendaraan tepat satu kali. Pada (2.5) menunjukkan aliran masuk dan keluar harus sama untuk semua *node* kecuali untuk depot. *Time window constraint* dan waktu pelayanan terdapat pada *constraint* (2.6) dan (2.7). *Constraint* (2.8) menunjukkan kendaraan harus kosong pada saat memulai rute perjalanan dan kembali ke depot. *Constraint* (2.9) merupakan akumulasi permintaan untuk semua *nodes* kecuali ditempat pembuangan. *Constraint* (2.10)

merupakan pembatas kapasitas kendaraan dan yang terakhir (2.11) dan (2.12) menentukan *non-negativity* dan *binary variable*.

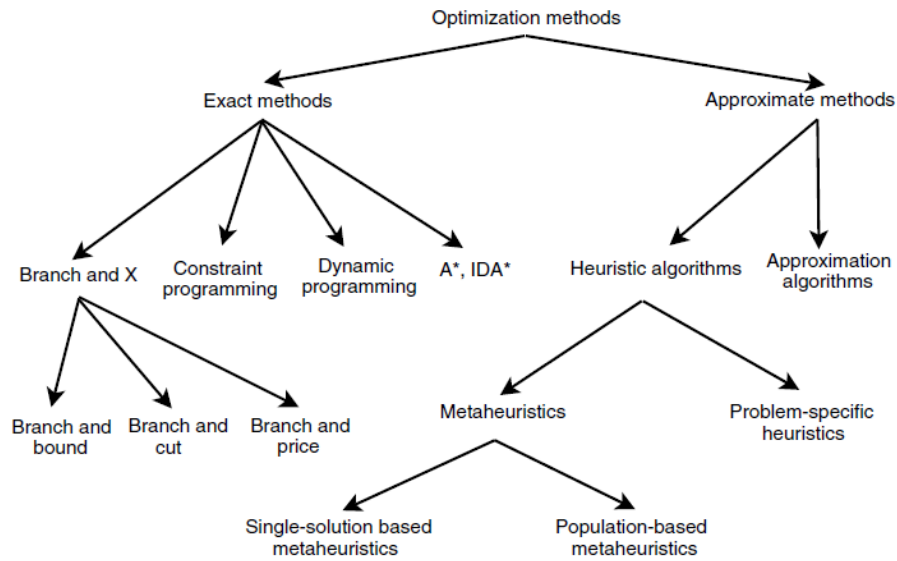
Model matematis ini masih memerlukan pengembangan model agar dapat sesuai dengan sistem yang akan diselesaikan dalam penelitian ini. Salah satu yang harus diubah adalah *constraint* (2.4) yang menyebutkan bahwa semua konsumen harus dilayani oleh kendaraan tepat satu kali sedangkan pada penelitian ini menggunakan sistem *split delivery* sehingga konsumen dapat dikunjungi lebih dari satu kali dengan atau tidak dengan menggunakan kendaraan yang sama.

2.4 Optimasi

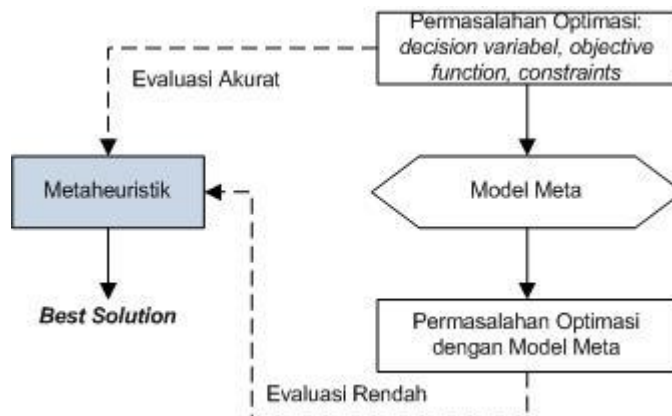
Menurut Santosa & Willy (2011), optimasi bisa dijelaskan sebagai suatu kumpulan formula matematis dan metode numerik untuk menemukan dan mengidentifikasi kandidat terbaik dari sekumpulan alternatif tanpa harus secara eksplisit menghitung dan mengevaluasi semua alternatif yang mungkin. Masalah optimasi biasanya dinyatakan dalam bentuk fungsi matematik. Berbagai jenis permasalahan optimasi menurut Talbi (2009) akan dapat diselesaikan dengan metode *exact* atau metode pendekatan, pembagian lebih jelasnya dapat dilihat pada Gambar 2.4. Metode *exact* mendapatkan solusi optimal dan menjamin optimalitasnya. Untuk masalah NP-complete, algoritma yang tepat adalah *non-polynomialtime algorithm* (kecuali $P = NP$). Metode pendekatan (atau heuristik) menghasilkan solusi berkualitas tinggi dalam waktu yang wajar untuk penggunaan praktis, tetapi tidak ada jaminan untuk menemukan solusi optimal global.

2.4.1 Metaheuristik

Santosa & Willy (2011) menyebutkan bahwa dalam dekade terakhir, kajian dan studi mengenai bidang metaheuristik untuk menyelesaikan permasalahan optimasi tumbuh secara signifikan, baik dari segi Teknik metaheuristik sendiri maupun dari sisi aplikasinya. Para peneliti telah menunjukkan kemampuan metode ini dalam memecahkan permasalahan kombinatorial, *hard combinatorial* dengan skala yang cukup besar dengan waktu komputasional yang kompetitif. Metaheuristik adalah metoda untuk mencari solusi yang memadukan interaksi antara prosedur pencarian lokal dan strategi yang lebih tinggi untuk



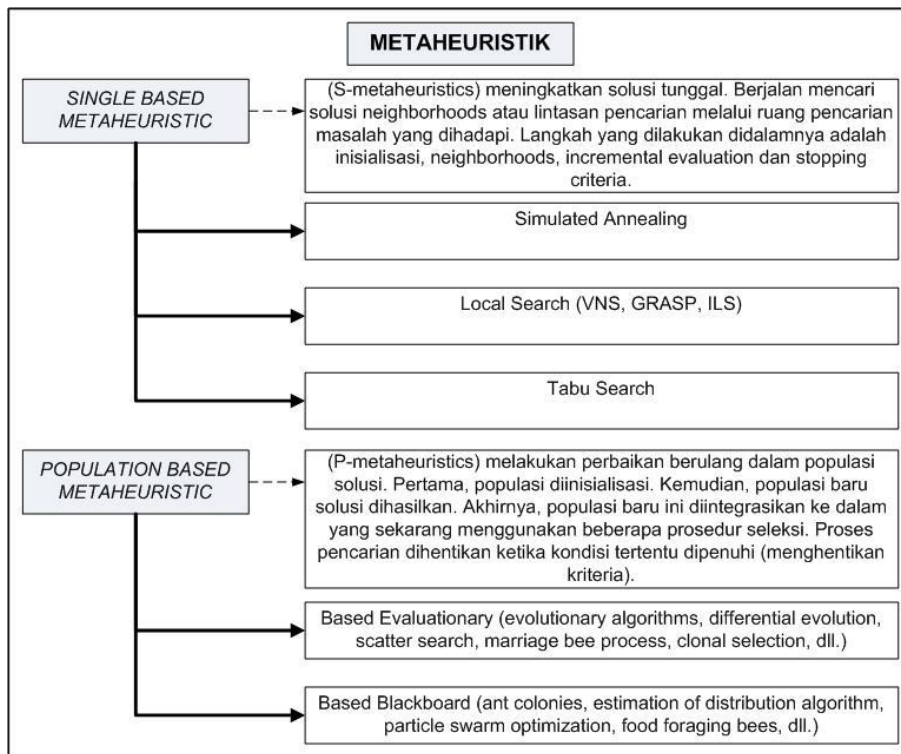
Gambar 2.4 Klasifikasi Metode Optimasi
(Sumber: Talbi, 2009)



Gambar 2.5 Optimasi dengan Metaheuristik
(Sumber: Talbi, 2009)

menciptakan proses yang mampu keluar dari optimal lokal dan melakukan pencarian di ruang solusi untuk menemukan solusi global. Bagan penyelesaian masalah optimasi dengan menggunakan metaheuristik ada pada Gambar 2.5.

VRP dan VRPTW keduanya merupakan contoh masalah optimasi kombinatorial NP-hard (El-Sherbeny, 2010). Permasalahan yang sulit diselesaikan (NP-hard) dengan ukuran sedang dan / atau struktur yang sulit dari sisi input datanya, menjadi salah satu contoh karakteristik masalah optimasi yang direkomendasikan untuk diselesaikan dengan metaheuristik. Ada banyak algoritma yang ada dalam metaheuristik yang dapat dibagi menjadi 2 karakteristik yang

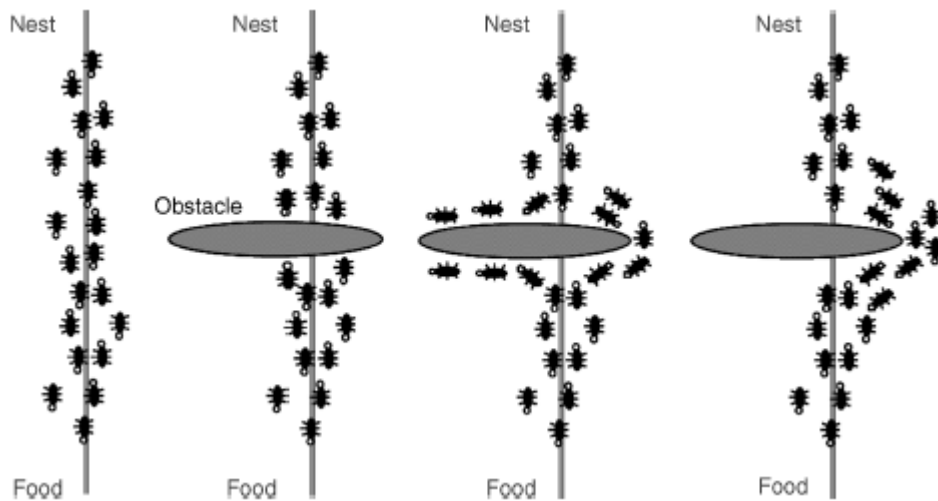


Gambar 2.6 Metode Metaheuristik

berbeda yaitu: *single solution based metaheuristic* dan *population based metaheuristic*. Bagan yang menampilkan perbedaan keduanya beserta algoritma apa saja yang termasuk didalamnya dapat dilihat pada gambar 2.6 (Talbi, 2009).

2.4.2 Ant Colony Optimization

Ant Colony Optimization (ACO) termasuk dalam kelompok *swarm intelligence*, yang merupakan salah satu jenis pengembangan paradigma yang digunakan untuk menyelesaikan masalah optimasi dimana inspirasi yang digunakan untuk memecahkan masalah tersebut berasal dari perilaku kumpulan semut. *Ant Based Techniques* pertama kali digunakan oleh Dorigo, *et al.* (1996) dengan menggunakan ACO untuk menyelesaikan kasus TSP. ACO biasanya digunakan untuk menyelesaikan *discrete optimization problem* dan persoalan yang kompleks dimana terdapat banyak variabel. Hasil yang didapatkan ACO memang tidak menjajikan akan menemukan solusi optimal tetapi akan mendekati solusi optimal tersebut (Santosa & Willy, 2011).



Gambar 2.7 Ilustrasi Perjalanan Semut Pada ACO
(Sumber: Talbi, 2009)

ACO meniru perilaku semut dalam mencari sumber makanan bermula dan kembali ke sarangnya yang ternyata secara alami semut akan melewati rute terpendek berdasarkan kuatnya aroma *pheromone* dan jarak lintasan. Dalam ACO setiap semut dalam kawanan yang berjalan akan meninggalkan *pheromone* (semacam zat kimia) pada jalur yang dilaluinya. Hal tersebut yang akan menjadi sebuah sinyal bagi sesama semut. Jalur yang pendek akan menyisakan sinyal yang lebih kuat. Semut berikutnya akan cenderung memilih untuk mengikuti jalur dengan sinyal yang paling kuat, sehingga jalur terpendek akan ditemui karena lebih banyak semut yang akan melewati jalur tersebut. Ilustrasi perjalanan semut dapat dilihat pada Gambar 2.7.

Pada awal ditemukannya algoritma ACO oleh penemunya digunakan untuk menyelesaikan persoalan *Travelling Salesman Problem* (TSP). Dalam Algoritma ACO untuk TSP, jika terdapat n kota akan ada $\binom{n(n-1)}{2}$ buah ruas, dan juga memiliki $\frac{(n-1)!}{2}$ rute yang mungkin. Jarak yang digunakan dalam TSP standar merupakan jarak simetris, artinya jarak antara kota r ke kota s sama dengan jarak kota s ke kota r $d(r, s) = d(s, r)$. Dalam TSP setiap semut memulai dan mengakhiri rutenya pada kota yang sama. Secara berulang, setiap semut akan menemukan rute yang dilaluinya hingga kembali ke kota awal yang sama. Pemilihan kota-kota yang akan dilalui didasarkan pada suatu fungsi probabilitas, dengan mempertimbangkan *visibility* (*invers* dari jarak) kota tersebut dan jumlah

pheromone yang terdapat pada ruas yang menghubungkan kota r dan kota s . semut akan memilih untuk bergerak menuju ke kota-kota yang dihubungkan dengan ruas yang pendek dan atau memiliki tingkat *pheromone* yang tinggi.

Berikut merupakan algoritma ACO untuk kasus TSP menurut Santosa & Ai (2017):

1. Langkah 1

Asumsikan jumlah semut sejumlah N . Tentukan jumlah *pheromone* awal τ_{ij}^1 yang sama untuk semua ruas antar kota. Agar lebih mudah untuk iterasi pertama nilai $\tau_{ij}^1 = 1$ untuk semua ruas ij . Tentukan *visibility* antar kota sebagai dengan *invers* jarak $\frac{1}{jarak}$ dari masing-masing kota ke kota yang lain atau dengan menggunakan persamaan 2.12. Set iterasi $t = 1$.

$$h = \frac{1}{jarak} \quad (2.13)$$

2. Langkah 2

a. Hitung probabilitas (p_{ij}) untuk memilih ruas atau nilai diskret x_{ij} menggunakan

$$p_k(r, s) = \begin{cases} \tau(r, s)^\alpha \eta(r, s)^\beta / \sum_{j \in M_k} \tau(r, u)^\alpha \eta(r, u)^\beta \\ 0 \end{cases} \quad (2.14)$$

jika $s \in M^k$ dan 0 untuk lainnya.

Nilai α merupakan bobot *pheromone* τ dan β adalah bobot yang mengontrol *visibility* terhadap tingkat *pheromone* τ .

b. Ruas tertentu akan dipilih oleh ruas k berdasarkan bilangan random dalam *range* (0,1). Untuk itu perlu juga menentukan *range* probabilitas kumulatif yang berkaitan dengan pilihan ruas. Jadi jika ada p kemungkinan nilai variabel, maka akan ada p pilihan *range* probabilitas. Ruas yang mempunyai nilai perkalian τ dan η yang besar mempunyai peluang besar untuk terpilih.

3. Langkah 3

a. Bangkitkan bilangan random r dalam *range* (0,1) satu untuk setiap semut untuk setiap ruas yang akan dipilih.. Tentukan nilai diskret yang mewakili ruas untuk semut k dengan menggunakan bilangan randm dari

langkah sebelumnya dan area probabilitas kumulatif. Setiap semut akan menjalani rute tertentu.

- b. Evaluasi nilai fungsi tujuan dengan cara menghitung jarak total setiap rute oleh semut k , f_k , $k = 1, 2, \dots, N$. Tentukan lintasan terbaik diantara N ruas atau lintasan yang sudah dipilih oleh semut-semut yang berbeda dengan persamaan

$$f_{best} = \min_{k=1,2,\dots,N} \{f_k\} \quad (2.15)$$

Gunakan jarak total yang ditempuh oleh setiap semut untuk menghitung nilai Persamaan 2.16

$$\Delta\tau(r, s) = \frac{1}{f_k} \quad (2.16)$$

$\Delta\tau(r, s)$ adalah *pheromone* yang ditambahkan pada setiap ruas yang dilalui seekor semut.

4. Langkah 4

Uji konvergensi dari proses. Dalam hal ini konvergensi dapat diartikan jika semua semut mengambil lintasan terbaik yang sama. Jika belum konvergen, koloni semut akan kembali ke sarang dan memulai pencarian makanan lagi. Set iterasi, $t = t + 1$, dan *update pheromone* untuk setiap ruas dengan

$$\tau_{ij}^{(t)} = \tau_{ij}^{old} + \sum_k \Delta\tau^{(k)} \quad (2.17)$$

Dimana τ_{ij}^{old} menyatakan jumlah *pheromone* dari iterasi sebelumnya yang tertinggal setelah penguapan

$$\tau_{ij}^{old} = (1 - \rho)\tau_{ij}^{t-1} \quad (2.18)$$

$\Delta\tau^k$ jumlah *pheromone* yang ditambahkan oleh semut terbaik k pada ruasnya dan dijumlahkan untuk semua semut yang menempuh ruas yang sama (jika ada lebih dari satu semut menempuh lintasan yang sama). $\rho \in (0,1)$ adalah parameter tingkat penguapan atau evaporasi. Dengan menggunakan nilai $\tau_{ij}^{(t)}$ lanjutkan ke langkah 2. Langkah 2,3 dan 4 diulang sampai proses konvergen tercapai atau berhenti disejumlah iterasi sebelum optimal tercapai.

ACO telah diterapkan diberbagai bidang, diantaranya ACO untuk VRP Bell & McMullen (2004), *Capacited Vehicle Routing Problem (CVRP)* Mazzeo & Loiseau (2004) dan pada tahun 2018 ACO tetap digunakan dalam pengembangan

kasus VRP diantaranya ada penelitian Tirkolaee, *et al.* (2018) yang menyelesaikan model *multi-trip capacited arc routing problem* untuk kasus pembuangan sampah perkotaan. Berdasarkan pengalaman empirik hasil penerapan ACO pada permasalahan yang bersifat kontinyu menghasilkan solusi yang kurang bagus sehingga perlu adanya modifikasi, karena awal ditemukannya ACO ditujukan untuk menyelesaikan permasalahan optimasi diskret.

2.5 Penelitian Terdahulu

Penelitian Ramdhani, *et al.* (2016) menggunakan objek penelitian yang sama dengan penelitian ini. Pada penelitian tersebut menggunakan model *vehicle assignment problem*, mencari jumlah kendaraan minimal dan bagaimana penugasan terbaik dari jumlah kendaraan yang dibutuhkan. Sedangkan penelitian ini akan membuat rute baru dengan jarak minimal yang kemudian dilakukan penugasan dari setiap kendaraan yang dibutuhkan. Li, *et al.* (2008) menggunakan istilah *vehicle scheduling problem* untuk menjadwalkan kendaraan dalam pengumpulan sampah dengan tujuan meminimalkan biaya operasional kendaraan dan *fixed cost*. Das & Bhattacharyya (2015) mengangkat kasus *covering salesman problem*, bagaimana dengan menggunakan 1 kendaraan dapat mengumpulkan sampah sesuai dengan permintaan dengan jarak minimal.

Pada kasus transportasi, model yang biasanya digunakan adalah VRP yang telah disesuaikan dengan kebutuhan masing-masing permasalahan. Begitu juga dengan kasus pada penelitian ini yaitu *MSW collection* yang menggunakan model dasar VRP yang disesuaikan lagi dengan kebutuhan. Misalnya terkait produk yang diangkut pada penelitian Abdulkader, *et al.* (2015) dan Reed, *et al.* (2014) menggunakan *Multi Compartement Vehicle Routing Problem (MCVRP)*, dimana muatan kendaraan yang menjadi fokus utamanya, 1 kendaraan dapat memuat lebih dari satu jenis barang dan tidak boleh tercampur, sehingga memudahkan proses *unloading*.

Terkait tempat pembuangan akhir yang harus dikunjungi sebelum kendaraan kembali ke depot, berarti pada VRP perlu ditambahkan terkait *intermediate facility*. Pada penelitian Son & Louati (2016), Buhrkal, *et al.* (2012), Benjamin & Beasley (2010) dan Li, *et al.*, (2008) menggunakan *multiple*

intermediate facility karena memiliki lebih dari satu tempat pembuangan akhir setelah muatan dalam kendaraan terisi penuh, *intermediate facility* yang dipilih adalah yang paling dekat berdasarkan tempat terakhir pengisian sebelum kendaraan bermuatan penuh. Sedangkan pada penelitian ini sama seperti yang digunakan Fitria, *et al.* (2009) yaitu menggunakan satu *intermediate facility* sebagai tempat pembuangan akhir.

Pada penelitian Buhrkal, *et al.* (2012) dan Benjamin & Beasley (2010) menggunakan *constraints time window* sebagai batasan jam operasional kerja, *constraints* ini juga akan muncul pada penelitian ini. Kendaraan yang digunakan dalam kasus *MSW collection* berbeda-beda sesuai dengan daerah masing-masing, namun rata-rata masing menggunakan kendaraan homogen dengan kapasitas yang sama seperti yang ada pada penelitian ini. Jenis kendaraan tersebut lain halnya dengan Son & Louati (2016) dan Markov, *et al.* (2015) yang menggunakan kendaraan heterogen dengan kapasitas kendaraan yang berbeda-beda.

Adanya *time window* dan kapasitas kendaraan dalam memenuhi kebutuhan seluruh LPS maka banyak penelitian yang berupaya untuk mencari solusi optimal diantaranya adalah dengan model VRP yang diusulkan oleh Buhrkal, *et al.* (2012), dimana kendaraan dapat melayani lebih dari satu rute dalam 1 kali periode waktu. Selain itu juga ada model VRP *split service* yang diusulkan oleh Suthikarnnarunai, N. (2008) dan Wilck IV & Cavalier (2012), dimana permintaan dalam suatu *node* dapat dibagi sehingga dalam satu *node* juga dapat dikunjungi oleh lebih dari satu kendaraan. Permintaan terus bertambah sehingga optimasi terkait pembuatan rute sangat penting. Penelitian ini menggunakan VRP *split service* dan menggunakan konsep lebih dari satu ritase untuk suatu kendaraan karena permintaan yang banyak dari keseluruhan LPS, maka hal ini akan terjadi apabila *time window* memungkinkan untuk melakukan perjalanan kembali dari LPA atau yang disebut dengan penambahan ritase kendaraan. *Split service* tidak bisa dihindari pada kasus *MSW collection* kota Surabaya, karena beberapa permintaan deterministik untuk beberapa LPS melebihi dari kapasitas kendaraan, sehingga perlu adanya pengambilan sampah kembali dengan atau tidak dengan kendaraan yang sama.

Banyak penelitian yang telah dilakukan terkait dengan pengembangan metode guna menyelesaikan permasalahan transportasi. Salah satu contoh diantaranya adalah Suthukarnnarunai (2008) yang telah mengembangkan dan menyelesaikan permasalahan transportasi pada rute bus sekolah dengan kendaraan yang *heterogeneous* karena kapasitas kendaraan yang berbeda. Penelitian ini mengembangkan model *exact integer programming* dan metode *sweep heuristic* dengan *2 opt exchange* untuk mendapatkan rute terbaik. Hasilnya *exact integer programming* dapat menyelesaikan beberapa solusi optimal dalam berbagai kasus. Sedangkan untuk metode heuristik dapat menemukan solusi yang baik untuk semua kasus dengan waktu komputasi yang cepat.

Penelitian Ramdhani, *et al.* (2016), Markov, *et al.* (2015) dan Das & Bhattacharyya (2015) menggunakan metode *exact* untuk pencarian solusinya. Markov, *et al.* (2015) dan Das & Bhattacharyya (2015) menggunakan tambahan metode heuristik dalam pencariannya sehingga dengan waktu yang lebih sedikit dapat menemukan solusi yang baik, sedangkan Ramdhani, *et al.* (2016) hanya menggunakan metode *exact* sehingga dibutuhkan waktu yang lama dan dalam waktu tertentu akhirnya *running* dihentikan untuk mendapatkan solusi sebelum mencapai nilai optimal.

Penggunaan metode heuristik lainnya juga dilakukan oleh Fitria, *et al.* (2009) yang menggunakan *sequential insertion*. Li, *et al.*, (2008) yang menggunakan *auction algorithm*. Buhrkal, *et al.* (2012) menggunakan *adaptive large neighbourhood search*. Keempat penelitian ini sudah berhasil mendapatkan solusi. Metode heuristik adalah metode pencarian solusi yang lebih baik dari pada *exact* dengan biaya komputasi yang wajar namun tidak dapat menjamin akan ditemukannya solusi optimal tetapi solusi yang mendekati optimal dan layak. Seringkali heuristik hanya dapat digunakan masalah spesifik, sehingga metode yang bekerja untuk satu masalah tidak dapat digunakan untuk memecahkan masalah yang berbeda, sehingga perlu ditingkatkan kembali terkait metode pencariannya.

Benjamin & Beasley (2010) menggunakan metode metaheuristik dalam pencarian solusi dan membandingkan dengan metode heuristik sebelumnya. Metode heuristik yang digunakan adalah *tabu search*, *variable neighbourhood*

search dan *variable neighbourhood tabu search*. Hasilnya dengan ketiga metode metaheuristik ini dapat mengurangi total jarak maupun total waktu komputasi yang digunakan. Reed, *et al.* (2014), Karadimas, *et al.* (2007), dan Bell & McMullen (2004) menggunakan ACO serta Abdulkader, *et al.* (2015) yang menambahkan 2 *local search* dalam metode ACO adalah pencarian solusi dengan model yang berbeda yaitu *population based* dimana perbaikan solusi dilakukan dalam populasi tidak menggunakan *single solution based*, hasilnya untuk masalah dengan ukuran yang besar dapat dilakukan dengan waktu yang lebih sedikit. Penambahan metode heuristik *local search* dibutuhkan untuk menyelesaikan masalah *multi compartement vehicle* yang efektif.

Penelitian ini merupakan permasalahan *Np-hard*, ukuran yang harus diselesaikan juga tidak sedikit sehingga akan digunakan metode ACO dalam pencarian solusinya dengan harapannya dapat mendapatkan solusi rute dengan total jarak yang lebih baik dari sebelumnya dan dalam waktu komputasi yang cepat. Posisi penelitian berdasarkan model yang digunakan dapat dilihat pada Tabel 2.3, posisi penelitian berdasarkan metode yang digunakan dapat dilihat pada Tabel 2.4, dan rangkuman penelitian sebelumnya dapat dilihat lebih detail pada tabel 2.5.

Tabel 2.2 Posisi Model Penelitian

Model yang Digunakan															
Penelitian	Son, L. H. & Louati, A. (2016)	Iliya Markova, Sacha Varoneb dan Michel Bierlairea (2016)	Nasir Ramdhani, Imam Baihaqi dan Nurhadi Siswanto (2016)	Abdulkader, M. M., Gajpal, Y. & Elmekawwy, T. Y. (2015)	Swaapan Das dan Bidyut Kr. Bhattacharya (2015)	Reed, M., Yiannakou, A. & Evering, R. (2014)	Karja Buhrkal, Allan Larsen Stefan Ropke (2012)	Wilck IV & Cavalier (2012)	Benjamin, A. M. & Beasley, J. E. (2010)	Lisye Fitria, Susy Susanty dan Suprayogi (2009)	Jing-Quan Li, Denis Borenstein, dan Pitu B. Mirchandani (2008)	N. Suthikarnarunai (2008)	Karadimas, N. V., Papatzelou, K. & Loumos, V. G. (2007)	Bell, J. E. & McMullen, P. R. (2004)	Penelitian ini
VRP														√	
<i>Heterogeneous vehicles routing problem</i>	√	√													
<i>VRP with Intermediete Facility</i>										√					√
<i>VRP with Multiple Intermediete Facility</i>	√						√		√		√				
<i>VRP with Time Window</i>							√		√						√
<i>VRP with Multipletrips</i>							√								√
<i>VRP with Split service</i>								√				√			√
<i>Multi Compartement Vehicle Routing Problem (MCVRP)</i>				√		√									
<i>Asymmetric Travelling Salesman Problem (ATSP)</i>													√		
<i>Covering Salesman Problem</i>					√										
<i>Vehicle Assignment Problem</i>			√												
<i>Vehicle Scheduling Problem</i>											√				

Tabel 2.3 Posisi Metode Penelitian

Metode Penyelesaian															
Penelitian	Son, L. H. & Louati, A. (2016)	Iliya Markova, Sacha Varoneb dan Michel Bierlairea (2016)	Nasir Ramdhani, Imam Baihaqi dan Nurhadi Siswanto (2016)	Abdulkader, M. M., Gajpal, Y. & Elmekkwaw, T. Y. (2015)	Swaapan Das dan Bidyut Kr. Bhattacharya (2015)	Reed, M., Yiannakou, A. & Evering, R. (2014)	Karja Buhrkal, Allan Larsen Stefan Ropke (2012)	Wick IV & Cavalier (2012)	Benjamin, A. M. & Beasley, J. E. (2010)	Lisye Fitria, Susy Susanty dan Suprayogi (2009)	Jing-Quan Li, Denis Borenstein, dan Pitu B. Mirchandani (2008)	N. Suthikarnarunai (2008)	Karadimas, N. V., Papatzelou, K. & Loumos, V. G. (2007)	Bell, J. E. & McMullen, P. R. (2004)	Penelitian ini
Exact:															
Mixed Integer Linear Programming		√			√										
Integer Linear Programming			√									√			
Heuristic:															
Heuristic Method					√			√							
Sequential Insertion										√					
Sweep Algorithm												√			
2 Opt Change				√								√			
Multiple Neighborhood Search		√													
Auction Algorithm											√				
Adaptive Large Neighborhood search							√								
Metaheuristic:															
Ant Colony Algorithm						√							√	√	√
Hybridized Ant Colony Algorithm				√											
Variable Neighbourhood Search									√						
Tabu Search									√						
Variable Neighbourhood Tabu Search									√						
Pendekatan Lain:															
ArcGIS Network Analyst	√														
Analytical Hierarchy Process	√														

Tabel 2.4 Penelitian Sebelumnya

NO	Penulis (Tahun)	Tujuan Penelitian	Objective Function	Karakteristik Sistem	Tipe permintaan	Tipe Kendaraan	Objek Penelitian
1	Son, L. H. & Louati, A. (2016)	Membangun model vehicle routing yang efektif untuk mengoptimalkan total jarak perjalanan kendaraan, emisi lingkungan dan biaya investasi	Memaksimalkan jumlah sampah yang dapat dikumpulkan dan meminimalkan emisi lingkungan oleh kendaraan	<i>Vehicle routing</i> untuk MSW <i>collection</i> dengan <i>multiple transfer station, gather sites</i> dan <i>inhomogeneous vehicle</i>	Deterministik	Heterogeneous	MSW <i>collection</i> di Kota Danang, Vietnam
2	Iliya Markova, Sacha Varoneb dan Michel Bierlairea (2016)	Solusi untuk permasalahan sampah daur ulang yang dikumpulkan dan dikirim menggunakan kendaraan yang heterogen dengan perbedaan volume dan berat drai kapasitas, biaya tetap kendaraan, biaya jarak perunit, dan biaya upah.	meminimumkan biaya dari total biaya tetap, biaya jarak perunit. Dan biaya upah serta total biaya relokasi oleh faktor berat.	heterogeneous fixed fleet vehicles routing problem	Deterministik	Heterogeneous	<i>Waste collection</i> industri di Geneva

Tabel 2.3 Penelitian Sebelumnya (Lanjutan)

NO	Penulis (Tahun)	Tujuan Penelitian	Objective Function	Karakteristik Sistem	Tipe permintaan	Tipe Kendaraan	Objek Penelitian
3	Nasir Ramdhani, Imam Baihaqi dan Nurhadi Siswanto (2016)	menentukan jumlah kendaraan dengan meminimasi jumlah kendaraan yang digunakan dengan penjadwalan keberangkatan kendaraan.	meminimumkan jumlah kendaraan yang dapat digunakan untuk memenuhi permintaan pengangkutan sampah.	<i>Vehicle Assignment Problem</i>	Deterministik	Homogen	Dinas Kebersihan dan Pertamanan Kota Surabaya
4	Abdulkader, M. M., Gajpal, Y. & Elmekawy, T. Y. (2015)	Menyelesaikan kasus VRP dimana kendaraan membawa produk yang berbeda	Menemukan total biaya dari keseluruhan jarak yang ditempuh oleh seluruh kendaraan	<i>Multi Compartement Vehicle Routing Problem (MCMVRP)</i>	Deterministik	Homogen	14 set CVRP dari penelitian sebelumnya (christofides)

Tabel 2.3 Penelitian Sebelumnya (Lanjutan)

NO	Penulis (Tahun)	Tujuan Penelitian	Objective Function	Karakteristik Sistem	Tipe permintaan	Tipe Kendaraan	Objek Penelitian
5	Swaapan Das dan Bidyut Kr. Bhattacharrya (2015)	Mengoptimalkan pengumpulan sampah dan skema transportasi yang fokus pada minimasi panjang dari masing-masing pengumpulan dan rute transportasi.	Minimum total biaya managemant (biaya pengumpulan, biaya transportasi, biaya proses, biaya pembuangan, biaya kendaraan dan biaya tetap), minimum biaya pengumpulan sampah di tingkat bawaaah dimana masing-masing pengumpul sampah dari berbagai sumber sampah, optimasi rute antara pusat pengumpulan sampah dan stasiun pengiriman untuk meminimalkan biaya transportasi sampah, dan meminimalkan jumlah dari biaya perjalanan dan biaya pengiriman kunjungan dari stasiun pengiriman ke tempat proses sampah.	<i>Covering salesman problem</i>	Deterministik	Heterogen	<i>Municipal waste collection</i>

Tabel 2.3 Penelitian Sebelumnya (Lanjutan)

NO	Penulis (Tahun)	Tujuan Penelitian	Objective Function	Karakteristik Sistem	Tipe permintaan	Tipe Kendaraan	Objek Penelitian
6	Reed, M., Yiannakou, A. & Evering, R. (2014)	Mendemonstrasikan kegunaan Ant Colony System (ACS) untuk menyelesaikan CVRP yang dimodelkan untuk pengumpulan sampah yang akan didaur ulang dari sampah masyarakat	Mengoptimalkan rute dengan kendaraan kompartemen	<i>CVRP with multiple compartement</i>	Deterministik	Homogeneous dengan kompartemen	14 set CVRP dari penelitian sebelumnya (christofides)
7	Karja Buhrkal, Allan Larsen Stefan Ropke (2012)	Dengan membangun algoritma adaptive large neighborhood untuk menyelesaikan permasalahan dan ilustrasi kegunaan algoritma untuk meningkatkan tujuan.	Menemukan rute untuk setiap kendaraan dengan meminimalkan biaya perjalanan dan kepuasan dari kapasitas kendaraan, dengan semua konsumen dilayani tepat satu kendaraan dalam batasan waktunya.	<i>Waste collection vehicle routing problem with time windows</i>	Deterministik	Homogeneous	Waste collection di North America
8	Wilck IV & Cavalier (2012)	Mengembangkan model heuristik untuk SDVRP	Menemukan rute minimum dengan total jarak minimum	<i>Split delivery vehicle routing problem (SDVRP)</i>	Deterministik	Homogen	11 data set dari Belenguer, et al.

Tabel 2.3 Penelitian Sebelumnya (Lanjutan)

NO	Penulis (Tahun)	Tujuan Penelitian	Objective Function	Karakteristik Sistem	Tipe permintaan	Tipe Kendaraan	Objek Penelitian
9	Benjamin, A. M. & Beasley, J. E. (2010)	Membandingkan beberapa macam metode metaheuristik dalam menentukan rute optimal termasuk jumlah kendaraan yang digunakan	Menemukan rute minimum dengan total jarak minimum	VRP with time window, driver rest period dan multiple disposal facilities	Deterministik	Homogen	2092 pelanggan dan 19 waste disposal facility
10	Lisye Fitria, Susy Susanty dan Suprayogi (2009)	mengembangkan model <i>Vehicle Routing Problem with Multiple Trips and Intermediate Facility</i> (VRPMTIF) untuk selanjutnya diterapkan pada pembuatan rute pengumpulan sampah.	meminimumkan jumlah kendaraan yang digunakan (lebih diprioritaskan) dan total waktu penyelesaian.	<i>Vehicle Routing Problem with Multiple trips and intermediate facility</i> (VRPMTIF)	Deterministik	Heterogen	Waste Collection di Bandung
11	Jing-Quan Li, Denis Borenstein, dan Pitu B. Mirchandani (2008)	Menyelidiki perencanaan operasional mingguan dari pengumpulan limbah padat di Porto Alegre.	Meminimumkan total operasi dan <i>fixed cost</i> kendaraan	<i>single depot vehicle scheduling problem</i> (SDVSP)	Deterministik	Homogen	<i>Solidwaste collection</i> di City of Porto Alegre, Brazil

Tabel 2.3 Penelitian Sebelumnya (Lanjutan)

NO	Penulis (Tahun)	Tujuan Penelitian	Objective Function	Karakteristik Sistem	Tipe permintaan	Tipe Kendaraan	Objek Penelitian
12	N. Suthikarnnarunai (2008)	Membangun model VRP dengan menggunakan metode sweep heuristic dengan perubahan 2 opt dan tour travelling salesman serta model integer programming untuk split delivery VRP.	meminimalkan total biaya perjalanan.	VRP <i>split delivery</i>	Deterministik	Heterogen	Pelayanan bus di Universitas Thai Chamber of Commerce (UTCC)
13	Karadimas, N. V., Papatzelou, K. & Loumos, V. G. (2007)	Mengidentifikasi skenario dengan biaya efektif untuk pengumpulan sampah, mengestimasi biaya pengerjaan dan mensimulasikan model	Meminimasi total panjang tour truk pengangkutan sampah dalam mengunjungi <i>loading spots</i>	<i>Asymmetric Travelling Salesman Problem (ATSP)</i>	Deterministik	Homogen	<i>Municipality waste collection Di Athens</i>
14	Bell, J. E. & McMullen, P. R. (2004)	Menambah kemampuan ACO untuk menyelesaikan persoalan VRP dalam ukuran yang besar	Menemukan rute minimum dengan jarak dan atau biaya minimal yang dapat melayani sejumlah pelanggan menggunakan banyak kendaraan tertentu	VRP	Deterministik	Homogen	3 kasus VRP dengan ukuran yang berbeda

(Halaman ini sengaja dikosongkan)

BAB III

METODOLOGI PENELITIAN

Pada bab ini telah diberikan gambaran kerangka berpikir dalam pelaksanaan penelitian ini yang terdiri dari beberapa tahapan yang saling berurutan. Adanya kerangka berpikir bertujuan agar tahapan dalam mengerjakan penelitian lebih terstruktur dan terarah dalam menjawab rumusan masalah.

3.1 Diagram Alir Penelitian

Penelitian ini dilakukan dalam beberapa tahap yang dapat dilihat pada Gambar 3.1. Penelitian dimulai dengan melakukan tahap pendahuluan yaitu studi pustaka dan studi lapangan, tahap pengembangan model, tahap pengerjaan model, tahap pembahasan dan analisis, tahap kesimpulan saran.

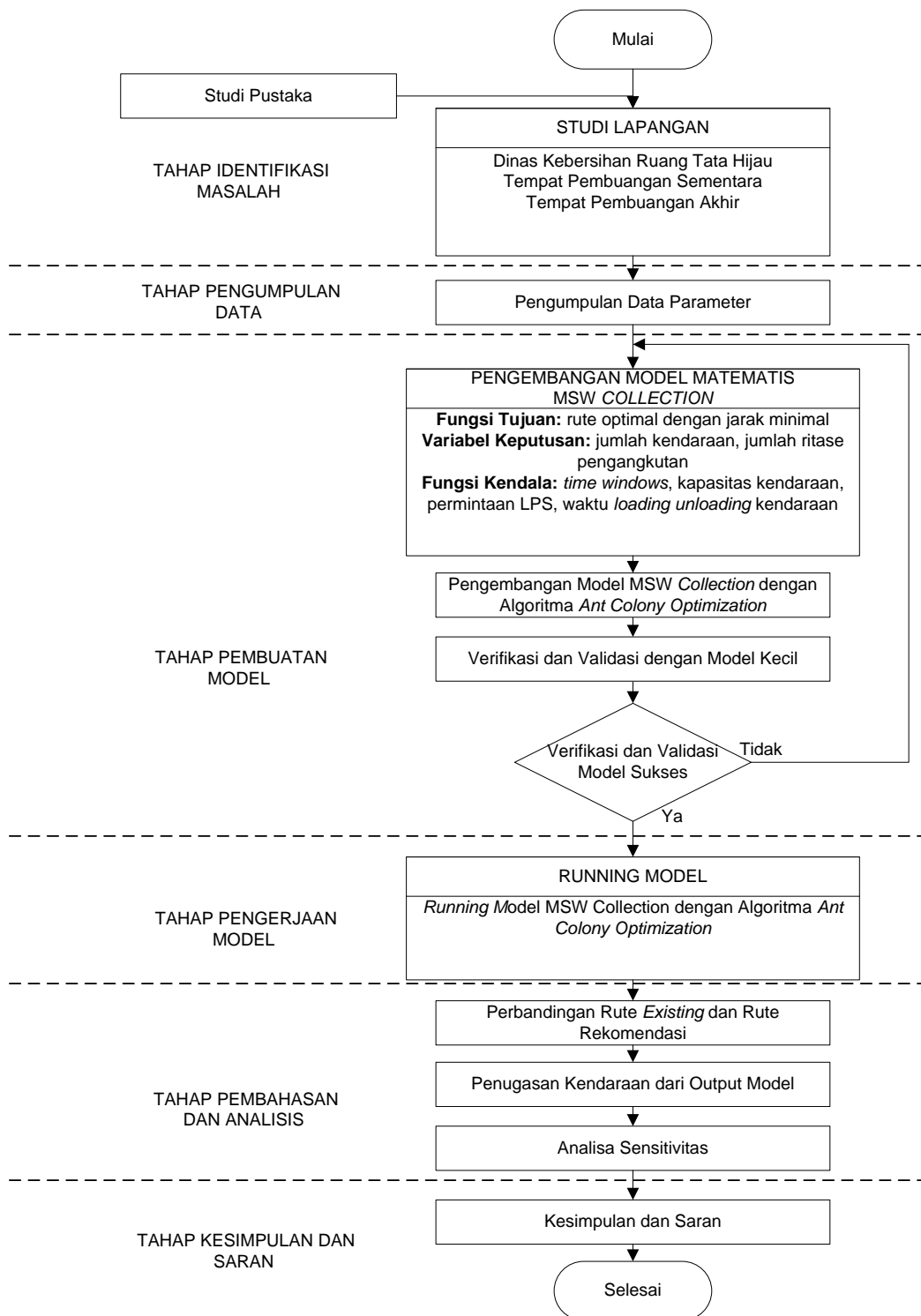
3.2 Tahap Pendahuluan

Pada tahap pendahuluan dilakukan pengumpulan studi pustaka dan studi lapangan. Studi pustaka berarti mengumpulkan sejumlah *literature* yang mendukung penelitian dari sumber buku, jurnal, artikel, dsb. Studi lapangan meliputi peninjauan objek yang diteliti yaitu, pengumpulan sampah di Kota Surabaya yang dikelola oleh DKRTH dengan fokus pada kendaraan truk *compactor* disekitar LPS dan 1 LPA. Studi pustaka dan studi lapangan sebenarnya akan banyak dilakukan secara bersamaan. Tujuannya adalah agar dapat mengetahui karakteristik sistem yang ada dan permasalahan yang ada sehingga dapat mengusulkan sebuah perbaikan.

3.3 Tahap Pengumpulan Data

Data yang digunakan didalam penelitian ini berasal dari data historis DKRTH, observasi langsung dan aplikasi *google maps*. Data yang dibutuhkan dalam penelitian ini adalah:

1. Data LPS yang dilayani oleh truk *compactor* seluruh titik yang ada di Kota Surabaya.



Gambar 3.1 Diagram Alir Penelitian

2. Jumlah *demand* masing-masing LPS dengan satuan kontainer.
3. Jumlah kendaraan dan kapasitas truk *existing* yang tersedia di depot.
4. *Time windows* untuk jam operasional depot, LPS dan LPA.
5. Data jarak tempuh antara LPS dengan depot, antar masing-masing LPS, antara LPS dengan LPA.
6. Data waktu tempuh antara LPS dengan depot, antar masing-masing LPS, antara LPS dengan LPA.
7. Waktu *loading* container dan *unloading* truck *compactor* di LPA.
8. Data rute dan penugasan kendaraan truk *compactor existing*.

3.4 Tahap pengembangan Model

Pada tahap pengembangan model terdapat 3 langkah penting didalamnya, yaitu: pengembangan model matematis sebagai model yang mempresentasikan kondisi sistem nyata dan dijadikan sebagai acuan untuk pembuatan model dengan pendekatan algoritma ACO, pengembangan model dengan algoritma ACO yang digunakan dalam pencarian nilai optimal pada penelitian ini dan terakhir model akan diverifikasi dan validasi agar layak dan mampu untuk menyelesaikan permasalahan pada penelitian ini. Sebelum melakukan pengembangan model matematis maupun model algoritma ACO, model konseptual akan dibuat sebagai dasar pembuatan model lainnya dan untuk menginformasikan kondisi sistem nyata sehingga lebih mudah untuk dipahami.

3.4.1 Pengembangan Model Matematis

Model matematis yang akan dikembangkan dalam penelitian ini merupakan model *mix integer linear programming*. Fungsi tujuan yang digunakan dalam penelitian ini adalah untuk menemukan rute optimal dengan jarak minimum, dengan variabel respon jarak seluruh rute yang dihasilkan.

Variabel keputusan dalam penelitian ini adalah sebagai berikut:

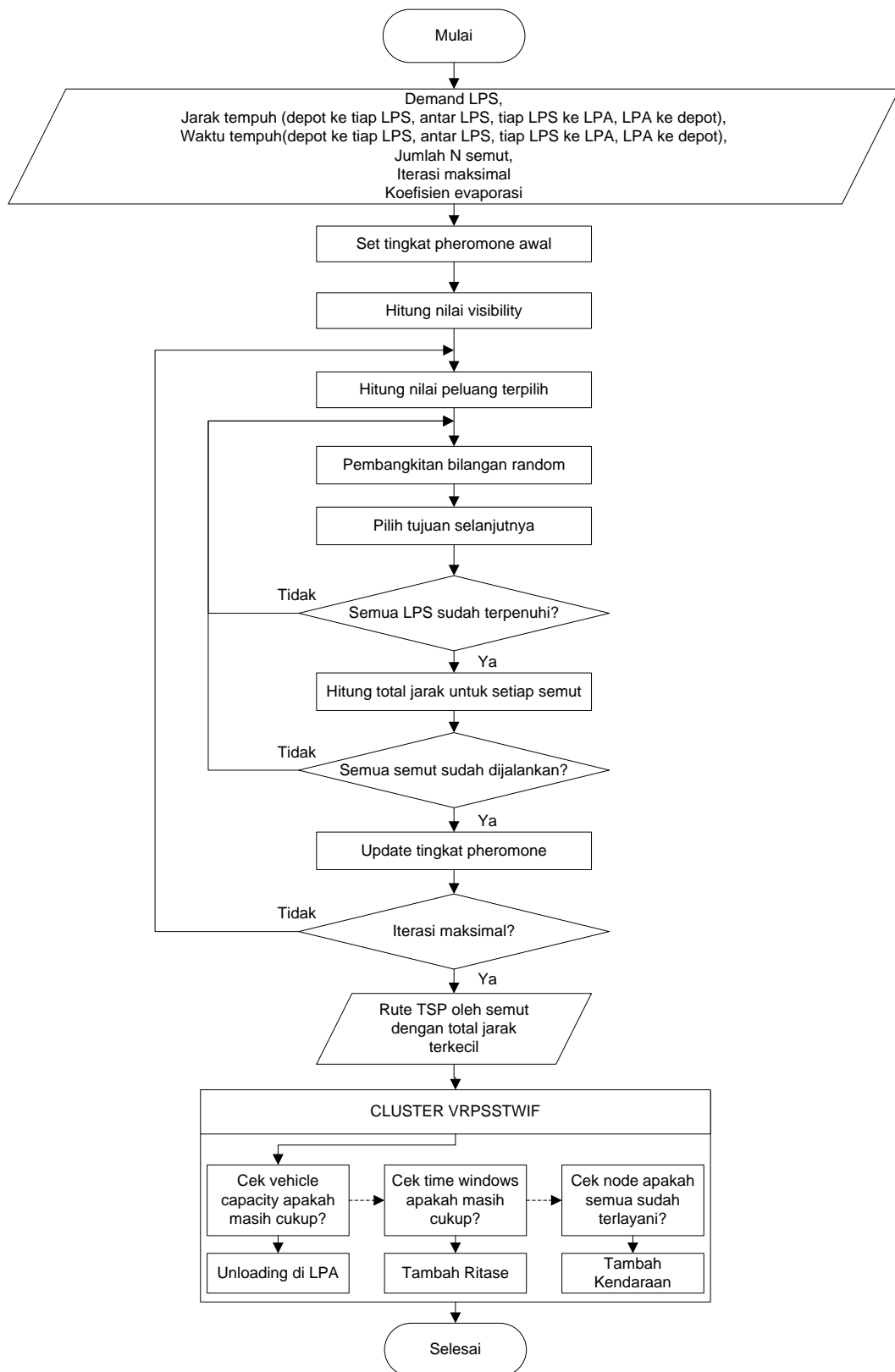
1. Berapa jumlah kendaraan yang digunakan untuk melayani permintaan?
2. Bagaimana rute dan penugasan kendaraan yang direkomendasikan?
3. Bagaimana pembagian *demand* pada setiap LPS nya?
4. Kapan waktu mulai kendaraan harus berangkat?

Terdapat beberapa *constraints* yang harus dipenuhi kedalam model yang dikembangkan, diantaranya berikut merupakan *global constraints* yang harus dipenuhi :

1. Semua kendaraan harus melakukan kegiatan operasional dalam *time window* yang sudah disesuaikan untuk masing-masing LPS dan LPA agar kendaraan dapat kembali pada jam operasional DKRTH yaitu pukul $a - 15.00$ WIB, a sendiri merupakan awal mulai kendaraan bisa berangkat dari depot yang nilainya nanti akan ditentukan dalam rute rekomendasi
2. Permintaan yang diangkut oleh kendaraan tidak boleh melebihi kapasitas. Kendaraan memiliki kapasitas 35 kontainer.
3. Kendaraan dapat memenuhi kebutuhan masing-masing kontainer yang ada di LPS-LPS yang sudah ditentukan sebelumnya.
4. Dalam memenuhi kebutuhan LPS, maka masing-masing LPS dapat dikunjungi oleh lebih dari 1 kali dan dengan kendaraan yang sama atau berbeda dalam waktu yang berbeda.
5. Waktu *loading* kendaraan adalah konstan 1 menit untuk 1 kontainer. Waktu *unloading* kendaraan adalah konstan 1 jam.
6. Semua kendaraan berangkat dan kembali ke depot dalam keadaan kosong.
7. Kendaraan yang sudah muatan penuh harus menuju LPA untuk mengkosongkan muatan.
8. Kendaraan tidak akan kembali ke depot apabila masih memungkinkan untuk mengambil sampah di LPS, mengantarkan sampah ke LPA dan kembali ke depot.
9. Sebelum kembali ke depot kendaraan harus melakukan pengisian BBM tepat setelah melakukan proses *unloading* pada kunjungan LPA yang terakhir.

3.4.2 Pengembangan Model dengan Algoritma *Ant Colony Optimization*

Pada Gambar 3.2 adalah algoritma ACO yang akan digunakan dalam penelitian ini. Terdapat 2 langkah utama yaitu: pembangkitan rute TSP dan pembagian menjadi beberapa sub rute yang disesuaikan dengan *constraints* yang ada, atau dalam kata lain penelitian ini menggunakan “*route first cluster second*”.



Gambar 3.2 Algoritma ACO

3.4.3 Verifikasi dan Validasi Model

Validasi merupakan proses penentuan apakah model yang dibuat benar-benar merupakan representasi dari sistem nyata yang dimodelkan. Perlu diperhatikan apakah semua elemen dan aktivitas yang ada pada sistem nyata sudah termodelkan pada model.

Pada verifikasi model dilakukan *running* model dengan menggunakan sampel kecil, apakah algoritma yang digunakan sudah dapat berjalan dengan baik sehingga pencarian rute optimal dapat dijalankan. Dari hasil proses *running* kemudian disesuaikan dengan hasil perhitungan manual. Apabila verifikasi dan validasi model sudah terpenuhi maka selanjutnya model siap digunakan untuk sampel yang lebih besar.

3.5 Tahap Pengerjaan Model

Setelah mendapatkan algoritma ACO untuk pencarian nilai optimal, maka langkah selanjutnya ada pada tahap pengerjaan model. Pada tahap pengerjaan model terdapat 1 langkah utama yaitu *running* model. *Running* model akan dilakukan sebanyak 10 replikasi dengan uji coba dengan berbagai parameter untuk mendapatkan nilai paling mendekati optimal. Selanjutnya parameter dengan nilai terbaik akan dilakukan 10 replikasi kembali untuk mendapatkan rute rekomendasi pada penelitian ini. *Running* model menggunakan bantuan *software* Matlab untuk membantu proses pencarian rute dengan jarak optimal

dan penugasan kendaraan.. Hasil dari *running* model adalah sejumlah rute dengan jarak dan jumlah kendaraan minimal. Pada langkah penugasan dilakukan pembagian kendaraan untuk masing-masing rute yang sesuai dengan jam operasional.

3.6 Tahap Pembahasan dan Analisis

Pada tahap ini sudah dilakukan *running* model sehingga hasil juga sudah didapatkan. Langkah awal pada tahap ini adalah membandingkan rute *existing* dan rute rekomendasi sementara sehingga mendapatkan beberapa pertimbangan tambahan dalam melakukan langkah selanjutnya yaitu penugasan kendaraan. Perbandingan akan dilakukan dengan melihat fungsi tujuan, waktu komputasi dan

jumlah kendaraan yang dibutuhkan. Langkah selanjutnya adalah melakukan analisa sensitivitas model dengan adanya perubahan jumlah permintaan pengangkutan di LPS yang sudah ada dan perubahan apabila jumlah LPS ditambah, bagaimana pengaruh jumlah kendaraan yang dibutuhkan.

3.7 Tahap Kesimpulan Saran

Pada tahap terakhir dalam penelitian ini ada penarikan kesimpulan dan saran penelitian. Kesimpulan dengan adanya implikasi praktis terhadap analisis yang telah dilakukan pada tahap sebelumnya yang harapannya dapat menjawab rumusan masalah dengan baik , serta memberikan saran perbaikan yang berguna untuk mengembangkan penelitian ini selanjutnya agar lebih baik.

(Halaman ini sengaja dikosongkan)

BAB IV

PENGUMPULAN DATA

Pada bab ini akan ditampilkan hasil pengumpulan data yang akan digunakan dalam penelitian ini. Data tersebut berkaitan dengan truk *compactor*, jarak tempuh, waktu tempuh dan kondisi penugasan kendaraan *existing*.

4.1 *Municipal Solid Waste Collection di Kota Surabaya*

MSW *collection* oleh kendaraan truk *compactor* di Kota Surabaya berawal dari kendaraan yang berangkat dari lokasi depot yaitu Kantor DKRTH dan dilanjutkan dengan proses *loading* pada LPS-LPS. Truk *compactor* memiliki kapasitas yang diasumsikan sebesar 35 kontainer, ketika kapasitas sudah penuh kendaraan akan melakukan proses *unloading* pada LPA. Proses *loading* diasumsikan secara konstan membutuhkan waktu 1 menit per kontainer dan proses *unloading* diasumsikan secara konstan membutuhkan waktu 60 menit. Kedua waktu tersebut didapatkan dari *interview* pengemudi truk *compactor* dan hasil observasi langsung di lapangan. Jam waktu keberangkatan kendaraan disesuaikan dengan LPS pertama yang akan dikunjungi oleh kendaraan. Kendaraan dapat melakukan proses pengangkutan kembali ke LPS-LPS setelah melakukan proses *unloading* dengan syarat kendaraan harus kembali ke depot sebelum atau tepat pada pukul 15.00 WIB.

Permasalahan rute pengangkutan sampah yang dimiliki DKRTH Kota Surabaya saat ini adalah jumlah kendaraan yang belum optimal untuk memenuhi permintaan pengangkutan di keseluruhan LPS yang ada di Surabaya, terlebih apabila pada saat ada kendaraan yang rusak. Di sisi lain permintaan pengangkutan sampah di masing-masing LPS semakin meningkat sejalan dengan meningkatnya pertumbuhan penduduk di Surabaya, sehingga perlu adanya optimasi serta alat bantu yang digunakan sebagai pertimbangan dalam penentuan rute secara baik dan cepat.

4.2 Pengumpulan Data

Data yang digunakan didalam penelitian ini diperoleh dari Dinas Kebersihan dan Ruang Terbuka Hijau Kota Surabaya dan dari hasil amatan jarak dan waktu langsung dengan bantuan *Google Maps*. Data yang didapatkan dari *Google Maps* diambil pada waktu jam operasional dan kedepannya akan diasumsikan bernilai konstan.

4.2.1 Data Kendaraan Truk *Compactor*

Pada data kendaraan penelitian ini berfokus pada jenis kendaraan truk *compactor*. Data truk *compactor* merupakan data yang memaparkan jumlah kendaraan yang dimiliki oleh DKRTH Kota Surabaya yang berada di depot. Truk *compactor* memiliki kapasitas yang sama $10 m^3$ atau dapat diasumsikan memiliki kapasitas sebesar 35 dalam satuan kontainer sampah yang ada pada setiap LPS nya. Data ini selanjutnya akan digunakan untuk melihat utilitas kendaraan yang dimiliki oleh seluruh kendaraan yang akan melayani beberapa LPS yang tersebar di Kota Surabaya. Data ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Data Kendaraan Truk *Compactor*

No	NO POLISI	TIPE KENDARAAN	No	NO POLISI	TIPE KENDARAAN
1	L9384NP	HINO/FG8JJKB GGJ (FG235JJ)	11	L9455NP	HINO/FG8JJKB GGJ (FG235JJ)
2	L9385NP	HINO/FG8JJKB GGJ (FG235JJ)	12	L9561NP	HINO/FG8JJKB GGJ (FG235JJ)
3	L9386NP	HINO/FG8JJKB GGJ (FG235JJ)	13	L9553NP	HINO/FG8JJKB GGJ (FG235JJ)
4	L9388NP	HINO/FG8JJKB GGJ (FG235JJ)	14	L9554NP	HINO/FG8JJKB GGJ (FG235JJ)
5	L9389NP	HINO/FG8JJKB GGJ (FG235JJ)	15	L9555NP	HINO/FG8JJKB GGJ (FG235JJ)
6	L9448NP	HINO/FG8JJKB GGJ (FG235JJ)	16	L9556NP	HINO/FG8JJKB GGJ (FG235JJ)
7	L9451NP	HINO/FG8JJKB GGJ (FG235JJ)	17	L9557NP	HINO/FG8JJKB GGJ (FG235JJ)
8	L9452NP	HINO/FG8JJKB GGJ (FG235JJ)	18	L9560NP	HINO/FG8JJKB GGJ (FG235JJ)
9	L9453NP	HINO/FG8JJKB GGJ (FG235JJ)	19	L9627NP	HINO/FG8JJKB GGJ (FG235JJ)
10	L9454NP	HINO/FG8JJKB GGJ (FG235JJ)	20	B9016SJA	HINO/FG8JJKB GGJ (FG235JJ)

Tabel 4.1 Data Kendaraan Truk *Compactor* (Lanjutan)

No	NO POLISI	TIPE KENDARAAN	No	NO POLISI	TIPE KENDARAAN
21	L9688NP	HINO/FG8JJKB GGJ (FG235JJ)	33	L9704NP	HINO/FG8JJKB GGJ (FG235JJ)
22	L9689NP	HINO/FG8JJKB GGJ (FG235JJ)	34	L9705NP	HINO/FG8JJKB GGJ (FG235JJ)
23	L9690NP	HINO/FG8JJKB GGJ (FG235JJ)	35	L9706NP	HINO/FG8JJKB GGJ (FG235JJ)
24	L9691NP	HINO/FG8JJKB GGJ (FG235JJ)	36	L9715NP	HINO/FG8JJKB GGJ (FG235JJ)
25	L9692NP	HINO/FG8JJKB GGJ (FG235JJ)	37	L9716NP	HINO/FG8JJKB GGJ (FG235JJ)
26	L9694NP	HINO/FG8JJKB GGJ (FG235JJ)	38	L9717NP	HINO/FG8JJKB GGJ (FG235JJ)
27	L9695NP	HINO/FG8JJKB GGJ (FG235JJ)	39	L9816NP	HINO/FG8JJKB GGJ (FG235JJ)
28	L9697NP	HINO/FG8JJKB GGJ (FG235JJ)	40	L9817NP	HINO/FG8JJKB GGJ (FG235JJ)
29	L9698NP	HINO/FG8JJKB GGJ (FG235JJ)	41	L9818NP	HINO/FG8JJKB GGJ (FG235JJ)
30	L9699NP	HINO/FG8JJKB GGJ (FG235JJ)	42	L9820NP	HINO/FG8JJKB GGJ (FG235JJ)
31	L9701NP	HINO/FG8JJKB GGJ (FG235JJ)	43	L9841NP	HINO/FG8JJKB GGJ (FG235JJ)
32	L9703NP	HINO/FG8JJKB GGJ (FG235JJ)	44	L9843NP	HINO/FG8JJKB GGJ (FG235JJ)

4.2.2 Data Lokasi *Node* yang Dikunjungi Truk *Compactor*

DKRTH Surabaya saat ini memiliki 1 lokasi depot truk *compactor* (*node* 1) sebagai tempat memulai pengambilan sampah dan tempat parkir setelah melakukan pengambilan sampah. Terdapat 80 LPS (*node* 2-81) yang tersebar di Wilayah Surabaya dimana pengambilan sampahnya dilayani oleh truk *compactor*. LPA terletak di Daerah Benowo (*node* 82) yang harus dikunjungi apabila kapasitas kendaraan sudah penuh. Data tersebut dapat dilihat pada Tabel 4.2, sedangkan alamat lengkap setiap nomor dapat dilihat pada Lampiran 1.

Tabel 4.2 Data Lokasi *Node* Kunjungan

NO	Lokasi	NO	Lokasi	NO	Lokasi
1	Depot DKRTH	5	Jl. Embong Malang	9	LPS. Bratang
2	LPS. Srikana	6	Jl. Karet	10	LPS. Kaliwaron
3	Jl. Pandegiling	7	JMP	11	LPS. Semut Kali
4	Jl. Urip Sumoharjo	8	Kemayoran Baru	12	LPS. Ngagel

Tabel 4.2 Data Lokasi *Node* Kunjungan (Lanjutan)

NO	Lokasi LPS	NO	Lokasi LPS	NO	Lokasi LPS
13	LPS. Gayung Pring	37	Taman Lansia	61	LPS Prapen Panjang Jiwo
14	LPS. Joyoboyo	38	Taman Prestasi	62	LPS Prapen
15	LPS. Bok Tong	39	Taman Joging Trek Pusura	63	Jl. Mulyorejo
16	Taman Flora	40	PMI	64	LPS Mojo Arum
17	Jl. Kalisumo	41	Balai Pemuda	65	LPS Bukit Barisan
18	LPS. Jemur Wonosari	42	DPRD	66	LPS. Babadan UNESA
19	LPS. Kertopaten	43	Stasiun Gubeng	67	LPS. Siwalan Kerto
20	LPS Nyamplungan	44	PDAM	68	LPS. Legundi
21	LPS. Candipuro	45	Jl. Dharmawangsa	69	LPS Kedung Anyar
22	LPS Pasar Genteng	46	RS. Bhayangkara	70	LPS. Pasar Kembang
23	LPS. Kayoon	47	LPS. Merapi	71	LPS. Karang Gayam
24	Taman Apsari	48	LPS. Simpang Dukuh	72	LPS. Wonokusumo Kidul
25	Siola	49	LPS. Kendalsari	73	LPS. Kalibokor
26	LPS. Tambak Rejo	50	LPS. Kebun Bibit Wonorejo	74	LPS. Sutorejo
27	Pasar Keputran	51	LPS. Peneleh	75	Lapangan Dwikora
28	Saluran Dinoyo	52	LPS. Demak Selatan	76	Jatisrono
29	Saluran Bengawan	53	LPS. Pandegiling	77	Endroso
30	Pasar Bunga Kayoon	54	LPS. Sulung	78	Tenggumung
31	Saluran Gadukan	55	LPS Endroso	79	Kertopaten
32	Saluran Mungsing	56	LPS Kangean	80	Kali Tebu
33	LPS Kejawen Putih Tambak	57	LPS Jeruk	81	Nyamplungan
34	LPS. Ketampon	58	LPS. Prapen 88	82	LPS Benowo
35	Taman Bungkul	59	LPS Prapen DKK		
36	Taman Sulawesi	60	LPS Raya Prapen		

4.2.3 Data Demand LPS

Setiap lokasi LPS memiliki jumlah *demand* berbeda-beda yang harus dilayani oleh kendaraan yang mengunjungi lokasi tersebut dengan batasan kapasitas kendaraan dan juga waktu pelayanan depot. Kendaraan truk *compactor* memiliki kapasitas yang diasumsikan sebesar 35 kontainer sampah yang ada pada setiap LPS. Setiap kendaraan harus kembali pada pukul 15.00 WIB di Depot DKRTH Surabaya. Jumlah *demand* di beberapa LPS lebih besar dari kapasitas kendaraan, sehingga

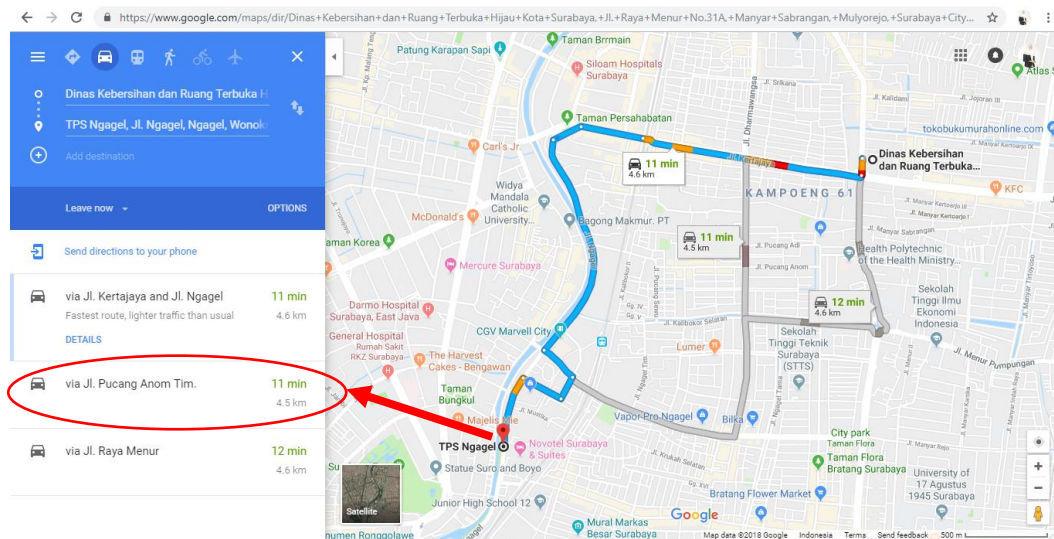
untuk setiap LPS dapat dikunjungi oleh lebih dari satu kali. Data ini selanjutnya digunakan untuk melihat penyebaran *demand* di Kota Surabaya yang harus dilayani setiap harinya oleh truk *compactor*. Data tersebut dapat dilihat pada Tabel 4.3.

Tabel 4.3 Data *Demand* LPS

NO	<i>Demand / Kontainer</i>	NO	<i>Demand / Kontainer</i>	NO	<i>Demand / Kontainer</i>	NO	<i>Demand / Kontainer</i>
2	140	22	12	42	2	62	20
3	13	23	55	43	2	63	22
4	10	24	4	44	3	64	35
5	2	25	6	45	2	65	140
6	1	26	90	46	15	66	70
7	2	27	40	47	24	67	70
8	1	28	20	48	23	68	70
9	140	29	15	49	33	69	80
10	70	30	40	50	40	70	62
11	70	31	7	51	25	71	70
12	60	32	4	52	110	72	70
13	20	33	30	53	140	73	70
14	108	34	50	54	35	74	80
15	23	35	12	55	15	75	6
16	15	36	1	56	35	76	6
17	1	37	2	57	20	77	15
18	70	38	5	58	20	78	1
19	30	39	10	59	20	79	30
20	20	40	4	60	20	80	18
21	20	41	2	61	20	81	20

4.2.4 Data Jarak Tempuh Antar Lokasi

Model yang digunakan pada penelitian ini merupakan pengembangan dari VRP, sehingga diperlukan beberapa pertimbangan diantaranya adalah jarak tempuh antar lokasi. Tujuan dari model yang ada pada penelitian ini adalah untuk meminimalisir total jarak yang ditempuh untuk semua kendaraan dalam suatu jam operasional. Jarak tempuh antar lokasi yaitu: depot ke setiap LPS, antar lokasi LPS, setiap lokasi LPS ke LPA dan dari LPA ke depot didapatkan dari aplikasi Google Maps dengan jarak minimum yang akan dipilih dari beberapa opsi yang disediakan. Satuan yang digunakan adalah kilometer. Jarak tempuh antar lokasi tidak berbanding lurus dengan waktu tempuhnya karena beberapa faktor yang terjadi di jalan raya misalnya kemacetan, jalan tol, dsb.



Gambar 4.1 Contoh Pengambilan Data Melalui Google Maps

Pada Gambar 4.1 adalah contoh pengambilan data untuk jarak tempuh dari *node* depot (1) menuju ke *node* LPA Ngagel (12). Jarak tempuh dari *node* 1 menuju *node* 12 adalah 4,5 km yang didapatkan dari nilai minimum dari beberapa pilihan alternatif jalan yang ada. Data jarak tempuh antar *node* lokasi dapat dilihat pada Lampiran 2.

4.2.5 Data Waktu Tempuh Antar Lokasi

Jarak tempuh antar lokasi tidak berbanding lurus dengan waktu tempuhnya sehingga perlu dilakukan pengumpulan data terkait waktu tempuh antar lokasi yaitu: depot ke setiap LPS, antar lokasi LPS, setiap lokasi LPS ke LPA dan dari LPA ke depot didapatkan dari aplikasi Google Maps dengan jarak minimum yang akan dipilih dari beberapa opsi yang disediakan. Bukan berarti waktu tercepat yang akan dipilih melainkan waktu yang didapatkan dari mengikuti jarak minimum kendaraan. Pada Satuan yang digunakan adalah menit. Pada Gambar 4.1 adalah contoh pengambilan data untuk jarak tempuh dari *node* depot (1) menuju ke *node* LPA Ngagel (12). Jarak tempuh dari *node* 1 menuju *node* 12 adalah 11 menit yang didapatkan dari nilai waktu tempuh untuk jarak minimum dari beberapa pilihan alternatif jalan yang ada. Data waktu tempuh antar lokasi dapat dilihat pada Lampiran 3.

(Halaman ini sengaja dikosongkan)

. BAB V







PENGEMBANGAN MODEL

Pada bab ini akan dijabarkan proses pengembangan model MSW *collection* di Kota Surabaya mulai dari pembuatan model konseptual, pengembangan model matematis maupun pengembangan algoritma ACO. Selain itu, pada bab ini akan dilakukan tahap verifikasi dan validasi untuk kedua model pengembangan.

5.1 Pengembangan Model Konseptual VRPSSTWIF untuk MSW *Collection* di Kota Surabaya

Model menurut Kamus Besar Bahasa Indonesia (KBBI) adalah pola (contoh, acuan, ragam dan sebagainya) dari sesuatu yang akan dibuat atau dihasilkan. Sedangkan Konsep menurut KBBI adalah gambaran mental dari objek, proses atau apapun yang ada diluar Bahasa, yang digunakan oleh akal budi untuk memahami hal-hal lain. Model konseptual pada penelitian ini akan digambarkan secara grafis untuk memahami proses pada sistem nyata sehingga akan lebih mudah dipahami. Simbol-simbol yang akan digunakan pada penelitian ini dapat dilihat pada Tabel 5.1, dan model konseptual dapat dilihat pada Gambar 5.1.

Tabel 5.1 Simbol Model Konseptual

NO	SIMBOL	NOTASI	KETERANGAN	NO	SIMBOL	NOTASI	KETERANGAN
1		{0,0'}	Depot di DKRTH	4		{1,2,...,l}	Truk <i>Compactor</i>
2		{1,2,...,n}	LPS	5		{1,2,...,q}	Kontainer Sampah
3		{m}	LPA	6		{g}	Pom Bensin



Gambar 5.1 Model Konseptual MSW *Collection* di Kota Surabaya

Setiap kendaraan akan berangkat dan kembali pada depot pada jam operasional $(a, b) = (a, 900)$. a disini fleksibel disesuaikan dengan rute awal yang akan dituju. Kendaraan harus kembali pada menit ke-900. Kendaraan 1 yang berwarna abu-abu melakukan pengambilan sampah pada LPS 1 dilanjutkan pada LPS 2. Pada Gambar 5.1 diasumsikan paling banyak kendaraan dapat mengangkut sebanyak 2 kontainer, oleh karena itu pada kendaraan 1 setelah mengangkut 1 kontainer dari LPS 1 dan 2 akan menuju ke LPA untuk melakukan proses *unloading*. Kendaraan 1 melakukan pengangkutan lagi pada LPS 2 sebanyak 2 kontainer dan melakukan proses *unloading* lagi, mengisi BBM dan kembali ke depot sebelum atau pas pada menit ke-900. Berarti LPS 2 dapat dilayani 2 kali dengan kendaraan 1.

Kendaraan 2 yang berwarna hijau mengawali perjalanan dengan mengambil sampah pada LPS 2 sebanyak 1 kontainer. Hal ini menyatakan bahwa pada LPS 2 dapat dilayani secara *split service*, yang artinya dapat dilayani lebih dari satu kali dengan kendaraan yang sama ataupun dengan kendaraan yang berbeda. Perjalanan kedua kendaraan 2 mengambil sampah pada LPS 3, namun kontainer yang dapat diambil hanya 1 karena muatan sudah tidak cukup dan akan dilakukan proses *unloading* pada LPA. Setelah sampai LPA kendaraan 2 tidak melakukan proses pengangkutan lagi. Hal ini mungkin terjadi karena waktu yang dibutuhkan untuk sampai depot kembali sudah tidak bisa sesuai dengan jam

operasional atau kembali ke depot melebihi menit ke-900. Sehingga dibutuhkan kendaraan ke 3 untuk mengangkut container pada LPS 3.

5.2 Pengembangan Model Matematis VRPSSTWIF untuk MSW Collection di Kota Surabaya

Model yang akan digunakan untuk MSW *collection* di Kota Surabaya adalah VRPSSTWIF. Model matematis yang digunakan mengacu pada penelitian Buhrkal, *et al.*, (2012). Fungsi tujuan pada model matematis sebelumnya terdapat pada Persamaan 2.1, sedangkan untuk *constraints* terdapat pada Persamaan 2.2 sampai 2.12. Perlu dilakukan pengembangan model dari yang sebelumnya agar dapat sesuai dengan sistem MSW *collection* di Kota Surabaya.

5.2.1 Skenario Pengembangan Model Matematis

Penelitian Buhrkal, *et al.*, (2012) perlu dimodifikasi agar sesuai dengan tujuan minimasi jarak total dari proses MSW *collection* di Kota Surabaya. Penelitian ini menggunakan referensi fungsi tujuan dari Abdulkader, *et al.*, (2015) dalam membuat notasi c_{ij} yang dijadikan sebagai jarak tempuh *arc* (i, j) . Pada penelitian tersebut memiliki fungsi tujuan meminimasi total biaya untuk melintasi *arc* (i, j) yang dilewati oleh seluruh kendaraan.

Terkait pemodelan proses *loading* yang dilakukan disetiap LPS dan *unloading* untuk setiap LPA, dari model acuan perlu ditambahkan parameter baru terkait *service time*, karena proses *loading* dan *unloading* pada penelitian ini berbeda, sehingga nilai *service time* tidak berlaku untuk keseluruhan. s_m dapat mewakili waktu proses *unloading* pada *node m* atau LPS. Terdapat 1 *node gas station* yang harus dikunjungi sesaat sebelum kembali ke depot untuk mengakhiri sebuah rute perjalanan dalam waktu pelayanan s_g .

Pada penelitian ini menggunakan *split service*, dimana pada untuk setiap *node* nya dapat dilayani lebih dari satu kali dengan menggunakan kendaraan yang sama ataupun berbeda, sehingga pada Persamaan 2.4 penelitian Buhrkal, *et al.*, (2012) yang menyebutkan bahwa semua konsumen harus dilayani oleh kendaraan tepat satu kali perlu dilakukan modifikasi. Penelitian Wilck IV & Cavalier (2012)

memberikan batasan bahwa setiap *customer* dikunjungi paling tidak satu kali yang artinya akan dilakukan *split demand* jika dibutuhkan agar dapat memenuhi kebutuhan *customer* dengan tidak melanggar batasan yang lain.

5.2.2 Notasi Model Matematis VRPSSTWIF

Indeks yang digunakan dalam pengembangan model VRPSSTWIF adalah sebagai berikut:

i, j indeks *node* dimana $i, j = 0, 1, \dots, n$

l indeks kendaraan dimana $l = 1, 2, \dots, k$

Himpunan dalam pengembangan model VRPSSTWIF adalah sebagai berikut:

V Himpunan *nodes* dimana $V = V^d \cup V^c \cup V^f \cup V^{gs}$

V^d Himpunan depot dimana terdapat sebuah depot sebagai titik mulai dan akhir perjalanan $V^d = \{0, 0'\}$

V^c Himpunan *customers* atau LPS dimana $V^c = \{1, 2, \dots, n\}$

V^f Himpunan *intermediate facility* dimana $V^f = \{n + 1\} = \{m\}$

V^{gs} Himpunan *gas station* dimana $V^{gs} = \{n + 2\} = \{g\}$

A Himpunan *arcs* dimana $A = \{(i, j) | i, j \in V, i \neq j\}$

K Himpunan kendaraan dimana $K = \{1, 2, \dots, l\}$

Q Himpunan *customers demand* dimana $Q = \{1, 2, \dots, q\}$

Parameter yang digunakan dalam pengembangan model VRPSSTWIF adalah sebagai berikut:

t_{ij} Waktu yang dibutuhkan dari *node* i ke j

s_i Waktu pelayanan pada *node* i (*loading time*)

s_m Waktu pelayanan pada *node* m (*unloading time*)

s_g Waktu pelayanan pada *node* g (pengisian bensin)

a_i Waktu paling awal *node* i dapat dilayani

b_i Waktu paling akhir *node* i dapat dilayani

c_{ij} jarak tempuh *arc* (i, j)

C Kapasitas kendaraan

Variabel keputusannya model VRPSSTWIF adalah:

X_{ijl}	$= \begin{cases} 1 & \text{jika dan hanya jika kendaraan } l \in K \text{ menggunakan arc } (i,j) \in A \\ 0 & \text{jika tidak} \end{cases}$
Y_{il}	$= \begin{cases} 1 & \text{jika dan hanya jika node } i \text{ dikunjungi kendaraan } l \in K \\ 0 & \text{jika tidak} \end{cases}$
d_{il}	Akumulatif <i>demand</i> (kontainer) yang ada pada <i>node</i> $i \in V$ oleh kendaraan $l \in K$
w_{il}	Waktu mulai pelayanan pada <i>node</i> $i \in V$ oleh kendaraan $l \in K$
q_{il}	Jumlah <i>demand</i> kontainer yang diambil dari <i>node</i> i oleh kendaraan $l \in K$
U_{ik}	Variabel bebas digunakan dalam <i>constraint</i> eliminasi <i>subtour</i>

5.2.3 Model Matematis VRPSSTWIF

Pengembangan model matematis VRPSSTWIF adalah sebagai berikut:

Fungsi tujuan:

Fungsi tujuan (5.1) diadopsi dari penelitian Abdulkader, *et al.*, (2015).

$$\text{Min} = \sum_{(i,j) \in A} c_{ij} \sum_{l \in K} X_{ijl} \quad (5.1)$$

Fungsi Tujuan 5.1 adalah persamaan untuk meminimasi jarak total yang ditempuh oleh semua kendaraan.

c_{ij} yang dimaksud pada persamaan ini tidak sama dengan model sebelumnya yang artinya adalah biaya perjalanan dari *node* i ke *node* j , melainkan merupakan jarak dari *node* i ke *node* j . X_{ijl} jika dan hanya jika kendaraan l mengunjungi segera *node* j setelah *node* i .

Fungsi kendala:

Capacity constraints

Capacity constraints 5.2, 5.5 dan 5.6 diadopsi dari penelitian Buhrkal, *et al.*, (2012). *Capacity constraint* 5.8 dan 5.9 dikembangkan dari penelitian penelitian Buhrkal, *et al.*, (2012) karena harus disesuaikan terkait indeks pada LPA dan *gas station*. *Capacity constraints* 5.3, 5.4, 5.7 dikembangkan dari penelitian Wilck IV & Cavalier (2012), pengembangan ini perlu dilakukan karena perbedaan indeks dan variabel yang harus disesuaikan lagi agar sesuai dengan kasus MSW *collection* di Kota Surabaya.

$$\sum_{i \in (0,0')} d_{il} = 0 \quad \forall l \in K \quad (5.2)$$

Kapasitas semua kendaraan yang keluar dan masuk ke depot memiliki muatan 0.

$$Q_i Y_{il} > q_{il} \quad \forall i \in V_c, l \in K \quad (5.3)$$

$$\sum_{i \in V^c} X_{ijl} = Y_{il} \quad \forall i \in V_c, l \in K \quad (5.4)$$

Kedua *constraints* menjamin variabel binary akan bernilai positif apabila material diambil pada *node* i dengan menggunakan kendaraan l .

$$d_{il} + q_{il} \leq d_{jl} + (1 - x_{ijl})M \quad \forall (i,j) \in V_c, j \in l \in V, l \in K \quad (5.5)$$

Akumulasi *demand* untuk setiap *node customer*.

$$d_{il} \leq C \quad \forall i \in V_c, l \in K \quad (5.6)$$

Akumulasi *demand* yang diambil oleh suatu kendaraan tidak dapat melebihi kapasitas kendaraan.

$$\sum_{l \in K} q_{il} = Q_i \quad \forall i \in V_c, l \in K \quad (5.7)$$

Jumlah *demand* yang diambil oleh kendaraan $l \in K$ pada suatu *node customer* sama dengan *demand* yang dimiliki oleh *node customer* $\forall i \in V_c$.

$$\sum_{i \in (m)} d_{il} = 0 \quad \forall l \in K \quad (5.8)$$

Kapasitas semua kendaraan yang keluar dari LPA memiliki muatan 0.

$$\sum_{i \in V} Q_i = Q_m \quad \forall i \in V_c \quad (5.9)$$

Jumlah seluruh *demand* pada *node customer* sama dengan *demand* pada LPA.

$$\sum_{i \in (g,g')} d_{il} = 0 \quad \forall l \in K \quad (5.10)$$

Kapasitas semua kendaraan yang keluar dan masuk ke *gas station* memiliki muatan 0.

Time windows constraints

Capacity constraints 5.13 dan 5.14 diadopsi dari penelitian Buhrkal, *et al.*, (2012).

Capacity constraint 5.11, 5.15 dan 5.9 dikembangkan dari penelitian penelitian Buhrkal, *et al.*, (2012) karena harus disesuaikan terkait indeks *node* dan karena adanya proses *unloading*. *Capacity constraints* 5.12 dibuat pada penelitian ini untuk menjamin waktu keberangkatan kendaraan fleksibel sesuai dengan rute awal yang dituju setelah kendaraan meninggalkan depot.

$$a_{0l} \leq w_{0l} \leq b_0 \quad 0 \in V, l \in K \quad (5.11)$$

Waktu mulai pelayanan pada depot adalah setelah atau pada saat pukul a_0 dan berakhir sebelum atau pada saat b_0 .

$$a_{0l} = b_j - t_{0jl} \quad 0 \in V, \forall j \in V_c \quad (5.12)$$

Waktu awal keberangkatan kendaraan dari depot adalah waktu jam buka *node* yang akan dituju dikurangi dengan waktu tempuh depot ke *node* tersebut.

$$a_i \leq w_{il} \leq b_i \quad \forall i \in \frac{V}{v^d} \quad (5.13)$$

Waktu mulai pelayanan pada *node customer* adalah setelah atau pada saat pukul a_i dan berakhir sebelum atau pada saat b_i .

$$w_{il} + s_{il} + t_{ij} \leq w_{il} + (1 - x_{ijl})M \quad \forall i \in V_c, l \in K \quad (5.14)$$

Akumulasi waktu pelayanan untuk setiap *node customer*.

$$w_{il} + s_m + t_{ij} \leq w_{il} + (1 - x_{ijl})M \quad \forall (i, j) \in A, l \in K \quad (5.15)$$

Akumulasi waktu pelayanan untuk *node LPA*.

Decision variable constraints

Capacity constraints 5.16, 5.17, 5.26 dan 5.27 diadopsi dari penelitian Buhrkal, *et al.*, (2012). *Capacity constraints* 5.18 dikembangkan dari penelitian Buhrkal, *et al.*, (2012) karena disesuaikan bahwa setiap *arc* bisa dilalui lebih dari 1 kali. *Capacity constraints* 5.21, 5.28 dan 5.22 diadopsi dari penelitian Wilck IV & Cavalier (2012). *Capacity constraints* 5.19 dan 5.20 dikembangkan dari penelitian Wilck IV & Cavalier (2012) karena harus disesuaikan terkait indeks *node*. *Capacity constraints* 5.23, 5.24 dan 5.25 dibuat pada penelitian ini untuk kendaraan akan mengisi bahan bakar di *gas station*.

$$\sum_{j \in V} X_{0jl} = 1 \quad \forall l \in K \quad (5.16)$$

$$\sum_{i \in V} X'_{i0l} = 1 \quad \forall l \in K \quad (5.17)$$

Semua kendaraan yang ada akan berangkat dimulai dari depot untuk mengunjungi *node* selanjutnya pada Fungsi Kendala 5.16 dan akan kembali ke depot pada Fungsi Kendala 5.17.

$$\sum_{i \in V} \sum_{l \in K} X_{ijl} \geq 1 \quad \forall j \in V_c \quad (5.18)$$

arc (i, j) bisa digunakan lebih dari 1 kali oleh kendaraan $l \in K$

$$\sum_{i \in V^c} Y_{il} \geq 1 \quad \forall l \in K \quad (5.19)$$

Untuk semua kendaraan $l \in K$ dapat mengunjungi *node* i lebih dari satu kali kunjungan

$$\sum_{l \in K} Y_{il} \geq 1 \quad \forall i \in V_c \quad (5.20)$$

Untuk semua *node customer* dapat dikunjungi lebih dari satu kali oleh kendaraan $l \in K$

$$\sum_{i \in V} X_{ipl} = \sum_{i \in K} X_{pjl} \quad \forall p = 1, \dots, n, l \in K \quad (5.21)$$

Menjamin bahwa kendaraan akan berkunjung dan meninggalkan *node customer*

$$u_{ik} - u_{jk} + nx_{ijk} \leq n - 1 \quad \forall i \in V_c, l \in K \quad (5.22)$$

Eliminasi subtour yang mungkin terjadi

$$X_{mgl} = X'_{g0l} \quad \forall l \in K \quad (5.23)$$

$$X_{mgl} = 1 \quad \forall l \in K \quad (5.24)$$

$$X'_{g0l} = 1 \quad \forall l \in K \quad (5.25)$$

Ketiga *constraints* sebelumnya menjamin bahwa sebelum kendaraan kembali ke depot maka akan melakukan pengisian bahan bakar tepat setelah pembuangan paling akhir pada LPA dan sebelum kembali ke depot.

$$d_{il} \geq 0 \quad \forall i \in V, l \in K \quad (5.26)$$

$$X_{ijl} \in \{0,1\} \quad \forall (i,j) \in A, l \in K \quad (5.27)$$

$$Y_{il} \in \{0,1\} \quad \forall i \in V, l \in K \quad (5.28)$$

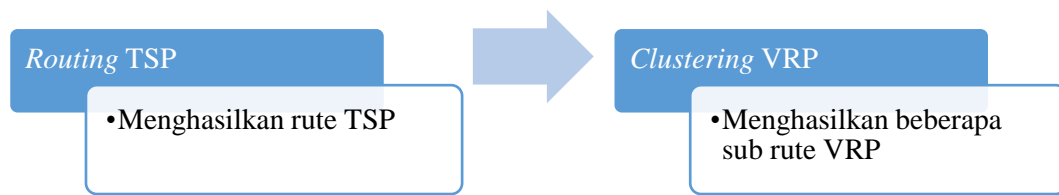
Ketiga *constraints* diatas menentukan *non-negativity* dan *binary variable*.

5.3 Pengembangan Algoritma ACO model VRPSSTWIF untuk MSW Collection di Kota Surabaya

Dalam pengembangan algoritma ACO untuk model VRPSSTWIF untuk kasus MSW *collection* di Kota Surabaya, penelitian ini menggunakan 2 tahapan utama seperti yang terlihat pada Gambar 5.2. Selanjutnya akan dijelaskan terkait parameter dan masing-masing tahapan algoritma yang akan digunakan dalam penelitian ini.

5.3.1 Parameter ACO untuk VRPSSTWIF

Berikut ini merupakan parameter yang digunakan dalam pengembangan algoritma ACO untuk model VRPSSTWIF untuk MSW *collection* di Kota Surabaya, beberapa diantaranya sudah merupakan ketetapan dalam suatu kasus dan



Gambar 5.2 Tahapan Algoritma ACO untuk Model VRPSSTWIF

sedangkan yang lain memiliki nilai ketidakpastian sehingga dalam penentuan nilainya perlu dilakukan beberapa percobaan dengan rekomendasi tertentu:

n Jumlah Kota

c_{ij} jarak tempuh (i, j)

a_i Waktu Awal Pelayanan TPS

b_0 Waktu Akhir pelayanan Depot

Keempat parameter sebelum ini merupakan parameter dengan nilai yang sudah ditentukan berdasarkan kasus tertentu.

h Nilai *visibility*

Nilai *visibility* dihitung dengan menggunakan Persamaan 2.13

r Bilangan *random*

Bilangan random dalam *range* $(0,1)$ untuk menentukan nilai diskrit yang mewakili ruas untuk setiap semut dalam menentukan rute selanjutnya yang akan dituju.

τ Indeks tingkat *pheromone* awal τ_{ij}^1

Nilai *pheromone* awal yang digunakan agar memudahkan perhitungan awal akan disamakan, maka nilai τ_{ij}^1 untuk setiap ruas yaitu 1.

N Jumlah Semut

iter Iterasi Maksimal

Tidak ada ketentuan pasti terkait berapa jumlah semut dan jumlah maksimal iterasi untuk suatu permasalahan tertentu, sehingga untuk mendapat nilai hasil terbaik perlu dilakukan beberapa kali *running* uji coba.

ρ Tingkat penguapan

Nilai dari tingkat penguapan adalah $\rho \in (0,1)$. Tingkat penguapan diasumsikan antara 0.5 sampai 0.8 (Santosa & Ai, 2017). Sedangkan beberapa penelitian lain Abdulkader, *et al.* (2015), Reed, *et al.* (2014), Dorigo & Stutzle

(2004) dan Bell & McMullen (2004) mengungkapkan bahwa $\rho = 0.9$ memberikan kesempatan yang bagus untuk memperbarui *pheromone* dalam perjalanan baru semut.

α Indeks tingkat *pheromone* $\tau_{i,j}$

β Indeks *visibility* h

Pada buku karangan Dorigo & Stutzle (2004) merekomendasikan nilai $\alpha=1$ dan $\beta=2$ karena dapat memberikan solusi yang baik. Hal ini didukung dengan beberapa penelitian setelahnya yang tepat menggunakan nilai yang sama Santosa & Ai (2017), Abdulkader, *et al.* (2015) dan Reed, *et al.* (2014).

5.3.2 Algoritma ACO untuk VRPSSTWIF

Berikut ini akan dibahas lebih detail terkait 2 tahapan utama algoritma ACO untuk VRPSSTWIF dalam kasus MSW *collection* di Kota Surabaya:

1. ACO TSP

Berikut ini merupakan algoritma yang digunakan dalam ACO untuk menghasilkan rute TSP, algoritma dapat dilihat pada Algoritma 5.1. Selanjutnya algoritma akan dirubah sedemikian rupa disesuaikan dengan bahasa pemrograman yang ada pada *Software Matlab*, koding lengkap ada pada Lampiran 5.

2. SUB RUTE VRP

Berikut ini merupakan algoritma yang digunakan dalam mencari rekomendasi rute untuk MSW *collection* di Kota Surabaya dengan pendekatan ACO. Pada Algoritma 5.2, terdapat didalamnya pengerjaan algoritma ACO TSP yang hasilnya akan digunakan sebagai dasar pembuatan sub rute VRP yang sudah disesuaikan dengan *constraints* yang ada. Koding lengkap dapat dilihat pada Lampiran 6.

Algoritma 5.1 ACO untuk TSP

Input: Jumlah Semut N , Jarak Tempuh c_{ij} (selain *node* LPA), Iterasi Maksimal *iter*.

Output: Rute Terbaik TSP, Total Jarak TSP.

Foreach $a = 1 : iter$ **do**

Reduksi Evaporasi menggunakan Persamaan 4.1

$$e = e - (e - e_r) * \frac{a}{iter} \quad (4.1)$$

Foreach $i = 1 : N$ **do**

Memulai perjalanan dari kota 1

end

Foreach $i = 1 : N$ **do**

Menghitung nilai *visibility* menggunakan Persamaan 2.13

Foreach $j = 1 : n$ **do**

Me-nol kan *visibility* kolom kota yang sudah terpilih

Menghitung probabilitas terpilihnya suatu kota selanjutnya dengan Persamaan 2.14

Membangkitkan bilangan random

Foreach $k = 1 : n$ **do**

Menghitung kumulatif peluang

Memilih tujuan kota baru yang belum dikunjungi dengan $r \leq s$, $s = kumulatif\ peluang\ k$

end

end

end

Mengakhiri rute perjalanan di kota 1

Foreach $i = 1 : N$ **do**

Jarak awal = 0

Foreach $k = 1 : n$ **do**

Jarak total = jarak awal + jarak dari kota 1 ke j + jarak dari kota I ke j + jarak dari kota j ke kota 1

end

Jarak total untuk rute yang dihasilkan setiap semut

end

Memilih jarak total terkecil dari seluruh perjalanan semut

Rute perjalanan adalah rute yang menghasilkan jarak total terkecil yang dilalui oleh seluruh semut

Update nilai evaporasi menggunakan Persamaan 2.18

Foreach $i = 1 : N$ **do**

Foreach $j = 1 : n$ **do**

Menghitung tambahan *pheromone* menggunakan Persamaan 2.16

Update nilai *pheromone* baru menggunakan Persamaan 2.17

end

end

Memilih jarak total terkecil dari seluruh perjalanan semut untuk seluruh iterasinya

Rute perjalanan adalah rute yang menghasilkan jarak total terkecil yang dilalui oleh seluruh semut pada setiap iterasi

End

Algoritma 5.1 VRP untuk MSW Collection di Kota Surabaya

Input: Semua Jarak Tempuh c_{ij} , Kapasitas Kendaraan C , Waktu Tempuh t_{ij} , Waktu Akhir pelayanan Depot b_0 , Waktu Awal Pelayanan TPS a_i , Jumlah Semut N , Iterasi Maksimal $iter$.

Output: Sub Rute Kendaraan, Total Jarak VRP, Waktu Komputasi.

Memisahkan jarak selain *node* LPA

Setting waktu mulai komputasi

Proses Algoritma ACO TSP

Memanggil kembali data jarak keseluruhan

Menentukan titik dari rute total $c = 2$, sub rute ke- $e = 1$

While $c < b$, $b =$ batas jumlah kota

Memulai sub rute dari depot (*node* 1)

Menentukan titik dari sub rute $f = 2$

Menentukan muatan awal = 0

Waktu keberangkatan = $a_i - t_{ij}$

Menentukan indeks perjalanan sebelumnya $Z = 1$, 1 berarti kendaraan berangkat dari *node customer* / LPS

While waktu $< b_0$ & $c < b$

If $Z < 2$

Pemilihan rute mempertimbangkan waktu dari *node* ke *node*, waktu ke LPA, waktu *loading*, waktu *unloading*, pengisian BBM dan balik ke depot tidak melebihi b_0 , jika $Z = 1$

Pemilihan rute mempertimbangkan waktu dari LPA ke *node*, waktu ke LPA, waktu *loading*, waktu *unloading*, pengisian BBM dan balik ke depot tidak melebihi b_0 , jika $Z = 2$

end

Muatan = 0

While muatan \leq kapasitas & $c < b$

If muatan \geq kapasitas

break

end

If muatan + *demand* $>$ kapasitas

(jika ya, berarti kendaraan tidak bisa mengambil keseluruhan *demand node*)

Cek perimbangan waktu

Memilih titik rute selanjutnya

Memperbarui waktu kumulatif dengan:

Waktu = waktu sebelumnya + waktu tempuh *node* sebelum ke *node* selanjutnya + waktu *loading*, jika $Z = 1$

Waktu = waktu sebelumnya + waktu tempuh dari LPA ke *node* selanjutnya + waktu *loading*, jika $Z = 2$

Memberbarui sisa *demand node*

Memperbarui muatan yang dibawa kendaraan

Menggeser titik sub rute $f = f + 1$

Memperbarui $Z = 1$

(jika tidak, berarti kendaraan bisa mengambil keseluruhan *demand node*)

Cek perimbangan waktu

Memilih titik rute selanjutnya

Memberbarui muatan dengan menambahkan *demand node*

Memperbarui waktu kumulatif dengan:

```

Waktu = waktu sebelumnya + waktu tempuh node sebelum ke node
selanjutnya + waktu loading, jika  $Z = 1$ 
Waktu = waktu sebelumnya + waktu tempuh dari LPA ke node
selanjutnya + waktu loading, jika  $Z = 2$ 
Menggeser titik sub rute  $f = f + 1$ 
Menggeser titik rute TSP  $c = c + 1$ 
Memperbarui  $Z = 1$ 
end
end
Memulai rute dari LPA
Memberbarui waktu kumulatif = waktu seblum + waktu tempuh node ke
LPA + waktu unloading
Menggeser titik sub rute
Memberbarui nilai  $Z = 2$ 
end
Mengembalikan kendaraan ke depot
Waktu tempuh kendaraan  $l =$  waktu sebelum + waktu tempuh dari LPA ke depot
Memberbarui nilai sub rute  $e = e + 1$ 
end
 $q =$  Panjang rute
for  $g = 1 : q$ 
rute = rute ( $g$ )
Menghitung jarak sub rute
end
Menjumlahkan total jarak
Setting waktu akhir komputasi

```

5.4 Verifikasi Validasi Model dan Algoritma

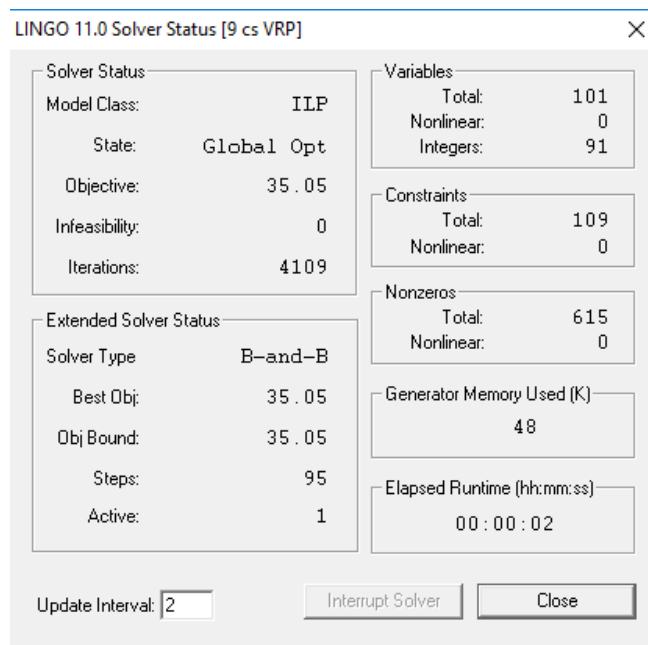
Validasi dan verifikasi model dan algoritma dilakukan dengan tujuan untuk mengetahui apakah model yang dibuat bisa mewakili persoalan serta algoritma yang disusun bisa menyelesaikan persoalan tersebut sehingga solusi yang dihasilkan tidak melanggar *constraints* apapun. Model matematis yang akan diselesaikan pada penelitian ini didekati dengan algoritma metaheuristik dengan pertimbangan bahwa permasalahan VRP merupakan *NP-hard problem*, terlebih untuk kasus *MSW collection* di Kota Surabaya permasalahannya sangat kompleks dan mempunyai *node customer* lebih dari 60. Apabila CVRP melayani 60 *customer* hal itu merupakan batas maksimum metode *exact* dapat berjalan secara optimal (Talbi, 2009).

Telah dilakukan percobaan untuk kasus sampel kecil dan sederhana untuk membandingkan perilaku metode *exact* dengan algoritma ACO sebagai metode metaheuristik yang nantinya akan digunakan dalam penelitian ini. Hasilnya pada percobaan 10 *node* CVRP untuk keduanya, algoritma ACO mampu mendekati

solusi optimal yang dihasilkan oleh metode *exact*, seperti yang terlihat pada Gambar 5.3 Dan 5.4. Nilai fungsi tujuan dari kedua penyelesaian ini sama yaitu 35.05. Selanjutnya akan dilakukan beberapa kali *running* untuk kedua cara tersebut dengan penambahan jumlah *node* untuk melihat perilaku metode penyelesaiannya. Pada algoritma ACO parameter yang digunakan adalah iterasi maksimal = 1500, $\alpha=1$, $\beta=2$, N semut = 5 dan $\rho = 0.9$. Hasil dari beberapa kali percobaan dapat dilihat pada Tabel 5.2.

Pada Tabel 5.1 diketahui bahwa algoritma dapat mengikuti dengan baik solusi dari metode *exact* karena gap fungsi tujuan yang dihasilkan tidak terlalu jauh, sedangkan dengan penambahan jumlah *node* metode *exact* mempunyai kelemahan terkait waktu komputasi. Pada penelitian ini diharapkan dapat memberikan rekomendasi rute dalam waktu yang cepat dan baik, sehingga selanjutnya akan dikembangkan metode ACO untuk merekomendasikan rute MSW *collection* di Kota Surabaya.

Pendekatan algoritma ACO rute dibuat secara “*route first cluster second*” dengan demikian solusi terlebih dahulu dihasilkan seperti halnya solusi pada persoalan *travelling salesman problem* (TSP), selanjutnya akan dipecah-pecah menjadi beberapa rute menyesuaikan *constraints* yang ada.



Gambar 5.3 CVRP *exact* 10 *node*

```

>> [jarak_total, rute] = VRP1(jar, jum_semut, iter, kapasitas, D)

rut =

     1     2     1

rut =

     1    10     6     7     8     5     3     4     1

rut =

     1     9     1

jaraksub =

     4.2000    25.4500    5.4000

comp_time =

     0.6406

```

Gambar 5.4 CVRP ACO 10 node

Tabel 5.2 Perbandingan *Exact* dan ACO CVRP

Jumlah Node	Exact CVRP				ACO CVRP			GAP Jarak	GAP Waktu
	Fungsi Tujuan	Keterangan	Sub Rute	Waktu	Fungsi Tujuan	Sub Rute	Waktu		
10 node	30.05	Global optimum	3	00.00.02	30.05	3	0.0113	0.0	00.00.08
15 node	72.35	Global optimum	5	00.09.59	72.85	6	0.9688	0.5	00.08.62
25 node	97.85	Feasible solution	7	02.30.57	102.15	8	4.6719	4.3	02.18.52
30 node	117.15	Feasible solution	9	05.20.08	120.2	10	7.3438	3.1	05.20.00
40 node	139.67	Feasible solution	10	10.20.07	152.82	11	7	13.2	10.20.00

Satuan yang digunakan: fungsi tujuan (km) dan waktu (jam)

5.4.1 Verifikasi dan Validasi Model

Validasi model didasarkan atas formulasi VRPSSTWIF yang telah dibangun untuk merepresentasikan sistem MSW *collection* di Kota Surabaya dengan truk *compactor*. Adanya pengembangan model dari *capacited* VRP maupun dari penelitian sebelumnya oleh Buhrkal, *et al.* (2012) membutuhkan adanya pembuktian terlebih dahulu. Verifikasi dan validasi model matematis akan diuji dengan menggunakan sampel kecil. Jarak tempuh, waktu tempuh dan *demand* untuk sampel kecil dapat dilihat pada Tabel 5.3, 5.4, dan 5.5.

Node no 1 menunjukkan depot, *node* 2 sampai 10 menunjukkan LPS-LPS dan *node* 82 menunjukkan LPA. Persoalan tersebut akan diselesaikan dengan kendaraan homogen yang memiliki kapasitas 35 kontainer. Waktu awal pelayanan untuk semua LPS sama yaitu pada menit ke 60 dan kendaraan harus sampai pada depot pada menit ke 720. Waktu *loading* adalah 1 menit/kontainer pada setiap LPS.

waktu *unloading* adalah 60 menit pada LPA. Waktu pengisian bahan bakar adalah 30 menit pada *gas station*.

Dilakukan *running* model TSP untuk menemukan rute awal dengan metode *exact* sehingga dijadikan dasar untuk *cluster* pembuatan rute selanjutnya. Hasil *running* model dapat dilihat pada Gambar 5.5, dengan rute **1-9-4-3-5-8-7-6-10-2-1** dengan total jarak tempuh 30.35 km.

VRP disebut dengan *m-TSP* atau TSP yang dipecah menjadi beberapa sub rute dengan kapasitas tertentu. Maka, langkah selanjutnya adalah *clustering* yang nantinya akan didapatkan sub rute. Tabel 5.6 menunjukkan cara pembuatan sub rute sedangkan untuk sub rute selanjutnya terdapat pada Lampiran 5.

Pada pembuatan sub rute 1 diketahui bahwa kendaraan akan berangkat pada menit ke 51 harena waktu tempuh dari depot ke tujuan selanjutnya (*node 9*) adalah 9, maka = 60-9. Status pada keterangan akan “AMBIL” apabila dalam pengambilan di *node* tersebut (*cycle time*, waktu tempuh antar *node*, *loading/unloading time* dan *service time* gs) sama dengan atau lebih kecil dari 720. Keterangan “STOP” digunakan jika sebaliknya atau diluar *time window* sistem.

Tabel 5.3 Jarak Tempuh Sampel Kecil

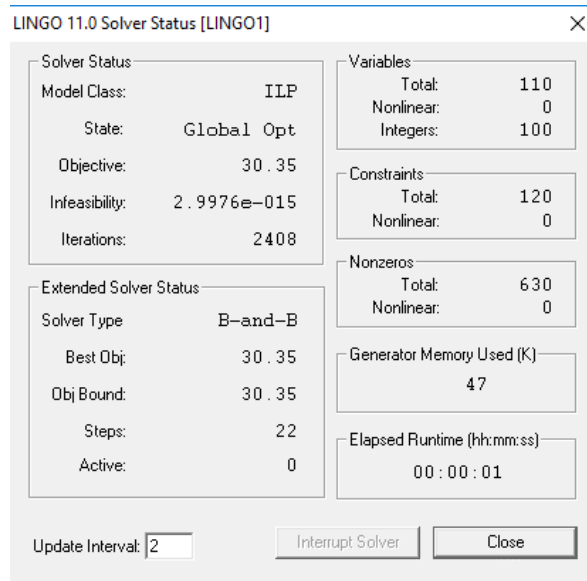
	1	2	3	4	5	6	7	8	9	10	82
1	0	2.1	3.6	3.1	4.7	7.3	8.1	7.5	2.7	4.8	23.1
2	2.1	0	4.7	4.2	4.8	6.8	7.4	6.9	3.4	3.8	25.3
3	3.6	4.7	0	1.7	2.9	5.7	6.3	5.7	4.6	6.5	18.8
4	3.1	4.2	1.7	0	3.3	6.1	6.7	6.1	4.2	6.1	20.4
5	4.7	4.8	2.9	3.3	0	2.8	3.4	2.8	7.6	6.5	21.8
6	7.3	6.8	5.7	6.1	2.8	0	0.65	2	9.4	7.8	21.5
7	8.1	7.4	6.3	6.7	3.4	0.65	0	1.7	9.3	7.7	21.3
8	7.5	6.9	5.7	6.1	2.8	2	1.7	0	10.4	9	21.9
9	2.7	3.4	4.6	4.2	7.6	9.4	9.3	10.4	0	6.6	27.5
10	4.8	3.8	6.5	6.1	6.5	7.8	7.7	9	6.6	0	24.5
82	23.1	25.3	18.8	20.4	21.8	21.5	21.3	21.9	27.5	24.5	0

Tabel 5.4 Waktu Tempuh Sampel Kecil

	1	2	3	4	5	6	7	8	9	10	82
1	0	6	10	7	9	17	17	16	9	8	64
2	6	0	16	11	15	18	20	19	10	10	45
3	10	16	0	9	9	17	18	16	14	18	49
4	7	11	9	0	14	21	24	22	12	16	50
5	9	15	9	14	0	7	9	7	26	15	49
6	17	18	17	21	7	0	2	4	21	16	33
7	17	20	18	24	9	2	0	5	22	17	34
8	16	19	16	22	7	4	5	0	25	19	35
9	9	10	14	12	26	21	22	25	0	15	51
10	8	10	18	16	15	16	17	19	15	0	44
82	64	45	49	50	49	33	34	35	51	44	0

Tabel 5.5 Demand Sampel Kecil

1	2	3	4	5	6	7	8	9	10	82
0	140	13	10	2	1	2	1	140	70	0



Gambar 5.5 TSP Exact dengan Lingo

Tabel 5.6 Sub Rute 1 Validasi dan Verifikasi Model

Sub Rute 1	1	9	82	9	82	9	82	9	gs	1
Jarak tempuh antar node		2.7	27.5	27.5	27.5	27.5	27.5	27.5	30	23.1
Jarak kumulatif		2.7	30.2	57.7	85.2	112.7	140.2	0		163.3
Demand		140	0	105	0	70	0	35		0
Pickup demand		35	0	35	0	35	0	35		0
Demand kumulatif		35	0	35	0	35	0	0		0
load/unloading time		35	60	35	60	35	60	35		0
Waktu tempuh antar node		9	51	51	51	51	51	51	64	
Cycle time	51	95	206	292	403	489	514	600	544	608
Keterangan		AMBIL		AMBIL		AMBIL		STOP		

Contoh perhitungan:

1. Waktu keberangkatan = waktu awal pelayanan – waktu tempuh antar node j
= $60-9 = 51$ (cycle time 1)
2. Loading time (non noe 82) = 1 menit * pickup demand
= $1*35 = 35$ (loading time 9)
3. Unloading time (node 82) = 60 menit
4. Cycle time = cycle time sebelum + (loading/unloading) + waktu tempuh antar node
= $51 + 35 + 9 = 95$ (cycle time 9)
5. "AMBIL" jika cycle time + jarak node ke LPA + service time di LPA + service time di gas station + jarak LPA ke Depot ≤ 900
6. "STOP" jika cycle time + jarak node ke LPA + service time di LPA + service time di gas station + jarak LPA ke Depot > 900

Tabel 5.7 Hasil Clustering Manual Metode Exact TSP

Sub Rute ke-	Jarak (km)	Waktu (menit)
1-9-82-9-82-9-82-1	163.3	608
1-9-82-4-3-5-8-7-6-10-82-10-82-1	164.75	720
1-10-2-82-2-82-2-82-1	158.2	674
1-2-82-2-82-1	101.1	743
Total	587.35	2475

Setelah didapatkan keterangan "STOP" dan *demand* pada LPS-LPS belum terambil maka akan dibuat sub rute baru, pada sampel kecil ini terdapat 4 rute. Pada tabel 5.7 merupakan rangkuman hasil perhitungan manual *clustering* sampel kecil. 4 rute berarti juga dalam kasus ini dibutuhkan 4 buah kendaraan. Jarak total yang akan ditempuh keempat kendaraan tersebut adalah 587.35 km dalam waktu 2475 menit. Selanjutnya akan dilakukan pengecekan terhadap *constraints* yang ada pada model matematis sebelumnya yang akan dirangkum pada Tabel 5.8. Hasil evaluasi perhitungan manual dari TSP *exact* sudah memenuhi *constraints*, selanjutnya akan dibandingkan dengan menggunakan TSP algoritma ACO dengan *software* Matlab.

Tabel 5.8 Evaluasi *Clustering Manual Rute Exact*

Evaluasi <i>Clustering Manual Rute Exact</i>	Sub Rute Ke-				
	1	2	3	4	
Fungsi Tujuan					
Rute yang ditempuh	1-9-82-9-82-9-82-1	1-9-82-4-3-5-8-7-6-10-82-10-82-1	1-10-2-82-2-82-2-82-1	1-2-82-2-82-2-82-1	
Jarak	163.3	164.75	158.2	101.1	
Total Jarak	587.35				
Constraints					
- Depot					
<i>Load</i>	0	√	√	√	√
<i>Time Windows</i>	$b \leq 720$	608 √	720 √	674 √	473 √
<i>Gs</i>	30 menit	√	√	√	√
- LPS					
<i>Demand</i>		105	105	105	64
<i>Total Capacity</i>	379	379 √			
<i>Time Windows</i>	$a \geq 60$	√	√	√	√
<i>Split Service</i>		√	√	√	√
<i>Intermediate Facility</i>		√	√	√	√
- LPA					
<i>Load</i>	0	√	√	√	√
<i>Time Windows</i>	60 menit	√	√	√	√

```

>> best_rute = ACOAlg (jar, jum_semut, iter)

jaraktot =

    30.3500    30.3500    30.3500    30.3500    30.3500    30.3500    30.3500

comp_time =

    0.7656

best_rute =

     1     2    10     6     7     8     5     3     4     9     1

```

Gambar 5.6 TSP ACO dengan Matlab

Tabel 5.9 Hasil *Clustering* Manual Metode ACO TSP

Sub Rute ke-	Jarak (km)	Waktu (menit)
1-2-82-2-82-2-82-1	151.7	664
1-2-82-10-82-10-82-1	132.1	649
1-6-7-8-5-3-4-9-82-9-82-1	154.35	558
1-9-82-9-82-9-82-1	152.3	688
Total	590.45	2585

Gambar 5.6 merupakan hasil *running* ACO yang paling mendekati optimal dengan menggunakan metode *exact* sebelumnya. *Running* model ACO dilakukan dengan jumlah semut = 7 dan iterasi maksimal = 1000. Jarak total yang dihasilkan sama dengan metode sebelumnya yaitu 30.35 km walaupun menempuh rute yang berbeda. Sehingga dapat dikatakan bahwa algoritma ACO untuk perutean TSP dapat dengan baik menghasilkan nilai yang mendekati optimal atau mendekati nilai yang dihasilkan dari metode *exact*.

Selanjutnya rute yang didapatkan akan dikerjakan secara manual sama dengan langkah sebelumnya untuk *clustering* menemukan beberapa sub rute dengan pertimbangan *constraints* yang ada. Pada tabel 5.8 merupakan rangkuman hasil perhitungan manual *clustering* sampel kecil dengan dasar hasil perutean TSP dengan menggunakan algoritma ACO. Sama seperti sebelumnya telah dihasilkan 4 rute berarti juga dalam kasus ini dibutuhkan 4 buah kendaraan. Jarak total yang akan ditempuh keempat kendaraan tersebut adalah 590.45 km dalam waktu 2559 menit. Selanjutnya akan dilakukan pengecekan terhadap *constraints* yang ada pada model matematis sebelumnya yang akan dirangkum pada Tabel 5.9.

Hasil yang didapatkan pada tahap ini adalah apabila dilakukan perhitungan manual dengan rute dasar dari model *exact* TSP maupun algoritma ACO TSP, keduanya dapat memenuhi *constraints* yang ada pada model matematis. Gap jarak yang dihasilkan dari perhitungan manual dari dasar rute TSP yang berbeda adalah $590.45 - 587.35 = 3.1$ km, hal ini berarti bahwa TSP yang dihasilkan melalui algoritma ACO dapat menghasilkan solusi yang mendekati nilai optimal atau dapat mengikuti metode *exact* dengan waktu yang lebih cepat. Selanjutnya akan dilakukan verifikasi dan validasi yang berkaitan dengan pengembangan algoritma ACO untuk permasalahan pada penelitian ini.

Tabel 5.10 Evaluasi Clustering Manual Rute ACO

Rute yang ditempuh		1-2-82-2-82-2-82-1	1-2-82-10-82-10-82-1	1-6-7-8-5-3-4-9-82-9-82-1	1-9-82-9-82-9-82-1
Jarak		151.7	132.1	154.35	152.3
Total Jarak		590.45			
Constraints					
- Depot					
Load	0	√	√	√	√
Time Windows	$b \leq 720$	664 √	649 √	558 √	688 √
Gs	30 menit	√	√	√	√
-LPS					
Demand		105	105	70	99
Total Capacity	379	379 √			
Time Windows	$a \geq 60$	√	√	√	√
Split Service		√	√	√	√
Intermediate Facility		√	√	√	√
- LPA					
Load	0	√	√	√	√
Time Windows	60 menit	√	√	√	√

5.4.2 Verifikasi dan Validasi Algoritma

Validasi algoritma dilaksanakan melakukan pembuktian terbalik yaitu hasil pencarian rute yang didapatkan akan dibandingkan dengan perhitungan manual. Proses validasi dijalankan menggunakan data sampel kecil yang sama dengan sebelumnya. Pada tahap sebelumnya telah diketahui bahwa algoritma ACO TSP dapat dengan baik mendekati nilai optimal dari metode *exact*. Selanjutnya pada tahap ini akan dibahas terkait algoritma *clustering* dalam pembuatan sub rute dengan menggunakan *Software Matlab*.

Sub rute yang dihasilkan dari TSP *exact* dengan menggunakan algoritma dapat dilihat pada Gambar 5.7. Sedangkan sub rute yang dihasilkan dari TSP ACO dengan menggunakan algoritma dapat dilihat pada Gambar 5.8. Hasil dari keduanya menunjukkan sub rute yang sama dengan yang dihasilkan dengan perhitungan manual pada tahap sebelum ini. Sehingga, algoritma ini siap untuk digunakan pada sampel yang lebih besar dalam waktu yang lebih cepat.

Sub rute yang dihasilkan dari TSP *exact* dengan menggunakan algoritma merupakan hasil dari tahap kedua dari algoritma ini yaitu *clustering*. Dikatakan valid apabila hasil sub rute yang didapatkan sama dengan perhitungan manualnya. Hal ini terjadi karena pada tahap *clustering* tidak dibangkitkan bilangan random seperti pada tahap pembuatan rute TSP yang membangkitkan bilangan random dalam penentuan rute yang selanjutnya akan dituju.

```
>> [jarak_total, jaraksub] = VRP(alljar, jum_semut, iter, kapasitas, D, wtempuh, maxt)

rut =

     1     9    11     9    11     9    11     1

rut =

     1     9    11     4     3     5     8     7     6    10    11    10    11     1

rut =

     1    10     2    11     2    11     2    11     1

rut =

     1     2    11     2    11     1

comp_time =

    0.0156
```

Gambar 5.7 Hasil *Running Clustering TSP Exact* dengan Matlab

```

>> [jarak_total, jaraksub] = VRP(alljar, jum_semut, iter, kapasitas, D, wtempuh, maxt)

rut =

     1     2    11     2    11     2    11     1

rut =

     1     2    11    10    11    10    11     1

rut =

     1     6     7     8     5     3     4     9    11     9    11     1

rut =

     1     9    11     9    11     9    11     1

comp_time =

    0.0469

```

Gambar 5.8 Hasil Running Clustering TSP ACO dengan Matlab

(Halaman ini sengaja dikosongkan)

BAB VI

HASIL DAN ANALISIS

Pada bab ini akan dijelaskan mengenai hasil dan analisis. Dengan menggunakan algoritma yang sudah dirancang. Dimulai dari percobaan model algoritma, perbandingan rute *existing* dengan rute yang direkomendasikan, penugasan kendaraan dan yang terakhir akan dilakukan Analisa sensitivitas.

6.1 Percobaan Model Algoritma

Setelah model algoritma tervalidasi dan terverifikasi, maka algoritma siap digunakan untuk sampel kasus yang lebih besar, yaitu kasus *MSW collection* di Kota Surabaya. Data yang dibutuhkan untuk proses *running* sudah ada dalam pembahasan bab selanjutnya. Percobaan ini membutuhkan uji beberapa kali sampel dengan beberapa replikasi untuk setiap calon parameter yang akan dipilih. Setelah parameter diputuskan akan dilakukan *running* model dengan parameter tersebut dalam beberapa replikasi untuk mengetahui nilai terbaik fungsi tujuan. Hasil yang didapatkan selanjutnya akan dilakukan analisa.

6.1.1 Parameter Model Algoritma untuk MSW Collection di Kota Surabaya

Dalam penelitian ini, pencarian rute *MSW collection* di Kota Surabaya menggunakan algoritma ACO. Di dalam algoritma ACO mengandung nilai random sehingga dalam penentuan nilai beberapa parameter perlu dilakukan beberapa kali perbandingan nilai parameter dalam beberapa replikasi untuk menentukan nilai parameter mana yang terbaik untuk digunakan dalam pencarian rute *MSW collection* di Kota Surabaya. Parameter yang akan diujikan adalah:

1. Evaporasi

Nilai evaporasi yang akan diuji adalah $\rho = (0.5, 0.6, 0.7, 0.8)$ (Santosa & Ai, 2017) dan $\rho = (0.9)$ (Abdulkader, *et al*, 2015; Reed, *et al*. 2014; Dorigo & Stutzle, 2004; dan Bell & McMullen, 2004).

2. Iterasi

Nilai iterasi yang akan diujikan adalah 500, 1000 dan 1500.

3. Jumlah semut

Jumlah semut yang akan diujikan adalah 5, 10 dan 15.

Pengujian semua algoritma dalam penelitian ini akan dilakukan dengan *software Matlab* dengan spesifikasi komputer: Intel® core™ i3-3110m CPU @2.40GHZ RAM 6 GB. Setiap kombinasi nilai parameter akan diujikan sebanyak 10 replikasi. Selanjutnya untuk setiap 10 replikasi akan dicari nilai total jarak minimum, rata-rata total jarak dan standart deviasinya. Perhitungan selengkapnya ditampilkan pada Lampiran 9. Rangkuman dari hasil pencarian nilai parameter terbaik dapat dilihat pada Tabel 6.1.

Tabel 6.1 Uji Coba Parameter

ρ	N	i	Min	Mean	SD	ρ	N	i	Min	Mean	SD		
0.5	5	500	4304	4422.8	75.11	0.6	5	500	4322.3	4406.4	46.91		
			35	35.7	0.67				35	35.9	0.57		
			2.7188	2.8141	0.07				2.75	2.8797	0.12		
		1000	4306.6	4369.4	30.45			1000	4353.6	4423.8	61.31		
			35	35.6	0.52				35	35.9	0.74		
			5.1094	5.2953	0.13				5.375	5.586	0.16		
			4336.7	4404.4	59.38				1500	4325.8	4421.2	78.51	
			35	35.9	0.57					35	36	0.47	
			7.625	8.0682	0.44					8.4531	8.6	0.11	
		10	500	4335.6	4402			47.44	10	500	4301.3	4365.3	48.22
				35	35.8			0.63			35	35.3	0.67
				5.25	5.425			0.12			5.7969	7.0172	2.97
	1000		4283.4	4426	85.11	1000	4310.4	4413.5		50.74			
			35	35.5	0.53		35	35.8		0.42			
			10.203	10.684	0.28		10.125	10.558		0.55			
			4259.2	4359.7	57.39		1500	4277.8		4384.8	53.86		
			35	35.2	0.42			35		35.8	0.63		
			15.125	15.534	0.33			15.203		15.948	0.74		
	15		500	4315.1	4386.3	56.21	15	500		4375.4	4422	46.84	
				35	35.7	0.67				34	35.6	0.84	
				7.9531	9.3422	2.91				8.0313	8.425	0.30	
		1000	4302.4	4387	56.12	1000		4220	4337.1	77.31			
			35	35.4	0.52			34	34.6	0.52			
			16.406	17.77	1.37			16.078	16.488	0.21			
			4295.7	4367	42.74			1500	4290.5	4372.6	45.00		
			35	35.5	0.71				35	35.7	0.48		
			22.609	23.091	0.29				24.359	25.578	0.76		

Tabel 6.1 Uji Coba Parameter (Lanjutan)

ρ	N	i	Min	Mean	SD	ρ	N	i	Min	Mean	SD	ρ	N	i	Min	Mean	SD			
0.7	5	500	4299.2	4413.5	101.38	0.8	5	500	4298.4	4431	75.62	0.9	5	500	4263.2	4402	63.06			
			35	35.8	0.63				35	36.1	0.74				35	36.1	0.57			
			2.4844	2.7157	0.13				2.7344	2.8781	0.16				2.7188	2.9672	0.40			
		1000	4281.5	4415.5	62.65			1000	4319.9	4446	63.13			1000	4311.9	4458.5	93.08			
			35	35.9	0.32				35	36.2	0.63				35	35.9	0.57			
			4.9844	5.2063	0.18				5.5	5.7688	0.24				5.5938	5.9516	0.26			
		1500	4331.3	4427.8	60.09			1500	4340.8	4420.8	60.83			1500	4351.8	4465.8	57.81			
			35	36.1	0.57				35	36.1	0.57				36	36.2	0.42			
			8.2813	8.625	0.24				7.8906	8.2203	0.20				8.1875	8.5094	0.26			
		10	500	4306.7	4400.3			59.65	10	500	4301.1			4404.7	81.55	10	500	4316.5	4409.6	66.34
				35	35.9			0.74			35			36	0.67			35	35.6	0.52
				5.3594	5.6594			0.23			5.2656			5.4641	0.19			5.5781	6.7485	1.29
	1000		4312.5	4429.2	56.54		1000	4306		4412.2	54.31		1000	4252.2	4368.8		70.05			
			35	36	0.67			35		35.8	0.42			35	35.5		0.53			
			10.25	10.667	0.23			10.719		11.347	0.51			11.828	13.723		2.73			
	1500		4320.7	4381.2	43.71		1500	4280.3		4376.6	60.58		1500	4306.4	4358.1		26.16			
			16	33.6	6.20			35		35.7	0.67			35	35.8		0.63			
			16.109	16.925	0.31			15.906		16.605	0.76			16.016	18.872		3.51			
	15		500	4289.5	4371.1		68.83	15		500	4268.8		4387.3	66.21	15		500	4295.9	4397	55.88
				35	35.5		0.53				35		35.6	0.52				35	35.5	0.71
				7.8594	8.2813		0.28				8.1875		9.8719	1.65				7.9219	8.3297	0.40
		1000	4297.7	4390.7	70.42		1000		4347	4427.6	58.93		1000	4297.7		4399.1	68.88			
			35	35.3	0.48				35	35.9	0.32			35		35.9	0.57			
			16.047	16.452	0.41				16.25	18.569	3.02			16.781		19.544	2.20			
		1500	4328.3	4412.3	55.73		1500		4318.4	4367	27.70		1500	4277.3		4366.1	43.05			
			35	35.7	0.48				35	35.5	0.53			35		35.5	0.53			
			25.641	26.55	0.78				23.938	454.53	3.2			23.313		28.508	3.20			

Berdasarkan Tabel 6.1 dapat diketahui bahwa:

1. Terkait waktu komputasi, apabila semakin besar jumlah semut maupun iterasinya maka waktu komputasi nilai rata-rata secara konstan yang dibutuhkan akan semakin tinggi. Sedangkan semakin besar nilai evaporasi tidak secara konstan mempengaruhi tinggi rendahnya waktu yang dibutuhkan, sesuai dengan contoh sampel kecil perbandingan waktu komputasi yang diambil pada Tabel 6.2.

Tabel 6.2 Parameter terhadap Waktu Komputasi

$\rho =$ 0.5	N			$\rho =$ 0.7	N			$\rho =$ 0.9	N		
	5	10	15		5	10	15		5	10	15
500	2.814	5.425	9.34	500	2.716	5.659	8.281	500	2.967	6.748	8.33
1000	5.295	10.68	17.77	1000	5.206	10.67	16.45	1000	5.952	13.72	19.54
1500	8.068	15.53	23.09	1500	8.625	16.93	26.55	1500	8.509	18.87	28.51

Waktu dalam satuan detik

2. Terkait dengan hasil fungsi tujuan yang didapatkan atau jarak totalnya, diketahui bahwa nilai minimum dan nilai rata-rata untuk semua parameter terangkum pada Tabel 6.3. Jarak total terkecil dimiliki oleh parameter

Tabel 6.3 Nilai Minimum Hasil Uji Coba

Keterangan	Min	Mean
Jarak Total	4252.2	4358.06
Parameter	[0.9,10,1000]	[0.9,10,1500]
Jumlah Kendaraan	35	35.2

Jarak dalam satuan km, parameter merupakan $[\rho, N, \text{iterasi}]$ dan kendaraan dalam satuan unit.

3. Kedua parameter terbaik mempunyai nilai evaporasi 0.9 dan jumlah semut sebanyak 10. Perbedaan hanya ada pada jumlah iterasi, selanjutnya kedua nilai tersebut dievaluasi lebih dalam. Pada iterasi 1000 menjadi nilai terbaik pada kategori nilai minimal, pada parameter tersebut memiliki rata-rata 4369 km dan standart deviasi sebesar 70.05 km. Pada iterasi 1500 menjadi nilai terbaik pada kategori rata-rata, pada parameter tersebut memiliki nilai minimal sebesar 26.16 km. Untuk kedua parameter tersebut bahkan bagi keseluruhan parameter jumlah kendaraan yang dibutuhkan sejumlah minimal 35.

Dari beberapa pertimbangan sebelumnya maka diputuskan pada penelitian ini menggunakan parameter nilai evaporasi $\rho = 0.9$, jumlah semut $N = 10$ dan iterasi

= 1500. Terpilihnya parameter ini karena iterasi 1500 memiliki standart deviasi terhadap nilai rata-rata perusahaan lebih kecil dibandingkan dnegan iterasi 1000.

6.1.2 Hasil *Running* Model Algoritma untuk MSW *Collection* di Kota Surabaya

Running model algoritma untuk MSW *collection* di Kota Surabaya dengan algoritma ACO menggunakan parameter yang sesuai dengan keputusan sub bab sebelum ini yaitu nilai evaporasi 0.9, jumlah semut 10 dan maksimal iterasi 1500. *Running* model dilakukan sebanyak 10 replikasi dan dipilih nilai terbaik dari 10 replikasi tersebut sebagai nilai fungsi tujuan yang didapatkan oleh model algoritma ACO.

Tabel 6.4 Hasil *Running* 10 Replikasi Model Algoritma

No	Fungsi Tujuan	Jumlah Kendaraan	Waktu	No	Fungsi Tujuan	Jumlah Kendaraan	Waktu
1	4335.2	35	16.5781	6	4314.6	36	17.4688
2	4291.5	36	15.8281	7	4343.2	36	15.8594
3	4344.3	35	15.5469	8	4333	36	16.0469
4	4311.9	36	17.6094	9	4393.2	35	16.3906
5	4265.7	35	15.6563	10	4408.5	35	16.4688

Satuan yang digunakan Fungsi tujuan (km), jumlah kendaraan (unit) dan waktu (detik)

Nilai minimum dari 10 replikasi yang dijalankan ada pada replikasi ke-5 dengan nilai fungsi tujuan sebesar 4265.7 km. Berikut ini merupakan hasil detail dari replikasi ke-5 sebagai replikasi yang memiliki nilai fungsi tujuan minimum:

```
>> [jarak_total, jaraksub] = VRP(alljar, jum_semut, iter, kapasitas, D, wtempuh, maxt)
rut 1=
  1 15 16 82 16 9 82 1
rut 2=
  1 9 82 9 82 1
rut 3=
  1 9 82 9 73 82 1
rut 4=
  1 73 82 73 17 12 82 1
rut 5=
  1 12 34 82 34 82 34 35 82 1
rut 6=
  1 28 29 82 14 82 1
rut 7=
  1 14 82 14 82 14 82 1
rut 8=
  1 46 13 82 18 82 1
```

rut 9=
1 18 82 59 60 82 1
rut 10=
1 60 62 61 82 61 58 49 82 1
rut 11=
1 49 50 82 50 33 82 1
rut 12=
1 33 74 82 74 82 1
rut 13=
1 74 82 74 63 64 82 1
rut 14=
1 64 10 82 10 82 10 82 1
rut 15=
1 21 43 56 82 56 23 82 1
rut 16=
1 23 82 23 30 82 30 82 1
rut 17=
1 40 25 42 48 82 41 39 38 24 22 5 82 1
rut 18=
1 3 53 82 53 82 1
rut 19=
1 53 82 53 82 53 82 1
rut 20=
1 27 82 27 4 70 82 1
rut 21=
1 70 82 70 69 82 69 82 1
rut 22=
1 69 65 82 65 82 65 82 1
rut 23=
1 65 82 65 47 82 47 82 1
rut 24=
1 52 82 52 82 52 82 1
rut 25=
1 52 11 82 11 82 11 54 82 1
rut 26=
1 54 51 32 31 82 31 8 6 7 19 82 19 79 82 1
rut 27=
1 75 20 81 82 81 77 55 82 1
rut 28=
1 55 76 72 82 72 82 72 78 82 1
rut 29=
1 80 26 82 26 82 1
rut 30=
1 26 82 26 71 82 1
rut 31=
1 71 82 71 68 82 68 82 1
rut 32=
1 68 37 45 2 82 2 82 1
rut 33=
1 2 82 2 82 2 44 82 1

```

rut 34=
 1 57 66 82 66 82 66 82 1
rut 35=
 1 67 82 67 82 36 82 1
comp_time =
 17.4063
jarak_total =
 4.4264e+03

```

6.2 Perbandingan Rute *Existing* dan Rute Rekomendasi

Rute *existing* yang selama ini digunakan di DKRTH Surabaya menggunakan kendaraan sejumlah 44 seperti yang disebutkan dalam bab sebelumnya. Dari rute *existing* diketahui waktu keberangkatan kendaraan dari depot dan waktu kendaraan kembali ke depot. *Gantt Chart* terkait *cycle time* untuk setiap perjalanan rute kendaraan dapat di lihat secara detail pada Lampiran 10. Selanjutnya akan dibuat *Gantt Chart* untuk rute rekomendasi yangh secara detail dapat dilihat pada Lampiran 11.

Pada *Gantt Chart* satuan yang digunakan adalah menit, contohnya ada pada Tabel 6.5. Dapat dilihat bahwa pada tabel tertulis nilai jadwal keberangkatan dan *cycle time*, keduanya merupakan nilai dari menit ke-, satuan yang digunakan adalah menit. Sebagai contoh sub rute 1 memulai perjalanan dari kotak berwarna kuning pada angka 295, artinya maksimal pada menit ke-295 kendaraan yang melakukan perjalanan pada sub rute 1 akan berangkat sehingga tepat pada menit ke-300 sudah sampai pada tujuan LPS pertama yang akan dilayani. Seluruh LPS baru dapat dilayani pada menit ke-300. Sub rute 1 diperkirakan akan kembali pada depot maksimal pada menit ke-720 yang artinya tidak melanggar *time window constraint* untuk setiap kendaraan maksimal kembali ke depot pada menit ke-900.

Tabel 6.5 *Gantt Chart Cycle Time Rute Existing*

Sub Rute	Jadwal Keberangkatan		Cycle Time																															
	290	295	300	315	330	345	360	375	390	405	420	435	450	465	480	495	510	525	540	555	570	585	600	615	630	645	660	675	690	705	720	900		
1																																		
2																																		

Perbandingan rute *existing* dengan rute yang direkomendasikan sementara pada penelitian ini adalah:

1. Rute *existing*

Total jarak yang ditempuh adalah 4611,1 km dengan menggunakan 44 kendaraan atau dengan kata lain terdapat 44 sub rute yang dijalankan. Terdapat 2 sub rute yang melanggar *time window constraint* penjelasan data terkait dapat dilihat pada Tabel 4.5.

2. Rute rekomendasi

Total jarak yang ditempuh adalah 4256.7 km dengan menggunakan 35 kendaraan atau dengan kata lain terdapat 35 sub rute yang dijalankan. Tidak ada sub rute yang melanggar *time window constraint*. Secara garis besar dapat dilihat pada *gantt chart* rute rekomendasi lebih padat dibandingkan dengan rute *existing*. Salah satu penyebabnya adalah diabaikannya rasio pemadatan pada kendaraan truk *compactor*.

6.3 Penugasan Truk *Compactor* untuk MSW *Collection* di Kota Surabaya

Berdasarkan rute rekomendasi yang dihasilkan pada tahap sebelumnya selanjutnya dilakukan penugasan kendaraan. Apabila murni tidak ada kebijakan lagi yang diambil maka kendaraan yang dibutuhkan adalah sebanyak 35 unit. Dan penugasan kendaraan terhadap suatu rute perjalanannya dapat dilihat pada Tabel 6.6.

Dari total kendaraan yang tersedia pada DKRTH saat ini yaitu 44 unit, rekomendasi penelitian ini hanya membutuhkan 35 unit kendaraan. Selisih dari keduanya adalah 9 unit. Sisa kendaraan ini nantinya akan digunakan sebagai mobil cadangan apabila permintaan tiba-tiba melonjak ataupun ada kendaraan yang *down*. Hal tersebut sangat penting bagi DKRTH Kota Surabaya karena pada saat saat tertentu kesulitan menjalankan proses MSW *collection* di Kota Surabaya terlebih ada tugas penyisiran jalan karena ada *event* tertentu. Disamping itu setiap harinya sampah harus selalu diambil karena jika tidak akan mengganggu pemukiman masyarakat Kota Surabaya.

Tabel 6.6 Penugasan Kendaraan Rekomendasi

No	NO POLISI	RUTE PERJALANAN	WAKTU KEBERANGKATAN (MENIT)	WAKTU KEMBALI (MENIT)	TOTAL JARAK (KM)
1	L9384NP	1 15 16 82 16 9 82 1	294	747	109.45
2	L9385NP	1 9 82 9 82 1	291	737	108.3
3	L9386NP	1 9 82 9 73 82 1	291	737	108.4
4	L9388NP	1 73 82 73 17 12 82 1	292	730	105.8
5	L9389NP	1 12 34 82 34 82 34 35 82 1	289	885	150.6
6	L9448NP	1 28 29 82 14 82 1	289	726	92.8
7	L9451NP	1 14 82 14 82 14 82 1	282	897	138.4
8	L9452NP	1 46 13 82 18 82 1	279	770	121.4
9	L9453NP	1 18 82 59 60 82 1	283	772	120.3
10	L9454NP	1 60 62 61 82 61 58 49 82 1	288	801	125.22
11	L9455NP	1 49 50 82 50 33 82 1	279	826	136.9
12	L9561NP	1 33 74 82 74 82 1	289	759	115.6
13	L9553NP	1 74 82 74 63 64 82 1	283	747	111.75
14	L9554NP	1 64 10 82 10 82 10 82 1	289	884	151.9
15	L9555NP	1 21 43 56 82 56 23 82 1	289	745	99.75
16	L9556NP	1 23 82 23 30 82 30 82 1	293	899	134.95
17	L9557NP	1 40 25 42 48 82 41 39 38 24 22 5 82 1	289	741	102.04
18	L9560NP	1 3 53 82 53 82 1	290	694	93.8
19	L9627NP	1 53 82 53 82 53 82 1	288	837	137.7
20	B9016SJA	1 27 82 27 4 70 82 1	293	744	94.25
21	L9688NP	1 70 82 70 69 82 69 82 1	281	860	138.9

Tabel 6.6 Penugasan Kendaraan Rekomendasi (Lanjutan)

No	NO POLISI	RUTE PERJALANAN	WAKTU KEBERANGKATAN (MENIT)	WAKTU KEMBALI (MENIT)	TOTAL JARAK (KM)
22	L9689NP	1 69 65 82 65 82 65 82 1	284	845	128.6
23	L9690NP	1 65 82 65 47 82 47 82 1	277	818	130.9
24	L9691NP	1 52 82 52 82 52 82 1	281	829	124.6
25	L9692NP	1 52 11 82 11 82 11 54 82 1	281	861	137.7
26	L9694NP	1 54 51 32 31 82 31 8 6 7 19 82 19 79 82 1	283	888	157.45
27	L9695NP	1 75 20 81 82 81 77 55 82 1	271	777	115.57
28	L9697NP	1 55 76 72 82 72 82 72 78 82 1	274	896	161.75
29	L9698NP	1 80 26 82 26 82 1	270	730	108.3
30	L9699NP	1 26 82 26 71 82 1	283	720	102.8
31	L9701NP	1 71 82 71 68 82 68 82 1	282	880	155.9
32	L9703NP	1 68 37 45 2 82 2 82 1	286	724	112.5
33	L9704NP	1 2 82 2 82 2 44 82 1	294	898	155.1
34	L9705NP	1 57 66 82 66 82 66 82 1	274	847	135
35	L9706NP	1 67 82 67 82 36 82 1	274	886	183.2

6.4 Analisa Sensitivitas

Pada sub bab ini akan dibahas mengenai analisa sensitivitas dari parameter yang dianggap memiliki pengaruh yang signifikan terhadap hasil keputusan yang dihasilkan oleh model. Parameter yang akan dilakukan Analisa sensitivitas adalah parameter jumlah *demand* yang akan di naikan nilai nya beberapa persen. Dari perubahan tersebut akan mempengaruhi jumlah kendaraan truk *compactor* yang digunakan dengan aturan dari berbagai *constraints* yang ada.

Tabel 6.7 Analisa Sensitivitas terhadap Perubahan *Demand*

Skenario	Total Demand	Fungsi Tujuan	Jumlah Kendaraan	Waktu Komputasi
Existing	2811	4611,1	44	-
Rekomendasi	2811	4256.7	35	15.6563
Naik 10% (LPS tetap)	3119	4863	39	17.6094
Naik 20% (LPS tetap)	3391	5223	43	18.2344
Naik 30% (LPS tetap)	3675	5826.1	48	18.2656
Naik 40% (LPS tetap)	3950	5965.6	52	18.7188
Naik 50% (LPS tetap)	4228	6494.2	55	18.7813

Berdasarkan Tabel 6.7 diatas diketahui bahwa terdapat perbedaan yang signifikan dari nilai fungsi tujuan yang dihasilkan diantara 5 skenario yang dibuat. Dengan mempertimbangkan *constraints* yang ada membuat perbedaan dalam *generate* sub rute. Ketika *demand* dinaikan beberapa persen *demand* LPS akan berubah sesuai dengan aturan kenaikan *demand*. Karena sub rute yang dihasilkan semakin naik maka kendaraan yang dibutuhkan akan semakin naik juga. Hal tersebut dipengaruhi dengan kapasitas maksimal kendaraan 35 kontainer dan adanya Batasan *time window* di depot DKRTH Surabaya.

Apabila jumlah kendaraan pada DKRTH Surabaya berjumlah 44 unit maka dengan kenaikan sekitar 20%, kendaraan yang dimiliki, diperkirakan masih mampu untuk mengambil sampah disetiap LPS dengan catatan semua kendaraan dalam keadaan tidak *down*. Apabila sampai 30% kenaikan *demand* yang ada maka dengan menggunakan 44 unit kendaraan diperkirakan tidak akan bisa mengambil seluruh sampah dalam satu hari operasional. Karena kenaikan jumlah produksi sampah pada saat tertentu beberapa kali terjadi, kebijakan yang ada pada DKRTH Surabaya adalah mengambil sampah lebih awal ataupun menambah jam kerja *driver* truk *compactor* sehingga pada menit ke-900 kendaraan belum sampai pada depot DKRTH Surabaya.

6.5 Analisis Hasil Penelitian

Parameter yang digunakan dalam uji model ACO untuk MSW *collection* di Kota Surabaya diperoleh dari uji coba beberapa parameter dengan 10 replikasi pada setiap kombinasinya. Hasil nya parameter yang terpilih merupakan parameter yang memiliki standart deviasi kecil dapat memberikan nilai fungsi tujuan yang

minimum. Rekomendasi rute yang dihasilkan oleh model algoritma ACO dengan parameter nilai evaporasi 0.9, jumlah semut 10 dan iterasi maksimal 1500 untuk MSW *collection* di Kota Surabaya menghasilkan nilai fungsi tujuan sebesar 4265.7 km. Nilai fungsi tujuan tersebut didapatkan dengan menggunakan jumlah kendaraan sebesar 35 unit. Rute tersebut dihasilkan dalam waktu komputasi sebesar 17.04 detik yang merupakan replikasi ke-5 dari 10 replikasi. Jumlah kendaraan yang digunakan masih dibawah jumlah kendaraan yang dimiliki oleh DKRTH pada saat ini. Sehingga dengan *demand* setiap LPS akan dapat terlayani semuanya dan juga masih memiliki truk *compactor* cadangan yang akan *stay* di depot.

Secara garis besar perbedaan yang dapat dilihat pada *gant chart* rute rekomendasi lebih padat dibandingkan dengan rute *existing*. Salah satu penyebabnya adalah diabaikannya rasio pemadatan pada kendaraan truk *compactor* dan beberapa kebijakan lain yang tidak dipertimbangkan atau secara sempurna dimasukkan dalam penelitian ini. Sebagai contoh waktu pengisian bensin, waktu tempuh yang didapatkan dari Google Maps dan antrian yang ada pada suatu *node*.

Penugasan kendaraan yang direkomendasikan dalam penelitian ini adalah menggunakan 35 unit kendaraan truk *compactor* sesuai dengan hasil uji menggunakan model ACO. Penugasan yang ada tidak melanggar *constraints* yang ada baik dari sisi kapasitas maupun *time window* dengan semua *demand* pada setiap LPS dapat dilayani semuanya.

Dari hasil analisa sensitivitas yang ada diketahui bahwa model algoritma yang digunakan dalam penelitian ini secara baik dapat beradaptasi dengan perubahan jumlah *demand* pada setiap LPS. Hasilnya diketahui bahwa apabila jumlah kendaraan pada DKRTH Surabaya berjumlah 44 unit maka dengan kenaikan sekitar 20%, kendaraan yang dimiliki, diperkirakan masih mampu untuk mengambil sampah disetiap LPS dengan catatan semua kendaraan dalam keadaan tidak *down*. Apabila sampai 30% kenaikan *demand* yang ada maka dengan menggunakan 44 unit kendaraan diperkirakan tidak akan bisa mengambil seluruh sampah dalam satu hari operasional.

BAB VII

KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan mengenai kesimpulan yang didapat pada penelitian berdasarkan tujuan yang ingin dicapai, hasil pengolahan data, dan analisa data, serta saran yang diajukan untuk penelitian selanjutnya.

7.1 Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut.

1. Telah dilakukan pengembangan model matematis dan algoritma ACO untuk pengambilan keputusan dalam menentukan rute kendaraan truk *compact*, jumlah kendaraan yang dibutuhkan dan jadwal keberangkatan untuk setiap rute dalam kasus MSW *collection* di Kota Surabaya.
2. Berdasarkan hasil uji model algoritma ACO untuk kasus didapatkan perbandingan antara rute *existing* dengan rute yang direkomendasikan. Membandingkan jarak total rute *existing* dengan rute yang direkomendasikan. Total jarak yang ditempuh oleh rute yang direkomendasikan adalah 4256.7 km dengan menggunakan 35 kendaraan. Hasil yang didapatkan lebih kecil dari rute *existing* selisih 354.4 km dan dengan selisih kendaraan 9 unit.
3. Penugasan kendaraan rekomendasi didasarkan oleh hasil uji model algoritma ACO. Kendaraan yang dibutuhkan adalah sebesar 35 unit truk *compact*. 9 kendaraan yang tidak melakukan perjalanan berfungsi sebagai cadangan ketika ada kenaikan *demand* pada LPS-LPS ataupun pengganti pada saat ada kendaraan yang *down*.
4. Berdasarkan hasil analisa sensitivitas didapatkan model algoritma yang digunakan dalam penelitian ini secara baik dapat beradaptasi dengan perubahan jumlah *demand* pada setiap LPS. Hasilnya sampai kenaikan 20% *demand* yang ada masih mampu untuk mengambil semua *demand* sampah pada setiap LPS yang ada.

7.2 Saran

Adapun saran pada penelitian ini adalah sebagai berikut:

1. Model dapat dikembangkan dengan *constraints* pengisian bensin secara *periodic*
2. Model dapat dikembangkan dengan *constraints* sistem antrian.
3. Mengintegrasikan jarak dan waktu tempuh secara *real time*.
4. Membuat *user interface* algoritma ACO sehingga dapat dengan mudah digunakan oleh orang awam sekalipun.

DAFTAR PUSTAKA

- Abdulkader, M. M., Gajpal, Y. & Elmekawy, T. Y., 2015. Hybridized Ant Colony Algorithm for The Multi Compartment Vehicle Routing Problem. *Applied Soft Computing*, Issue 37, pp. 196-203.
- Belien, J., Boeck, D. L. & Ackere, J. V., 2014. Municipal Solid Waste Collection and Management Problems: A Literature Review. *Transportation Science*, 48(1), pp. 78-102.
- Bell, J. E. & McMullen, P. R., 2004. Ant Colony Optimization Techniques for The Vehicle Routing Problem. *Advance Engineering Informations*, Issue 18, pp. 41-48.
- Benjamin, A. M. & Beasley, J. E., 2010. Metaheuristics for the Waste Collection Vehicle Routing Problem with Time Windows, Driver Rest and Multiple Disposal Facilities. *Computers & Operations Research*, Issue 37, pp. 2270-2280.
- Buhrkal, K., Larsen, A. & Ropke, S., 2012. The Waste Collection Vehicle Routing Problem with Time Windows in a City Logistics Context. *Social and Behavioral Sciences*, Issue 39, pp. 241-254.
- Das, S. & Bhattacharyya, B. K., 2015. Optimization of Municipal Solid Waste Collection and Transportation Routes. *Waste Managemant*.
- Dorigo, M. & Stutzle, T., 2004. *Ant Colony Optimization*. London: The MIT Press.
- Dror, M., Laporte, G. & Trudeau, P., 1994. Vehicle Routing with Split Deliveries. *Discrete Applied Mathematic*, Issue 50, pp. 239-254.
- El-Sherbeny, N. A., 2010. Vehicle Routing with Time Windows: An Overview of Exact, Heuristic and Metaheuristic Methods. *Science*, Issue 22, pp. 121-131.

- Fitria, L., Susanty, S. & Suprayogi, 2009. Penentuan Rute Truk Pengumpulan dan Pengangkutan Sampah di Bandung. *Jurnal Teknik Industri*, 11(1), pp. 51-60.
- Glover, F., 1990. Tabu Search: A Tutorial. *Interfaces*, 20(4), pp. 74-94.
- Hoornweg, D. & Tata, P. B., 2012. *WHAT A WASTE: A Global Review of Solid Waste Management*. Washington DC: The World Bank.
- Karadimas, N. V., Papatzelou, K. & Loumos, V. G., 2007. Optimal Solid Waste Collection Routes Identified by the Ant Colony System Algorithm. *Waste Management & Research*, Issue 25, pp. 139-147.
- Kuo, R., Kuo, P., Chen, Y. R. & Zulvia, F., 2016. application of metaheuristics based clustering algorithm to item assignment in a synchronized zone order picking system. *applied soft computing*, pp. 143 - 150.
- Li, J.-Q., Borenstein, D. & Mirchandani, P. B., 2008. Truck Scheduling for Solid Waste Collection in the City of Porto Alegre, Brazil. *Management Science*, Issue 36, pp. 1133-1149.
- Markov, I., Varone, S. & Bierlaire, M., 2015. Integrating a Heterogeneous Fixed Fleet and a Flexible Assignment of Destination Depots in the Waste Collection VRP with Intermediate Facilities. *Transportation Research Part B*, pp. 1-18.
- Mazzeo, S. & Loiseau, I., 2004. An Ant Colony Algorithm for the Capacitated Vehicle Routing. *Electronic Notes in Discrete Mathematics*, Issue 18, pp. 181-186.
- Reed, M., Yiannakou, A. & Evering, R., 2014. An Ant Colony Algorithm for the Multi-Compartment Vehicle Routing Problem. *Applied Soft Computing*, Issue 15, pp. 169-176.

- Reed, M., Yiannakou, A. & Evering, R., 2014. An Ant Colony Algorithm for the Multi-Compartment Vehicle Routing Problem. *Applied Soft Computing*, Issue 15, pp. 169-176.
- Santosa, B. & Ai, T. J., 2017. *Pengantar Metaheuristik Implementasi dengan Matlab*. Surabaya: ITS Tekno Sains.
- Santosa, B. & Willy, P., 2011. *Metoda Metaheuristik, Konsep dan Implementasi*. 1 ed. s.l.:Guna Widya.
- Sipra, A. T., Gao, N. & Sarwar, H., 2018. Municipal Solid Waste (MSW) Pyrolysis for Bio-Fuel Production: A Review of Effect of MSW Components and Catalysts. *Fuel Processing Technology*, Issue 175, pp. 131-147.
- Son, L. H. & Louati, A., 2016. Modelling Municipal Solid Waste Collection: A Generalized Vehicle Routing Model with Multiple Transfer Stations, Gather Sites and Inhomogeneous Vehicles in Time Windows. *Waste Managemant*, Issue 52, pp. 34-49.
- Suprayogi & Priyandari, Y., 2017. Tabu Search for the Vehicle Routing Problem with Multiple Trips, Time Windows and Simultaneous Delivery-Pickup. *Jurnal Teknik Industri*, 19(2), pp. 75-82.
- Talbi, E.-. G., 2009. *Metaheuristic from Design to Implementation*. Canada: John Wiley & Sons,.
- Tirkolae, E. B. et al., 2018. An Improved Ant Colony Optimization for the Multi-trip Capacitated Arc Routing Problem. *Computers and Electrical Engineering*, pp. 1-15.
- Toth, P. & Vigo, D., 2002. *The Vehicle Routing Problem*. Philadelphia: Siam.
- Wilck IV, J. H. & Cavalier, T. M., 2012. A Construction Heuristic for the Split Delivery Vehicle Routing Problem. *American Journal of Operations Research*, Issue 2, pp. 153-162.

(Halaman ini sengaja dikosongkan)

LAMPIRAN 1

Berikut ini merupakan alamat lengkap dari setiap *node* lokasi yang dikunjungi oleh truk *compactor*:

NO	Lokasi	Alamat
1	Depot DKRTH	Dinas Kebersihan dan Ruang Terbuka Hijau Kota Surabaya, Jl. Raya Menur No.31A, Manyar Sabrangan, Mulyorejo, Surabaya City, East Java 60285
2	LPS. Srikana	Jl. Srikana, Airlangga, Gubeng, Kota SBY, Jawa Timur 60286
3	Jl. Pandegiling	Jl. Pandegiling, Wonorejo, Tegalsari, Kota SBY, Jawa Timur
4	Jl. Urip Sumoharjo	Jl. Urip Sumoharjo, Embong Kaliasin, Genteng, Kota SBY, Jawa Timur
5	Jl. Embong Malang	Jl. Embong Malang, Genteng, Kota SBY, Jawa Timur
6	Jl. Karet	Jl. Karet, Bongkaran, Pabean Cantian, Kota SBY, Jawa Timur
7	JMP	Jembatan Merah Plaza, Jl. Taman Jayengrono No.2 - 4, South Krembangan, Krembangan, Surabaya City, East Java 60175
8	Kemayoran Baru	Jl. Kemayoran Baru, Krembangan, Kota SBY, Jawa Timur
9	LPS. Bratang	LPS Bratang, Baratajaya, Gubeng, Surabaya City, East Java 60284
10	LPS. Kaliwaron	Jl. Kaliwaron, Mojo, Gubeng, Kota SBY, Jawa Timur
11	LPS. Semut Kali	Jl. Semut Kali, Bongkaran, Pabean Cantian, Kota SBY, Jawa Timur
12	LPS. Ngagel	TPS Ngagel, Jl. Ngagel, Ngagel, Wonokromo, Surabaya City, East Java 60246
13	LPS. Gayung Pring	Jl. Gayungsari, Menanggal, Gayungan, Kota SBY, Jawa Timur
14	LPS Joyoboyo	Jl. Joyoboyo, Sawunggaling, Wonokromo, Kota SBY, Jawa Timur
15	LPS. Bok Tong	Manyar Sabrangan, Mulyorejo, Surabaya City, East Java
16	Taman Flora	Taman Flora Bratang Surabaya, Jl. Bratang Binangun, Baratajaya, Gubeng, Surabaya City, East Java 60284
17	Jl. Kalisumo	Jl. Ngagel Tama, Gubeng, Kota SBY, Jawa Timur 60283
18	LPS. Jemur Wonosari	Jemur Wonosari, Wonocolo, Surabaya City, East Java
19	LPS. Kertopaten	Jl. Kertopaten, Simolawang, Simokerto, Kota SBY, Jawa Timur
20	LPS Nyamplungan	Tps Nyamplungan, Jl. Nyamplungan No.219, Ampel, Semampir, Kota SBY, Jawa Timur 60151
21	LPS. Candipuro	Jl. Candipuro, Pacar Keling, Tambaksari, Kota SBY, Jawa Timur 60131
22	LPS Pasar Genteng	Genteng Market, Jl. Genteng Besar, Genteng, Surabaya City, East Java 60275

NO	Lokasi	Alamat
23	LPS Kayoon	TPS Kayon, Jl. Kayon, Embong Kaliasin, Genteng, Surabaya City, East Java 60271
24	Taman Apsari	Taman Apsari, Jl. Taman Apsari No.63, Embong Kaliasin, Genteng, Kota SBY, Jawa Timur 60271
25	Siola	siola, Jl. Genteng Kali No.123, Genteng, Surabaya City, East Java 60275
26	LPS. Tambak Rejo	Tambakrejo, Simokerto, Surabaya City, East Java
27	Pasar Keputran	Keputran Market, Jl. Keputran, Keputran, Tegalsari, Surabaya City, East Java 60265
28	Saluran Dinoyo	Jl. Serayu, Keputran, Tegalsari, Kota SBY, Jawa Timur
29	Saluran Bengawan	Jl. Bengawan, Darmo, Wonokromo, Kota SBY, Jawa Timur
30	Pasar Bunga Kayoon	Kayoon Flower Market, Jl. Kayon, Embong Kaliasin, Genteng, Surabaya City, East Java 60271
31	Saluran Gadukan	Jl. Moro Krembangan, Morokrembangan, Krembangan, Kota SBY, Jawa Timur 60178
32	Saluran Mungsing	Jl. Ikan Mungsing, Perak Bar., Krembangan, Kota SBY, Jawa Timur 60177
33	LPS Kejawaen Putih Tambak	Kejawaan Putih Tamba, Mulyorejo, Surabaya City, East Java
34	LPS. Ketampon	Jl. Ketampon, DR. Soetomo, Tegalsari, Kota SBY, Jawa Timur 60264
35	Taman Bungkul	Taman Bungkul, Jl. Taman Bungkul, Darmo, Wonokromo, Surabaya City, East Java 60241
36	Taman Sulawesi	Taman Persahabatan, Jl. Sulawesi No.67, Ngagel, Wonokromo, Kota SBY, Jawa Timur 60265
37	Taman Lansia	Taman Lansia, Jl. Kalimantan No.12, Gubeng, Surabaya City, East Java 60281
38	Taman Prestasi	Taman Prestasi, Jl. Ketabang Kali, Ketabang, Genteng, Kota SBY, Jawa Timur 60272
39	Taman Joging Trek Pusura	Jl. Yos Sudarso, Genteng, Kota SBY, Jawa Timur
40	PMI	Indonesian Red Cross Blood Bank, Jl. Embong Ploso No.7-15, Embong Kaliasin, Genteng, Surabaya City, East Java 60271
41	Balai Pemuda	Balai Pemuda Surabaya, Jl. Gubernur Suryo No.15, Embong Kaliasin, Genteng, Kota SBY, Jawa Timur 60271
42	DPRD	DPRD, Jl. Embong Malang No.25-31, Kedungdoro, Tegalsari, Kota SBY, Jawa Timur 60261
43	Stasiun Gubeng	Surabaya Gubeng, Pacar Keling, Tambaksari, Kota SBY, Jawa Timur 60131
44	PDAM	PDAM Surya Sembada Kota Surabaya, Jl. Mayjen Prof. Dr. Moestopo No.2, Pacar Keling, Tambaksari, Surabaya City, East Java 60131
45	Jl. Dharmawangsa	Jl. Dharmawangsa, Airlangga, Gubeng, Kota SBY, Jawa Timur

NO	Lokasi	Alamat
46	RS. Bhayangkara	RS Bhayangkara Surabaya H.S Samsoeri Mertojoso, Jalan Ahmad Yani No. 116, Ketintang, Gayungan, Kota SBY, Jawa Timur 60231
47	LPS. Merapi	Jl. Merapi, Sawahan, Kec. Sawahan, Kota SBY, Jawa Timur
48	LPS. Simpang Dukuh	Jl. Simpang Dukuh, Genteng, Kota SBY, Jawa Timur
49	LPS. Kendalsari	Jl. Raya Kendalsari, Rungkut, Kota SBY, Jawa Timur
50	LPS. Kebun Bibit Wonorejo	Kebun Bibit Wonorejo, Jl. Raya Kendalsari Blok RK No.70, Wonorejo, Rungkut, Surabaya City, East Java 60297
51	LPS. Peneleh	Jl. Makam Peneleh, Peneleh, Genteng, Kota SBY, Jawa Timur 60274
52	LPS. Demak Selatan	Jl. Demak Sel., Tembok Dukuh, Bubutan, Kota SBY, Jawa Timur 60173
53	LPS. Pandegiling	Jl. Pandegiling, Wonorejo, Tegalsari, Kota SBY, Jawa Timur
54	LPS. Sulung	Jl. Sulung, Alun-alun Contong, Bubutan, Kota SBY, Jawa Timur
55	LPS Endroso	Jl. Endroso, Wonokusumo, Semampir, Kota SBY, Jawa Timur
56	LPS Kangean	Taman Brmmain, Jl. Kangean, Sananwetan, Gubeng, Kota Blitar, Jawa Timur 60281
57	LPS Jeruk	Jl. Raya Jeruk, Lakarsantri, Kota SBY, Jawa Timur
58	LPS. Prapen 88	Jl. Raya Prapen No.88, Panjang Jiwo, Tenggilis Mejoyo, Kota SBY, Jawa Timur 60299
59	LPS Prapen DKK	Jl. Raya Jemursari, Prapen, Tenggilis Mejoyo, Kota SBY, Jawa Timur
60	LPS Raya Prapen	Jl. Raya Saronojiwo, Panjang Jiwo, Tenggilis Mejoyo, Kota SBY, Jawa Timur 60299
61	LPS Prapen Panjang Jiwo	Jl. Raya Prapen, Panjang Jiwo, Tenggilis Mejoyo, Kota SBY, Jawa Timur
62	LPS Prapen	Jl. Raya Saronojiwo, Panjang Jiwo, Tenggilis Mejoyo, Kota SBY, Jawa Timur 60299
63	Jl. Mulyorejo	Jl. Mulyorejo, Mulyorejo, Kota SBY, Jawa Timur 60115
64	LPS Mojo Arum	Jl. Mojo Arum I, Mojo, Gubeng, Kota SBY, Jawa Timur 60285
65	LPS Bukit Barisan	Jl. Bukit Barisan, Petemon, Kec. Sawahan, Kota SBY, Jawa Timur 60252
66	LPS. Babadan UNESA	Jl. Babatan UNESA, Kota SBY, Jawa Timur 60213
67	LPS. Siwalan Kerto	Siwalankerto, Wonocolo, Surabaya City, East Java
68	LPS. Legundi	Jl. Legundi, Ketabang, Genteng, Kota SBY, Jawa Timur 60272
69	LPS Kedung Anyar	Jl. Kedung Anyar I, Kota SBY, Jawa Timur 60251

NO	Lokasi	Alamat
70	LPS. Pasar Kembang	Jl. Kupang-Pasar Kembang, Wonorejo, Tegalsari, Kota SBY, Jawa Timur
71	LPS. Karang Gayam	Jl. Karang Gayam, Tambaksari, Kota SBY, Jawa Timur 60136
72	LPS. Wonokusumo Kidul	Jl. Wonokusumo Kidul, Semampir, Kota SBY, Jawa Timur
73	LPS. Kalibokor	Jl. Kalibokor, Kertajaya, Gubeng, Kota SBY, Jawa Timur
74	LPS. Sutorejo	Depo Sutorejo, Jl. Kalisari Timur, Kalisari, Mulyorejo, Kota SBY, Jawa Timur 60113
75	Lapangan Dwikora	Jl. Sawah Pulo Kulon, Ujung, Semampir, Kota SBY, Jawa Timur 60155
76	Jatisrono	Jl. Jatisrono, Ujung, Semampir, Kota SBY, Jawa Timur 60155
77	Endroso	Jl. Endroso, Wonokusumo, Semampir, Kota SBY, Jawa Timur
78	Tenggumung	Jl. Tenggumung Wetan, Kota SBY, Jawa Timur
79	Kertopaten	Jl. Kertopaten, Simolawang, Simokerto, Kota SBY, Jawa Timur
80	Kali Tebu	Jl. Platuk Donomulyo, Kenjeran, Kota SBY, Jawa Timur
81	Nyamplungan	Jl. Nyamplungan Balokan, Ampel, Semampir, Kota SBY, Jawa Timur 60151
82	LPA Benowo	Power House Landfill Gas Power Plant, Jl. Romokalisari I, Romokalisari, Benowo, Kota SBY, Jawa Timur 60192

LAMPIRAN 3

Berikut ini merupakan data waktu tempuh antar node:

Table with 82 columns and 82 rows, titled 'WAKTU TEMPUL DALAM SATUAN MENIT'. The table contains numerical data representing travel times between nodes, with some cells highlighted in yellow.

No	Nopol	Jarak & Waktu	Rute kendaraan														Jumlah Jarak	Jumlah Ritase	Gs	Service Time	Jadwal Keberangkatan	Jadwal kepulangan	Waktu Idle	
			1	73	82	73	82	1																
41	L9818NP	Rute	1	73	82	73	82	1										312	2	30	190	292	722	178
		Jarak		2.5	25.7	25.7	25.7	23.1										102.7						
		Waktu		8	46	46	46	64										210						
		Demand		35	0	35	0											70						
42	L9820NP	Rute	1	74	82	74	82	1										314	2	30	190	283	752	148
		Jarak		5.5	27.7	27.7	27.7	23.1										111.7						
		Waktu		17	56	56	56	64										249						
		Demand		35	0	35	0											70						
43	L9841NP	Rute	1	75	76	77	78	82	79	82	1							551	2	30	178	271	721	179
		Jarak		8.7	1.3	0.75	1.9	26	22.6	22.6	23.1							106.95						
		Waktu		29	7	4	8	50	40	40	64							242						
		Demand		6	6	15	1	0	30	0								58						
44	L9843NP	Rute	1	80	82	81	82	1										327	2	30	168	270	703	197
		Jarak		9.2	27	24.2	24.2	23.1										107.7						
		Waktu		30	57	42	42	64										235						
		Demand		18	0	30	0											48						

LAMPIRAN 5

Berikut ini merupakan koding dari algoritma ACO untuk TSP:

```
function best_rute = ACOAlg (jar, jum_semud, iter)
%jar adalah matriks koordinat
m = jum_semud; %jumlah semud
n = length(jar); %jumlah kota
e = 0.6; %evaporasi
alpha = 1; %indeks Tau
beta = 1; %indeks visibility
tau = 0.1*ones(n); %tho awal
for a = 1:iter %pengulangan
    e = 0.6-(0.6-0.2)*a/(iter); %reduksi evaporasi
    for i = 1:m %untuk semua semud
        rute(i,1) = 1; %dimulai dari kota 1
    end
    for i = 1:m %untuk semua semud
        h = 1./jar; %visibility
        for j = 1:n %untuk semua kota tujuan
            c = rute(i,j);
            h(:,c) = 0;
            %me-nol-kan baris kota terpilih agar tidak ke kota
            %sendiri
            koef1 = (tau(c,:).^alpha).*(h(c,:).^beta);
            koef2 = (sum(koef1));
            p = (1/koef2).*koef1; %probability
            r = rand; %bil random
            s = 0; %inital kumulatif peluang
            for k = 1:n
                s = s+p(k); %kumulatif prob kota ke k
                if r <= s
                    rute(i,j+1) = k; %pilihan kota k sebagai tujuan
                    break
                end
            end
        end
    end
    end
    rute;
    rute_semud = horzcat(rute,rute(:,1)); %dikembalikan ke depot
    for i = 1:m %untuk semua semud
        jarak = 0;
        for j = 1:n %untuk semua kota
            jarak= jarak + jar(rute_semud(i,j),rute_semud(i,j+1));
            %hitung jarak
        end
        f(i) = jarak;
    end
    jaraktot = f;
    [minf,idk] = min(f); %pilih jarak total terkecil
    ter = rute_semud(idk,:); %rute dengan jarak terkecil
    tau = (1-e)*tau; %evaporasi
```

```

for i = 1:m %untuk semua semut
    for j = 1:n %untuk semua kota
        dt = 1/f(i); %delta tau
        tau(rute_semut(i,j),rute_semut(i,j+1)) =
            tau(rute_semut(i,j),rute_semut(i,j+1))+dt; %tho baru
    end
end
[min_d(a),num] = min(jaraktot);
best_rute(a,:) = rute_semut(num,:);
end
best_rute = best_rute(end,:); %rute terbaik
jaraktot

```

LAMPIRAN 6

Berikut ini merupakan koding dari pembuatan sub rute VRP:

```
function [jarak_total, jaraksub] = VRP(alljar, jum_semud, iter,
kapasitas, D, wtempuh, maxt)
%alljar = jarak
%jum_semud = jumlah semut untuk alg ACO
%kapasitas = kapasitas truk
%iter = jumlah iterasi
%wtempuh = waktu tempuh ke node
%maxt = total waktu maksimal
jar = alljar(1:(length(alljar)-1),1:(length(alljar)-1)); %jarak
tanpa IF
start_time = cputime; %waktu mulai komputasi
ruteopt = feval('ACOAlg', jar, jum_semud, iter); %pencarian rute
[a,b] = size(ruteopt); %b = panjang rute
IF = length(alljar); %posisi IF
c = 2; %titik dari rute total
e = 1; %subrute ke-
while c < b %batas jumlah kota
    rute{e}(1) = 1; %dimulai dari depot
    f = 2; %titik dari subrute
    carried = 0; %muatan awal
    waktu = 300 - wtempuh(ruteopt(c-1),ruteopt(c));
    %waktu mulai berangkat
    z = 1; %indeks untuk melihat dari node atau dari IF
    while waktu < maxt && c < b %batas waktu
        if z < 2
            if (waktu + wtempuh(ruteopt(c-1),ruteopt(c)) +
                D(ruteopt(c)) + wtempuh(ruteopt(c),IF) + 60 + 30 +
                wtempuh(IF,1)) > maxt
                %pemilihan rute mempertimbangkan waktu ke node, ke IF
                dan balik ke depot tidak melebihi waktu max (jika
                dari node sebelumnya)
                break;
            end
        else
            if (waktu + wtempuh(IF,ruteopt(c)) + D(ruteopt(c)) +
                wtempuh(ruteopt(c),IF) + 60 + 30 + wtempuh(IF,1)) >
                maxt
                %pemilihan rute mempertimbangkan waktu ke node, ke IF
                dan balik ke depot tidak melebihi waktu max (jika
                dari IF)
                break;
            end
        end
        carried = 0; %refresh muatan
        while carried <= kapasitas && c < b %batasan kapasitas
            if carried >= kapasitas
                break;
            end
            if (carried + D(ruteopt(c))) > kapasitas
                %batasan sisa kapasitas mampu menambah demand
                if z < 2
```

```

        if (waktu + wtempuh(ruteopt(c-1),ruteopt(c)) +
D(ruteopt(c)) + wtempuh(ruteopt(c),IF) + 60 + 30
+ wtempuh(IF,1)) > maxt
%pemilihan rute mempertimbangkan waktu ke node,
ke IF dan balik ke depot tidak melebihi waktu max
(jika dari node sebelumnya)
            break;
        end
    else
        if (waktu + wtempuh(IF,ruteopt(c)) +
D(ruteopt(c)) + wtempuh(ruteopt(c),IF) + 60 + 30
+ wtempuh(IF,1)) > maxt
%pemilihan rute mempertimbangkan waktu ke node,
ke IF dan balik ke depot tidak melebihi waktu max
(jika dari IF)
            break;
        end
    end
end

if z < 2
waktu = waktu + wtempuh(ruteopt(c-1),ruteopt(c)) +
(kapasitas - carried); %adjust waktu tempuh ke
node jika dari node sebelumnya
else
waktu = waktu + wtempuh(IF,ruteopt(c)) +
(kapasitas - carried); %adjust waktu tempuh ke
node jika dari IF
end

rute{e}(f) = ruteopt(c); %memilih titik rute
D(ruteopt(c)) = D(ruteopt(c)) - (kapasitas -
carried); %demand sisa di c
carried = carried + (kapasitas - carried);
%adjust muatan hingga penuh
f = f + 1; %mengeser titik subrute
z = 1; %refresh z
else
if z < 2
    if (waktu + wtempuh(ruteopt(c-1),ruteopt(c)) +
D(ruteopt(c)) + wtempuh(ruteopt(c),IF) + 60 +
30 + wtempuh(IF,1)) > maxt
%pemilihan rute mempertimbangkan waktu ke
node, ke IF dan balik ke depot tidak melebihi
waktu max (jika dari node sebelumnya)
            break;
        end
    else
        if (waktu + wtempuh(IF,ruteopt(c)) +
D(ruteopt(c)) + wtempuh(ruteopt(c),IF) + 60 +
30 + wtempuh(IF,1)) > maxt
%pemilihan rute mempertimbangkan waktu ke
node, ke IF dan balik ke depot tidak melebihi
waktu max (jika dari IF)
            break;
        end
    end
end
end

```

```

rute{e}(f) = ruteopt(c); %memilih titik rute
carried = carried + D(ruteopt(c)); %adjust muatan
if z < 2
    waktu = waktu + wtempuh(ruteopt(c-
1),ruteopt(c))+ D(ruteopt(c));
    %adjust waktu tempuh ke node jika dari node
    sebelumnya
else
    waktu = waktu + wtempuh(IF,ruteopt(c))+
D(ruteopt(c)); %adjust waktu tempuh ke node jika
dari IF
end
f = f + 1; %mengeser titik subrute
c = c + 1; %mengeser titik rute total
z = 1; %refresh z
end
end
rute{e}(f) = IF; %menuju ke IF jika kapasitas penuh atau
waktu lewat
waktu = waktu + wtempuh(ruteopt(c),IF)+ 60; %adjust waktu
tempuh ke IF
f = f + 1; %mengeser titik subrute
z = z + 1; %berikutnya dari IF
end
rute{e}(f) = 1; %kembali ke depot
waktu = waktu + wtempuh(IF,1)+ 30; %adjust waktu tempuh ke
depot
e = e + 1; %mencari subrute berikutnya
end
q = length(rute);
for g = 1:q
    rut = rute{g}
    jaraksub(g) = feval('hitungjarak',rut,alljar); %menghitung
jarak subrute
end
jarak_total = sum(jaraksub); %menghitung total jarak subrute
comp_time = cputime - start_time %waktu komputasi

function totjarak = hitungjarak(x,alljar)
n = length(x); %panjang rute
totjarak = 0; %total jarak awal
for i = 1:n - 1
    totjarak = totjarak + alljar(x(i), x(i+1)); %perhitungan
jarak subrute
end
totjarak;

```

LAMPIRAN 7

Berikut ini merupakan Tabel Pembuatan sub rute 2 sampai 4 dengan metode *exact* TSP pada tahap peruteaanya:

Sub Rute 2	1	9	82	4	3	5	8	7	6	10	82	10	82	10	gs	1
Jarak tempuh antar node		2.7	27.5	20.4	1.7	2.9	2.8	1.7	0.65	7.8	24.5	7.8	24.5	24.5	30	23.1
Jarak kumulatif		2.7	30.2	50.6	52.3	55.2	58	59.7	60.35	68.15	92.65	100.45	124.95	0		148.05
Demand		35	0	10	13	2	1	2	1	70	0	64	0	29		0
Pickup demand		35	0	10	13	2	1	2	1	6	0	35	0	29		0
Demand kumulatif		35	0	10	23	25	26	28	29	35	0	35	0	0		0
load/unloading time		35	60	10	13	2	1	2	1	6	60	35	60	29		0
Waktu tempuh antar node		9	51	50	9	9	7	5	2	16	44	44	44	44		64
Cycle time	51	95	206	266	288	299	307	314	317	339	443	522	626	699	656	720
Keterangan		AMBIL		AMBIL	AMBIL	AMBIL	AMBIL	AMBIL	AMBIL	AMBIL		AMBIL		STOP		

Sub Rute 3	1	10	2	82	2	82	2	82	2	gs	1
Jarak tempuh antar node		4.8	3.8	25.3	25.3	25.3	25.3	25.3	25.3	30	23.1
Jarak kumulatif		4.8	8.6	33.9	59.2	84.5	109.8	135.1	0		158.2
Demand		29	140	0	134	0	99	0	64		0
Pickup demand		29	6	0	35	0	35	0	35		0
Demand kumulatif		29	35	0	35	0	35	0	0		0
load/unloading time		29	6	60	35	60	35	60	35		0
Waktu tempuh antar node		8	10	45	45	45	45	45	45		64
Cycle time	52	89	105	210	290	395	475	580	660	610	674
Keterangan		AMBIL	AMBIL		AMBIL		AMBIL		STOP		

Sub Rute 3	1	2	82	2	82	gs	1
Jarak tempuh antar node		2.1	25.3	25.3	25.3	30	23.1
Jarak kumulatif		2.1	27.4	52.7	78		101.1
Demand		64	0	29	0		0
<i>Pickup demand</i>		35	0	29	0		0
<i>Demand kumulatif</i>		35	0	29	0		0
<i>load/unloading time</i>		35	60	29	60		0
Waktu tempuh antar node		6	45	45	45		64
<i>Cycle time</i>	54	95	200	274	379	409	473
Keterangan		AMBIL		AMBIL			

LAMPIRAN 8

Berikut ini merupakan Tabel Pembuatan sub rute dengan metode ACO TSP pada tahap peruteanya:

Sub Rute 1	1	2	82	2	82	2	82	2	gs	1
Jarak tempuh antar node		2.1	25.3	25.3	25.3	25.3	25.3	25.3	30	23.1
Jarak kumulatif		2.1	27.4	52.7	78	103.3	128.6	0		151.7
Demand		140	0	105	0	70	0	35		0
Pickup demand		35	0	35	0	35	0	35		0
Demand kumulatif		35	0	35	0	35	0	0		0
load/unloading time		35	60	35	60	35	60	35		0
Waktu tempuh antar node		6	45	45	45	45	45	45		64
Cycle time	54	95	200	280	385	465	570	650	600	664
Keterangan		AMBIL		AMBIL		AMBIL		STOP		

Sub Rute 2	1	2	82	10	82	10	82	6	gs	1
Jarak tempuh antar node		2.1	25.3	20.4	20.4	20.4	20.4	21.5	30	23.1
Jarak kumulatif		2.1	27.4	47.8	68.2	88.6	109	130.5		132.1
Demand		35	0	70	0	35	0	1		0
Pickup demand		35	0	35	0	35	0	1		0
Demand kumulatif		35	0	35	0	35	0	1		0
load/unloading time		35	60	35	60	35	60	1		0
Waktu tempuh antar node		6	45	44	44	44	44	33		64
Cycle time	39	80	185	264	368	447	551	585	581	649
Keterangan		AMBIL		AMBIL		AMBIL		STOP		

Sub Rute 3	1	6	7	8	5	3	4	9	82	9	82	9	gs	1
Jarak tempuh antar node		7.3	0.65	1.7	2.8	2.9	1.7	4.2	27.5	27.5	27.5	27.5	30	23.1
Jarak kumulatif		7.3	7.95	9.65	12.45	15.35	17.05	21.25	48.75	76.25	103.75	131.25		154.35
Demand		1	2	1	2	13	10	140	0	134	0	99		0
Pickup demand		1	2	1	2	13	10	6	0	35	0	35		0
Demand kumulatif		1	3	4	6	19	29	35	0	35	0	35		0
load/unloading time		1	60	1	60	13	60	6	60	35	60	35		0
Waktu tempuh antar node		17	2	5	7	9	9	12	51	51	51	51		64
Cycle time	28	46	108	114	181	203	272	290	383	376	494	580	524	558
Keterangan		AMBIL	AMBIL	AMBIL	AMBIL	AMBIL	AMBIL	AMBIL		AMBIL		STOP		

Sub Rute 4	1	9	82	9	82	9	82	gs	1
Jarak tempuh antar node		2.7	25.3	25.3	25.3	25.3	25.3	30	23.1
Jarak kumulatif		2.7	28	53.3	78.6	103.9	129.2		152.3
Demand		99	0	64	0	29	0		0
Pickup demand		35	0	35	0	29	0		0
Demand kumulatif		35	0	35	0	29	0		0
load/unloading time		35	60	35	60	29	60		0
Waktu tempuh antar node		9	51	51	51	51	51		64
Cycle time	51	95	206	292	403	483	594	624	688
Keterangan		AMBIL		AMBIL		AMBIL			

LAMPIRAN 9

Berikut ini merupakan perhitungan 10 replikasi dalam pencarian nilai parameter terbaik:

Evaporasi	Jumlah Semut	Iterasi	Keterangan	Sampling										Min	Mean	SD	
				1	2	3	4	5	6	7	8	9	10				
0.5	5	500	Jarak (KM)	4484.9	4495.2	4473.5	4400.9	4363.7	4364.6	4377.4	4304	4547.4	4415.9	4304	4422.75	75.11152	
			Kendaraan (Unit)	36	36	36	36	35	36	35	35	37	35	35	35	35.7	0.674949
			Running time (Detik)	2.8125	2.8281	2.8438	2.9688	2.8281	2.7188	2.8125	2.8125	2.7656	2.75	2.7188	2.81407	0.067295	
		1000	Jarak (KM)	4379.7	4306.6	4390.2	4370.5	4413.7	4376.5	4381.4	4387	4351.4	4336.7	4306.6	4369.37	30.44799	
			Kendaraan (Unit)	36	35	36	36	36	36	35	35	36	35	35	35.6	0.516398	
			Running time (Detik)	5.4219	5.1875	5.1094	5.125	5.2969	5.2656	5.25	5.4063	5.375	5.5156	5.1094	5.29532	0.133895	
		1500	Jarak (KM)	4424	4484.4	4366.7	4453	4378.9	4372.2	4348.1	4507.8	4372.2	4336.7	4336.7	4404.4	59.38189	
			Kendaraan (Unit)	36	36	36	35	37	36	36	36	36	35	35	35.9	0.567646	
			Running time (Detik)	8.4844	7.625	7.8438	7.7656	8.0313	7.7344	8.2344	7.8594	8.0313	9.0719	7.625	8.06815	0.435169	
	10	500	Jarak (KM)	4450.3	4335.6	4446.4	4451.8	4449.8	4339.4	4348.8	4398.9	4399.9	4398.8	4335.6	4401.97	47.43574	
			Kendaraan (Unit)	36	35	35	36	36	35	36	37	36	36	35	35.8	0.632456	
			Running time (Detik)	5.5	5.5313	5.5156	5.2969	5.2656	5.25	5.4844	5.375	5.4688	5.5625	5.25	5.42501	0.117582	
		1000	Jarak (KM)	4407.3	4283.4	4533.6	4491.6	4373	4484.5	4426.9	4541	4369.9	4348.4	4283.4	4425.96	85.10881	
			Kendaraan (Unit)	36	35	36	35	35	36	36	35	35	36	35	35.5	0.527046	
			Running time (Detik)	10.6563	10.4844	10.8281	10.6875	10.75	11.1094	10.9531	10.2031	10.8594	10.3125	10.2031	10.68438	0.282678	
		1500	Jarak (KM)	4378.2	4321.3	4398.8	4315.5	4259.2	4358.2	4471.8	4333.2	4388.1	4372.7	4259.2	4359.7	57.39336	
			Kendaraan (Unit)	35	35	35	35	35	36	36	35	35	35	35	35.2	0.421637	
			Running time (Detik)	15.1406	15.5	15.1719	15.125	15.5	15.7188	15.4219	16.0469	15.9063	15.8125	15.125	15.53439	0.330223	
	15	500	Jarak (KM)	4459.4	4332	4315.1	4395.7	4362.6	4449.8	4350.1	4333.8	4464.6	4400.3	4315.1	4386.34	56.21131	
			Kendaraan (Unit)	36	37	35	36	36	35	35	36	36	35	35	35.7	0.674949	
			Running time (Detik)	8.1094	8.2031	7.9531	8.0781	8.4531	8.2344	17.5313	8.7031	9.2969	8.8594	7.9531	9.34219	2.907224	
		1000	Jarak (KM)	4392	4458.3	4408.7	4371.1	4326.9	4478.7	4302.4	4376.7	4416.4	4338.4	4302.4	4386.96	56.11778	
			Kendaraan (Unit)	35	36	35	35	36	36	35	35	36	35	35	35.4	0.516398	
			Running time (Detik)	17.2188	16.7656	17.1406	16.4063	17.5625	17.8438	19.5313	20.8594	17.3281	17.0469	16.4063	17.77033	1.373628	
1500		Jarak (KM)	4356.6	4456.8	4352.2	4365.2	4335	4295.7	4382.8	4376.7	4347.3	4401.7	4295.7	4367	42.73703		
		Kendaraan (Unit)	36	35	35	35	35	35	36	35	37	36	35	35.5	0.707107		
		Running time (Detik)	22.6094	23.2031	23.1563	23.3125	23.125	23.4063	23.0469	22.9844	23.4531	22.6094	22.6094	23.09064	0.293402		
0.6	5	500	Jarak (KM)	4389.7	4383.8	4322.3	4416	4421.7	4396	4437.7	4423.7	4371.4	4501.3	4322.3	4406.36	46.91226	
			Kendaraan (Unit)	37	36	36	36	36	36	36	35	36	35	35	35.9	0.567646	
			Running time (Detik)	3.125	2.8281	3.0313	2.9688	2.8281	2.7969	2.8594	2.8125	2.7969	2.75	2.75	2.8797	0.121051	
		1000	Jarak (KM)	4414.2	4429.2	4432.9	4549.6	4397.9	4363.1	4439.5	4365.1	4493.3	4353.6	4353.6	4423.84	61.31478	
			Kendaraan (Unit)	36	36	35	36	35	36	36	37	37	35	35	35.9	0.737865	
			Running time (Detik)	5.8125	5.4688	5.375	5.5156	5.5938	5.8125	5.7813	5.5313	5.4375	5.5313	5.375	5.58596	0.160574	
		1500	Jarak (KM)	4420.6	4397.2	4382.6	4443.1	4609	4381.9	4325.8	4359.3	4477.3	4415.4	4325.8	4421.22	78.50894	
			Kendaraan (Unit)	36	36	36	36	37	35	36	36	36	36	35	36	0.471405	
			Running time (Detik)	8.7813	8.5938	8.7656	8.5156	8.5313	8.5156	8.4531	8.5781	8.5469	8.7188	8.4531	8.60001	0.114644	
	10	500	Jarak (KM)	4464.8	4396.2	4394.9	4347.4	4374.5	4354.7	4332.9	4307.5	4379.1	4301.3	4301.3	4365.33	48.21902	
			Kendaraan (Unit)	35	37	35	35	35	35	35	35	36	35	35	35.3	0.674949	
			Running time (Detik)	5.875	6.2656	6.0313	5.9844	6.2969	6.1719	6.0156	15.4531	6.2813	5.7969	5.7969	7.0172	2.969097	
		1000	Jarak (KM)	4414.6	4398	4374.1	4476.5	4426.2	4431.2	4378	4471.6	4454.3	4310.4	4310.4	4413.49	50.73626	
			Kendaraan (Unit)	36	36	36	36	36	35	35	36	36	36	35	35.8	0.421637	
			Running time (Detik)	11.4375	11.5313	10.2031	10.3438	10.2031	10.125	10.9688	10.2031	10.3906	10.1719	10.125	10.55782	0.545478	
		1500	Jarak (KM)	4361.3	4277.8	4392	4447.8	4383.2	4363.5	4363.3	4477.2	4407.6	4374.1	4277.8	4384.78	53.86472	
			Kendaraan (Unit)	35	36	35	37	36	35	36	36	36	36	35	35.8	0.632456	
			Running time (Detik)	15.9219	15.3125	17.3281	15.4531	15.8281	15.9688	17.1875	15.4063	15.875	15.2031	15.2031	15.94844	0.74229	
	15	500	Jarak (KM)	4483.2	4500	4413.7	4375.4	4387.5	4383.5	4393.7	4384	4424.2	4474.5	4375.4	4421.97	46.84295	
			Kendaraan (Unit)	36	37	36	36	35	36	35	35	36	34	34	35.6	0.843274	
			Running time (Detik)	8.6563	8.7031	8.3125	8.3906	8.75	8.0625	8.2813	8.875	8.0313	8.1875	8.0313	8.42501	0.301158	
		1000	Jarak (KM)	4403.1	4359.4	4220	4398.6	4413.9	4322.3	4336.8	4426.9	4256.9	4233	4220	4337.09	77.3148	
			Kendaraan (Unit)	34	35	34	35	35	35	35	34	34	35	34	34.6	0.516398	
			Running time (Detik)	16.6875	16.4375	16.4688	16.8438	16.4531	16.0781	16.5781	16.3125	16.625	16.3906	16.0781	16.4875	0.212321	
1500		Jarak (KM)	4333.6	4387.5	4422.8	4384	4392.7	4313.8	4290.5	4383.5	4424.2	4393.7	4290.5	4372.63	44.99561		
		Kendaraan (Unit)	36	36	36	35	35	36	35	36	36	36	35	35.7	0.483046		
		Running time (Detik)	24.3594	26.0469	25.0469	26.375	26.1875	24.8906	25.125	25.7969	25.2188	26.7344	24.3594	25.57814	0.75906		

Evaporasi	Jumlah Semut	Iterasi	Keterangan	Sampling										Min	Mean	SD
				1	2	3	4	5	6	7	8	9	10			
0.7	5	500	Jarak (KM)	4586.2	4395.4	4528.5	4329.8	4395.3	4482.9	4483.7	4304.3	4329.5	4299.2	4299.2	4413.48	101.3818
			Kendaraan (Unit)	36	35	36	36	36	36	36	35	35	37	35	35.8	0.632456
			Running time (Detik)	2.8594	2.5781	2.7656	2.6719	2.6563	2.8438	2.6563	2.7969	2.4844	2.8438	2.4844	2.71565	0.126474
		1000	Jarak (KM)	4419.8	4388.7	4281.5	4428.5	4459.3	4452.2	4440.7	4338.7	4479.5	4466.2	4281.5	4415.51	62.65144
			Kendaraan (Unit)	36	35	36	36	36	36	36	36	36	36	35	35.9	0.316228
			Running time (Detik)	5.5	5.0938	5.0469	5.3906	5.3594	4.9844	5.0469	5.3281	5.1563	5.1563	4.9844	5.20627	0.175187
	1500	Jarak (KM)	4372.5	4466.9	4527.5	4390.5	4435.6	4371.3	4331.3	4438.8	4473.8	4469.9	4331.3	4427.81	60.08608	
		Kendaraan (Unit)	35	36	36	36	36	36	36	36	37	37	35	36.1	0.567646	
		Running time (Detik)	8.5	8.75	8.9531	8.8594	8.7813	8.2813	8.375	8.8281	8.5625	8.3594	8.2813	8.62501	0.239119	
	10	500	Jarak (KM)	4395.8	4323.1	4415.4	4401.9	4416.3	4478.7	4306.7	4414.5	4493	4357.3	4306.7	4400.27	59.64534
			Kendaraan (Unit)	37	36	36	37	36	36	35	35	36	35	35	35.9	0.737865
			Running time (Detik)	5.4844	5.4844	5.7656	5.7344	5.6563	5.3594	5.875	5.625	6.125	5.4844	5.3594	5.65939	0.226872
		1000	Jarak (KM)	4427.6	4396.5	4403.9	4485	4490.3	4399.9	4442.4	4429.8	4503.8	4312.5	4312.5	4429.17	56.54489
			Kendaraan (Unit)	36	35	35	36	37	36	37	36	36	36	35	36	0.666667
			Running time (Detik)	10.7969	10.8594	10.4688	10.4531	10.7656	10.7188	10.6094	10.7031	10.25	11.0469	10.25	10.6672	0.229533
	1500	Jarak (KM)	4424	4335.4	4320.7	4466.5	4360	4356.6	4358.2	4399	4395.4	4396.3	4320.7	4381.21	43.71106	
		Kendaraan (Unit)	36	16	35	36	35	35	36	36	36	35	16	33.6	6.203941	
		Running time (Detik)	16.7969	16.9531	17.0938	16.9844	17	17.125	16.9219	16.1094	17.2656	17	16.1094	16.92501	0.312995	
	15	500	Jarak (KM)	4326.2	4363.3	4452.1	4384.2	4298.8	4375.1	4306.4	4420	4289.5	4495	4289.5	4371.06	68.83136
			Kendaraan (Unit)	35	35	36	35	36	36	35	36	35	36	35	35.5	0.527046
			Running time (Detik)	8.5	8.4844	8.1094	7.8594	8.125	8.2188	8.0781	8.7813	8.1406	8.5156	7.8594	8.28126	0.277078
		1000	Jarak (KM)	4454	4344.6	4407.4	4297.7	4352.16	4496.3	4486.6	4312.4	4352.7	4403.3	4297.7	4390.716	70.41712
			Kendaraan (Unit)	35	35	35	35	35	36	36	35	36	35	35	35.3	0.483046
			Running time (Detik)	17.2188	16.2813	16.1875	16.0469	16.4219	16.0625	16.0938	16.9375	16.4531	16.8125	16.0469	16.45158	0.407946
1500	Jarak (KM)	4377.7	4442	4441.6	4363.9	4375	4368.6	4328.3	4481.5	4459.2	4485.3	4328.3	4412.31	55.73203		
	Kendaraan (Unit)	35	36	36	35	36	35	36	36	36	36	35	35.7	0.483046		
	Running time (Detik)	26.9063	25.8906	26.0781	27.6406	26.6406	28.0469	26.3438	25.6406	26.3594	25.9531	25.6406	26.55	0.780663		
0.8	5	500	Jarak (KM)	4524.2	4503.8	4337.1	4409.3	4412.7	4437.3	4.4428e+0	4484.3	4298.4	4471.9	4298.4	4431	75.61847
			Kendaraan (Unit)	36	35	35	36	37	37	36	36	36	37	35	36.1	0.737865
			Running time (Detik)	3.0781	2.8438	3.2188	2.8906	2.8281	2.7344	2.75	2.75	2.8906	2.7969	2.7344	2.87813	0.156393
		1000	Jarak (KM)	4413.6	4514	4319.9	4410	4500.3	4383.5	4476.2	4451.4	4507	4484.5	4319.9	4446.04	63.13138
			Kendaraan (Unit)	37	37	36	35	36	36	36	36	36	37	35	36.2	0.632456
			Running time (Detik)	5.5	5.5781	5.9063	6.3594	5.6719	5.7656	5.75	5.6406	5.8438	5.6719	5.5	5.76876	0.2398
	1500	Jarak (KM)	4522.7	4340.8	4376.2	4497	4428.5	4378.4	4365.5	4454.5	4458.6	4385.5	4340.8	4420.77	60.83216	
		Kendaraan (Unit)	37	36	36	36	35	36	36	36	37	36	35	36.1	0.567646	
		Running time (Detik)	8.4375	8.2188	8.4844	8.2344	8.3438	8.2188	7.9219	8.3281	8.125	7.8906	7.8906	8.22033	0.197436	
	10	500	Jarak (KM)	4558.3	4380.2	4418.1	4495.4	4327.2	4410.7	4327.4	4367.8	4301.1	4460.8	4301.1	4404.7	81.55338
			Kendaraan (Unit)	36	36	36	36	37	35	35	36	37	36	35	36	0.666667
			Running time (Detik)	5.5313	5.4375	5.2656	5.2656	5.3906	5.375	5.8438	5.4063	5.375	5.75	5.2656	5.46407	0.192804
		1000	Jarak (KM)	4486	4372.1	4363.9	4306	4453.3	4450.3	4403.6	4432.8	4396.9	4457	4306	4412.19	54.30986
			Kendaraan (Unit)	36	35	36	36	36	36	36	35	36	36	35	35.8	0.421637
			Running time (Detik)	11.2031	11.2031	11.1719	11.3438	11.6719	10.7188	11.6719	10.9844	10.9688	12.5313	10.7188	11.3469	0.511525
	1500	Jarak (KM)	4407.1	4280.3	4393.4	4349.7	4386.7	4509.8	4316.4	4371.8	4362.7	4388.2	4280.3	4376.61	60.57951	
		Kendaraan (Unit)	36	35	35	37	36	35	36	35	36	36	35	35.7	0.674949	
		Running time (Detik)	17.25	16.08	15.9063	16.8281	16.7969	17.2969	15.9219	18.0625	15.9688	15.9375	15.9063	16.60489	0.759122	
	15	500	Jarak (KM)	4390.6	4425.7	4425.4	4268.8	4422.5	4438.1	4351.8	4459.4	4279.4	4410.9	4268.8	4387.26	66.21262
			Kendaraan (Unit)	35	35	36	36	36	35	36	36	36	35	35	35.6	0.516398
			Running time (Detik)	12.7344	8.8594	12.1563	8.1875	11.0156	8.2969	9.5781	8.7656	10.5781	8.5469	8.1875	9.87188	1.65422
		1000	Jarak (KM)	4458	4440.9	4347	4405.9	4524.6	4361.8	4444.1	4458.4	4481	4354.3	4347	4427.6	58.92771
			Kendaraan (Unit)	35	36	36	36	36	36	36	36	36	36	35	35.9	0.316228
			Running time (Detik)	24.1875	17.2031	17.4219	16.6406	23.3281	16.25	16.9844	20.7031	16.2969	16.6719	16.25	18.56875	3.02437
1500	Jarak (KM)	4380.9	4391.5	4350.8	4377.6	4371	4322.3	4318.4	4389.6	4371.6	4395.8	4318.4	4366.95	27.70007		
	Kendaraan (Unit)	35	35	35	36	36	36	36	35	35	36	35	35.5	0.527046		
	Running time (Detik)	24.0313	25.0625	24.5313	24.375	26.25	4322.3	24.9375	24.1094	23.9375	25.7344	23.9375	454.5269	1358.997		

Evaporasi	Jumlah Semut	Iterasi	Keterangan	Sampling										Min	Mean	SD	
				1	2	3	4	5	6	7	8	9	10				
0.9	5	500	Jarak (KM)	4490.6	4440.9	4389.6	4404.6	4401.6	4353.7	4395.2	4469.4	4263.2	4410.9	4263.2	4401.97	63.06115	
			Kendaraan (Unit)	37	36	36	36	36	35	36	36	36	37	36	35	36.1	0.567646
			Running time (Detik)	4.0781	2.8594	2.7656	2.7188	2.8281	2.75	2.7813	3	3.0313	2.8594	2.7188	2.9672	2.9672	0.403589
		1000	Jarak (KM)	4398.1	4311.9	4500.2	4568.6	4406.7	4516	4507.3	4560.1	4494.8	4320.8	4311.9	4458.45	93.07886	
			Kendaraan (Unit)	36	36	35	35	36	36	36	37	36	36	35	35.9	0.567646	
			Running time (Detik)	5.7344	5.8906	6.2813	5.7969	5.8594	5.5938	6.1094	6.1094	5.7344	6.4063	5.5938	5.95159	0.263266	
		1500	Jarak (KM)	4439.4	4469.8	4564.6	4463.4	4351.8	4403	4488.1	4493.4	4488.9	4495.2	4351.8	4465.76	57.80908	
			Kendaraan (Unit)	36	37	36	36	36	36	36	36	37	36	36	36.2	0.421637	
			Running time (Detik)	9.0625	8.3125	8.7344	8.4375	8.5781	8.1875	8.5	8.6719	8.3438	8.2656	8.1875	8.50938	0.262928	
	10	500	Jarak (KM)	4379.5	4389.4	4354.6	4487.6	4504.2	4370.8	4375.9	4413.9	4316.5	4504	4316.5	4409.64	66.34153	
			Kendaraan (Unit)	35	36	35	36	35	36	36	35	36	36	35	35.6	0.516398	
			Running time (Detik)	8.2813	5.5781	7.6875	5.6094	8.625	5.6719	5.9531	6.125	5.7188	8.2344	5.5781	6.74845	1.285323	
		1000	Jarak (KM)	4403.3	4398.8	4393.7	4417.1	4459.3	4322.8	4363.4	4252.2	4256.8	4420.2	4252.2	4368.76	70.04765	
			Kendaraan (Unit)	35	36	35	35	36	35	36	35	36	36	35	35.5	0.527046	
			Running time (Detik)	12.2656	12.4063	11.8281	12.0938	11.8438	17.5625	18.875	12.1875	11.9219	16.25	11.8281	13.72345	2.726544	
		1500	Jarak (KM)	4336.9	4350	4306.4	4365.8	4409.9	4357.4	4368.7	4360.3	4356.6	4368.6	4306.4	4358.06	26.16326	
			Kendaraan (Unit)	36	36	35	35	37	35	36	36	36	36	35	35.8	0.632456	
			Running time (Detik)	16.0156	16.3438	16.0781	23.2188	16.625	16.375	20.2656	16.1719	23.7344	23.8906	16.0156	18.87188	3.505509	
	15	500	Jarak (KM)	4425.3	4482.2	4405.1	4375.9	4406.8	4330.1	4375	4460.2	4413.2	4295.9	4295.9	4396.97	55.8811	
			Kendaraan (Unit)	35	37	36	35	35	35	36	35	35	36	35	35.5	0.707107	
			Running time (Detik)	8.8906	8.7656	8.6563	8.1719	8.1875	8	7.9219	8.7813	7.9375	7.9844	7.9219	8.3297	0.395588	
		1000	Jarak (KM)	4341.5	4522.6	4329.6	4466	4297.7	4390.1	4415.8	4427.5	4441.4	4359.2	4297.7	4399.14	68.88442	
			Kendaraan (Unit)	35	36	36	36	35	36	36	36	36	37	35	35.9	0.567646	
			Running time (Detik)	17.6406	19.3281	16.7813	21.2656	23.3125	19.7813	19.9844	16.7813	22.0156	18.5469	16.7813	19.54376	2.197305	
1500		Jarak (KM)	4396.9	4306	4384.9	4403.2	4400.4	4346	4390.1	4277.3	4383.6	4372.3	4277.3	4366.07	43.04664		
		Kendaraan (Unit)	35	36	35	36	35	35	36	35	36	36	35	35.5	0.527046		
		Running time (Detik)	24.8594	23.3125	27.0156	31.5156	27.5781	31.2656	25.9688	29.5156	32.2344	31.8125	23.3125	28.50781	3.202708		

