



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS184853

OPTIMASI RUTE DISTRIBUSI PENGANGKUTAN SAMPAH DI KABUPATEN BANGKALAN DENGAN MENGUNAKAN ALGORITMA GENETIKA

OPTIMIZATION OF WASTE TRANSPORT DISTRIBUTION ROUTE IN BANGKALAN REGENCY USING GENETIC ALGORITHM

SARAH FAUZIYYAH SEKARSARI
NRP 0521154000076

Dosen Pembimbing
Edwin Riksakomara, S.Kom., MT.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR - IS1848523

OPTIMASI RUTE DISTRIBUSI PENGANGKUTAN SAMPAH DI KABUPATEN BANGKALAN DENGAN MENGUNAKAN ALGORITMA GENETIKA

SARAH FAUZIYAH SEKARSARI
NRP 0521154000076

Dosen Pembimbing
Edwin Riksakomara, S.Kom., MT.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknik Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

FINAL PROJECT - IS1848523

OPTIMIZATION OF WASTE TRANSPORT DISTRIBUTION ROUTE IN BANGKALAN REGENCY USING GENETIC ALGORITHM

SARAH FAUZIYAH SEKARSARI
NRP 0521154000076

Supervisor
Edwin Riksakomara, S.Kom., MT.

INFORMATION SYSTEMS DEPARTMENT
Information and Communication Technology Faculty
Sepuluh Nopember Institut of Technology
Surabaya 2019

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

**OPTIMASI RUTE DISTRIBUSI PENGANGKUTAN
SAMPAH DI KABUPATEN BANGKALAN DENGAN
MENGUNAKAN ALGORITMA GENETIKA**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

SARAH FAUZIYAH SEKARSARI
NRP. 0521154000076

Surabaya, Juli 2019

**KEPALA
DEPARTEMEN SISTEM INFORMASI**



Mahendrawati ER, S.t., M.Sc., Ph.D
NIP. 19761011 200604 2 001

Halaman ini sengaja dikosongkan

LEMBAR PERSETUJUAN

OPTIMASI RUTE DISTRIBUSI PENGANGKUTAN SAMPAH DI KABUPATEN BANGKALAN DENGAN MENGUNAKAN ALGORITMA GENETIKA

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

SARAH FAUZIYYAH SEKARSARI

NRP. 0521154000076

Disetujui Tim Penguji : Tanggal Ujian : 05 Juli 2019
Periode Wisuda : September 2019

Edwin Riksakomara, S.Kom, MT (Pembimbing I)

Wiwik Anggraeni, S.Si., M.Kom. (Penguji I)

Faizal Mahananto, S.Kom., M.Eng, Ph.D. (Penguji II)



Halaman ini sengaja dikosongkan

OPTIMASI RUTE DISTRIBUSI PENGANGKUTAN SAMPAH DI KABUPATEN BANGKALAN DENGAN MENGUNAKAN ALGORITMA GENETIKA

Nama : SARAH FAUZIYYAH
SEKARSARI
NRP : 0521 15 4000 0076
Departemen : SISTEM INFORMASI FTIK-ITS
Dosen Pembimbing : Edwin Riksakomara, S.Kom., MT.

ABSTRAK

Sampah adalah salah satu masalah klasik yang dihadapi berbagai wilayah di Indonesia, tetapi juga kompleks untuk ditangani. Salah satu wilayah Indonesia yang memiliki masalah tersebut adalah Kabupaten Bangkalan, Madura, Jawa Timur. Adanya peningkatan jumlah penduduk di Kabupaten Bangkalan menimbulkan meningkatnya volume sampah yang dihasilkan sehingga dikhawatirkan permasalahan sampah tersebut mengganggu aktivitas masyarakat sehari-hari.

Penanganan permasalahan sampah tersebut dapat diatasi dengan adanya pengelolaan sampah yang tepat dan melibatkan seluruh pemangku kepentingan, termasuk pemerintah daerah. Salah satu proses operasional pengelolaan sampah yang dikelola oleh pemerintah daerah adalah pengangkutan dan pengumpulan sampah yang sistematis. Sayangnya, rute pengangkutan dan pengumpulan sampah yang dilewati armada-armada pengangkut sampah masih bersifat subjektif dan cenderung menghasilkan rute yang panjang dan kurang optimal. Untuk mengurangi besarnya biaya operasional yang disebabkan oleh panjangnya rute pengangkutan dan pengumpulan sampah, maka diperlukan adanya optimasi pada rute tersebut.

Salah satu cara untuk mengoptimalkan rute armada-armada pengangkut sampah tersebut adalah dengan menerapkan penyelesaian Vehicle Routing Problem (VRP). Agar

penyelesaian VRP optimal, maka masing-masing armada harus melewati titik-titik tersebut tepat satu kali. Untuk menyelesaikan permasalahan VRP, diperlukan adanya algoritma sebagai panduan untuk mendapatkan rute yang optimal. Salah satu algoritma yang dapat digunakan adalah Algoritma Genetika.

Hasil yang diperoleh dari penelitian tugas akhir ini adalah adanya pengurangan jumlah jarak tempuh sebesar 10,66% dari total jarak tempuh seluruh armada sebesar 181,850 km berkurang menjadi 167,967 km. Dengan pengurangan rute ini dapat memangkas atau mengurangi biaya operasional untuk melakukan pengangkutan sampah.

Kata Kunci: Pengangkutan Sampah, Vehicle Routing Problem, Algoritma Genetika

OPTIMIZATION OF WASTE TRANSPORT DISTRIBUTION ROUTE IN BANGKALAN REGENCY USING GENETIC ALGORITHM

Name : SARAH FAUZIYYAH
SEKARSARI
NRP : 0521 15 4000 0076
Department : SISTEM INFORMASI FTIK-ITS
Supervisor : Edwin Riksakomara, S.Kom., MT.

ABSTRACT

Waste is a common problem in several regions in Indonesia, yet it is a complex problem to handle. This problem has been faced by Bangkalan regency, Madura, East Java. Since the total of Bangkalan population increased each year, the amount of its waste has increased significantly. This may cause interference with the population's convenience.

As the amount of Bangkalan Regency's waste increasing, the waste problem becomes more complex. This problem can be handled by proper waste management and involving all stakeholders, including the local government. One of the operational processes of waste management managed by the local government is integrated transportation and waste management. Unfortunately, waste collection routes are still subjective and quite less optimal. To reduce operational costs caused by collecting waste with less optimal routes, the optimization of those routes is needed.

Vehicle Routing Problem (VRP) is a combinatorial optimization that can optimize waste collection routes. In order to complete the optimal VRP, each vehicle must pass these points of locations exactly once. To solve VRP problems, it is necessary to have an algorithm as a guide to get the optimal route. Genetic Algorithm can be used as VRP's algorithm.

The results obtained from this final project are reduction in the number of the mileage of 10.66% from the total mileage of the entire fleet of 181.850 km reduced to 167.967 km. By reducing this route, it can cut or reduce operational costs to carry out waste collection.

Keywords: Waste Collection, Vehicle Routing Problem, Genetic Algorithm

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, berkat, dan hidayah-Nya yang telah memberikan anugerah dan tuntunan kepada penulis sehingga penulis dapat menyelesaikan tugas akhir dengan judul **“OPTIMASI RUTE DISTRIBUSI PENGANGKUTAN SAMPAH DI KABUPATEN BANGKALAN DENGAN MENGGUNAKAN ALGORITMA GENETIKA”** yang menjadi salah satu syarat kelulusan di Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Penyusunan tugas akhir ini senantiasa mendapatkan dukungan dari berbagai pihak baik dalam bentuk doa, motivasi, semangat, kritik, saran, serta berbagai bantuan lainnya. Untuk itu, secara khusus penulis akan menyampaikan ucapan terima kasih kepada:

1. Allah SWT yang telah memberikan hidayah serta atas ijin-Nya pula saya mampu menyelesaikan tugas akhir ini.
2. Segenap keluarga besar penulis, terutama kedua orang tua, Bapak Shoma Sari serta Ibu Dewie Sekar yang senantiasa mendoakan, memberikan motivasi, dan kebutuhan materiil maupun non-materiil sehingga penulis mampu menyelesaikan pendidikan S1 ini dengan baik. Serta adik penulis, Salma.
3. Ibu Mahendrawati ER, S.t., M.Sc., Ph.D., selaku Ketua Jurusan Sistem Informasi ITS, Bapak Nisfu Asrul Sani, S.Kom, M.Sc selaku KaProdi S1 Sistem Informasi ITS serta seluruh dosen pengajar beserta staf dan karyawan Jurusan Sistem Informasi, FTIK ITS Surabaya selama penulis menjalani kuliah.
4. Bapak Edwin Riksakomara, S.Kom, MT. selaku dosen pembimbing yang telah meluangkan waktu untuk membimbing, memberi arahan, dan memberikan ilmu kepada penulis selama pengerjaan tugas akhir ini.

5. Ibu Wiwik Anggraeni, S.Si., M.Kom. dan Bapak Faizal Mahananto, S.Kom., M.Eng, Ph.D. selaku dosen penguji yang telah memberikan masukan dan saran kepada penulis selama pengerjaan tugas akhir ini.
6. Bapak Dadang selaku Kepala Bidang Pengelolaan Sampah dan Limbah B3 DLH Bangkalan serta Bapak Ishak Sudibyo selaku Kepala DLH Bangkalan yang telah memberikan kesempatan dan bantuan kepada penulis dalam pengerjaan tugas akhir ini.
7. Teman penulis, Titis Adi, yang telah senantiasa membantu penulis dalam mengerjakan penelitian sehingga penulis mampu menyelesaikan tugas akhir ini.
8. Teman-teman Lannister khususnya Inge dan Andre, serta Mbak Linda yang telah menemani penulis dalam melakukan penelitian.
9. Serta semua pihak yang telah membantu dalam pengerjaan tugas akhir ini yang belum tersebut namanya.

Terima kasih atas segala bantuan, dukungan, serta doanya. Semoga Allah SWT senantiasa melimpahkan anugerah serta membalas kebaikan yang telah diberikan kepada penulis. Penulis pun menyadari bahwa tugas akhir ini memiliki kekurangan, oleh karena itu penulis mengharapkan saran dan masukan demi kebaikan penulis dan tugas akhir ini. Akhir kata, penulis berharap bahwa tugas akhir ini dapat memberikan kebermanfaatan.

Surabaya, Juli 2019
Penulis,

Sarah Fauziyyah Sekarsari

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
LEMBAR PERSETUJUAN.....	ix
LEMBAR PERSETUJUAN.....	ix
ABSTRAK.....	xi
ABSTRACT.....	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL.....	xxiii
DAFTAR KODE PROGRAM.....	xxv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	5
1.6. Relevansi.....	6
BAB II TINJAUAN PUSTAKA.....	9
2.1. Penelitian Sebelumnya.....	9
2.2. Dasar Teori.....	11
2.2.1. DLH Kabupaten Bangkalan.....	11
2.2.2. Pengelolaan Sampah Kabupaten Bangkalan.....	11
2.2.3. Optimasi.....	13
2.2.4. <i>Vehicle Routing Problem</i> (VRP).....	15
2.2.5. Algoritma Genetika.....	18
BAB III METODOLOGI.....	25
3.1. Diagram Tahapan Pelaksanaan Tugas Akhir.....	25
3.2. Tahapan Pelaksanaan Tugas Akhir.....	25
3.2.1. Identifikasi Masalah.....	25
3.2.2. Tinjauan Pustaka.....	26
3.2.3. Pengumpulan dan Pengelolaan Data.....	26
3.2.4. Pemodelan Permasalahan.....	27
3.2.5. Penerapan Algoritma Genetika.....	27

3.2.6.	Evaluasi	28
3.2.7.	Visualisasi Hasil Penerapan Algoritma Genetika pada Peta Digital	28
3.2.8.	Penyusunan Laporan Tugas Akhir	28
BAB IV	PERANCANGAN.....	29
4.1	Pengumpulan Data	29
4.1.1.	Lokasi Tempat Pembuangan Sampah (TPS) dengan Sistem <i>Stationary Container System</i> (SCS).....	29
4.1.2.	Rute Armada Truk Pengangkut Sampah dengan Sistem <i>Stationary Container System</i> (SCS)	30
4.2	Inisialisasi Algoritma Genetika terhadap Program Pencarian Rute Pengangkutan Sampah.....	30
4.3	Perancangan <i>Use Case Diagram</i> Aplikasi	32
4.3.1.	Penentuan <i>Actor</i> , <i>Use Case</i> , dan <i>Relation</i>	33
4.3.2.	Perancangan <i>Use Case Diagram</i>	33
BAB V	IMPLEMENTASI	37
5.1.	Implementasi Algoritma Genetika pada Program...37	
5.1.1.	Input Data	37
5.1.2.	Menentukan Jumlah Populasi dan Generasi....37	
5.1.3.	Inisialisasi Populasi	37
5.1.4.	Seleksi	38
5.1.5.	Rekombinasi/ <i>Crossover</i>	39
5.1.6.	Mutasi.....	39
5.1.7.	Menampilkan Hasil keluaran.....	40
5.2.	Pengembangan Visualisasi.....	40
5.2.1.	Basis Data.....	41
5.2.2.	Input Data Titik Lokasi	44
5.2.3.	Pengelolaan Algoritma Genetika.....	48
5.2.4.	Visualisasi Hasil Pengelolaan Algoritma Genetika	49
5.3	Antar Muka Aplikasi.....	52
5.3.1.	Index Antar Muka	53
5.3.2.	Input Data Antar Muka.....	53

5.3.3.	Tampilkan Hasil Antar Muka.....	58
5.4	Penjelasan Alur Kerja Sistem Aplikasi.....	58
5.4.1.	Index Alur Kerja	59
5.4.2.	Input Data Alur Kerja	59
5.4.3.	Tampilkan Hasil Alur Kerja.....	62
BAB VI HASIL DAN PEMBAHASAN		65
6.1.	Lingkungan Uji Coba	65
6.2.	Uji Fungsional dan Non-Fungsional.....	66
6.2.1.	Uji Fungsional.....	66
6.2.2.	Uji Non-Fungsional.....	66
6.3.	Uji Coba Program Pencarian Rute.....	68
6.4.	Optimasi Rute	71
6.5.	Perbandingan Hasil Program Algoritma Genetika ..	72
6.5.1.	Perbandingan Jarak Tempuh Rute	72
6.5.2.	Perbandingan Visualisasi Rute.....	74
6.6.	Hasil Rekomendasi Rute Baru.....	76
BAB VII KESIMPULAN DAN SARAN		77
7.1.	Kesimpulan	77
7.2.	Saran	77
DAFTAR PUSTAKA		79
BIODATA PENULIS		81
LAMPIRAN A: Lokasi Tempat Pembuangan Sampah (TPS) dengan Sistem Stationary Container System (SCS).....		83
LAMPIRAN B: Rute Armada Truk Pengangkut Sampah dengan Sistem Stationary Container System (SCS).....		85
LAMPIRAN C: PERBANDINGAN VISUALISASI RUTE.....		87
1.	Perbandingan Rute Armada ke-1	87
2.	Perbandingan Rute Armada ke-2	89
3.	Perbandingan Rute Armada ke-3	92
4.	Perbandingan Rute Armada ke-4	94
5.	Perbandingan Rute Armada ke-5	96
6.	Perbandingan Rute Armada ke-6	98

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1 Relevansi Usulan Tugas Akhir dengan Bidang- Bidang Keilmuan Lab. RDIB	7
Gambar 2.1 Hauled Container System (HCS).....	12
Gambar 2.2 Stationary Container System (SCS)	13
Gambar 2.3 Ilustrasi VRP	16
Gambar 2.4 Ilustrasi Populasi, Gen, dan Individu atau Kromosom	19
Gambar 2.5 Ilustrasi Rekombinasi (Crossover)	21
Gambar 2.6 Proses Algoritma Genetika.....	23
Gambar 3.1 Diagram Tahapan Pelaksanaan Tugas Akhir	25
Gambar 4.1 Ilustrasi Komponen Algoritme Genetika terkait Pengangkutan Sampah.....	31
Gambar 4.2 Perancangan Use Case Diagram.....	34
Gambar 5.1 Basis Data Aplikasi	41
Gambar 5.2 Tampilan Index.....	53
Gambar 5.3 Tampilan Input Data.....	54
Gambar 5.4 Tampilan Tambah Data	55
Gambar 5.5 Tampilan Upload Data	57
Gambar 5.6 Tampilan Hasil	58
Gambar 5.7 Diagram Aplikasi Sistem pada Halaman Index..	59
Gambar 5.8 Diagram Aplikasi Sistem pada Halaman Input Data.....	60
Gambar 5.9 Diagram Aplikasi Sistem pada Halaman Tampilkan Hasil	63
Gambar 6.1 Diagram uji coba parameter $\alpha = 0,05$	70
Gambar 6.2 Diagram uji coba parameter $\alpha = 0,5$	70
Gambar 6.3 Diagram uji coba parameter $\alpha = 1$	71
Gambar 6.4 Contoh Visualisasi Rute Saat Ini	75
Gambar 6.5 Contoh Visualisasi Rute Baru	75

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya 1	9
Tabel 2.2 Penelitian Sebelumnya 2	10
Tabel 4.1 Lokasi Tempat Pembuangan Sampah (TPS) dengan Sistem Stationary Container System (SCS).....	29
Tabel 4.2 Rute Armada Truk Pengangkut Sampah.....	30
Tabel 4.3 Use Case UC1	34
Tabel 4.4 Use Case UC2	35
Tabel 4.5 Use Case UC3	36
Tabel 5.1 Tabel 'ga1'	41
Tabel 5.2 Tabel 'ga2'	42
Tabel 5.3 Tabel Data yang Disimpan.....	46
Tabel 6.1 Lingkungan Uji Coba Perangkat Keras.....	65
Tabel 6.2 Lingkungan Uji Coba Perangkat Lunak.....	65
Tabel 6.3 Uji Fungsional.....	66
Tabel 6.4 Uji Non-Fungsional Skalabilitas	66
Tabel 6.5 Uji Non-Fungsional Kompatibilitas terhadap Sistem Operasi.....	67
Tabel 6.6 Uji Non-Fungsional Kompatibilitas terhadap Penjelajah Web	67
Tabel 6.7 Uji Non-Fungsional Kompatibilitas terhadap Perangkat Keras	67
Tabel 6.8 Uji Non-Fungsional Reliabilitas.....	67
Tabel 6.9 Hasil Uji Coba dengan Mengubah Parameter	68
Tabel 6.10 Hasil Uji Coba Program	71
Tabel 6.11 Perbandingan Jarak Tempuh Rute.....	72
Tabel 6.12 Perbandingan Total Jarak Tempuh Rute	74
Tabel 6.13 Hasil Rekomendasi Rute Baru	76

Halaman ini sengaja dikosongkan

DAFTAR KODE PROGRAM

Kode Program 5.1 Input Data	37
Kode Program 5.2 Menentukan Jumlah Populasi dan Generasi	37
Kode Program 5.3 Inisialisasi Populasi.....	38
Kode Program 5.4 Seleksi.....	38
Kode Program 5.5 Inversi variabel 'i_distances'	39
Kode Program 5.6 Rekombinasi/Crossover	39
Kode Program 5.7 Mutasi	40
Kode Program 5.8 inisialisasi Variabel 'genbest'	40
Kode Program 5.9 Keluaran Variabel 'genbest'	40
Kode Program 5.10 Konfigurasi Basis Data	44
Kode Program 5.11 Penambahan Data Titik.....	45
Kode Program 5.12 Penyimpanan Data Titik	45
Kode Program 5.13 Pemasukan Data yang ke dalam Basis Data	46
Kode Program 5.14 Upload Data Titik	47
Kode Program 5.15 Penghapusan Data Titik.....	48
Kode Program 5.16 Pengambilan Data dari Basis Data.....	49
Kode Program 5.17 Penambahan Data Hasil pada Basis Data	50
Kode Program 5.18 Penggabungan tabel 'ga1' dengan 'ga2' ..	50
Kode Program 5.19 Google Maps API Key.....	50
Kode Program 5.20 Pengambilan Data untuk Visualisasi	51
Kode Program 5.21 Penyusunan Array '\$result'	51
Kode Program 5.22 Pengaturan Google Maps API	51
Kode Program 5.23 Penyusunan Variabel '\$stmt'	55

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Pada bagian pendahuluan menyajikan mengenai latar belakang, perumusan masalah, batasan, tujuan dari penelitian, serta relevansi tugas akhir. Uraian pada bab ini membantu untuk memberikan gambaran umum mengenai permasalahan dan pemecahan masalah Tugas Akhir ini.

1.1. Latar Belakang

Sampah adalah salah satu masalah klasik yang dihadapi berbagai wilayah di Indonesia, tetapi juga kompleks untuk ditangani. Salah satu permasalahan yang cukup sering terjadi mengenai sampah adalah meningkatnya volume sampah yang dihasilkan di wilayah-wilayah di Indonesia. Per tahun 2017, setiap orang diperkirakan menghasilkan 0,7 kilogram sampah setiap harinya. Dengan mempertimbangkan jumlah penduduk Indonesia, sampah yang dihasilkan mampu mencapai puluhan bahkan ratusan juta ton per tahunnya dan akan terus meningkat seiring meningkatnya jumlah penduduk Indonesia [1]. Selain jumlah penduduk yang terus meningkat, peningkatan volume sampah juga terjadi akibat faktor geografis, sosial, ekonomi, budaya, serta kemajuan teknologi [2].

Salah satu wilayah Indonesia adalah Kabupaten Bangkalan yang terletak di bagian barat Pulau Madura, Jawa Timur. Kabupaten Bangkalan berbatasan dengan Laut Jawa, Selat Madura, dan Kabupaten Sampang. Kabupaten ini memiliki luas lebih dari tiga kali Kota Surabaya, yakni 1.260,14 kilometer persegi dan dibagi menjadi 18 kecamatan berisi 273 desa dan 8 kelurahan [3]. Jumlah penduduk di Kabupaten Bangkalan pada tahun 2017 adalah sebesar 970.894 jiwa, di mana jumlah penduduk tersebut telah mengalami kenaikan dibanding tahun sebelumnya (2016) dengan jumlah penduduk sebesar 962.773 jiwa [3]. Peningkatan jumlah penduduk di Kabupaten Bangkalan ini semakin pesat sejak munculnya suatu kemajuan teknologi di bidang transportasi, yakni adanya Jembatan

Suramadu yang menghubungkan Kota Surabaya di Pulau Jawa dan Kabupaten Bangkalan di Pulau Madura [4].

Adanya peningkatan jumlah penduduk di Kabupaten Bangkalan cenderung akan mengakibatkan meningkatnya volume sampah yang dihasilkan. Menurut Kepala Dinas Lingkungan Hidup Kabupaten Bangkalan 2017, Ishak Sudiby, di tahun 2017 volume sampah meningkat 100 persen dibandingkan tahun sebelumnya, utamanya sampah yang berasal dari rumah makan yang menjamur karena adanya pertambahan penduduk maupun pengunjung di Kabupaten Bangkalan [5]. Selain itu, tidak menutup kemungkinan volume sampah di Kabupaten Bangkalan akan terus meningkat seiring berjalannya waktu [6]. Penumpukan sampah tersebut dikhawatirkan akan mengganggu aktivitas masyarakat sehari-hari.

Penanganan permasalahan sampah tersebut dapat diatasi dengan adanya pengelolaan sampah yang tepat dan menyeluruh. Menurut Koordinator Kemitraan Kota Hijau, Nirwono Joga, pengelolaan sampah harus melibatkan seluruh pemangku kepentingan, mulai dari pengelolaan di sumber penghasil sampah hingga ke pemrosesan terakhir. Salah satu pihak yang ikut andil dalam mengelola sampah adalah pemerintah daerah. Pemerintah daerah diharapkan mampu mengatur kebijakan pengelolaan sampah, baik secara strategis maupun operasional, sesuai kapasitas daerah yang dikelola [7]. Salah satu proses operasional pengelolaan sampah yang dikelola oleh pemerintah daerah adalah pengangkutan dan pengumpulan sampah yang sistematis.

Pengangkutan dan pengumpulan sampah dimulai dari mengumpulkan sampah di masing-masing rumah tangga, kemudian dikumpulkan di TPS terdekat agar selanjutnya diangkut oleh armada pengangkut sampah. Setelah armada tersebut mengangkut sampah di TPS-TPS yang telah ditentukan, kumpulan sampah tersebut akan diantar ke TPA (Tempat Pembuangan Akhir). Masing-masing armada telah dialokasikan untuk menangani pengambilan sampah dari TPS

menuju TPA. Berdasarkan lokasi TPS itulah maka setiap armada akan menentukan rute-rute yang harus dilalui. Namun rute yang dilalui masing-masing armada masih berdasarkan subjektivitas pengemudi armada sehingga rute yang digunakan belum menggunakan sistem khusus.

Sering kali pengemudi armada pengangkut sampah memilih rute berdasarkan inisiatifnya sendiri, yang secara umum belum didasarkan pada suatu konsep tertentu. Hal ini dapat menghasilkan rute pengangkutan sampah yang panjang. Semakin panjang rute pengangkutan sampah, semakin besar pula biaya yang dikeluarkan. Untuk mengurangi biaya operasional yang tinggi, maka diperlukan adanya optimasi pada rute tersebut.

Rute optimal didapatkan dari rute terpendek bagi masing-masing armada untuk mengumpulkan sampah dari TPS-TPS yang telah ditentukan menuju TPA. Salah satu cara untuk menemukan rute terpendek armada adalah dengan menerapkan penyelesaian *Vehicle Routing Problem* atau VRP.

VRP merupakan suatu permasalahan yang bertujuan untuk mencapai kondisi optimal sehingga mampu meminimalisir pengeluaran yang dibutuhkan. Dalam VRP, pengeluaran yang dimaksud berupa jarak rute yang harus ditempuh atau biaya yang dikeluarkan untuk menempuh rute tersebut. VRP memiliki satu atau lebih depot, beberapa armada, node atau titik yang harus dikunjungi masing-masing armada, serta jalur atau rute yang akan digunakan armada agar sampai di titik-titik yang telah disesuaikan. Agar penyelesaian VRP optimal, maka masing-masing armada harus melewati titik-titik tersebut tepat satu kali [8].

Permasalahan VRP dapat diselesaikan dengan menggunakan algoritma sebagai panduan untuk mendapatkan rute yang optimal. Salah satu algoritma yang dapat digunakan adalah Algoritma Genetika. Dengan adanya implementasi VRP dengan algoritma genetika, diharapkan pengangkutan dan pengumpulan sampah dapat berjalan secara optimal sehingga

permasalahan penumpukan sampah di TPS-TPS dapat ditekan. Berdasarkan penelitian sebelumnya, *Using Genetic Algorithm in Implementing Capacitated Vehicle Routing Problem* oleh Mazin Abed Mohammed, algoritma genetika merupakan algoritma yang efisien untuk digunakan dalam penyelesaian permasalahan VRP, utamanya CVRP. Dari penelitian lainnya, *Vehicle-Routing Problem Optimization for Municipal Solid Waste Collection using Genetic Algorithm : The Case of Southern Nablus City* oleh Ramiz Assaf dan Yahya Saleh, penerapan penyelesaian VRP dengan algoritma genetika mampu menghemat biaya yang dikeluarkan hingga 66%. Hal ini membuktikan bahwa luaran dari performa algoritma genetika tersebut sangat membantu dalam menyusun rute optimal [8] [9].

Tugas akhir ini bertujuan untuk mengimplementasikan penyelesaian VRP dengan algoritma genetika pada pengangkutan sampah di Kabupaten Bangkalan, Madura, Jawa Timur dengan menghasilkan rute terpendek yang perlu dilalui sehingga mampu menghasilkan rute yang optimal.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, permasalahan yang akan diangkat pada tugas akhir ini adalah sebagai berikut:

- 1) Bagaimana penerapan penyelesaian VRP (*Vehicle Routing Problem*) dengan algoritma genetika ke dalam sistem optimasi rute pengangkutan sampah di Kabupaten Bangkalan?
- 2) Apakah hasil optimasi rute pengangkutan sampah di Kabupaten Bangkalan lebih optimal dibandingkan rute pengangkutan sebelum dioptimasi?

1.3. Batasan Masalah

Batasan masalah tugas akhir ini adalah sebagai berikut:

- 1) Rute pengangkutan sampah yang dioptimasi merupakan rute pengangkutan sampah yang digunakan oleh Dinas Lingkungan Hidup Kabupaten Bangkalan.

- 2) Data yang digunakan adalah data primer dari Dinas Lingkungan Hidup Pemerintah Kabupaten Bangkalan yang dikumpulkan pada tahun 2019.
- 3) Metode yang digunakan adalah penyelesaian VRP (*Vehicle Routing Problem*) dengan algoritma genetika.
- 4) Visualisasi dilakukan pada peta digital.
- 5) Pengangkutan sampah tidak dipengaruhi kondisi lain/tidak biasa yang mempengaruhi seperti jalan sempit, macet, penutupan jalan karena demo, atau armada mengalami kerusakan.
- 6) Pengangkutan sampah tidak dipengaruhi oleh waktu dan dilakukan selama jam kerja.
- 7) Pengangkutan sampah tidak membahas kebijakan politis terkait dengan pembuangan sampah.
- 8) Algoritma genetika hanya diterapkan pada pengangkutan sampah dengan sistem *Stationary Container System* (SCS) yang dilakukan dengan *Dump Truck*.

1.4. Tujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah sebagai berikut:

- 1) Menerapkan VRP (*Vehicle Routing Problem*) dengan algoritma genetika untuk pengangkutan sampah di Kabupaten Bangkalan.
- 2) Mengoptimasikan rute pengangkutan sampah di Kabupaten Bangkalan.

1.5. Manfaat Penelitian

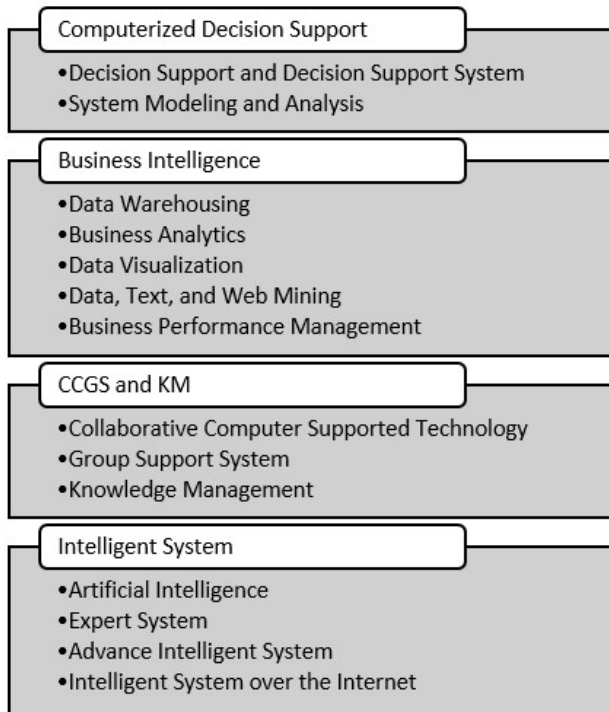
Manfaat yang dapat diperoleh dari tugas akhir ini adalah sebagai berikut:

- 1) Mendapatkan rute terpendek untuk pengangkutan sampah di seluruh TPS (Tempat Pembuangan Sementara) di Kabupaten Bangkalan.
- 2) Meminimalisir penggunaan sumber daya untuk pengangkutan sampah di Kabupaten Bangkalan (armada, bahan bakar, karyawan, dan lain-lain).

1.6. Relevansi

Tugas akhir ini memiliki relevansi dengan matakuliah yang ada di Departemen Sistem Informasi, Fakultas Teknologi dan Informasi, ITS, yakni mata kuliah Riset Operasi dan Optimasi Kombinatorik. Mata kuliah Riset Operasi adalah mata kuliah matematika terapan yang memaparkan penggunaan model-model matematika, statistika, dan algoritma untuk mendapatkan solusi optimal pada suatu permasalahan. Solusi optimal dalam Riset Operasi umumnya berupa nilai maksimal atau nilai minimal dari sebuah fungsi objektif yang disesuaikan dengan permasalahan. Mata kuliah Optimasi Kombinatorik dan Heuristik, mata kuliah kedua yang memiliki relevansi dengan tugas akhir ini, merupakan mata kuliah lanjutan dari Riset Operasi. Matakuliah ini berfokus pada permasalahan kombinatorik yang menghasilkan solusi optimum dari percobaan terhadap seluruh kemungkinan yang ditemukan. Permasalahan kombinatorik yang dimaksud adalah permasalahan yang berhubungan dengan pemilihan (seleksi), penyusunan, perangkaian, dan aktivitas lainnya dari kumpulan objek.

Berdasarkan kedua mata kuliah tersebut, tugas akhir ini sesuai dengan ranah penelitian di Laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB) yang ada di Departemen Sistem Informasi ITS, seperti yang terlihat pada Gambar 1.1. Dari empat bidang ilmu yang dicakup Lab. RDIB, tugas akhir ini mencakup dua bidang ilmu, yakni bidang ilmu sistem pendukung keputusan terkomputerisasi (*Computerized decision support*) dan sistem cerdas (*Intelligent system*).



Gambar 1.1 Relevansi Usulan Tugas Akhir dengan Bidang-Bidang Keilmuan Lab. RDIB

Halaman ini sengaja dikosongkan

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai penelitian sebelumnya dan dasar teori yang menjadi acuan dalam pengerjaan tugas akhir. Penelitian sebelumnya memberikan gambaran mengenai apa saja yang telah dilakukan dan dasar teori memberikan gambaran secara umum dari tugas ini.

2.1. Penelitian Sebelumnya

Tabel 2.1 dan Tabel 2.2 adalah penelitian-penelitian sebelumnya yang berkaitan dengan tugas akhir ini.

Tabel 2.1 Penelitian Sebelumnya 1

Judul Paper	<i>Using Genetic Algorithm in Implementing Capacitated Vehicle Routing Problem</i>
Penulis ; tahun	Mazin Abed Mohammed, Moh. Sharifuddin Ahmad, Salama A. Mostafa ; 2012
Deskripsi umum penelitian	Penelitian ini berfokus pada penggunaan algoritma genetika pada <i>Capacitated Vehicle Routing Problem</i> (CVRP). Objektif yang digunakan pada penelitian ini adalah ; menemukan solusi atas suatu permasalahan CVRP dengan algoritma genetika, mengimplementasi algoritma genetika pada CVRP dalam optimasi layanan armada pengangkut, serta meminimalisir jarak, waktu, dan biaya dalam rute yang digunakan dalam permasalahan CVRP dengan algoritma serupa. Impelementasi algoritma genetika pada penyelesaian CVRP menggunakan aplikasi MATLAB. Berdasarkan hasil dari eksperimen yang didapatkan, penelitian ini membuktikan bahwa algoritma genetika dalam penyelesaian CVRP mampu meningkatkan kecepatan algoritmanya dari representasi gen yang digunakan. Pada saat yang sama, algoritma genetika mampu menghasilkan solusi yang berkualitas tinggi dengan nilai <i>fitness</i> yang baik. Penelitian ini menyimpulkan bahwa algoritma genetika mampu menghasilkan solusi atas permasalahan CVRP yang kompleks dan memiliki

	skala besar dengan hasil yang berkualitas tinggi secara efisien [8].
Keterkaitan penelitian	Penelitian ini menggunakan algoritma genetika untuk menyelesaikan permasalahan VRP, utamanya CVRP. Selain itu, implementasi dari penelitian ini juga menggunakan aplikasi MATLAB.

Tabel 2.2 Penelitian Sebelumnya 2

Judul Paper	<i>Vehicle-Routing Optimization for Municipal Solid Waste Collection Using genetic Algorithm: The case of Southern Nablus City</i>
Penulis ; tahun	Ramiz Assaf, Yahya Saleh ; 2017
Deskripsi umum penelitian	Penelitian ini berfokus pada pencarian rute optimal untuk pengumpulan sampah padat di Kota Nablus, Palestina. Pengumpulan sampah ini dicakup oleh pemerintah setempat sehingga pencarian rute dilakukan di bawah naungan regulasi. Pencarian rute optimal dilakukan dengan penyelesaian <i>Vehicle Routing Problem</i> (VRP) serta algoritma genetika sebagai algoritmanya. Data yang digunakan dalam penelitian ini adalah data dari pemerintah setempat yang dikumpulkan pada tahun 2016. Luas wilayah yang menjadi objek penelitian ini adalah 605 km ² dengan total penduduk sebanyak 351.400 jiwa (2016). Armada pengangkut sampah yang digunakan berjumlah dua armada dan tempat pembuangan sementara sampah padat tersebar di 1230 titik. Dikarenakan armada yang digunakan memiliki batas atau limit, maka jenis VRP yang digunakan adalah CVRP atau <i>Capacitated-VRP</i> . Rute optimal yang didapatkan memiliki selisih yang cukup jauh dari rute sebelumnya, yakni 66% lebih menguntungkan, baik dari segi jarak yang ditempuh (Dari rata-rata 312 km menjadi 104,74 km) maupun waktu yang dibutuhkan (dari 7 jam per perjalanan menjadi 2,3 jam per perjalanan) [9].
Keterkaitan penelitian	Memiliki studi kasus yang sama yakni pengumpulan sampah dari beberapa tempat pembuangan sementara dengan algoritma genetika pula. Selain itu penelitian ini juga menggunakan CVRP yang

	memperhitungkan kapasitas armada pengangkut sampah.
--	---

2.2. Dasar Teori

Subbab ini menjelaskan dasar teori yang digunakan sebagai landasan pengerjaan tugas akhir.

2.2.1. DLH Kabupaten Bangkalan

Dinas Lingkungan Hidup (DLH) Kabupaten Bangkalan merupakan salah satu bidang kedinasan dalam pemerintahan di Kabupaten Bangkalan. Berdasarkan Peraturan Bupati Bangkalan No. 44 Tahun 2016 tentang Kedudukan, Susunan Organisasi, Tugas dan Fungsi, Serta Tata Kerja Dinas Lingkungan Hidup, Dinas Lingkungan Hidup Kabupaten Bangkalan berfokus pada keempat bidang:

- 1) Bidang Pelayanan Tata Lingkungan, yang meliputi inventarisasi, RPPLH, KLHS, kajian dampak lingkungan hidup, pemeliharaan lingkungan hidup, pertamanan, dan pemakaman.
- 2) Bidang Pengelolaan Sampah dan Limbah B3, yang meliputi pengelolaan sampah, pengelolaan limbah B3, dan pengembangan fasilitas teknis.
- 3) Bidang Pengendalian Pencemaran dan Kerusakan Lingkungan, yang meliputi pencegahan, penanggulangan, pemulihan, dan pengendalian kerusakan lingkungan hidup.
- 4) Bidang Penataan Lingkungan Hidup, yang meliputi pengawasan dan peningkatan kapasitas lingkungan hidup dan penanganan pengaduan dan penataan hukum.

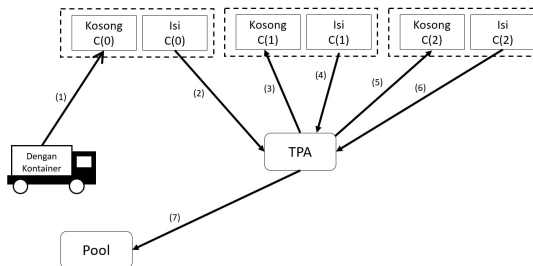
2.2.2. Pengelolaan Sampah Kabupaten Bangkalan

Berdasarkan jenis armada yang digunakan, yakni *Arm Roll Truck* dan *Dump Truck*, Dinas Lingkungan Hidup Kabupaten Bangkalan memiliki dua sistem pengangkutan sampah. Di antaranya adalah:

1) *Hauled Container System (HCS)*

Pengangkutan sampah dengan *Hauled Container System* atau HCS dilakukan oleh armada dengan tipe *Armroll Truck*, di mana armada tersebut membawa keseluruhan sampah beserta kontainer yang ditentukan, kemudian mengganti kontainer tersebut dengan kontainer lain yang kosong. Kontainer yang diganti memiliki jenis maupun kapasitas yang sama dengan yang dibawa. Proses HCS (Gambar 2.1) secara detail dapat dijelaskan:

- a. *Arm Roll Truck* datang ke lokasi TPS dengan membawa kontainer kosong.
- b. Sesampainya di lokasi, *Arm Roll Truck* menurunkan kontainer kosong yang dibawanya.
- c. Kontainer yang ingin diangkat diangkat dengan *Arm Roll Truck* (*loading process*).
- d. *Arm Roll Truck* membawa kontainer berisi sampah ke TPA.
- e. *Arm Roll Truck* melakukan *unloading* sampah di zona penimbunan di TPA.
- f. Setelah kontainer yang diangkat *Arm Roll Truck* kosong, armada tersebut menuju ke TPS selanjutnya untuk menukar kontainer tersebut dengan kontainer yang berisi sampah.
- g. Proses berakhir ketika *Arm Roll Truck* kembali menuju *pool*.

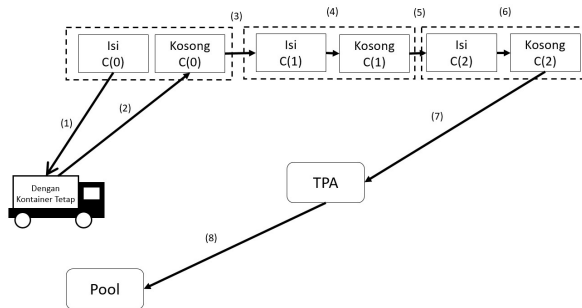


Gambar 2.1 Hauled Container System (HCS)

2) *Stationary Container System (SCS)*

Pengangkutan sampah dengan *Stationary Container System (SCS)* dilakukan oleh armada dengan tipe *Dump Truck*. Secara umum, sistem pengangkutan ini menggunakan TPS dengan kontainer yang tidak bisa dipindahkan. Proses SCS (Gambar 2.2) secara detail adalah sebagai berikut:

- a. *Dump Truck* datang ke lokasi TPS dengan kondisi kontainer kosong.
- b. Sesampainya di lokasi, *Dump Truck* melakukan loading sampah.
- c. Sesudah mengangkut/*loading* pada TPS pertama, *Dump Truck* berangkat ke TPS selanjutnya.
- d. Loading dilakukan hingga *Dump Truck* mencapai TPS terakhir.
- e. *Dump Truck* menuju ke TPA untuk melakukan *unloading*.
- f. *Dump Truck* kembali ke pool.



Gambar 2.2 Stationary Container System (SCS)

2.2.3. Optimasi

Optimasi adalah suatu aksi atau proses untuk mencapai suatu hasil yang optimal. Dalam ilmu matematika komputer, nilai optimal yang dimaksud dapat berupa nilai minimal atau nilai maksimal dari suatu fungsi riil, disesuaikan dengan permasalahan yang ada. Perolehan nilai optimal dilakukan

dengan cara pemilihan nilai yang berpotensi menjadi nilai optimal secara sistematis. Pemilihan atau pencarian nilai optimal tersebut adalah bentuk dari permasalahan optimasi.

Permasalahan Optimasi

Permasalahan optimasi sering kali muncul pada aktivitas sehari-hari, dengan berbagai bentuk nilai optimal yang berbeda-beda. Nilai optimal dapat berupa jumlah, waktu, kapasitas, jarak, biaya, dan lainnya. Berikut ini adalah contoh dari permasalahan optimasi:

- 1) Menentukan rute terpendek yang harus dilewati untuk mengunjungi lokasi-lokasi tertentu.
- 2) Meminimalisir penggunaan container pada pengiriman barang menggunakan transportasi laut di pelabuhan.
- 3) Meminimalisir jumlah karyawan dengan tetap mempertimbangkan kualitas produk yang dihasilkan.
- 4) Mengatur penjadwalan angkutan umum di suatu wilayah.

Selain permasalahan-permasalahan tersebut, optimasi juga dapat menjadi solusi pada permasalahan lain di berbagai bidang keilmuan.

Penyelesaian Permasalahan Optimasi

Penyelesaian permasalahan optimasi memiliki beberapa metode, di antaranya adalah Metode Eksak, Metode Heuristik, Metode Metaheuristik, dan Algoritma *Greedy*. Secara umum metode eksak digunakan untuk menemukan solusi optimal yang sudah pasti (eksak), sementara metode aproksimasi, metaheuristik, dan algoritma greedy digunakan untuk menemukan solusi paling optimal atau mendekati solusi optimal.

- 1) Metode Eksak
Metode eksak atau metode konvensional merupakan metode yang menggunakan perhitungan matematis biasa. Algoritma yang digunakan dalam metode eksak

di antaranya adalah *dynamic programming*, *constraint programming*, *branch and bound*, dan A^* .

2) Metode Aproksimasi

Metode aproksimasi dibagi menjadi dua sub-kelas, yakni algoritma aproksimasi dan metode heuristik. Algoritma aproksimasi adalah algoritma yang mampu menjamin optimasi solusi yang didapatkan. Metode heuristik dapat mencari solusi yang baik dalam skala permasalahan yang besar, namun tidak bisa menjamin bahwa solusi yang “baik” tersebut sebagai solusi yang optimal atau mendekati optimal. Metode heuristik sendiri dibagi menjadi dua bagian, yaitu heuristik khusus (*specific heuristic*) dan metaheuristik.

3) Metode Metaheuristik

Metode metaheuristik adalah algoritma yang mampu menyelesaikan hampir seluruh permasalahan optimasi. Berbeda dengan metode heuristik, metode metaheuristik memiliki metodologi yang lebih umum namun juga memiliki penyelesaian permasalahan yang kompleks, sehingga dapat dikatakan sebagai *upper-level of heuristic*. Metaheuristik mampu mengatasi permasalahan dalam skala yang besar dengan hasil yang memuaskan serta waktu yang cukup singkat. Metode ini memiliki dua orientasi yang berbeda, yakni *population-based* yang berfokus pada konsep eksplorasi, dan *single-solution-based* yang berfokus pada konsep eksploitasi.

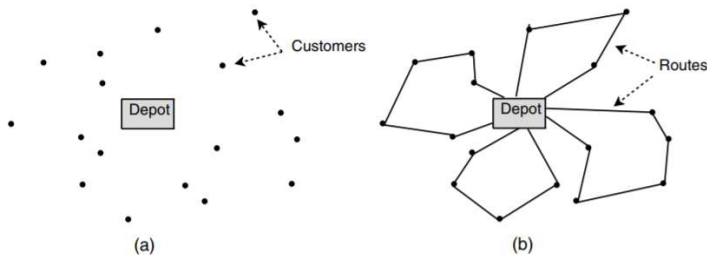
4) Algoritma Greedy

Algoritma greedy adalah algoritma yang bersifat konstruktif. Sesuai dengan sifatnya, algoritma ini mengkonstruksi suatu solusi dari awal dengan cara menentukan nilai pada suatu variabel keputusan hingga solusi tersebut selesai dibuat [10].

2.2.4. Vehicle Routing Problem (VRP)

Vehicle Routing Problem atau VRP adalah salah satu permasalahan optimasi kombinatorial yang dikemukakan oleh Dantzig dan Ramser pada tahun 1959. Permasalahan VRP

termasuk dalam *NP-Hard (Non-deterministic Polynomial-Hard)*, di mana tidak ada algoritma khusus yang dapat menghasilkan solusi optimal dalam waktu polinomial sehingga satu-satunya cara untuk mencari solusi yang paling mendekati optimal adalah dengan mencoba satu per satu seluruh kemungkinan yang ada. Secara teori, VRP merupakan perkembangan dari *Travelling Salesman Problem (TSP)*. Permasalahan VRP memiliki tujuan untuk mengoptimalkan rute armada kendaraan pengangkut agar dapat mengunjungi pelanggan-pelanggan yang tersebar di berbagai lokasi. Armada kendaraan pengangkut yang digunakan berjumlah dua atau lebih. Lokasi pelanggan atau lokasi yang harus dikunjungi harus dilewati tepat satu kali agar rute yang dihasilkan tidak terlalu panjang dan mendekati optimal [10] [11]. Ilustrasi VRP dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi VRP

Secara umum, VRP dapat digambarkan sebagai berikut ; $G = (V, A)$, G adalah suatu grafik dengan V sebagai titik yang harus dilalui dan A adalah representasi gabungan antara $\{i, j\}$. Nilai V adalah sebagai berikut ; $V = \{0, 1, 2, \dots, n\}$ di mana $V=0$ adalah titik mulainya. $A = \{(V_i, V_j) : V_i, V_j \in V, i < j\}$ adalah representasi dari jarak antara lokasi i dengan j yang dimulai dari i dan berakhir di j . Biaya yang dikeluarkan, c_{ij} , memiliki asosiasi dengan setiap jarak pada $\{i, j\} \in C$ [12]. Formula 1 adalah notasi umum yang digunakan dalam formulasi VRP.

$$C = \min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^K c_{ij} x_{ij}^k \quad (1)$$

C menunjukkan total biaya pengeluaran selama perjalanan yang harus diminimalisir. Untuk setiap n yang dikunjungi, terdapat node i dan j masing-masing satu yang merepresentasikan satu node dengan node lainnya. Dalam suatu perjalanan dari node i ke j , dibutuhkan satu kendaraan yang direpresentasikan dengan k . c_{ij} merupakan biaya yang dikeluarkan dalam suatu perjalanan dari node i ke j , sementara x_{ij} merepresentasikan adanya rute atau jalan dari node i ke node j oleh satu kendaraan, yang bernilai 1 jika terdapat rute dan 0 jika antara node i dan node j tidak berhubungan oleh suatu rute [8] [9].

Seiring berjalannya waktu serta semakin banyaknya implementasi penyelesaian VRP, bentuk dari permasalahan VRP memiliki banyak variasi, antara lain *Vehicle Routing Problem with Time Windows* (VRPTW), *Site-Dependent Vehicle Routing Problem* (SDVRP), *Multi-Depot Vehicle Routing Problem* (MDVRP), *Vehicle Routing Problem with Backhauls* (VRPB), *Vehicle Routing Problem with Pickup and Delivery* (VRPPD), *Capacitated Vehicle Routing Problem* (CVRP), dan lain-lain [11].

Karakteristik Vehicle Routing Problem (VRP)

Berikut ini adalah karakteristik-karakteristik yang dimiliki oleh *Vehicle Routing Problem* (VRP) [8].

- 1) Memiliki objektif atau tujuan umum untuk mengurangi jumlah jarak dalam satu perjalanan.
- 2) Kendaraan-kendaraan yang digunakan berasal dari satu depot yang sama, kecuali untuk *Multi-Depot Vehicle Routing Problem* (MDVRP). Dalam penelitian ini, depot yang digunakan adalah Kantor Dinas Lingkungan Hidup Kabupaten Bangkalan sebagai titik mula perjalanan armada truk pengangkut sampah.
- 3) Setiap kendaraan mengunjungi sekumpulan titik (direpresentasikan sebagai pelanggan) dan kembali

pada depot. Dalam penelitian ini, titik yang dimaksud adalah TPS yang harus dikunjungi.

- 4) Kebutuhan pelanggan yang dikunjungi tidak lebih dari kapasitas kendaraan yang digunakan. Dalam relevansinya dengan penelitian ini, maka volume sampah pada TPS-TPS yang harus dikunjungi tidak boleh melebihi kapasitas armada truk pengangkut sampah.
- 5) Seluruh titik harus dikunjungi oleh satu kendaraan sebanyak tepat satu kali.

2.2.5. Algoritma Genetika

Algoritma genetika ditemukan pertama kali oleh John Holland pada tahun 1960-an. Menurut Holland, algoritma genetika merupakan bentuk abstrak dari evolusi biologis yang dikembangkan menjadi suatu kerangka (*framework*) dalam ilmu komputasi. Algoritma genetika menggunakan konsep seleksi, rekombinasi, dan mutasi atas individu-individu yang disediakan dalam suatu populasi. Secara garis besar, algoritma genetika akan mengambil individu terbaik sebagai solusi atas suatu permasalahan. Algoritma genetika akan menilai baik-tidaknya suatu individu dengan melakukan perbandingan antarindividu berdasarkan nilai *fitness* yang dimiliki masing-masing individu [8] [12].

Komponen Algoritma Genetika

Berikut ini adalah komponen-komponen yang ada dalam algoritma genetika.

- 1) Gen
Gen merupakan komponen yang menggambarkan suatu unit nilai. Nilai yang dimaksud disebut juga sebagai alel. Sekumpulan gen yang berisi nilai-nilai tertentu disebut individu atau kromosom. Dalam algoritma genetika, alel dalam gen dapat berupa biner, integer, karakter, maupun kombinatorial. Jika direpresentasikan

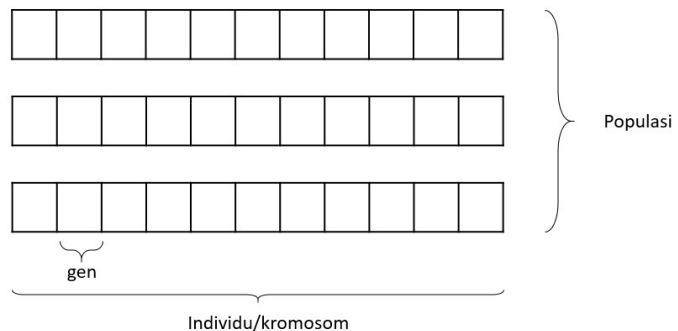
dengan konsep VRP, maka gen adalah gambaran dari lokasi yang harus dilalui dalam satu kali perjalanan [8].

2) Individu/Kromosom

Individu atau kromosom merupakan kumpulan dari beberapa gen. Dalam algoritma genetika, suatu individu mencakup suatu solusi yang berpotensi sebagai solusi atas permasalahan yang dihadapi. Jika dihubungkan dengan VRP, maka individu atau kromosom merupakan rute-rute yang harus dilalui dalam satu kali perjalanan [8].

3) Populasi

Populasi merupakan kumpulan dari individu atau kromosom [8]. Ilustrasi dari populasi, gen, dan individu atau kromosom dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi Populasi, Gen, dan Individu atau Kromosom

4) Nilai *Fitness*

Nilai *fitness* merupakan nilai yang digunakan untuk menentukan individu mana yang memiliki gen-gen dengan alel-alel terbaik sebagai solusi yang diharapkan. Secara umum, nilai *fitness* adalah gambaran dari seberapa bagus suatu individu atau kromosom. Dalam VRP, nilai *fitness* merupakan jumlah jarak yang dihasilkan oleh alel-alel dalam kumpulan gen pada suatu individu. Karena VRP memiliki konsep untuk menghasilkan jarak sependek mungkin, maka semakin kecil nilai *fitness* suatu individu, semakin baik kualitas

individu tersebut dan semakin besar pula kesempatan individu tersebut untuk menjadi solusi akhir [8] [9].

Proses Algoritma Genetika

Berikut ini adalah proses dari algoritma genetika, yang terdiri dari inialisasi populasi, inialisasi nilai *fitness*, seleksi, rekombinasi/*crossover*, dan mutasi.

1) Inialisasi Populasi

Langkah pertama dalam penerapan algoritma genetika adalah menginisialisasi populasi yang berisi kumpulan individu atau kromosom. Menginisialisasi populasi merupakan langkah yang memiliki pengaruh besar terhadap keseluruhan proses pada algoritma genetika. Semakin besar populasi atau semakin banyak kromosom yang ada dalam populasi, semakin banyak pula waktu dan sumber daya yang dibutuhkan dalam melakukan penyelesaian masalah dengan algoritma genetika. Sebaliknya, semakin sedikit individu dalam suatu kromosom, semakin sedikit pula waktu dan sumber daya yang diperlukan. Namun, solusi yang didapatkan dari populasi dengan ukuran besar memiliki hasil yang cukup signifikan dengan solusi awal dibandingkan dengan populasi berukuran kecil [8].

2) Inialisasi Nilai *Fitness*

Inialisasi nilai *fitness* dilakukan untuk menentukan individu mana yang memiliki gen-gen terbaik. Inialisasi nilai *fitness* digunakan untuk menentukan individu mana yang akan dipilih pada langkah seleksi [8] [12].

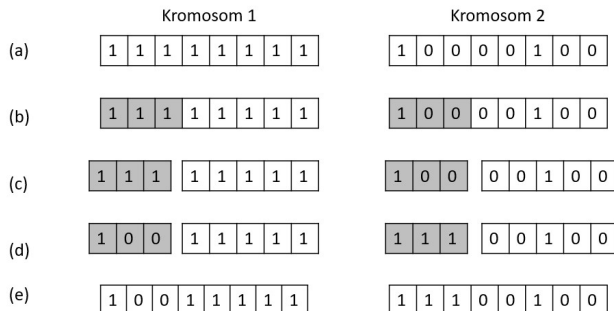
3) Seleksi

Langkah ketiga dari algoritma genetika adalah proses seleksi individu-individu dalam populasi. Pemilihan individu dilakukan berdasarkan nilai *fitness*-nya—semakin bagus nilai *fitness* yang dimiliki suatu

individu, semakin besar kemungkinan individu tersebut untuk terpilih. Dalam algoritma genetika, akan dipilih dua individu dengan nilai *fitness* terbaik [8].

4) Rekombinasi (*Crossover*)

Setelah melalui proses seleksi, dua individu terpilih akan dilakukan proses rekombinasi gen. Pada langkah keempat ini, algoritma genetika akan memilih secara acak tempat gen mana yang akan ditukar pada kedua individu terpilih. Setelah memilih tempat gen, proses rekombinasi antara kedua individu terpilih akan dilakukan dan menghasilkan individu dengan identitas gen yang baru. Sebagai contoh, terdapat dua individu berisi gen dalam bentuk string, yakni 10000100 dan 11111111. Tempat gen yang dipilih adalah 3 tempat gen paling awal. Jika dilakukan proses rekombinasi, maka kedua individu tersebut akan memiliki gen yang berbeda, yakni 11100100 dan 10011111, sesuai dengan ilustrasi pada Gambar 2.5. Secara umum, proses rekombinasi ini memiliki konsep yang sama dengan organisme haploid yang hanya berisi sepasang individu kromosom [8] [12].



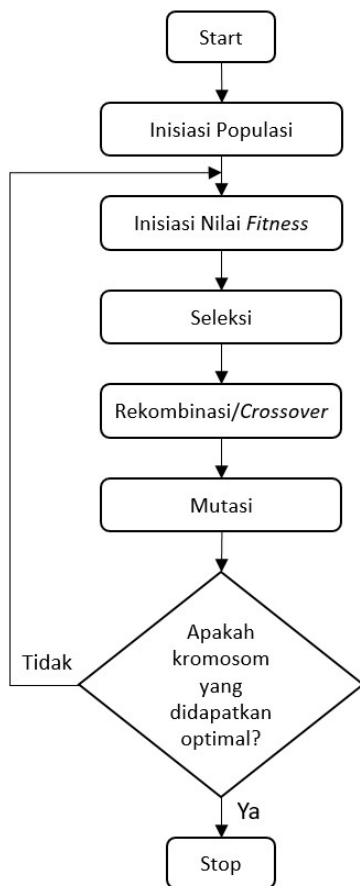
Gambar 2.5 Ilustrasi Rekombinasi (Crossover)

5) Mutasi

Langkah selanjutnya adalah proses mutasi atas dua individu yang sebelumnya telah direkombinasi. Algoritma genetika akan secara acak memilih tempat gen mana yang akan dilakukan mutasi, lalu memutasi tempat gen tersebut. Mutasi dilakukan dengan cara inversi gen yang dipilih. Sebagai contoh, suatu individu berbentuk string 00000100 akan dilakukan mutasi pada tempat kedua, sehingga menghasilkan kromosom 01000100 [12].

Dalam permasalahan VRP, proses mutasi ini akan melalui tahap pengecekan atau testing, di mana individu yang telah dimutasi akan dicek validasinya. Pengecekan validasi individu akan dilakukan dengan melihat rute antara dua gen yang berurutan. Jika antara kedua gen tersebut dapat dilalui suatu rute, maka individu tersebut dinyatakan valid. Jika tidak ada rute antara kedua gen tersebut, maka individu tersebut tidak valid dan tidak bisa dikalkulasi sebagai solusi potensial [8].

Jika setelah proses mutasi dilakukan belum ada individu yang memiliki nilai *fitness* yang optimal, maka individu-individu hasil mutasi tersebut ditempatkan pada populasi baru. Populasi baru tersebut akan melewati proses seleksi, rekombinasi, dan mutasi pula. Sekumpulan proses tersebut dihitung sebagai satu iterasi yang menghasilkan satu generasi. Proses iterasi seperti pada Gambar 2.6, akan berhenti pada generasi tertentu ketika suatu individu sudah memiliki nilai *fitness* yang optimal [8].



Gambar 2.6 Proses Algoritma Genetika

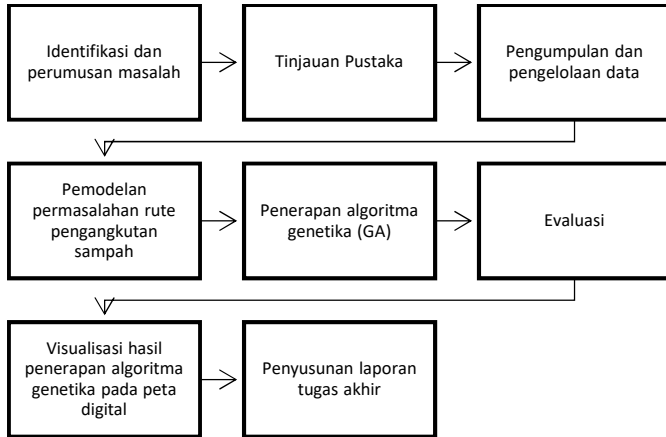
Halaman ini sengaja dikosongkan

BAB III METODOLOGI

Pada bab ini dijelaskan mengenai langkah-langkah penelitian dalam pengerjaan tugas akhir ini agar dapat diselesaikan secara sistematis dan terarah, serta disajikan pula jadwal kegiatan pengerjaan tugas akhir.

3.1. Diagram Tahapan Pelaksanaan Tugas Akhir

Dalam Gambar 3.1 dijabarkan tahapan penelitian tugas akhir dan uraian tahapan yang digunakan dalam bentuk diagram.



Gambar 3.1 Diagram Tahapan Pelaksanaan Tugas Akhir

3.2. Tahapan Pelaksanaan Tugas Akhir

Tahapan pelaksanaan tugas akhir meliputi kegiatan:

3.2.1. Identifikasi Masalah

Identifikasi dan perumusan masalah adalah tahap pertama pengerjaan tugas akhir ini. Tahap ini memiliki tujuan untuk mencari permasalahan yang ada pada proses pengangkutan sampah di Kabupaten Bangkalan, Madura, Jawa Timur. Perumusan masalah disusun berdasarkan kondisi eksisting

proses pengangkutan sampah di Bangkalan agar nantinya proses pencarian solusi akan sesuai dengan yang dibutuhkan.

3.2.2. Tinjauan Pustaka

Pencarian solusi atas permasalahan yang telah didapatkan harus memiliki metode, algoritma, serta cara kerja yang benar dan tepat. Karenanya, dibutuhkan tinjauan pustaka untuk menentukan metode yang digunakan untuk tugas akhir ini. Dalam proses tinjauan pustaka, dilakukan proses pencarian metode berdasarkan penelitian-penelitian sebelumnya dan/atau buku-buku yang memiliki permasalahan serupa dengan tugas akhir ini. Hal-hal yang ditinjau adalah penelitian-penelitian sebelumnya yang terkait, Dinas Lingkungan Hidup Kabupaten Bangkalan, sistem pengangkutan dan pengumpulan sampah Kabupaten Bangkalan, optimasi, *Vehicle Routing Problem* (VRP), dan algoritma genetika.

3.2.3. Pengumpulan dan Pengelolaan Data

Proses ini bertujuan untuk mencari data dan informasi mengenai proses pengambilan sampah di Kabupaten Bangkalan, komponen-komponen yang mendukung, kebijakan yang diterapkan, dan data lainnya yang berhubungan dengan tugas akhir ini. Pengambilan data dilakukan di Dinas Lingkungan Hidup Kabupaten Bangkalan, Madura, Jawa Timur.

Berikut ini adalah jenis data yang diperlukan untuk penelitian ini.

1) Data Sekunder

Data sekunder merupakan data yang diperoleh dari buku, artikel, jurnal, dan lain-lain yang memaparkan data secara umum. Data sekunder digunakan sebagai pendukung data primer dan/atau penyusunan data primer.

2) Data Primer

Data primer merupakan data yang diperoleh dari sumber asli, tanpa perantara maupun media perantara. Data primer diperoleh dengan proses wawancara, pengambilan data fisik secara langsung, dan/atau lainnya sesuai dengan kesepakatan.

3.2.4. Pemodelan Permasalahan

Setelah mendapatkan data yang dibutuhkan, dilakukan pemodelan permasalahan rute pengangkutan sampah. Dalam proses ini data yang diperoleh akan diteliti, disusun, kemudian dimodelkan berdasarkan perumusan masalah yang disusun pada bab sebelumnya agar nantinya pemrosesan data menjadi lebih mudah dan menghasilkan luaran yang diinginkan. Berikut ini adalah komponen-komponen yang akan dimodelkan pada tahap ini:

- 1) Koordinat lokasi seluruh TPS di Kabupaten Bangkalan yang dikelola oleh Dinas Lingkungan Hidup Kabupaten Bangkalan.
- 2) Rute pengangkutan sampah yang digunakan.
- 3) Jarak antar lokasi TPS yang dikelola oleh Dinas Lingkungan Hidup Kabupaten Bangkalan.

3.2.5. Penerapan Algoritma Genetika

Untuk mendapatkan luaran yang diinginkan, maka dilakukan penerapan algoritma genetika pada permasalahan terkait. Penerapan algoritma genetika dilakukan dengan menggunakan alat (*tool*) berupa MATLAB, aplikasi yang menunjang komputasi permasalahan teknik dengan kinerja tinggi (*high-performance*).

Masukan atau input dari algoritma genetika ini adalah gen-gen dan individu-individu yang belum dilakukan proses seleksi, rekombinasi/*crossover*, dan mutasi, sementara keluaran atau output dari algoritma genetika adalah individu berisi gen-gen

dengan nilai *fitness* terbaik yang merupakan solusi paling optimal dalam suatu permasalahan. Jika dihubungkan dengan VRP, maka gen merupakan representasi dari lokasi titik yang harus dikunjungi, individu merupakan rute-rute yang merepresentasikan solusi, dan populasi merupakan sekumpulan solusi-solusi atas permasalahan VRP. Luaran algoritma genetika dalam VRP adalah individu dengan nilai *fitness* terbaik yang menggambarkan rute pengangkutan sampah terpendek. Individu tersebut harus berkolerasi dengan fungsi tujuannya.

3.2.6. Evaluasi

Proses evaluasi merupakan penilaian terhadap luaran yang ditemukan berdasarkan penerapan algoritma genetika yang dilakukan dengan MATLAB. Luaran ini akan dibandingkan dengan kondisi eksisting proses pengumpulan dan pengangkutan sampah di Kabupaten Bangkalan. Dari proses ini, dapat dinilai apakah metode yang digunakan pada tugas akhir ini sesuai atau tidak dengan permasalahan yang ada. Penilaian ini didasarkan oleh perbandingan antara luaran dari performa algoritma genetika dengan kondisi eksisting.

3.2.7. Visualisasi Hasil Penerapan Algoritma Genetika pada Peta Digital

Setelah melakukan evaluasi terhadap luaran performa algoritma genetika, visualisasi dilakukan untuk menggambarkan luaran tersebut. Hal ini bertujuan agar luaran performa algoritma genetika mudah dipahami. Visualisasi terhadap luaran tersebut dilakukan pada peta digital berbasis web.

3.2.8. Penyusunan Laporan Tugas Akhir

Penyusunan laporan tugas akhir merupakan dokumentasi hasil penelitian dan luaran dalam bentuk buku tugas akhir, paper, dan/atau jurnal ilmiah. Dokumentasi dilakukan sesuai dengan format yang telah ditentukan oleh Departemen Sistem Informasi ITS.

BAB IV PERANCANGAN

Dalam bab ini akan dijelaskan mengenai rancangan dari tugas akhir yang berkaitan dengan subyek dan obyek penelitian, pemilihan subyek dan obyek penelitian, dan bagaimana penelitian akan dilakukan.

4.1 Pengumpulan Data

Pada subbab berikut akan dilakukan pengumpulan data yang dibutuhkan dalam pengerjaan tugas akhir. Proses perolehan data didapatkan dari permintaan data secara langsung kepada Dinas Lingkungan Hidup Kabupaten Bangkalan. Beberapa data yang diperoleh adalah sebagai berikut:

4.1.1. Lokasi Tempat Pembuangan Sampah (TPS) dengan Sistem *Stationary Container System* (SCS)

Lokasi tempat pembuangan sampah dengan sistem *Stationary Container System* (SCS) dapat dilihat pada Tabel 4.1. Informasi pada Tabel 4.1 mencakup beberapa nama TPS/Depo serta titik latitud dan longitud TPS/Depo terkait. Keseluruhan data dapat dilihat pada Lampiran A.

Tabel 4.1 Lokasi Tempat Pembuangan Sampah (TPS) dengan Sistem *Stationary Container System* (SCS)

No.	Depo/TPS	Latitud	Longitud
1.	Depo Perum Griya Abadi	-7.0522527	112.729527
2.	Depo Senenan	-7.0392918	112.7406358
3.	Depo Jl. Mayjen Sungkono	-7.030617	112.751247
4.	Kantor BRI	-7.0424095	112.7386265
5.	Kantor Pajak	-7.0420144	112.738886
6.	Lampu Merah Senenan	-7.0388905	112.7409389
...

4.1.2. Rute Armada Truk Pengangkut Sampah dengan Sistem *Stationary Container System* (SCS)

Sebagian informasi mengenai rute armada pengangkut sampah beserta pengemudi dijabarkan pada Tabel 4.2. Keseluruhan data terkait dapat dilihat pada Lampiran B.

Tabel 4.2 Rute Armada Truk Pengangkut Sampah

No.	Waktu	Pengemudi	Rute
1.	Pagi	Djumaidi	TPS Masjid Junok
2.			TPS Perum Halim 1
3.			TPS Perum Halim 2
4.			TPS Perum Griya Utama
5.			TPS Radio Amana FM
6.			TPS Dishutbun
7.			TPS Perum Cendana
8.		Imam Subakti	Depo Perum Griya Abadi
9.			TPS IKIP
10.			TPS Coca Cola
11.			TPS Dr. Surata
12.			TPS Mushola Sak-Sak Timur
...

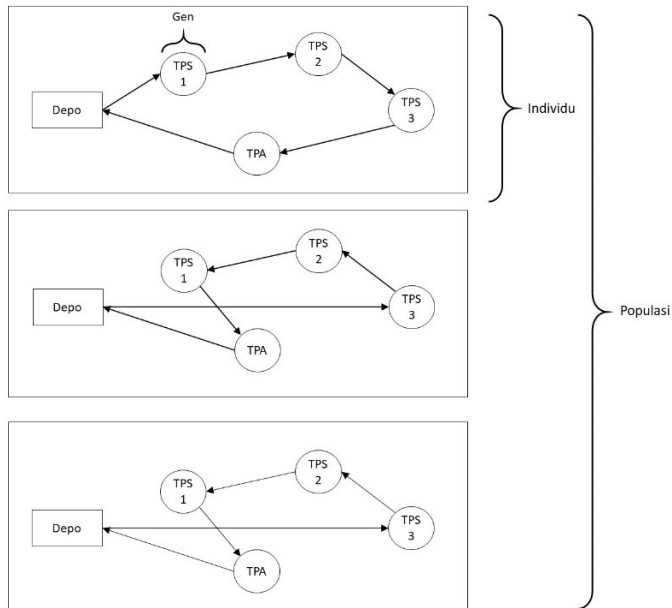
4.2 Inisialisasi Algoritma Genetika terhadap Program Pencarian Rute Pengangkutan Sampah

Untuk melakukan pencarian rute pengangkutan sampah dengan menggunakan algoritma genetika dalam ilmu komputasi, diperlukan adanya inisiasi komponen algoritma genetika yang berhubungan dengan pencarian rute. Komponen-komponen algoritma genetika yang berhubungan dengan pencarian rute adalah sebagai berikut:

1. Gen

Gen pada permasalahan pencarian rute merupakan titik lokasi yang harus dikunjungi. Dalam permasalahan pencarian rute pengangkutan sampah, gen merupakan titik TPS yang harus dikunjungi oleh armada truk pengangkut sampah serta TPA sebagai gen yang harus

dikunjungi terakhir. Ilustrasi mengenai gen pada pencarian rute pengangkutan sampah dapat dilihat pada Gambar 4.1.



Gambar 4.1 Ilustrasi Komponen Algoritme Genetika terkait Pengangkutan Sampah

2. Individu/Kromosom

Dalam algoritma genetika, individu atau kromosom merupakan sekumpulan gen yang membentuk suatu solusi. Dalam pencarian rute pengangkutan sampah, individu merupakan sekumpulan gen yang membentuk suatu rute. Ilustrasi mengenai individu terkait dengan pengangkutan sampah dapat dilihat pada Gambar 4.1.

3. Populasi

Dalam pencarian rute pengangkutan sampah, populasi direpresentasikan sebagai sekumpulan rute dalam program. Ilustrasi mengenai populasi terkait

pengangkutan sampah dapat dilihat pada Gambar 4.1, di mana populasi yang tersedia berjumlah tiga populasi.

4. Nilai *Fitness*

Nilai *fitness* merupakan nilai yang menyatakan bagus-tidaknya suatu solusi. Dalam pencarian rute pengangkutan sampah, nilai *fitness* yang dimaksud adalah jarak tempuh suatu populasi atau rute.

Selain menentukan komponen algoritma genetika, diperlukan pula inisiasi parameter yang menentukan kinerja algoritma dalam proses pencarian rute. Parameter-parameter yang digunakan adalah sebagai berikut:

1. Alpha

Alpha (α) merupakan parameter yang mempengaruhi kinerja algoritma genetika dalam proses rekombinasi. Parameter ini akan menentukan tempat gen mana yang akan dilakukan rekombinasi antara dua populasi.

2. Beta

Beta (β) merupakan parameter yang dapat menentukan gen mana yang akan dilakukan mutasi. Parameter tersebut mempengaruhi kinerja algoritma genetika dalam proses mutasi.

3. Gamma (γ)

Untuk melakukan mutasi, diperlukan dua gen dalam satu populasi. Karenanya, dibutuhkan parameter Gamma (γ) sebagai parameter gen kedua dalam proses mutasi, selain gen dengan parameter beta.

4.3 Perancangan *Use Case Diagram* Aplikasi

Pada subbab berikut akan dilakukan perancangan *use case* diagram aplikasi yang meliputi penentuan komponen, perancangan, dan penjelasan mengenai *use case diagram* yang akan menjabarkan cara kerja penyusunan aplikasi dalam tugas akhir.

4.3.1. Penentuan *Actor*, *Use Case*, dan *Relation*

Berikut ini adalah *actor*, *use case*, dan relasi antarkomponen dalam pengembangan aplikasi.

Actor

Actor atau aktor merupakan pengguna (*user*) atau sistem yang berperan sebagai subjek. Dalam aplikasi ini, akan dilibatkan satu aktor, yakni *User* atau pengguna. *User* dapat berinteraksi secara langsung dengan sistem tanpa perantara aktor lainnya.

Use Case

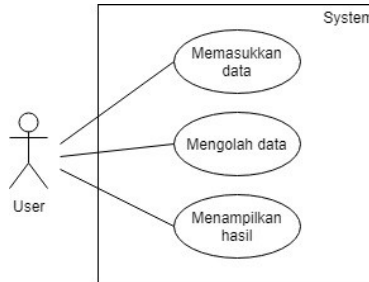
Use case merupakan aksi atau kegiatan yang dilakukan dalam suatu sistem. Secara umum, *use case* mendefinisikan interaksi-interaksi yang terjadi antara sistem dengan aktor. Dalam aplikasi ini, terdapat tiga *use case*, yakni; Memasukkan Data, Mengolah Data, dan Menampilkan Hasil.

Relation

Relation atau relasi merupakan hubungan yang terjadi dalam sistem. Relasi yang dimaksud melibatkan seluruh komponen diagram, seperti aktor dan *use case*. Dalam aplikasi ini, terdapat tiga relasi yakni; relasi antara *User* dengan *use case* Memasukkan Data, relasi antara *User* dengan *use case* Mengolah Data, dan relasi antara *User* dengan *use case* Menampilkan Hasil.

4.3.2. Perancangan *Use Case Diagram*

Perancangan Diagram *Use Case* dilakukan setelah menentukan komponen-komponen diagram berupa *Actor*, *Relation*, dan *Use Case*. Berdasarkan subbab sebelumnya, telah ditentukan satu *actor*, tiga *use case*, dan tiga relasi. Hasil rancangan Diagram *Use Case* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Perancangan Use Case Diagram

Setiap *use case* memiliki tabel masing-masing yang menjelaskan kode serta nama *use case*, aktor yang terlibat, deskripsi singkat, kondisi awal, *flow of event*, *alternate flow*, informasi terkait, dan kondisi akhir. Penjelasan mengenai masing-masing *use case* dapat dilihat pada Tabel 4.3, Tabel 4.4, dan Tabel 4.5.

Tabel 4.3 Use Case UC1

Kode	UC1
Use Case	Memasukkan Data
Aktor	<i>User</i>
Deskripsi	Proses memasukkan informasi-informasi yang diperlukan untuk pengolahan data.
Kondisi Awal	Tidak ada data/informasi dalam sistem.
Flow of Event	<ol style="list-style-type: none"> 1. <i>User</i> memilih fitur ‘Tambah Data’. 2. <i>User</i> memasukkan nomor ID, keterangan lokasi, serta titik lintang dan longitud pada lokasi terkait. 3. <i>User</i> menggunakan fitur ‘Simpan Data’ untuk menyimpan informasi-informasi mengenai titik-titik lokasi dalam bentuk CSV (jika dibutuhkan). 4. Sistem menampilkan titik-titik lokasi yang telah dimasukkan.
Alternate flow	<ol style="list-style-type: none"> 1. <i>User</i> memilih fitur ‘Upload Data’.

	<ol style="list-style-type: none"> 2. <i>User</i> memilih file .csv berisi nomor ID, keterangan lokasi, serta titik lintang dan bujur lokasi dengan format yang telah ditentukan pada fitur ‘Simpan Data’. 3. Sistem menampilkan titik-titik lokasi yang telah dimasukkan.
Exceptional flow	<ul style="list-style-type: none"> ▪ Data tidak dapat disimpan karena kurangnya informasi yang diberikan. ▪ File yang diunggah tidak sesuai dengan format yang ditentukan sehingga data tidak bisa disimpan.
Informasi terkait	Nomor ID lokasi, keterangan lokasi, titik lintang lokasi, titik bujur lokasi.
Kondisi Akhir	Sistem berhasil mendapatkan dan menampilkan informasi terkait yang telah dimasukkan sebelumnya.

Tabel 4.4 Use Case UC2

Kode	UC2
Use Case	Mengolah Data
Aktor	<i>User</i>
Deskripsi	Proses pengelolaan data dan informasi yang didapatkan dengan algoritma genetika.
Kondisi Awal	Sistem memiliki informasi-informasi yang dibutuhkan dalam pengelolaan data.
Flow of Event	<ol style="list-style-type: none"> 1. <i>User</i> memilih fitur ‘Simpan Titik’ dan ‘Simpan Jarak’ agar sistem dapat mengelola informasi mengenai titik-titik lokasi serta jarak antartitik. 2. <i>User</i> memilih fitur ‘Jalankan TSP’. 3. Sistem mengelola data yang dimiliki berupa jumlah titik dan jarak antartitik.5 4. Sistem membuat file .csv berisi hasil pengelolaan data dengan algoritma genetika. <i>File</i> yang dimaksud berisi urutan titik yang harus dilalui supaya mencapai rute yang optimal.

Alternate flow	-
Exceptional flow	<ul style="list-style-type: none"> ▪ Kurangnya data/informasi yang diberikan memungkinkan gagalnya perolehan hasil pengelolaan data.
Informasi terkait	Jumlah titik lokasi, jarak antartitik lokasi.
Kondisi Akhir	Hasil pengelolaan data dieksport dalam bentuk file .csv.

Tabel 4.5 Use Case UC3

Kode	UC3
Use Case	Menampilkan Hasil
Aktor	<i>User</i>
Deskripsi	Proses menampilkan hasil pengelolaan data dalam bentuk peta digital.
Kondisi Awal	Hasil pengelolaan data telah didapatkan sistem.
Flow of Event	<ol style="list-style-type: none"> 1. <i>User</i> memilih fitur ‘Tampilkan Hasil’. 2. Sistem melakukan penggabungan data antara hasil pengelolaan data dengan data yang telah dimasukkan pada <i>Use Case 1</i>. 3. Sistem menghasilkan urutan rute yang optimal di dalam <i>database</i>. 4. Sistem menampilkan urutan rute yang optimal dalam bentuk peta digital.
Alternate flow	-
Exceptional flow	Peta digital tidak akan muncul jika data yang dibutuhkan tidak lengkap.
Informasi terkait	Hasil pengelolaan data, nomor ID lokasi, keterangan lokasi, titik latitud lokasi, titik longitud lokasi.
Kondisi Akhir	Hasil pengelolaan data berupa rute optimal berhasil divisualisasi dengan peta digital.

BAB V IMPLEMENTASI

Bab ini menjelaskan proses pelaksanaan penelitian tugas akhir dan proses implementasi optimasi rute dengan algoritma genetika serta penyusunan visualisasi hasil implementasi.

5.1. Implementasi Algoritma Genetika pada Program

Bagian ini akan dijelaskan mengenai pengembangan program menggunakan algoritma genetika dalam beberapa langkah sebagai berikut:

5.1.1. Input Data

Dalam aplikasi ini, data yang diperlukan adalah koordinat titik yang harus dikunjungi dan jarak antartitik. Kedua data tersebut ditampung dalam file .xls dan .csv. Proses input kedua data tersebut ke dalam MATLAB dapat dilihat dalam Kode Program 5.1.

```
titik = xlsread('D:\KULIAH\TA\MATLAB\data\titik1.xlsx');  
jarak = csvread('D:\KULIAH\TA\MATLAB\data\jarak1.csv');
```

Kode Program 5.1 Input Data

5.1.2. Menentukan Jumlah Populasi dan Generasi

Jumlah populasi dan jumlah generasi masing-masing direpresentasikan oleh variabel 'populasi' dan 'generasi'. Selain itu, diperlukan juga penentuan jumlah titik yang harus dilalui dengan variabel 'n_titik'. Adapun penentuan jumlah populasi dan generasi dapat dilihat pada Kode Program 5.2.

```
n_titik = length(titik);  
populasi=1000;  
generasi=100;
```

Kode Program 5.2 Menentukan Jumlah Populasi dan Generasi

5.1.3. Inisialisasi Populasi

Inisialisasi populasi merupakan pembuatan populasi dari titik-titik yang sudah dimasukkan sebelumnya. Untuk memilih secara acak anggota dari masing-masing populasi, digunakan fungsi *randperm*. Pengacakan yang dilakukan tidak boleh melibatkan titik pertama dan titik terakhir, sehingga fungsi *randperm* hanya berlaku pada posisi titik kedua hingga posisi titik kedua terakhir. Proses inisialisasi populasi dapat dilihat pada Kode Program 5.3.

```
G = zeros(populasi, n_titik);
for perpopulasi = 1:populasi
    G(perpopulasi,:) = [1 randperm(n_titik-2)+1 n_titik];
end
```

Kode Program 5.3 Inisialisasi Populasi

5.1.4. Seleksi

Untuk menjalankan seleksi, diperlukan adanya jarak antartitik yang direpresentasikan oleh variabel 'dstances'. Perhitungan jarak pada variabel 'dstances' meliputi jarak antara satu titik dengan titik selanjutnya serta jarak antara titik terakhir dengan titik pertama. Pada Kode Program 5.4, proses seleksi akan memilih individu dengan jarak 'dstances' terbesar terlebih dahulu.

```
for perpopulasi = 1:populasi
    Gx = G(perpopulasi,:);
    distance = 0;
    for i = 1:n_titik - 1
        distance = distance + jarak(Gx(i), Gx(i+1));
    end
    distance = distance + jarak(Gx(n_titik), Gx(1));
    dstances(perpopulasi) = distance;
end
```

Kode Program 5.4 Seleksi

Pada Kode Program 5.5, dilakukan inversi dengan variabel 'i_dstances' untuk menentukan jarak terpendek.


```
i_dstances = 1./dstances;
prob = i_dstances/sum(i_dstances);
```

Kode Program 5.5 Inversi variabel 'i_dstances'

5.1.5. Rekombinasi/*Crossover*

Proses rekombinasi atau *crossover* dilakukan dengan merekombinasikan suatu individu atau generasi (direpresentasikan dengan variabel 'gen1') dengan individu lainnya (direpresentasikan dengan variabel 'gen2'). Proses rekombinasi ini tidak mempengaruhi posisi titik pertama dan titik terakhir. Jumlah gen yang akan dilakukan rekombinasi bergantung pada parameter 'alpha'. Hasil rekombinasi adalah individu baru berupa anak dari kedua individu terpilih yang direpresentasikan sebagai 'gench1' dan 'gench2'. Proses rekombinasi dapat dilihat pada Kode Program 5.6.

```
gens = G(i,:);
children = zeros(populasi, n_titik);
for jumlahpair = 1:(populasi/2)
    a1 = 1+2*(jumlahpair-1);
    a2 = 2+2*(jumlahpair-1);
    gen1 = gens(a1,:);
    gen2 = gens(a2,:) ;
    for no_titik = 2:(n_titik-1)
        cp = ceil((no_titik)*alpha);
    end

    gench1 = crossover_anak(gen1,gen2,cp);
    gench2 = crossover_anak(gen2,gen1,cp);
    children(a1,:) = gench1;
    children(a2,:) = gench2;
end
G = children; |
```

Kode Program 5.6 Rekombinasi/*Crossover*

5.1.6. Mutasi

Proses mutasi melibatkan suatu individu terpilih dengan jarak terpendek. Titik-titik yang merupakan gen-gen dari individu tersebut akan dirombak dengan fungsi *fliplr* tanpa melibatkan titik pertama atau titik terakhir. Sesuai Kode Program 5.7, titik

kedua hingga titik kedua terakhir direpresentasikan sebagai variabel 'no_titik'. Perombakan posisi masing-masing titik menggunakan parameter 'beta' dan 'gamma' serta *ceil* sebagai pembulat. Proses mutasi dapat dilihat pada Kode Program 5.7.

```
for perpopulasi = 1:populasi
    n1 = ceil(no_titik*beta);
    n2 = ceil(no_titik*gamma);
end
G(perpopulasi,gench1:gench2) = fliplr(G(perpopulasi,gench1:gench2));
```

Kode Program 5.7 Mutasi

5.1.7. Menampilkan Hasil keluaran

Hasil dari seluruh proses algoritma genetika akan direpresentasikan dengan variabel 'genbest', seperti yang dijabarkan pada Kode Program 5.8.

```
G(1,:) = genbest;
```

Kode Program 5.8 inialisasi Variabel 'genbest'

Contoh dari hasil optimasi algoritma genetika dijabarkan pada Kode program 5.9, di mana masing-masing titik diwakilkan oleh urutan titik awalnya.

```
genbest =
    1     6     5     2     3     4     7     8     9
```

Kode Program 5.9 Keluaran Variabel 'genbest'

5.2. Pengembangan Visualisasi

Pada bagian ini akan dijelaskan mengenai visualisasi dari hasil pencarian rute optimal oleh algoritma genetika yang telah dijabarkan pada subbab sebelumnya. Visualisasi dirancang dalam bentuk aplikasi web yang terdiri dari empat kegiatan utama; pembuatan basis data, pemasukkan data titik lokasi, pengelolaan algoritma, dan visualisasi hasil pengelolaan algoritma genetika dengan peta digital.

5.2.1. Basis Data

Dalam aplikasi ini, dibutuhkan basis data (*database*) untuk menyimpan informasi maupun data terkait lokasi, pengelolaan algoritma genetika, dan visualisasi hasil pengelolaan. Basis data yang dimiliki berjumlah satu basis data, yang berisi dua tabel dengan fungsionalitas yang berbeda. Pada Gambar 5.1, dapat dilihat bahwa basis data 'ga' memiliki tabel 'ga1' dan 'ga2'.



Gambar 5.1 Basis Data Aplikasi

Tabel 'ga1'

Tabel 'ga1' memiliki fungsi untuk menyimpan data terkait titik lokasi yang akan diolah dengan algoritma genetika. Sesuai dengan Tabel 5.1, berikut adalah penjabaran mengenai kolom yang dimiliki Tabel 'ga1'.

Tabel 5.1 Tabel 'ga1'

id_koordinat	ket_koordinat	lat	lng
1	Dinas Lingkungan Hidup Bangkalan	-7.0476101	112.7323577
2	POM BENSIN JUNOK	-7.0298799	112.7578553
3	TPS SMAN 1	-7.0297794	112.7587738
4	SMA 4	-7.0213075	112.764092
5	TPS RE. MARTADINATA	-7.046374	112.728703
6	PONPES AL KHOLILYAH	-7.028921	112.754748
7	TPA	-7.1031674	112.7308058

Tabel 'ga1' terdiri dari kolom:

a. id_koordinat

Kolom 'id_koordinat' berisi ID koordinat yang menjabarkan urutan titik lokasi yang dimasukkan ke dalam basis data. Urutan titik lokasi dalam kolom ini diisi berdasarkan rute yang sedang digunakan sebelum dilakukan optimalisasi. Kolom 'id_koordinat' bertipe integer.

b. ket_koordinat

Kolom 'ket_koordinat' berisi keterangan mengenai titik lokasi yang dimasukkan ke dalam basis data. Kolom ini memiliki tipe *text*.

c. lat

Kolom 'lat' berisi koordinat lintang dari titik lokasi yang dimasukkan. Kolom ini bertipe *varchar* untuk menghindari pembulatan dan penggunaan matematis terhadap koordinat lintang.

d. lng

Kolom 'lng' berisi koordinat bujur dari titik lokasi yang dimasukkan. Kolom ini bertipe *varchar* untuk menghindari pembulatan dan penggunaan matematis terhadap koordinat bujur.

Tabel 'ga2'

Tabel 'ga2' berfungsi untuk menyimpan informasi mengenai hasil pengelolaan algoritma genetika. Sesuai dengan Tabel 5.2, berikut adalah penjabaran mengenai kolom yang dimiliki Tabel 'ga2'.

Tabel 5.2 Tabel 'ga2'

id	awal	akhir	id_urutan
15	1	1	1561476771
16	2	5	1561476771
17	3	3	1561476771
18	4	2	1561476771
19	5	4	1561476771
20	6	6	1561476771
21	7	7	1561476771

Tabel 'ga2' terdiri dari kolom:

- a. id
Kolom 'id' menjabarkan ID dari hasil yang dimasukkan ke dalam basis data. Hasil yang dimaksud dihasilkan oleh langkah menampilkan hasil keluaran pada subbab 5.2.7. Tipe kolom 'id' adalah integer dengan fitur *auto increment*, di mana nomor ID akan bertambah dengan sendirinya seiring masuknya data ke dalam basis data.
- b. awal
Kolom 'awal' berisi ID koordinat yang menjabarkan urutan titik lokasi yang dimasukkan ke dalam basis data. Urutan titik lokasi dalam kolom ini diisi berdasarkan rute yang sedang digunakan sebelum dilakukan optimalisasi. Nilai dari Kolom 'awal' didapatkan dari Kolom 'id_urutan' pada Tabel 'g1'.
- c. akhir
Kolom 'akhir' merupakan urutan titik lokasi setelah dilakukan optimalisasi dengan algoritma genetika. Sama dengan Kolom 'awal', Kolom 'akhir' juga bertipe integer.
- d. id_urutan
Kolom 'id_urutan' berisi waktu (*timestamp*) input data hasil optimalisasi. Kolom ini memiliki tipe varchar untuk menghindari pembulatan serta perhitungan matematis terhadap ID urutan.

Basis data 'ga' memerlukan koneksi dengan PHP agar aplikasi dapat berjalan. Koneksi basis data dilakukan oleh fungsi 'getConnection' dengan PDO dalam file 'config.php' sesuai dengan kode Program 5.10.

```

<?php
class Config{
    private $host = "localhost";
    private $db_name = "ga";
    private $username = "root";
    private $password = "";
    public $conn;

    public function getConnection(){
        $this->conn = null;

        try{
            $this->conn = new PDO("mysql:host=" . $this->host . ";dbname="
                . $this->db_name, $this->username, $this->password);
        }
        catch(PDOException $exception){
            echo "Connection error: " . $exception->getMessage();
        }

        return $this->conn;
    }
}

```

Kode Program 5.10 Konfigurasi Basis Data

5.2.2. Input Data Titik Lokasi

Dalam kegiatan input data titik lokasi, terdapat empat sub-kegiatan yang terdiri dari penambahan, penyimpanan, *upload*, serta penghapusan data titik lokasi.

a. Penambahan Data Titik Lokasi

Penambahan data dilakukan oleh fungsi 'insert' sesuai dengan Kode Program 5.11. Data yang diperlukan adalah ID titik lokasi, keterangan lokasi, serta koordinat latitud dan longitud lokasi. Terdapat empat variabel yang digunakan dalam fungsi ini, yakni; variabel 'id' sebagai ID titik lokasi, variabel 'nm' sebagai keterangan titik lokasi, variabel 'lt' sebagai titik latitud, dan variabel 'ln' sebagai titik longitud.

```

function insert(){
    $query = "insert into ".$this->table_name." values(?,?,?,?)";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $this->id);
    $stmt->bindParam(2, $this->lt);
    $stmt->bindParam(3, $this->ln);
    $stmt->bindParam(4, $this->nm);

    if($stmt->execute()){
        return true;
    }else{
        return false;
    }
}

```

Kode Program 5.11 Penambahan Data Titik

b. Penyimpanan Data Titik Lokasi

Penyimpanan data titik lokasi dilakukan untuk menyimpan suatu rute agar dapat digunakan lagi nantinya. Data disimpan dalam bentuk *.csv* (*comma-delimited string*). Penyimpanan data dilakukan dengan bantuan fungsi ‘setActiveSheetIndex’ pada file PHPExcel.php yang dijelaskan pada Kode Program 5.12.

```

public function setActiveSheetIndex($pIndex = 0)
{
    $numSheets = count($this->_workSheetCollection);

    if ($pIndex > $numSheets - 1) {
        throw new PHPExcel_Exception(
            "You tried to set a sheet active by the out of bounds index: {$pIndex}. The actual number of sheets is {$numSheets}."
        );
    } else {
        $this->_activeSheetIndex = $pIndex;
    }
    return $this->getActiveSheet();
}

```

Kode Program 5.12 Penyimpanan Data Titik

Sesuai dengan Kode Program 5.13, ID titik lokasi, keterangan lokasi, serta koordinat lintang dan longitud lokasi disimpan dengan masing-masing lokasi berada dalam satu baris yang sama. Untuk kolom lintang dan longitud, diperlukan adanya

identifikasi tipe variabel agar nilai pada kolom latitud dan longitud tetap dalam tipe *varchar* atau *string*.

```

$csv = new PHPEXcel();
$csv->getProperties();

$numrow = 1;
while($data = $sql->fetch()) {
    $csv->setActiveSheetIndex(0)->setCellValue('A'.$numrow, $data['
        id_koordinat']);
    $csv->setActiveSheetIndex(0)->setCellValue('B'.$numrow, $data['
        ket_koordinat']);
    $csv->setActiveSheetIndex(0)->setCellValueExplicit('C'.$numrow, $data['
        lat'], PHPEXcel_Cell_DataType::TYPE_STRING);
    $csv->setActiveSheetIndex(0)->setCellValueExplicit('D'.$numrow, $data['
        lng'], PHPEXcel_Cell_DataType::TYPE_STRING);
    $numrow++;
}

$sql = $conn->prepare("SELECT id_koordinat, ket_koordinat, lat, lng FROM gal
ORDER BY id_koordinat");
$sql->execute();

```

Kode Program 5.13 Pemasukan Data yang ke dalam Basis Data

Tampilan dari data yang telah disimpan dalam bentuk .csv dapat dilihat pada Tabel 5.3.

Tabel 5.3 Tabel Data yang Disimpan

	A	B	C	D	E	F	G
1	1,"Dinas Lingkungan Hidup Bangkalan",	"-7.0476101",	"112.7323577"				
2	2,"POM BENSIN JUNOK",	"-7.0298799",	"112.7578553"				
3	3,"TPS SMAN 1",	"-7.0297794",	"112.7587738"				
4	4,"SMA 4",	"-7.0213075",	"112.764092"				
5	5,"TPS RE. MARTADINATA",	"-7.046374",	"112.728703"				
6	6,"PONPES AL KHOLILIAH",	"-7.028921",	"112.754748"				
7	7,"TPA",	"-7.1031674",	"112.7308058"				
8							
9							

c. Upload Data Titik Lokasi

Upload atau pengunggahan data titik lokasi dilakukan untuk memasukkan data mengenai lokasi-lokasi yang dibutuhkan dalam melakukan optimasi rute. Pengunggahan hanya dapat dilakukan pada file berbentuk .csv, dengan format yang telah ditentukan oleh fitur penyimpanan data titik lokasi pada subbab sebelumnya. Data maupun informasi yang diperlukan adalah ID titik lokasi, keterangan lokasi, serta koordinat latitud dan

longitud lokasi terkait yang akan dioptimalisasi dengan algoritma genetika. Kode Program 5.14 menampilkan cara kerja pengunggahan data titik lokasi, serta memasukannya ke dalam basis data. Variabel 'getData' merupakan *array* yang berisi nilai ID titik lokasi, keterangan lokasi, serta koordinat latitud dan longitud lokasi.

```

if(isset($_POST["Import"])){
    $filename=$_FILES["file"]["tmp_name"];
    if($_FILES["file"]["size"] > 0) {
        $file = fopen($filename, "r");
        while (($getData = fgetcsv($file, 10000, ",") !== FALSE) {
            $sql = "INSERT into ga1 (
                id_koordinat,ket_koordinat,lat,lng) values ('".$
                getData[0]."', '".$getData[1]."', '".$getData[2]."', '
                '.$getData[3]."'");
            $conn->prepare($sql)->execute();
        }
        fclose($file);
    }
}

```

Kode Program 5.14 Upload Data Titik

d. Penghapusan Data Titik Lokasi

Opsi penghapusan data dapat dilakukan pada masing-masing baris yang terdiri dari ID titik lokasi, keterangan lokasi, serta koordinat latitud dan longitud lokasi. Sesuai dengan Kode Program 5.15, data yang dipilih dihapus dari Tabel 'ga1'. Proses penghapusan data titik lokasi juga mempengaruhi nilai dari Tabel 'ga2' yang memunculkan hasil akhir optimasi algoritma genetika.

```

function delete(){
    $query = "DELETE FROM " . $this->table_name . " WHERE
        id_koordinat = ?; DELETE FROM ga2";

    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $this->id);

    if($result = $stmt->execute()){
        return true;
    }else{
        return false;
    }
}

```

Kode Program 5.15 Penghapusan Data Titik

5.2.3. Pengelolaan Algoritma Genetika

Dalam kegiatan pengelolaan algoritma genetika, terdapat beberapa sub-kegiatan di antaranya pengolahan data dan pengolahan algoritma genetika untuk mendapatkan rute yang optimal.

a. Pengelolaan Data

Proses pengelolaan data dilakukan agar MATLAB mampu mendapatkan informasi mengenai jumlah titik lokasi beserta jarak antartitik dalam bentuk matriks *adjacency*. Data akan diekspor dalam bentuk .csv dan .xls oleh PHP, yang kemudian diolah oleh MATLAB. Kode program 5.16 menunjukkan bagaimana informasi mengenai titik lokasi diekspor ke dalam file berbentuk .xls. Adapun data yang diperoleh adalah koordinat lintang dan longitudo lokasi dari Tabel 'ga1' basis data 'ga'.

```

<?php
header("Content-type: application/vnd-ms-excel");
header("Content-Disposition: attachment; filename=titik.xls");
?>

<table>
  <?php
  $host = "localhost";
  $db_name = "ga";
  $username = "root";
  $password = "";
  $conn = new PDO("mysql:host=" . $host . ";dbname=" . $db_name, $username, $
    password);

  $sql = $conn->prepare("SELECT id_koordinat, lat, lng FROM ga1 ORDER BY
    id_koordinat");
  $sql->execute();

  while($data = $sql->fetch()){ // Ambil semua data dari hasil eksekusi $sql
    echo "<tr>";
    echo "<td>". $data['lat']. "</td>";
    echo "<td>". $data['lng']. "</td>";
    echo "</tr>";
  }
  ?>
</table>

```

Kode Program 5.16 Pengambilan Data dari Basis Data

b. Pengelolaan Algoritma Genetika

Proses pengelolaan algoritma genetika dilakukan oleh MATLAB, seperti yang telah dijelaskan pada subbab 5.1.

5.2.4. Visualisasi Hasil Pengelolaan Algoritma Genetika

Dalam kegiatan visualisasi, terdapat tiga sub-kegiatan di dalamnya, yakni penambahan hasil algoritma ke dalam basis data, penggabungan basis data, serta visualisasi hasil menggunakan peta digital.

a. Penambahan Data Hasil

Penambahan data berupa hasil algoritma genetika yang telah dijalankan sebelumnya dilakukan untuk menyimpan hasil tersebut ke dalam basis data. Basis data yang digunakan adalah Basis Data 'ga' dengan Tabel 'ga2'. Berdasarkan Kode Program 5.17, dapat dilihat bahwa terdapat tiga data yang dimasukkan; data pada kolom 'awal', 'akhir', dan 'id_urutan'.

```

define('CSV_PATH','C:/Users/user/Downloads/'); // specify CSV file path
$csv_file = CSV_PATH . "result.csv"; // Nama file
$csvfile = fopen($csv_file, 'r');
$theData = fgets($csvfile);

$csv_array = explode(",", $theData);
$id = strtotime('s');

for($i=0 ; $i<count($csv_array) ; $i++){
    $sql = "INSERT INTO ga2(akhir, awal, id_urutan) VALUES (?, ?, ?)";
    $conn->prepare($sql)->execute([$csv_array[$i], $i+1, $id]);
}

```

Kode Program 5.17 Penambahan Data Hasil pada Basis Data

b. Penggabungan Tabel Basis Data

Kedua tabel dalam Basis Data 'ga', 'ga1' dan 'ga2', memerlukan koneksi agar hasil yang didapatkan pada sub-kegiatan sebelumnya dapat divisualisasikan. Penggabungan dilakukan dengan fitur SQL 'JOIN' antara kolom 'id_urutan' dari Tabel 'ga1' dengan kolom 'awal' dari Tabel 'ga2'. Proses penggabungan dua tabel tersebut dapat dilakukan dengan Kode Program 5.18 menggunakan fitur 'JOIN'.

```

$data = $conn->query("SELECT ga1.lat FROM ga1 JOIN ga2 ON ga1.
id_koordinat = ga2.akhir ORDER BY ga2.awal")->fetchAll();

```

Kode Program 5.18 Penggabungan tabel 'ga1' dengan 'ga2'

c. Visualisasi Hasil dengan Peta Digital

Visualisasi hasil optimalisasi rute dengan peta digital dilakukan dengan bantuan Google Maps API, sehingga dibutuhkan inisialisasi kunci API (API key) dalam pemrograman PHP. Inisialisasi kunci API dapat dilihat pada Kode Program 5.19.

```

<script async defer src="https://maps.googleapis.com/maps/api/
js?key=AIzaSyBhDrCa4P6ZpJl1a2fkn1qKd-MD_pReVYM&callback=initMap">
</script>

```

Kode Program 5.19 Google Maps API Key

Untuk mengambil data dari basis data, digunakan fitur 'SELECT' pada SQL sesuai dengan Kode Program 5.20.

```

function finalWaypoint($end){
    $query = "SELECT ga1.lat, ga1.lng FROM ga1 JOIN ga2 ON ga1.
        id_koordinat = ga2.lat2 WHERE ga2.lng2 != 1 && ga2.lng2
        != $end ORDER BY ga2.lng2";
    $stmt = $this->conn->prepare( $query );
    $stmt->execute();

    return $stmt;
}

```

Kode Program 5.20 Pengambilan Data untuk Visualisasi

Untuk menggunakan data mengenai koordinat latitude dan longitude dari basis data yang dibutuhkan dalam proses visualisasi, diperlukan adanya *array* berisi titik-titik lokasi. Penyesunan *array* dapat dilihat pada Kode Program 5.21.

```

<?php
$result = array();
while ($row = $way->fetch(PDO::FETCH_ASSOC)){
    $result[] = sprintf("{location:{lat: %s, lng:
        %s},stopover:true}", $row['lat'], $row['lng']);
    }

    $stmt->execute();
    $data = $stmt->fetchAll();
    $number = count($data)-1;
    $resultHasil = join(', ', $result);
?>

```

Kode Program 5.21 Penyesunan Array '\$result'

Adapun untuk penyesuaian peta digital dengan hasil optimalisasi rute dari basis data dapat dilihat pada Kode Program 5.22.

```

function DisplayRoute(directionsService, directionsDisplay, map) {
    directionsService.route({
        origin: {lat:<?php print_r($data[0]['lat'])?>, lng:<?php
            print_r($data[0]['lng'])?>},
        destination: {lat:<?php print_r($data[$number]['lat'])?>, lng
            :<?php print_r($data[$number]['lng'])?>},
        waypoints: [<?php print_r ($resultHasil); ?>],
        optimizeWaypoints: false,
        travelMode: 'DRIVING'
    }, function(response, status) {
        if (status === 'OK') {
            directionsDisplay.setDirections(response);
        } else {
            window.alert('Directions request failed due to ' + status);
        }
    });
}

```

Kode Program 5.22 Pengaturan Google Maps API

Fitur optimalisasi *waypoints* pada aplikasi ini dinonaktifkan untuk mendapatkan rute optimal yang konsisten dan tidak dipengaruhi kondisi lain/tidak biasa, sesuai dengan batasan masalah tugas akhir poin keenam. Untuk tipe perjalanan, digunakan fitur ‘DRIVING’ agar rute yang dipilihkan oleh peta digital mampu dilewati oleh armada truk pengangkut sampah. Berdasarkan Kode Program 5.22, terdapat tiga informasi penting mengenai titik lokasi yang digunakan dalam menampilkan peta digital, yakni ‘origin’, ‘destination’, dan ‘waypoints’.

- **Origin**
‘origin’ menunjukkan titik lokasi di mana rute dimulai. Berdasarkan daftar titik lokasi yang ada pada basis data, maka titik lokasi yang menjadi ‘origin’ adalah titik lokasi dengan kolom ‘id_urutan’ bernilai 1. Dikarenakan *array* dimulai dari 0, maka ‘origin’ diinisialisasi sebagai *array* ke-0 dalam *array* ‘data’.
- **Destination**
‘destination’ menunjukkan titik lokasi terakhir yang dikunjungi sebelum kembali ke titik awal. Untuk menginisialisasi latitud dan longitud titik ‘destination’, diperlukan adanya variabel ‘number’ yang bernilai jumlah titik lokasi yang ada pada basis data, seperti yang dapat dilihat pada Kode Program 5.22.
- **Waypoints**
‘waypoints’ menunjukkan lokasi-lokasi yang dilewati setelah titik pertama (‘origin’) dan sebelum titik terakhir (‘destination’). Pengambilan data mengenai titik lokasi dari basis data untuk ‘waypoints’ dapat dilihat pada Kode Program .5.20.

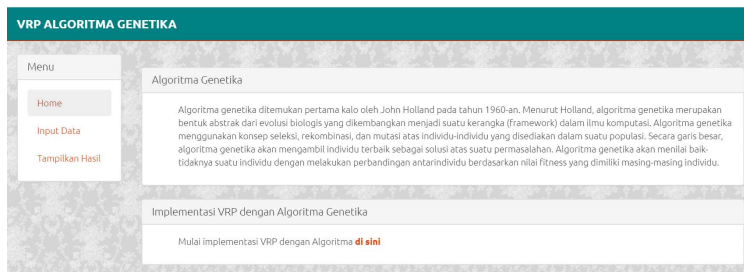
5.3 Antar Muka Aplikasi

Pada bagian ini akan dijelaskan mengenai desain antar muka aplikasi, yang terdiri dari tiga halaman utama; Index, Input

Data, dan Tampilkan Hasil. Tujuan dari pembuatan desain antar muka ini adalah untuk memudahkan *user* dalam menggunakan aplikasi.

5.3.1. Index Antar Muka

Index adalah halaman pertama yang ditampilkan pada *user*. Dikarenakan aplikasi ini bersifat terbuka atau *open*, *user* tidak perlu melakukan membuat akun dan *log in* agar dapat menjalankan aplikasi. Gambar 5.2 menunjukkan tampilan dari Index yang disimpan dalam file ‘index.php’.

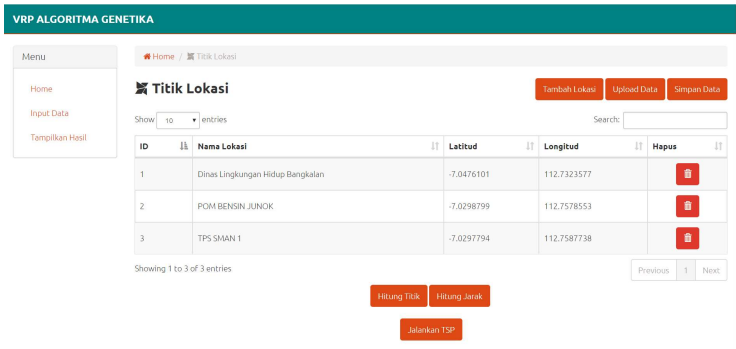


Gambar 5.2 Tampilan Index

Halaman Index terdiri dari menu, penjelasan mengenai algoritma genetika, serta *link* yang menuju pada halaman Input Data.

5.3.2. Input Data Antar Muka

Halaman Input Data merupakan halaman di mana kegiatan input data titik lokasi dilakukan. Sesuai dengan sub-kegiatan pada poin 5.2.2, halaman Input Data memiliki fitur menambah, menyimpan, *upload*, serta menghapus data titik lokasi. Desain antar muka halaman Input Data yang disimpan dalam file ‘nilai.php’ dapat dilihat pada Gambar 5.3.



Gambar 5.3 Tampilan Input Data

Agar dapat menampilkan titik lokasi yang telah dimasukkan ke dalam basis data, dilakukan pemanggilan data sesuai dengan Kode Program 5.23.

```

<table width="100%" class="table table-striped table-bordered" id="tabeldata">
  <thead>
    <tr>
      <th>ID</th>
      <th>Nama Lokasi</th>
      <th>Latitud</th>
      <th>Longitud</th>
      <th width="100px">Hapus</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td style="vertical-align:middle;"><?php echo $row['id_koordinat'] ?></td>
      <td style="vertical-align:middle;"><?php echo $row['ket_koordinat'] ?></td>
      <td style="vertical-align:middle;"><?php echo $row['lat'] ?></td>
      <td style="vertical-align:middle;"><?php echo $row['lng'] ?></td>
      <td class="text-center" style="vertical-align:middle;">
        <a href="nilai-hapus.php?id=<?php echo $row['id_koordinat'] ?>" onclick="return confirm('Yakin ingin menghapus data?')" class="btn btn-danger"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></a>
      </td>
    </tr>
  </tbody>
</table>
  </?php ?>
  </?php ?>
  </tbody>
</table>
  </?php ?>
  </?php ?>
  </tbody>
</table>

```

Kode Program 5.23 Pemanggilan Data

Adapun variabel 'stmt' didapatkan dari fungsi 'readAll' seperti yang dapat dilihat pada Kode Program 5.24. Dalam Kode

Program 5.24, Tabel ‘gal’ diwakilkan oleh ‘\$this->table_name’.

```
function readAll(){
    $query = "SELECT * FROM ".$this->table_name." ORDER BY id_koordinat ASC";
    $stmt = $this->conn->prepare( $query );
    $stmt->execute();

    return $stmt;
}
```

Kode Program 5.23 Penyusunan Variabel ‘\$stmt’

Selain ‘nilai.php’, terdapat halaman lain yang berhubungan dengan halaman Input Data, yakni Tambah Data dan Upload Data.

▪ Tambah Data

Halaman Tambah Data berfungsi untuk menambah data mengenai titik lokasi yang akan dilakukan optimalisasi dengan algoritma genetika. Desain antar muka halaman Tambah Data yang disimpan dalam file ‘nilai-baru.php’ dapat dilihat pada Gambar 5.4.

The screenshot shows a web application interface for 'VRP ALGORITMA GENETIKA'. On the left is a 'Menu' sidebar with options: Home, Input Data, and Tampilkan Hasil. The main content area has a breadcrumb trail: Home / Titik / Tambah Data. Below this is a form titled 'Tambah Koordinat' with the following fields: ID, Nama Lokasi, Latitud, and Longitud. At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

Gambar 5.4 Tampilan Tambah Data

Untuk mendapatkan visualisasi seperti yang dapat dilihat pada Gambar 5.4, diperlukan adanya Kode Program 5.25 dan 5.26.

```

include_once 'includes/nilai.inc.php';
$eks = new Nilai($db);

$eks->lt = $_POST['lt'];
$eks->ln = $_POST['ln'];
$eks->nm = $_POST['nm'];
$eks->id = $_POST['id'];

if($eks->insert()){

```

Kode Program 5.25 Inisialisasi Variabel Data

```

<form method="post">
  <div class="form-group">
    <label for="ln">ID</label>
    <input type="text" class="form-control" id="id" name="id"
      required>
  </div>
  <div class="form-group">
    <label for="nm">Nama Lokasi</label>
    <input type="text" class="form-control" id="nm" name="nm"
      required>
  </div>
  <div class="form-group">
    <label for="lt">Latitud</label>
    <input type="text" class="form-control" id="lt" name="lt"
      required>
  </div>
  <div class="form-group">
    <label for="ln">Longitud</label>
    <input type="text" class="form-control" id="ln" name="ln"
      required>
  </div>
  <button type="submit" class="btn btn-primary"><span class="fa
fa-save"></span> Simpan</button>
  <button type="button" onclick="location.href='nilai.php'" class
="btn btn-success"><span class="fa fa-history"></span> Kembali
</button>
</form>

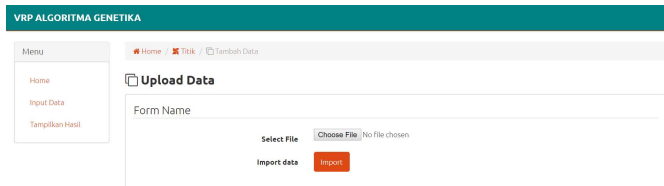
```

Kode Program 5.26 Tampilan Keseluruhan Data

Kode program 5.26 menampilkan inisialisasi variabel dengan penjabaran sebagai berikut; 'lt' bernilai koordinat latitud, 'ln' bernilai koordinat longitud, 'id' bernilai ID titik lokasi, dan 'nm' bernilai nama atau keterangan titik lokasi. Kode Program 5.26 menampilkan kolom ID, keterangan lokasi, koordinat latitud dan longitud pada lokasi terkait, serta tombol simpan.

▪ Upload Data

Halaman Upload Data bertujuan untuk mengunggah data titik lokasi yang telah disimpan sebelumnya dengan fitur Simpan Data. Desain antar muka halaman Upload Data yang disimpan dalam file ‘uploaddata.php’ dapat dilihat pada Gambar 5.5.



Gambar 5.5 Tampilan Upload Data

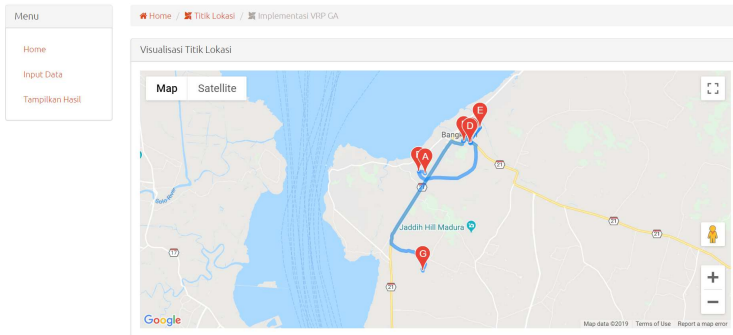
Untuk mendapatkan visualisasi seperti pada Gambar 5.5, diperlukan adanya Kode Program 5.27. Kode Program 5.27 menampilkan form yang berisi tombol ‘Choose File’ untuk memilih file, serta tombol ‘Import’ untuk memasukkan data pada file yang diunggah ke dalam basis data.

```
<form class="form-horizontal" action="uploaddataa.php" method="post" name
="upload_excel" enctype="multipart/form-data">
  <fieldset>
    <!-- Form Name -->
    <legend>Form Name</legend>
    <!-- File Button -->
    <div class="form-group">
      <label class="col-md-4 control-label" for="filebutton">
        Select File</label>
      <div class="col-md-4">
        <input type="file" name="file" id="file" class="
        input-large">
      </div>
    </div>
    <!-- Button -->
    <div class="form-group">
      <label class="col-md-4 control-label" for="singlebutton">
        Import data</label>
      <div class="col-md-4">
        <btn type="submit" id="submit" name="Import" class="
        btn btn-primary button-loading" data-loading-text="
        Loading...">Import</button>
      </div>
    </div>
  </fieldset>
</form>
```

Kode Program 5.27 Tampilan Upload Data

5.3.3. Tampilkan Hasil Antar Muka

Halaman Tampilkan Hasil merupakan halaman yang menampilkan fitur visualisasi peta digital yang menggambarkan hasil optimasi rute dengan bantuan Google Maps API. Desain antar muka halaman Tampilkan Hasil yang disimpan dalam file 'hasil.php' dapat dilihat pada Gambar 5.6.



Gambar 5.6 Tampilan Hasil

Sesuai dengan Kode Program 5.28, peta digital divisualisasikan dengan fungsi 'initMap'.

```
function initMap() {
  var directionsService = new google.maps.DirectionsService;
  var directionsDisplay = new google.maps.DirectionsRenderer;
  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 7,
  });
  directionsDisplay.setMap(map);
  directionsDisplay.setPanel(document.getElementById('right-panel'));
  calculateAndDisplayRoute(directionsService, directionsDisplay, map);
}
```

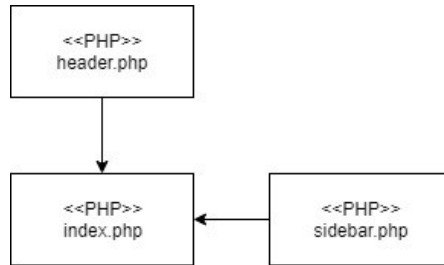
Kode Program 5.28 Tampilan Peta

5.4 Penjelasan Alur Kerja Sistem Aplikasi

Subbab ini menjelaskan alur kerja sistem aplikasi yang telah disusun pada subbab 5.2 dan 5.3. Alur kerja sistem aplikasi dibagi menjadi tiga bagian sesuai dengan halaman antar muka aplikasi, yakni Index, Input Data, dan Tampilkan Hasil.

5.4.1. Index Alur Kerja

Diagram aplikasi sistem pada halaman Index dapat dilihat pada Gambar 5.7.



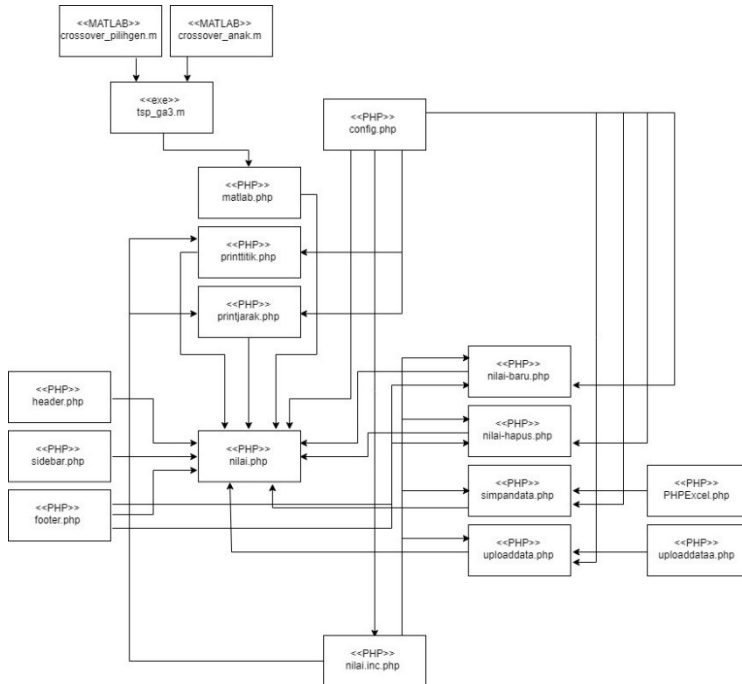
Gambar 5.7 Diagram Aplikasi Sistem pada Halaman Index

Halaman Index melibatkan tiga file PHP, yakni 'index.php', 'header.php', dan 'sidebar.php'.

- index.php
File 'index.php' merupakan file utama pada halaman Index.
- header.php
File 'header.php' berisi kode program yang memiliki tanggung jawab dalam pengembangan *front-end* sisi atas aplikasi.
- sidebar.php
File 'sidebar.php' berisi kode program yang menampilkan menu aplikasi di sisi kiri.

5.4.2. Input Data Alur Kerja

Diagram aplikasi sistem pada halaman Input Data dapat dilihat pada Gambar 5.8. Untuk file 'sidebar.php' dan 'header.php', keduanya memiliki fungsi yang sama seperti yang telah dijelaskan pada bagian 5.4.1.



Gambar 5.8 Diagram Aplikasi Sistem pada Halaman Input Data

Halaman Nilai mencakup 15 file PHP dan tiga file MATLAB (.m).

- nilai.php
File ‘nilai.php’ merupakan file utama dari halaman Input Data yang menampilkan ID, keterangan, serta koordinat lintud dan longitud titik lokasi yang disimpan dalam basis data.
- footer.php
File ‘footer.php’ berisi kode program yang memiliki responsibilitas dalam tampilan notifikasi jika data berhasil dihapus atau ditambahkan ke dalam basis data. ‘footer.php’ juga menampilkan notifikasi jika terjadi

kegagalan dalam penghapusan maupun penambahan data.

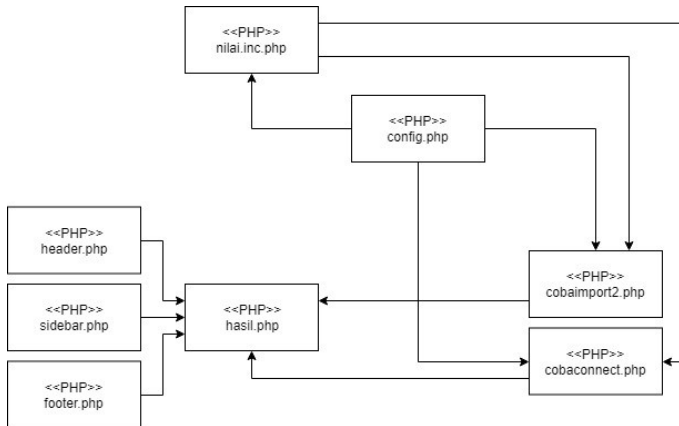
- nilai-baru.php
File 'nilai-baru.php' berisi kode program yang dapat melakukan penambahan data mengenai titik lokasi.
- nilai-hapus.php
File 'nilai-hapus.php' berisi kode program yang dapat melakukan penghapusan data mengenai titik lokasi yang ada dalam basis data.
- simpandata.php
File 'simpandata.php' berisi kode program yang dapat melakukan penyimpanan data suatu rute berisi titik-titik lokasi yang akan dioptimasi.
- PHPExcel.php
File 'PHPExcel.php' berisi kode program yang dapat membantu 'simpandata.php' melakukan penyimpanan data.
- uploaddata.php
File 'uploaddata.php' berisi kode program yang menampilkan halaman khusus untuk mengunggah data rute yang telah disimpan sebelumnya.
- uploaddataa.php
File 'uploaddataa.php' berisi kode program berupa fungsi yang dapat membantu pengunggahan serta penyimpanan data yang diunggah ke dalam basis data. Dengan kata lain, file 'uploaddataa.php' merupakan fungsi dari file 'uploaddata.php'.
- printjarak.php
File 'printjarak.php' berisi kode program yang dapat melakukan penyimpanan informasi mengenai jarak antartitik dalam bentuk matriks *adjacency*.
- printtitik.php
File 'printtitik.php' berisi kode program yang dapat melakukan penyimpanan informasi mengenai titik lokasi yang akan dioptimasi.

- **matlab.php**
File ‘matlab.php’ berisi kode program yang dapat menjalankan file ‘tsp_ga3.exe’ untuk melakukan optimasi dengan algoritma genetika.
- **config.php**
File ‘config.php’ berisi kode program yang menyambungkan file-file PHP dalam aplikasi dengan basis data. File ‘config.php’ digunakan pada berbagai macam fitur yang berhubungan dengan data titik lokasi, seperti penambahan dan penghapusan data titik lokasi, pengunggahan dan penyimpanan data rute, dan lainnya.
- **nilai.inc.php**
File ‘nilai.inc.php’ berisi fungsi-fungsi yang digunakan oleh file-file PHP lainnya. Fungsi-fungsi yang dicakup oleh ‘nilai.inc.php’ berupa fitur-fitur yang tersedia pada aplikasi.
- **tsp_ga3.m**
File ‘tsp_ga3.m’ merupakan kode program MATLAB yang merupakan file utama dalam melakukan proses algoritma genetika.
- **crossover_pilihgen.m**
File ‘crossover_pilihgen.m’ mencakup kode program MATLAB yang dapat membantu file ‘tsp_ga3.m dalam proses rekombinasi dalam algoritma genetika.
- **crossover_anak.m**
File ‘crossover_pilihgen.m’ mencakup kode program MATLAB yang dapat membantu file ‘tsp_ga3.m dalam proses rekombinasi dalam algoritma genetika.

5.4.3. Tampilkan Hasil Alur Kerja

Diagram aplikasi sistem pada halaman Tampilkan Hasil dapat dilihat pada Gambar 5.9. Untuk file ‘sidebar.php’ dan ‘header.php’, keduanya memiliki fungsi yang sama seperti yang telah dijelaskan pada bagian 5.4.1. Untuk file ‘footer.php’, ‘config.php’, dan ‘nilai.inc.php’, file-file tersebut memiliki

fungsi yang sama seperti yang telah dijelaskan pada bagian 5.4.1.



Gambar 5.9 Diagram Aplikasi Sistem pada Halaman Tampilkan Hasil

Halaman Tampilkan Hasil melibatkan delapan file PHP.

- hasil.php
File 'hasil.php' merupakan file utama dari halaman Tampilkan Hasil yang menampilkan rute hasil optimasi menggunakan algoritma genetika pada peta digital dengan bantuan Google Maps API.
- cobaimport2.php
File 'cobaimport2.php' berisi fungsi yang mampu memasukkan hasil optimasi algoritma genetika yang dijalankan oleh file 'tsp_ga3.exe' ke dalam basis data.
- cobacconnect.php

File 'cobacconnect.php' berisi fungsi yang mampu melakukan penggabungan antara data titik lokasi dengan hasil optimasi algoritma genetika di dalam basis data untuk membantu visualisasi hasil optimasi pada peta digital.

Halaman ini sengaja dikosongkan

BAB VI HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dari pembuatan serta implementasi optimasi dengan algoritma genetika. Bab ini akan menjelaskan lingkungan uji coba, uji coba program optimasi rute, perbandingan hasil uji coba, hasil rekomendasi rute baru, serta analisa hasil pengujian.

6.1. Lingkungan Uji Coba

Pada subbab Lingkungan Uji Coba ini akan menjelaskan terkait lingkungan pengujian dalam melakukan implementasi penelitian tugas akhir terkait optimasi rute pengangkutan sampah di Kabupaten Bangkalan. Spesifikasi perangkat keras yang digunakan dalam implementasi ditunjukkan pada Tabel 6.1.

Tabel 6.1 Lingkungan Uji Coba Perangkat Keras

Perangkat Keras	Spesifikasi
Jenis	Laptop ASUS A456U
Processor	Intel Core i7-7500U 3.56Hz
RAM	8 GB

Untuk spesifikasi perangkat lunak yang digunakan dalam pengerjaan tugas akhir ditunjukkan pada Tabel 6.2.

Tabel 6.2 Lingkungan Uji Coba Perangkat Lunak

Perangkat Lunak	Fungsi
Windows 10 64 bit	Sistem Operasi
Microsoft Excel 2013	Mengolah data
XAMPP Control Panel v3.2.3	Sebagai server untuk melakukan uji coba
MATLAB R2018b	Mengolah kode program MATLAB
Sublime Text 3	Mengolah kode program aplikasi berbasis web
Google Chrome	<i>Browser</i> untuk melakukan uji coba aplikasi

6.2. Uji Fungsional dan Non-Fungsional

Pada subbab ini akan dijelaskan mengenai hasil uji fungsional dan non-fungsional terhadap aplikasi.

6.2.1. Uji Fungsional

Uji fungsional mencakup pengujian terhadap masing-masing *use case* aplikasi. Hasil pengujian fungsional aplikasi dapat dilihat pada Tabel 6.3.

Tabel 6.3 Uji Fungsional

No.	<i>Use Case</i>	Keterangan
1.	Memasukkan data	Berhasil
2.	Mengolah data	Berhasil
3.	Menampilkan hasil	Berhasil

6.2.2. Uji Non-Fungsional

Uji non-fungsional mencakup pengujian terhadap skalabilitas, kompatibilitas, serta reliabilitas aplikasi.

Pengujian skalabilitas mencakup seberapa banyak titik lokasi yang dapat dioptimasi oleh aplikasi. Hasil pengujian skalabilitas aplikasi dapat dilihat pada Tabel 6.4.

Tabel 6.4 Uji Non-Fungsional Skalabilitas

No.	Jumlah Titik yang Diuji	Keterangan	Status
1.	10	Berhasil	-
2.	15	Berhasil	-
3.	20	Berhasil	-
4.	25	Berhasil	-
5.	30	Gagal	MAX_WAYPOINTS EXCEEDED

Uji kompatibilitas mencakup seberapa jauh aplikasi dapat dijalankan pada sistem operasi, penjelajah web (*web browser*), maupun perangkat keras yang berbeda. Hasil pengujian

kompabilitas aplikasi berdasarkan sistem operasi yang digunakan dapat dilihat pada Tabel 6.5.

Tabel 6.5 Uji Non-Fungsional Kompatibilitas terhadap Sistem Operasi

No.	Sistem Operasi	Status
1.	Windows 10	Berjalan Normal
2.	Windows 8	Berjalan Normal

Hasil pengujian kompabilitas aplikasi berdasarkan penjelajah web yang digunakan dapat dilihat pada Tabel 6.6.

Tabel 6.6 Uji Non-Fungsional Kompatibilitas terhadap Penjelajah Web

No.	Penjelajah Web	Status
1.	Google Chrome	Berjalan Normal
2.	Internet Explorer	Berjalan Normal
3.	Mozilla Firefox	Berjalan Normal

Hasil pengujian kompabilitas aplikasi berdasarkan perangkat keras yang digunakan dapat dilihat pada Tabel 6.7.

Tabel 6.7 Uji Non-Fungsional Kompatibilitas terhadap Perangkat Keras

No.	Perangkat Keras Processor	Status
1.	i7	Berjalan Normal
2.	i5	Berjalan Normal

Uji reliabilitas mencakup seberapa jauh aplikasi dapat digunakan kembali jika terjadi pemutusan koneksi internet, penonaktifan aplikasi pendukung XAMPP, serta penonaktifan perangkat keras ketika aplikasi sedang berjalan. Hasil pengujian reliabilitas aplikasi dapat dilihat pada Tabel 6.8.

Tabel 6.8 Uji Non-Fungsional Reliabilitas

No.	Aksi yang Dilakukan	Keterangan
1.	Pemutusan jaringan internet	Berjalan Normal
2.	Penonaktifan XAMPP	Berjalan Normal

No.	Aksi yang Dilakukan	Keterangan
3.	Penonaktifan perangkat keras	Berjalan Normal

6.3. Uji Coba Program Pencarian Rute

Uji coba program pencarian rute dilakukan dengan mengubah parameter 'alpha' yang merepresentasikan kinerja proses rekombinasi, serta parameter 'beta' dan 'gamma' yang merepresentasikan kinerja proses mutasi dalam algoritma genetika. Uji coba dilakukan pada satu rute, yakni rute yang dilewati oleh armada ke-1 dengan pengemudi Djumaidi. Nilai *fitness* yang menyatakan nilai solusi direpresentasikan dengan variabel 'distance'. Semakin kecil nilai solusi yang didapatkan, semakin bagus solusi tersebut.

Nilai-nilai yang digunakan untuk masing-masing parameter adalah 0,05, 0,5, dan 1. Uji coba pada kedua parameter pertama, 0,05 dan 0,5, dilakukan untuk membuktikan ada atau tidaknya pola tertentu dari solusi-solusi yang didapatkan dengan rentang parameter 10^1 . Parameter uji coba terakhir, 1, dilakukan untuk mengetahui optimalisasi solusi dengan nilai maksimal parameter.

Hasil uji coba program pencarian rute dapat dilihat pada Tabel 6.9.

Tabel 6.9 Hasil Uji Coba dengan Mengubah Parameter

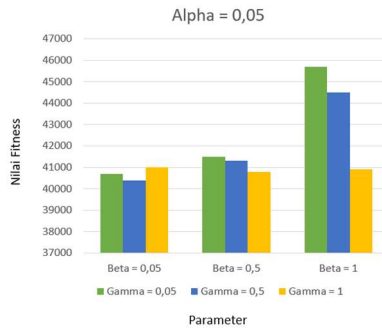
Skenario ke-	Alpha	Beta	Gamma	Nilai <i>Fitness</i>
1	0,05	0,05	0,05	40.700
2			0,5	40.400
3			1	41.000
4		0,5	0,05	41.500
5			0,5	41.300
6			1	40.800
7		1	0,05	45.700
8			0,5	44.500
9			1	40.900

Skenario ke-	Alpha	Beta	Gamma	Nilai <i>Fitness</i>
10	0,5	0,05	0,05	44.900
11			0,5	40.800
12			1	40.800
13		0,5	0,05	41.200
14			0,5	40.500
15			1	40.700
16		1	0,05	40.400
17			0,5	40.400
18			1	41.200
19	1	0,05	0,05	40.500
20			0,5	44.900
21			1	40.700
22		0,5	0,05	40.500
23			0,5	40.400
24			1	40.400
25		1	0,05	40.700
26			0,5	40.700
27			1	40.500

Hasil Uji Coba Program Pencarian Rute

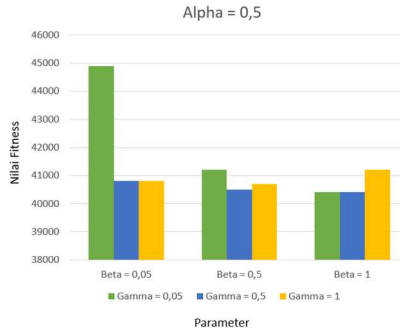
Berdasarkan Tabel 6.9, dapat disimpulkan bahwa Skenario ke-2, 16, 17, 23, dan 24 mampu menghasilkan jumlah jarak terpendek dibandingkan seluruh skenario yang ada dengan solusi yang didapatkan sebesar 40.400.

Diagram mengenai solusi yang didapatkan dari parameter alpha = 0,05 dapat dilihat pada Gambar 6.1. Selain parameter gamma = 1, nilai *fitness* yang didapatkan semakin baik jika parameter beta yang digunakan semakin kecil.



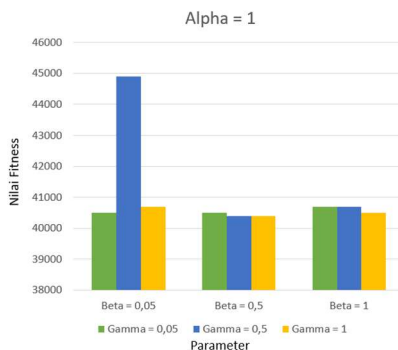
Gambar 6.1 Diagram uji coba parameter alpha = 0,05

Diagram mengenai solusi yang didapatkan dari parameter alpha = 0,5 dapat dilihat pada Gambar 6.2. Untuk parameter gamma = 0,05, semakin besar nilai parameter beta, semakin baik nilai *fitness* yang dihasilkan. Hal ini tidak berlaku bagi skenario dengan nilai parameter gamma 0,5 dan 1.



Gambar 6.2 Diagram uji coba parameter alpha = 0,5

Diagram mengenai solusi yang didapatkan dari parameter alpha = 1 dapat dilihat pada Gambar 6.3. Keseluruhan skenario dengan parameter alpha = 1 tidak memiliki pola khusus terhadap nilai *fitness* yang dihasilkan.



Gambar 6.3 Diagram uji coba parameter alpha = 1

6.4. Optimasi Rute

Subbab ini menjelaskan mengenai pengujian terhadap program optimasi rute yang telah dirancang dengan pemrograman MATLAB. Hasil yang diharapkan dari program adalah adanya rute yang lebih optimal dibandingkan dengan rute yang ada saat ini, sehingga hasil program tersebut dapat menjadi rekomendasi dalam pengambilan keputusan.

Uji coba dilakukan dengan menjalankan algoritma genetika pada MATLAB dengan jumlah populasi sebesar 1000 dan generasi berjumlah 50. Skenario yang digunakan adalah Skenario 2, di mana nilai parameter alpha = 0,05, beta = 0,05, dan gamma = 0,5. Skenario 2 merupakan salah satu dari lima skenario yang mampu menghasilkan nilai fitness paling optimal. Hasil kerja program untuk setiap rute pengangkutan sampah di Kabupaten Bangkalan dapat dilihat pada Tabel 6.10.

Tabel 6.10 Hasil Uji Coba Program

No.	Armada ke-	Pengemudi	Hasil Urutan Rute
1.	1	Djumaidi	1,6,7,2,3,5,4,8,9
2.	2	Imam Subakti (Pagi)	1,8,4,6,5,7,2,3,9
3.	3	Abdul Rahman	1,2,3,8,4,5,9,6,7,10

No.	Armada ke-	Pengemudi	Hasil Urutan Rute
4.	4	Edy Hartono	1,5,8,7,6,4,3,2,9
5.	5	Samsul Hidayat	1,4,5,2,3,7,6,8
6.	6	Imam Subakti (Siang)	1,5,3,2,4,6,7

6.5. Perbandingan Hasil Program Algoritma Genetika

Subbab ini akan menjelaskan perbandingan antara hasil optimasi dengan kondisi saat ini pada rute pengangkutan sampah di Kabupaten Bangkalan. Perbandingan dilakukan berdasarkan jarak tempuh rute beserta visualisasinya.

6.5.1. Perbandingan Jarak Tempuh Rute

Perbandingan jarak tempuh rute baru hasil optimasi dengan rute saat ini dijelaskan pada Tabel 6.11 dan 6.12.

Tabel 6.11 Perbandingan Jarak Tempuh Rute

Armada ke-	Rute Saat Ini			Rute Baru		
	Dari Titik ke-	Menuju Titik ke-	Jarak (km)	Dari Titik ke-	Menuju Titik ke-	Jarak (km)
1	1	2	5,30	1	6	1,50
	2	3	1,30	6	7	0,12
	3	4	0,40	7	2	3,60
	4	5	0,17	2	3	1,30
	5	6	2,20	3	5	0,24
	6	7	0,12	5	4	0,17
	7	8	13,70	4	8	15,60
	8	9	9,30	8	9	9,30
	9	1	8,60	9	1	8,60
Jumlah Jarak			41,09	Jumlah Jarak		40,43
2	1	2	3,70	1	8	0,80
	2	3	0,24	8	4	4,20
	3	4	0,60	4	6	0,24
	4	5	3,70	6	5	0,60
	5	6	0,35	5	7	3,20
	6	7	0,11	7	2	0,11

Armada ke-	Rute Saat Ini			Rute Baru		
	Dari Titik ke-	Menuju Titik ke-	Jarak (km)	Dari Titik ke-	Menuju Titik ke-	Jarak (km)
	7	8	4,30	2	3	0,35
	8	9	9,30	3	9	11,70
	9	1	8,60	9	1	8,60
Jumlah Jarak			30,90	Jumlah Jarak		29,80
3	1	2	0,072	1	2	0,072
	2	3	0,7	2	3	0,70
	3	4	0,85	3	8	0,45
	4	5	0,35	8	4	0,40
	5	6	0,12	4	5	0,35
	6	7	0,55	5	9	0,07
	7	8	0,85	9	6	0,11
	8	9	0,052	6	7	0,55
	9	10	9,20	7	10	8,50
	10	1	8,60	10	1	8,60
Jumlah Jarak			21,344	Jumlah Jarak		19,802
4	1	2	2,10	1	5	1,90
	2	3	0,081	5	8	0,90
	3	4	0,40	8	7	0,26
	4	5	0,75	7	6	0,65
	5	6	0,27	6	4	0,45
	6	7	0,65	4	3	0,40
	7	8	0,26	3	2	0,081
	8	9	10,80	2	9	9,60
	9	1	8,60	9	1	8,60
Jumlah Jarak			23,911	Jumlah Jarak		22,841
5	1	2	3,60	1	4	3,60
	2	3	0,60	4	5	0,074
	3	4	0,45	5	2	0,25
	4	5	0,074	2	3	0,60
	5	6	1,40	3	7	0,35
	6	7	1,10	7	6	0,14
	7	8	11,50	6	8	11,40
	8	1	8,60	8	1	8,60
Jumlah Jarak			27,324	Jumlah Jarak		25,014
6	1	2	4,70	1	5	0,55
	2	3	0,081	5	3	5,90
	3	4	1,50	3	2	0,13

Armada ke-	Rute Saat Ini			Rute Baru		
	Dari Titik ke-	Menuju Titik ke-	Jarak (km)	Dari Titik ke-	Menuju Titik ke-	Jarak (km)
	4	5	5,80	2	4	1,30
	5	6	4,70	4	6	1,70
	6	7	11,90	6	7	11,90
	7	1	8,60	7	1	8,60
Jumlah Jarak			37,281	Jumlah Jarak		30,080
Total			181,850	Total		167,967

Tabel 6.12 Perbandingan Total Jarak Tempuh Rute

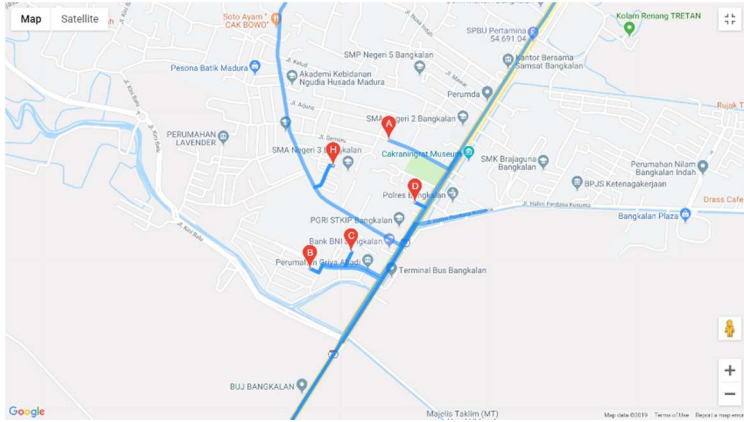
Armada ke-	Jarak Tempuh	
	Rute Saat Ini (km)	Rute Baru (km)
1	41,090	40,430
2	30,900	29,800
3	21,344	19,802
4	23,911	22,841
5	27,324	25,014
6	37,281	30,080
Total	181,850	167,967
Selisih	13,883	
Persentase Selisih	10,66%	

Berdasarkan Tabel 6.11 dan 6.12, dapat dilihat bahwa total jarak tempuh pada rute lama sebesar 181,850 km, sementara total jarak tempuh rute baru sebesar 167,967 km. Dari total jarak tempuh tersebut, didapatkan selisih antara rute lama dan rute baru sebesar 13,883 km.

6.5.2. Perbandingan Visualisasi Rute

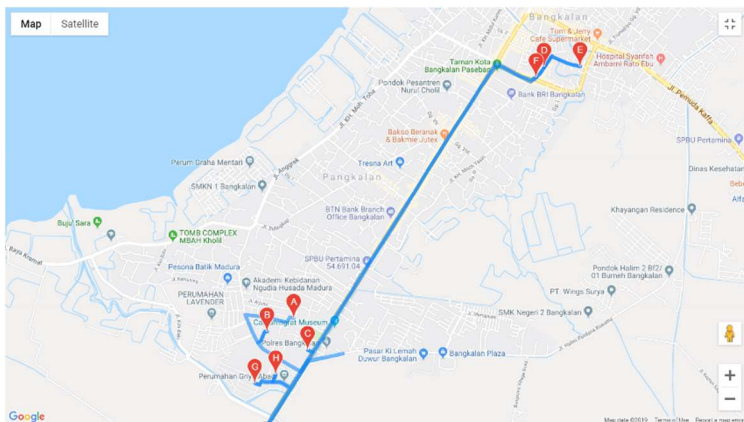
Subbab ini akan memaparkan visualisasi rute lama dan rute baru, sebagai pembanding visual. Perbandingan visualisasi rute dilakukan oleh masing-masing armada.

Gambar 6.1 menampilkan contoh visualisasi rute armada kedua saat ini, yang belum dilakukan optimasi.



Gambar 6.4 Contoh Visualisasi Rute Saat Ini

Gambar 6.2 menampilkan contoh visualisasi rute armada kedua yang dihasilkan yang direkomendasikan. Rute tersebut dimulai dari Pool – TPS SMA 3 – TPS Coca Cola – TPS Musholla Sak-Sak Timur – TPS Dr. Surata – TPS Jl. Kartini – Depo Perumahan Griya Abadi – TPS IKIP – dan berakhir di TPA.



Gambar 6.5 Contoh Visualisasi Rute Baru

Hasil visualisasi keseluruhan rute dapat dilihat pada Lampiran C.

6.6. Hasil Rekomendasi Rute Baru

Subbab ini menjelaskan hasil rekomendasi rute baru yang telah dirancang sebelumnya. Hasil rekomendasi rute baru dapat dilihat pada Tabel 6.13.

Tabel 6.13 Hasil Rekomendasi Rute Baru

Armada ke-	Pengemudi	Rekomendasi Rute
1	Djumaidi	Pool – TPS Radio Amana FM – TPS DISHUTBUN – TPS Masjid Junok – TPS Perumahan Halim 2 – TPS Perumahan Griya Utama – TPS Perumahan Halim 1 – TPS Perumahan Griya Utama – TPA
2	Imam Subakti (Pagi)	Pool – TPS SMA 3 – TPS Coca Cola – TPS Musholla Sak-Sak Timur – TPS Dr. Surata – TPS Jl. Kartini – Depo Perumahan Griya Abadi – TPS IKIP – TPA
3	Abdul Rahman	Pool – TPS SMAN 2 – TPS Pengadilan – Kantor BRI – Depo Senenan – TPS PUMARA – Kantor Pajak – TPS Stadion – TPS Pertanian - TPA
4	Edy Hartono	Pool – TPS RM Balado – TPS Pondok Pesantren KH. Kholil – TPS Jl. KHM. Kholil Gang VII – TPS Kelurahan Demangan – Titik Embun – TPS Las Pak Ujang – Lampu Merah Senenan – TPA
5	Samsul Hidayat	Pool – TPS SDN Kraton 1 – TPS PLN – TPS SMPN 1 – TPS Toko Bok-Bok – TPS Jl. Mayjen Sungkono – Depo Jl. Mayjen Sungkono – TPA
6	Imam Subakti (Siang)	Pool – TPS RE. Martadinata – TPS SMAN 1 – Pom Bensin Junok – SMA 4 – Pondok Pesantren Al-Kholiliyah – TPA

BAB VII

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan kesimpulan yang didapat dari pengerjaan penelitian tugas akhir ini dan saran yang dapat diberikan untuk penelitian selanjutnya.

7.1. Kesimpulan

Berdasarkan hasil yang telah diuraikan pada bagian sebelumnya, kesimpulan yang dapat diambil adalah:

- 1) Algoritma genetika memiliki kapabilitas dalam optimasi rute pengangkutan sampah. Penghematan jarak yang dihasilkan oleh algoritma genetika sebesar 10,66%.
- 2) Visualisasi rekomendasi rute yang dihasilkan dilakukan dengan menggunakan fitur Google Maps API.

7.2. Saran

Berdasarkan hasil penelitian tugas akhir ini, maka saran yang dapat diberikan untuk membantu penelitian selanjutnya adalah sebagai berikut:

- 1) Data yang digunakan merupakan data yang diperoleh dari Dinas Lingkungan Hidup Kabupaten Bangkalan mengenai rute pengangkutan sampah di Kota Bangkalan. Ada baiknya jika aplikasi yang dikembangkan digunakan untuk mengoptimasi rute pengangkutan sampah di seluruh kecamatan dan kelurahan Kabupaten Bangkalan.
- 2) Penelitian tugas akhir ini hanya mencakup pengangkutan sampah bertipe *Stationary Container System* (SCS), sehingga ada baiknya jika aplikasi dapat diterapkan untuk optimasi rute pengangkutan sampah tipe *Stationary Container System* (SCS) dan *Hauled Container System* (HCS).

- 3) Algoritma yang digunakan dalam tugas akhir ini adalah algoritma genetika. Penggunaan algoritma yang lain seperti PSO (*Particle Swarm Optimization*), algoritma semut, atau algoritma lebah dapat dilakukan untuk membantu pencarian rute optimal.
- 4) Optimasi rute sebaiknya memperhitungkan volume TPS, jumlah dan kapasitas armada truk pengangkut sampah, kondisi jalan, jam kerja, dan berbagai macam kebijakan yang terkait dengan pengelolaan pengangkutan sampah di Kabupaten Bangkalan.

DAFTAR PUSTAKA

- [1] Tim VIVA, “Miris, Satu Orang Hasilkan 0,7Kg Sampah Tiap Harinya,” 25 Januari 2019. [Online]. Available: <https://www.viva.co.id/gaya-hidup/kesehatan-intim/1058125-miris-satu-orang-hasilkan-0-7kg-sampah-tiap-harinya>.
- [2] B. Chandra, Pengantar Kesehatan Lingkungan, Jakarta: EGC, 2006.
- [3] Badan Pusat Statistik Kabupaten Bangkalan, Kabupaten Bangkalan Dalam Angka, Bangkalan: BPS Kabupaten Bangkalan, 2018.
- [4] M. Effendi dan R. Hendarto, “Dampak Pembangunan Jembatan Suramadu terhadap Perekonomian Pulau Madura (Studi Kasus Kabupaten Bangkalan),” *Diponegoro Journal of Economics*, pp. 185-197, 2014.
- [5] Madura News Media, “Volume Sampah Di Kota Bangkalan Meningkat 100 Persen/Tahun,” 23 Maret 2017. [Online]. Available: <http://maduranewsmedia.com/terkini/volume-sampah-di-kota-bangkalan-meningkat-100-persentahun/>.
- [6] D. Heryanto, “Dalam Sehari, Volume Sampah Di Bangkalan Capai 212 Meter Kubik,” 3 November 2015. [Online]. Available: <https://www.maduracorner.com/dalam-sehari-volume-sampah-di-bangkalan-capai-212-meter-kubik/>.
- [7] N. Joga, “Bersama Menuntaskan Sampah,” Investor Daily Indonesia, 23 Februari 2019. [Online]. Available: <https://id.beritasatu.com/home/bersama->

- menuntaskan-sampah/185783. [Diakses 13 Maret 2019].
- [8] M. A. Mohammed, M. S. Ahmad dan S. A. Mostafa, "Using Genetic Algorithm in Implementing Capacitated Vehicle Routing Problem," *International Conference on Computer & Information Science (ICCIS)*, pp. 257-262, 2012.
- [9] R. Assaf dan Y. Saleh, "Vehicle-Routing Optimization for Municipal Solid Waste Collection Using Genetic Algorithm: The Case of Southern Nablus City," *De Gruyter*, vol. 26, no. 3, pp. 43-57, 2017.
- [10] E.-G. Talbi, *Metaheuristics - From Design to Implementation*, New Jersey: John Wiley & Sons, 2009.
- [11] B. L. Golden, S. Raghavan dan E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New*, New York: Springer Science & Business Media, 2008.
- [12] M. Mitchell, *An Introduction to Genetic Algorithms*, Massachusetts: MIT Press, 1998.

BIODATA PENULIS



Penulis bernama lengkap Sarah Fauziyyah Sekarsari, lahir di Semarang, Jawa Tengah pada tanggal 26 Oktober 1997. Penulis merupakan anak pertama dari dua bersaudara. Pada tahun 2003 hingga 2007, penulis menempuh pendidikan dasar di SDIT Al-Hikmah Jakarta, dilanjutkan dengan pendidikan dasar di SD Al-Falah Surabaya pada tahun 2007 hingga 2009. Penulis melanjutkan

studinya di SMP Al-Hikmah Surabaya pada tahun 2009 hingga 2012, dan SMA Al-Hikmah Surabaya pada tahun 2012 hingga lulus pada tahun 2015. Selanjutnya penulis melanjutkan pendidikan tinggi di Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember (ITS) Surabaya angkatan tahun 2015 dengan NRP 05215 4000 0076.

Selama menempuh pendidikan tinggi di ITS, penulis aktif mengikuti berbagai kegiatan, di antaranya adalah menjadi Ketua Divisi ITS TV tahun 2016 dan 2017, Ketua Divisi IM HMSI tahun 2017, staf Dokumentasi Kreatif ISE 2016, dan Ketua Divisi Dokumentasi Kreatif ISE 2017. Selain aktif dalam organisasi dan kepanitiaan, penulis juga mengikuti beberapa pelatihan diantaranya SAP University Alliance Course yang diadakan Departemen Sistem Informasi. Di tahun terakhir penulis menyusun tugas akhir di bidang Rekayasa Data dan Intelegensia Bisnis.

Untuk mengetahui lebih lanjut mengenai penelitian ini maupun hal yang berkaitan dengan penulis, dapat menghubungi e-Mail penulis, ssarahfauziyyah@gmail.com

Halaman ini sengaja dikosongkan

LAMPIRAN A: Lokasi Tempat Pembuangan Sampah (TPS) dengan Sistem Stationary Container System (SCS)

Lokasi Tempat Pembuangan Sampah (TPS) dengan *Sistem Stationary Container System* (SCS) di Kabupaten Bangkalan berdasarkan data dari DLH Bangkalan yang diperoleh pada tahun 2019.

No.	Depo/TPS	Latitud	Longitud
1.	Depo Perum Griya Abadi	-7.0522527	112.729527
2.	Depo Senenan	-7.0392918	112.7406358
3.	Depo Jl. Mayjen Sungkono	-7.030617	112.751247
4.	Kantor BRI	-7.0424095	112.7386265
5.	Kantor Pajak	-7.0420144	112.738886
6.	Lampu Merah Senenan	-7.0388905	112.7409389
7.	Pom Bensin Junok	-7.0298799	112.7578553
8.	TPS Masjid Junok	-7.0319093	112.7614287
9.	TPS Perum Halim 1	-7.0459718	112.7583679
10.	TPS Perum Halim 2	-7.0433001	112.7606889
11.	TPS Perum Griya Utama	-7.0451257	112.759677
12.	TPS Radio Amana FM	-7.0499284	112.7409017
13.	TPS Dishutbun	-7.04997	112.7419873
14.	TPS Perum Cendana	-7.1584922	112.7183424
15.	TPS IKIP	-7.0515541	112.7312933
16.	TPS Coca Cola	-7.0497227	112.7334729
17.	TPS Dr. Surata	-7.0289564	112.7536042
18.	TPS Musholla Sak-Sak Timur	-7.028981	112.751016
19.	TPS Jl. Kartini	-7.029668	112.750313
20.	TPS SMA 3	-7.0484579	112.7304228
21.	TPS Pengadilan	-7.0457495	112.7364912
22.	TPS Pumara	-7.0419622	112.7389796
23.	TPS Stadion	-7.042861	112.7384475
24.	TPS Pertanian	-7.047231	112.7356852
25.	TPS Las Pak Ujang	-7.038561	112.740236
26.	TPS RM Balado	-7.0362812	112.742544
27.	TPS Kel. Demangan	-7.034473	112.742187
28.	TPS Jl. KHM. Kholil Gg. VII	-7.030417	112.741165
29.	TPS PONPES KH. Kholil	-7.030486	112.742662

30.	TPS SMPN 1	-7.0256039	112.7516524
31.	TPS Toko Bok-Bok	-7.028936	112.753943
32.	TPS SDN Kraton 1	-7.0272079	112.7506507
33.	TPS PLN	-7.0265472	112.7505692
34.	TPS Jl. Mayjen Sungkono	-7.030580	112.752154
35.	TPS SMAN 1	-7.0297794	112.7587738
36.	SMA 4	-7.0213075	112.764092
37.	TPS RE. Martadinata	-7.046374	112.728703
38.	PONPES Al-Kholiliyah	-7.028921	112.754748
39.	Titik Embun	-7.036453	112.738668

LAMPIRAN B: Rute Armada Truk Pengangkut Sampah dengan Sistem Stationary Container System (SCS)

Rute armada truk pengangkut sampah dengan *Stationary Container System* (SCS) di Kabupaten Bangkalan berdasarkan data dari DLH Bangkalan yang diperoleh pada tahun 2019.

No.	Waktu	Pengemudi	Rute
1.	Pagi	Djumaidi	TPS Masjid Junok
2.			TPS Perum Halim 1
3.			TPS Perum Halim 2
4.			TPS Perum Griya Utama
5.			TPS Radio Amana FM
6.			TPS Dishutbun
7.			TPS Perum Cendana
8.			Imam Subakti
9.		TPS IKIP	
10.		TPS Coca Cola	
11.		TPS Dr. Surata	
12.		TPS Musholla Sak-Sak Timur	
13.		TPS Jl. Kartini	
14.		TPS SMA 3	
15.		Abdul Rahman	TPS Pengadilan
16.			Depo Senenan
17.			TPS Pumara
18.			TPS Stadion
19.			TPS Pertanian
20.			Kantor BRI
21.			Kantor Pajak
22.		Edy Hartono	Lampu Merah Senenan
23.			TPS Las Pak Ujang
24.			Titik Embun
25.			TPS RM Balado
26.			TPS Kel. Demangan
27.			TPS Jl. KHM. Kholil Gg. VII
28.			TPS PONPES KH. Kholil
29.		Samsul Hidayat	TPS SMPN 1
30.			TPS Toko Bok-Bok
31.			TPS SDN Kraton 1

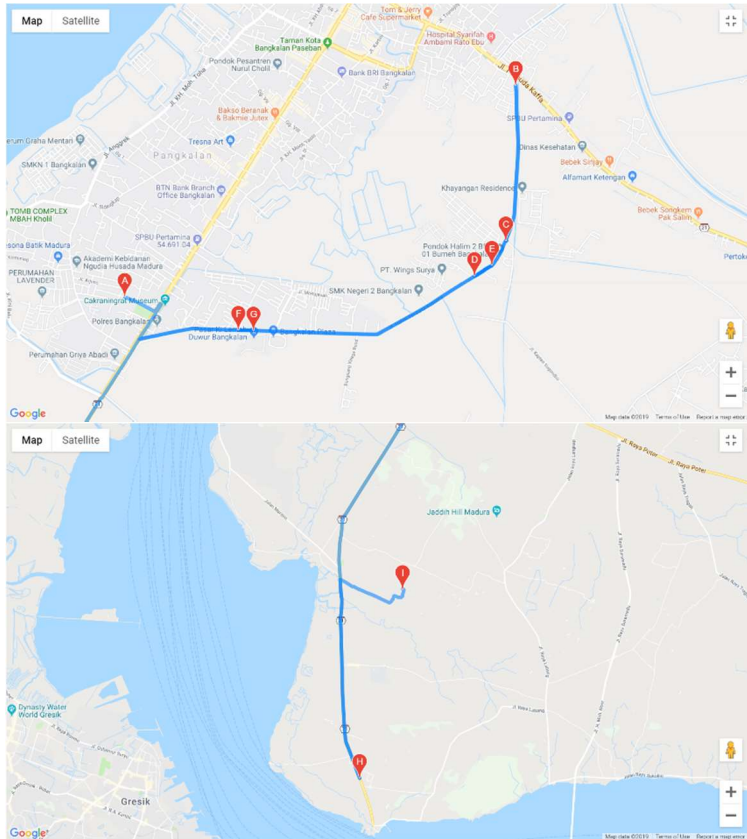
32.			TPS PLN
33.			Depo Jl. Mayjen Sungkono
34.			TPS Jl. Mayjen Sungkono
35.	Siang	Imam Subakti	Pom Bensin Junok
36.			TPS SMAN 1
37.			SMA 4
38.			TPS RE. Martadinata
39.			PONPES Al-Kholiliyah

LAMPIRAN C: PERBANDINGAN VISUALISASI RUTE

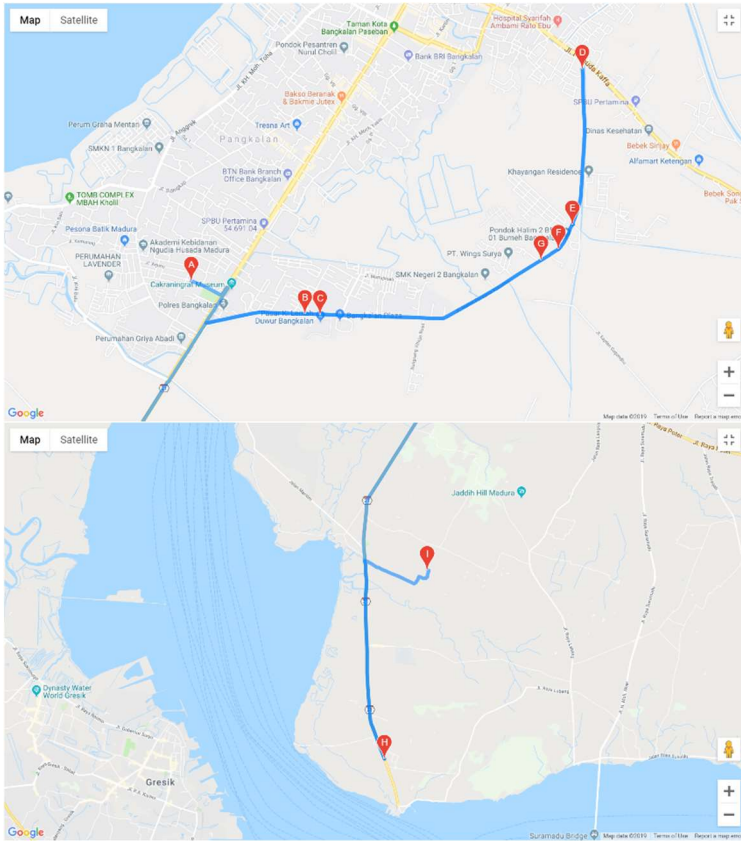
Visualisasi rute didapatkan dari hasil optimasi rute yang dilakukan pada aplikasi tugas akhir.

1. Perbandingan Rute Armada ke-1

a. Rute Lama

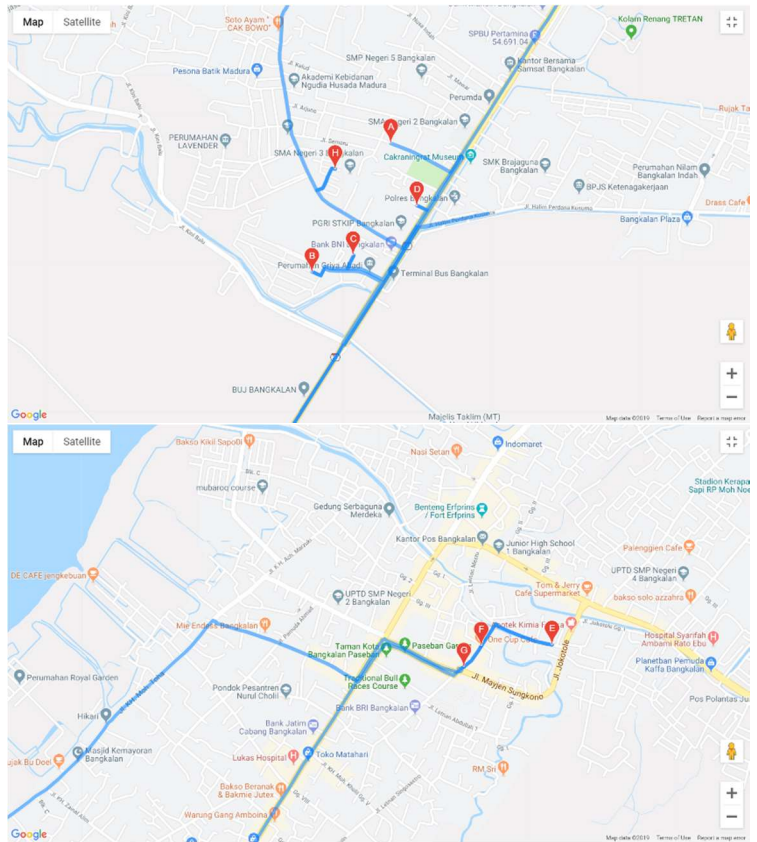


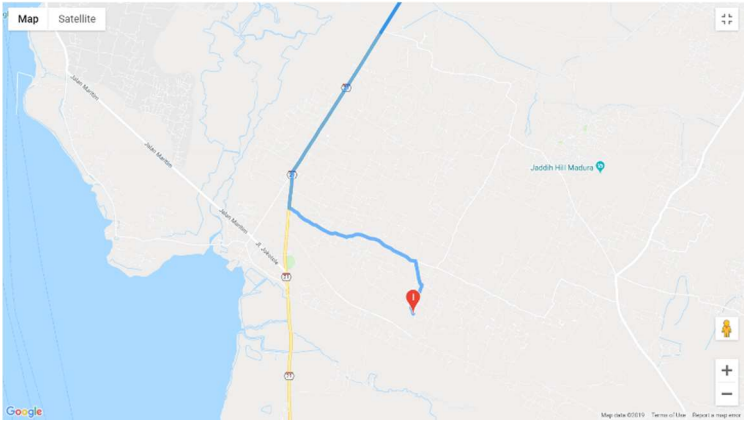
b. Rute Baru



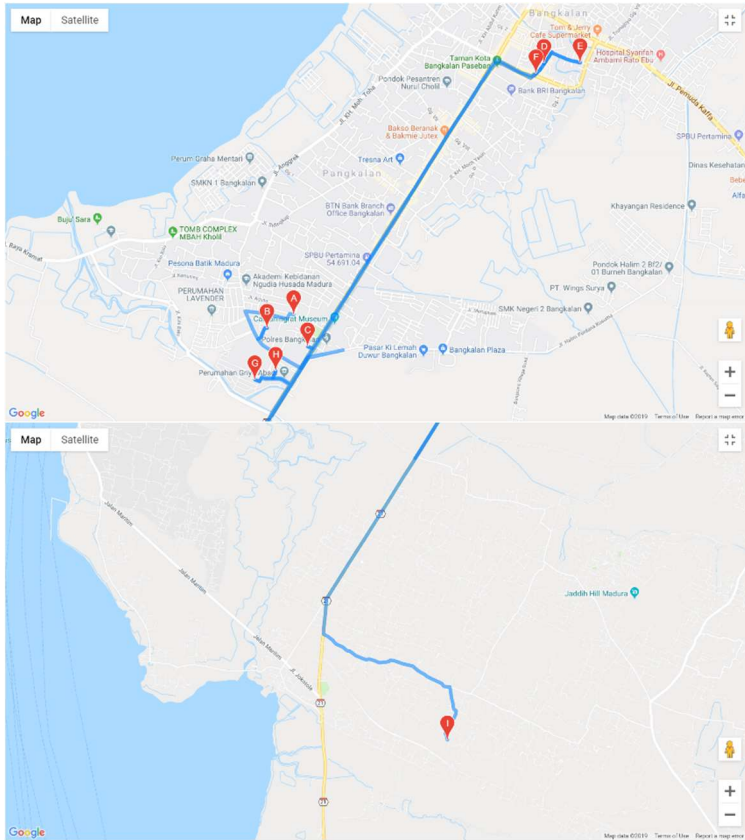
2. Perbandingan Rute Armada ke-2

a. Rute Lama



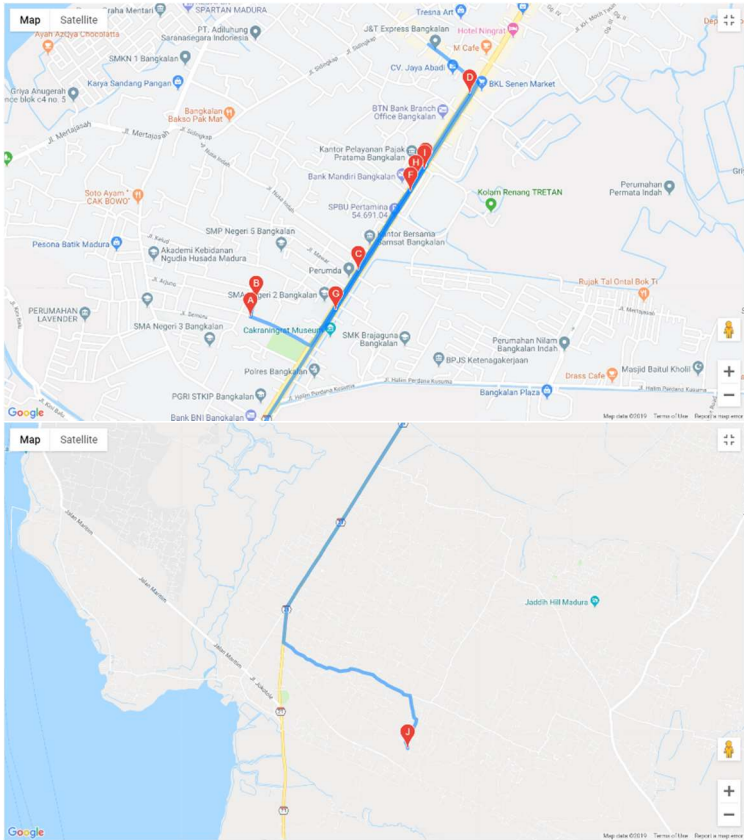


b. Rute Baru

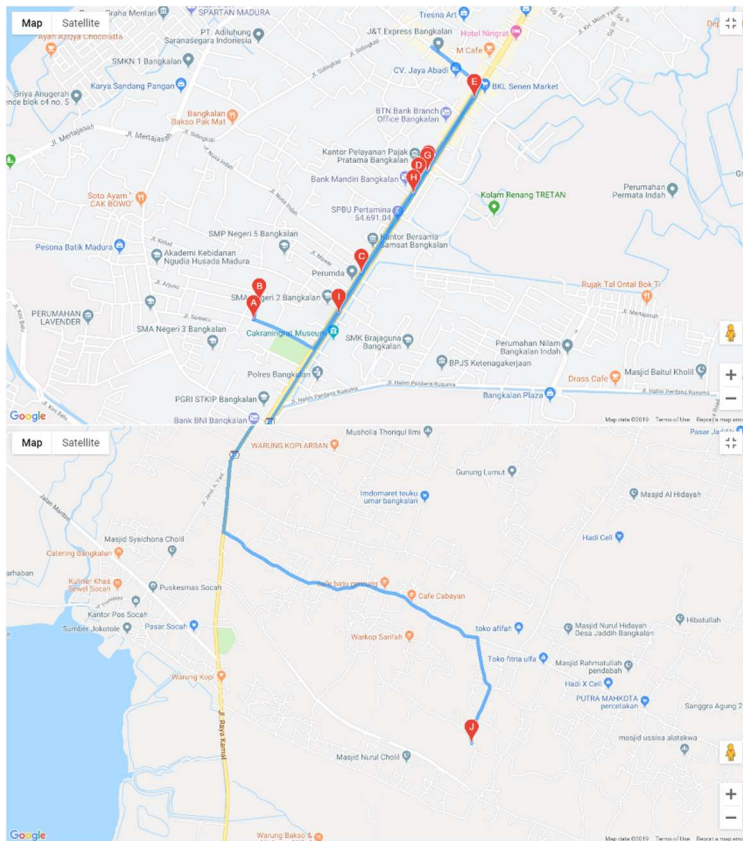


3. Perbandingan Rute Armada ke-3

a. Rute Lama

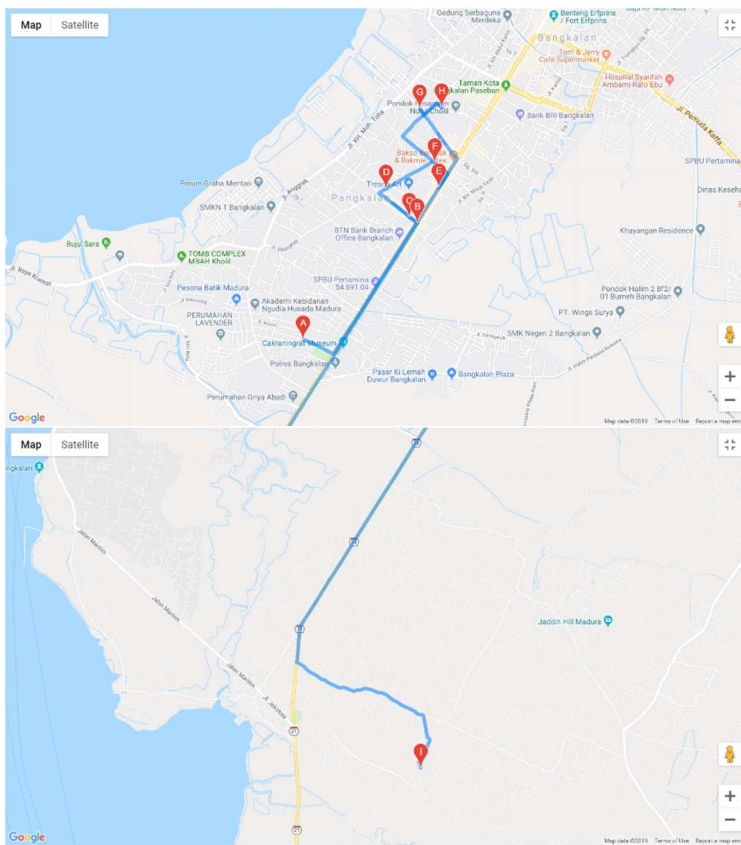


b. Rute Baru

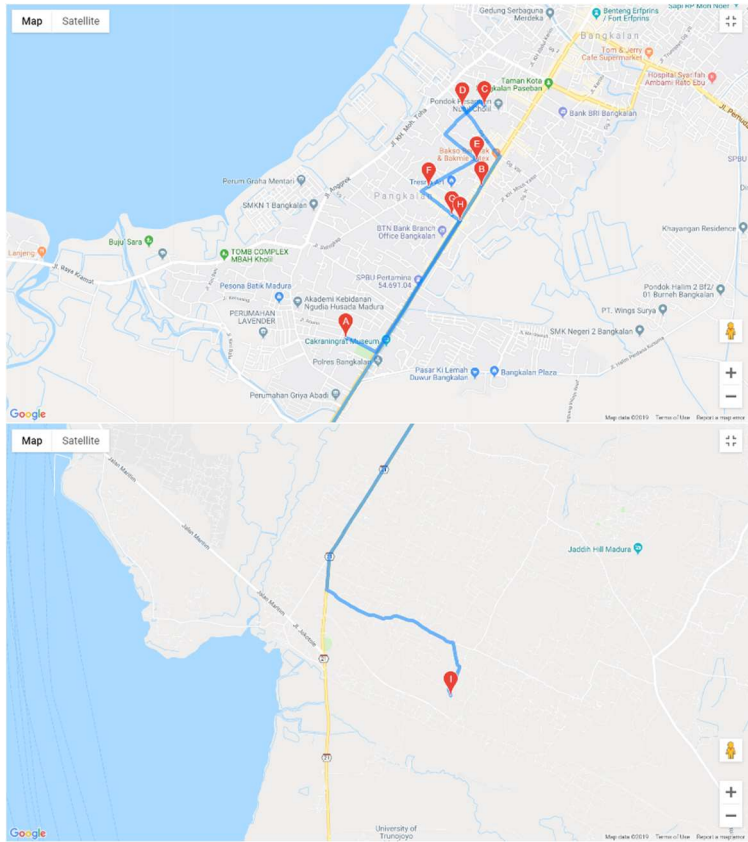


4. Perbandingan Rute Armada ke-4

a. Rute Lama

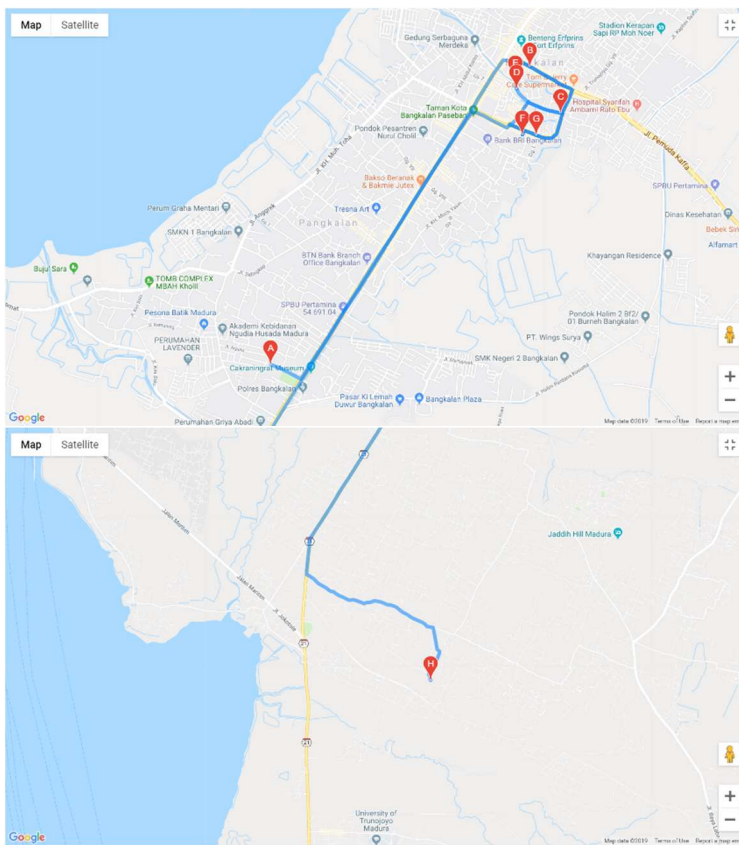


b. Rute Baru

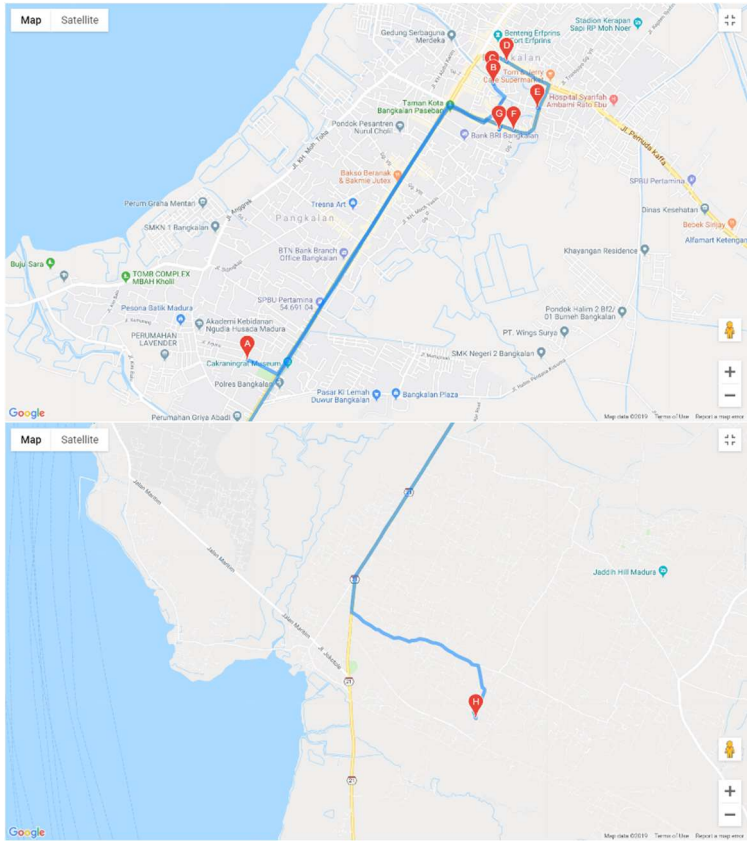


5. Perbandingan Rute Armada ke-5

a. Rute Lama

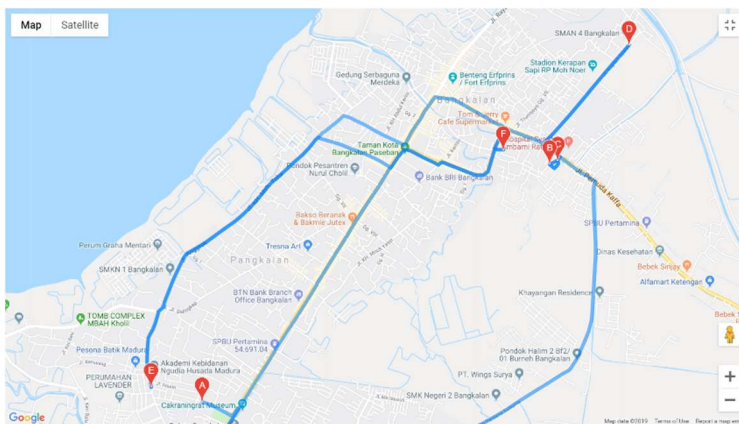


b. Rute Baru



6. Perbandingan Rute Armada ke-6

a. Rute Lama



b. Rute Baru

