

**PENERAPAN ALGORITMA NEAREST NEIGHBOR UNTUK OPTIMASI RUTE  
PENGAMBILAN DAN PEMBUANGAN SAMPAH DI KOTA YOGYAKARTA  
BERBASIS WEB**

**(Studi Kasus :Dinas Lingkungan Hidup Kota Yogyakarta)**

**TUGAS AKHIR**

Sebagai syarat untuk memperoleh gelar sarjana S-1 di Program Studi Informatika,  
Jurusan Teknik Informatika, Fakultas Teknik Industri, Universitas Pembangunan  
Nasional “Veteran” Yogyakarta



**Disusun Oleh:**

Fadel Al Cholid  
123140079

**PROGRAM STUDI INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2021**

**HALAMAN PENGESAHAN PEMBIMBING**

**PENERAPAN ALGORITMA NEAREST NEIGHBOR UNTUK OPTIMASI RUTE  
PENGAMBILAN DAN PEMBUANGAN SAMPAH DI KOTA YOGYAKARTA  
BERBASIS WEB**

**(Studi Kasus :Dinas Lingkungan Hidup Kota Yogyakarta)**

Disusun Oleh:

**FADEL AL CHOLID**

123140079

Telah di uji dan dinyatakan lulus oleh pembimbing  
pada tanggal: 26 November 2021

Menyetujui,  
Pembimbing I

Pembimbing II

**Nur Heri Cahyana, S.T., M.Kom.**  
NIP. 1960 09 22 1984 03 1001

**Frans Richard Kodong, S.T., M.Kom.**  
NIK. 2 6202 95 0006 1

Mengetahui,  
Koordinator Program Studi

**Dr. Heriyanto, A.Md., S.Kom.,M.Cs.**  
NIK. 2 7706 11 0301 1

## HALAMAN PENGESAHAN PENGUJI

### PENERAPAN ALGORITMA NEAREST NEIGHBOR UNTUK OPTIMASI RUTE PENGAMBILAN DAN PEMBUANGAN SAMPAH DI KOTA YOGYAKARTA BERBASIS WEB

(Studi Kasus: Dinas Lingkungan Hidup Kota Yogyakarta)

**Disusun Oleh:**

Fadel Al Cholid  
123140079

Telah diuji dan dinyatakan lulus oleh pembimbing dan penguji

Pada Tanggal: 26 November 2021

Menyetujui,  
Pembimbing I



**Nur Heri Cahyana, S.T., M.Kom**  
NIP. 1960 09 22 1984 03 1001

Pembimbing II



**Frans Richard Kodong, S.T., M.Kom.**  
NIK. 2 6202 95 0006 1

Menyetujui,  
Penguji I



**Dr. Heriyanto, A.Md., S.Kom., M.Cs.**  
NIK. 2 7706 11 0301 1

Penguji II



**Rifki Indra Perwira, S.Kom., M.Eng.**  
NIK. 2 8307 12 04181

### SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan dibawah ini, saya:

Nama : Fadel Al Cholid

NIM : 123140079

Menyatakan bahwa karya ilmiah saya yang berjudul **Penerapan Algoritma Nearest Neighbor Untuk Optimasi Rute Pengambilan Dan Pembuangan Sampah Di Kota Yogyakarta Berbasis Web.**

Merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya. Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta

Pada tanggal 29 November 2021

Menyatakan



(Fadel Al Cholid)

NIM. 123140079



## PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Fadel Al Cholid

NIM : 123140079

Fakultas/Prodi : Teknik Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir:

**Penerapan Algoritma Nearest Neighbor Untuk Optimasi Rute Pengambilan Dan Pembuangan Sampah Di Kota Yogyakarta Berbasis Web.**

Adalah hasil kerja saya sendiri dan benar bebas dari plagiasi kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (Sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta,

nembuat pernyataan

  
(Fadel Al Cholid)  
NIM. 123140079

## HALAMAN PERSEMBAHAN

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Penerapan Algoritma *Nearest Neighbor* Untuk Optimasi Rute Pengambilan Dan Pembuangan Sampah Di Kota Yogyakarta Berbasis Web”.

Dengan ini saya persembahkan karya ini untuk kedua orang tua saya dan saudara-saudara saya. Untuk orang tua terima kasih atas kasih sayang yang tak ternilai harganya dari saya lahir hingga saya sebesar ini. Terima kasih juga untuk doa-doa yang sudah dikirimkan kepada saya dan hal-hal baik yang telah diberikan.

## ABSTRAK

Membangun sebuah perangkat lunak optimasi rute pengambilan sampah di kota Yogyakarta untuk membantu Dinas Lingkungan Hidup Kota Yogyakarta dalam rute pengambilan sampah dan pembuangan sampah menggunakan perhitungan waktu berdasarkan volume bak truk dan volume tempat sampah.

Perancangan/metode/pendekatan: Menggunakan metode *Nearest Neighbor* untuk menentukan jalur pengambilan dan pembuangan sampah tercepat menuju TPS dan TPA. Hasil: Dari Membandingkan waktu titik TPS awal dengan titik TPS lainnya, maka akan mendapatkan waktu tercepat, lalu membandingkan waktu titik TPA maka akan ditemukan jalur pengambilan tercepat yang akan dituju oleh *driver* dan rute akan dimunculkan pada halaman jalur pickup.

Keaslian/ *state of the art*: Perbedaan penelitian ini dengan penelitian sebelumnya adalah pada bagian metode penentuan jalur dan objek penelitian. Pada metode ini data titik TPS dan titik TPA akan dianalisa lalu dilakukan proses import ke *database*. Kemudian hasil analisa data titik TPS awal dan titik TPA akan diproses menggunakan metode *Nearest Neighbor*. *Output* dari sistem ini berupa rute tercepat yang diambil oleh setiap *driver* dalam menjalankan tugas pengambilan dan pembuangan sampah.

**Kata kunci:** Jalur Pengambilan Dan Pembuangan sampah, *Google Maps*, *Nearest Neighbor*, Sampah.

## ABSTRACT

*Objective: To build a software optimization of garbage collection routes in the city of Yogyakarta to assist the Yogyakarta City Environmental Service in the route of garbage collection and disposal of waste using time calculations based on the volume of the tailgate and the volume of the trash bin.*

*Design/method/approach: Using method Nearest Neighbor to determine the fastest garbage collection and disposal path to the TPS and TPA.*

*Result: Comparing the time of the initial tps point with other tps points, it will get the fastest time, then comparing the time of the landfill point, the fastest pick-up path will be found by the driver and the route will appear on the pickup line page.*

*Authenticity/ state of the art: The difference between this study and previous research is in the method of determining the path and object of research. In this method, the TPS and TPA point data will be analyzed and then imported into the database. Then the results of the initial TPS point data analysis and the TPA point will be processed using the method Nearest Neighbor. The output of this system is the fastest route taken by each driver in carrying out their garbage collection and disposal tasks.*

***Keywords: Intake and Disposal Path, Google Maps, Nearest Neighbors, Garbage.***



## PRAKATA

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Penerapan Algoritma *Nearest Neighbor* Untuk Optimasi Rute Pengambilan Dan Pembuangan Sampah Di Kota Yogyakarta Berbasis Web (Studi Kasus: Dinas Lingkungan Hidup Kota Yogyakarta)”.

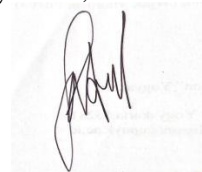
Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana strata satu pada Jurusan Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Laporan ini tidak dapat terselesaikan tanpa bantuan dari berbagai pihak, baik secara langsung maupun tidak langsung. Pada kesempatan ini penulis ingin menyampaikan ungkapan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang senantiasa memberikan petunjuk dan kemudahan.
2. Kedua orang tua saya tercinta, Bapak Edi Driyan Tiktono, Ibu Mulyanah, serta nenek saya Mbah Witri atas kasih sayang tak terhingga sepanjang masa, nasihat, doa serta dukungan baik secara moral maupun material yang telah diberikan.
3. Saudara saya Ghina Al Ghonia, Mbak Friska, Om Yanto, Bulek Ari, Om Feri yang telah memberikan semangat, Motivasi dan Doa untuk Menyelesaikan Tugas Akhir ini.
4. Teman baik saya Widi Kristanti, Rofiudin, Della, Aini Marwah, Imam Abdullah, Rifky, Luqmen, Ahzam, Warsana, Ageng, Candra Kharisma, Ergan Pramudita, Fajar, Iqbal, Raka Merdeka, Rezki Suprastio, Donny, Rama, Yudhis yang telah memberikan semangat dan motivasi serta membantu dalam penyelesaian Tugas Akhir ini.
5. Bapak Nur Heri Cahyana, S.T., M.Kom selaku dosen pembimbing I yang telah membimbing penulis dalam mengerjakan tugas akhir ini.
6. Bapak Frans Richard Kodong, S.T., M.Kom selaku dosen pembimbing II yang telah membimbing penulis dalam mengerjakan Tugas Akhir ini.
7. Bapak Dr. Heriyanto, A.Md., S.Kom., M.Cs dan Bapak Rifki Indra Perwira, S.Kom., M.Eng selaku penguji yang telah memberikan kritik dan saran dalam Tugas Akhir ini, sehingga Tugas Akhir ini menjadi lebih baik.
8. Dosen dan Mentor saya yang telah melatih, mendidik dan mengarahkan hingga menjadi manusia yang berilmu dan berakhlak.
9. Keluarga Besar Teknik Informatika UPN Veteran Yogyakarta, terutama Angkatan 2014 yang selalu menemani dalam proses perjuangan dari awal sampai ke titik ini. Mungkin disini saya tidak bisa menyebutkan satu persatu nama kalian, yang jelasnya setiap nama kalian ada cerita yang sangat luar biasa. Terimakasih Teknik Informatika 2014.

Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari sempurna baik dari segi penyusunan, bahasa, maupun dari penulisan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pembaca guna menjadi acuan agar penulis bisa menjadi lebih baik lagi di masa mendatang.

Yogyakarta, 15 November 2021



Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN PEMBIMBING .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN PENGUJI.....</b>	<b>iii</b>
<b>SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR .....</b>	<b>iv</b>
<b>PERNYATAAN BEBAS PLAGIASI.....</b>	<b>v</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vi</b>
<b>ABSTRAK.....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>PRAKATA .....</b>	<b>ix</b>
<b>DAFTAR ISI .....</b>	<b>x</b>
<b>DAFTAR GAMBAR .....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>DAFTAR MODUL PROGRAM .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
1.6 Metodologi Penelitian Dan Pengembangan Sistem.....	4
1.4.1 Metodologi Penelitian .....	4
1.4.2 Metode Pengembangan Sistem.....	4
1.7 Sistematika Penulisan .....	5
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1 Pengolahan Sampah.....	7
2.2 Nearest Neighbor .....	7
2.3 PHP .....	7
2.4 Metode Pengembangan Sistem.....	9
2.5 Data Flow Diagram (DFD).....	10
2.6 Flowchart .....	11
2.7 Database.....	12
2.8 Entity Relationship Diagram (ERD).....	12
2.9 Tinjauan Pustaka.....	14
<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>16</b>
3.1 Metode Penelitian .....	16
3.1.1 Data Wawancara.....	16
3.2 Metode Pengembangan Sistem.....	16
3.3 Survei Data .....	18
3.1.2 Studi Pustaka .....	18
3.1.3 Observasi.....	18
3.4 ANALISIS KEBUTUHAN SISTEM.....	19
3.1.4 Kebutuhan Perangkat Lunak (Software) .....	19

3.1.5 Kebutuhan Perangkat Keras ( <i>Hardware</i> ) .....	19
3.5 Perancangan Sistem .....	19
3.6 Arsitektur Sistem .....	20
3.7 Perancangan Proses .....	21
3.1.6 Data Flow Diagram (DFD) Level 0 .....	21
3.1.7 Data Flow Diagram (DFD) Level 1 .....	22
3.8 Flowchart .....	24
3.9 Perancangan Basis Data .....	26
3.1.8 Entity Relationship Diagram (ERD) .....	26
3.1.9 Perancangan Struktur Tabel .....	27
3.1.10 Relasi antar tabel (RAT) .....	30
3.10 Sample Perhitungan Metode .....	31
3.11 Perancangan Antarmuka .....	34
3.1.11 Struktur Menu .....	34
3.1.12 Perancangan User Interface .....	36
3.12 Perancangan Pengajuan Sistem .....	49
<b>BAB IV HASIL, PENGUJIAN, DAN PEMBAHASAN .....</b>	<b>52</b>
4.1 Hasil Penelitian .....	52
4.1.1 Tampilan Halaman Login .....	52
4.1.2 Tampilan Halaman Jalur Pickup .....	53
4.1.3 Tampilan Halaman Detail Perhitungan .....	55
4.1.4 Tampilan Halaman Sektor .....	58
4.1.5 Tampilan Halaman TPS .....	60
4.1.6 Tampilan Halaman Rute .....	61
4.1.7 Tampilan Halaman Kendaraan .....	62
4.1.8 Tampilan Halaman Kendaraan TPA .....	64
4.1.9 Tampilan Halaman TPA .....	66
4.1.10 Tampilan Halaman User .....	68
4.2 Pengujian Sistem .....	69
4.2.1 Alpha Test .....	70
4.2.2 Pengujian Beta Test .....	71
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>74</b>
5.1 Kesimpulan .....	74
5.2 Saran .....	74
<b>DAFTAR PUSTAKA .....</b>	<b>75</b>

## DAFTAR GAMBAR

Gambar 2. 1 Algoritma <i>Nearest Neighbor</i> titik awal ke titik pengambilan .....	8
Gambar 2. 2 Algoritma <i>Nearest Neighbor</i> terpilih pertama ke titik pengambilan berikutnya .....	8
Gambar 2. 3 Algoritma <i>Nearest Neighbor</i> urutan pengambilan dari titik awal ke semua rute .....	8
Gambar 2. 4 Tahapan Metode Waterfall .....	9
Gambar 3. 1 Tahapan metode <i>waterfall</i> .....	18
Gambar 3. 2 Arsitektur Sistem .....	20
Gambar 3. 3 Data Flow Diagram (DFD) Level 0 .....	22
Gambar 3. 4 Data Flow Diagram (DFD) Level 1 .....	23
Gambar 3. 5 Flowchart Sistem .....	25
Gambar 3. 6 Entity Relationship Diagram .....	26
Gambar 3. 7 Relasi Antar Tabel (RAT) .....	30
Gambar 3. 8 Struktur Menu Admin .....	35
Gambar 3. 9 Struktur Menu Driver .....	35
Gambar 3. 10 Login Form .....	36
Gambar 3. 11 Halaman Utama User .....	37
Gambar 3. 12 Halaman Jalur Pickup .....	37
Gambar 3. 13 Halaman Sektor .....	38
Gambar 3. 14 Halaman Tabel TPS .....	38
Gambar 3. 15 Halaman Rute .....	39
Gambar 3. 16 Halaman Add Sektor .....	39
Gambar 3. 17 Halaman Add TPS .....	40
Gambar 3. 18 Halaman Edit TPS .....	40
Gambar 3. 19 Halaman Edit Rute .....	41
Gambar 3. 20 Halaman Kendaraan .....	41
Gambar 3. 21 Halaman Add kendaraan .....	42
Gambar 3. 22 Halaman Edit Kendaraan .....	43
Gambar 3. 23 Halaman Kendaraan TPA .....	43
Gambar 3. 24 Halaman Add Kendaraan TPA .....	44
Gambar 3. 25 Halaman Edit Kendaraan TPA .....	44
Gambar 3. 26 Halaman TPA .....	45
Gambar 3. 27 Halaman Add TPA .....	45
Gambar 3. 28 Halaman Edit TPA .....	46
Gambar 3. 29 Halaman User .....	46
Gambar 3. 30 Halaman Add User .....	47
Gambar 3. 31 Halaman Edit User .....	47
Gambar 3. 32 Halaman Detail Perhitungan .....	48
Gambar 4. 1 Tampilan Halaman Login .....	52
Gambar 4. 2 Tampilan Halaman Jalur Pickup .....	53
Gambar 4. 3 Tampilan Halaman Detail Perhitungan .....	55
Gambar 4. 4 Tampilan Halaman Sektor .....	58
Gambar 4. 5 Tampilan Halaman TPS .....	60
Gambar 4. 6 Tampilan Halaman Rute .....	61
Gambar 4. 7 Tampilan Halaman Kendaraan .....	63
Gambar 4. 8 Tampilan Halaman Kendaraan TPA .....	64
Gambar 4. 9 Tampilan Halaman TPA .....	66
Gambar 4. 10 Tampilan Halaman User .....	68

## DAFTAR TABEL

Tabel 2. 1 Simbol dalam DFD .....	11
Tabel 2. 2 Simbol dalam Flowchart .....	11
Tabel 2. 3 Simbol dalam Flowchart (Lanjutan) .....	12
Tabel 2. 4 Simbol dalam ERD .....	14
Tabel 2. 5 Penelitian sebelumnya.....	15
Tabel 3. 1 Pembagian Hak Akses Pengguna.....	21
Tabel 3. 2 Tabel Data Pengguna .....	27
Tabel 3. 3 Tabel Kendaraan .....	27
Tabel 3. 4 Tabel Kendaraan TPA.....	27
Tabel 3. 5 Tabel Rute dari TPA .....	27
Tabel 3. 6 Tabel Rute ke TPA.....	28
Tabel 3. 7 Tabel Rute TPS .....	29
Tabel 3. 8 Tabel Sektor .....	29
Tabel 3. 9 Tabel TPA .....	29
Tabel 3. 10 Tabel TPS.....	29
Tabel 3. 11 Data Titik Awal Keberangkatan TPA .....	31
Tabel 3. 12 Data Kendaraan .....	31
Tabel 3. 13 Perbandingan waktu tercepat dari titik keberangkatan.....	31
Tabel 3. 14 Hasil Data TPS Yang Terdekat Dari Titik Keberangkatan .....	31
Tabel 3. 15 Perhitungan waktu dari titik awal menuju Jalan Pramuka .....	32
Tabel 3. 16 Perhitungan waktu dari Jalan Pramuka menuju Jembatan Tungkak .....	32
Tabel 3. 17 Perhitungan waktu dari Jembatan Tungkak menuju Jl Singasingamangaraja .....	32
Tabel 3. 18 Perhitungan waktu dari Jalan Singasingamangaraja menuju Jalan Veteran .....	32
Tabel 3. 19 Waktu Tempuh Dari TPS Titik Akhir Menuju ke TPA .....	32
Tabel 3. 20 Perhitungan waktu dari TPA Piyungan menuju Rs Pratama.....	33
Tabel 3. 21 Perhitungan waktu dari Rs Pratama menuju Pasar Pujokusuman .....	33
Tabel 3. 22 Perhitungan waktu dari Pasar Pujokusuman menuju Lp Wirogunan .....	33
Tabel 3. 23 Perhitungan waktu dari Lp Wirogunan menuju Jalan Gajah .....	33
Tabel 3. 24 Perhitungan waktu dari TPS Titik Akhir Menuju TPA.....	33
Tabel 3. 25 Hasil Perhitungan Akhir dari TPS ke TPS .....	34
Tabel 3. 26 Tabel Rancangan <i>Alpha Test</i> Pada <i>Admin</i> .....	50
Tabel 3. 27 Bobot Penilaian Pengujian Alpha .....	50
Tabel 3. 28 Pengujian <i>Beta Test</i> .....	51
Tabel 3. 29 Bobot Penilaian Pengujian Beta.....	51
Tabel 4. 1 Bobot Penilaian Pengujian <i>Alpha Test</i> .....	70
Tabel 4. 2 Jumlah Pengujian <i>Alpha Test</i> .....	70
Tabel 4. 3 Skala Penilaian Pengujian Beta Test.....	72
Tabel 4. 4 Jumlah Pengujian Beta Test .....	72

## DAFTAR MODUL PROGRAM

Modul Program 4. 1 <i>Source Code</i> Tampilan Halaman Login.....	53
Modul Program 4. 2 <i>Source Code</i> Tampilan Jalur Pickup.....	54
Modul Program 4. 3 Lanjutan <i>Source Code</i> Tampilan Jalur Pickup.....	55
Modul Program 4. 4 <i>Source code</i> Tampilan Halaman Detail Perhitungan .....	56
Modul Program 4. 5 Lanjutan <i>Source Code</i> Tampilan Halaman Detail Perhitungan .....	58
Modul Program 4. 6 <i>Source Code</i> Tampilan Halaman Sektor.....	59
Modul Program 4. 7 Lanjutan <i>Source Code</i> Tampilan Halaman Sektor.....	59
Modul Program 4. 8 <i>Source Code</i> Tampilan Halaman TPS .....	61
Modul Program 4. 9 <i>Source Code</i> Tampilan Halaman Rute.....	61
Modul Program 4. 10 Lanjutan <i>Source Code</i> Tampilan Halaman Rute.....	62
Modul Program 4. 11 <i>Source Code</i> Tampilan Halaman Kendaraan .....	64
Modul Program 4. 12 <i>Source Code</i> Tampilan Halaman Kendaraan Tpa.....	66
Modul Program 4. 13 <i>Source Code</i> Tampilan Halaman TPA.....	67
Modul Program 4. 14 <i>Source Code</i> Tampilan Halaman User.....	69



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Daerah Istimewa Yogyakarta (DIY) merupakan kota wisata dan pendidikan yang padat penduduk dengan jumlah pertumbuhan dan perkembangan yang baik dari segi ekonomi dan segi pembangunan, tentunya tingkat limbah sampah juga ikut meningkat. Dari data Dinas Lingkungan Hidup kota Yogyakarta mengatakan bahwa pada tahun 2018 jumlah sampah yang ditangani mencapai 549,740 ton pertahun, kemudian pada tahun 2019 mengalami peningkatan dengan jumlah sampah yang ditangani mencapai 583,800 ton pertahun. Untuk jumlah TPS di kota Yogyakarta berjumlah 68 TPS dan kendaraan yang dipunya Dinas Lingkungan Hidup kota Yogyakarta berjumlah 142 kendaraan.

Sampah pada dasarnya merupakan suatu bahan yang terbuang atau dibuang dari sumber hasil aktivitas manusia maupun proses-proses alam yang tidak mempunyai nilai ekonomi, bahkan dapat mempunyai nilai negatif karena dalam penanganannya, baik untuk membuang atau membersihkannya memerlukan biaya yang cukup besar. Selain itu karakteristik dari sampah adalah bau, sampah juga bisa menimbulkan penyakit diare. Sampah merupakan konsekuensi dari adanya aktivitas masyarakat. Setiap aktivitas manusia pasti menghasilkan buangan atau sampah. Seiring tumbuhnya kota, bertambah pula beban yang harus diterima kota tersebut. Sampah akan memberikan efek negatif apabila penanganannya tidak dilakukan secara cermat dan serius, yaitu mengakibatkan terjadinya perubahan keseimbangan lingkungan yang merugikan atau tidak diharapkan, sehingga dapat mencemari lingkungan baik terhadap tanah, air, maupun udara. Pengelolaan sampah secara efektif harus dijalankan oleh semua pihak, baik masyarakat maupun pemerintah. Semua pihak bertanggung jawab terhadap penanganan sampah sehingga tidak lagi menimbulkan masalah.

Dinas Lingkungan Hidup di kota Yogyakarta khususnya bidang pengelolaan persampahan mempunyai tugas membantu kepala dinas dalam merumuskan kebijakan, mengkoordinasikan, membina, mengawasi, dan mengendalikan program dibidang pengelolaan persampahan. Saat ini pengambilan sampah dari TPS ke TPA yang dilakukan Dinas Lingkungan Hidup dalam pemilihan jalur pengambilan sampah masih bersifat memperkirakan (*intuisi*). Optimasi rute merupakan salah satu faktor penting dalam melakukan pengambilan dan pembuangan sampah. Karena pengambilan sampah yang diperkirakan belum terstruktur, maka membutuhkan waktu yang kurang efektif.

Beberapa penelitian yang terkait dengan optimasi rute diantaranya, penelitian yang dilakukan oleh (Andayani & Perwitasari, 2014) dengan judul “Optimasi Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma *Dijkstra*”. Pada penelitian ini metode yang digunakan adalah metode *Dijkstra*, dalam penggunaannya metode ini menggunakan parameter jarak, waktu dan beban biaya yang ditanggung oleh kantor dinas. Sedangkan pada penelitian (Abadi et al., 2014) yang berjudul “Optimasi Rute Kendaraan Distribusi Produk Roti Menggunakan Metode *Nearest Neighbor* dan Metode *Sequential Insertion*”. Pada penelitian ini dilakukan peluang penentuan rute kendaraan distribusi produk

roti dari titik terdekat lebih besar. Sedangkan pada penelitian oleh (Kasus et al., 2014) dengan judul “Optimasi Rute Distribusi Es Balok Menggunakan *Algoritma Nearest Neighbor* dan *Local Search*”. Pada penelitian ini dilakukan penentuan rute terpendek dan tercepat karena dalam penelitian ini hanya menentukan es balok mencair dalam hitungan jam. Sedangkan penelitian yang dilakukan oleh (Arinalhaq et al., 2013) dengan judul “Optimasi Rute Kendaraan Pengangkutan Sampah Dengan Menggunakan Metode *Nearest Neighbor*”. Pada penelitian ini dilakukan menentukan rute kendaraan pengangkut sampah serta belum diterapkannya penelitian ini pada sistem operasi berbasis android maupun berbasis web serta dibutuhkan penelitian lanjutan terkait penelitian ini dengan menimbang jarak rute. Sedangkan penelitian yang dilakukan oleh (Pratama Dicky., 2019) dengan judul “Optimasi Travelling Salesman Problem With Time Windows (TSP-TW) Pada Rute Obyek Wisata Di Provinsi Yogyakarta Menggunakan Metode *Algoritma Genetika* Berbasis Android”. Pada penelitian ini menentukan rute obyek wisata di Yogyakarta serta waktu tempuh, jadwal buka tempat wisata, dan jadwal tutup tempat wisata.

Dengan permasalahan tersebut maka dilakukan penelitian dengan membuat aplikasi optimasi rute pengambilan dan pembuangan sampah berbasis web. Algoritma yang digunakan untuk menyelesaikan optimasi rute pada penelitian ini adalah *Nearest Neighbor*. *Nearest Neighbor* adalah sebuah teknik dalam menyelesaikan permasalahan rute dengan cara menentukan jarak tercepat dari titik awal penjemputan. Metode ini merupakan metode yang sederhana dalam memecahkan masalah rute dan merupakan solusi awal dalam pemilihan titik tercepat berdasarkan asumsi jumlah kendaraan dalam setiap evakuasi. Kelebihan menggunakan algoritma *Nearest Neighbor* yaitu karena semua solusi titik yang diperoleh lebih efisien dalam menentukan titik TPS pertama yang harus dituju oleh *driver* berdasarkan perhitungan waktu.

Berdasarkan permasalahan tersebut, penelitian ini akan merancang suatu aplikasi berbasis web dengan menggunakan algoritma *Nearest Neighbor*. Penelitian ini diterapkan untuk membantu Dinas Lingkungan Hidup Kota Yogyakarta dalam menentukan rute waktu tempuh yang efisien dalam pengambilan sampah yang akan dituju.

Dengan menggunakan aplikasi ini proses pengambilan sampah dari TPS ke TPA jadi efisien serta mengoptimalkan kinerja tenaga kerja.

Definisi efisien pada penelitian ini adalah proses yang mengharuskan penyelesaian pekerjaan dengan tepat waktu, cepat dan memuaskan. Sehingga efisien berkaitan erat dengan tepat waktu tanpa harus mengeluarkan biaya yang berlebihan.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah dikemukakan, maka dapat dirumuskan masalah yang ada sebagai berikut:

Bagaimana melakukan optimasi rute dengan menggunakan metode *Nearest Neighbor* agar waktu yang dihitung dapat menghasilkan waktu yang efisien.

## 1.3 Batasan Masalah

Untuk memperoleh pembahasan yang mengarah pada tujuan dan memudahkan proses pengolahan data, maka ditetapkan batasan–batasan masalah terhadap penelitian yang dilakukan. Berikut adalah batasan–batasan masalah dalam penelitian ini:

1. Penelitian ini hanya berlaku di kota Yogyakarta.
2. Variabel yang digunakan dalam penelitian ini adalah waktu tempuh, rute efisien, dan faktor kemacetan tidak diperhitungkan pada penelitian ini.
3. Aplikasi ini tidak memperhitungkan jarak, biaya dan jadwal pengambilan sampah.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun sebuah perangkat lunak optimasi rute pengambilan sampah di kota Yogyakarta dengan menggunakan metode *Nearest Neighbor* untuk membantu Dinas Lingkungan Hidup kota Yogyakarta dalam rute pengambilan sampah dan pembuangan sampah menggunakan perhitungan waktu berdasarkan volume bak truk dan volume tempat sampah.

### 1.5 Manfaat Penelitian

Aplikasi ini diharapkan dapat membantu Dinas Lingkungan Hidup dalam menentukan rute yang efisien untuk pengambilan dan pembuangan sampah, serta diharapkan dapat membantu kota Yogyakarta agar terbebas dari limbah sampah yang menumpuk.

### 1.6 Metodologi Penelitian Dan Pengembangan Sistem

Metodologi penelitian dan pengembangan sistem merupakan urutan langkah-langkah sistematis dalam membangun sistem yang dibuat. Dalam metodologi penelitian ini terdapat dua jenis metode penelitian pengumpulan data dan pengembangan.

#### 1.4.1 Metodologi Penelitian

Metode penelitian pengumpulan data terdiri dari studi observasi, dan studi pustaka.

##### 1. Observasi

Observasi ini langsung dilakukan ke instansi yang bersangkutan, kebetulan dengan sistem yang akan dibangun ini bekerja sama dengan Dinas Lingkungan Hidup kota Yogyakarta. Observasi yang dilakukan di Dinas Lingkungan Hidup kota Yogyakarta adalah pengumpulan data yang berisi tentang informasi rute, jumlah armada, dan jumlah TPS yang tercatat.

##### 2. Wawancara

Wawancara ini merupakan proses pengumpulan data yang dilakukan dengan proses tanya jawab. Wawancara dilakukan dengan salah satu staff Dinas Lingkungan Hidup kota Yogyakarta di bidang Teknisi Lapangan Pengurangan Sampah. Wawancara dengan salah satu staff Dinas Lingkungan Hidup tersebut membicarakan mengenai alur proses rute pengambilan sampah, jumlah armada truck, serta jumlah titik TPS

#### 1.4.2 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan perangkat lunak (*software*) ini adalah metodologi air terjun (*Waterfall*). Metode *waterfall* adalah metode pengembangan *software* yang menyediakan pendekatan alur hidup perangkat lunak yang teratur serta mempunyai tahapan-tahapan yang jelas. Tahapan dari metode ini dimulai dari survei sistem, analisa sistem, desain sistem, pembuatan sistem, implementasi sistem, pemeliharaan sistem (Imam Rofi'i, Hendrawan, Pareza Alam Jusia, 2015).

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut.

Tahapan tahapan dari metode *waterfall* adalah sebagai berikut:

1. Requirements Analysis

Dalam tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. System and Software design

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementation

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

4. Integration and System Testing

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. Operation & maintenance

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

## 1.7 Sistematika Penulisan

Penelitian ini terdiri dari 5 (lima) bab yang disusun secara sistematis, sistematika penulisannya sebagai berikut :

## BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Dalam bab ini berisi tentang teori-teori dari beberapa penelitian yang sudah dilakukan dan menjadi landasan yang mendasari penelitian, teori tersebut meliputi kajian literatur memuat karya ilmiah yang sudah ada.

## **BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM**

Pada bab ini berisi mengenai metode penelitian, pengumpulan data dan analisis data. Analisis data dibuat dengan mengkombinasikan pengumpulan data yang diperoleh. Serta cara bagaimana aktifitas kerangka kerja dalam tiap tahapnya serta gambaran umum perancangan sistem, meliputi perancangan mekanisme sistem.

## **BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN**

Bab ini berisi secara rinci penerapan dari analisis data dan pengolahan data yang dibuat menjadi aplikasi penentuan jalur pengambilan dengan menampilkan antarmuka program disertai cara kerja dan penggunaan program.

## **BAB V PENUTUP**

Dalam bab ini berisi penutup dari penjelasan yang berisi tentang kesimpulan dan saran dari hasil aplikasi agar dapat dilakukan perbaikan dimasa yang akan datang.



## BAB II TINJAUAN PUSTAKA

### 2.1 Pengolahan Sampah

Pengelolaan sampah adalah pengumpulan, pengangkutan, pemrosesan, daur ulang, atau pembuangan dari material sampah. Material sampah yang dihasilkan biasanya dampak dari aktivitas manusia, dan biasanya dikelola untuk mengurangi dampaknya terhadap kesehatan, lingkungan, atau estetika. Pengelolaan sampah juga dilakukan untuk memulihkan sumber daya alam.

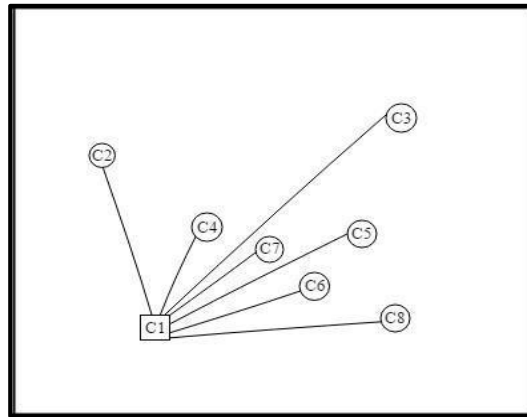
### 2.2 Nearest Neighbor

Metode *nearest neighbor* merupakan sebuah teknik dalam menyelesaikan permasalahan rute dengan cara menentukan jarak tercepat dari titik awal penjemputan. Metode ini merupakan metode yang sederhana dalam memecahkan masalah rute dan merupakan solusi awal dalam pemilihan titik terdekat berdasarkan asumsi jumlah kendaraan dalam setiap evakuasi (Abadi et al., 2014)).

Cara kerja metode ini adalah sebagai berikut. Pertama-tama, semua kendaraan masih kosong. Dimulai dari rute kendaraan pertama, metode ini memasukkan (*insert*) satu persatu titik terdekat (*Nearest Neighbor*) yang belum dikunjungi ke dalam rute, selama memasukkan titik tersebut ke dalam rute, kendaraan tidak melanggar batasan kapasitas maksimum kendaraan tersebut (Amri et al., 2014). Contoh kerja algoritma *Nearest Neighbor* dapat dilihat pada gambar 2.1, gambar 2.2, dan gambar 2.3.

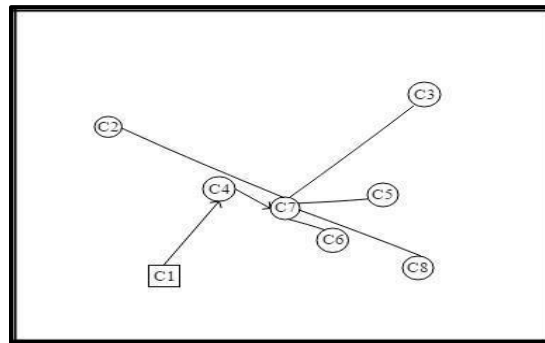
### 2.3 PHP

PHP merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*. PHP memungkinkan untuk membuat halaman *web* yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL namun PHP juga mendukung sistem manajemen database *Oracle*, *Microsoft Access*, *Interbase*, *d-base*, *PostgreSQL*, dan sebagainya. (Sukri Adrianto1, Yenus Novia Fitri, 2019)



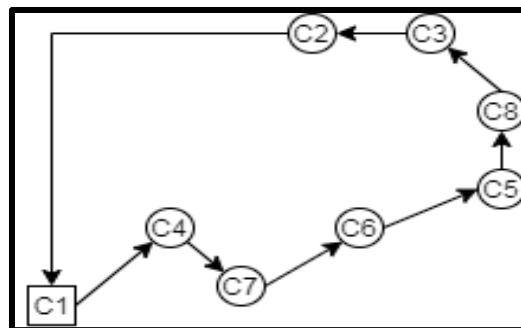
**Gambar 2. 1** Algoritma *Nearest Neighbor* titik awal ke titik pengambilan

Pada gambar 2.1 diatas terdapat beberapa titik TPS, diantaranya ada C1, C2, C3, C4, C5, C6, C7, dan C8.



**Gambar 2. 2** Algoritma *Nearest Neighbor* terpilih pertama ke titik pengambilan berikutnya

Pada gambar 2.2 diatas terdapat beberapa titik TPS. C1 sebagai sebagai titik awal keberangkatan. Kemudian algoritma memilih rute tercepat, terpilihlah C4 sebagai rute selanjutnya. Kemudian setelah C4 tuntas, algoritma memilih C7 sebagai rute tercepat selanjutnya.

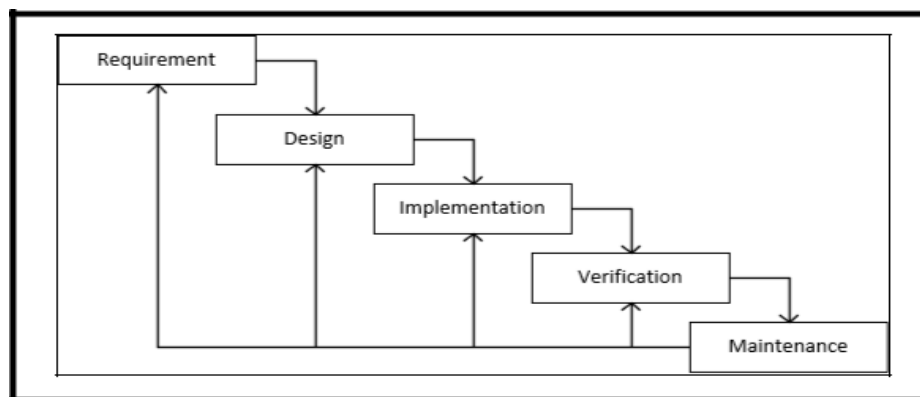


**Gambar 2. 3** Algoritma *Nearest Neighbor* urutan pengambilan dari titik awal ke semua rute

Pada gambar 2.3 diatas terdapat beberapa titik TPS dengan C1 sebagai titik awal keberangkatan dan C3, C4, C5, C6, C7, C8, dan C2. Pada awalnya titik awal memilih 1 diantara rute lainnya dan akan memilih rute tercepat dari beberapa titik pengambilan tersebut. Kemudian setelah C1 tuntas, algoritma akan menghitung kembali rute berikutnya dari titik terakhir ke beberapa titik yang akan dituju, dan seterusnya sampai titik yang akan dituju yaitu titik untuk kembali ke titik awal (C1). Dari contoh gambar diatas terpilihlah jalur rute C1 (1)→C4 (2) → C7 (3) → C6 (4) → C5 (5) → C8 (6) → C3 (7) → C2 (8) → C1 (9).

#### 2.4 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah metode *waterfall* (siklus air terjun). Metode *waterfall* menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*construction*) serta penyerahan sistem kepada pengguna (*deployment*) yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Supandi et al., 2018). Tahapan dalam Metode *Waterfall* dapat dilihat pada gambar 2.4.



**Gambar 2. 4** Tahapan Metode Waterfall

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut.

Tahapan tahapan dari metode *waterfall* adalah sebagai berikut:

1. *Requirements Analysis*

Dalam tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

## 2. *System and Software design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

## 3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

## 4. *Integration and System Testing*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

## 5. *Operation & maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

### 2.5 Data Flow Diagram (DFD)

Menurut (Muttaqin, 2014) DFD adalah Diagram yang menggunakan notasi simbol untuk menggambarkan arus data sistem. Ada beberapa simbol yang digunakan pada DFD, yaitu :

#### 1. Entitas Eksternal (External Entity)

*Entitas eksternal (external entity)* merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain yang berada pada lingkungan luarnya yang berinteraksi dengan sistem dengan memberikan input atau menerima output.

#### 2. Proses (Process)

*Proses (process)* menunjukkan pada bagian yang mengubah input menjadi *output*, yaitu menunjukkan bagaimana satu atau lebih *input* diubah menjadi beberapa *output*. Setiap proses mempunyai nama, nama dari proses ini menunjukkan apa yang dikerjakan proses.



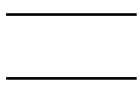
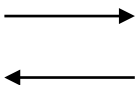
### 3. Simpanan Data (Data Store)

*Data Store* merupakan simpanan dari data yang dapat berupa suatu file atau database pada sistem komputer.

### 4. Arus Data (Data Flow)

Arus Data (*data flow*) di DFD diberi simbol panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

**Tabel 2. 1** Simbol dalam DFD



Simbol	Keterangan
	Entitas Eksternal
	Proses
	Data Store
	Data Flow

## 2.6 Flowchart




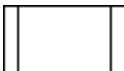



*Flowchart* merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. (Ilham Akhsanu Ridlo, 2017).

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart* :

**Tabel 2. 2** Simbol dalam Flowchart

Simbol	Nama	Fungsi
	<i>Terminator</i>	Permulaan atau akhir program
	Garis alir ( <i>flow line</i> )	Arah aliran program

**Tabel 2. 3** Simbol dalam Flowchart (Lanjutan)

	<i>Preparation</i>	Proses inisiasi atau pemberian harga awal
	Proses	Proses perhitungan atau proses pengolahan data
	<i>Input/output data</i>	Proses <i>input/output</i> data, parameter, informasi
	<i>Predefined process</i> (sub program)	Permulaan sub program atau proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>On page connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
	<i>Off page connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda

## 2.7 Database

Pengertian *database* adalah kumpulan informasi dan data yang saling berhubungan satu sama lain, dimana data tersebut tersimpan di simpanan luar komputer dan diperlukan *software* tertentu untuk memanipulasinya. *Database* merupakan suatu kesatuan yang dibentuk dari gabungan tabel dan file, dimana setiap tabel terdiri dari *record* yang disusun atas *field-field* yang ada didalamnya. (Simarmata, 2008)

## 2.8 Entity Relationship Diagram (ERD)

Menurut (Agarina & Karim, 2018) *Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi yang terdiri dari objek-objek dasar sehingga mempunyai hubungan atau relasi antar objek-objek tertentu. Relasi antar objek dilukiskan dengan menggunakan simbol-simbol grafis tertentu.

### 1. Entitas

*Entitas* adalah sesuatu objek yang dapat didenifinisikan dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.



## 2. Atribut

Setiap entitas mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut memiliki sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

## 3. Kardinalitas/Derajat Relasi

Relasi adalah hubungan antara suatu himpunan dengan himpunan entitas yang lainnya. Pada penggambaran diagram hubungan entitas, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas lainnya. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu *database*. Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa:

### 1. Satu ke Satu (*One to One*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

### 2. Satu ke Banyak (*One to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.


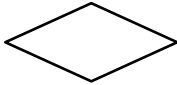
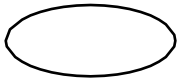

### 3. Banyak ke Satu (*Many to One*)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya dengan entitas B.

### 4. Banyak ke Banyak (*Many to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

**Tabel 2. 4** Simbol dalam ERD

Simbol	Nama	Keterangan
	Entitas	Entitas merupakan suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi	Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
	Atribut	Atribut berfungsi mendeskripsikan karakter entitas.
	Garis	Garis merupakan penghubung antara entitas, atribut dan relasi.

## 2.9 Tinjauan Pustaka

Adapun penelitian yang dilakukan sebelumnya ada beberapa variasi dan keunggulan masing-masing, berikut penelitian sebelumnya di antaranya :

1. Penelitian yang dilakukan oleh (Andayani Sri, 2014) dengan judul **Optimasi Rute Terpendek Pengambilan Sampah Di Kota Merauke Menggunakan Algoritma Dijkstra**. Pada penelitian ini dilakukan penentuan rute terpendek di kota Merauke menggunakan metode Algoritma Dijkstra.
2. Penelitian oleh (Abadi et al, 2014) dengan judul **Optimasi Rute Kendaraan Distribusi Produk Roti Menggunakan Metode Nearest Neighbor Dan Metode Sequential Insertion**. Pada Penelitian ini dilakukan peluang penentuan rute kendaraan distribusi produk roti dari titik terdekat lebih besar.
3. Penelitian oleh (Kasus et al, 2014) dengan judul **Optimasi Rute Distribusi Es Balok Menggunakan Algoritma Nearest Neighbor Dan Local Search**. Pada penelitian ini dilakukan penentuan rute terpendek dan tercepat karena dalam penelitian ini hanya menentukan es balok mencair dalam hitungan jam.
4. Penelitian oleh (Arinalhaq et al, 2013) dengan judul **Optimasi Rute Kendaraan Pengangkutan Sampah Dengan Menggunakan Metode Nearest Neighbor**. Pada penelitian ini dilakukan menentukan rute kendaraan pengangkut sampah serta belum diterapkannya penelitian ini pada sistem operasi berbasis android maupun berbasis web serta dibutuhkan penelitian lanjutan terkait penelitian ini dengan menimbang jarak rute.

5. Penelitian yang dilakukan oleh (Pratama Dicky, 2019) dengan judul **Optimasi Travelling Salesman Problem With Time Windows (TSP-TW) Pada Rute Obyek Wisata Di Provinsi D.I.Yogyakarta Menggunakan Algoritma Genetika Berbasis Android**. Namun perbedaanya adalah menentukan rute obyek wisata di Yogyakarta serta waktu tempuh, jadwal buka, dan jadwal tutup obyek wisata.

**Tabel 2. 5** Penelitian sebelumnya

No	Paper	Author
1.	Optimasi Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma <i>Dijkstra</i>	(Andayani Sri, 2014)
2.	Optimasi Rute Kendaraan Distribusi Produk Roti Menggunakan Metode Nearest Neighbor dan Metode Sequential Insertion	(Abadi et al., 2014)
3.	Optimasi Rute Distribusi Es Balok Menggunakan Algoritma Nearest Neighbor dan Local Search	(Hutasoit et al., 2014)
4.	Optimasi Rute Kendaraan Pengangkutan Sampah dengan Menggunakan Metode <i>Nearest Neighbor</i>	(Arinalhaq et al., 2013)
5.	Optimasi Travelling Salesman Problem With Time Windows (TSP-TW) Pada Rute Obyek Wisata Di Provinsi D.I.Yogyakarta Menggunakan Algoritma Genetika Berbasis Android	(Pratama Dicky., 2019)

## BAB III METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

Pada bab ini membahas mengenai beberapa aspek yaitu pengambilan data dinas dengan wawancara yang digunakan sebagai acuan dalam penelitian penerapan metode *Nearest Neighbor* untuk penentuan rute dari TPS menuju ke TPA sebagai media tanggap pengangkutan sampah menggunakan teknologi *Google Maps API* berbasis web. Metode *Nearest Neighbour* merupakan sebuah teknik dalam menyelesaikan permasalahan rute dengan cara menentukan waktu terdekat dari titik awal keberangkatan. Metode ini merupakan metode yang sederhana dalam memecahkan masalah rute dan merupakan solusi awal dalam pemilihan titik TPS terdekat berdasarkan asumsi jumlah kendaraan dalam setiap titik TPS dan titik TPA. Cara kerja metode ini adalah sebagai berikut. Pertama-tama, semua rute kendaraan masih kosong. Dimulai dari rute kendaraan pertama, metode ini memasukkan (*insert*) satu persatu titik pengambilan terdekat (*nearest neighbor*) yang belum dikunjungi ke dalam rute. Selama memasukkan titik pengambilan tersebut ke dalam rute, kendaraan tidak melanggar batasan kapasitas maksimum kendaraan tersebut. (Amri et al., 2014)

Metode yang diterapkan untuk pengembangan sistem dalam penelitian ini adalah metode *waterfall*. Metode pengembangan sistem (*waterfall*) memiliki tahap-tahap sebagai berikut yaitu analisis kebutuhan sistem, desain sistem implementasi dan pengujian unit, pengujian sistem, perawatan (*maintenance*).

#### 3.1.1 Data Wawancara

Data yang digunakan dalam penelitian ini adalah data wawancara yang diperoleh dari Dinas Lingkungan Hidup Kota Yogyakarta yang terletak di Jl Bima Sakti, Demangan, Gondokusuman, Kota Yogyakarta. Data yang diambil adalah Data sarana prasarana, data rute antar TPS ke TPA dan data jumlah TPS di kota Yogyakarta (Dikunjungi pada Senin, 15 Maret 2021 pukul 09.00).

### 3.2 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah metode *waterfall* (siklus air terjun). Metode *waterfall* menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*construction*) serta penyerahan *system* kepada pengguna (*deployment*) yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Supandi et al., 2018).

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut:

#### 1. *Requirements Analysis*

Dalam tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

#### 2. *System and Software design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

#### 3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

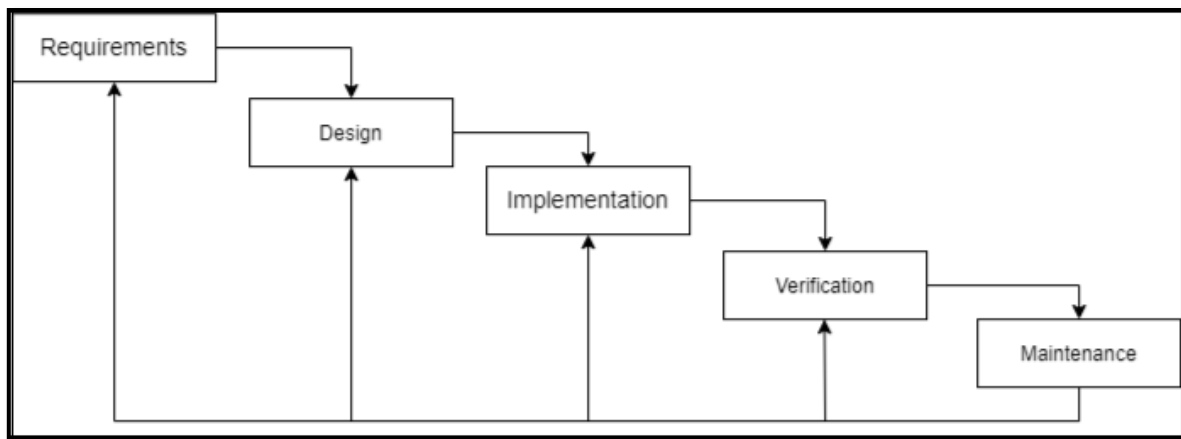
#### 4. *Integration and System Testing*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

#### 5. *Operation & maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Dalam penelitian ini hanya pada tahap pengujian tanpa ada tahap *maintenance* atau pemeliharaan.



**Gambar 3. 1** Tahapan metode *waterfall*

Pada tahap metodologi penelitian, proses penyelesaian dilakukan dengan tiga tahap. Tahap pertama dilakukan dengan pengumpulan data yang terdiri dari tiga tahap yaitu survey data, studi pustaka dan observasi.

### 3.3 Survei Data

Pengumpulan data bertujuan untuk mendapatkan informasi mengenai apa saja yang dibutuhkan untuk mendukung pembuatan sebuah sistem. Pengumpulan data pada penelitian ini terdapat beberapa tahapan yaitu studi pustaka dan observasi. Tahapan ini mencakup pengumpulan informasi. Dari tahapan tersebut dilakukan analisis masalah dengan pengumpulan data yang berkaitan dengan penentuan jalur pengambilan dan pembuangan sampah. Data yang digunakan merupakan data dari Dinas Lingkungan Hidup kota Yogyakarta.

#### 3.1.2 Studi Pustaka

Studi pustaka merupakan proses pengumpulan buku atau jurnal penelitian sebagai referensi yang memiliki tujuan untuk memperkuat materi pembahasan dalam penelitian yang dilakukan dan mempelajari dasar dasar yang digunakan. Khususnya mempelajari penentuan jalur evakuasi, metode *nearest neighbor* dan pembuatan aplikasi berbasis web.

#### 3.1.3 Observasi

Observasi merupakan sebuah proses pencatatan atau pengamatan untuk mendapatkan sebuah informasi yang dibutuhkan dalam melakukan penelitian. Observasi ini langsung dilakukan ke Dinas Lingkungan Hidup kota Yogyakarta untuk melakukan penelitian tugas akhir. Observasi yang dilakukan di Dinas Lingkungan Hidup kota Yogyakarta adalah pengumpulan data yang berisi tentang data jumlah sarana prasarana, data jalur TPS ke TPS, data TPS ke TPA, data lokasi TPS, dan data lokasi TPA.



### 3.4 ANALISIS KEBUTUHAN SISTEM

Analisis kebutuhan sistem dalam hal ini menganalisa hal-hal yang berkaitan dengan sistem yang akan dibangun. Tujuan dari analisis kebutuhan sistem adalah memahami kebutuhan dari sistem yang mewadahi kebutuhan sistem dan kebutuhan pengguna.

Dalam sistem ini untuk melakukan penentuan jalur rute TPS ke TPS tercepat, TPS ke TPA tercepat untuk mendapatkan optimasi waktu yang efisien.

#### 3.1.4 Kebutuhan Perangkat Lunak (*Software*)

Perangkat Lunak yang digunakan untuk penelitian ini antara lain :

1. Sistem Operasi yang digunakan adalah Microsoft Windows.
2. Web Server menggunakan XAMMP.
3. Database menggunakan MySQL.
4. Web Browser yang digunakan adalah Microsoft Edge dan Google Chrome.
5. Bahasa Pemrograman yang digunakan adalah PHP.
6. Sublime Text 3.
7. Program Pendukung yang digunakan Draw Io.
8. Microsoft Office Home Student 2019.
9. yED Graph Editor.

#### 3.1.5 Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk melakukan penelitian ini adalah Laptop Dell Vostro 5470, mouse, dan printer. Spesifikasi laptop antara lain :

1. Intel(R) Core(TM) i3-4030U CPU @ 1,9 GHz
2. SSD dengan kapasitas 500 GB
3. RAM dengan kapasitas 4 GB.
4. Ukuran LCD dengan layar 14-inch.

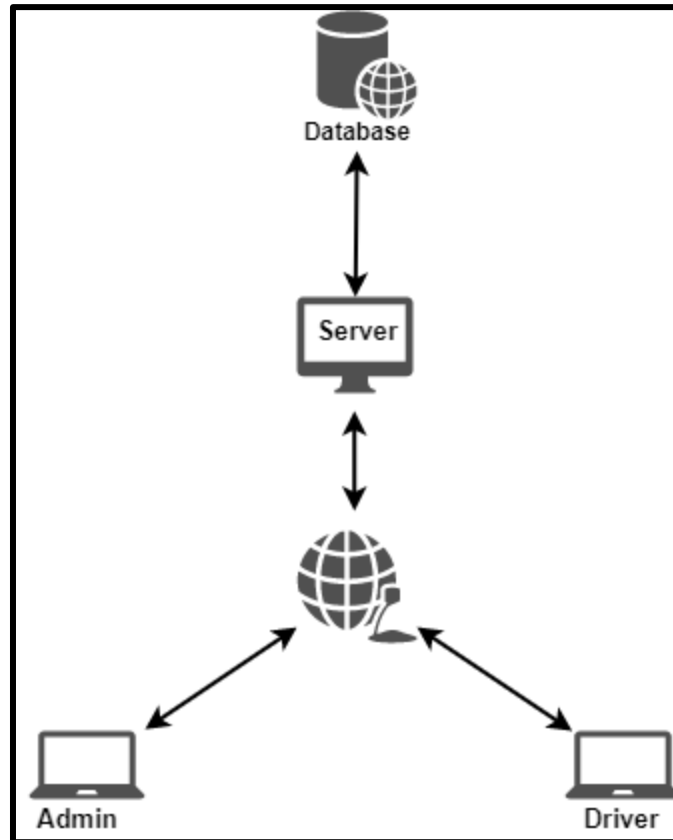
### 3.5 Perancangan Sistem

Dalam perancangan sistem diperlukan langkah identifikasi kebutuhan sistem, dalam hal ini penting dilakukan agar sistem yang di bentuk sesuai dengan kebutuhan. Perancangan system ini terdiri dari :

1. Arsitektur Komputer
2. Perancangan Proses
3. *Flowchart*
4. Perancangan Basis Data
5. Perancangan *User Interface*

### 3.6 Arsitektur Sistem

Perancangan arsitektur sistem menggambarkan proses berjalannya suatu sistem. Arsitektur sistem ini terdiri dari admin dan *driver*. Admin dan *driver* terhubung dengan server. Server terhubung dengan database untuk mengelola atau menampilkan sebuah data. Tampilannya dapat dilihat pada gambar 3.2.



**Gambar 3. 2** Arsitektur Sistem

Arsitektur sistem yang dibangun terdiri dari beberapa komponen yaitu pengguna (*user*), komputer (*system*), internet, dan *database*. Pengguna (*user*) dalam sistem ini adalah admin dan *driver*. Dimana fungsi admin sebagai operator aplikasi dan *driver* berfungsi memonitoring aplikasi. Server terhubung dengan internet, dan database untuk mengelola atau menampilkan data. Pembagian hak akses seperti pada tabel 3.1.

**Tabel 3. 1** Pembagian Hak Akses Pengguna

No	Halaman	Admin	Driver	Keterangan
1	Halaman Jalur Pengambilan	R	R	Admin Memonitoring Semua Rute Pengambilan , Driver hanya membaca rute pribadi.
2	Halaman Rute Menuju TPS	CRED	R	Admin memasukan Data TPS dan TPA Driver hanya dapat Membaca data TPS dan TPA.
3	Halaman Kendaraan	CRED	R	Admin yang dapat Memasukan data Kendaraan, Driver hanya dapat membaca data kendaraan.
4	Halaman Kendaraan TPA	CRED	-	Hanya Admin yang dapat Mengakses data Kendaraan.
5	Halaman TPA	CRED	R	Admin yang dapat Memasukan data TPA, Driver hanya dapat membaca data TPA.
6	Halaman User	CRED	-	Hanya Admin yang dapat Mengsakses data Data User.

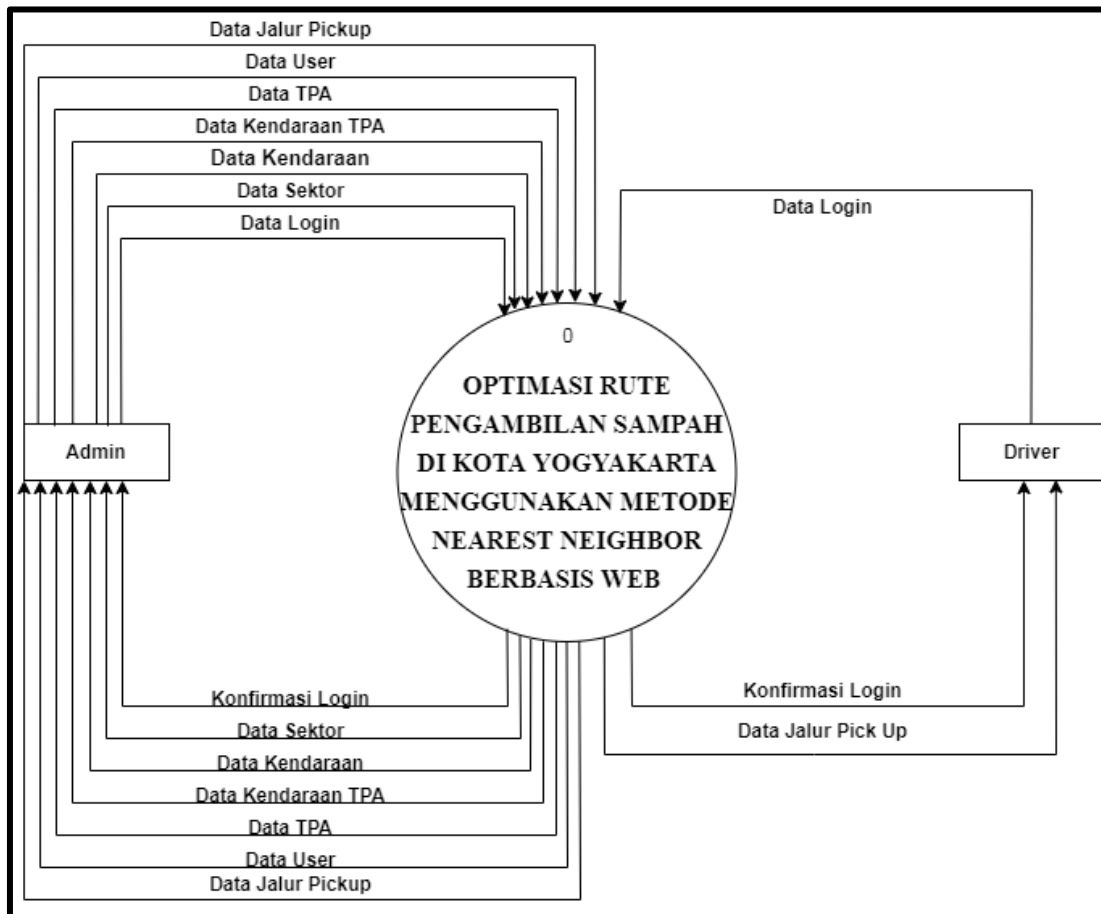
Keterangan: C= Create, R=Read, E=Edit, D=Delete

### 3.7 Perancangan Proses

Pada segmen ini perancangan dibagi menjadi perancangan proses dengan membuat *Data Flow Diagram* (DFD), perancangan basis data (ERD, perancangan tabel dan RAT), perancangan struktur menu dan perancangan antar muka (*user interface*). Perancangan proses merupakan tahapan yang digunakan untuk menggambarkan proses dibangunnya suatu sistem. Perancangan proses akan digambarkan dalam *Data Flow Diagram* (DFD) dari level 0 sampai dengan level 1, dan perancangan *user interface*.

#### 3.1.6 Data Flow Diagram (DFD) Level 0

DFD level 0 merupakan gambaran secara keseluruhan dari sistem yang dibuat berupa aliran masuk dan aliran keluar yang ditunjukkan dengan arah anak panah. Melalui DFD tersebut dapat dilihat bahwa sistem ini memiliki 2 entitas yaitu Admin dan Driver. Hasil gambar DFD level 0 dapat dilihat pada gambar 3.3.

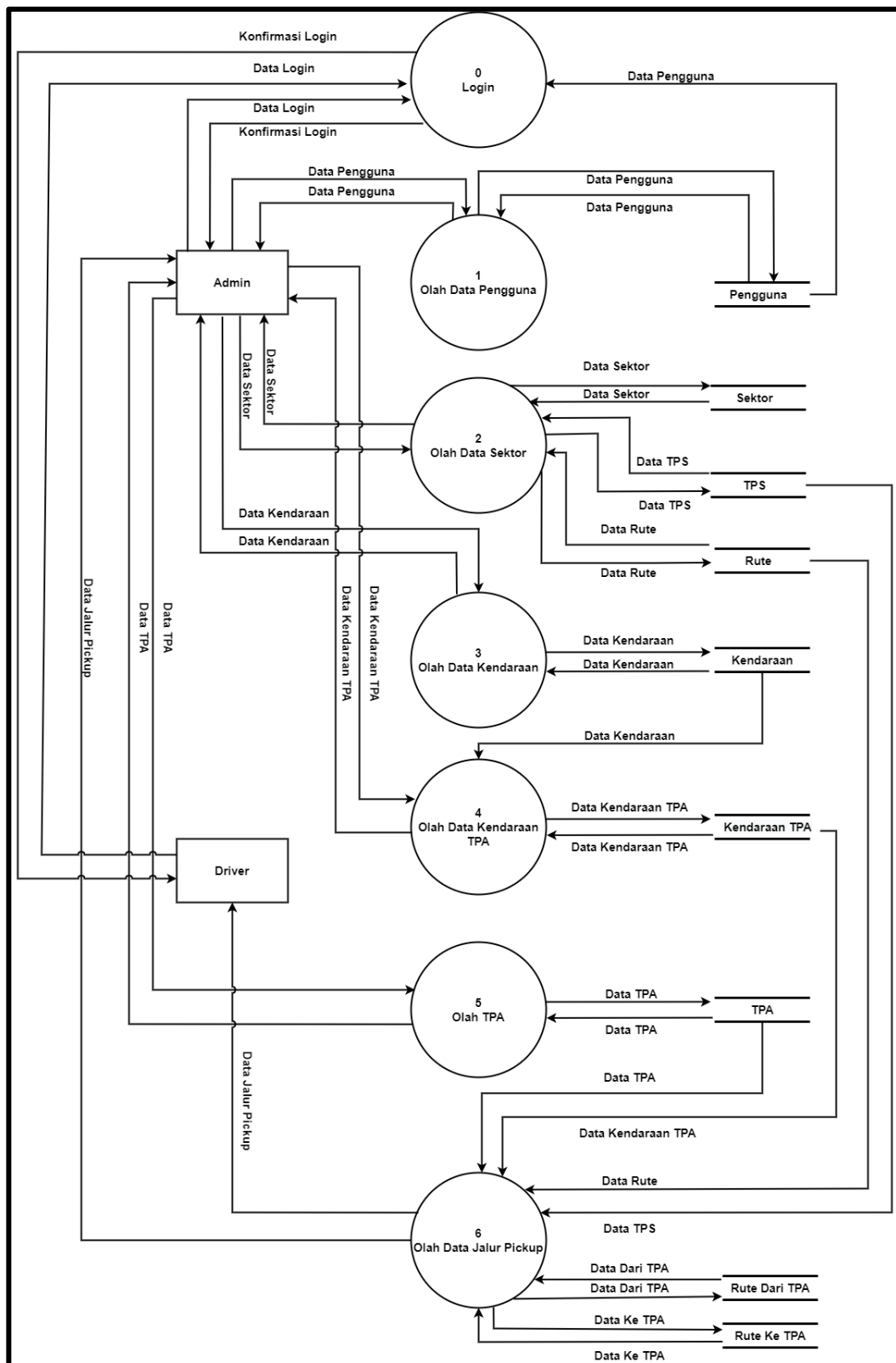


**Gambar 3. 3** Data Flow Diagram (DFD) Level 0

DFD level 0 menggambarkan *entitas* admin dan *driver* memiliki perbedaan terlihat dari aliran data yang masuk dan keluar dari *entitas* tersebut. *Entitas* admin memiliki aliran data masuk dan data keluar dalam sistem, maka secara tidak langsung entitas admin berfungsi sebagai pengolah semua data dalam sistem proses tersebut. Sedangkan *entitas driver* hanya dapat melihat informasi yang ada didalam sistem proses tersebut.

### 3.1.7 Data Flow Diagram (DFD) Level 1

DFD level 1 merupakan turunan dari proses DFD level 0, dalam level ini menggambarkan dengan jelas perjalanan aliran data *entitas*, proses dan basis data. DFD level 1 mempunyai proses utama yaitu *Proses Login*, Olah Data Pengguna, Olah Data Sektor, Olah Data Kendaraan, Olah Data Kendaraan TPA, Olah Data TPA dan Proses Olah Data Jalur Pickup. DFD level 1 dapat dilihat pada gambar 3.4.



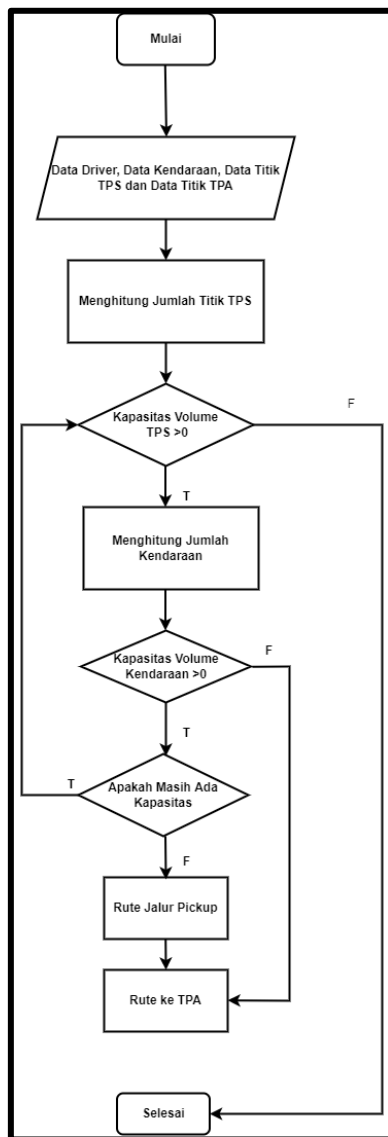
Gambar 3. 4 Data Flow Diagram (DFD) Level 1

Dalam DFD level 1 sistem ini memiliki 7 proses yaitu:

1. Proses 0 Login  
Proses ini admin dan *driver* melakukan login dengan id dan password.
2. Proses 1 Olah Data Pengguna  
Proses ini hanya admin yang dapat menambah dan mengedit Pengguna atau driver.
3. Proses 2 Data Sektor  
Proses ini hanya admin dan *driver* hanya menerima informasi tentang jalur Pengambilan saja dan tidak bisa melakukan olah data.
4. Proses 3 Olah Data Kendaraan  
Proses ini admin dapat menambah kendaraan dan hanya diakses oleh admin.
5. Proses 4 Olah Data Kendaraan TPA  
Proses ini hanya bisa diakses oleh admin dan admin bisa menambah, mengedit dan menghapus data.
6. Proses 5 Olah Data TPA  
Proses ini hanya bisa diakses oleh admin dan admin bisa mengedit dan menambah TPA.
7. Proses 6 Olah Data Jalur Pickup  
Proses ini admin dapat membuat, melihat, mengubah dan menghapus data yang dibutuhkan yang nantinya digunakan dalam proses jalur pengambilan, sedangkan *driver* hanya dapat menerima atau melihat informasi.

### 3.8 Flowchart

*Flowchart* adalah bagan yang menunjukkan alir atau arus (*flow*) di dalam program atau prosedur sistem secara logika. Pada penelitian ini terdapat *flowchart* yang digunakan untuk menggambarkan alur proses yang digunakan untuk menentukan titik awal keberangkatan, dan penentuan jalur pengambilan yang terefisien menuju TPS dan TPA.



**Gambar 3. 5** Flowchart Sistem

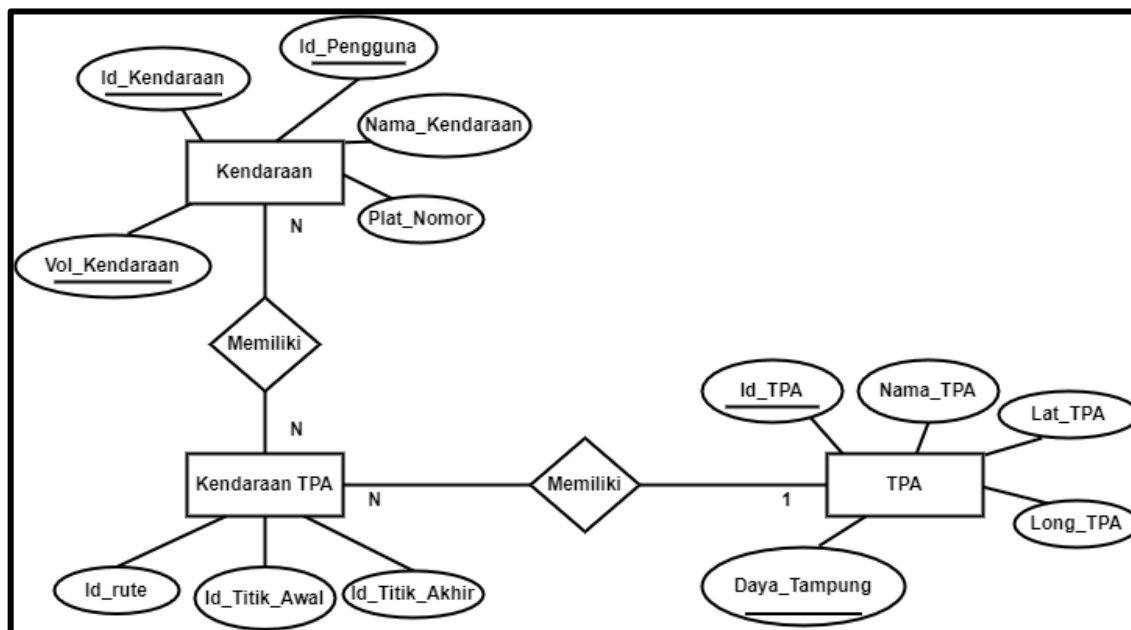
Pada *Flowchart* dimulai dengan pengambilan data *driver*, data kendaraan, data titik TPS dan data titik TPA. Lalu selanjutnya menghitung jumlah titik TPS, apakah masih ada sampah yang berada di titik TPS. Jika masih ada, mendefinisikan total kendaraan kemudian mengambil waktu setiap titik TPS. Jika sudah, *driver* mengambil waktu terpendek yang sudah ditentukan oleh perhitungan fitur *google map* dan mengambil sampah yang ada di TPS tersebut jika masih ada sisa kapasitas maka kembali ke proses sebelumnya. Jika kapasitas sudah penuh maka *driver* tinggal menuju TPA. Proses tersebut akan berulang sampai tidak ada sampah yang berada di titik TPS.

### 3.9 Perancangan Basis Data

Basis data merupakan salah satu komponen yang penting dalam membangun sistem. Basis data berfungsi sebagai media penyimpanan dan pengolahan data sebagai arsitektur dasar sebuah sistem. Dalam merancang basis data, ada beberapa tahapan yang harus dilalui yaitu merancang *Entity Relation Diagram* (ERD), struktur tabel, Relasi Antar Tabel (RAT).

#### 3.1.8 Entity Relationship Diagram (ERD)

Tujuan utama dari ERD adalah untuk menunjukkan struktur objek data (*entity*) dan hubungan (*relationship*) yang ada pada objek tersebut. ERD juga berguna sebagai professional sistem, karena memperlihatkan hubungan antara data store dan *Data Flow Diagram* (DFD). *Entity Relationship Diagram* (ERD) pada perancangan basis data dapat dilihat pada gambar 3.6.



**Gambar 3. 6** Entity Relationship Diagram



### 3.1.9 Perancangan Struktur Tabel

Untuk memenuhi kebutuhan sistem dalam penyimpanan data terutama dalam jumlah data yang besar maka diperlukan suatu basis data. Dibawah ini akan dijelaskan tabel yang dibutuhkan antara lain.

#### 1. Tabel Pengguna

Tabel ini berfungsi untuk menyimpan data-data login admin dan *driver*

**Tabel 3. 2** Tabel Data Pengguna

Nama Field	Tipe Data	Constraint	Keterangan
id_pengguna	int (11)	Primary Key	Id Pengguna
nama	varchar (50)	Not Null	Nama Pengguna
username	varchar (8)	Not Null	Id Login
password	varchar (100)	Not Null	Pasword User
level	enum('admin','driver')	Not Null	Level Pengguna

#### 2. Tabel Kendaraan

Tabel ini berfungsi untuk menyimpan data Kendaraan.

**Tabel 3. 3** Tabel Kendaraan

Nama Field	Tipe Data	Constraint	Keterangan
id_kendaraan	int (11)	Primary Key	Id Kendaraan
id_pengguna	int (11)	Not Null	IdPenguna
nama_kendaraan	varchar (50)	Not Null	Nama Kendaraan
plat_nomor	varchar (50)	Not Null	Plat Nomor
volume_kendaraan	int (11)	Not Null	Volume Kendaraan

#### 3. Tabel Kendaraan TPA

Tabel ini berfungsi untuk menyimpan data Kendaraan TPA.

**Tabel 3. 4** Tabel Kendaraan TPA

Nama Field	Tipe Data	Constraint	Keterangan
id_kendaraan_tpa_	int (11)	Primary Key	Id Kendaraan
id_kendaraan	int (11)	Not Null	Id Pengguna
id_tpa	int (11)	Not Null	Nama Kendaraan

#### 4. Tabel Rute dari TPA

Tabel ini berfungsi untuk menyimpan data keberangkatan titik awal dari TPA menuju ke TPS

**Tabel 3. 5** Tabel Rute dari TPA

Nama Field	Tipe Data	Constraint	Keterangan
id_rute	int (11)	Primary Key	Id Rute
id_titik_awal	int (11)	Null	Id Titik Awal
id_titik_akhir	int (11)	Null	Id Titik Akhir
waktu	float	Null	Waktu Yang Diperlukan

waktu_text	varchar (200)	Null	Waktu Dalam Menit
origin	text	Null	Nama Titik Awal
destination	text	Null	Nama Titik Akhir

## 5. Tabel Rute ke TPA

Tabel ini berfungsi untuk menyimpan data rute menuju ke TPA dari TPS

**Tabel 3. 6** Tabel Rute ke TPA

Nama Field	Tipe Data	Constraint	Keterangan
id_rute	int (11)	Primary Key	Id Rute
id_titik_awal	int (11)	Null	Id Titik Awal
id_titik_akhir	int (11)	Null	Id Titik Akhir
waktu	float	Null	Waktu Yang Diperlukan
waktu_text	varchar (200)	Null	Waktu Dalam Menit
origin	text	Null	Nama Titik Awal
destination	text	Null	Nama Titik Akhir

## 6. Tabel Rute TPS

Tabel ini berfungsi untuk menyimpan data Rute TPS yang akan dituju

**Tabel 3. 7** Tabel Rute TPS

Nama Field	Tipe Data	Constraint	Keterangan
id_rute	int (11)	Primary Key	Id Rute
id_titik_awal	int (11)	Null	Id Titik Awal
id_titik_akhir	int (11)	Null	Id Titik Akhir
waktu	float	Null	Waktu Yang Diperlukan
waktu_text	varchar (200)	Null	Waktu Dalam Menit
origin	text	Null	Nama Titik Awal
destination	text	Null	Nama Titik Akhir

## 7. Tabel Sektor

Tabel ini berfungsi untuk menyimpan data Sektor

**Tabel 3. 8** Tabel Sektor

Nama Field	Tipe Data	Constraint	Keterangan
id_sektor	int (11)	Primary Key	Id Sektor
nama_sektor	varchar (100)	Not Null	Nama Sektor
kapasitas_sektor	int (11)	Not Null	Kapasitas Sektor

## 8. Tabel TPA

Tabel ini berfungsi untuk menyimpan data TPA

**Tabel 3. 9** Tabel TPA

Nama Field	Tipe Data	Constraint	Keterangan
id_tpa	int (11)	Primary Key	Id TPA
nama_tpa	varchar (100)	Null	Nama TPA
lat_tpa	varchar (50)	Null	Latitude TPA
long_tpa	varchar(50)	Null	Longitude TPA
daya_tampung	int (11)	Null	Daya Tampung TPA

## 9. Tabel TPS

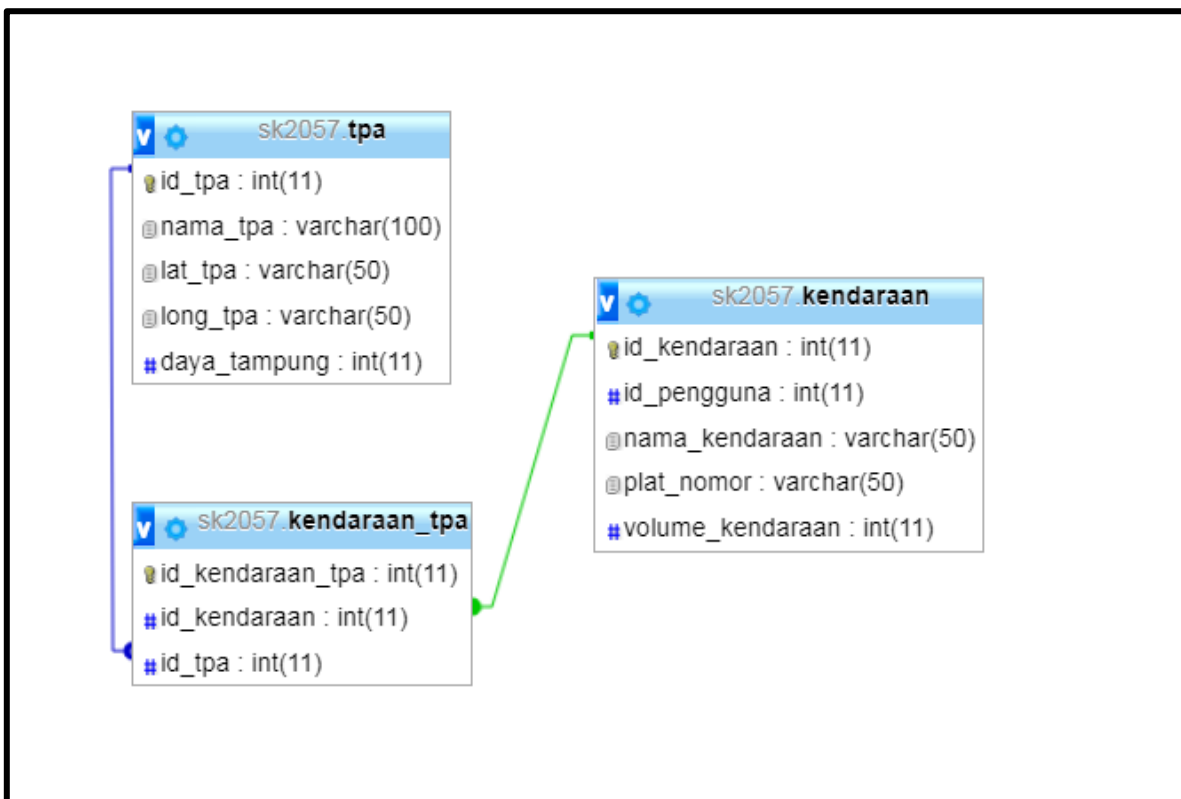
Tabel ini berfungsi untuk menyimpan data TPS

**Tabel 3. 10** Tabel TPS

Nama Field	Tipe Data	Constraint	Keterangan
id_tps	int (11)	Primary Key	Id TPS
nama_tps	varchar (100)	Null	Nama TPA
lat_tps	varchar (50)	Null	Latitude TPS
long_tps	varchar (50)	Null	Longitude TPS
id_sektor	int (11)	Null	Id Sektor
Volume	int (11)	Null	Volume TPS

### 3.1.10 Relasi antar tabel (RAT)

Terdapat tiga tabel yaitu pengguna, kendaraan, kendaraan\_tpa, sektor, tpa, tps, rute\_dari\_tpa, rute\_ke\_tpa, dan rute\_tps. Perancangan Relasi Antar Tabel dapat dilihat pada gambar 3.8.



**Gambar 3. 7** Relasi Antar Tabel (RAT)

### 3.10 Sample Perhitungan Metode

Pada sample perhitungan ini menggunakan data *sample* penentuan jalur pengambilan *driver* sektor Ngasem Gading. Sedangkan kendaraan yang digunakan untuk *sample* perhitungan ini adalah Amroll yang dikendarai oleh *driver* Sumpuna. Untuk *sample* perhitungan dari yang dilakukan oleh sistem dapat dilihat dibawah ini.

**Tabel 3. 11** Data Titik Awal Keberangkatan TPA

Nama TPS	Longitude	Latitude
TPA Piyungan	110.3947255	-7.8696834

**Tabel 3. 12** Data Kendaraan

Kendaraan	Driver	Plat Nomor	Daya Tampung
Amroll	Driver Sumpuna	AB 9092 UA	2000

**Tabel 3. 13** Perbandingan waku tercepat dari titik keberangkatan

Nama TPS	Longitude	Latitude	Daya Tampung(m20	tu/Detik
Jembatan Tungkak	110.3743523	-7.8160657	500	1064
Jalan Gajah	110.3847158	-7.8040062	500	1230
Jalan Pramuka	110.3866386	-7.8210272	500	930
Jl Sisingamangaraja	110.3729489	-7.8200816	500	941
Jalan Veteran	110.390849	-7.8094062	500	1115
Lp Wirogunan	110.3760161	-7.8037198	500	1204
Pasar Pujokusuman	110.3669065	-7.8139609	500	1299
Rs Pratama	110.3717171	-7.8155951	500	1183

**Tabel 3. 14** Hasil Data TPS Yang Terdekat Dari Titik Keberangkatan

Nama Titik Awal	Nama TPS	Waktu/Detik	Urutan
Piyungan	Jalan Pramuka	930	1
Piyungan	Jl Singsingamangaraja	941	2
Piyungan	Jembatan Tungkak	1064	3
Piyungan	Jalan Veteran	1115	4
Piyungan	Rs Pratama	1183	5
Piyungan	Lp Wirogunan	1204	6
Piyungan	Jalan Gajah	1230	7

Dari data yang sudah ada maka penentuan jalur pengambilan setiap *driver* menggunakan metode *nearest neighbor* dengan membandingkan waktu terpendek yang ditempuh dari titik awal *driver* menuju titik pembuangan. sebagai berikut :

1. Penentuan Jalur Pengambilan *Driver* Sumpana yang menggunakan kendaraan Dump Truck.  
Pemilihan Rute Titik Awal TPA Piyungan —→ Jalan Pramuka

**Tabel 3. 15** Perhitungan waktu dari titik awal menuju Jalan Pramuka

Nomor	Nama TPS	Waktu/dtk
1	Jalan Pramuka	930
2	Jalan Singsingamngaraja	941
3	Jembatan Tungkak	1064
4	Jalan Veteran	1115

2.

2. Pemilihan Rute TPA Piyungan—→ Jalan Pramuka—→ Jembatan Tungkak.

**Tabel 3. 16** Perhitungan waktu dari Jalan Pramuka menuju Jembatan Tungkak

Nomor	Nama TPS	Waktu/dtk
1	Jembatan Tungkak	344
2	Jalan Veteran	407
3	Jalan Singsingamangaraja	415

**Tabel 3. 17** Perhitungan waktu dari Jembatan Tungkak menuju Jl Singsingamangaraja

Nomor	Nama Barak	Waktu/dtk
1	Jalan Singsingamangaraja	191
2	Jalan Veteran	695

**Tabel 3. 18** Perhitungan waktu dari Jalan Singsingamangaraja menuju Jalan Veteran

Nomor	Nama Barak	Waktu/dtk
1	Jalan Veteran	536

**Tabel 3. 19** Waktu Tempuh Dari TPS Titik Akhir Menuju ke TPA

Nama TPS Akhir	Nama TPA	Waktu/dtk
Jalan Veteran	Piyungan	1115

Jadi setelah kendaraan keliling menuju TPS dan ternyata volume kendaraan sudah penuh maka kendaraan akan kembali menuju TPA. Jika kendaraan keliling TPS tapi volume belum penuh maka kendaraan akan menuju ke TPS selanjutnya. Karena TPS masing-masing memiliki 500m<sup>2</sup> dan volume kendaraan adalah 2000m<sup>2</sup>, jika kendaraan sudah mengelilingi 4 TPS maka seharusnya sudah penuh lalu kembali lagi ke titik awal.

### 3. Pemilihan Rute Lanjutan TPA Piyungan —► Rs Pratama

**Tabel 3. 20** Perhitungan waktu dari TPA Piyungan menuju Rs Pratama

Nomor	Nama TPS	Waktu/dtk
1	Rs Pratama	1183
2	Lp Wirogunan	1204
3	Jalan Gajah	1230
4	Pasar Pujokusuman	1299

**Tabel 3. 21** Perhitungan waktu dari Rs Pratama menuju Pasar Pujokusuman

Nomor	Nama TPS	Waktu/dtk
1	Pasara PujoKusuman	277
2	Lp Wirogunan	596
3	Jalan Gajah	746

**Tabel 3. 22** Perhitungan waktu dari Pasar Pujokusuman menuju Lp Wirogunan

Nomor	Nama Barak	Waktu/dtk
1	Lp Wirogunan	608
2	Jalan Gajah	770

**Tabel 3. 23** Perhitungan waktu dari Lp Wirogunan menuju Jalan Gajah

Nomor	Nama Barak	Waktu/dtk
1	Jalan Gajah	358

**Tabel 3. 24** Perhitungan waktu dari TPS Titik Akhir Menuju TPA

Nama TPS Akhir	Nama TPA	Waktu/dtk
Jalan Gajah	Piyungan	1230

**Tabel 3. 25** Hasil Perhitungan Akhir dari TPS ke TPS

Nama Titik Awal	Nama TPS	Waktu/Detik
Piyungan	Jalan Pramuka	930
Jalan Pramuka	Jembatan Tungkak	344
Jembatan Tungkak	Jl Singsingamangaraja	191
Jl Singsingamangaraja	Jalan Veteran	536
Jalan Veteran	TPA Piyungan	1115
TPA Piyungan	Rs Pratama	1183
Rs Pratama	Pasar Pujokusuman	277
Pasar Pujokusuman	Lp Wirogunan	608
Lp Wirogunan	Jalan Gajah	358
Jalan Gajah	TPA Piyungan	1230

### 3.11 Perancangan Antarmuka

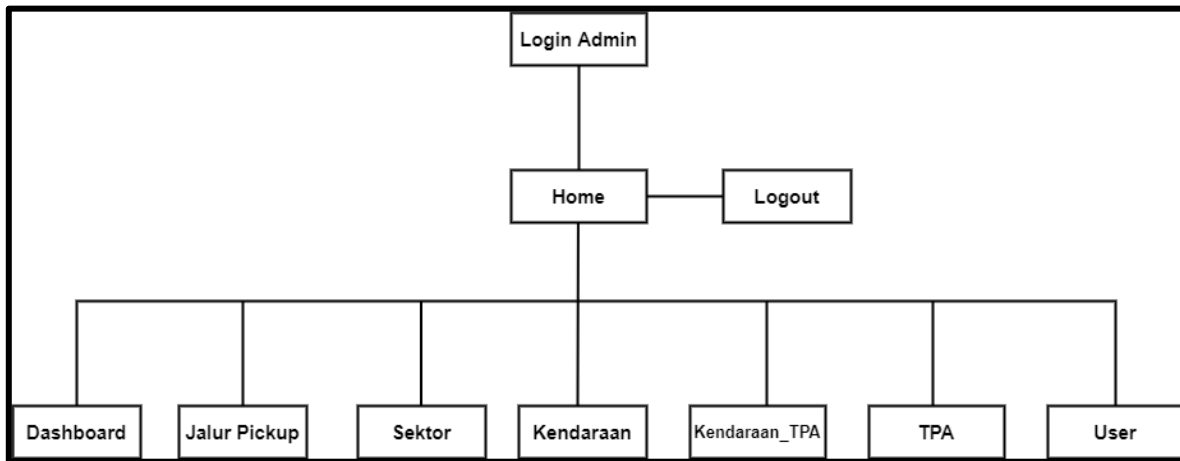
Perancangan antar muka merupakan bagian dimana terjadi komunikasi antara pengguna dengan sistem. Faktor tampilan juga mempengaruhi kemudahan dalam mengoperasikan sebuah sistem, sehingga bagaimana membuat suatu tampilan yang *interaktif* agar dapat dipahami oleh pengguna.

#### 3.1.11 Struktur Menu

Desain struktur menu pada penerapan metode *Nearest Neighbor* untuk penentuan jalur pengambilan dan pembuangan sampah menggunakan fitur *Google Maps* berbasis web terdiri dari menu login sebelum masuk ke sistem. *User* yang mengakses terdiri dari admin dan *driver* dan level aksesnya berbeda. Pada admin terdapat menu dashboard, jalur pickup, sektor, kendaraan, kendaraan TPA, TPA dan user. Sedangkan pada *driver* terdapat menu Dashboard, jalur pickup, sektor, kendaraan, dan TPA. Tampilannya dapat dilihat pada Gambar 3.8. dan gambar 3.9.

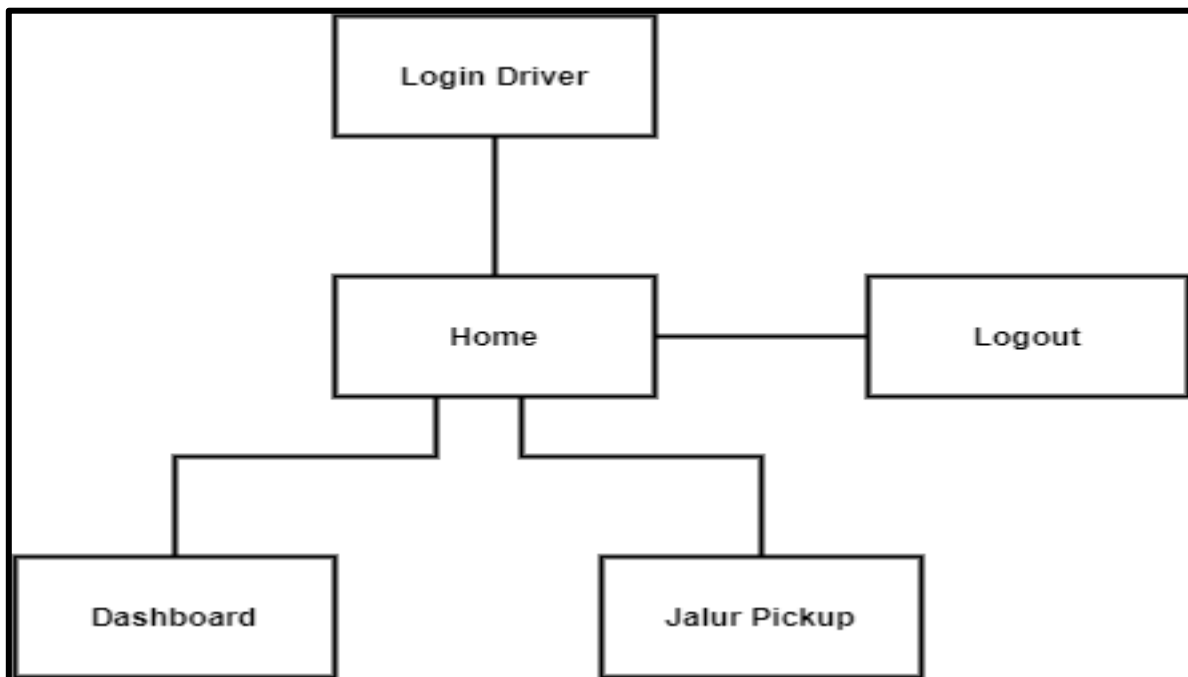


## 1. Menu Admin



**Gambar 3. 8** Struktur Menu Admin

## 2. Menu Driver



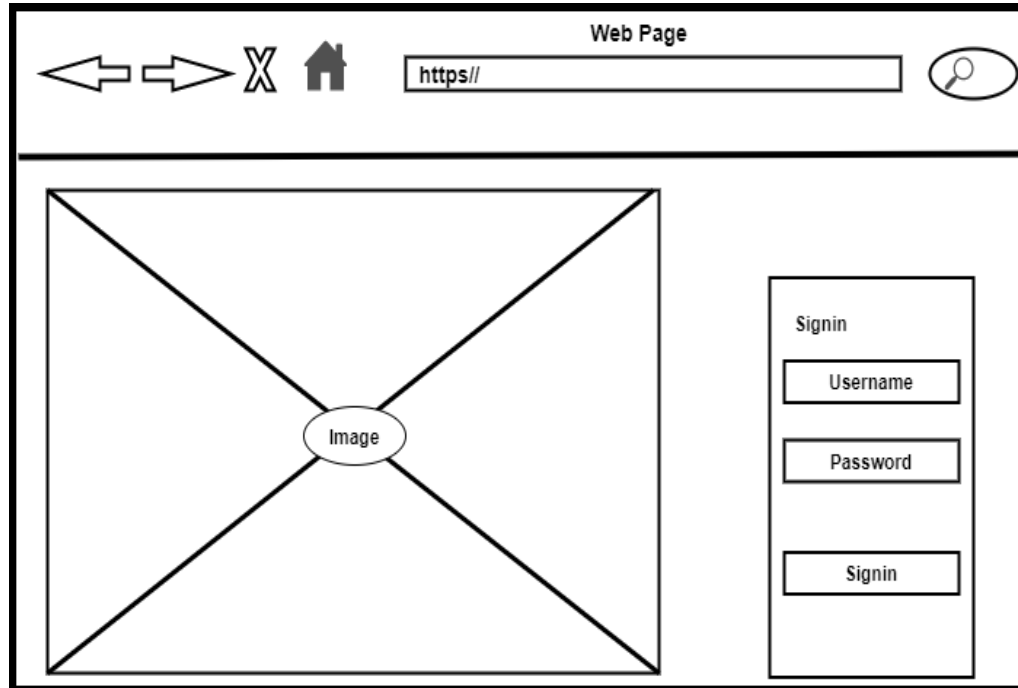
**Gambar 3. 9** Struktur Menu Driver

### 3.1.12 Perancangan User Interface

Perancangan user interface merupakan bentuk tampilan aplikasi yang akan di buat oleh peneliti, adapun rancangannya sebagai berikut:

#### 1. Halaman Login

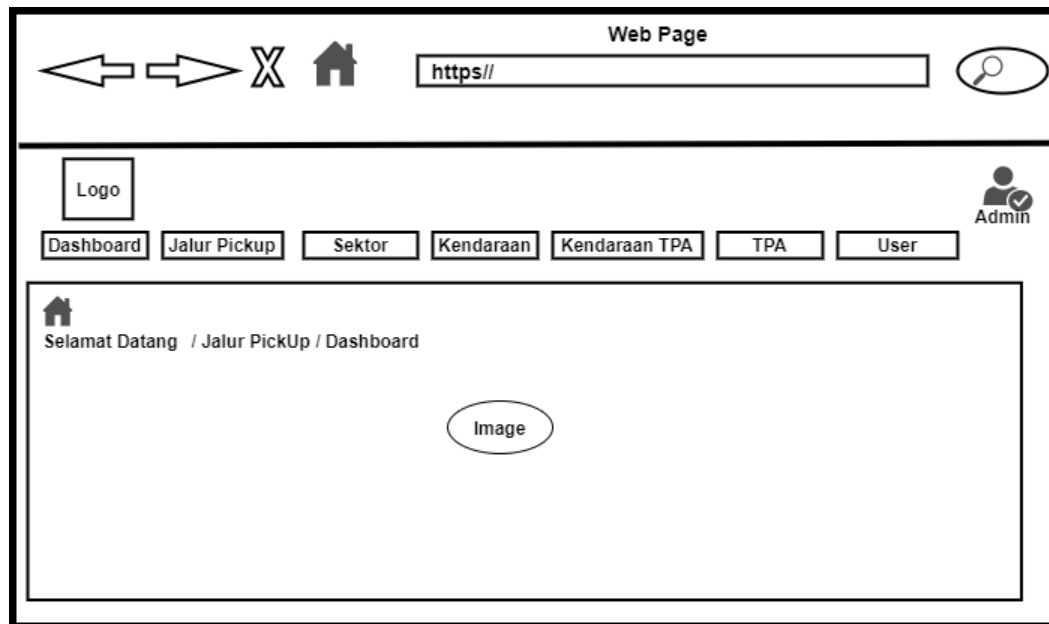
Halaman ini merupakan tampilan awal dari aplikasi sebelum admin atau *driver* masuk dalam sistem. Pada halaman login terdapat input *username* dan password jika username dan password benar maka akan dilanjutkan ke halaman berikutnya.



Gambar 3. 10 Login Form

## 2. Halaman Utama

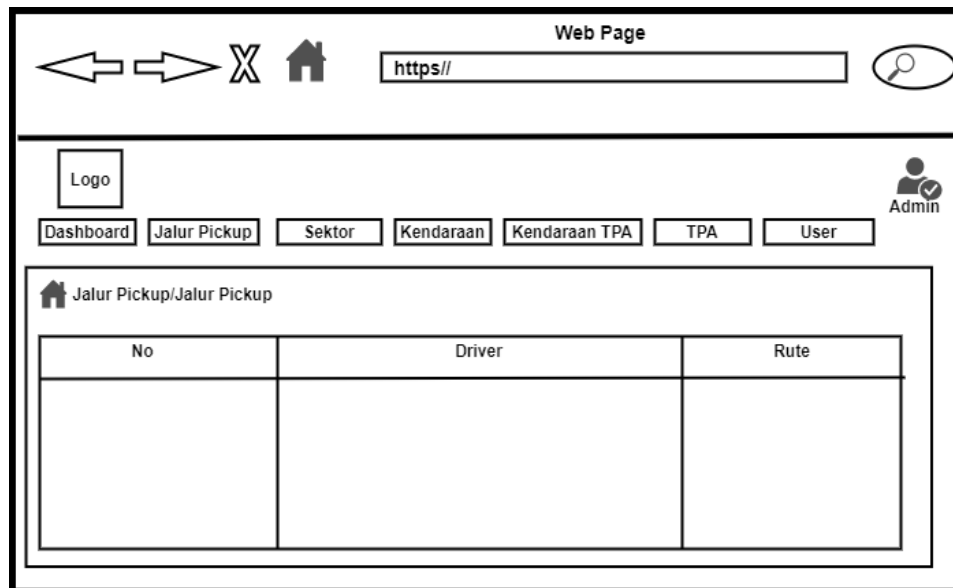
Halaman ini merupakan tampilan informasi *user*, dapat di akses oleh admin yang memberikan informasi tentang data-data pengguna dimulai dari *username*, password, level, beserta aksi tambah, edit dan *delete*.



**Gambar 3.11** Halaman Utama User

## 3. Halaman Jalur Pickup

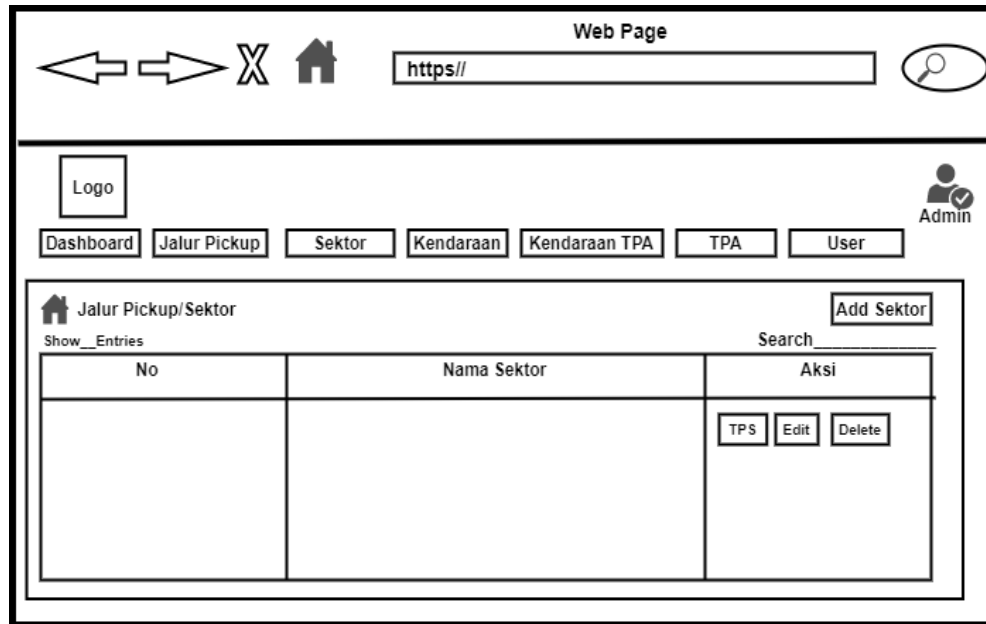
Halaman ini merupakan tampilan informasi jalur pickup. Pada halaman terdapat tampilan rute *driver*, detail perhitungan urutan rute titik awal pengambilan dan titik akhir pembuangan.



**Gambar 3.12** Halaman Jalur Pickup

4. Halaman Sektor

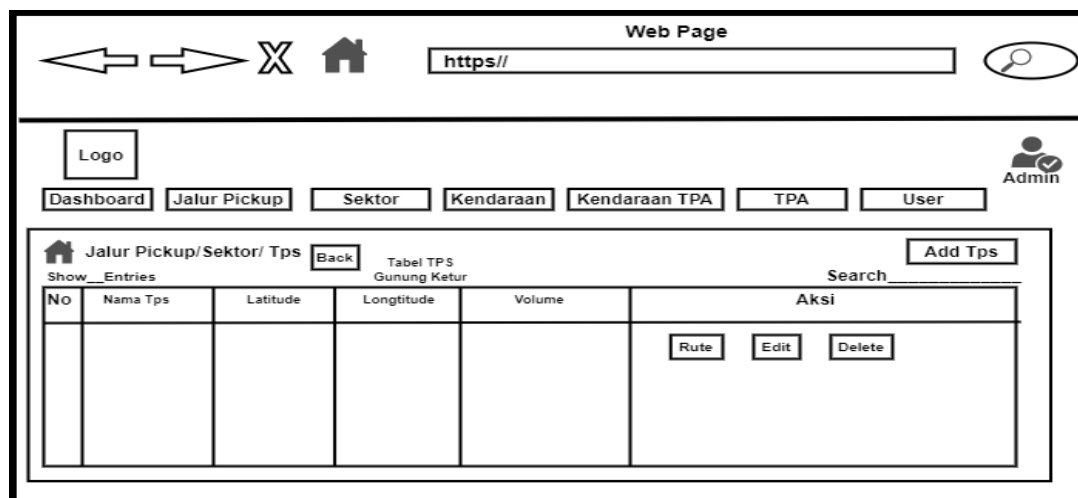
Halaman ini merupakan tampilan nama sektor yang berisi TPS dan rute. Didalam halaman sektor dapat mengedit maupun menambah. Pada halaman ini hanya bisa diakses oleh admin. Pada halaman terdapat *button back* untuk kembali ke menu sebelumnya.



Gambar 3. 13 Halaman Sektor

5. Halaman Tabel TPS

Halaman ini merupakan tampilan urutan nama TPS, *latitude*, *longitude* dan volume. Di halaman ini bisa mengedit maupun menambah. Pada halaman tersebut hanya dapat di akses oleh admin.



Gambar 3. 14 Halaman Tabel TPS

6. Halaman Tabel Rute

Halaman ini merupakan tampilan tabel rute yang di akses oleh admin. Pada halaman dusun terdapat nama TPS, waktu dan aksi untuk edit dan *delete*.

**Gambar 3. 15** Halaman Rute

7. Halaman Add Sektor

Halaman ini merupakan tampilan *add* sektor yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input nama sektor.

**Gambar 3. 16** Halaman Add Sektor

8. Halaman Add TPS

Halaman ini merupakan tampilan daftar TPS yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input sektor, nama TPS, *latitude*, *longitude*, dan volume TPS.

Web Page

https://

Jalur Pickup/Sektor/ Add\_tps

Add TPS

Sektor \_\_\_\_\_

Nama TPS \_\_\_\_\_

Latitude \_\_\_\_\_

Longitude \_\_\_\_\_

Volume \_\_\_\_\_

Cancel Save

**Gambar 3. 17** Halaman Add TPS

9. Halaman Edit TPS

Halaman ini merupakan tampilan edit TPS yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input nama TPS, *latitude*, *longitude* dan volume TPS.

Web Page

https://

Jalur Pickup/Sektor/ Edit\_tps

Edit TPS

Sektor \_\_\_\_\_

Nama TPS \_\_\_\_\_

Latitude \_\_\_\_\_

Longitude \_\_\_\_\_

Volume \_\_\_\_\_

Cancel Save

**Gambar 3. 18** Halaman Edit TPS

#### 10. Halaman Edit Rute

Halaman ini merupakan tampilan edit rute yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input rute asal, rute tujuan dan waktu.

Web Page

https://

Jalur Pickup/Sektor/ Rute\_Edit

Edit Rute

Asal \_\_\_\_\_

Tujuan \_\_\_\_\_

Waktu \_\_\_\_\_

Cancel Save

Gambar 3. 19 Halaman Edit Rute

#### 11. Halaman Kendaraan

Halaman ini merupakan tampilan kendaraan. Pada halaman kendaraan terdapat aksi edit, delete dan button add kendaraan.

Web Page

https://

Jalur Pickup/Kendaraan

Logo

Dashboard Jalur Pickup Sektor Kendaraan Kendaraan TPA TPA User Admin Aksi

Show Entries Tabel Kendaraan Search

No	Kendaraan	Driver	Plat Nomor	Volume Kendaraan	Aksi
					Edit Delete

Add Kendaraan

Gambar 3. 20 Halaman Kendaraan

## 12. Halaman Add Kendaraan

Halaman ini merupakan tampilan *add* kendaraan yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input nama kendaraan, *driver*, plat nomor dan volume kendaraan.

The screenshot shows a web browser window with the title 'Web Page'. The address bar contains 'https://'. The page content is as follows:

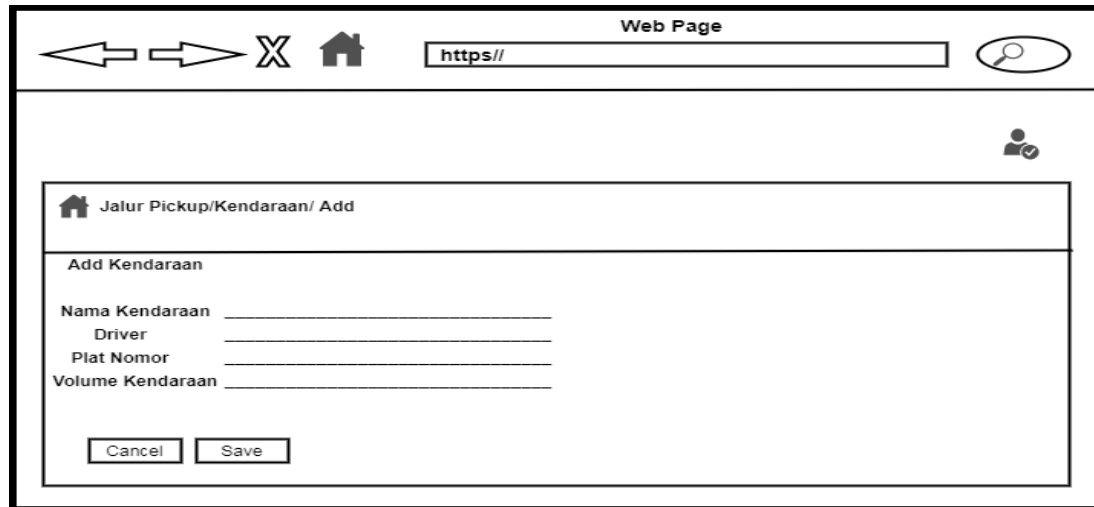
- Breadcrumb: Jalur Pickup/Kendaraan/ Add
- Title: Add Kendaraan
- Form fields:
  - Nama Kendaraan \_\_\_\_\_
  - Driver \_\_\_\_\_
  - Plat Nomor \_\_\_\_\_
  - Volume Kendaraan \_\_\_\_\_
- Buttons:

**Gambar 3. 21** Halaman Add kendaraan



### 13. Halaman Edit kendaraan

Halaman ini merupakan tampilan *add* kendaraan yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input nama kendaraan, *driver*, plat nomor dan volume kendaraan.



Web Page

https://

Jalur Pickup/Kendaraan/ Add

Add Kendaraan

Nama Kendaraan \_\_\_\_\_

Driver \_\_\_\_\_

Plat Nomor \_\_\_\_\_

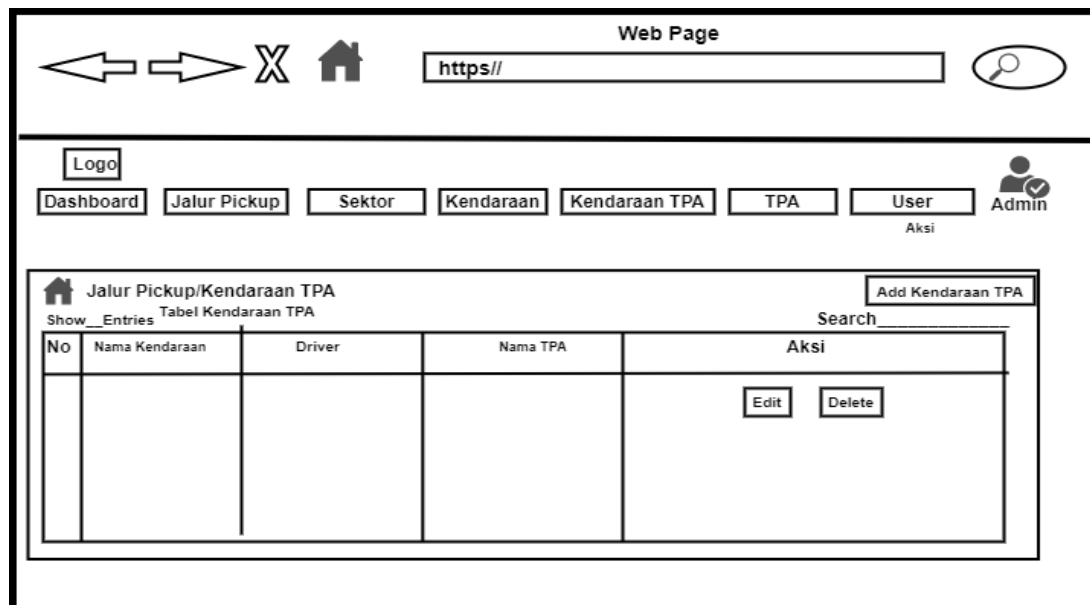
Volume Kendaraan \_\_\_\_\_

Cancel Save

**Gambar 3. 22** Halaman Edit Kendaraan

### 14. Halaman Kendaraan TPA

Halaman ini merupakan tampilan kendaraan TPA. Pada halaman kendaraan TPA terdapat aksi edit, *delete* dan *button add* kendaraan TPA.



Web Page

https://

Jalur Pickup/Kendaraan TPA

Logo

Dashboard Jalur Pickup Sektor Kendaraan Kendaraan TPA TPA User Admin

Aksi

Show Entries Tabel Kendaraan TPA Search

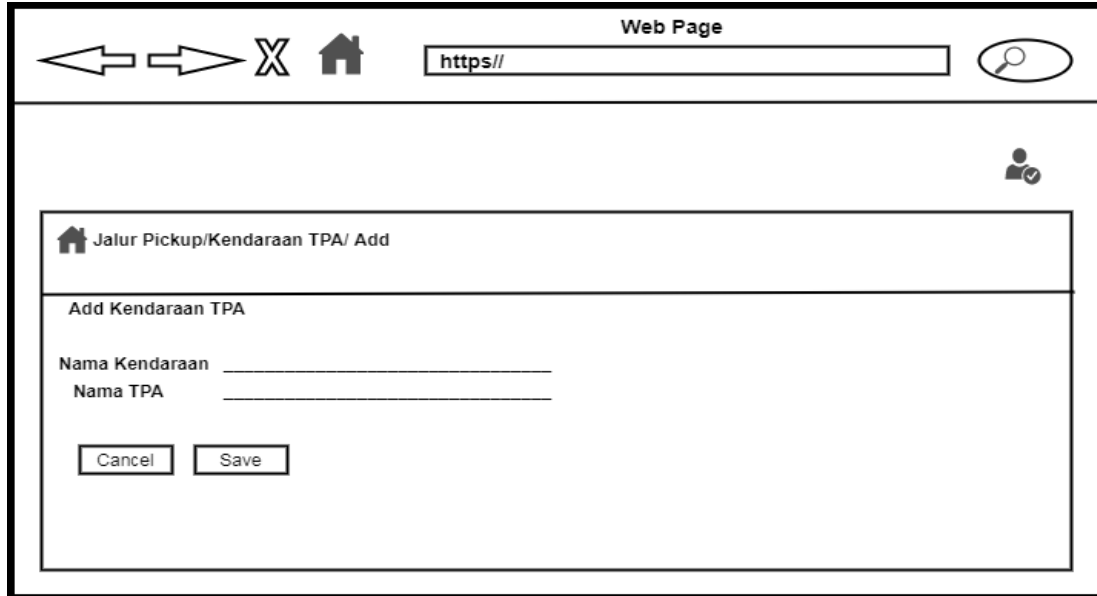
No	Nama Kendaraan	Driver	Nama TPA	Aksi
				Edit Delete

Add Kendaraan TPA

**Gambar 3. 23** Halaman Kendaraan TPA

#### 15. Halaman Add Kendaraan TPA

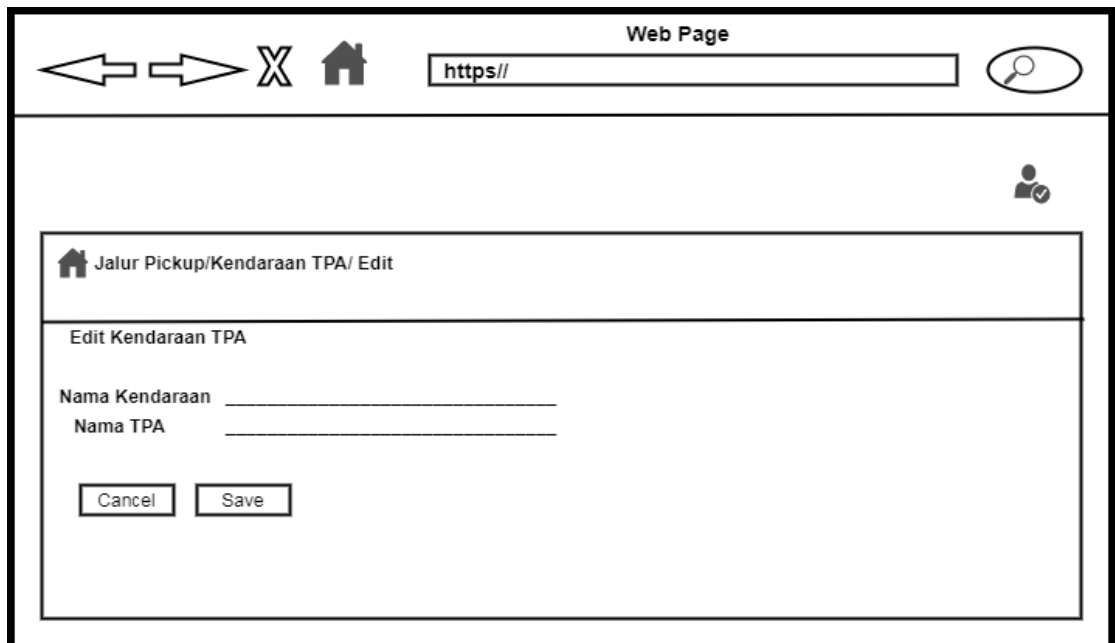
Halaman ini merupakan tampilan add kendaraan TPA yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi input nama kendaraan, dan nama TPA.



**Gambar 3. 24** Halaman Add Kendaraan TPA

#### 16. Halaman Edit Kendaraan TPA

Halaman ini merupakan tampilan edit kendaraan TPA yang hanya dapat di akses oleh admin. Pada halaman ini terdapat aksi edit nama kendaraan, dan nama TPA.



**Gambar 3. 25** Halaman Edit Kendaraan TPA

### 17. Halaman TPA

Halaman ini merupakan tampilan TPA. Pada halaman TPA terdapat aksi *edit*, *delete* dan *button add* TPA.

The screenshot shows a web browser window titled "Web Page" with a URL bar containing "https://". The page has a navigation bar with a "Logo" button and a menu with items: "Dashboard", "Jalur Pickup", "Sektor", "Kendaraan", "Kendaraan TPA", "TPA", "User", and an "Admin" user icon. The main content area is titled "Jalur Pickup/TPA" and includes a "Show Entries" button and a "Tabel TPA" label. Below this is a table with the following structure:

No	Nama TPA	Longitude	Latitude	Aksi
				<div>Edit Delete</div>

There is also a "Search" input field and an "Add TPA" button in the top right corner of the main content area.

Gambar 3. 26 Halaman TPA

### 18. Halaman Add TPA

Halaman ini merupakan tampilan add TPA. Pada halaman TPA terdapat aksi input nama TPA, *longitude*, dan *latitude*.

The screenshot shows a web browser window titled "Web Page" with a URL bar containing "https://". The page has a navigation bar with a "Logo" button and a menu with items: "Dashboard", "Jalur Pickup", "Sektor", "Kendaraan", "Kendaraan TPA", "TPA", "User", and an "Admin" user icon. The main content area is titled "Jalur Pickup/TPA/ Add" and includes a "Add TPA" label. Below this is a form with the following fields:

Nama TPA \_\_\_\_\_  
Longitude \_\_\_\_\_  
Longitude \_\_\_\_\_

There are "Cancel" and "Save" buttons at the bottom of the form.

Gambar 3. 27 Halaman Add TPA

#### 19. Halaman Edit TPA

Halaman ini merupakan tampilan *edit* TPA. Pada halaman TPA terdapat aksi input nama TPA, *longitude*, dan *latitude*.

Web Page

https://

Jalur Pickup/TPA/ Edt

Edit TPA

Nama TPA \_\_\_\_\_

Longitude \_\_\_\_\_

Longitude \_\_\_\_\_

Cancel Save

Gambar 3. 28 Halaman Edit TPA

#### 20. Halaman User

Halaman ini merupakan tampilan halaman *user*. Pada halaman *user* terdapat aksi *edit*, *delete* dan *button add user*.

Web Page

https://

Logo

Dashboard Jalur Pickup Sektor Kendaraan Kendaraan TPA TPA User Admin

Jalur Pickup/User

Show\_Entries Data User Search

No	Nama	Username	Aksi
			Edit Delete

Add User

Gambar 3. 29 Halaman User

## 21. Halaman Add User

Halaman ini merupakan tampilan Add User. Pada halaman pengungsian terdapat aksi input Nama user, Username, password, dan memilih level.

Web Page

https://

Jalur Pickup/User/Add

Add User

Nama \_\_\_\_\_

Username \_\_\_\_\_

Password \_\_\_\_\_

Level \_\_\_\_\_

Cancel Save

**Gambar 3. 30** Halaman Add User

## 22. Halaman Edit User

Halaman ini merupakan tampilan Edit User. Pada halaman pengungsian terdapat aksi input Nama user, Username, password, dan memilih level user.

Web Page

https://

Jalur Pickup/User/Edit

Edit User

Nama \_\_\_\_\_

Username \_\_\_\_\_

Password \_\_\_\_\_

Level ☐ Admin ☐ Driver

Cancel Save

**Gambar 3. 31** Halaman Edit User

### 2.3 Halaman Detail Perhitungan

Halaman ini merupakan tampilan detail perhitungan rute dari titik awal sampai ke titik akhir dan hanya admin yang bisa mengakses

Web Page

https://

Logo

Admin

Dashboard Jalur Pickup Sektor Kendaraan Kendaraan TPA TPA User

Jalur Pickup/Jalur Pick up/Perhitungan  
Detail Perhitungan Driver/Jenis Kendaraan

Show\_\_Entries Rute Titik Awal ke Titik Akhir Search

No	Nama TPS	Waktu
----	----------	-------

**Gambar 3. 32** Halaman Detail Perhitungan

### 3.12 Perancangan Pengajuan Sistem

Pada penelitian ini pengujian yang dilakukan merupakan pengujian *black box*. Pengujian *black box* juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performasi, kesalahan inisialisasi dan akhir program (Pressman, 2010). Maka dengan demikian pengujian *black box* menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Bisa dikatakan bahwa kasus uji yang dibuat untuk melakukan pengujian *black box* testing harus dibuat dengan kasus benar dan kasus salah. Karena tujuan dari metode pengujian *black box* adalah mendapatkan kesalahan sebanyak-banyaknya. Setelah memperoleh hasil ujinya, kesalahan atau hasil yang tidak sesuai dengan yang diinginkan ataupun ketidaksesuaian tersebut dicatat untuk selanjutnya diperbaiki. Adapun pengujian yang dilakukan yaitu:

#### 1. Pengujian *Alpha test*

Pengujian *alpha test* dilakukan pada sisi pengembang oleh seorang pengguna. Pengujian *alpha* dilakukan pada sebuah lingkungan yang terkontrol dan pengoperasiannya diawasi oleh pengembang. Pengujian *alpha* biasanya dilakukan saat pelatihan atau *training* aplikasi. Adapun tabel pengujian *alpha test* untuk Admin Aplikasi dapat dilihat pada tabel 3.25.

**Tabel 3. 26** Tabel Rancangan *Alpha Test* Pada Admin

No	Detail Item	Data Pengujian	Penilaian				
			SB	B	CB	KB	SKB
1	Halaman Dashboard	Menampilkan halaman <i>home</i> .					
2	Halaman Jalur Pickup	Menampilkan data jalur evakuasi					
		Menampilkan data detail rute setiap driver					
		Menampilkan data detail perhitungan					
		Menampilkan data perhitungan setiap driver					
3	Halaman Sektor	Menampilkan data dusun					
		Menampilkan data titik kumpul					
		Keberhasilan proses tambah data					
		Keberhasilan proses <i>edit</i> data					
4	Halaman Kendaraan	Keberhasilan proses hapus data					
		Menampilkan data kendaraan					
		Keberhasilan proses tambah data					
		Keberhasilan proses <i>edit</i> data					
5	Halaman Kendaran TPA	Keberhasilan proses hapus data					
		Menampilkan data Kendaraan Evakuasi					
		Keberhasilan proses tambah data					
		Keberhasilan proses Edit data					
6	Halaman TPA	Keberhasilan proses hapus data					
		Menampilkan data Pengungsian					
		Keberhasilan proses tambah data					
		Keberhasilan proses Edit data					
7	Halaman User	Keberhasilan proses hapus data					
		Keberhasilan proses Edit data					
		Keberhasilan proses tambah data					
		Menampilkan halaman User					
9	<i>Logout</i>	Keberhasilan <i>logout</i> atau menutup aplikasi					

Bobot penilaian untuk pengujian *alpha test* menggunakan 5 skala penilaian, skala penilaian yang digunakan sesuai dengan tabel 3.26.

**Tabel 3. 27** Bobot Penilaian Pengujian Alpha

No	Skala	Bobot	Keterangan
1	Sangat Baik	5	SB
2	Baik	4	B
3	Cukup Baik	3	CB
4	Kurang Baik	2	KB
5	Sangat Kurang Baik	1	SKB



## 2. Pengujian *Beta Test*

Pengujian *beta* adalah pengujian yang dilakukan oleh lingkungan pengguna, dimana lingkungan perangkat lunak tidak lagi dapat dikendalikan oleh pengembang. Pengujian *beta* merupakan pengujian yang dilakukan secara objektif secara langsung mengenai tanggapan pengguna terhadap aplikasi penerapan *Algoritma Nearest Neighbor* untuk optimasi rute pengambilan dan pembuangan sampah menggunakan teknologi Google Maps berbasis web. Pada pengujian *beta test* ini ada 10 item uji. Daftar item yang akan diuji sesuai dengan tabel 3.27.

**Tabel 3. 28** Pengujian *Beta Test*

No	Item Uji	Penilaian				
		SS	S	CS	KS	STS
1	Aplikasi memiliki tampilan yang mudah digunakan					
2	Aplikasi mudah dioperasikan oleh <i>user</i> (pengguna)					
3	Aplikasi sesuai dengan kebutuhan user (pengguna)					
4	Tampilan aplikasi sesuai dengan kebutuhan user					
5	Halaman Jalur Pickup mudah dimengerti					
6	Halaman data Sektor mudah dimengerti					
7	Halaman data titik TPS mudah dimengerti					
8	Halaman data kendaraan mudah dimengerti					
9	Halaman data TPA mudah dimengerti					
10	Hasil penentuan rute Pengambilan dan Pembuangan Sampah mudah dimengerti					

Bobot penilaian untuk pengujian *beta test* menggunakan 5 skala penilaian, skala penilaian yang digunakan sesuai dengan tabel 3.28.

**Tabel 3. 29** Bobot Penilaian Pengujian Beta

No	Skala	Bobot	Keterangan
1	Sangat Setuju	5	SS
2	Setuju	4	S
3	Cukup Setuju	3	CS
4	Kurang Setuju	2	KS
5	Sangat Kurang Setuju	1	SKS

## BAB IV HASIL, PENGUJIAN, DAN PEMBAHASAN

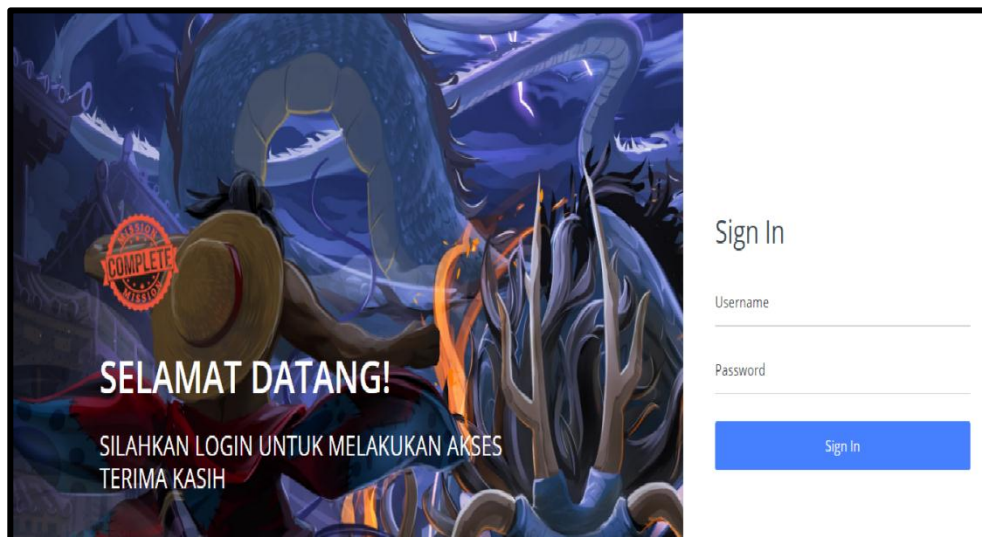
Hasil, pengujian, dan pembahasan merupakan tahap dimana sistem siap dioperasikan pada keadaan sebenarnya. Pada pembahasan ini akan diketahui apakah sistem yang dibuat sudah sesuai dengan analisis dan perancangan yang telah dilakukan sebelumnya. Sehingga bisa mengetahui apakah sistem yang sudah dibuat sudah benar-benar dapat menghasilkan tujuan yang ada pada penelitian ini.

### 4.1 Hasil Penelitian

Hasil penelitian berisikan penerapan Metode *Nearest Neighbor* dalam penentuan jalur pengambilan sampah di Kota Yogyakarta. Pada bagian ini menyajikan tampilan aplikasi dan modul program dari aplikasi penentuan jalur pengambilan tersebut. Adapun modul program dan tampilan aplikasi sebagai berikut:

#### 4.1.1 Tampilan Halaman Login

Halaman *login* admin dan *driver* pada aplikasi pengambilan dan pembuangan sampah dapat dilihat pada gambar 4.1 di bawah ini.



**Gambar 4. 1** Tampilan Halaman Login

Tampilan halaman login mempunyai potongan *source code* dengan modul program 4.1

```
<?php
session_start();
error_reporting(0);
if ($_SESSION['sk2057']) {
header('location:module.php?module=home');
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<div class="auth-side-form">
<div class=" auth-content">

<h3 class="mb-4 f-w-400">Sign In</h3>
<form action="cek_login.php" method="POST">
<div class="form-group mb-3">
<label class="floating-label" for="username">Username</label>
<input type="text" class="form-control" id="username" placeholder="" name="username">
</div>
<div class="form-group mb-4">
<label class="floating-label" for="Password">Password</label>
<input type="password" class="form-control" id="Password" placeholder="" name="password">
</div>

```

**Modul Program 4. 1** *Source Code* Tampilan Halaman Login

#### 4.1.2 Tampilan Halaman Jalur Pickup

Tampilan halaman jalur pickup tampilan halaman ini bisa dilihat pada gambar 4.2 dibawah ini.

NO	DRIVER	RUTE
1	HARSONO WIDAGDO	PIYUNGAN => SMA 3 => JL PRAMUKA => JEMBATAN TUNGKAK => RS PRATAMA PIYUNGAN => SMP 1 => SAGAN => KRICAK => RW 4 BENER PIYUNGAN

**Gambar 4. 2** Tampilan Halaman Jalur Pickup

Tampilan halaman jalur pickup mempunyai potongan *source code* seperti modul program 4.3.

```
<?php
// if (isset($_SESSION['sk2057']['level'])) {
if ($_SESSION['sk2057']['level'] == 'admin') {
?>
<div class="row">
<!-- Zero config table start -->
<div class="col-sm-12">
<div class="card">
<div class="card-header">
<h5>Jalur Pickup Sampah</h5>
</div>
```

#### Modul Program 4.2 Source Code Tampilan Jalur Pickup

```
<div class="card-body">
<div class="row align-items-center m-1-0">
<div class="col-sm-6">
</div> </div>
<div class="dt-responsive table-responsive">
<table id="simpletable3" class="table table-striped table-bordered wrap">
<thead> <tr>
<th>No</th>
<th>Driver</th>
<th>Rute</th>
<!-- <th>Aksi</th> -->
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach ($rute_perkendaraan as $key => $value) {
?>
<tr>
<td><?=$no++ ?></td>
<td><?=$driver[$key]['nama'] ?></td>
<td>
<?php
// print_r($value);
$temp = explode(" => ", $value);
$maks_id = max(array_keys($temp));
$hasil = array();
foreach ($temp as $key2 => $value2) {
$temp2 = explode(".", $value2);
// print_r($temp2);
if ($temp2[0] == 'tps') {
array_push($hasil, $tps[$temp2[1]]['nama_tps'] . ' <button class="btn btn-
danger btn-sm mb-1"> <i class="fa fa-sort-up"> </i> </button>');
}elseif ($temp2[0] == 'tpa') {
// array_push($hasil, "<br>");
if ($key2 == 0) {
array_push($hasil, $tpa[$temp2[1]]['nama_tpa'] . ' <button class="btn btn-
```

```

success btn-sm mb-1"> <i class="fa fa-sort-down"> </i> </button> <br>' );
}else{
array_push($hasil, "<br>" . $tpa[$temp2[1]]['nama_tpa'] . ' <button
class="btn btn-success btn-sm mb-1"> <i class="fa fa-sort-down"> </i>
</button> <br>' );

```

### Modul Program 4.3 Lanjutan Source Code Tampilan Jalur Pickup

#### 4.1.3 Tampilan Halaman Detail Perhitungan

Tampilan halaman detail perhitungan in merupakan detail rute mana yang dipilih oleh setiap *driver* yang ditentukan berdasarkan waktu yang terpendek. Tampilan halaman tersebut dapat dilihat pada gambar 4.3 dibawah ini.

NO	NAMA TPS	WAKTU
1	SMA 3	006
2	JL SINGSINGMANGARAJA	973
3	JL VETERAN	1116
4	Jl RAJAH	1764
5	LP WIROGUNAN	1294

Gambar 4.3 Tampilan Halaman Detail Perhitungan

Tampilan halaman detail perhitungan mempunyai potongan *source code* seperti modul program 4.5

```

<h5>Detail Perhitungan</h5>
<br>
<h5><?= $r['nama_tps'] ?></h5>
</div>
<div class="card-body">
<?php
?>
<div class="col-sm-12">
<ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
<?php
foreach ($kendaraan as $key => $value) {
?>
<li cl?>
<div class="col-sm-12">
<ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
<?php
foreach ($kendaraan as $key => $value) {
?>ass="nav-item">
<a class="nav-link <?= $key == 1 ? 'active' : null ?>" id="kendaraan<?=

```

```

$key ?>-tab" data-toggle="pill" href="#kendaraan<?= $key ?>" role="tab"
aria-controls="kendaraan<?= $key ?>" aria-selected="true"><?=
$value['driver'] . " / " . $value['nama_kendaraan'] ?></a>
</li>
<?php
}
?>
</ul>
<div class="tab-content" id="pills-tabContent">
<?php
$key_awal = 1;
foreach ($kendaraan as $key => $value_kendaraan) {
?>

```

#### Modul Program 4.4 Source code Tampilan Halaman Detail Perhitungan

```

<div class="tab-pane fade <?= $key == 1 ? 'show active' : null ?>"
id="kendaraan<?= $key ?>" role="tabpanel" aria-labelledby="kendaraan<?=
$key ?>-tab">
<?php
$gabung = '';
$start_table = array();
$number = 1;
unset($start_posisi);
unset($end_posisi);
foreach ($map_rute_perkendaraan[$key] as $key2 => $value2) {
$temp_detail_rute_tps = $arr_sisa_volume;
$temp_detail_rute_ke_tpa = $arr_sisa_volume2;
$temp_detail_rute_dari_tpa = $arr_sisa_volume3;
if ($gabung == '') {
$gabung = $start_posisi[1];
if ($key > $key_awal) {
$key_awal = $key;
$gabung = '';
}}
if (!(isset($start_posisi))) {
$start_posisi = $value2;
}else{
$end_posisi = $value2;
if ((isset($end_posisi['sisa']))) {
$gabung = $gabung . " => " . $end_posisi[1];
}else{
$gabung = $gabung . " => " . $end_posisi[1];
echo '<h4>' . $number++ . '. ' . $gabung . '</h4>' ;
echo "<br>";
array_push($start_table, $end_posisi);
$terpilih = 0;
$drop = 0;
for ($i=0; $i < (count($start_table) - 1); $i++) {
if ((isset($start_table[$i]['sisa']))) {
if ((isset($start_table[$i+1]['sisa']))) {
if (isset($start_posisi['kosong'])) {
foreach ($start_posisi['kosong'] as $key5 => $value5) {
if ((isset($start_table[$i+1])) AND ($key5 == $start_table[$i+1][0])) {
// do nothing
$do_nothing = '';

```

```

}else{
unset($temp_detail_rute_tps[$start_table[$i][0]][$key5]);
}
}
}
$table_detail = $temp_detail_rute_tps[$start_table[$i][0]];
$judul = $tps[$start_table[$i][0]][1];
$tiipe = 'TPS';
$drop = $drop + $start_table[$i]['volume_sampah'];
}else{
$table_detail = $temp_detail_rute_ke_tpa[$start_table[$i][0]];
$judul = $tps[$start_table[$i][0]][1];
$tiipe = 'TPA';
$drop = $drop + $start_table[$i]['volume_sampah'];
}
}else{
if (isset($start_table[$i]['kosong'])) {
foreach ($start_table[$i]['kosong'] as $key5 => $value5) {
unset($temp_detail_rute_dari_tpa[$start_table[$i][0]][$key5]);
}}
$table_detail = $temp_detail_rute_dari_tpa[$start_table[$i][0]];
$judul = $tpa[$start_table[$i][0]][1];
$tiipe = 'TPS';
}echo "<pre>";
echo "</pre>";
?>
<h6 style="color:#4680ff;">Dari <?= $judul ?> ke : </h6>
<table id="simpletable<?= $no_table++ ?>" class="table table-striped table-bordered nowrap">
<thead><tr>
<th>No</th>
<th>Nama <?= $tiipe ?></th>
<th>Waktu</th>
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach ($table_detail as $key3 => $value3){
?>
<tr>
<td><?= $no++ ?></td>
<td><?= $tiipe == 'TPS' ? $tps[$key3][1] : $tpa[$key3][1] ?></td>
<td><?= $value3 ?></td>
</tr>
<?php}
?>
</tbody>
</table>
<?php
($tiipe == 'TPS') {
echo "
Yang terpilih adalah ". $start_table[$i+1][1] . " membawa ".
$start_table[$i+1]['volume_sampah'] . " dari " .
($start_table[$i+1]['sisa'] + $start_table[$i+1]['volume_sampah']) . " ( ".
$start_table[$i+1]['volume'] . " )"

```

```

;
$terpilih = 0;
}
elseif ($tipe == 'TPA') {
echo "
Yang terpilih adalah ". $end_posisi['nama_tpa'] . " diturunkan ". $drop;
;
$terpilih = 1; }
}
echo "<hr>";
$start_table = array();
$gabung = '';
}
$start_posisi = $end_posisi;
}
array_push($start_table, $start_posisi);
}
?>

```

#### Modul Program 4. 5 Lanjutan *Source Code* Tampilan Halaman Detail Perhitungan

##### 4.1.4 Tampilan Halaman Sektor

Tampilan halaman sektor merupakan isi dari data sektor dan data Tps yang nantinya sebagai titik pengambilan *driver*. Tampilan halaman tersebut dapat dilihat pada gambar 4.4 dibawah ini.

NO	NAMA SEKTOR	AKSI
1	GUNUNG KETUR	TPS Edit Delete
2	KRASAK	TPS Edit Delete
3	MALIOBORO - KRANGGAN	TPS Edit Delete
4	NGASEM GADING	TPS Edit Delete

**Gambar 4. 4** Tampilan Halaman Sektor

Tampilan halaman sektor mempunyai potongan *source code* seperti dengan modul program 4.7.

```

<div class="card-header">
<h5>Tabel Sektor</h5>
</div>
<div class="card-body">
<div class="row align-items-center m-1-0">
<div class="col-sm-6">

```



```

</div>
<div class="col-sm-6 text-right">
<button class="btn btn-success btn-sm btn-round has-ripple"
onclick="location.href = '?module=sektor&act=add';" ><i class="feather
icon-plus"></i> Add Sektor</button>
</div>
</div>
<div class="dt-responsive table-responsive">
<table id="simpletable" class="table table-striped table-bordered nowrap">
<thead>
<tr>
<th>No</th>
<th>Nama Sektor</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>

```

### Modul Program 4. 6 Source Code Tampilan Halaman Sektor

```

<?php
$stmtpil = mysqli_query($koneksi, "SELECT * FROM sektor ORDER BY
nama_sektor ASC");
$no = 1;
while ($r = mysqli_fetch_array($stmtpil)) {
    ?>
    <tr>
    <td><?= $no++ ?></td>
    <td><?= $r['nama_sektor'] ?></td>
    <td>
    <a href="module.php?module=sektor&act=tps&id=<?php echo $r['id_sektor'];
    ?>" class="btn btn-primary btn-sm"><i class="feather icon-
    book"></i>&nbsp;TPS </a>
    <a href="module.php?module=sektor&act=edit&id=<?php echo $r['id_sektor'];
    ?>" class="btn btn-info btn-sm"><i class="feather icon-edit"></i>&nbsp;Edit
    </a>
    <a href="<?php echo $aksi; ?>module=sektor&act=delete&id=<?php echo
    $r['id_sektor']; ?>" onclick="return confirm('Apa Anda yakin menghapus <?=
    $r[nama_sektor] ?>?')" class="btn btn-danger btn-sm"><i class="feather
    icon-trash-2"></i>&nbsp;Delete </a>
    </td>
    </tr>

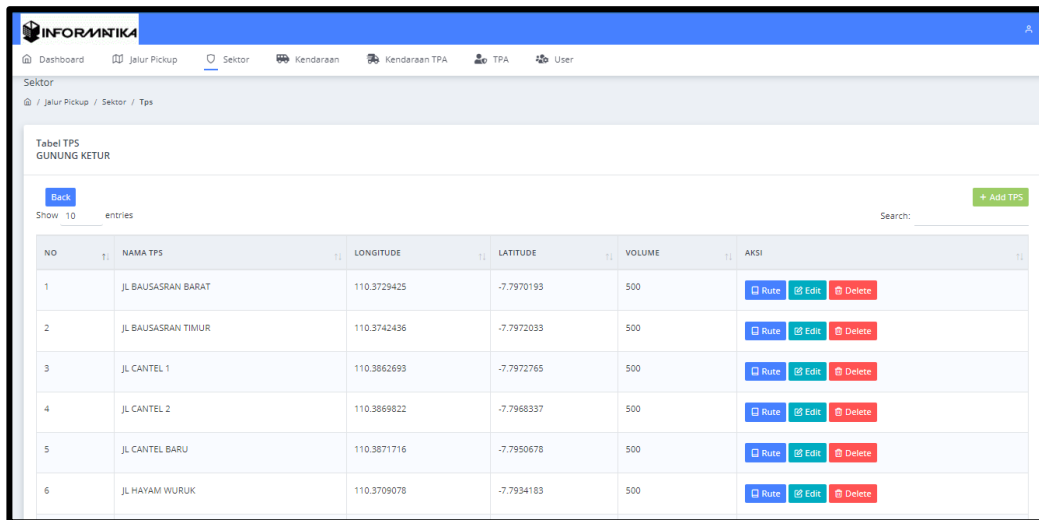
```

---

**Modul Program 4.7** Lanjutan *Source Code* Tampilan Halaman Sektor

#### 4.1.5 Tampilan Halaman TPS

Tampilan halaman Tps merupakan halaman data titik Tps. Tampilan halaman Tps dapat dilihat pada gambar 4.5 dibawah ini.



The screenshot shows a web application interface for 'INFORMATIKA'. The main content area displays a table titled 'Tabel TPS GUNUNG KETUR'. The table has columns for NO, NAMA TPS, LONGITUDE, LATITUDE, VOLUME, and AKSI. There are 6 rows of data. Each row has three action buttons: 'Route', 'Edit', and 'Delete'.

NO	NAMA TPS	LONGITUDE	LATITUDE	VOLUME	AKSI
1	JL BAUSASRAN BARAT	110.3729425	-7.7970193	500	<a href="#">Route</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	JL BAUSASRAN TIMUR	110.3742436	-7.7972033	500	<a href="#">Route</a> <a href="#">Edit</a> <a href="#">Delete</a>
3	JL CANTEL 1	110.3862693	-7.7972765	500	<a href="#">Route</a> <a href="#">Edit</a> <a href="#">Delete</a>
4	JL CANTEL 2	110.3869822	-7.7968337	500	<a href="#">Route</a> <a href="#">Edit</a> <a href="#">Delete</a>
5	JL CANTEL BARU	110.3871716	-7.7950678	500	<a href="#">Route</a> <a href="#">Edit</a> <a href="#">Delete</a>
6	JL HAYAM WURUK	110.3709078	-7.7934183	500	<a href="#">Route</a> <a href="#">Edit</a> <a href="#">Delete</a>

Gambar 4. 5 Tampilan Halaman TPS

Tampilan halaman Tps mempunyai potongan *source code* seperti dengan modul program 4.8.

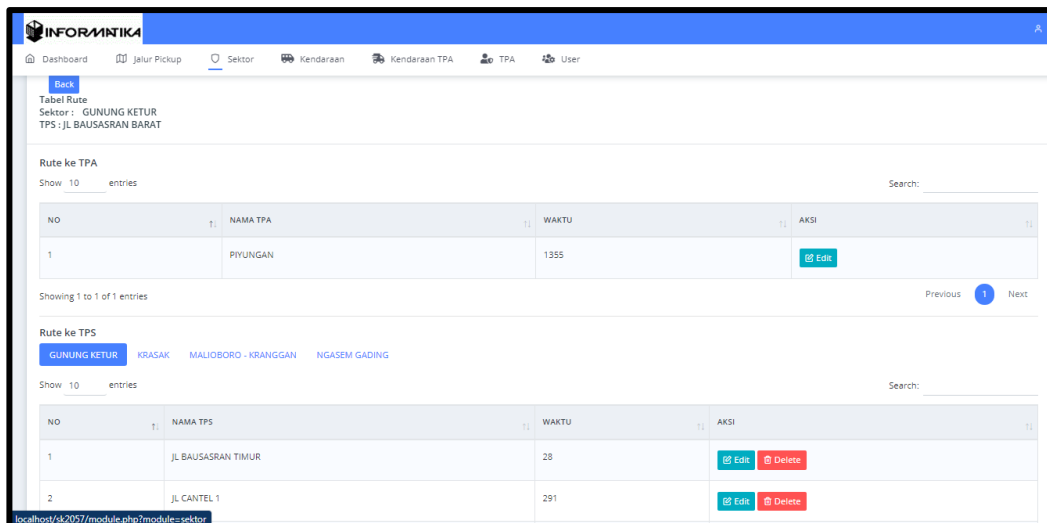
```
<div class="row">
<!-- Zero config table start -->
<div class="col-sm-12">
<div class="card">
<div class="card-header">
<h5>Tabel TPS </h5>
<br>
<h5><? = $r['nama_sektor'] ?></h5>
</div>
<div class="card-body">
<div class="row align-items-center m-l-0">
<div class="col-sm-6">
<button class="btn btn-primary btn-sm btn-round has-ripple"
onclick="location.href = '?module=sektor';" ><i class="feather icon-back"></i> Back</button>
</div>
<div class="col-sm-6 text-right">
<button class="btn btn-success btn-sm btn-round has-ripple"
onclick="location.href = '?module=sektor&act=add_tps&id_sektor=<?php echo $ _GET['id']; ?>';" ><i class="feather icon-plus"></i> Add TPS</button>
```

```
</div>
</div>
```

## Modul Program 4. 8 Source Code Tampilan Halaman TPS

### 4.1.6 Tampilan Halaman Rute

Tampilan halaman rute merupakan tampilan rute yang digunakan oleh *driver* yang ada pada aplikasi tersebut. Tampilan halaman rute ini bisa dilihat pada gambar 4.6 dibawah ini.



Gambar 4. 6 Tampilan Halaman Rute

Tampilan halaman rute mempunyai potongan *source code* seperti dengan modul program 4.10.

```
<div class="row">
<!-- Zero config table start -->
<div class="col-sm-12">
<div class="card">
<div class="card-header">
<div class="row align-items-center m-l-0">
<div class="col-sm-6">
<button class="btn btn-primary btn-sm btn-round has-ripple"
onclick="location.href = '?module=sektor&act=tps&id=?= $id_sektor ?>';"
><i class="feather icon-back"></i> Back</button>
</div>
</div>
```

## Modul Program 4. 9 Source Code Tampilan Halaman Rute

```
<h5>Tabel Rute </h5>
<br>
<h5>Sektor : </h5>
<h5><?= $r_detail['nama_sektor'] ?></h5>
<br>
<h5>TPS : <?= $r_detail['nama_tps'] ?></h5>
</div>
```

```

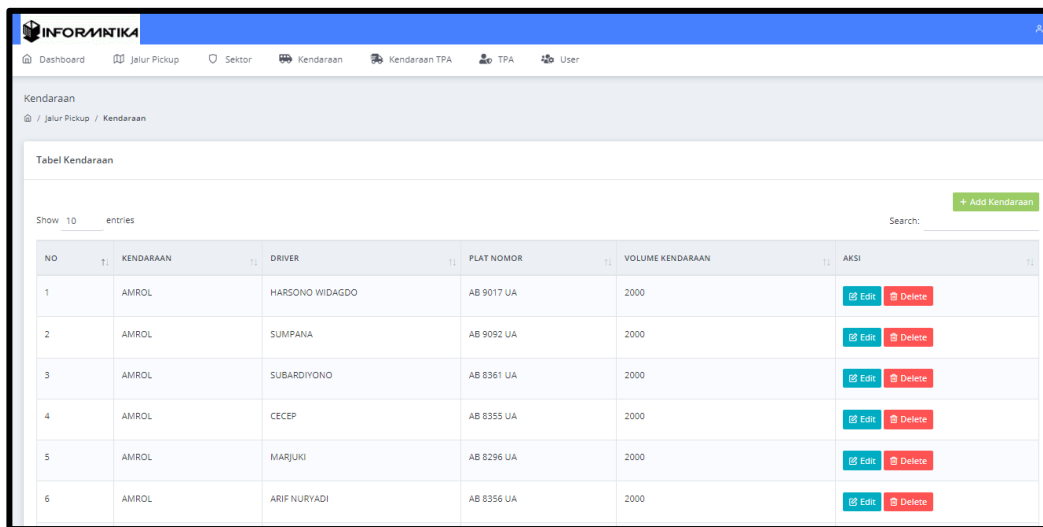
<div class="card-body">
<h5>Rute ke TPA</h5>
<div class="dt-responsive table-responsive">
<table id="simpletable2" class="table table-striped table-bordered nowrap">
<thead>
<tr>
<th>No</th>
<th>Nama TPA</th>
<th>Waktu</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php
$stampil = mysqli_query($koneksi, "SELECT * FROM tpa ORDER BY nama_tpa
ASC");
$no = 1;
while ($r = mysqli_fetch_array($stampil)) {
$stampil2 = mysqli_query($koneksi, "SELECT * FROM tpa t left join
rute_ke_tpa r on t.id_tpa=r.id_titik_akhir WHERE id_tpa='$r[id_tpa]' AND
id_titik_awal='$_GET[id]' ORDER BY nama_tpa ASC");
$r2 = mysqli_fetch_array($stampil2)
?>
<tr>
<td><?= $no++ ?></td>
<td><?= $r['nama_tpa'] ?></td>
<td><?= (empty($r2['waktu'])) ? 0 : $r2['waktu'] ?></td>
<td>
<a href="module.php?module=sektor&act=rute_edit&id_asal=<?php echo
$_GET['id']; ?>&id_tujuan=<?php echo $r['id_tpa'];
?>&tiptpe=rute_ke_tpa&id_rute=<?php echo $r2['id_rute']; ?>&id_sektor=<?php
echo $id_sektor; ?>" class="btn btn-info btn-sm"><i class="feather icon-
edit"></i>&nbsp;Edit </a>
</td>
</tr>
<?php
}
?>

```

#### Modul Program 4.10 Lanjutan Source Code Tampilan Halaman Rute

##### 4.1.7 Tampilan Halaman Kendaraan

Tampilan halaman kendaraan ini merupakan data kendaraan yang nantinya didaftarkan sebagai kendaraan pengambilan dan pembuangan sampah. Tampilan halaman tersebut dapat dilihat pada gambar 4.7 dibawah ini.



**Gambar 4. 7** Tampilan Halaman Kendaraan

Tampilan halaman kendaraan mempunyai potongan *source code* seperti dengan modul program 4.11.

```
<h5>Tabel Kendaraan</h5>
</div>
<div class="card-body">
<div class="row align-items-center m-l-0">
<div class="col-sm-6">
</div>
<div class="col-sm-6 text-right">
<button class="btn btn-success btn-sm btn-round has-ripple"
onclick="location.href = '?module=kendaraan&act=add';" ><i class="feather
icon-plus"></i> Add Kendaraan</button>
</div>
</div>
<div class="dt-responsive table-responsive">
<table id="simpletable" class="table table-striped table-bordered nowrap">
<thead>
<tr>
<th>No</th>
<th>Kendaraan</th>
<th>Driver</th>
<th>Plat Nomor</th>
<th>Volume Kendaraan</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php
$stampil = mysqli_query($koneksi, "SELECT * FROM kendaraan k JOIN pengguna p
ON k.id_pengguna=p.id_pengguna ORDER BY nama_kendaraan ASC");
$no = 1;
while ($r = mysqli_fetch_array($stampil)) {
?>
```

```

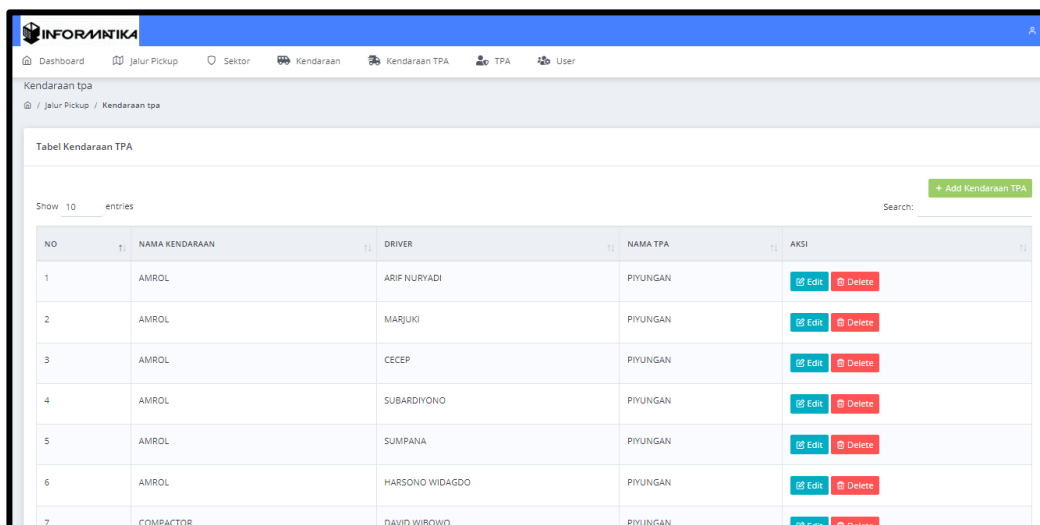
<tr>
<td><?= $no++ ?></td>
<td><?= $r['nama_kendaraan'] ?></td>
<td><?= $r['nama'] ?></td>
<td><?= $r['plat_nomor'] ?></td>
<td><?= $r['volume_kendaraan'] ?></td>
<td>
<a href="module.php?module=kendaraan&act=edit&id=<?php echo
$r['id_kendaraan']; ?>" class="btn btn-info btn-sm"><i class="feather icon-
edit"></i>&nbsp;Edit </a>
<a href="<?php echo $aksi; ?>?module=kendaraan&act=delete&id=<?php echo
$r['id_kendaraan']; ?>" onclick="return confirm('Apa Anda yakin menghapus
<?= $r[nama_kendaraan] ?>?>?>'" class="btn btn-danger btn-sm"><i
class="feather icon-trash-2"></i>&nbsp;Delete </a>
</td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>

```

**Modul Program 4. 11** *Source Code* Tampilan Halaman Kendaraan

#### 4.1.8 Tampilan Halaman Kendaraan TPA

Tampilan halaman kendaraan Tpa merupakan data kendaraan yang sudah didaftarkan sebagai kendaraan pengambilan dan pembuangan sampah. Tampilan halaman tersebut dapat dilihat pada gambar 4.8 dibawah ini.



NO	NAMA KENDARAAN	DRIVER	NAMA TPA	AKSI
1	AMROL	ARIF NURYADI	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>
2	AMROL	MARJUKI	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>
3	AMROL	CECEP	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>
4	AMROL	SUBARDIYONO	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>
5	AMROL	SUMPANA	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>
6	AMROL	HARSONO WIDAGDO	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>
7	COMPACTOR	DAVID WIBOWO	PIYUNGAN	<a href="#">Edit</a> <a href="#">Delete</a>

**Gambar 4. 8** Tampilan Halaman Kendaraan TPA



```

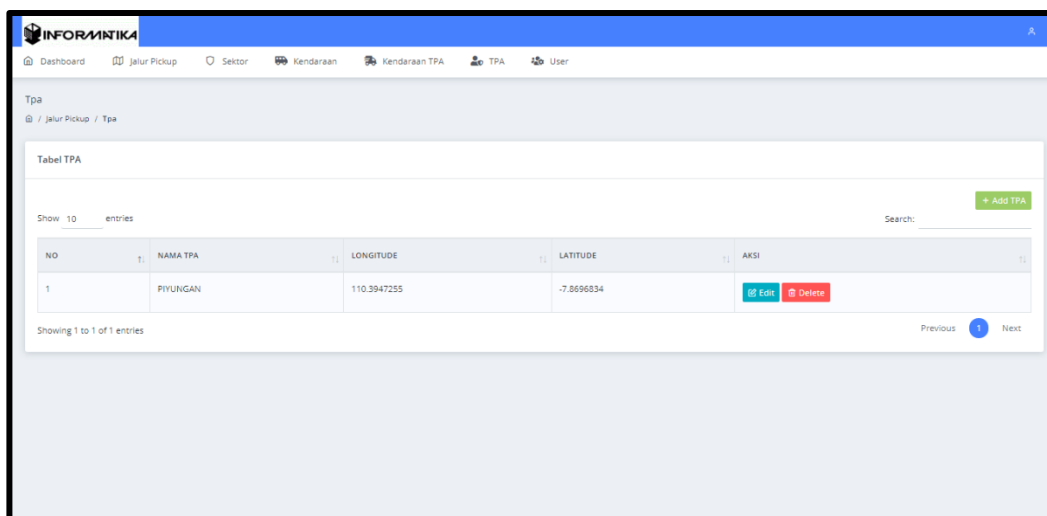
</td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>

```

#### Modul Program 4. 12 Source Code Tampilan Halaman Kendaraan Tpa

##### 4.1.9 Tampilan Halaman TPA

Tampilan halaman Tpa merupakan data Tpa yang nantinya akan digunakan sebagai tujuan akhir. Tampilan halaman tersebut dapat dilihat pada gambar 4.9 di bawah ini.



Gambar 4. 9 Tampilan Halaman TPA

Tampilan halaman Tpa mempunyai potongan *source code* seperti dengan modul program 4.13.

```

<h5>Tabel TPA</h5>
</div>
<div class="card-body">
<div class="row align-items-center m-l-0">
<div class="col-sm-6">
</div>
<div class="col-sm-6 text-right">
<button class="btn btn-success btn-sm btn-round has-ripple"
onclick="location.href = '?module=tpa&act=add';" ><i class="feather icon-

```



```

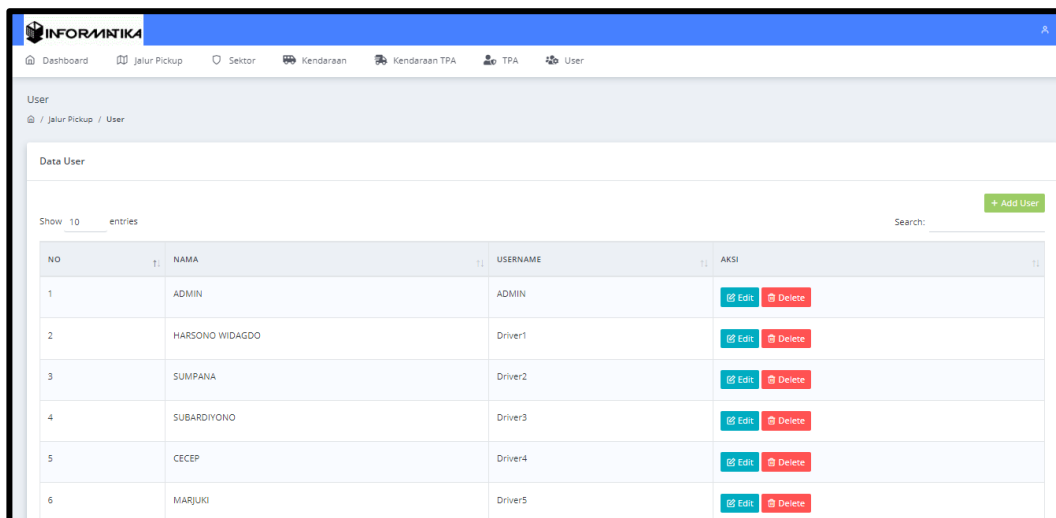
plus"></i> Add TPA</button>
</div>
</div>
<div class="dt-responsive table-responsive">
<table id="simpletable" class="table table-striped table-bordered nowrap">
<thead>
<tr>
<th>No</th>
<th>Nama TPA</th>
<th>Longitude</th>
<th>Latitude</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php
$stampil = mysqli_query($koneksi, "SELECT * FROM tpa ORDER BY nama_tpa
ASC");
$no = 1;
while ($r = mysqli_fetch_array($stampil)) {
?>
<tr>
<td><?= $no++ ?></td>
<td><?= $r['nama_tpa'] ?></td>
<td><?= $r['long_tpa'] ?></td>
<td><?= $r['lat_tpa'] ?></td>
<td>
<a
href="module.php?module=tpa&act=edit&id=<?php echo $r['id_tpa']; ?>"
class="btn btn-info btn-sm"><i class="feather icon-edit"></i>&nbsp;Edit
</a>
<a
href="<?php echo $aksi; ?>?module=tpa&act=delete&id=<?php echo
$r['id_tpa']; ?>" onclick="return confirm('Apa Anda yakin menghapus <?=
$r[nama_tpa] ?>?>?')" class="btn btn-danger btn-sm"><i class="feather icon-
trash-2"></i>&nbsp;Delete </a>
</td>
</tr>
</tbody>
<?php
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>

```

#### Modul Program 4.13 Source Code Tampilan Halaman TPA

#### 4.1.10 Tampilan Halaman User

Tampilan halaman *user* merupakan data pengguna yang dapat mengakses aplikasi. Tampilan halaman tersebut dapat dilihat pada gambar 4.10 di bawah ini.



NO	NAMA	USERNAME	AKSI
1	ADMIN	ADMIN	<a href="#">Edit</a> <a href="#">Delete</a>
2	HARSONO WIDAGDO	Driver1	<a href="#">Edit</a> <a href="#">Delete</a>
3	SUMPANA	Driver2	<a href="#">Edit</a> <a href="#">Delete</a>
4	SUBARDIYONO	Driver3	<a href="#">Edit</a> <a href="#">Delete</a>
5	CECEP	Driver4	<a href="#">Edit</a> <a href="#">Delete</a>
6	MARJUKI	Driver5	<a href="#">Edit</a> <a href="#">Delete</a>

**Gambar 4. 10** Tampilan Halaman User

Tampilan halaman *user* mempunyai potongan *source code* seperti dengan modul program 4.14.

```
<h5>Data User</h5>
</div>
<div class="card-body">
<div class="row align-items-center m-l-0">
<div class="col-sm-6">
</div>
<div class="col-sm-6 text-right">
<button class="btn btn-success btn-sm btn-round has-ripple"
onclick="location.href = '?module=user&act=add';" ><i class="feather icon-
plus"></i> Add User</button>
</div>
</div>
<div class="dt-responsive table-responsive">
<table id="simpletable" class="table table-striped table-bordered no-wrap">
<thead>
<tr>
<th>No</th>
```

```
<th>Nama</th>
<th>Username</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php
include 'koneksi.php';
$no = 1;
$data = mysqli_query($koneksi,"select * from pengguna");
while($r = mysqli_fetch_array($data)){
?>
<tr>
<td><?php echo $no++; ?></td>
<td><?php echo $r['nama']; ?></td>
<td><?php echo $r['username']; ?></td>
echo 'level'
<td>
<a
href="module.php?module=user&act=edit&id=<?php echo $r['id_pengguna']; ?>"
class="btn btn-info btn-sm"><i class="feather icon-edit"></i>&nbsp;Edit
</a>
<a
href="<?php echo $aksi; ?>?module=user&act=delete&id=<?php echo
$r['id_pengguna']; ?>" onclick="return confirm('Apa Anda yakin menghapus
<?= $r[nama] ?>?>?>)" class="btn btn-danger btn-sm"><i class="feather icon-
trash-2"></i>&nbsp;Delete </a>
</td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
```

#### Modul Program 4. 14 Source Code Tampilan Halaman User

## 4.2 Pengujian Sistem

Pengujian pada penelitian ini merupakan pengujian pada *website*. Pengujian *website* ini berfokus pada spesifikasi fungsional dari perangkat lunak. Pada pengujian *black box* akan dilakukan pengamatan hasil dari eksekusi pada *website* tersebut. Tujuan dari pengujian ini yaitu untuk mencari dan mengetahui kesalahan yang terjadi pada aplikasi sehingga dapat dilakukan perbaikan. Pengujian akan menjadi acuan dalam pengembangan pada antarmuka *website*. Pengujian *black box* ini menguji antarmuka yang terdapat pada *website*.

#### 4.2.1 Alpha Test

Pengujian dilakukan oleh responden yang akan menguji dengan menjalankan setiap fungsi yang telah dijabarkan pada setiap poin pengujian. Penilaian dilakukan dengan menentukan skala penilaian yang akan diisi berdasarkan kesesuaian dari hasil pengujian. Poin-poin pengujian beserta hasil pengujian *alpha test* dapat dilihat pada tabel 4.2.

Bobot penilaian untuk pengujian *alpha test* menggunakan 5 skala penilaian, skala penilaian yang digunakan sesuai dengan tabel 4.1.

**Tabel 4. 1** Bobot Penilaian Pengujian *Alpha Test*

No	Bobot	Range	Penilaian	Keterangan
1	5	100%-81%	SB	Sangat Baik
2	4	80%-61%	B	Baik
3	3	60%-41%	CB	Cukup Baik
4	2	40%-21%	KB	Kurang Baik
5	1	20%-0%	SKB	Sangat Kurang Baik

##### 1. Pengujian Alpha Test Pada Halaman Admin

Hasil pengujian *alpha test* dilakukan sebanyak 3 responden. Pengujian ini menggunakan 10 pengujian menu dan 28 detail pengujian. Pada pengujian ini menggunakan 5 skala penilaian, skala penilaian sesuai dengan table 4.11. Skala penilaian yang tersedia sudah meliputi fungsi pengoperasian program, tampilan *user interface*, dan interaksi pembelajaran. Pada table 4.2 adalah total jumlah pengujian *alpha test* dari 3 responden.

**Tabel 4. 2** Jumlah Pengujian *Alpha Test*

No	Pengujian	Detail Pengujian	Penilaian				
			SB	B	CB	KB	SKB
1	Halaman Dashboard	Menampilkan Halaman Home	3	0	0	0	0
2	Halaman Jalur Pickup	Menampilkan data jalur pickup	1	2	0	0	0
		Menampilkan data detail perhitungan	2	1	0	0	0
		Menampilkan data perhitungan setiap driver	2	0	1	0	0
3	Halaman Sektor	Menampilkan data sektor	1	2	0	0	0
		Menampilkan data tps	1	2	0	0	0
		Menampilkan data rute	1	2	0	0	0
		Keberhasilan proses tambah data	1	2	0	0	0
		Keberhasilan proses <i>edit</i> data	1	2	0	0	0
		Keberhasilan proses hapus data	2	1	0	0	0
4	Halaman Kendaraan	Menampilkan data kendaraan	2	1	0	0	0
		Keberhasilan proses tambah data	2	0	1	0	0
		Keberhasilan proses <i>edit</i> data	3	0	0	0	0
		Keberhasilan proses hapus data	2	1	0	0	0

5	Halaman Kendaraan TPA	Menampilkan data Kendaraan tpa	1	1	1	0	0
		Keberhasilan proses tambah data	2	0	1	0	0
		Keberhasilan proses Edit data	2	0	1	0	0
		Keberhasilan proses hapus data	2	1	0	0	0
6	Halaman TPA	Menampilkan data tpa	2	1	0	0	0
		Keberhasilan proses tambah data	3	0	0	0	0
		Keberhasilan proses Edit data	3	0	0	0	0
		Keberhasilan proses <i>edit</i> data	3	0	0	0	0
		Keberhasilan proses hapus data	3	0	0	0	0
7	Halaman User	Menampilkan halaman User	1	2	0	0	0
		Keberhasilan proses tambah data	2	1	0	0	0
		Keberhasilan proses Edit data	3	0	0	0	0
		Keberhasilan proses hapus data	2	1	0	0	0
8	<i>Logout</i>	Keberhasilan <i>logout</i> atau menutup aplikasi	3	0	0	0	0
<b>Total</b>			<b>56</b>	<b>23</b>	<b>5</b>	<b>0</b>	<b>0</b>

Adapun skala penilaian yaitu:

SB = Sangat Baik      CB = Cukup Baik      SKB = Sangat Kurang baik  
 B = Baik      KB = Kurang Baik

Hasil pengujian menggunakan metode uji validasi *alpha test* dapat diperoleh kesimpulan bahwa responden memberikan jawaban sangat baik sebanyak 56 jawaban, baik 23 jawaban, cukup baik 5 jawaban, kurang baik 0 jawaban, dan sangat kurang baik 0 jawaban. Untuk menghitung persentase dari jawaban pengisian *alpha test* rumusnya sebagai berikut:

Persentase =

$$\frac{\text{Jumlah}}{\text{Jumlah jawaban} \times \text{Jumlah Responden}} \times 100\%$$

Hasil dari persentase dan pengujian *alpha test* sebagai berikut:

Sangat Baik      = 56/84 x 100      = 67%  
 Baik      = 23/84 x 100      = 27%  
 Cukup Baik      = 5/84 x 100      = 6%  
 Kurang Baik      = 0/84 x 100      = 0%  
 Sangat Kurang Baik      = 0/84 x 100      = 0%

#### 4.2.2 Pengujian Beta Test

Pengujian *beta test* adalah pengujian yang dilakukan oleh pemakai oleh lingkungan pemakai, perangkat lunak tidak lagi dapat dikendalikan oleh pengembang. Pengujian *beta* merupakan pengujian yang dilakukan secara objektif secara langsung mengenai tanggapan pengguna terhadap aplikasi. Hasil pengujian *beta test* dilakukan sebanyak 10 responden. Pengujian ini menggunakan 10 item uji. Pada pengujian ini menggunakan 5 skala penilaian, skala penilaian sesuai dengan tabel 4.3.

**Tabel 4. 3** Skala Penilaian Pegujian Beta Test

No	Bobot	Range	Penilaian	Keterangan
1	5	100%-81%	SS	Sangat Setuju
2	4	80%-61%	S	Setuju
3	3	60%-41%	CS	Cukup Setuju
4	2	40%-21%	KS	Kurang Setuju
5	1	20%-0%	SKS	Sangat Kurang Setuju

Tabel 4.4 merupakan total jumlah penilaian dari keseluruhan *form* pengujian yang ada. Pengujian *beta test* di uji oleh 10 responden. Item uji yang digunakan dalam pengujian *beta test* terdiri dari 10 item.

**Tabel 4. 4** Jumlah Pengujian Beta Test

No	Item Uji	Penilaian				
		SS	S	CS	KS	SKS
1	Aplikasi memiliki tampilan yang mudah digunakan	4	6	0	0	0
2	Aplikasi mudah dioperasikan oleh <i>user</i> (pengguna)	5	3	2	0	0
3	Aplikasi sesuai dengan kebutuhan user (pengguna)	4	3	3	0	0
4	Tampilan aplikasi sesuai dengan kebutuhan user	4	4	2	0	0
5	Halaman Jalur Pickup mudah dimengerti	3	4	3	0	0
6	Halaman data Sektor mudah dimengerti	5	3	2	0	0
7	Halaman data Tps mudah dimengerti	4	4	2	0	0
8	Halaman data kendaraan mudah dimengerti	4	4	2	0	0
9	Halaman data Tpa mudah dimengerti	4	5	1	0	0
10	Hasil penentuan rute pengambilan dan pembuangan sampah mudah dimengerti	3	4	3	0	0
<b>Total</b>		<b>40</b>	<b>40</b>	<b>20</b>	<b>0</b>	<b>0</b>

Adapun skala penilaian yaitu:

SB = Sangat Baik

CB = Cukup Baik

SKB = Sangat Kurang baik

B = Baik

KB = Kurang Baik

Hasil pengujian menggunakan metode uji validasi *beta test* dapat diperoleh kesimpulan bahwa responden memberikan jawaban sangat baik sebanyak 40 jawaban, baik 40 jawaban, cukup baik 20 jawaban, kurang baik 0 jawaban, dan sangat kurang baik 0 jawaban. Untuk menghitung persentase dari jawaban pengisian *beta test* rumusnya sebagai berikut:

Persentase =

$$\frac{\text{Jumlah}}{\text{Jumlah jawaban} \times \text{Jumlah Responden}} \times 100\%$$

Hasil dari persentase dan pengujian *beta test* sebagai berikut:

Sangat Baik	= 40/100 x 100	= 40%
Baik	= 40/100 x 100	= 40%
Cukup Baik	= 20/100 x 100	= 20%
Kurang Baik	= 0/100 x 100	= 0%
Sangat Kurang Baik	= 0/100 x 100	= 0%

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan dari hasil analisis, perancangan dan pembahasan yang telah dilakukan sebelumnya, maka dihasilkan sebuah sistem penentuan jalur pengambilan sampah. Dari hasil penelitian, kesimpulan yang dapat diambil antara lain:

1. Penentuan jalur pengambilan dan pembuangan sampah yang efisien menggunakan metode *Nearest Neighbor* menggunakan data di dinas lingkungan hidup kota Yogyakarta.
2. Pembuatan Aplikasi web ini menggunakan fitur *Google Maps* pada pencarian waktu tempuh yang diambil oleh *driver*, dengan menggunakan bahasa pemrograman PHP.
3. Berdasarkan pengujian *alpha test* didapatkan presentase akhir sangat baik sebanyak 67%, baik 27%, cukup baik 6%, kurang baik 0%, dan sangat kurang baik 0%.
4. Berdasarkan pengujian *beta test* didapatkan presentase akhir sangat baik sebanyak 40%, baik 40%, cukup baik 20%, kurang baik 0%, dan sangat kurang baik 0%.

### 5.2 Saran

Pada penelitian yang telah dilakukan terdapat beberapa saran sebagai masukan agar dapat dilakukan pengembangan sistem selanjutnya, yaitu:

1. Dilakukan perbandingan metode untuk membandingkan hasil masing-masing metode.
2. Dapat digunakan pertimbangan kemacetan dan jadwal pada saat pengambilan dan pembuangan sampah untuk mengantisipasi keterlambatan waktu.
3. Dapat menambah fitur map agar memudahkan *driver*.
4. Dapat diterapkan melalui sistem pelaporan versi android agar masyarakat dapat melapor lokasi TPS mana yang sampahnya sudah menumpuk.
5. Refrensi yang valid menyangkut data yang dibutuhkan, penelitian yang pernah ada, dan metode-metode tentang optimasi jalur sangat dibutuhkan untuk membantu selama proses penelitian.



## DAFTAR PUSTAKA

- Abadi, C., Susanty, S., & Adianto, H. (2014). *Penentuan Rute Kendaraan Distribusi Produk Roti Menggunakan Metode Nearest Neighbor dan Metode Sequential Insertion* \*. 01(03), 152–163.
- Andayani, S., & Perwitasari, E. W. (2014). Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra. *Aeminar Nasional Teknologi Informasi & Komunikasi Terapan (SEMANTIK)*, 164–170.
- Arinalhaq, F., Imran, A., & Fitria, L. (2013). Penentuan Rute Kendaraan Pengangkutan Sampah dengan Menggunakan Metode Nearest Neighbour ( Studi Kasus PD Kebersihan Kota. *Jurnal Online Institut Teknologi Nasional*, 1(1), 22–32.
- Hutasoit, C. S., Susanty, S., & Imran, A. (2014). Penentuan Rute Distribusi Es Balok Menggunakan Algoritma Nearest Neighbour dan Local Search (Studi kasus di PT X). *Reka Integra*, 02(02), 268–276.
- Kharisma, Hartati &. (2015). “OPTIMASI RUTE DISTRIBUSI BARANG FROZEN DI PT. SUKANDA DJAYA DENGAN MENGGUNAKAN METODE NEAREST NEIGHBOUR (STUDI KASUS: PT. SUKANDA DJAYA PEKANBARU) Misra.”: 84–90.
- Armandi, Ekky; Purwani, Annie; Linarti, Utaminingsih. (2019). Optimasi Rute Pengangkutan Sampah Kota Yogyakarta Menggunakan Hybrid Genetic Algorithm. *Jurnal Ilmiah Teknik Industri*, 18(2), 236-244.
- Supriyo, Prapto Tri; Aman, Amril; Bakhtiar, Toni; Hanum, Farida (2017). Model Optimasi Pengelolaan Sampah Perkotaan: Penentuan Lokasi Pembuangan Sampah. *Seminar Nasional Matematika*, 701-708.
- Oktavia, Chendrasari Wahyu; Natalia, Christine; Adigunawan, Indra (2019). Penentuan Jalur Rute Distribusi Produk Fast Moving Consumer Goods (FMCG) Dengan Menggunakan Metode Nearest Neighbor (Studi Kasus: PT. XYZ). *Jurnal Al-Azhar Indonesia Seri Sains Dan Teknologi*. 5(2), 101.
- Pratama, Dicky (2019). Optimasi *Travelling Salesman Problem With Time Windows* (TSP-TW) Pada Rute Obyek Wisata Di Provinsi D.I. Yogyakarta Menggunakan *Algoritma Genetika* Berbasis Android. *Jurnal Wisata*. 30-41.
- Pratama P Albiyan; JH Frans; Utomo Sudiyo (2019). Optimilisasi Rute Pengangkutan Sampah Berbasis Sistem Informasi Geografis Di Kupang. *Jurnal Teknik Sipil*. 8(1), 12.
- HAR Lubis; Maulana Andrean; RB Frazila (2016). Penerapan Konsep *Vehicle Routing Problem* Dalam Kasus Pengangkutan Sampah Di Perkotaan. *Jurnal Teknik Sipil ITB*. 23(1), 213-222.
- Ambarizki, Prismeida Putri Dara (2016). Optimasi Sistem Pengangkutan Sampah Berdasarkan Kapasitas Kendaraan Pengangkutan Dan Kondisi Kontainer Sampah Di Kota Surabaya Barat. *Jurnal Institute Sepuluh November*. 25, 201.
- HP Lestari (2013). Penerapan Algoritma Koloni Semut Untuk Optimasi Rute Distribusi Pengangkutan Sampah Di Kota Yogyakarta. *Jurnal Sains Dasar*. 2, 1.

E Fitriana (2019). Optimasi Pengangkutan Sampah Di Kecamatan Medan Tembung Menggunakan Metode *Saving Matrix*. *Undergraduate Thesis*. 122.

ATPD Akhsan; Z Zainuddin; Achmad Andani (2019). Optimasi Rute Menggunakan Algoritma *Greedy* Pada Pengangkutan Sampah Di Kota Makassar. *IT Jurnal*. 10(1).