



**PENERAPAN ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE  
TERPENDEK TRUK PENGANGKUT SAMPAH  
(STUDI KASUS : KABUPATEN JEMBER)**

**SKRIPSI**

Oleh

**Lucky Indrayu Hapsari**

**NIM 13241010111**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**



**PENERAPAN ALGORITMA DJIKSTRA DALAM PENCARIAN RUTE  
TERPENDEK TRUK PENGANGKUT SAMPAH  
(STUDI KASUS : KABUPATEN JEMBER)**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

Oleh

**Lucky Indrayu Hapsari**

**NIM 132410101011**

**PROGRAM STUDI SISTEM INFORMASI**

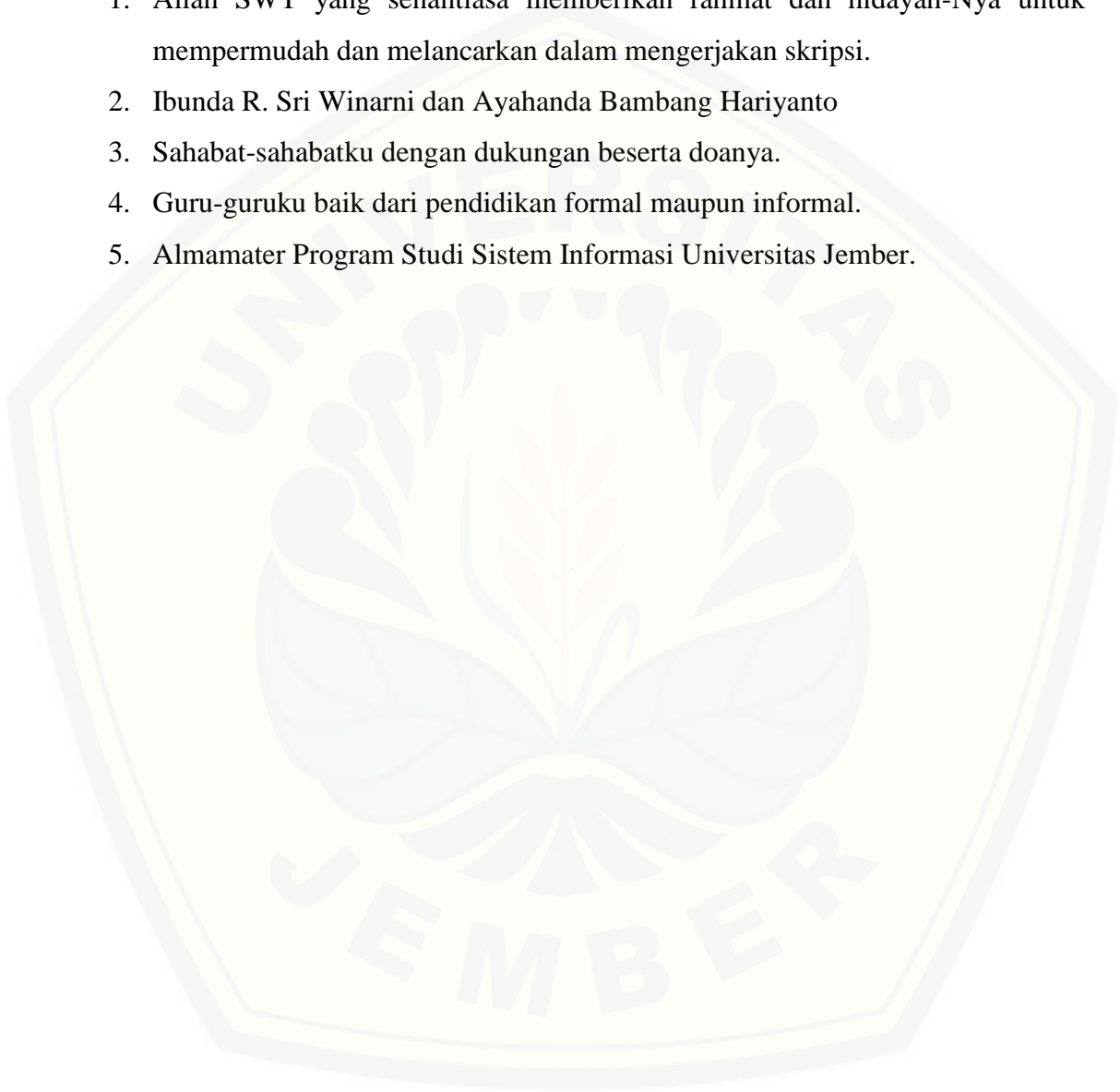
**UNIVERSITAS JEMBER**

**2017**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi.
2. Ibunda R. Sri Winarni dan Ayahanda Bambang Hariyanto
3. Sahabat-sahabatku dengan dukungan beserta doanya.
4. Guru-guruku baik dari pendidikan formal maupun informal.
5. Almamater Program Studi Sistem Informasi Universitas Jember.



**MOTO**

*“This too shall pass”*



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

nama : Lucky Indrayu Hapsari

NIM : 132410101011

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan Algoritma Dijkstra Dalam Perhitungan Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 14 Juli 2017

Yang menyatakan,

Lucky Indrayu Hapsari

NIM 132410101011

**SKRIPSI**

**PENERAPAN ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE  
TERPENDEK TRUK PENGANGKUT SAMPAH  
(STUDI KASUS : KABUPATEN JEMBER)**

Oleh

Lucky Indrayu Hapsari

NIM 13241010111

Pembimbing :

Dosen Pembimbing Utama : Prof. Drs. Slamim, M.Comp.Sc.,Ph.D

Dosen Pembimbing Pemdamping : Nelly Oktavia Adiwijaya, S.Si., MT

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember)” telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 14 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Drs. Slamir M.Comp.Sc., Ph.D  
NIP. 19670420 1992011001

Nelly Oktavia Adiwijaya, S.Si., MT  
NIP. 198410242009122008

**PENGESAHAN PENGUJI**

Skripsi berjudul “Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember)”, telah diuji dan disahkan pada:

hari,tanggal : Kamis, 13 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Anang Andrianto, ST., MT.

NIP. 196906151997021002

Yanuar Nurdiansyah, ST., M.Cs.

NIP. 19820101 2010121004

Mengesahkan  
Ketua Program Studi

Prof. Drs. Slamini, M.Comp.Sc.,Ph.D

NIP. 19670420 1992011001



## RINGKASAN

Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember); Lucky Indrayu Hapsari, 132410101011 2017, 141 halaman ; Program Studi Sistem Informasi Universitas Jember.

Aktivitas manusia dalam memanfaatkan alam selalu meninggalkan sisa yang dianggap sudah tidak berguna lagi sehingga diperlakukan sebagai barang buangan, yaitu sampah dan limbah (Widyatmoko dan Sintorini, 2002). Timbulan sampah di Kabupaten Jember yang tidak terlayani banyak yang dibiarkan membusuk dan dibuang ke lahan kosong. Jarak dari Tempat Pembuangan Sementara (TPS) ke Tempat Pembuangan Akhir (TPA) juga menjadi dasar masalah. Proses pengangkutan sampah yang menjadi salah satu faktor penghambat yang seharusnya dapat dicari solusinya dengan mencari rute terpendek yang dapat dilalui kendaraan pengangkut sampah dalam melakukan proses pengangkutan sampah dari sumber ataupun dari TPS ke TPA. Untuk mendukung usaha pengangkutan sampah dapat kita terapkan dalam sebuah sistem. Dalam Teori Graf terdapat kajian yang membahas tentang pencarian rute terpendek. Salah satunya ialah algoritma Dijkstra. Algoritma Dijkstra memiliki konsep untuk pencarian rute terpendek, yakni memilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Berdasarkan penelitian yang telah dilakukan sebelumnya dan konsep dari algoritma Dijkstra itu sendiri dapat dilihat bahwa algoritma Dijkstra cocok untuk diterapkan sebagai solusi pemecahan masalah pada penelitian ini dan diterapkan dalam pembangunan sebuah sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember.

## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan “Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember dan selaku dosen pembimbing akademik yang telah mendampingi penulis sebagai mahasiswa;
2. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Nelly Oktavia Adiwijaya, S.Si., MT selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember.
4. Ibunda R. Sri Winarni dan Ayahanda Bambang Hariyanto yang selalu mendukung dan mendoakan.
5. Ikatan Keluarga Besar Soemo Winardi yang telah mendukung dengan do'a, nasehat dan motivasi.
6. *Aunty* Lely Triana Avia yang selalu setia menemani dan mendo'akan serta memberi semangat yang luar biasa.
7. Keluarga bebeb: Devie depoy, Karina lele, Andin twin, Vita nobita beserta Nisaa jubaedah, Adul Jihadul, Talitha, Arief, Yofanda, Mas Affan, Mbak Ainul, Mas Marceli, Mas Willi, Akbar, Keke, Dzul, Raras, Bunga, Dinda, Mbak Hesti, *thank you guys, you are such a blessings.*
8. Muhammad Huda Gembul Muttaqin yang selalu menemani, membantu, dan mendo'akan, semoga semua niat baiknya dilancarkan dan disegerakan.

9. Keluarga besar Laboratorium Basis Data.
10. Teman-teman seperjuangan INTENTION angkatan 2013.
11. Teman-teman Program Studi Sistem Informasi di semua angkatan.
12. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 13 Juli 2017

Penulis

**DAFTAR ISI**

SKRIPSI.....	i
SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iii
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN.....	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL.....	xvii
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu.....	5
2.2 Definisi Sampah.....	6
2.3 Peraturan Terkait Pengangkutan Sampah.....	7

2.6	Graf.....	8
2.7	Algoritma Dijkstra.....	9
2.8	Model <i>Waterfall</i> .....	13
BAB 3. METODOLOGI PENELITIAN.....		14
3.1	Jenis Penelitian .....	14
3.2	Tempat dan waktu penelitian .....	14
3.3	Tahapan Penelitian .....	14
3.1.1	Tahapan Analisis Kebutuhan .....	15
3.1.2	Tahapan Desain Sistem .....	18
3.1.3	Tahapan Implementasi Sistem .....	19
3.1.4	Tahapan Pengujian Sistem .....	19
3.1.5	Tahapan Pemeliharaan Sistem .....	20
BAB 4. PENGEMBANGAN SISTEM.....		21
4.1	Analisis Kebutuhan Sistem .....	21
4.1.2	Kebutuhan Non-Fungsional .....	22
4.2	Desain Sistem .....	22
4.2.1	<i>Business Process</i> .....	22
4.2.2	<i>Use Case Diagram</i> .....	23
4.2.3	<i>Use Case Scenario</i> .....	25
4.2.4	<i>Sequence Diagram</i> .....	27
4.2.5	<i>Activity Diagram</i> .....	29
4.2.6	<i>Class Diagram</i> .....	31
4.2.7	<i>Entity Relationship Diagram (ERD)</i> .....	35
4.2	Penulisan Kode Program .....	35
4.3.1	Kode Program Mengelola Data TPS dan TPA .....	35

4.3.2 Kode Program Melihat Pembuangan Sampah .....	35
4.3.3 Kode Program Melihat <i>User</i> .....	36
4.3.4 Kode Program Mengelola Data Pembuangan Sampah .....	36
4.3.5 Kode Program Melihat Peta .....	42
4.3.6 Kode Program Verifikasi Pembuangan Sampah.....	42
4.3 Pengujian Sistem .....	42
4.3.1 Pengujian <i>White Box</i> .....	42
4.3.2 Pengujian <i>Black Box</i> .....	47
BAB 5. HASIL DAN PEMBAHASAN.....	49
5.1 Hasil Penelitian.....	49
5.1.1 Perhitungan Metode Dijkstra dalam Pembuangan Sampah.....	49
5.1.2 Penentuan Jalur Terpendek Pembuangan Sampah dalam Sistem ...	50
5.2 Hasil Pembangunan Aplikasi .....	53
5.2.1 Halaman <i>Login</i> .....	53
5.2.2 Halaman Dashboard .....	53
5.2.3 Halaman TPS dan TPA .....	55
5.2.4 Halaman Pembuangan Sampah.....	55
5.2.5 Halaman Manajemen User .....	56
5.2.6 Halaman Pembuangan Sampah.....	56
5.2.7 Halaman TPS dan TPA .....	57
5.2.8 Halaman Verifikasi Data Pembuangan Sampah .....	57
5.3 Pembahasan .....	58
5.3.1 Analisis Algoritma Dijkstra .....	58
5.3.2 Analisis Model <i>Waterfall</i> .....	59
BAB 6. PENUTUP .....	60

6.1	Kesimpulan.....	60
6.2	Saran.....	60
DAFTAR PUSTAKA .....		61
LAMPIRAN.....		63
A.	<i>Use Case Scenario</i> .....	63
A.1.	Skenario <i>Use Case</i> Mengelola Data TPS dan TPA .....	63
A.2	Skenario <i>Use Case</i> Melihat Data Pembuangan Sampah.....	66
A.3	Skenario <i>Use Case</i> Mengelola <i>User</i> .....	66
A.4	Skenario <i>Use Case</i> Mengelola Pembuangan Sampah.....	69
A.5	Skenario <i>Use Case</i> Melihat Peta .....	70
A.6	Skenario <i>Use Case</i> Verifikasi Data Pembuangan Sampah .....	71
A.7	Skenario <i>Use Case Login</i> .....	72
A.8	Skenario <i>Use Case Log Out</i> .....	73
B.	<i>Sequence Diagram</i> .....	73
B.1.	<i>Sequence Diagram</i> Mengelola Data TPS dan TPA.....	73
B.2	<i>Sequence Diagram</i> Mengelola Data <i>User</i> .....	74
B.3	<i>Sequence Diagram</i> Melihat Data TPS dan TPA .....	75
B.4	<i>Sequence Diagram</i> Melihat Data Pembuangan Sampah.....	75
B.5	<i>Sequence Diagram Login</i> .....	75
B.6	<i>Sequence Diagram</i> Verifikasi Data Pembuangan Sampah .....	76
C.	<i>Activity Diagram</i> .....	77
C.1	<i>Activity Diagram</i> Mengelola Data TPS dan TPA.....	77
C.2	<i>Activity Diagram</i> Melihat Data Pembuangan Sampah.....	78
C.3	<i>Activity Diagram</i> Pengelolaan Data <i>User</i> .....	79
C.4	<i>Activity Diagram</i> Melihat TPS dan TPA.....	79

C.5	<i>Activity Diagram</i> Mengelola Data Pembuangan Sampah .....	80
D.	Kode Program.....	82
D.1	Kode Program Pengelolaan Data Sampah .....	82
D.2	Kode Program Melihat Data Pembuangan Sampah.....	93
D.3	Kode Program Pengelolaan Data User .....	96
D.4	Kode Program Melihat TPS dan TPA .....	108
E.	Pengujian <i>White Box</i> .....	120
E.1	Kelas Controller.....	120
E.2	Kelas Model.....	122
F.	Pengujian <i>Black Box</i> .....	123
F.1	Fitur Pengelolaan Data Akun.....	123
F.2	Fitur Mengelola Data TPS dan TPA.....	124
F.3	Fitur Melihat Data Pembuangan Sampah .....	125
F.4	Fitur Melihat Data TPS dan TPA .....	126



**DAFTAR GAMBAR**

Gambar 2. 1 Graf Terhubung dan Graf Tidak Terhubung .....	9
Gambar 2. 2 Diagram Alir Algoritma Dijkstra .....	10
Gambar 2. 3 Contoh Graf penentuan jalur terpendek .....	11
Gambar 2. 4 Model Waterfall .....	13
Gambar 3. 1 Diagram tahap analisis data.....	16
Gambar 3. 2 Diagram alir penentuan jalur terpendek .....	17
Gambar 4. 1 Business Process .....	22
Gambar 4. 2 Use Case Diagram .....	23
Gambar 4. 3 Sequence Diagram Pengelolaan Pembuangan Sampah .....	29
Gambar 4. 4 Activity Diagram Mengelola Data Pembuangan Sampah.....	30
Gambar 4. 5 Classs Diagram.....	31
Gambar 4. 6 Entity Relation Diagram .....	35
Gambar 4. 7 Diagram Alir Function _construct.....	43
Gambar 4. 8 Diagram Alir Function cobaakun() .....	44
Gambar 4. 9 Diagram Alir Function profil().....	44
Gambar 4. 10 Diagram Alir Function detail(\$id_user).....	45
Gambar 4. 11 Diagram Alir Function add().....	46
Gambar 5. 1 Representasi Graf Sistem .....	50
Gambar 5. 2 Kode Program Fungsi pgr_dijkstra .....	52
Gambar 5. 3 Penentuan Rute Terpendek TPS Talangsari – TPA Pakusari .....	52
Gambar 5. 4 Halaman Utama Aplikasi .....	53
Gambar 5. 5 Halaman Dashboard Admin .....	54
Gambar 5. 6 Halaman Dashboard Sopir .....	54
Gambar 5. 7 Halaman Dashboard TPA.....	54
Gambar 5. 8 Halaman Manajemen TPS dan TPA .....	55
Gambar 5. 9 Halaman Pembuangan Sampah.....	55
Gambar 5. 10 Halaman Manajemen User .....	56
Gambar 5. 11 Halaman Pembuangan Sampah.....	56
Gambar 5. 12 Halaman TPS dan TPA .....	57
Gambar 5. 13 Halaman Verifikasi Pembuangan Sampah.....	57

**DAFTAR TABEL**

Tabel 2. 1 Penyelesaian Graf dengan menggunakan Algoritma Dijkstra .....	12
Tabel 4. 1 Definsi Aktor.....	24
Tabel 4. 2 Definisi Use Case.....	24
Tabel 4. 3 Skenario Use Case Pengelolaan Pembuangan Sampah (Sopir) .....	26
Tabel 4. 4 Kode Program Controller Pembuangan .....	36
Tabel 4. 5 Kode Program Kelas Controller Route .....	38
Tabel 4. 6 Kode Program Kelas Pembuangan Model.....	40
Tabel 4. 7 Kode Program Kelas Route Model .....	41
Tabel 4. 8 Test Case Function _construct() .....	43
Tabel 4. 9 Test Case Function cobaakun().....	44
Tabel 4. 10 Test Case Function profil() .....	45
Tabel 4. 11 Test Case Function detail(\$id_user) .....	46
Tabel 4. 12 Test Case Function add() .....	47
Tabel 4. 13 Pengujian Black Box Mengelola Pembuangan Sampah (Sopir).....	47
Tabel 5. 1 Panjang Jalan TPS Talang Sari - TPA Pakusari .....	49
Tabel 5. 2 Perhitungan Representasi Graf dari Algoritma Dijkstra.....	50
Tabel 5. 3 Relasi Antara Tabel Jalan dan Tabel Jenis Kendaraan .....	51
Tabel 5. 4 Nilai cost dan reverse_cost .....	52

## BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### 1.1 Latar Belakang

Aktivitas manusia dalam memanfaatkan alam selalu meninggalkan sisa yang dianggap sudah tidak berguna lagi sehingga diperlakukan sebagai barang buangan, yaitu sampah dan limbah (Widyatmoko dan Sintorini, 2002). Sampah adalah suatu buangan berupa padat yang membawa berbagai jenis penyakit, menurunkan sumber daya, menimbulkan polusi, menyumbat saluran air dan berbagai akibat negatif lainnya.

Di kota-kota besar untuk menjaga kebersihan sering kali menyingkirkan sampah ke tempat yang jauh dari pemukiman atau yang biasa disebut Tempat Pembuangan Akhir (TPA). Menurut data dari PJU Kebersihan Kabupaten Jember, kondisi sampah Kabupaten Jember saat ini sekitar 5.538,47 M<sup>3</sup>/hari namun yang terangkut ke TPA hanya 700,53 M<sup>3</sup>/hari atau sama dengan 12.64 % dari timbulan sampah yang ada. Timbulan sampah yang tidak terlayani banyak yang dibiarkan membusuk dan dibuang ke lahan kosong yang ada di Kabupaten Jember.

Umumnya pengumpulan sampah di Tempat Pembuangan Sementara (TPS) dilakukan secara tercampur. Sedangkan pengangkutan sampah baik dari sumber maupun dari TPS ke TPA yang seharusnya mampu terangkut setiap hari belum dapat diaplikasikan dengan baik dan teratur sehingga tak jarang suatu TPS dapat menampung sampah lebih banyak dari daya tampung sebenarnya. Timbunan sampah lebih dari dua hari ini menimbulkan bau tak sedap, lalat, dan lindi yang dapat meluber ke jalan. Masalah lainnya ialah panjang lintasan dari satu TPS atau TPA ke TPS dan TPA yang saling berjauhan sehingga menjadi faktor lain penghambat proses pengangkutan sampah. Selain itu manajemen angkut sampah

yang belum memiliki prosedur yang baik juga menjadi kendala penghambat proses pengangkutan sampah.

Proses pengangkutan sampah seperti pada masalah yang telah diuraikan diatas seharusnya dapat dicari solusinya dengan mencari rute terpendek yang dapat dilalui kendaraan pengangkut sampah dalam melakukan proses pengangkutan sampah dari sumber ataupun dari TPS ke TPA.

Untuk mendukung usaha pengangkutan sampah dapat kita terapkan dalam sebuah sistem. Dalam Teori Graf terdapat kajian yang membahas tentang pencarian rute terpendek. Terdapat beberapa algoritma yang dapat digunakan untuk membantu pemecahan solusi dari masalah pencarian rute terpendek antara lain yaitu Algoritma Dijkstra dan *Ant Colony Optimize (ACO)*.

Dilihat dari sistem manual pengangkutan sampah yang diterapkan saat ini, konsep yang dapat disimpulkan dari uraian masalah diatas dapat diterapkan kedalam graf dengan mempartisi titik-titik koordinat TPS dengan tujuan TPA yang sama terlebih dahulu sehingga dapat dihitung rute terpendeknya dengan algoritma pencarian rute terpendek.

Pada penelitian sebelumnya mengenai sistem aplikasi penentuan rute terpendek pada jaringan multi moda transportasi umum menggunakan Algoritma Dijkstra yang dilakukan oleh Sofyan Arifianto, Penelitian tersebut dilakukan menggunakan Algoritma Dijkstra dan dianggap cocok karena mudah digunakan oleh *user*. Algoritma ini dapat menentukan jalur terpendek dari *graph* berbobot yang bobotnya bernilai lebih besar dari nol (positif), dari titik awal ke semua titik yang dikehendaki, sehingga nantinya akan ditemukan jalur terpendek dari titik awal dan titik tujuan yang diinputkan. Untuk mempermudah penggunaan sistem ini, maka *user* hanya menginputkan tentang informasi titik awal, dan titik tujuan. (Arifianto, 2012)

Algoritma tersebut memiliki konsep untuk pencarian rute terpendek, yakni memilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Berdasarkan penelitian yang telah dilakukan sebelumnya dan konsep dari algoritma Dijkstra itu sendiri dapat dilihat bahwa algoritma Dijkstra cocok untuk

diterapkan sebagai solusi pemecahan masalah pada penelitian ini dan diterapkan dalam pembangunan sebuah sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember.

## 1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini adalah sebagai berikut:

1. Bagaimana mencari rute terpendek truk pengangkut sampah di Kabupaten Jember dengan menggunakan Algoritma Dijkstra?
2. Bagaimana mengembangkan aplikasi pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember dengan menggunakan Algoritma Dijkstra?

## 1.3 Tujuan

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Mengimplementasikan Algoritma Dijkstra untuk pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember.
2. Mengembangkan aplikasi pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember menggunakan Algoritma Dijkstra.

## 1.4 Batasan Masalah

Penulis memberikan batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan aplikasi. Berikut adalah batasan masalah yang dicantumkan:

1. Objek penelitian adalah TPS dan TPA di Kabupaten Jember.
2. Batasan wilayah yang akan diuji dalam penelitian ini adalah terbatas di Kabupaten Jember.
3. Daya tampung truk pengangkut sampah sama.
4. Parameter bobot berdasarkan pada jarak yang ditempuh.

### **1.5 Sistematika Penulisan**

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literatur, jurnal, dan internet.

### 2.1 Penelitian Terdahulu

Adapun penelitian terdahulu sehingga penelitian ini muncul adalah sebagai berikut :

Penelitian sebelumnya dilakukan oleh Desi Wulandari, yang telah mengembangkan sistem pencarian rute terpendek pengangkutan sampah dengan menggunakan algoritma *Ant Coloni Optimization (ACO)*. Pada penelitian ini penulis menerapkan algoritma *ACO* untuk mencari rute terpendek pengangkutan sampah di Surabaya sehingga berdampak pada efisiensi waktu dan biaya yang diperlukan selama proses pengangkutan sampah. Selain itu hasil dari penelitian tersebut ialah penggunaan sistem yang telah dibangun menggunakan *ACO* berdasarkan letak titik yang sudah ditentukan lebih optimal daripada rute yang selama ini dilalui oleh armada kebersihan.

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan Algoritma Dijkstra yaitu dengan judul “Sistem Informasi Geografis Pencarian SPBU Terdekat dan Penentuan Jalur Terpendek Menggunakan Algoritma Dijkstra di Kabupaten Jember Berbasis Web” yang dilakukan oleh Abdul Roqib dari Program Studi Sistem Informasi Universitas Jember pada tahun 2014. Pada penelitian ini dijelaskan bagaimana merancang basis data spasial dan atribut stasiun pengisian bahan bakar serta membangun sistem informasi geografis berbasis web pencarian SPBU terdekat dan penentuan jalur terdekat di Kabupaten Jember menggunakan algoritma Dijkstra. Hasil dari penelitian ini adalah algoritma Dijkstra dapat menentukan jalur tercepat dengan mengganti nilai *cost* pada *pgrouting* dengan nilai waktu tempuh yang di dapat dari

perhitungan jarak dibagi kecepatan dan Hasil Perhitungan jalur terpendek bergantung pada nilai kriteria, *cost* dan *reverse\_cost*. (Roqib, 2014).

## 2.2 Definisi Sampah

Berbagai aktivitas dilakukan oleh manusia untuk memenuhi kesejahteraan hidupnya dengan memproduksi barang dari sumber daya alam. Disamping menghasilkan barang yang dikonsumsi manusia dihasilkan pula bahan buangan yang sudah tidak dibutuhkan lagi oleh manusia. Bahan buangan makin bertambah banyak, hal ini erat hubungannya dengan makin bertambahnya jumlah penduduk. Bahan buangan ini dikenal dengan istilah sampah (*wastes*) berbentuk padat, cair, dan gas.

Sampah adalah limbah yang bersifat padat terdiri dari zat organik dan zat anorganik yang dianggap tidak berguna dan harus dikelola agar tidak membahayakan lingkungan dan melindungi investasi pembangunan. Sampah perkotaan adalah sampah yang timbul di kota (SNI 19-2454-2002).

Pengertian sampah menurut SNI 13-1990-F tentang Tata Cara Pengelolaan Teknik Sampah Perkotaan didefinisikan sebagai limbah yang bersifat padat, terdiri dari zat organik dan zat anorganik yang dianggap tidak berguna lagi dan harus dikelola agar tidak membahayakan lingkungan dan melindungi investasi pembangunan. Sampah yang merupakan hasil sampingan dari aktivitas manusia telah menimbulkan permasalahan yang sangat kompleks, antara lain:

1. Masalah estetika dan kenyamanan.
2. Merupakan sarang atau tempat berkumpulnya berbagai binatang yang dapat menjadi vektor penyakit.
3. Menyebabkan terjadinya polusi udara, air dan tanah.
4. Menyebabkan terjadinya penyumbatan saluran-saluran air buangan dan drainase.



### 2.3 Peraturan Terkait Pengangkutan Sampah

Pengangkutan sampah menurut UU no 18 Tahun 2008 tentang Pengelolaan Sampah, merupakan bagian dari penanganan sampah. Pengangkutan didefinisikan sebagai dalam bentuk membawa sampah dari sumber dan atau dari tempat penampungan sampah sementara atau dari TPS 3R (reduce, reuse, recycle) menuju ke tempat pengolahan sampah terpadu atau tempat pemrosesan akhir. Beberapa acuan normatif juga mencantumkan tentang pengaturan pengangkutan sampah, antara lain:

1. Pedoman Standar Pelayanan Minimal Pedoman Penentuan Standar Pelayanan Minimal Bidang Penataan Ruang, Perumahan dan Pemukiman dan Pekerjaan Umum (Keputusan Menteri Permukiman Dan Prasarana Wilayah No. 534/KPTS/M/2001). Pedoman ini mencakup pelayanan minimal untuk pengelolaan sampah secara umum dalam wilayah pemukiman perkotaan dimana 80% dari total jumlah penduduk terlayani terkait dengan pengelolaan sampah. Khusus untuk pengangkutan dicantumkan bahwa jenis alat angkut mempengaruhi pelayanan, sebagai berikut:

- a. Dump truck dengan kapasitas 6 m<sup>3</sup> dapat melayani pengangkutan untuk 700 kk – 1000 kk sedangkan dengan kapasitas 8 m<sup>3</sup> untuk 1500 kk – 2000 kk (jumlah ritasi 2-3/ hari).

- b. Armroll truck dengan kontainer 8 m<sup>3</sup> juga dapat melayani 2000 kk – 3000 kk (jumlah ritasi 3-5/hari). Armroll truck merupakan kendaraan angkut yang dilengkapi sistem hidrolis untuk mengangkat bak dan membongkar muatannya. Pengisian muatan masih tetap secara manual dengan tenaga kerja. Truk ini memiliki kapasitas yang bervariasi, yaitu 6m<sup>3</sup>, 8m<sup>3</sup>, dan 10m<sup>3</sup>. Dalam pengangkutan sampah, efisiensi penggunaan armroll truck dapat dicapai apabila memenuhi beberapa kriteria yaitu jumlah trip atau ritasi perhari minimum 5 dan jumlah kru maksimum 1 orang. Agar tidak mengganggu lingkungan selama perjalanan ke TPA, kontainer sebaiknya memiliki tutup dan tidak rembes

sehingga lindi tidak mudah tercecer. Kontainer yang tidak memiliki tutup sebaiknya dilengkapi dengan tutup terpal selama pengangkutan.

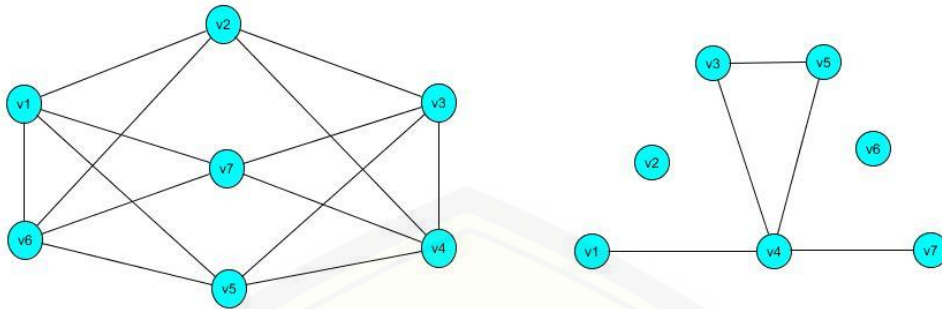
c. Compactor truck 8 m<sup>3</sup> mampu melayani 2500 kk 2. SNI 19-2454-2002, tata cara teknik operasional pengelolaan sampah perkotaan. SNI ini mengatur tentang pola pengangkutan dan operasional pengangkutan.

2. SNI 03-3243-2008, pengelolaan sampah pemukiman. SNI mengatur tentang kebutuhan sarana untuk pengangkutan sampah yang dipengaruhi oleh tipe rumah dan tingkat pelayanan serta jenis alat angkut.

## 2.6 Graf

Sebuah graf  $G$  merupakan pasangan himpunan  $(V(G), E(G))$ , dimana  $V(G)$  adalah himpunan berhingga tak kosong dari elemen yang disebut titik, dan  $E(G)$  adalah sebuah himpunan (mungkin kosong) dari pasangan tak terurut  $\{u, v\}$  dari titik-titik  $u, v \in V(G)$  yang disebut sisi. Titik  $u$  dan  $v$  dimisalkan merupakan titik-titik dari graf  $G$ .  $u$  dikatakan bertetangga dengan  $v$  jika terdapat sebuah sisi  $e$  yang menghubungkan  $u$  dan  $v$ , yaitu  $e = uv$ . Selanjutnya kita sebut  $v$  tetangga dari  $u$ . Himpunan semua tetangga dari  $u$  disebut ketetanggan dari  $u$  dan dinotasikan dengan  $N(u)$ . Kedua titik  $u$  dan  $v$  dapat juga disebut bersisian dengan sisi  $e$  (Slamin, 2009).

Sebuah graf  $G$  disebut terhubung jika untuk sembarang dua titik yang berbeda  $u$  dan  $v$  dari  $G$  terdapat sebuah lintasan antara  $u$  dan  $v$ . Jika tidak maka  $G$  disebut tak terhubung. Contoh graf terhubung dan tidak terhubung terdapat pada gambar 2.1.



Gambar 2. 1 Graf Terhubung dan Graf Tidak Terhubung

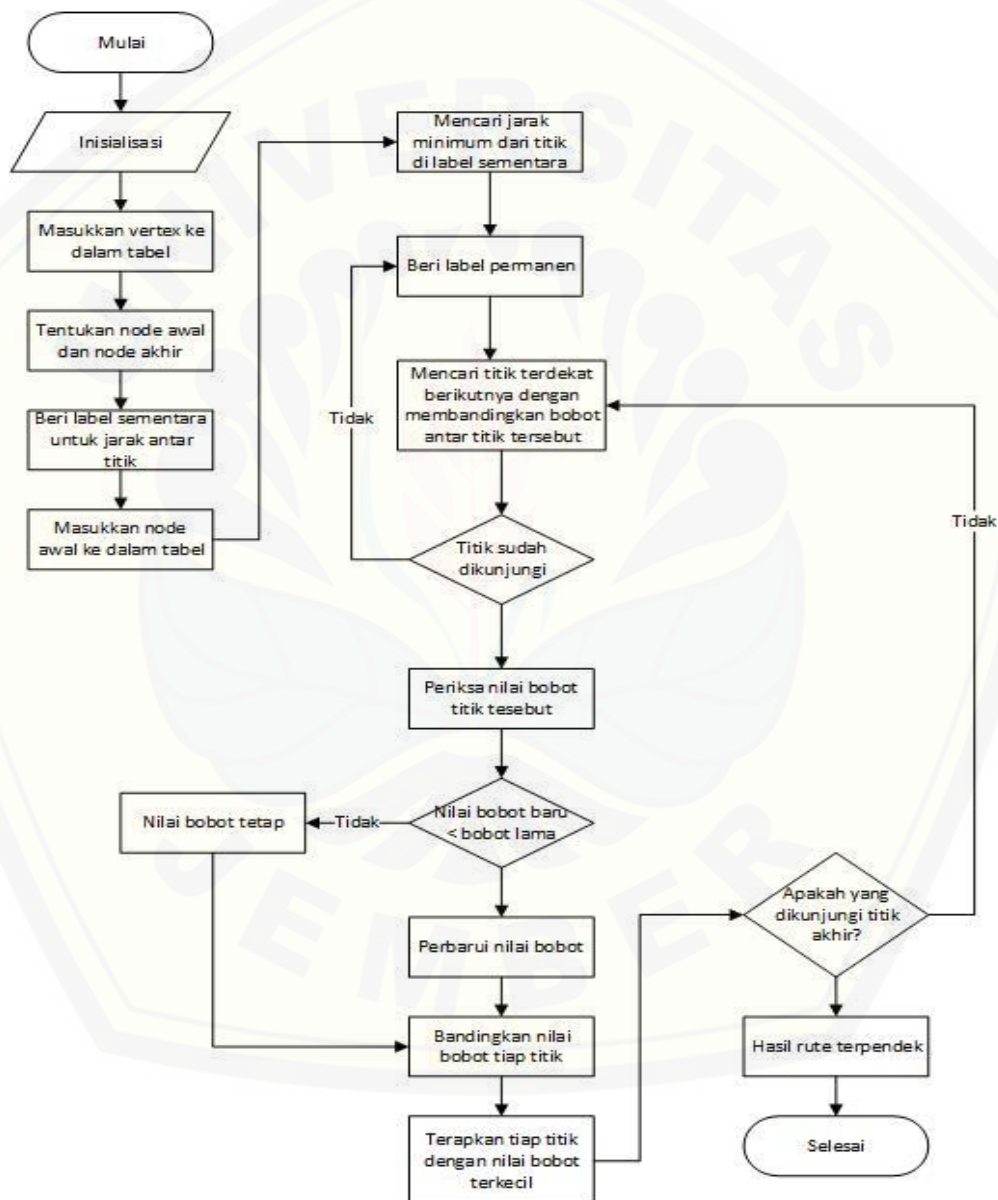
## 2.7 Algoritma Dijkstra

Pada dasarnya, algoritma ini merupakan salah satu bentuk algoritma *greedy*. Algoritma ini termasuk algoritma pencarian graf yang digunakan untuk menyelesaikan masalah lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki *cost* sisi negatif, dan menghasilkan sebuah pohon lintasan terpendek. Algoritma ini sering digunakan pada *routing*. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah (Fakhri, Penerapan Algoritma Dijkstra dalam Pencarian Solusi Maximum Flow Problem, 2008).

Algoritma Dijkstra merupakan salah satu bentuk algoritma *greedy*. Algoritma ini termasuk algoritma pencarian graf yang digunakan untuk menyelesaikan masalah lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki *cost* sisi negatif dan menghasilkan sebuah pohon lintasan terpendek (Fakhri, 2008).

Algoritma Dijkstra menggunakan *adjacent list* untuk merepresentasikan sebuah jaringan. Secara garis besar algoritma dijkstra membagi semua *node* menjadi dua, kemudian dimasukkan ke dalam tabel yang berbeda, yaitu tabel permanen dan tabel temporal. Tabel permanen berisi *node* awal dan *node-node* yang telah melalui proses pemeriksaan dan labelnya telah diubah dari temporal menjadi permanen. Tabel temporal berisi *node-node* yang berhubungan dengan *node* pada tabel permanen (Purwananto & dkk, 2005).

Pemilihan rute dalam algoritma Dijkstra dilakukan dengan *Best First Search* (BFS), menurut T. Sutojo, dkk, *best first search* merupakan kombinasi dari metode *depth first search* dan *breadth first search* dimana pencarian diperbolehkan mengunjungi *node* pada level lebih rendah jika *node* pada level lebih tinggi memiliki nilai heuristik lebih buruk. Dari uraian di atas algoritma Dijkstra dapat di gambarkan ke dalam diagram alir seperti Gambar 2.2



Gambar 2. 2 Diagram Alir Algoritma Dijkstra

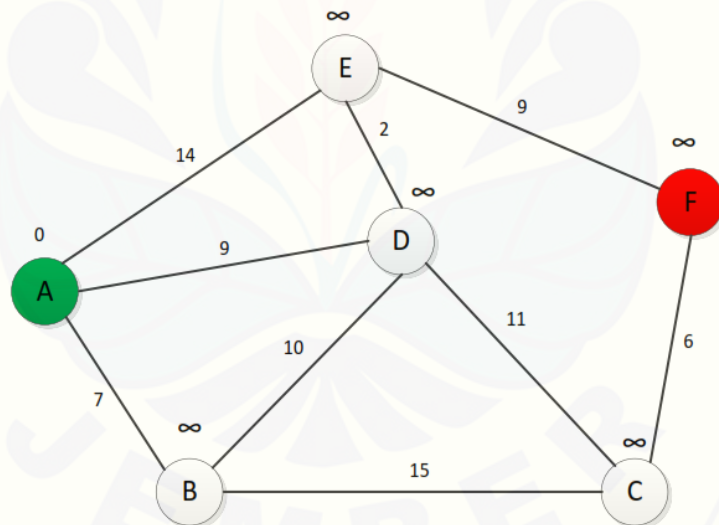
(Sumber: (Gusmao & dkk, 2013))

Sebuah graf berarah yang berbobot (*weighted directed graph*)  $G$  dan sumber *vertices* dalam  $G$  dan  $V$  merupakan himpunan semua *vertices* dalam *graph*  $G$ , dalam hal ini graf berarah yang berbobot dan sumber *vertices* dapat disimpulkan sebagai inputan dari algoritma Dijkstra. Setiap sisi dari graf ini adalah pasangan *vertices*  $(u, v)$  yang melambangkan hubungan dari *vertex*  $u$  ke *vertex*  $v$ . Himpunan semua tepi disebut  $E$  (Wibowo & Wicaksono, 2012).

Untuk menghitung bobot dari semua sisi dapat menggunakan fungsi pada rumus 1.

$$W: E \rightarrow [0, \infty] \dots \dots \dots [2]$$

Contoh bentuk pengimplementasian algoritma Dijkstra pada graf untuk menentukan jalur terpendek dari *node* A ke F pada Gambar 2.3 dan penyelesaiannya pada tabel 2.1.



Gambar 2. 3 Contoh Graf penentuan jalur terpendek

(Sumber: (Roqib, 2014))

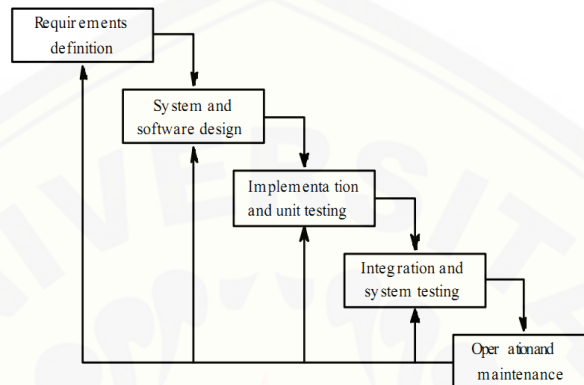
Tabel 2. 1 Penyelesaian Graf dengan menggunakan Algoritma Dijkstra

(Sumber: Raqib, 2014)

Langkah	Vertex sumber	Jarak vertex lain	Jalur pada graf
1.	A	$A \rightarrow B = 7$ $A \rightarrow D = 9$ $A \rightarrow E = 14$	
2.	B	$B \rightarrow C$ $7 + 15 = 22$ $B \rightarrow D$ $7 + 10 = 17$	
3.	D	$D \rightarrow C$ $9 + 11 = 20$ $D \rightarrow E$ $9 + 2 = 11$	
4.	E	$E \rightarrow F$ $11 + 9 = 20$	

## 2.8 Model Waterfall

Model *waterfall* merupakan metode yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test dan pemeliharaan (Sommerville, 2003). Gambaran model *waterfall* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Model Waterfall

(Sommerville, 2003)

Berdasarkan Gambar 2.4 dapat dilihat bahwa dalam penggunaan model *Waterfall* dilakukan secara bertahap dan apabila ada kesalahan maka harus kembali ke tahap sesuai pada Gambar 2.4. Tahap pertama yaitu mengumpulkan kebutuhan-kebutuhan dalam pembuatan sistem, yaitu dengan wawancara atau studi pustaka yang ada. Tahap kedua dengan mendesain sistem yang akan dibuat. Tahap ketiga mulai mengimplementasikan koding ke dalam sistem. Tahap selanjutnya integrasi dan pengujian sistem dan tahap terakhir pemeliharaan sistem.

### **BAB 3. METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tahapan dari penelitian.

#### **3.1 Jenis Penelitian**

Jenis penelitian yang digunakan dalam penelitian ini merupakan penelitian kuantitatif dan kualitatif. Penelitian kuantitatif yang akan dilakukan adalah pada tahap pengumpulan data dan identifikasi kebutuhan yang kemudian didukung dengan adanya penelitian kualitatif pada saat melakukan wawancara kepada semua pihak yang berkaitan dengan adanya sistem pencarian lintasan terpendek menggunakan algoritma Dijkstra.

#### **3.2 Tempat dan waktu penelitian**

Penelitian dilakukan di Kabupaten Jember. Waktu penelitian dilakukan selama 4 (empat) bulan dimulai pada bulan Februari sampai dengan bulan Juni 2017.

#### **3.3 Tahapan Penelitian**

Penelitian mengenai sistem pencarian rute terpendek truk pengangkut sampah ini menggunakan model pengembangan *System Development Life Cycle* (SDLC) *waterfall* seperti yang telah dijelaskan pada bab tinjauan pustaka subbab 2.8. Pengembangan aplikasi ini menggunakan model *waterfall* untuk menyesuaikan dengan kebutuhan aplikasi. Aplikasi yang dibangun merupakan aplikasi dalam skala yang kecil, Sumber Daya Manusia (SDM) yang membangun aplikasi sebanyak 1 orang, dan menyesuaikan dengan *user* yang akan menjadi pengguna aplikasi.



### 3.1.1 Tahapan Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat menganalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data yang didapat oleh peneliti yaitu data TPS dan TPA, koordinat letak tempat pembuangan sampah, rute jalan yang dilalui oleh truk pengangkut sampah. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem. Tahap yang digunakan adalah sebagai berikut:

Penelitian yang dilakukan pada kantor DPU Cipta Karya dan Dinas Kebersihan Kabupaten Jember serta seluruh TPS dan TPA di Kabupaten Jember. Penelitian yang dilaksanakan dalam waktu empat bulan dimulai pada bulan Februari sampai dengan bulan Juni 2017. Tahapan analisis kebutuhan pada penelitian yang telah dilakukan tersebut dimulai dengan tahap pengumpulan data meliputi studi pustaka dan wawancara. Tahapan yang selanjutnya yaitu tahap pengolahan data dan gambaran umum sistem yang akan dibangun.

#### 1. Tahap Pengumpulan Data

Tahap pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan dalam beberapa tahap pengumpulan data sebagai berikut:

##### a. Studi Pustaka

Studi pustaka ini dilakukan dengan tujuan sebagai dasar pembahasan penyusunan dasar teori yang digunakan dalam penelitian. Sumber yang digunakan sebagai studi pustaka berupa buku, jurnal, SOP Dinas Kebersihan, dan karya ilmiah dari penelitian yang sejenis sebelumnya. Data yang didapatkan dalam studi pustaka yaitu penggunaan algoritma Dijkstra dalam penelitian dan pernyataan-pernyataan yang mendukung untuk perancangan sistem ini.

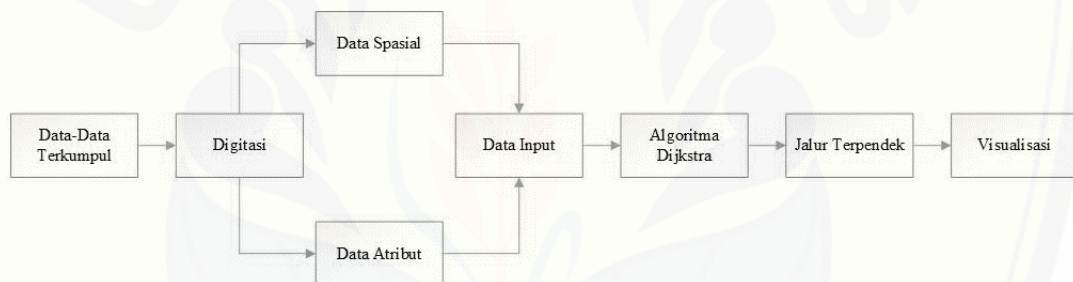
##### b. Wawancara

Wawancara merupakan salah satu cara untuk mengumpulkan data dengan mengajukan pertanyaan langsung kepada narasumber. Dalam

penelitian ini peneliti melakukan wawancara langsung kepada narasumber terkait dengan alur pembuangan sampah di Kabupaten Jember. Para narasumber yang dimaksud, yaitu Kepala Dinas Kebersihan, staf IT Dinas Kebersihan, sopir truk sampah dan staf TPA Pakusari. Dari wawancara yang telah dilakukan diketahui bahwa setiap TPA di Kabupaten Jember telah memiliki daftar tetap untuk TPS-TPS yang bertujuan akhir di TPA tersebut dan data alur kerja proses pembuangan sampah itu sendiri.

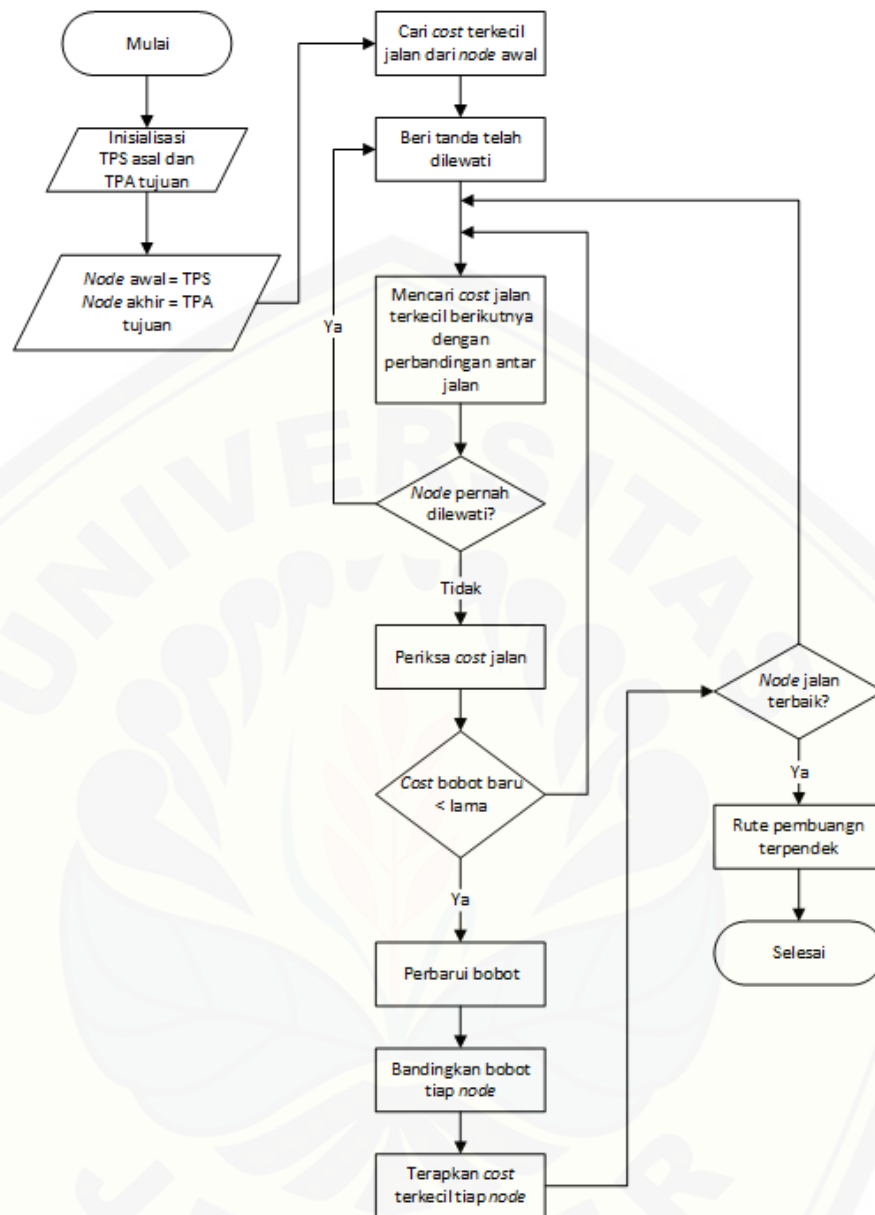
## 2. Tahap Analisis Data

Tahap analisis data dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Langkah selanjutnya yaitu menganalisa data dengan menggunakan algoritma Dijkstra. Tahap analisis data selengkapnya dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Diagram tahap analisis data

Data yang menjadi *input* untuk menentukan jalur terpendek adalah data lokasi TPS dan TPA dan status data jalan yang ada di Kabupaten Jember. Data *input* untuk algoritma Dijkstra ada dua macam, yaitu lokasi titik awal dan lokasi TPA tujuan yang telah ditentukan berdasarkan SOP Dinas Kebersihan. Output dari proses algoritma Dijkstra ini yaitu jalur terpendek dari titik lokasi awal menuju TPA tujuan. Detail proses analisis untuk penentuan jalur terpendek dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Diagram alir penentuan jalur terpendek

### 3. Gambaran Umum Sistem

Sistem yang akan dibuat yaitu aplikasi penentu jalur terpendek yang dapat dilalui oleh truk sampah dalam pembuangan sampah di Kabupaten Jember dengan menggunakan algoritma Dijkstra berbasis web. Aplikasi ini dibangun dengan tujuan proses pembuangan sampah ke Tempat Pembuangan Akhir (TPA) menjadi lebih efektif dan efisien. Sehingga proses pembuangan sampah di Kabupaten Jember lebih teratur dan tidak menyebabkan timbunan sampah di TPS. Aktor

yang terlibat dalam aplikasi ini yaitu *Administrator* yang bertugas untuk manajemen pengguna serta data-data yang ada di dalam sistem, Sopir yang menggunakan sistem dalam memilih jalur terpendek dalam perjalanan menuju TPA Kabupaten Jember, dan TPA yang akan memverifikasi data pembuangan sampah.

### 3.1.2 Tahapan Desain Sistem

Tahap yang selanjutnya yaitu desain sistem menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object-Oriented Design (OOD)*. Pemodelan UML yang akan digunakan sebagai berikut:

#### 1. *Business Process*

*Business process* digunakan untuk menggambarkan masukan data yang dibutuhkan oleh sistem (*input*), keluaran data yang dihasilkan oleh sistem (*output*), media dari sistem (*uses*), dan tujuan dari pembuatan sistem (*goal*). *Input* yang dibutuhkan berupa data TPS dan TPA, pembuangan sampah, data jalan, data titik awal posisi. *Output* yang dihasilkan oleh sistem berupa data jalur terpendek menuju TPA, data rekap pembuangan sampah, data rekap pengguna.

#### 2. *Use Case Diagram*

*Use case diagram* merupakan dokumentasi yang menggambarkan fitur dan aktor yang dapat mengakses fitur tersebut pada sistem yang akan dibangun. Aktor yang dapat mengakses aplikasi yaitu *Administrator*, Sopir dan TPA. Fitur yang disediakan oleh aplikasi yaitu fitur *login*, mengelola data user, mengelola data TPS dan TPA, mengelola data pembuangan sampah, verifikasi data pembuangan sampah dan rekomendasi rute terdekat dalam proses pembuangan sampah ke TPA.

#### 3. *Use Case Scenario*

*Use case scenario* digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *use case diagram* dan menjelaskan keadaan yang akan terjadi pada suatu *event* tertentu. *Use case scenario* menggambarkan keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada suatu kondisi tertentu.

#### 4. *Sequence Diagram*

*Sequence diagram* adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antarobjek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu. Interaksi antarobjek tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada *use case diagram*, tampilan sistem, kontroler, model, dan pesan yang disampaikan jika terjadi suatu *event* tertentu.

#### 5. *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur terjadi, *decision* yang terjadi, dan bagaimana alur berakhir.

#### 6. *Class Diagram*

*Class diagram* adalah sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan kelas, *function*, dan atribut yang dibutuhkan oleh seorang programmer dalam membangun sistem.

#### 7. *Entity Relationship Diagram (ERD)*

*Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antarrelasi.

### 3.1.3 Tahapan Implementasi Sistem

Pada tahap ini desain yang telah dibuat akan diimplementasikan ke dalam kode program. Hal yang dilakukan dalam tahap implementasi antara lain:

- a. Penulisan kode program (*programming*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
- b. Manajemen basisdata menggunakan PostgreSQL.

### 3.1.4 Tahapan Pengujian Sistem

Pengujian aplikasi dilakukan dengan 2 cara, yaitu:

1. *White Box Testing* merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program yang ada. Pengujian *white box* menggunakan *cyclomatic Complexity* (CC) atau biasa disebut juga dengan *conditional complexity*. *Cyclomatic complexity* adalah alat pengukuran untuk mengindikasikan kekompleksitasan dari program dengan cara menelusuri nomor dari jalur yang independen melalui *source codenya*. Dikembangkan oleh Thomas J. McCabe, Sr. pada tahun 1976, *cyclomatic complexity* digunakan di semua fase pada *software lifecycle*, dimulai dari fase desain untuk menjaga agar software dapat dipercaya dan mudah untuk di tes, dan juga agar bisa terorganisir. CC merupakan metrik tradisional yang menghitung tingkat kompleksitas suatu *method/procedure*. Metrik ini bisa diterapkan pada pemrograman berorientasi objek untuk menghitung kompleksitas suatu method. CC secara langsung tidak bisa digunakan untuk mengukur kompleksitas kelas, karena adanya pewarisan dalam code. Namun CC bisa menghitung kompleksitas kelas jika dikombinasikan dengan pengukuran lain. Rumus CC dapat dilihat pada rumus 2.

$$CC = (E - N) + 2 \dots \dots \dots [2]$$

Keterangan:

E= *Edge*

N= *Node*

2. *Black Box Testing* merupakan cara pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal kerja. Pengujian dilaksanakan dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Pengujian *black box* dilakukan oleh *user* yang mengoperasikan aplikasi. Dalam aplikasi ini *user* meliputi Admin, Sopir dan TPA.

### 3.1.5 Tahapan Pemeliharaan Sistem

Sistem yang telah selesai akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan pada sistem.

## BAB 4. PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan sistem pencarian rute terpendek truk pengangkut sampah. Tahap pengembangan sistem dilaksanakan berdasarkan model *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, dan pengujian sistem.

### 4.1 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem merupakan tahapan yang penting dalam pengembangan sebuah sistem informasi. Analisis kebutuhan yang dilakukan sesuai dengan data yang diperoleh pada tahapan yang telah dijelaskan pada bab metodologi penelitian subbab 3.1.1. Seluruh kebutuhan didefinisikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional.

#### 4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang akan dilakukan oleh sistem. Kebutuhan fungsional dari aplikasi ini adalah sebagai berikut:

1. Sistem mampu mengelola data user meliputi (lihat, tambah, ubah, dan hapus).
2. Sistem mampu mengelola data TPS dan TPA meliputi (lihat, tambah, ubah, dan hapus).
3. Sistem mampu mengelola data pembuangan sampah meliputi (lihat dan tambah).
4. Sistem mampu menampilkan hasil rute dengan panjang lintasan terdekat dalam proses pembuangan sampah dari TPS yang dipilih ke TPA yang telah ditentukan.
5. Sistem mampu memverifikasi data pembuangan sampah.

#### 4.1.2 Kebutuhan Non-Fungsional

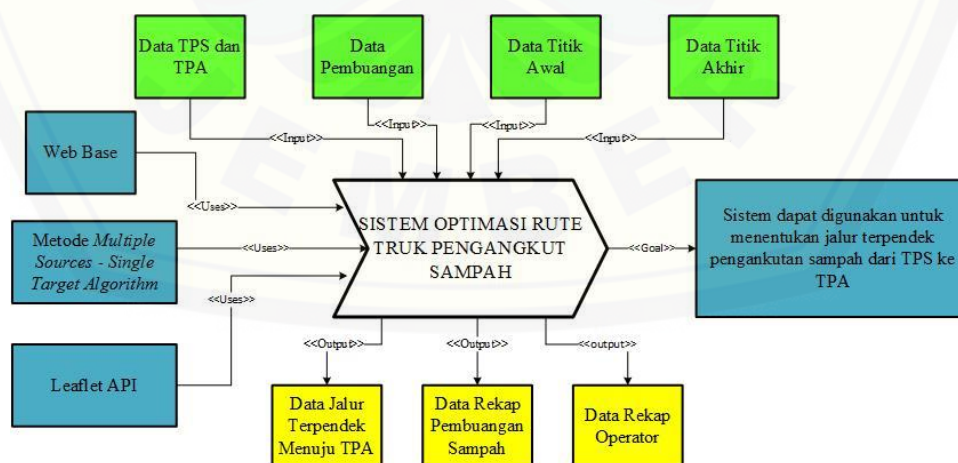
Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan non-fungsional menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non-fungsional aplikasi ini yaitu sistem menggunakan *username* dan *password* untuk autentifikasi akses terhadap sistem.

## 4.2 Desain Sistem

Desain sistem yang dibuat berdasarkan gambaran sistem yang telah dijelaskan pada bab metodologi penelitian subbab 3.1.1. Desain yang dibuat meliputi *business process*, *use case diagram*, *use case scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan ERD.

#### 4.2.1 Business Process

*Business Process* adalah suatu kumpulan aktivitas yang terstruktur untuk mencapai suatu tujuan tertentu atau untuk menghasilkan sebuah produk. Ada beberapa komponen di dalamnya, meliputi data yang menjadi masukan (*input*), data masukan yang kemudian diolah menjadi data keluaran (*output*), media yang digunakan (*uses*), dan tujuan yang ingin dicapai (*goal*). *Business Process* aplikasi ini dapat dilihat pada Gambar 4.1.

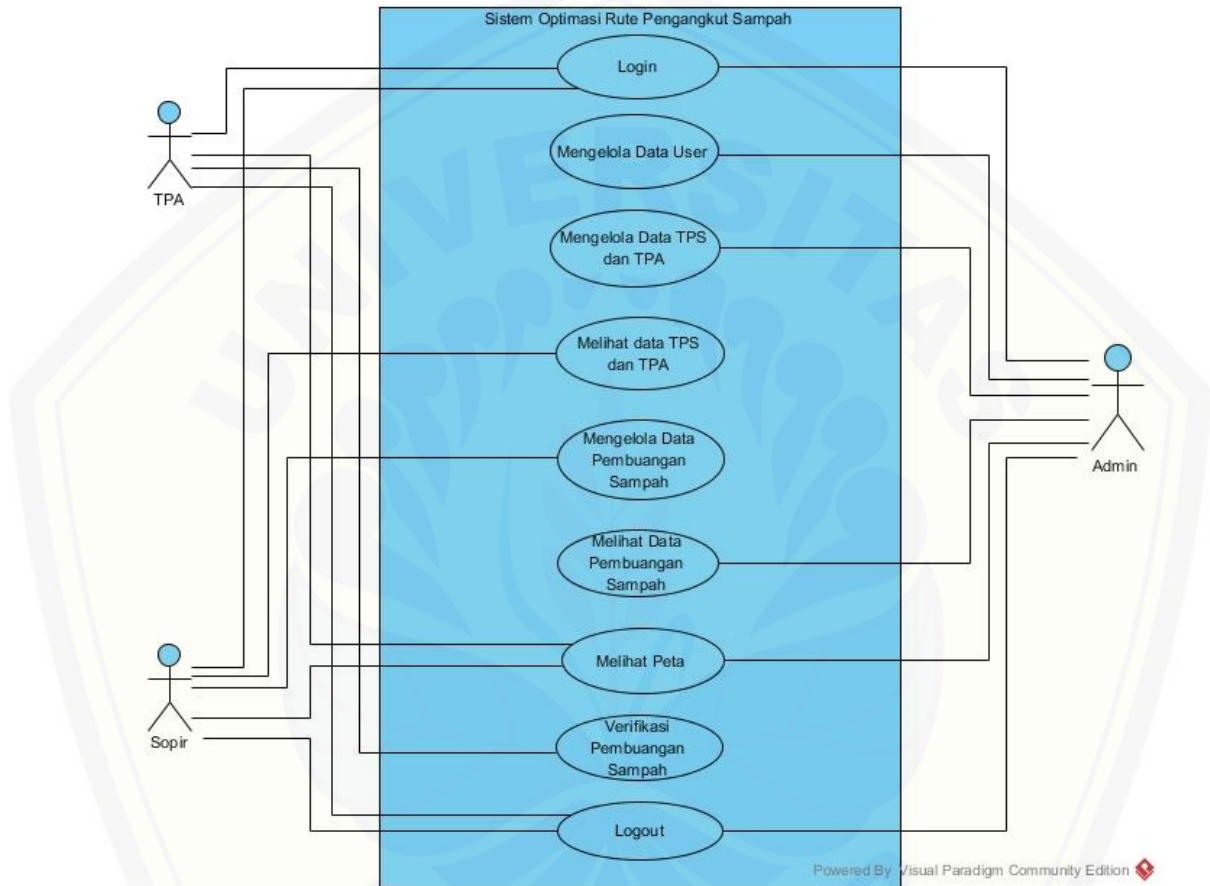


Gambar 4. 1 Business Process



#### 4.2.2 Use Case Diagram

*Use Case Diagram* merupakan dokumentasi yang menggambarkan fitur dan aktor yang dapat mengakses fitur tersebut pada sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember. *Use Case Diagram* aplikasi ini dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Use Case Diagram

Penjelasan tentang definisi aktor dan definisi *use case* dalam *use case diagram* pada Gambar 4.2.

##### 1. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor-aktor sebagai pengguna dari sistem pencarian rute terpendek truk pengangkut sampah Kabupaten Jember yang akan dibangun. Terdapat tiga aktor seperti yang dijelaskan pada Tabel 4.1.

Tabel 4. 1 Definisi Aktor

No.	Aktor	Deskripsi
1.	<i>Admin</i>	Aktor <i>Admin</i> memiliki hak akses secara penuh. <i>Admin</i> dapat melakukan <i>login</i> dan mengelola data sistem secara keseluruhan, meliputi data user, TPS dan TPA, pembuangan sampah.
2.	Sopir	Sopir digeneralisasi sebagai responden yang memiliki hak akses untuk mengisi data pembuangan sampah serta pengelolaan jalur terpendek. Sopir juga memiliki hak akses melihat data TPS dan TPA.
3.	TPA	TPA digeneralisasi sebagai aktor yang berada di lokasi tujuan pembuangan sampah. Memiliki hak akses untuk melakukan verifikasi pada saat proses pembuangan sampah.

## 2. Definisi *Use Case*

Definisi *use case* merupakan penjelasan dari masing-masing *use case* atau fitur-fitur dari sistem pencarian rute terpendek truk pengangkut sampah Kabupaten Jember yang akan dibangun. Terdapat tiga aktor seperti yang dijelaskan pada Tabel 4.2.

Tabel 4. 2 Definisi Use Case

No.	<i>Use Case</i>	Deskripsi
1.	Login	Merupakan alur dari aktor dan sistem jika akan masuk dalam sistem.
2.	Mengelola Data User	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data pengguna.

3.	Mengelola Data TPS dan TPA	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data TPS dan TPA.
4.	Melihat data TPS dan TPA	Menggambarkan proses melihat data TPS dan TPA
5.	Mengelola Data Pembuangan Sampah	Menggambarkan proses mengedit, menghapus, dan melihat data pembuangan sampah pada pengguna <i>Admin</i> .
6.	Melihat data pembuangan sampah	Menggambarkan proses melihat peta yang terdapat marker lokasi-lokasi TPS dan TPA.
7.	Melihat Peta	Menggambarkan proses melihat peta yang berisi kordinat TPS dan TPA dalam marker
8.	Verifikasi Pembuangan Sampah	Menggambarkan proses verifikasi data pembuangan sampah.
9.	Logout	Proses keluar dari sistem

#### 4.2.3 Use Case Scenario

*Use case scenario* digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* seperti pada Gambar 4.2.

##### 1. Skenario *Use Case* Mengelola Data User

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* mengelola data user dapat dilihat pada lampiran A.

##### 2. Skenario *Use Case* mengelola data TPS dan TPA

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* mengelola data user dapat dilihat pada lampiran A.

##### 3. Skenario *Use Case* melihat data TPS dan TPA

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* melihat data TPS dan TPA dapat dilihat pada lampiran A.

#### 4. Skenario *Use Case* mengelola Data Pembuangan Sampah

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan data pembuangan sampah dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Skenario Use Case Pengelolaan Pembuangan Sampah (Sopir)

<b>Nama Use Case</b>	Mengelola Data Pembuangan Sampah
<b>Aktor</b>	Sopir
<b>Deskripsi Singkat</b>	Sopir akan memproses data pembuangan sampah untuk selanjutnya akan diproses dengan menampilkan sebuah peta yang sudah dihubungkan dengan metode Dijkstra
<b>Prekondisi</b>	Data Pembuangan Sampah
<b>Prakondisi</b>	Data Pembuangan Sampah berhasil diproses
<b>Event Flow</b>	
<b>Skenario Normal: Rute Pembuangan Sampah</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik Menu Pembuangan Sampah	
	2. Menampilkan halaman form Pembuangan Sampah meliputi <i>maps</i> , nama, ID, longitude, latitude, tanggal, dan sopir.
3. Mengisi data pembuangan sampah	
4. Klik tombol rute	
	5. Menampilkan jalur pada peta yang digunakan dalam perjalanan

	pembuangan sampah.
<b>Skenario Normal: Tambah Pembuangan Sampah</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
4. Klik tombol simpan	
	5. Menyimpan ke database
<b>Skenario Alternatif: Form Data Tidak Lengkap</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
4a. Klik tombol simpan	
	5a. Menampilkan warning “ <i>please fill out this field</i> ”

5. Skenario *Use Case* Melihat Data Pembuangan Sampah

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* melihat data pembuangan sampah dapat dilihat pada lampiran A.

6. Skenario *Use Case* Verifikasi Data Pembuangan Sampah

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case* verifikasi pembuangan sampah dapat dilihat pada lampiran A.

7. Skenario *Use Case Login*

Penjelas urutan aksi actor dan reaksi sistem pada scenario normal dan skenario alternative scenario *use case Login* dapat dilihat pada lampiran A

8. Skenario *Use Case Log out*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case logout* dapat dilihat pada lampiran A.

#### 4.2.4 Sequence Diagram

*Sequence diagram* adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antarobjek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu pada sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember.

1. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.

2. *Sequence Diagram Mengelola Data User*

Penggambaran *sequence diagram* mengelola data user digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.

3. *Sequence Diagram Mengelola Data TPS dan TPA*

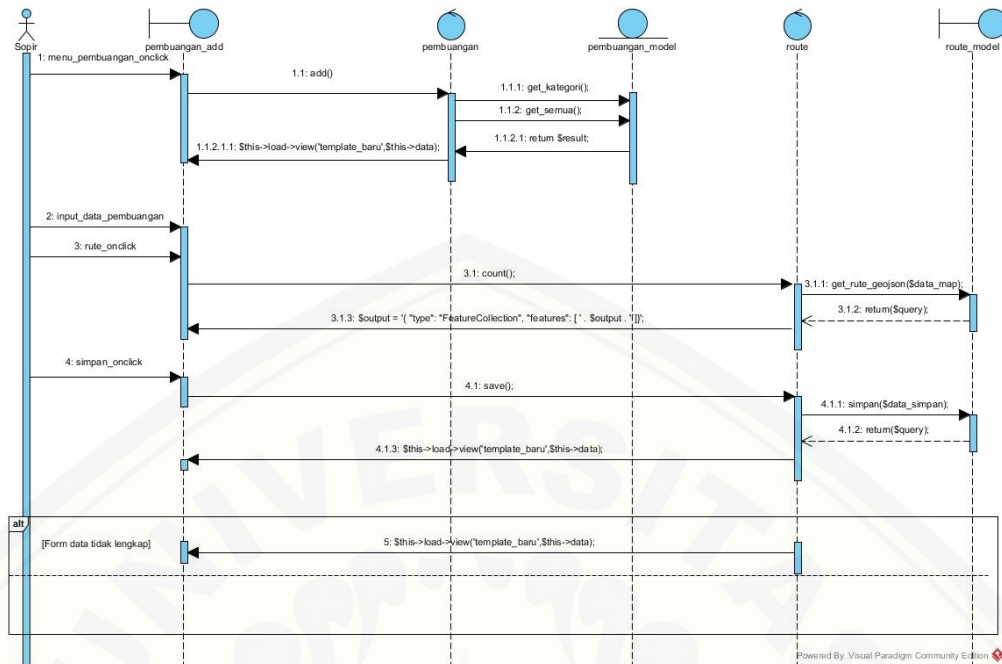
Penggambaran *sequence diagram* mengelola data TPS dan TPA digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3.

4. *Sequence Diagram Melihat Data TPS dan TPA*

Penggambaran *sequence diagram* melihat data TPS dan TPA digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.

5. *Sequence Diagram Mengelola Data Pembuangan Sampah*

Penggambaran *sequence diagram* mengelola data pembuangan sampah digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3



Gambar 4. 3 Sequence Diagram Pengelolaan Pembuangan Sampah

#### 6. *Sequence Diagram* Melihat Data Pembuangan Sampah

Penggambaran *sequence diagram* melihat data pembuangan sampah digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.

#### 7. *Sequence Diagram* Melihat Peta

Penggambaran *sequence diagram* melihat peta digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.

#### 8. *Sequence Diagram* Verifikasi Data Verifikasi Pembuangan Sampah

Penggambaran *sequence diagram* verifikasi data pembuangan sampah digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.

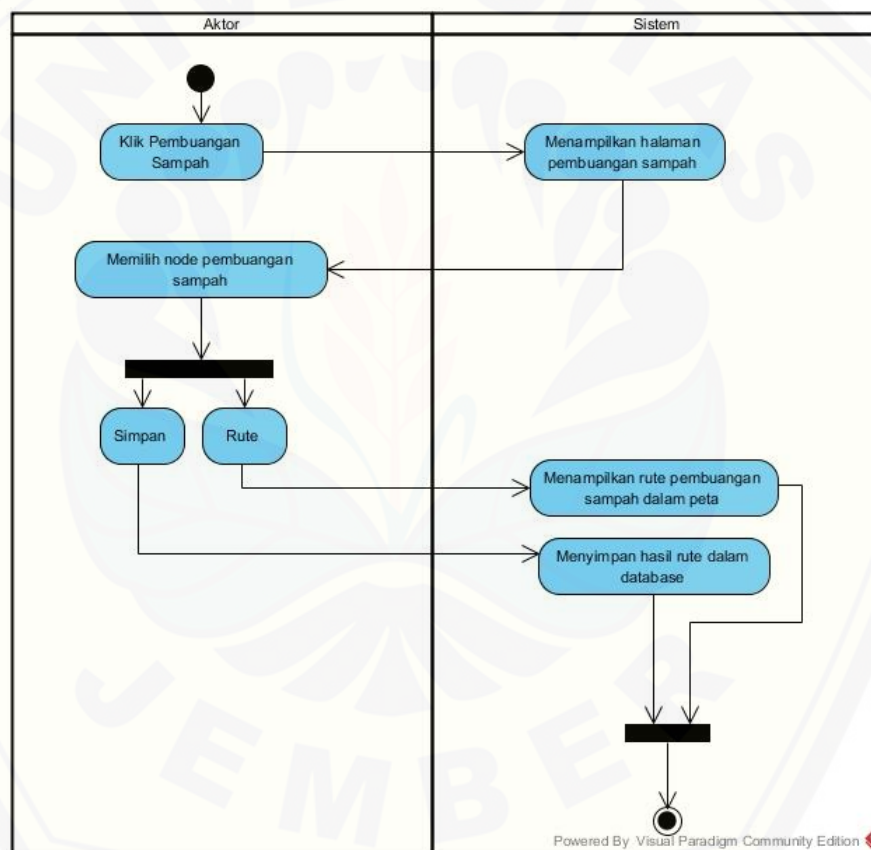
### 4.2.5 *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas pada sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember.

#### 1. *Activity Diagram* Login

*Activity diagram* login dapat dilihat pada lampiran C.

2. *Activity diagram* Mengelola Data User  
*Activity diagram* pengelolaan data User dapat dilihat pada lampiran C.
3. *Activity Diagram* Mengelola Data TPS dan TPA  
*Activity diagram* mengelola TPS dan TPA dapat dilihat pada lampiran C.
4. *Activity Diagram* Melihat Data TPS dan TPA  
*Activity diagram* melihat data TPS dan TPA dapat dilihat pada lampiran C.
5. *Activity Diagram* Mengelola Data Pembuangan Sampah  
*Activity diagram* mengelola data pembuangan sampah dapat dilihat pada Gambar 4.4



Gambar 4. 4 Activity Diagram Mengelola Data Pembuangan Sampah

6. *Activity Diagram* Melihat Data Pembuangan Sampah  
*Activity diagram* melihat data pembuangan sampah dapat dilihat pada lampiran C.
7. *Activity Diagram* Melihat Peta  
*Activity diagram* melihat peta dapat dilihat pada lampiran C.

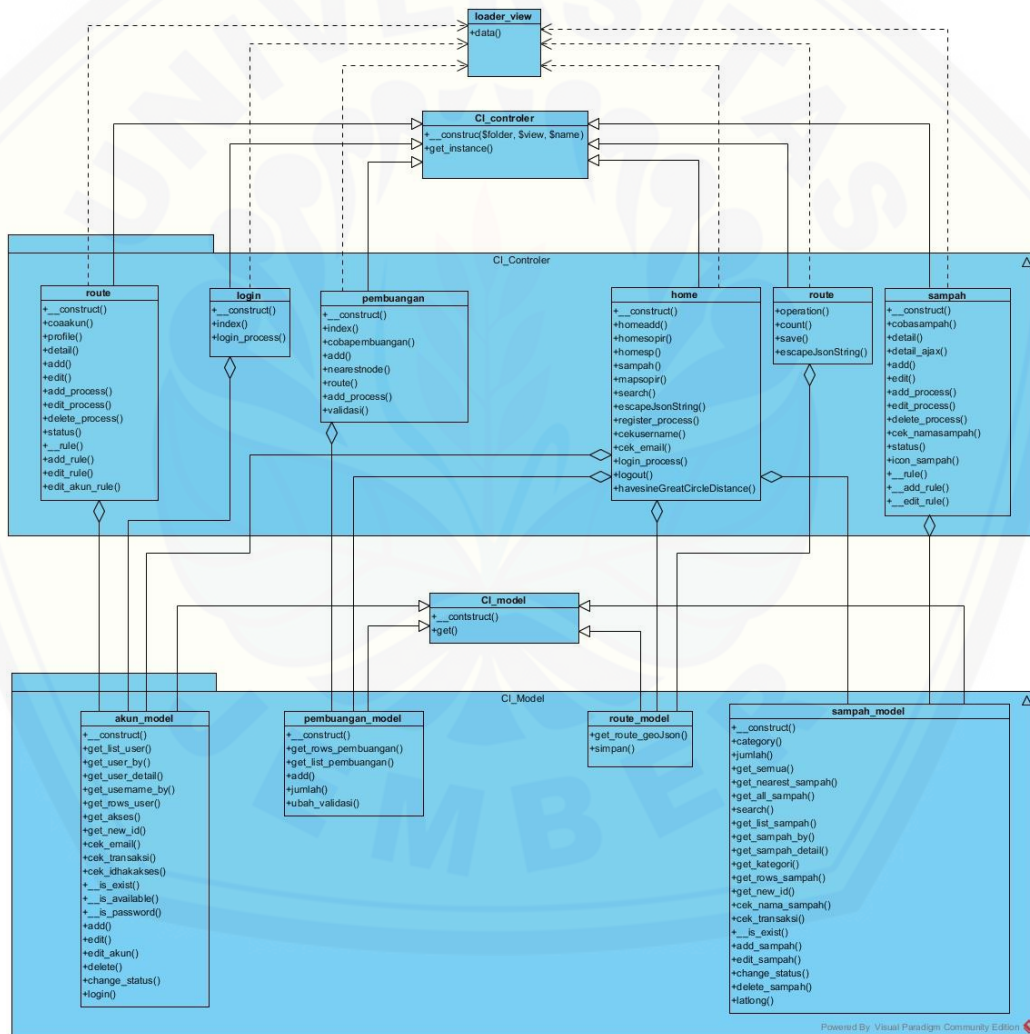


8. Activity Diagram Verifikasi Pembuangan Sampah

Activity diagram verifikasi data pembuangan sampah dapat dilihat pada lampiran C.

4.2.6 Class Diagram

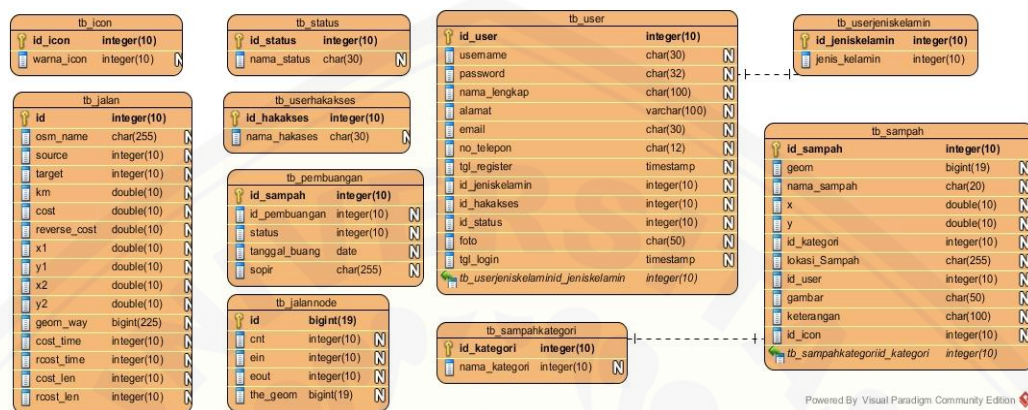
Class Diagram menggambarkan hubungan antarkelas yang digunakan untuk membangun suatu sistem. Class Diagram sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember dilihat pada Gambar 4.5.



Gambar 4. 5 Class Diagram

#### 4.2.7. Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD sistem pencarian rute terpendek truk pengangkut sampah di Kabupaten Jember dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Entity Relation Diagram

## 4.2 Penulisan Kode Program

Desain yang telah dibuat akan diimplemetasikan ke dalam kode program. Penulisan kode program untuk setiap fitur seperti yang telah digambarkan pada *use case diagram* Gambar 4.2 ditulis dalam 3 bagian kelas, meliputi kelas *view*, *controller*, dan model.

### 4.3.1 Kode Program Mengelola Data TPS dan TPA

Kode program pengelolaan data *user* terletak pada kelas *sampah\_list*, *sampah\_add*, *sampah\_edit*, *sampah\_detail*, *controller* *sampah*, dan *sampah\_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

### 4.3.2 Kode Program Melihat Pembuangan Sampah

Kode program melihat data pembuangan terletak pada kelas *pembuangan\_list*, *controller* *pembuangan*, dan *pembuangan\_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

#### 4.3.3 Kode Program Melihat *User*

Kode program pengelolaan data akun terletak pada kelas `akun_add`, `akun_detail`, `akun_edit`, `akun_list`, *controller* akun, dan `akun_model`. Penulisan kode program ini dapat dilihat pada lampiran D.

#### 4.3.4 Kode Program Mengelola Data Pembuangan Sampah

Kode program pengelolaan data pembuangan sampah terletak pada kelas *controller* pembuangan, *controller* route, `pembuangan_model`, dan `route_model`. Penulisan kode program ini dapat dilihat pada Tabel 4.4 sampai dengan Tabel 4.8.

Tabel 4. 4 Kode Program Controller Pembuangan

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Pembuangan extends CI_Controller {
7.	
8.	private \$data;
9.	
10.	function __construct() {
11.	parent::__construct();
12.	\$this->load->model('sampah_model');
13.	\$this->load->model('akun_model');
14.	\$this->load->model('routing_model');
15.	\$this->load->model('pembuangan_model');
16.	
17.	\$this->data = array(
18.	'title' => NULL,
19.	'content' => NULL,
20.	'menu' => NULL,
21.	'akses' => \$this->session->userdata('sampah_hakakses'),
22.	'username' => \$this->session->userdata('sampah_username'),
23.	);
24.	}
25.	
26.	function index() {
27.	\$this->list_pembuangan();
28.	}
29.	
30.	function cobapembuangan(){
31.	\$this->data['menu'] = 1;
32.	\$this->data['title'] = 'Pembuangan Sampah';
33.	\$this->data['content'] = \$this->load->view('pembuangan_list',array(
34.	'data' => \$this->pembuangan_model->get_list_pembuangan(
35.	),TRUE);
36.	\$this->load->view('template_baru',\$this->data);
37.	}
38.	
39.	function add(){
40.	\$this->data['menu'] = 1;
41.	\$this->data['title'] = 'Tambah Pembuangan';
42.	\$this->data['content'] = \$this->load->view('pembuangan_add',array(

```

43.     'kategori' => $this->sampah_model->get_kategori(),
44.     'sampah' => $this->sampah_model->get_semua()
45.     ),TRUE);
46.
47.     $this->load->view('template_baru',$this->data);
48. }
49.
50. function nearest_node() {
51.     $data = array(
52.         'x' => $this->input->post('x'),
53.         'y' => $this->input->post('y')
54.     );
55.
56.     $result = $this->routing_model->get_nearest_coordinate($data);
57.     $result2 = $this->routing_model->get_nearest_network($data);
58.
59.     if ($result2['osm_name'] != "") {
60.         $nama_jalan = $result2['osm_name'];
61.     } else {
62.         $nama_jalan = 'No name';
63.     }
64.
65.     echo $result['lat'] . ' ' . $result['lng'];
66.     echo '|';
67.     echo $nama_jalan;
68. }
69.
70. function route() {
71.     $data = array(
72.         'x1' => $this->input->post('x1'),
73.         'y1' => $this->input->post('y1'),
74.         'x2' => $this->input->post('x2'),
75.         'y2' => $this->input->post('y2'),
76.         'kendaraan' => $this->input->post('kendaraan'),
77.         'jalur' => $this->input->post('jalur'),
78.         'id' => $this->input->post('id')
79.     );
80.
81.     $rs = $this->routing_model->get_route_geojson($data);
82.     $output = "";
83.     $rowOutput = "";
84.     foreach ($rs as $row) {
85.         $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry":
86.         ' . $row['geojson'] . ', "properties": {';
87.         $props = "";
88.         $id = "";
89.         foreach ($row as $key => $val) {
90.             if ($key != "geojson") {
91.                 $props .= (strlen($props) > 0 ? ',' : '') . '"" . $key . '":' . $this-
92.                 >escapeJsonString($val) . '""';
93.             }
94.             if ($key == "id") {
95.                 $id .= ',"id":' . $this->escapeJsonString($val) . '""';
96.             }
97.         }
98.
99.         $rowOutput .= $props . ' }';
100.        $rowOutput .= $id;
101.        $rowOutput .= ' }';
102.        $output .= $rowOutput;
103.    }
104.

```

105.	\$output = '{ "type": "FeatureCollection", "features": [ '. \$output . ' ]}';
106.	echo \$output;
107.	}
108.	
109.	function add_process(){
110.	\$data = array(
111.	'id_sampah' => \$this->input->post('id_sampah'),
112.	'tanggal_buang' => \$this->input->post('tanggal_buang'),
113.	'sopir' => \$this->input->post('sopir')
114.	);
115.	
116.	\$data2 = array(
117.	'lat' => \$this->input->post('lat'),
118.	'long' => \$this->input->post('long')
119.	);
120.	\$hasil = \$this->pembuangan_model->add(\$data);
121.	if (\$hasil) {
122.	redirect('home/mapssopir');
123.	} else {
124.	echo 'gagal';
125.	}
126.	
127.	}
128.	
129.	function validasi(\$id_pembuangan){
130.	\$this->pembuangan_model->ubah_validasi(\$id_pembuangan);
131.	redirect('pembuangan/cobapembuangan');
132.	
133.	}
134.	
135.	}

Tabel 4. 5 Kode Program Kelas Controller Route

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Route extends CI_Controller {
7.	
8.	public function count(){
9.	
10.	\$this->load->model('route_model');
11.	
12.	\$idS = \$this->input->post('iddd');
13.	
14.	// id 1 - 31 ke pakusari
15.	if (\$idS < 32) {
16.	\$data_map = array(
17.	'x1' => '113.759123400001',
18.	'y1' => '-8.169869900001',
19.	'x2' => \$this->input->post('x'),
20.	'y2' => \$this->input->post('y'),
21.	'id' => 'id'
22.	);
23.	
24.	// id 32 - 34 ke ambul
25.	}elseif(\$idS > 31 && \$idS < 35){
26.	\$data_map = array(

```

27.         'x1' => "",
28.         'y1' => "",
29.         'x2' => $this->input->post('x'),
30.         'y2' => $this->input->post('y'),
31.         'id' => 'id'
32.     );
33. }
34.
35. // id 35 - 38 ke tanggul
36. elseif($idS > 34 && $idS < 39){
37.     $data_map = array(
38.         'x1' => "",
39.         'y1' => "",
40.         'x2' => $this->input->post('x'),
41.         'y2' => $this->input->post('y'),
42.         'id' => 'id'
43.     );
44. }
45.
46. // id 39 - 41 ke kecong
47. elseif($idS > 38 && $idS < 42){
48.     $data_map = array(
49.         'x1' => "",
50.         'y1' => "",
51.         'x2' => $this->input->post('x'),
52.         'y2' => $this->input->post('y'),
53.         'id' => 'id'
54.     );
55. }
56.
57. // id 42 - 46 ke balung
58. elseif($idS > 41 && $idS < 47){
59.     $data_map = array(
60.         'x1' => "",
61.         'y1' => "",
62.         'x2' => $this->input->post('x'),
63.         'y2' => $this->input->post('y'),
64.         'id' => 'id'
65.     );
66. }
67.
68. // lebih dari 46 ke pakusari
69. else{
70.     $data_map = array(
71.         'x1' => '113.759123400001',
72.         'y1' => '-8.169869900001',
73.         'x2' => $this->input->post('x'),
74.         'y2' => $this->input->post('y'),
75.         'id' => 'id'
76.     );
77. }
78.
79. $rs = $this->route_model->get_route_geojson($data_map);
80.
81. $output = "";
82. $rowOutput = "";
83. foreach ($rs as $row) {
84.     $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry":
85.     ' . $row['geojson'] . ', "properties": {';
86.     $props = "";
87.     $id = "";
88.     foreach ($row as $key => $val) {

```

89.	if (\$key != "geojson") {
90.	\$props .= (strlen(\$props) > 0 ? ',' : ') . "" . \$key . "" . \$this-
91.	>escapeJsonString(\$val) . "";
92.	}
93.	if (\$key == "id") {
94.	\$id .= ',' . "id": "" . \$this->escapeJsonString(\$val) . "";
95.	}
96.	}
97.	\$rowOutput .= \$props . ' ';
98.	\$rowOutput .= \$id;
99.	\$rowOutput .= ' ';
100.	\$output .= \$rowOutput;
101.	}
102.	\$output = '{ "type": "FeatureCollection", "features": [ ' . \$output . ' ]}';
103.	echo \$output;
104.	}
105.	
106.	public function save()
107.	{
108.	\$this->load->model('route_model');
109.	\$data_simpan = array(
110.	'id_sampah'                    => \$this->input-
111.	>post('id_sampah'),
112.	'tanggal_buang'             => \$this->input-
113.	>post('tanggal_buang'),
114.	'sopir'                     => \$this->input-
115.	>post('sopir')
116.	);
117.	if(\$this->route_model->simpan(\$data_simpan)){
118.	echo "Sukses";
119.	}else{
120.	show_404();
121.	}
122.	}
123.	
124.	function escapeJsonString(\$value) {
125.	\$escapers = array("\\", "/", "\", "\n", "\r", "\t", "\x08", "\x0c");
126.	\$replacements = array("\\\\", "\\/", "\\\"", "\\n", "\\r", "\\t", "\\f", "\\b");
127.	\$result = str_replace(\$escapers, \$replacements, \$value);
128.	return \$result;
129.	}
130.	}

Tabel 4. 6 Kode Program Kelas Pembuangan Model

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Pembuangan_Model extends CI_Model {
7.	
8.	function __construct() {
9.	parent::__construct();
10.	}
11.	
12.	function get_rows_pembuangan() {
13.	return \$this->db->query("select id_pembuangan from tb_pembuangan");
14.	}>num_rows();
15.	}

16.	
17.	function get_list_pembuangan() {
18.	\$usr = \$this->session->userdata('sampah_hakakses');
19.	if(\$usr==1){
20.	\$data = \$this->db->query("select * from
21.	tb_pembuangan a join tb_sampah b on (a.id_sampah = b.id_sampah)
22.	order by id_pembuangan desc
23.	");
24.	}
25.	else {
26.	\$data = \$this->db->query("select * from
27.	tb_pembuangan a join tb_sampah b on (a.id_sampah = b.id_sampah)
28.	order by id_pembuangan desc
29.	");
30.	}
31.	return \$data->result_array();
32.	}
33.	
34.	
35.	function add(\$data){
36.	\$data = \$this->db->query("INSERT INTO public.tb_pembuangan(
37.	id_sampah, tanggal_buang, sopir)
38.	VALUES (
39.	
40.	" . \$data['id_sampah'] . ",
41.	" . \$data['tanggal_buang'] . ",
42.	" . \$data['sopir'] . ")");
43.	
44.	return \$data;
45.	}
46.	
47.	function jumlah(){
48.	return \$this->db->query ("select COUNT(id_pembuangan) as jumlah from
49.	tb_pembuangan");
50.	}
51.	
52.	function ubah_validasi(\$id_pembuangan){
53.	\$this->db->query("
54.	update tb_pembuangan set
55.	status = 1 where id_pembuangan=\$id_pembuangan");
56.	}
57.	}

Tabel 4. 7 Kode Program Kelas Route Model

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Route_model extends CI_Model {
7.	
8.	function get_route_geojson(\$input) {
9.	\$data = \$this->db->query("SELECT seq, st_asgeojson(geom_way) as geojson
10.	FROM pgr_dijkstra(
11.	'SELECT id, source, target, cost_len as cost, rcost_len as reverse_cost FROM
12.	tb_jalan j, tb_jalanjeniskendaraan jk
13.	where j.id = jk.id_jalan',
14.	(select source from tb_jalan j, tb_jalanjeniskendaraan jk where j.id = jk.id_jalan
15.	order by st_distance(st_makepoint(x1, y1), st_makepoint(\$input[x1], \$input[y1]))



16.	limit 1),
17.	(select target from tb_jalan j, tb_jalanjeniskendaraan jk where j.id = jk.id_jalan
18.	order by st_distance(st_makepoint(x2, y2), st_makepoint(\$input[x2], \$input[y2]))
19.	limit 1),
20.	false, true
21.	) d,
22.	tb_jalan j
23.	WHERE d.id2 = j.id");
24.	return \$data->result_array();
25.	}
26.	
27.	function simpan(\$d){
28.	return \$this->db->query("INSERT INTO tb_pembuangan(id_sampah,
29.	tanggal_buang, sopir)
30.	VALUES('\$d[id_sampah]', '\$d[tanggal_buang]', '\$d[sopir]");
31.	}
32.	}

#### 4.3.5 Kode Program Melihat Peta

Kode program melihat peta terletak pada kelas *sampah\_list*, *controller* sampah, dan *sampah\_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

#### 4.3.6 Kode Program Verifikasi Pembuangan Sampah

Kode program verifikasi pembuangan sampah terletak pada kelas *dashboard*, *controller* home, dan *sampah\_model*. Penulisan kode program ini dapat dilihat pada lampiran D.

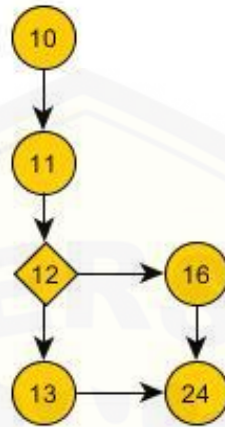
### 4.3 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*.

#### 4.3.1 Pengujian White Box

Pengujian *white box* pada aplikasi penpembuanganan bahan bakar minyak ini dengan cara menggambar diagram alir, menghitung kompleksitas siklomatiknya (CC), dan membuat tabel pengujian *test case*. Pengujian kompleksitas siklomatik pada fitur manajemen akun yang dilakukan oleh *admin* dapat dilihat pada Gambar 4.7 sampai dengan Gambar 4.11 dan tabel pengujian

test case dapat dilihat pada Tabel 4.9 sampai dengan Tabel 4.13, sedangkan untuk fitur yang lain dapat dilihat pada lampiran E.



Gambar 4. 7 Diagram Alir Function \_construct

$$CC = E - N + 2 = 6 - 6 + 2 = 2$$

Tabel 4. 8 Test Case Function \_construct()

Test case	<b>Jika userdata kosong</b>
Target yang Diharapkan	Kembali ke halaman home
Hasil pengujian	Benar
Path/jalur	10-11-12-13-24
Test case	Jika userdata tidak kosong
Target yang Diharapkan	Menampilkan halaman yang sesuai dengan hak akses masing-masing user
Hasil pengujian	Benar
Path/jalur	10-11-12-16-24

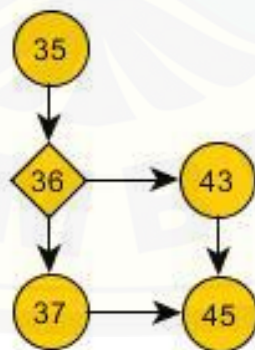


Gambar 4. 8 Diagram Alir Function cobaakun()

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4. 9 Test Case Function cobaakun()

Test case	<b>Jika login berhasil</b>
Target yang Diharapkan	Menampilkan daftar akun
Hasil pengujian	Benar
Path/jalur	26-27-33

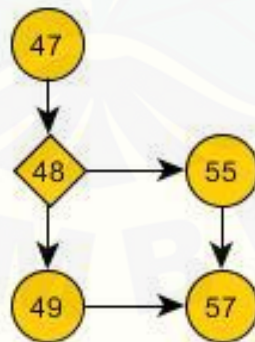


Gambar 4. 9 Diagram Alir Function profil()

$$CC = E - N + 2 = 5 - 5 + 2 = 2$$

Tabel 4. 10 Test Case Function profil()

Test case	<b>Jika berhasil memilih tombol detail</b>
Target yang Diharapkan	Menampilkan profil akun yang dipilih
Hasil pengujian	Benar
Path/jalur	35-36-37-45
Test case	Jika tidak berhasil memilih tombol detail
Target yang Diharapkan	Kembali ke halaman home
Hasil pengujian	Benar
Path/jalur	35-36-43-45

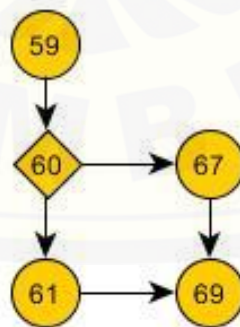


Gambar 4. 10 Diagram Alir Function detail(\$id\_user)

$$CC = E - N + 2 = 5 - 5 + 2 = 2$$

Tabel 4. 11 Test Case Function detail(\$id\_user)

Test case	<b>Jika berhasil memilih tombol detail</b>
Target yang Diharapkan	Menampilkan detail akun yang dipilih
Hasil pengujian	Benar
Path/jalur	47-48-49-57
Test case	Jika tidak berhasil memilih tombol detail
Target yang Diharapkan	Kembali ke halaman home
Hasil pengujian	Benar
Path/jalur	47-48-55-57



Gambar 4. 11 Diagram Alir Function add()

$$CC = E - N + 2 = 5 - 5 + 2 = 2$$

Tabel 4. 12 Test Case Function add()

Test case	<b>Jika berhasil memilih tombol tambah data</b>
Target yang Diharapkan	Menampilkan form tambah data akun
Hasil pengujian	Benar
Path/jalur	59-60-61-69
Test case	Jika tidak berhasil memilih tombol tambah data
Target yang Diharapkan	Kembali ke halaman home/cobakun
Hasil pengujian	Benar
Path/jalur	59-60-67-69

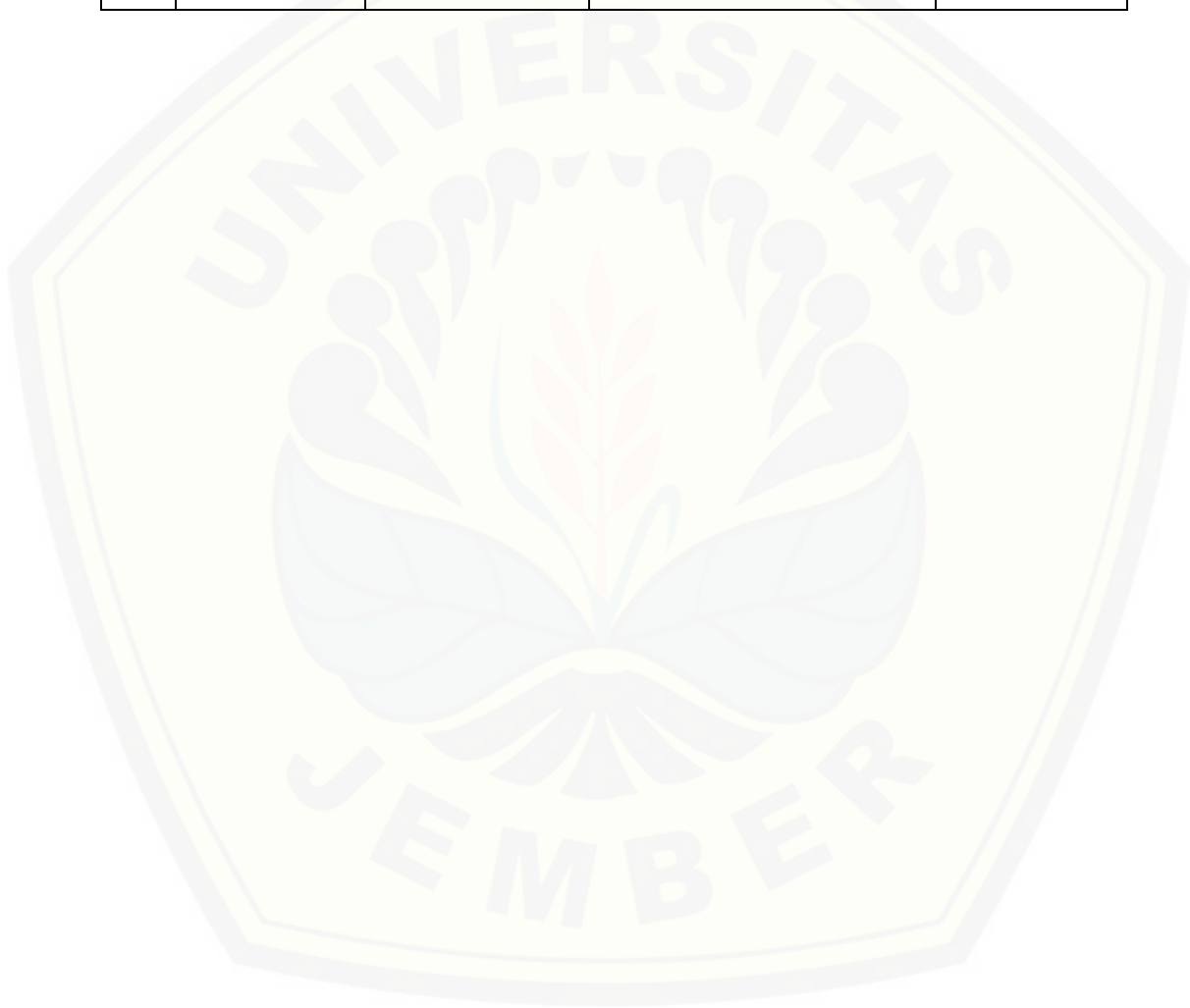
#### 4.3.2 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan oleh *user* dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Pengujian *black box* untuk fitur mengelola pembuangan sampah dapat dilihat pada Tabel 4.14. Sedangkan untuk pengujian fitur yang lain dapat dilihat pada lampiran F.

Tabel 4. 13 Pengujian Black Box Mengelola Pembuangan Sampah (Sopir)

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Mengelola	Memilih menu	Menampilkan form	Berhasil

	pembuangan sampah	Pembuangan Sampah	pembuangan sampah	
		Memilih tombol Rute	Menampilkan jalur terpendek dalam peta	Berhasil
		Memilih tombol Simpan	Menyimpan data ke <i>database</i>	Berhasil



## BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan dari penelitian yang dilakukan adalah:

1. Dibandingkan dengan rute awal manual pembuangan sampah, penerapan Algoritma Dijkstra dalam sistem ini mampu menghasilkan rute terpendek selama proses pengangkutan sampah.
2. Pengembangan rute menggunakan algoritma Dijkstra berdasarkan letak koordinat yang sudah ditentukan lebih pendek daripada rute yang selama ini dilalui oleh armada kebersihan. Sistem dapat menghasilkan rute 3% - 5% lebih pendek dari rute manual yang biasa dilalui armada pengangkut sampah.
3. Hasil pada penentuan rute hanya melewati jalan yang telah dilakukan *tracking*, sehingga truk tidak akan melewati jalan yang kecil. Karena *tracking* dilakukan pada jalan-jalan yang seharusnya dilewati oleh truk sampah.

### 6.2 Saran

Beberapa saran berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu:

1. Diperlukan penyempurnaan dalam *tracking* untuk mendapatkan data jalan dan alternatif rute yang lebih lengkap.
2. Disarankan pada penelitian selanjutnya dapat menginputkan banyak titik koordinat awal dengan tujuan TPA yang sama agar dapat menghasilkan rute yang lebih optimal.

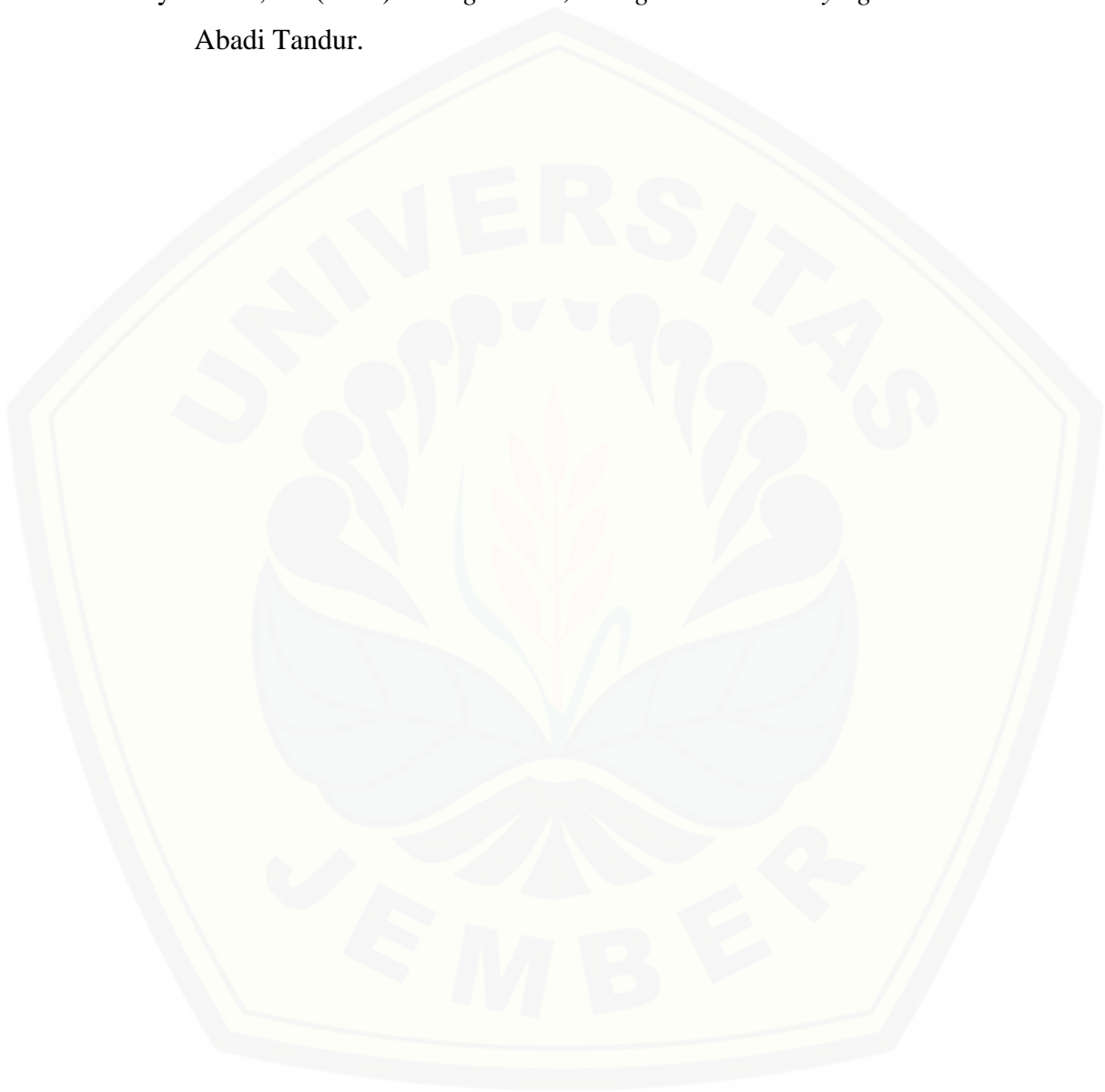


**DAFTAR PUSTAKA**

- Arifianto, S. (2012). Sistem Aplikasi Penentuan Rute Terpendek Pada Jaringan Multi Moda Transportasi Umum Menggunakan Algoritma Dijkstra. *Tesis. Program Studi Magister Sistem Informasi Universitas Diponegoro Semarang.*
- Arifin, E. B. (2006). Pertumbuhan Kota Jember dan Munculnya Budaya Pandhalungan. *Konferensi Nasional Sejarah VIII.*
- Fakhri. (2008). Penerapan Use the "Insert Citation" button to add citations to this document.
- Algoritma Dijkstra Dalam Pencarian Solusi Maximum Flow Problem. *Makalah If2251 Strategi Algoritmik.*
- Gusmao, A., & dkk. (2013). Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra. *Jurnal EECCIS.*
- Purwananto, Y., & dkk. (2005). Implementasi Dan Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya. *Jurnal Penelitian dan Pengembangan TELEKOMUNIKASI.*
- Roqib, A. (2014). Sistem Informasi Geografis Pencarian SPBU Terdekat dan Penentuan Jalur Terpendek Menggunakan Algoritma Dijkstra di Kabupaten Jember Berbasis Web.
- Slamin. (2009). *DESAIN JARINGAN: Pendekatan Teori Graf.* Jember: Jember University Press.
- Sommerville, I. (2003). *Software Engineering (Rekayasa Perangkat Lunak).* Jakarta: Erlangga.

Wibowo, A., & Wicaksono, A. (2012). Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metode Algoritma Dijkstra. *JUITA ISSN : 2086-9398*.

Widyatmoko, S. (2002). *Menghindari, Mengolah dan Menyingkirkan* . Jakarta: Abadi Tandır.



## LAMPIRAN

A. *Use Case Scenario*A.1. Skenario *Use Case* Mengelola Data TPS dan TPA

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* Mengelola Data TPS dan TPA dapat dilihat pada Tabel 1.

Tabel 1 Skenario *Use Case* Pengelolaan Data TPS dan TPA

<b>Nama Use Case</b>	Mengelola Data TPS dan TPA
<b>Aktor</b>	<i>Admin</i>
<b>Deskripsi Singkat</b>	<i>Admin</i> akan mengelola data TPS dan TPA
<b>Prekondisi</b>	Data TPS dan TPA
<b>Prakondisi</b>	1. Data TPS dan TPA berhasil ditambahkan 2. Data TPS dan TPA berhasil diubah 3. Data TPS dan TPA berhasil dihapus
<b>Event Flow</b>	
<b>Skenario Normal: Melihat Data TPS DAN TPA</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu TPS dan TPA	
	2. Menampilkan halaman TPS dan TPA yaitu tabel data TPS dan TPA yang telah dimasukkan yang meliputi no, nama, alamat, kategori, opsi beserta tombol detail, ubah dan hapus
<b>Skenario Normal: Melihat Detail Data TPS dan TPA</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Detail	

	4. Menampilkan halaman detail TPS dan TPA meliputi nama tempat pembuangan, kategori, alamat, keterangan, foto, dan koordinat lokasi tempat pembuangan dalam peta.
<b>Skenario Normal: Tambah Data TPS DAN TPA</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Tambah Data	
	4. Menampilkan form tambah data TPS dan TPA meliputi nama tempat pembuangan, kategori, alamat, latitude, longitude, gambar, dan keterangan.
5. Klik tombol simpan	
	6. Menampilkan halaman TPS dan TPA.
<b>Skenario Alternatif: Form Data Tidak Lengkap</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
5a. Klik tombol Simpan	
	6a. Menampilkan warning “please fill out this field”
<b>Skenario Normal: Batal Tambah Data TPS dan TPA</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
5a. Klik tombol Batal	
	6a. Menampilkan halaman TPS dan TPA.
<b>Skenario Normal: Ubah Data TPS dan TPA</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Ubah	

	4. Menampilkan form ubah data TPS dan TPA meliputi nama tempat pembuangan, kategori, alamat, gambar, latitude, longitude, gambar, dan keterangan.
5. Mengubah form ubah data TPS dan TPA	
6. Klik tombol Simpan	
	7. Menyimpan ke database
	8. Menampilkan halaman manajemen TPS dan TPA
<b>Skenario Alternatif: Form Data Tidak Lengkap</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
6a. Klik tombol Simpan	
	7a. Menampilkan warning “please fill out this field”
<b>Skenario Normal: Memilih Tombol Batal</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
6b. Klik tombol Batal	
	7b. Menampilkan kembali halaman TPS dan TPA
<b>Skenario Normal: Hapus Data TPS dan TPA</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Hapus	
	4. Menghapus data pada database
	5. Menampilkan halaman manajemen TPS dan TPA

### A.2 Skenario *Use Case* Melihat Data Pembuangan Sampah

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan data pembuangan sampah dapat dilihat pada Tabel 2.

Tabel 2 Skenario *Use Case* Melihat Data Pembuangan Sampah

<b>Nama <i>Use Case</i></b>	Melihat Data Pembuangan Sampah
<b>Aktor</b>	<i>Admin</i>
<b>Deskripsi Singkat</b>	<i>Admin</i> akan melihat data pembuangan sampah
<b>Prekondisi</b>	Data pembuangan sampah
<b>Prakondisi</b>	Data pembuangan sampah berhasil dilihat
<b><i>Event Flow</i></b>	
<b>Skenario Normal: Melihat Data Pembuangan Sampah</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu Pembuangan Sampah	2. Menampilkan halaman pembuangan sampah meliputi no, sopir, tujuang, tanggal, status yang telah dimasukkan.

### A.3 Skenario *Use Case* Mengelola *User*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan *user* dapat dilihat pada Tabel 3.

Tabel 3 Skenario *Use Case* Mengelola *User*

<b>Nama <i>Use Case</i></b>	Mengelola Data User
<b>Aktor</b>	<i>Admin</i>
<b>Deskripsi Singkat</b>	<i>Admin</i> akan mengelola data manajemen user

<b>Prekondisi</b>	Data manajemen user
<b>Prakondisi</b>	<ol style="list-style-type: none"> <li>1. Data manajemen user berhasil ditambahkan</li> <li>2. Data manajemen user berhasil diubah</li> <li>3. Data manajemen user berhasil dihapus</li> </ol>
<i>Event Flow</i>	
<b>Skenario Normal: Melihat Data Manajemen User</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu Manajemen User	
	2. Menampilkan halaman user meliputi tabel data user yang terdiri dari no, <i>username</i> , hak akses, status beserta tombol detail, ubah dan hapus
<b>Skenario Normal: Melihat Detail Data Manajemen Akun</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Detail	
	4. Menampilkan halaman detail akun meliputi foto, <i>username</i> , <i>password</i> , nama lengkap, jenis kelamin, alamat, email, no.telepon, dan hak akses.
<b>Skenario Normal: Tambah Data Manajemen User</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Tambah	
	4. Menampilkan form tambah data akun meliputi <i>username</i> , <i>password</i> , nama lengkap, jenis kelamin, no. telepon, email, hak akses, foto, dan

	alamat.
5. Klik tombol Simpan	
	6. Menyimpan ke <i>database</i>
	7. Menampilkan halaman manajemen user.
<b>Skenario Alternatif: Form Data Tidak Lengkap</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
5a. Klik tombol Simpan	
	6a. Menampilkan <i>warning</i> “ <i>please fill out this field</i> ”
<b>Skenario Normal: Batal Tambah Data Manajemen User</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
5a. Klik tombol Batal	
	6a. Menampilkan halaman manajemen user.
<b>Skenario Normal: Ubah Data Manajemen User</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Ubah	
	4. Menampilkan form ubah data akun meliputi <i>username, password, nama lengkap, jenis kelamin, no. telepon, email, hak akses, foto, dan alamat.</i>
5. Mengubah form ubah data manajemen user	
6. Klik tombol Simpan	
	7. Menyimpan ke <i>database</i>
	8. Menampilkan halaman manajemen user
<b>Skenario Alternatif: Form Data Tidak Lengkap</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>



6a. Klik tombol Simpan	
	7a. Menampilkan <i>warning</i> “ <i>please fill out this field</i> ”
<b>Skenario Normal: Memilih Tombol Batal</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
6b. Klik tombol Batal	
	7b. Menampilkan kembali halaman manajemen akun
<b>Skenario Normal: Hapus Data Manajemen User</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
3. Klik tombol Hapus	
	4. Menghapus data pada <i>database</i>
	5. Menampilkan halaman manajemen akun

#### A.4 Skenario *Use Case* Mengelola Pembuangan Sampah

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case* mengelola pembuangan sampah dapat dilihat pada Tabel 4.

Tabel 4 Skenario *Use Case* Mengelola Pembuangan Sampah

<b>Nama <i>Use Case</i></b>	Mengelola Data Pembuangan Sampah
<b>Aktor</b>	Sopir
<b>Deskripsi Singkat</b>	Sopir akan memproses data pembuangan sampah untuk selanjutnya akan diproses dengan menampilkan sebuah peta yang sudah dihubungkan dengan algoritma Dijkstra
<b>Prekondisi</b>	Data Pembuangan Sampah
<b>Prakondisi</b>	Data Pembuangan Sampah berhasil diproses

<i>Event Flow</i>	
<b>Skenario Normal: Rute Pembuangan Sampah</b>	
Aksi Aktor	Reaksi Sistem
4. Klik Menu Pembuangan Sampah	
	5. Menampilkan halaman form Pembuangan Sampah meliputi <i>maps</i> , nama, ID, longitude, latitude, tanggal, dan sopir.
6. Klik tombol rute	
	7. Menampilkan jalur pada peta yang digunakan dalam perjalanan pembuangan sampah.
<b>Skenario Normal: Tambah Pembuangan Sampah</b>	
Aksi Aktor	Reaksi Sistem
6. Klik tombol simpan	
	7. Menyimpan ke <i>database</i>
<b>Skenario Alternatif: Form Data Tidak Lengkap</b>	
Aksi Aktor	Reaksi Sistem
4a. Klik tombol simpan	
	5a. Menampilkan warning " <i>please fill out this field</i> "

#### A.5 Skenario *Use Case* Melihat Peta

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case* melihat peta dapat dilihat pada Tabel 5.

Tabel 5 Skenario *Use Case* Melihat Peta

<b>Nama Use Case</b>	Melihat Peta
<b>Aktor</b>	Admin, Sopir, dan TPA
<b>Deskripsi Singkat</b>	Admin, Sopir, dan TPA akan melihat

	peta tempat pembuangan sampah
<b>Prekondisi</b>	Peta tempat pembuangan sampah
<b>Prakondisi</b>	Peta tempat pembuangan sampah berhasil dilihat
<i>Event Flow</i>	
<b>Skenario Normal: Melihat Peta tempat pembuangan sampah</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu Dashboard	
	2. Menampilkan halaman peta tempat pembuangan sampah beserta marker.

#### A.6 Skenario *Use Case* Verifikasi Data Pembuangan Sampah

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case* verifikasi pembuangan sampah dapat dilihat pada Tabel 6.

Tabel 6 Skenario *Use Case* Verifikasi Data Pembuangan Sampah

<b>Nama <i>Use Case</i></b>	Verifikasi Data Pembuangan Sampah
<b>Aktor</b>	TPA
<b>Deskripsi Singkat</b>	TPA akan melakukan verifikasi data pembuangan sampah
<b>Prekondisi</b>	Data pembuangan sampah belum terverifikasi
<b>Prakondisi</b>	Data pembuangan sampah berhasil diverifikasi
<i>Flow Events</i>	
<b>Skenario Normal: Verifikasi Data Pembuangan Sampah</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu Verifikasi Pembuangan Sampah	

	2. Menampilkan halaman data pembuangan sampah meliputi no, sopir, tujuan, tanggal, status, info dan tombol verifikasi
3. Klik tombol Verifikasi pada data pembuangan yang dipilih	
	4. Menampilkan data pembuangan sampah yang telah terverifikasi

#### A.7 Skenario *Use Case Login*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case login* dapat dilihat pada Tabel 6.

Tabel 7 Skenario *Use Case Login*

<b>Nama Usecase</b>	Login
<b>Aktor</b>	Admin, Sopir, dan TPA
<b>Prakondisi</b>	Halaman awal sistem dan belum masuk
<b>Pascakondisi</b>	<i>Dashboard</i>
<b>Skenario Normal: login</b>	
Aksi Aktor	Reaksi Sistem
1. Masukkan <i>username</i> dan <i>password</i>	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>username</i> dan <i>password</i>
	4. Menampilkan halaman <i>dashboard</i>
<b>Skenario Alternatif: Username dan Password Kosong</b>	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>username</i> dan <i>password</i>
	4. Menampilkan halaman <i>login</i> sistem
<b>Skenario Alternatif: Username atau password salah</b>	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>username</i> dan <i>password</i>

	4. Menampilkan halaman <i>login</i> sistem
--	--

A.8 Skenario *Use Case Log Out*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal, skenario *use case log oout* dapat dilihat pada Tabel 6.

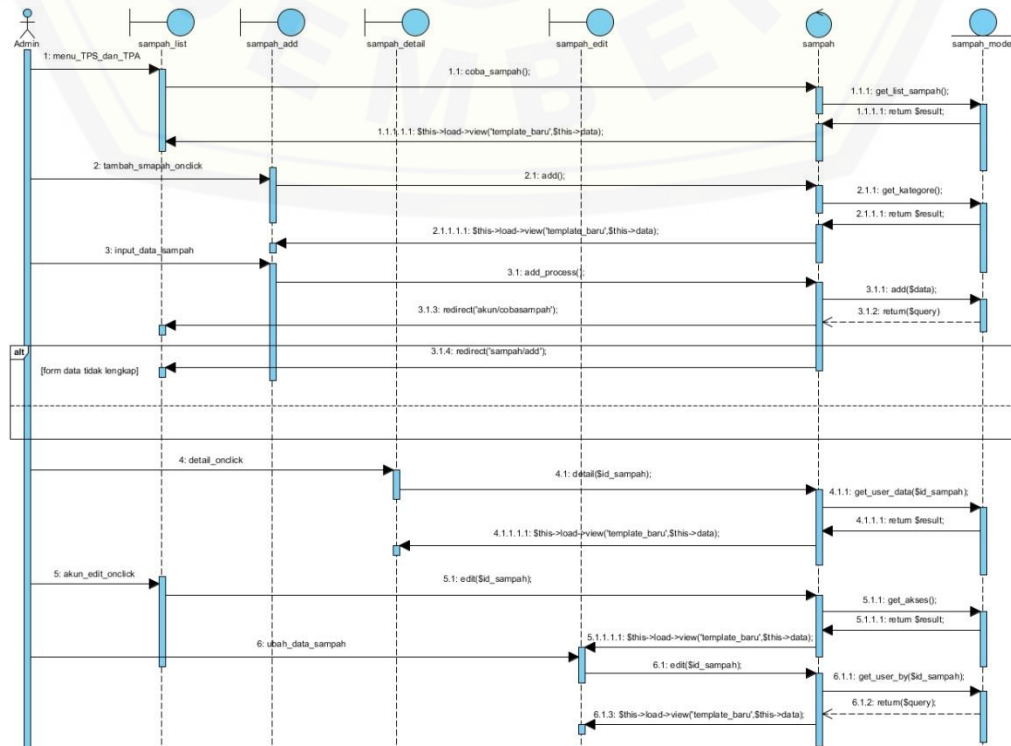
Tabel 8 Skenario *Use Case Log Out*

<b>Nama Usecase</b>	<i>Logout</i>
<b>Aktor</b>	Admin, Sopir, dan TPA
<b>Deskripsi</b>	Fitur untuk keluar dari sistem.
<b>Prakondisi</b>	Halaman <i>Dashboard</i>
<b>Pascakondisi</b>	Berhasil keluar dari sistem dan berada pada halaman <i>login</i>
<b>Skenario Normal: <i>Logout</i></b>	
Aksi Aktor	Reaksi Sistem
1. Klik tombol <i>Log Out</i>	
	2. Menampilkan halaman <i>login</i>

B. *Sequence Diagram*

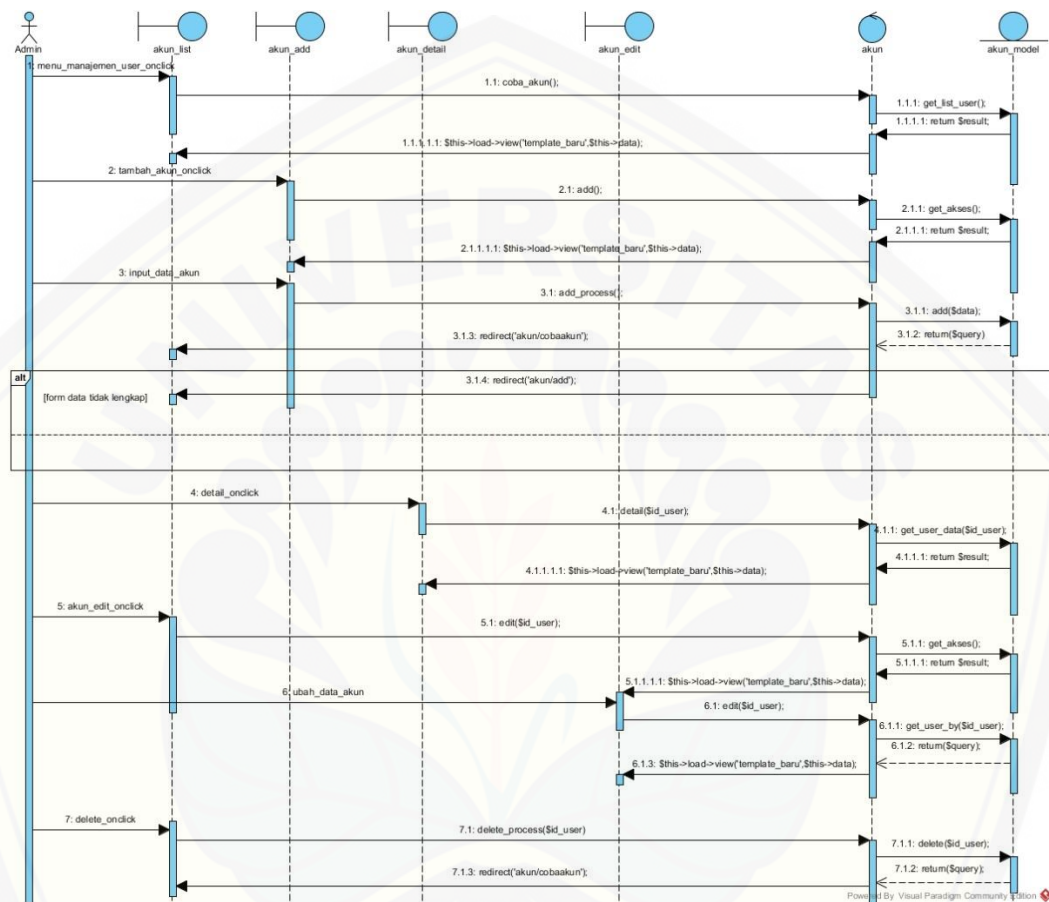
B.1. *Sequence Diagram* Mengelola Data TPS dan TPA

Penggambaran *sequence diagram* mengelola data TPS dan TPA digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 1.



## B.2 Sequence Diagram Mengelola Data User

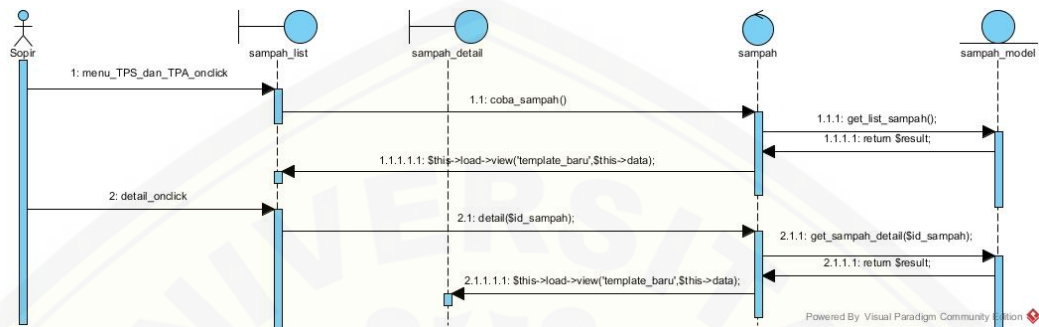
Penggambaran *sequence diagram* mengelola user digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 2.



Gambar 2 Sequence Diagram Mengelola Data User

### B.3 Sequence Diagram Melihat Data TPS dan TPA

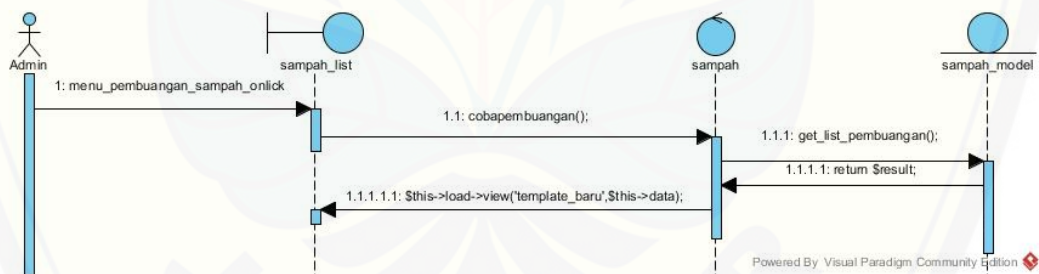
Penggambaran *sequence diagram* melihat data TPS dan TPA digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 3.



Gambar 3 Sequence Diagram Melihat Data TPS dan TPA

### B.4 Sequence Diagram Melihat Data Pembuangan Sampah

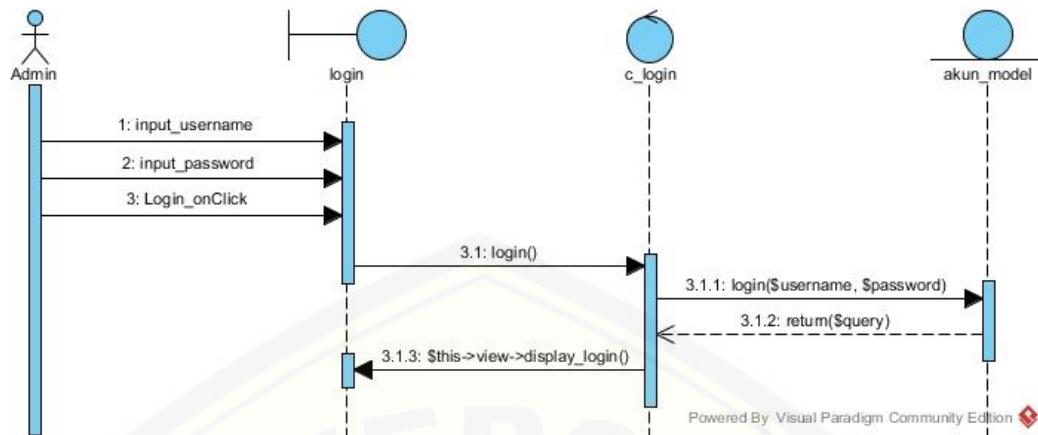
Penggambaran *sequence diagram* melihat pembuangan sampah gunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.



Gambar 4 Sequence Diagram Melihat Data Pembuangan Sampah

### B.5 Sequence Diagram Login

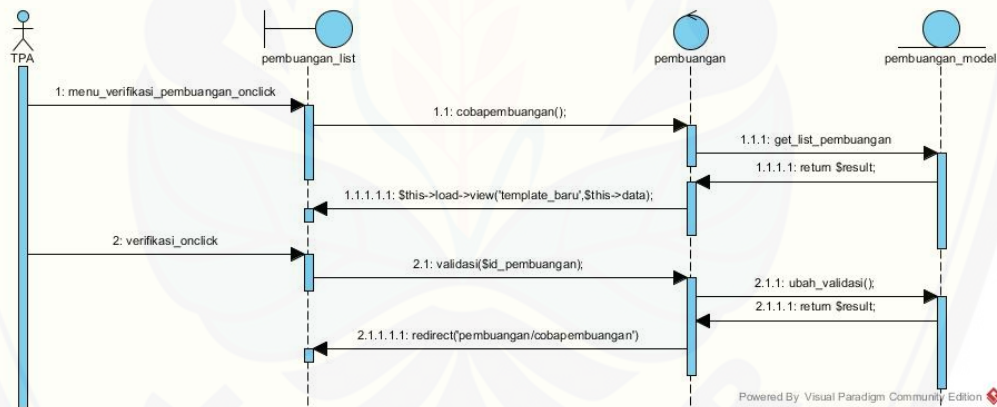
Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 5.



Gambar 5 Sequence Diagram Melihat Peta

### B.6 Sequence Diagram Verifikasi Data Pembuangan Sampah

Penggambaran *sequence diagram* verifikasi pembuangan sampah digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 6.



Gambar 6 Sequence Diagram Verifikasi Data Pembuangan Sampah

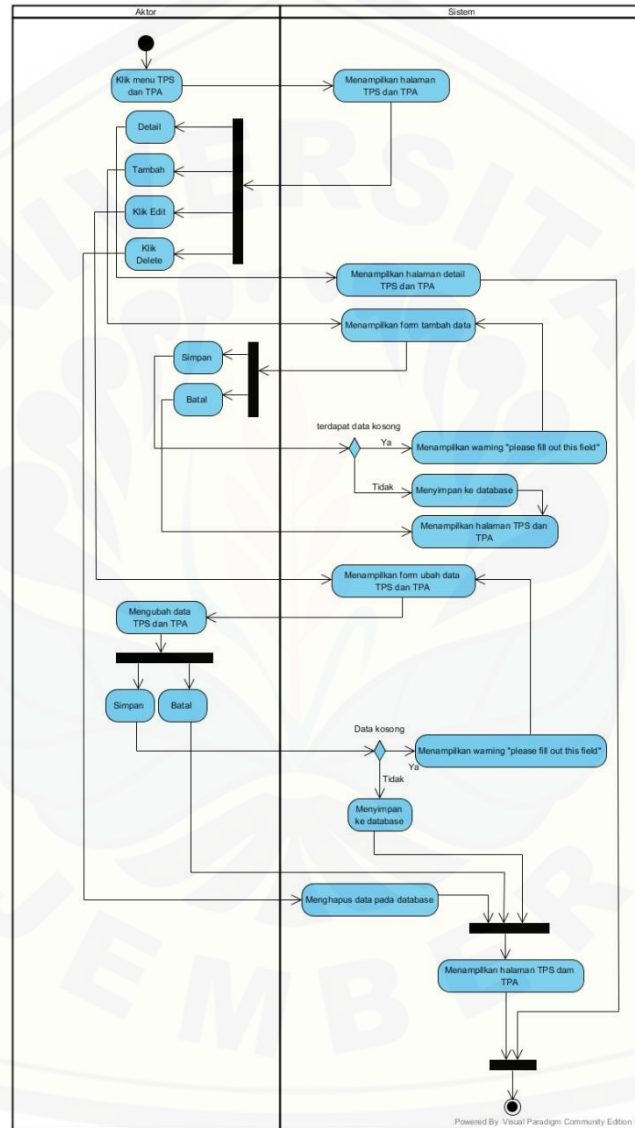


C. Activity Diagram

Activity diagram menggambarkan alur aktivitas pada sistem pencarian rute terpendek truk pengangkut sampah.

C.1 Activity Diagram Mengelola Data TPS dan TPA

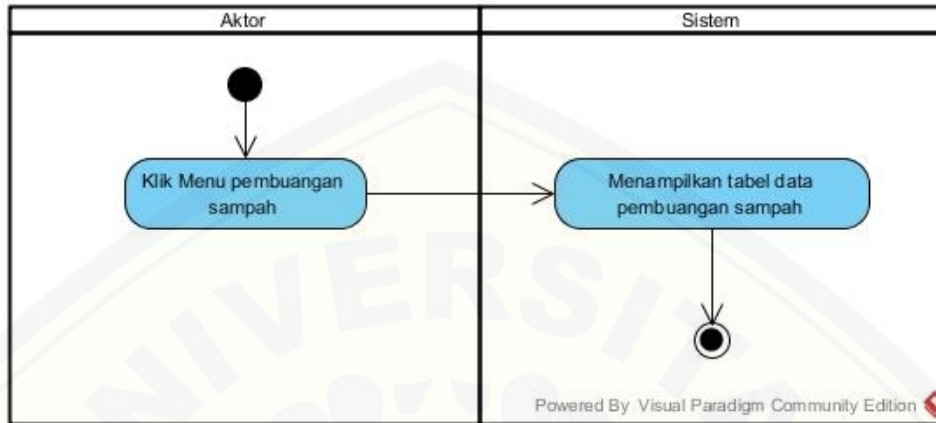
Activity diagram mengelola data TPS dan TPA dapat dilihat pada Gambar 7.



Gambar 7 Activity Diagram Mengelola Data TPS dan TPA

### C.2 *Activity Diagram* Melihat Data Pembuangan Sampah

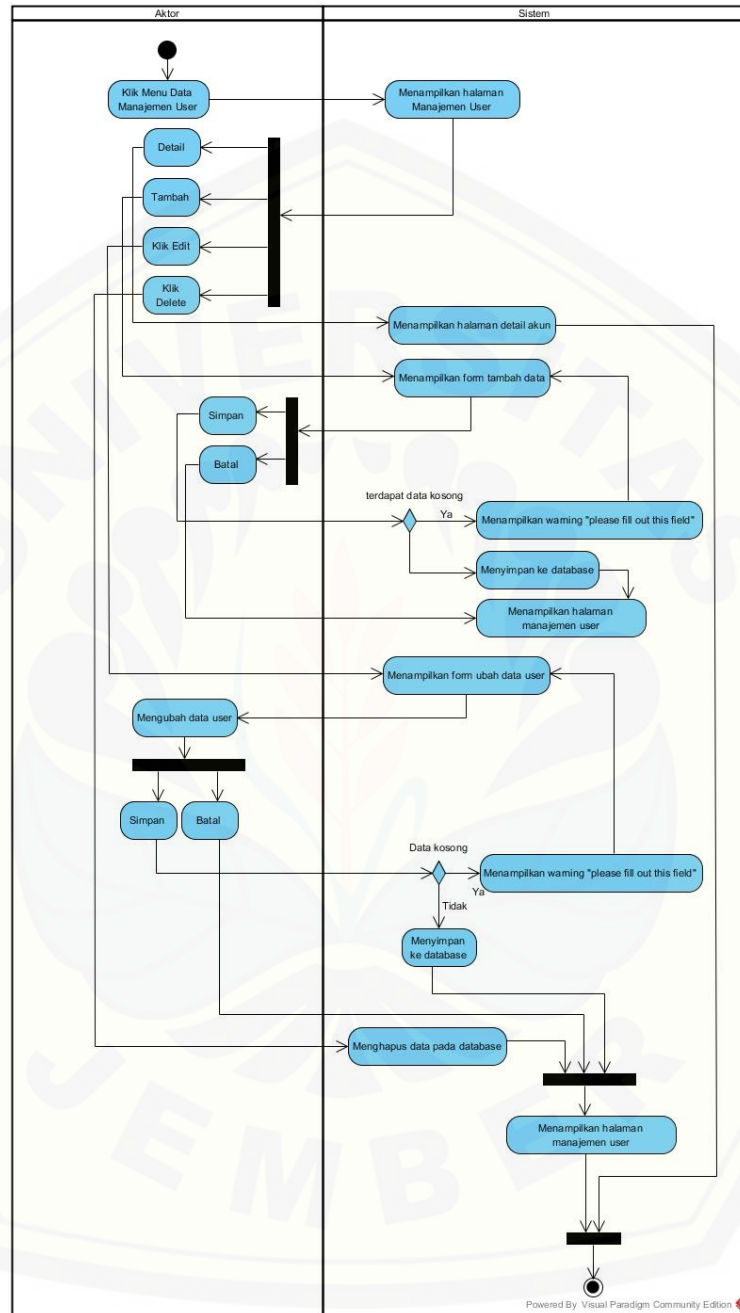
*Activity diagram* melihat data pembuangan sampah dapat dilihat pada Gambar 8.



Gambar 8 *Activity Diagram* Melihat Data Pembuangan Sampah

C.3 Activity Diagram Pengelolaan Data User

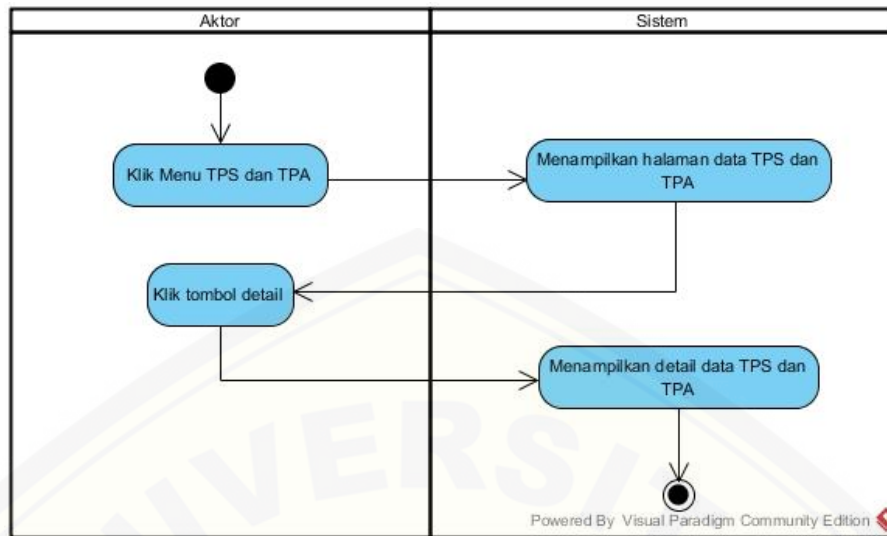
Activity diagram pengelolaan data User dapat dilihat pada Gambar 9.



Gambar 9 Activity Diagram Pengelolaan Data Akun

C.4 Activity Diagram Melihat TPS dan TPA

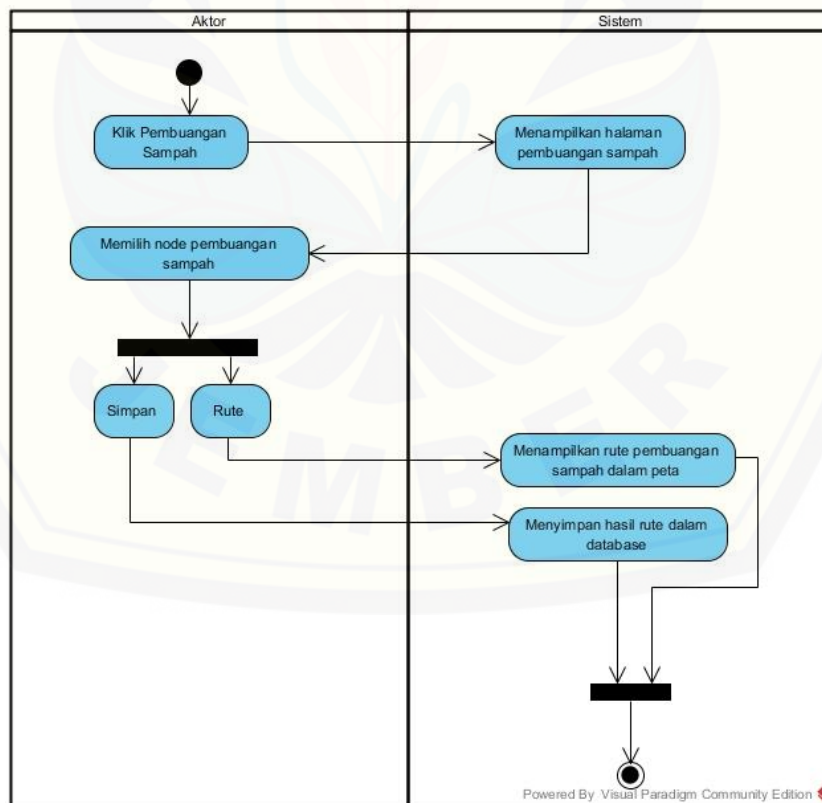
Activity diagram melihat data TPS dan TPA dapat dilihat pada Gambar 10.



Gambar 10 Activity Diagram Melihat Data TPS dan TPA

C.5 *Activity Diagram* Mengelola Data Pembuangan Sampah

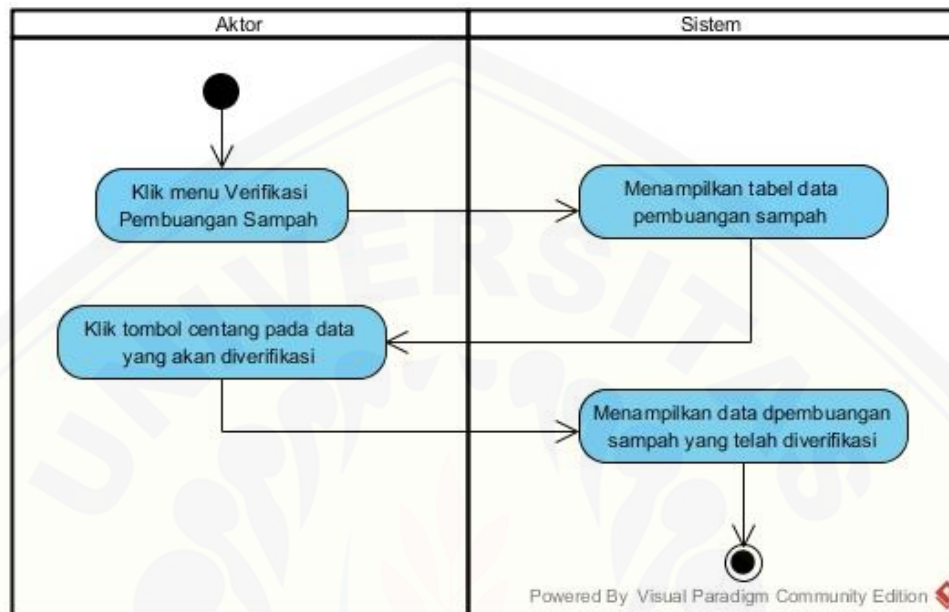
*Activity diagram* mengelola data pembuangan sampah dapat dilihat pada Gambar 11.



Gambar 11 *Activity Diagram* Melihat Peta

### C.6 Activity Diagram Verifikasi Data Pembuangan Sampah

Activity diagram verifikasi data pembuangan sampah dapat dilihat pada Gambar 12.



Gambar 12 Activity Diagram Verifikasi Data Pembuangan Sampah

## D. Kode Program

### D.1 Kode Program Pengelolaan Data Sampah

Kode program pengelolaan data sampah terletak pada kelas samp dan sampah\_model.

#### 1. Kelas sampah

Penulisan kode program sampah dapat dilihat pada Tabel 7.

Tabel 7 Kode Program Kelas sampah

1.	<?php
2.	if (!defined('BASEPATH'))
3.	exit('No direct script access allowed');
4.	class Sampah extends CI_Controller {
5.	
6.	private \$data;
7.	
8.	function __construct() {
9.	parent::__construct();
10.	
11.	\$this->load->model('sampah_model');
	\$this->load->model('akun_model');
12.	
13.	\$this->data = array(
14.	'title' => NULL,
15.	'content' => NULL,
16.	'menu' => NULL,
17.	'akses' => \$this->session->userdata('sampah_hakakses'),
18.	'username' => \$this->session->userdata('sampah_username'),
19.	'nama_lengkap' => \$this->session->userdata('nama_lengkap')
20.	);
21.	}
22.	// memanggil function list sampah untuk ditampilkan di halaman awal manajemen sampah
23.	function cobasampah(){
24.	\$this->data['menu'] = 1;
25.	\$this->data['title'] = 'Manajemen Sampah';
26.	\$this->data['content'] = \$this->load->view('sampah_list',array(
27.	'data' => \$this->sampah_model->get_list_sampah(
28.	),TRUE);
29.	\$this->load->view('template_baru',\$this->data);
30.	}
31.	
32.	
33.	// menampilkan detail ketika user mengklik icon sampah
34.	function detail(\$id) {
35.	\$data_sampah = \$this->sampah_model->get_sampah_by(\$id);
36.	\$this->data['title'] = \$data_sampah['nama_sampah'];
37.	\$this->data['content'] = \$this->load->view(
38.	'sampah_detail', array(
39.	'id_sampah' => \$id,
40.	'data' => \$data_sampah,
41.	), TRUE);
42.	
43.	\$this->load->view('template_baru', \$this->data);

```

44.     }
45. // fungsi ajax untuk menampilkan detail sampah
46.
47.     function detail_ajax() {
48.         $id = $this->input->post('id');
49.         $data_sampah = $this->sampah_model->get_sampah_by($id);
50.         if ($data_sampah['gambar'] == "") {
51.             $gambar = 'no-image.png';
52.         } else {
53.             $gambar = $data_sampah['gambar'];
54.         }
55.
56.         echo '|';
58.         echo '<h1>' . $data_sampah['nama_sampah'] . '</h1>';
59.         echo 'Alamat : ' . $data_sampah['lokasi_sampah'] . '<br/>';
60.         echo 'Kategori : ' . $data_sampah['nama_kategori'] . '<br/>';
61.         echo 'Keterangan : ' . $data_sampah['keterangan'] . '<br/>';
62.         echo '| ' . $rate;
63.     }
64.
65.
66.
67. //menampilkan halaman input sampah
68.     function add() {
69.         if ($this->data['akses'] != NULL) {
70.             $this->data['title'] = 'Tambah Tempat Pembuangan';
71.             $this->data['content'] = $this->load->view(
72.                 'sampah_add', array(
73.                     'kategori' => $this->sampah_model->get_kategori(),
74.                     ), TRUE);
75.             $this->load->view('template_baru', $this->data);
76.         } else {
77.             redirect('home');
78.         }
79.     }
80.
81.
82. //menampilkan halaman edit, dengan mengambil id sampah
83.     function edit($id) {
84.         if ($this->data['akses'] != NULL) {
85.             $this->data['title'] = 'Edit Data Sampah';
86.             $this->data['content'] = $this->load->view('sampah_edit', array(
87.                 'data' => $this->sampah_model->get_sampah_detail($id),
88.                 'kategori' => $this->sampah_model->get_kategori(),
89.                 ), TRUE);
90.             $this->load->view('template_baru', $this->data);
91.         } else {
92.             redirect('home');
93.         }
94.     }
95. // fungsi yang dijalankan dalam proses menambah sampah
96.     function add_process() {
97.         if ($this->data['akses'] != NULL) {
98.             if ($this->__add_rule()) {
99.                 if ($_FILES['image']['name'] == "") {
100.                     $gambar = 1;
101.                 } else {
102.                     $gambar = $_FILES['image']['name'];
103.                 }
104.                 $data = array(
105.                     'nama_sampah' => strtoupper($this->input->post('nama_sampah')),

```

```

103.         'lokasi' => $this->input->post('lokasi'),
104.         'id_kategori' => $this->input->post('kategori'),
105.         'gambar' => $gambar,
106.         'lat' => $this->input->post('lat'),
107.         'long' => $this->input->post('long'),
108.         'keterangan' => $this->input->post('keterangan'),
109.         // 'id_icon' => $this->icon_sampah($this->input->post('kategori'))
110.         'id_icon' => $this->input->post('kategori')
111.     );
112.     if ($data['gambar'] == 1) {
113.         $sukses = $this->sampah_model->add_sampah($data);
114.         if ($sukses) {
115.             $this->session->set_userdata('message', '1,Tempat Pembuangan telah
116. ditambahkan. ');
117.             redirect('sampah/cobasampah');
118.         } else {
119.             $this->session->set_userdata('message', '2,Tempat Pembuangan
120. ditambahkan. ');
121.             redirect('sampah/add');
122.         }
123.     } else {
124.         $config_image = array(
125.             'upload_path' => './images/',
126.             'allowed_types' => 'gif|jpg|png|jpeg',
127.             'max_size' => '500000',
128.             'overwrite' => TRUE,
129.             'remove_spaces' => FALSE,
130.             'file_name' => $_FILES['image']['name']
131.         );
132.
133.         $this->load->library('upload');
134.         $this->upload->initialize($config_image);
135.
136.         if ($this->upload->do_upload('image')) {
137.             $sukses = $this->sampah_model->add_sampah($data);
138.             if ($sukses) {
139.                 $this->session->set_userdata('message', '1,Tempat Pembuangan telah
140. ditambahkan. ');
141.                 redirect('sampah/cobasampah/');
142.             } else {
143.                 $this->session->set_userdata('message', '2,Tempat Pembuangan gagal
144. ditambahkan. ');
145.                 redirect('sampah/cobasampah/');
146.             }
147.         } else {
148.             $this->session->set_userdata('message', '2,Upload gambar gagal. ');
149.             redirect('sampah/add');
150.         }
151.     }
152. } else {
153.     $this->add();
154. }
155. } else {
156.     redirect('home');
157. }
158.
159. }
160. // fungsi menyimpan perubahan dalam proses edit
161. function edit_process() {
162.     if ($this->data['akses'] != NULL) {
163.         if ($this->__edit_rule()) {
164.             if ($_FILES['image']['name'] == "") {

```



```

165.     $gambar = 1;
166.   } else {
167.     $gambar = $_FILES['image']['name'];
168.   }
169.
170.   $data = array(
171.     'id_sampah' => $this->input->post('id_sampah'),
172.     'nama_sampah' => strtoupper($this->input->post('nama_sampah')),
173.     'lokasi' => $this->input->post('lokasi'),
174.     'id_kategori' => $this->input->post('kategori'),
175.     'gambar' => $gambar,
176.     'lat' => $this->input->post('lat'),
177.     'long' => $this->input->post('long'),
178.     'keterangan' => $this->input->post('keterangan'),
179.     'id_icon' => $this->input->post('id_icon'));
180.   );
181.
182.   if ($data['gambar'] == 1) {
183.     $sukses = $this->sampah_model->edit_sampah($data);
184.     if ($sukses) {
185.       $this->session->set_userdata('message', '1, Data sampah telah diperbarui. ');
186.       redirect('sampah/cobasampah/' . $this->input->post('id_sampah'));
187.     } else {
188.       $this->session->set_userdata('message', '2, Data sampah gagal diperbarui. ');
189.       redirect('sampah/edit/' . $this->input->post('id_sampah'));
190.     }
191.   } else {
192.     $config_image = array(
193.       'upload_path' => './images/',
194.       'allowed_types' => 'gif|jpg|png|jpeg',
195.       'max_size' => '500000',
196.       'overwrite' => TRUE,
197.       'remove_spaces' => FALSE,
198.       'file_name' => $_FILES['image']['name']
199.     );
200.     $this->load->library('upload');
201.     $this->upload->initialize($config_image);
202.     if ($this->upload->do_upload('image')) {
203.       $sukses = $this->sampah_model->edit_sampah($data);
204.       if ($sukses) {
205.         $this->session->set_userdata('message', '1, Data sampah telah diperbarui. ');
206.         redirect('sampah/edit/' . $this->input->post('id_sampah'));
207.       } else {
208.         $this->session->set_userdata('message', '2, Data sampah gagal diperbarui. ');
209.         redirect('sampah/edit/' . $this->input->post('id_sampah'));
210.       }
211.     } else {
212.       $this->session->set_userdata('message', '2, Upload gambar gagal. ');
213.       redirect('sampah/edit/' . $this->input->post('id_sampah'));
214.     }
215.   }
216.   } else {
217.     $this->edit($this->input->post('id_sampah'));
218.   }
219.   } else {
220.     redirect('home');
221.   }
222. }
223. // proses menghapus id berdasarkan sampah yang dipilih
224. function delete_process($id) {
225.   if ($this->data['akses'] != NULL) {
226.     if ($this->sampah_model->cek_transaksi($id) == TRUE) {

```

```

227.     $sukses = $this->sampah_model->delete_sampah($id);
228.     if ($sukses == true) {
229.         $this->session->set_userdata('message', '1, Sampah telah dihapus.');
```

230. redirect('sampah/cobasampah');

231. } else {

232. \$this->session->set\_userdata('message', '2, Sampah gagal dihapus.');

233. redirect('sampah/list\_sampah/');

234. }

235. } else {

236. \$this->session->set\_userdata('message', '2,Telah ada transaksi dengan tabel lain.

237. Sampah tidak boleh dihapus.');

238. redirect('sampah/list\_sampah/');

239. }

240. } else {

241. redirect('home');

242. }

243. }

244. // fungsi dalam tambah sampah, untuk mengecek adakah nama yang sama

245. function cek\_namasampah() {

246. \$validateValue = \$\_REQUEST['fieldValue'];

247. \$validateId = \$\_REQUEST['fieldId'];

248.

249. \$arrayToJs = array();

250. \$arrayToJs[0] = \$validateId;

251.

252. \$cek = \$this->sampah\_model->cek\_namasampah(\$validateValue);

253. if (\$cek == true) {

254. \$arrayToJs[1] = true;

255. echo json\_encode(\$arrayToJs);

256. } else {

257. for (\$x = 0; \$x < 1000000; \$x++) {

258. if (\$x == 990000) {

259. \$arrayToJs[1] = false;

260. echo json\_encode(\$arrayToJs);

261. }

262. }

263. }

264. }

265.

266.

267. function icon\_sampah(\$input){

268. \$icon = 1;

269. for (\$i = 0; \$i < count(\$input); \$i++) {

270. if (\$input[\$i] == 2) {

271. \$icon += 1;

272. }

273. if (\$input[\$i] == 4) {

274. \$icon += 2;

275. }

276. }

277. return \$icon;

278. }

279.

280. private function \_\_rule(\$rule) {

281. \$this->load->library('form\_validation');

282. \$this->form\_validation->set\_rules(\$rule);

283. if (\$this->form\_validation->run()) {

284. return TRUE;

285. } else {

286. return FALSE;

287. }

288. }

```
289. // menambahkan rule baru ke dalam peta dan algoritma
290. private function __add_rule() {
291.     $rule = array
292.     (
293.         array
294.         (
295.             'field' => 'nama_sampah'
296.             , 'label' => 'nama sampah'
297.             , 'rules' => 'required'
298.         )
299.     , array
300.     (
301.         'field' => 'lat'
302.         , 'label' => 'latitude'
303.         , 'rules' => 'trim|required'
304.     )
305.     , array
306.     (
307.         'field' => 'long'
308.         , 'label' => 'longitude'
309.         , 'rules' => 'trim|required'
310.     )
311.     , array
312.     (
313.         'field' => 'lokasi'
314.         , 'label' => 'lokasi'
315.         , 'rules' => 'trim|required'
316.     )
317. );
318. return $this->__rule($rule);
319. }
320. // fungsi edit rule, ketika sampah diedit
321. private function __edit_rule() {
322.     $rule = array
323.     (
324.         array
325.         (
326.             'field' => 'nama_sampah'
327.             , 'label' => 'nama sampah'
328.             , 'rules' => 'required'
329.         )
330.     , array
331.     (
332.         'field' => 'lat'
333.         , 'label' => 'latitude'
334.         , 'rules' => 'trim|required'
335.     )
336.     , array
337.     (
338.         'field' => 'long'
339.         , 'label' => 'longitude'
340.         , 'rules' => 'trim|required'
341.     )
342.     , array
343.     (
344.         'field' => 'lokasi'
345.         , 'label' => 'lokasi'
346.         , 'rules' => 'trim|required'
347.     )
348. );
349. return $this->__rule($rule);
350. }
```

351.	
352.	}
353.	
354.	?>
355.	

## 2. Kelas sampah\_model

Penulisan kode program sampah\_model dapat dilihat pada Tabel 8.

Tabel 8 Kode Program Kelas sampah\_model

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Sampah_Model extends CI_Model {
7.	
8.	function __construct() {
9.	parent::__construct();
10.	}
11.	
12.	function category(\$id) {
13.	if (\$id != 0) {
14.	\$kategori = " and tb_sampah.id_kategori = \$id ";
15.	} else {
16.	\$kategori = "";
17.	}
18.	return \$kategori;
19.	}
20.	
21.	function jumlah(){
22.	return \$this->db->query ("select COUNT(id_sampah) as jumlah from tb_sampah
23.	where id_kategori = 1");
24.	}
25.	
26.	function get_semua(){
27.	return \$this->db->query("select * from tb_sampah")->result_array();
28.	}
29.	
30.	
31.	// fungsi untuk menampilkan pom terdekat
32.	
33.	function get_nearest_sampah(\$input) {
34.	\$kategori = \$this->category(\$input['kategori']);
35.	\$data = \$this->db->query("select distinct(tb_sampah.id_sampah), nama_sampah, x,
36.	y, tb_sampah.id_kategori, gambar, id_icon, st_distance_spheroid(geom,
37.	st_setsrid(st_makepoint(\$input[x],\$input[y]), 4326),
38.	'SPHEROID[\"WGS84\",6378137,298.25728])/ 1000 as jarak, st_asgeojson(geom) as
39.	geojson
40.	from tb_sampah, tb_sampahKategori
41.	where tb_sampah.id_kategori = tb_sampahKategori.id_kategori
42.	and tb_sampah.id_status = 1
43.	and st_distance_spheroid(geom, st_setsrid(st_makepoint(\$input[x],\$input[y]),
44.	4326), 'SPHEROID[\"WGS84\",6378137,298.25728])/1000 <= \$input[radius] \$kategori
45.	\$produk \$layanan
46.	order by st_distance_spheroid(geom,
47.	st_setsrid(st_makepoint(\$input[x],\$input[y]), 4326),

```

48. 'SPHEROID["WGS84",6378137,298.25728])/1000 asc");
49.     return $data->result_array();
50. }
51. // menampilkan semua koordinat sampah
52. function get_all_sampah($input) {
53.     $kategori = $this->category($input['kategori']);
54.     $data = $this->db->query("select distinct(tb_sampah.id_sampah), nama_sampah, x,
55. y, tb_sampah.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
56.     from tb_sampah, tb_sampahkategori
57.     where tb_sampah.id_kategori = tb_sampahKategori.id_kategori
58.     $kategori");
59.     return $data->result_array();
60. }
61.
62. function search($input) {
63.     $kategori = $this->category($input['kategori']);
64.
65.     $data = $this->db->query("select distinct(tb_sampah.id_sampah), nama_sampah, x,
66. y, tb_sampah.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
67.     from tb_sampah, tb_sampahkategori
68.     where tb_sampah.id_kategori = tb_sampahKategori.id_kategori
69.     and nama_sampah ilike '%$input[keyword]%' $kategori");
70.     return $data->result_array();
71. }
72.
73. // get
74. function get_list_sampah() {
75.     $data = $this->db->query("select * from
76.     tb_sampah, tb_sampahkategori
77.     where tb_sampah.id_kategori = tb_sampahkategori.id_kategori
78.     order by nama_sampah asc");
79.     return $data->result_array();
80. }
81.
82. function get_sampah_by($id) {
83.     $data = $this->db->query("select *
84.     from tb_sampah, tb_sampahkategori
85.     where tb_sampah.id_kategori = tb_sampahkategori.id_kategori
86.     and id_sampah = $id");
87.     return $data->row_array();
88. }
89.
90.
91. function get_sampah_detail($id) {
92.     $data = $this->db->query("select *
93.     from tb_sampah, tb_sampahkategori
94.     where tb_sampah.id_kategori = tb_sampahkategori.id_kategori
95.     and id_sampah = $id");
96.     return $data->row_array();
97. }
98.
99. function get_sampah_form() {
100.     $data = $this->db->query("select * from tb_sampah");
101.     return $data->result_array();
102. }
103.
104.
105. function get_kategori() {
106.     return $this->db->query("select * from tb_sampahkategori")->result_array();
107. }
108.
109. function get_rows_sampah() {

```

```

110.     return $this->db->query("select id_sampah from tb_sampah")->num_rows();
111.     }
112.
113.     function get_new_id() {
114.         $data = $this->db->query("select max(id_sampah) as id from tb_sampah");
115.         $last = $data->row_array();
116.         if ($last['id'] == 0 || $last['id'] == NULL) {
117.             return 1;
118.         } else {
119.             return ($last['id'] + 1);
120.         }
121.     }
122.
123.     // cek
124.     function cek_namasampah($nama){
125.         $data = $this->db->query("select * from tb_sampah where nama_sampah =
126.         '".strtoupper($nama)."'");
127.         if($data->num_rows() > 0){
128.             return FALSE;
129.         } else{
130.             return TRUE;
131.         }
132.     }
133.
134.     function cek_transaksi($id_sampah){
135.         $data1 = $this->db->query("select * from tb_rating where id_sampah =
136.         $id_sampah");
137.         $data2 = $this->db->query("select * from tb_review where id_sampah =
138.         $id_sampah");
139.         if($data1->num_rows() > 0 || $data2->num_rows() > 0){
140.             return FALSE;
141.         } else{
142.             return TRUE;
143.         }
144.     }
145.
146.     function __is_exist($id) {
147.         $this->db->where('id_sampah', $id);
148.         $data = $this->db->get('tb_sampah');
149.         if ($data->num_rows() > 0) {
150.             return TRUE;
151.         } else {
152.             return FALSE;
153.         }
154.     }
155.
156.     // set
157.     function add_sampah($input) {
158.         $new_id = $this->get_new_id();
159.         $user = $this->session->userdata('sampah_iduser');
160.         $this->db->trans_begin();
161.         if ($input['gambar'] == 1) {
162.             $this->db->query("
163.             insert into tb_sampah (
164.                 id_sampah,
165.                 nama_sampah,
166.                 lokasi_sampah,
167.                 id_kategori,
168.                 keterangan,
169.                 id_user,
170.                 id_icon,
171.                 x,

```

```

172.         y,
173.         geom
174.     ) values (
175.         "" . $new_id . "",
176.         "" . $input['nama_sampah'] . "",
177.         "" . $input['lokasi'] . "",
178.         "" . $input['id_kategori'] . "",
179.         "" . $input['keterangan'] . "",
180.         "" . $user . "",
181.         "" . $input['id_kategori'] . "",
182.         $input[long],
183.         $input[lat],
184.         ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
185.     )
186. ");
187. } else {
188.     $this->db->query("
189.     insert into tb_sampah (
190.     id_sampah,
191.     nama_sampah,
192.     lokasi_sampah,
193.     id_kategori,
194.     keterangan,
195.     gambar,
196.     id_user,
197.     id_icon,
198.     x,
199.     y,
200.     geom
201.     ) values (
202.         "" . $new_id . "",
203.         "" . $input['nama_sampah'] . "",
204.         "" . $input['lokasi'] . "",
205.         "" . $input['id_kategori'] . "",
206.         "" . $input['keterangan'] . "",
207.         "" . $input['gambar'] . "",
208.         "" . $user . "",
209.         "" . $input['id_kategori'] . "",
210.         $input[long],
211.         $input[lat],
212.         ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
213.     )
214. ");
215. }
216.
217. if ($this->db->trans_status() === FALSE) {
218.     $this->db->trans_rollback();
219.     return FALSE;
220. } else {
221.     $this->db->trans_commit();
222.     return TRUE;
223. }
224. }
225.
226. function edit_sampah($input) {
227.     $this->db->trans_begin();
228.     if ($input['gambar'] == 1) {
229.         $this->db->query("
230.         update tb_sampah set
231.         nama_sampah = '$input[nama_sampah]',
232.         lokasi_sampah = '$input[lokasi]',
233.         id_kategori = $input[id_kategori],

```

```

234.         keterangan = '$input[keterangan]',
235.         id_icon = $input[id_icon],
236.         x = $input[long],
237.         y = $input[lat],
238.         geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
239.         where id_sampah = $input[id_sampah]
240.     ");
241. } else {
242.     $this->db->query("
243.         update tb_sampah set
244.         nama_sampah = '$input[nama_sampah]',
245.         lokasi_sampah = '$input[lokasi]',
246.         id_kategori = $input[id_kategori],
247.         keterangan = '$input[keterangan]',
248.         gambar = '$input[gambar]',
249.         id_icon = $input[id_icon],
250.         x = $input[long],
251.         y = $input[lat],
252.         geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
253.         where id_sampah = $input[id_sampah]
254.     ");
255. }
256.
257. if ($this->db->trans_status() === FALSE) {
258.     $this->db->trans_rollback();
259.     return FALSE;
260. } else {
261.     $this->db->trans_commit();
262.     return TRUE;
263. }
264. }
265.
266.
267. function delete_sampah($id) {
268.     if ($this->__is_exist($id) {
269.         $this->db->trans_begin();
270.         $this->db->query("delete from tb_sampah where id_sampah = $id");
271.         if ($this->db->trans_status() === FALSE) {
272.             $this->db->trans_rollback();
273.             return FALSE;
274.         } else {
275.             $this->db->trans_commit();
276.             return TRUE;
277.         }
278.     } else {
279.         return FALSE;
280.     }
281. }
282.
283. function latlong(){
284.     $data = $this->db->query("select x as lng, y as lat, nama_sampah from
285.     tb_sampah");
286.     return $data->result_array();
287. }
288.
289. }
290.
291. ?>
292.

```



## D.2 Kode Program Melihat Data Pembuangan Sampah

Kode program melihat data pembuangan sampah terletak pada kelas pembuangan dan pembuangan\_model.

### 1. Kelas pembuangan

Penulisan kode program pembuangan dapat dilihat pada Tabel 9.

Tabel 9 Kode Program Kelas pembuangan

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Pembuangan extends CI_Controller {
7.	
8.	private \$data;
9.	
10.	function __construct() {
11.	parent::__construct();
12.	\$this->load->model('sampah_model');
13.	\$this->load->model('akun_model');
14.	\$this->load->model('routing_model');
15.	\$this->load->model('pembuangan_model');
16.	
17.	\$this->data = array(
18.	'title' => NULL,
19.	'content' => NULL,
20.	'menu' => NULL,
21.	'akses' => \$this->session->userdata('sampah_hakakses'),
22.	'username' => \$this->session->userdata('sampah_username'),
23.	);
24.	}
25.	
26.	function index() {
27.	\$this->list_pembuangan();
28.	}
29.	
30.	function cobapembuangan(){
31.	\$this->data['menu'] = 1;
32.	\$this->data['title'] = 'Pembuangan Sampah';
33.	\$this->data['content'] = \$this->load->view('pembuangan_list',array(
34.	'data' => \$this->pembuangan_model->get_list_pembuangan()
35.	),TRUE);
36.	\$this->load->view('template_baru',\$this->data);
37.	}
38.	
39.	function add(){
40.	\$this->data['menu'] = 1;
41.	\$this->data['title'] = 'Tambah Pembuangan';
42.	\$this->data['content'] = \$this->load->view('pembuangan_add',array(
43.	'kategori' => \$this->sampah_model->get_kategori(),
44.	'sampah' => \$this->sampah_model->get_semua()
45.	),TRUE);
46.	
47.	\$this->load->view('template_baru',\$this->data);
48.	}

```

49.
50. function nearest_node() {
51.     $data = array(
52.         'x' => $this->input->post('x'),
53.         'y' => $this->input->post('y')
54.     );
55.
56.     $result = $this->routing_model->get_nearest_coordinate($data);
57.     $result2 = $this->routing_model->get_nearest_network($data);
58.
59.     if ($result2['osm_name'] != "") {
60.         $nama_jalan = $result2['osm_name'];
61.     } else {
62.         $nama_jalan = 'No name';
63.     }
64.
65.     echo $result['lat'] . ' ' . $result['lng'];
66.     echo '|';
67.     echo $nama_jalan;
68. }
69.
70. function route() {
71.     $data = array(
72.         'x1' => $this->input->post('x1'),
73.         'y1' => $this->input->post('y1'),
74.         'x2' => $this->input->post('x2'),
75.         'y2' => $this->input->post('y2'),
76.         'kendaraan' => $this->input->post('kendaraan'),
77.         'jalur' => $this->input->post('jalur'),
78.         'id' => $this->input->post('id')
79.     );
80.
81.     $rs = $this->routing_model->get_route_geojson($data);
82.     $output = "";
83.     $rowOutput = "";
84.     foreach ($rs as $row) {
85.         $rowOutput = (strlen($rowOutput) > 0 ? ',' : '') . '{"type": "Feature", "geometry":
86.         ' . $row['geojson'] . ', "properties": {';
87.         $props = "";
88.         $id = "";
89.         foreach ($row as $key => $val) {
90.             if ($key != "geojson") {
91.                 $props .= (strlen($props) > 0 ? ',' : '') . "" . $key . ":" . $this-
92.                 >escapeJsonString($val) . ""';
93.             }
94.             if ($key == "id") {
95.                 $id .= '"id":' . $this->escapeJsonString($val) . ""';
96.             }
97.         }
98.
99.         $rowOutput .= $props . '>';
100.        $rowOutput .= $id;
101.        $rowOutput .= '>';
102.        $output .= $rowOutput;
103.    }
104.
105.    $output = '{ "type": "FeatureCollection", "features": [ ' . $output . ' ]}';
106.    echo $output;
107. }
108.
109. function add_process(){
110.     $data = array(

```

111.	'id_sampah' => \$this->input->post('id_sampah'),
112.	'tanggal_buang' => \$this->input->post('tanggal_buang'),
113.	'sopir' => \$this->input->post('sopir')
114.	);
115.	
116.	\$data2 = array(
117.	'lat' => \$this->input->post('lat'),
118.	'long' => \$this->input->post('long')
119.	);
120.	\$hasil = \$this->pembuangan_model->add(\$data);
121.	if (\$hasil) {
122.	redirect('home/mapssopir');
123.	} else {
124.	echo 'gagal';
125.	}
126.	
127.	}
128.	
129.	function validasi(\$id_pembuangan){
130.	\$this->pembuangan_model->ubah_validasi(\$id_pembuangan);
131.	redirect('pembuangan/cobapembuangan');
132.	
133.	}
134.	
135.	}
136.	

## 2. Kelas pembuangan\_model

Penulisan kode program pembuangan\_model dapat dilihat pada Tabel 10.

Tabel 10 Kode Program Kelas pembuangan\_model

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Pembuangan_Model extends CI_Model {
7.	
8.	function __construct() {
9.	parent::__construct();
10.	}
11.	
12.	function get_rows_pembuangan() {
13.	return \$this->db->query("select id_pembuangan from tb_pembuangan")->
14.	num_rows();
15.	}
16.	
17.	function get_list_pembuangan() {
18.	\$usr = \$this->session->userdata('sampah_hakakses');
19.	if(\$usr==1){
20.	\$data = \$this->db->query("select * from
21.	tb_pembuangan a join tb_sampah b on (a.id_sampah = b.id_sampah)
22.	order by id_pembuangan desc
23.	");
24.	}
25.	else {
26.	\$data = \$this->db->query("select * from

27.	tb_pembuangan a join tb_sampah b on (a.id_sampah = b.id_sampah)
28.	order by id_pembuangan desc
29.	");
30.	}
31.	return \$data->result_array();
32.	}
33.	
34.	
35.	function add(\$data){
36.	\$data = \$this->db->query("INSERT INTO public.tb_pembuangan(
37.	id_sampah, tanggal_buang, sopir)
38.	VALUES (
39.	
40.	" . \$data['id_sampah'] . ",
41.	" . \$data['tanggal_buang'] . ",
42.	" . \$data['sopir'] . ")");
43.	
44.	return \$data;
45.	}
46.	
47.	function jumlah(){
48.	return \$this->db->query ("select COUNT(id_pembuangan) as jumlah from
49.	tb_pembuangan");
50.	}
51.	
52.	function ubah_validasi(\$id_pembuangan){
53.	\$this->db->query("
54.	update tb_pembuangan set
55.	status = 1 where id_pembuangan=\$id_pembuangan");
56.	}
57.	}
58.	

### D.3 Kode Program Pengelolaan Data User

Kode program pengelolaan data user terletak pada kelas akun dan akun\_model.

#### 1. Kelas akun

Penulisan kode program akun dapat dilihat pada Tabel 11.

Tabel 11 Kode Program Kelas akun

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Akun extends CI_Controller {
7.	
8.	private \$data;
9.	
10.	function __construct() {
11.	parent::__construct();
12.	if (\$this->session->userdata('sampah_hakakses') == NULL) {
13.	redirect('home');

```
14.     }
15.
16.     $this->load->model('akun_model');
17.     $this->data = array(
18.         'title' => NULL,
19.         'content' => NULL,
20.         'menu' => NULL,
21.         'akses' => $this->session->userdata('sampah_hakakses'),
22.         'username' => $this->session->userdata('sampah_username'),
23.     );
24. }
25.
26. function cobaakun(){
27.     $this->data['menu'] = 1;
28.     $this->data['title'] = 'Manajemen User';
29.     $this->data['content'] = $this->load->view('akun_list,array(
30.         'data' => $this->akun_model->get_list_user()
31.     ),TRUE);
32.     $this->load->view('template_baru',$this->data);
33. }
34.
35. function profile() {
36.     if ($this->data['akses'] != NULL) {
37.         $this->data['title'] = 'Profil';
38.         $this->data['content'] = $this->load->view(
39.             'akun_detail', array('data' => $this->akun_model->get_user_detail($this-
40. >session->userdata('sampah_iduser'))), TRUE);
41.
42.         $this->load->view('template', $this->data);
43.     } else {
44.         redirect('home');
45.     }
46. }
47.
48. function detail($id_user) {
49.     if ($this->data['akses'] != NULL) {
50.         $this->data['title'] = 'Detail Akun';
51.         $this->data['content'] = $this->load->view(
52.             'akun_detail', array('data' => $this->akun_model-
53. >get_user_detail($id_user)), TRUE);
54.
55.         $this->load->view('template_baru', $this->data);
56.     } else {
57.         redirect('home');
58.     }
59. }
60.
61. function add() {
62.     if ($this->data['akses'] == 1) {
63.         $this->data['title'] = 'Tambah Akun';
64.         $this->data['content'] = $this->load->view(
65.             'akun_add', array(
66.                 'akses' => $this->akun_model->get_akses(), TRUE);
67.         $this->load->view('template_baru', $this->data);
68.     } else {
69.         redirect('home/cobaakun');
70.     }
71. }
72.
73. function edit($id_user) {
74.     if ($this->data['akses'] != NULL) {
75.         $this->data['title'] = 'Edit Akun';
```

```

76.     $this->data['content'] = $this->load->view(
77.         'akun_edit', array(
78.             'kode' => NULL,
79.             'akses' => $this->akun_model->get_akses(),
80.             'data' => $this->akun_model->get_user_by($id_user), TRUE);
81.     $this->load->view('template_baru', $this->data);
82. } else {
83.     redirect('home');
84. }
85. }
86.
87. function add_process() {
88.     if ($this->data['akses'] != NULL) {
89.         if ($this->__add_rule()) {
90.             if ($_FILES['foto']['name'] == "") {
91.                 $foto = 1;
92.             } else {
93.                 $foto = $_FILES['foto']['name'];
94.             }
95.             $data = array(
96.                 'id_user' => "",
97.                 'username' => $this->input->post('username'),
98.                 'password' => $this->input->post('password'),
99.                 'nama_lengkap' => $this->input->post('nama_lengkap'),
100.                'id_jeniskelamin' => $this->input->post('jenis_kelamin'),
101.                'no_telepon' => $this->input->post('no_telepon'),
102.                'email' => $this->input->post('email'),
103.                'alamat' => $this->input->post('alamat'),
104.                'id_hakakses' => $this->input->post('hak_akses'),
105.                'foto' => $foto
106.            );
107.            if ($data['foto'] == 1) {
108.                $sukses = $this->akun_model->add($data);
109.                if ($sukses) {
110.                    $this->session->set_userdata('message', '1,User telah ditambahkan.');
```

```

138.         $this->session->set_userdata('message', '2,Upload foto gagal.');
```

```

139.         redirect('user/add');
```

```

140.     }
141. }
142. } else {
143.     $this->add();
144. }
145. } else {
146.     redirect('home');
```

```

147. }
148. }
149.
150. function edit_process($kode = NULL) {
151.     if ($this->data['akses'] != NULL) {
152.         if ($this->__edit_rule()) {
153.             if ($_FILES['foto']['name'] == "") {
154.                 $foto = 1;
155.             } else {
156.                 $foto = $_FILES['foto']['name'];
157.             }
158.             $data = array(
159.                 'id_user' => $this->input->post('id_user'),
160.                 'nama_lengkap' => $this->input->post('nama_lengkap'),
161.                 'id_jeniskelamin' => $this->input->post('jenis_kelamin'),
162.                 'no_telepon' => $this->input->post('no_telepon'),
163.                 'email' => $this->input->post('email'),
164.                 'alamat' => $this->input->post('alamat'),
165.                 'id_hakakses' => $this->input->post('hak_akses'),
166.                 'foto' => $foto
167.             );
168.
169.             if ($data['foto'] == 1) {
170.                 $sukses = $this->akun_model->edit($data);
171.                 if ($kode == NULL) {
172.                     if ($sukses) {
173.                         $this->session->set_userdata('message', '1,User telah diupdate.');
```

```

174.                         redirect('akun/cobaakun/' . $this->input->post('id_user'));
175.                     } else {
176.                         $this->session->set_userdata('message', '2,User gagal diupdate.');
```

```

177.                         redirect('akun/cobaakun/' . $this->input->post('id_user'));
178.                     }
179.                 } else {
180.                     if ($sukses) {
181.                         $this->session->set_userdata('message', '1,Profil telah diupdate.');
```

```

182.                         redirect('akun/edit_account/');
183.                     } else {
184.                         $this->session->set_userdata('message', '2,Profil gagal diupdate.');
```

```

185.                         redirect('akun/edit_account/');
186.                     }
187.                 }
188.             } else {
189.                 $config_image = array(
190.                     'upload_path' => './images/',
191.                     'allowed_types' => 'gif|jpg|png|jpeg',
192.                     'max_size' => '500000',
193.                     'overwrite' => TRUE,
194.                     'remove_spaces' => FALSE,
195.                     'file_name' => $_FILES['foto']['name']
196.                 );
197.                 $this->load->library('upload');
```

```

198.                 $this->upload->initialize($config_image);
199.                 if ($this->upload->do_upload('foto')) {
```

```
200.     $sukses = $this->user_model->edit($data);
201.     if ($kode == NULL) {
202.         if ($sukses) {
203.             $this->session->set_userdata('message', '1,User telah diupdate.');
```

204. redirect('akun/cobaakun/' . \$this->input->post('id\_user'));
205. } else {
206. \$this->session->set\_userdata('message', '2,User gagal diupdate.');

207. redirect('akun/cobaakun/' . \$this->input->post('id\_user'));
208. }
209. } else {
210. if (\$sukses) {
211. \$this->session->set\_userdata('message', '1,Profil telah diupdate.');

212. redirect('akun/edit\_account/');
213. } else {
214. \$this->session->set\_userdata('message', '2,Profil gagal diupdate.');

215. redirect('akun/edit\_account/');
216. }
217. }
218. } else {
219. \$this->session->set\_userdata('message', '2,Upload foto gagal.');

220. if (\$kode == NULL) {
221. redirect('akun/edit/' . \$this->input->post('id\_user'));
222. } else {
223. redirect('akun/edit\_account/');

224. }
225. }
226. }
227. } else {
228. if (\$kode == NULL) {
229. \$this->edit(\$this->input->post('id\_user'));
230. } else {
231. \$this->edit\_account();
232. }
233. }
234. } else {
235. redirect('home');

236. }
237. }
238.
239. function edit\_password\_process() {
240. if (\$this->data['akses'] != NULL) {
241. \$data = array(
242. 'id\_user' => \$this->input->post('id\_user'),
243. 'username' => \$this->input->post('username'),
244. 'password' => \$this->input->post('password'),
245. 'password\_baru' => \$this->input->post('password\_baru'),
246. 'password\_konfirmasi' => \$this->input->post('password\_konfirmasi'),
247. );
248. \$sis\_password = \$this->user\_model->\_\_is\_password(\$data['id\_user'],
249. \$data['password']);
250. if (\$sis\_password) {
251. if (\$data['password\_baru'] == \$data['password\_konfirmasi']) {
252. \$sukses = \$this->user\_model->edit\_account(\$data);
253. if (\$sukses) {
254. \$this->session->set\_userdata('message', '1>Password telah diupdate.');

255. redirect('akun/edit\_akun/');
256. } else {
257. \$this->session->set\_userdata('message', '2>Password gagal diupdate.');

258. redirect('akun/edit\_akun/');
259. }
260. } else {
261. \$this->session->set\_userdata('message', '2,Konfirmasi password salah.');



```
262.         redirect('akun/edit_akun/');
263.     }
264.     } else {
265.         $this->session->set_userdata('message', '2,Password salah.');
```

266. redirect('akun/edit\_akun/');

```
267.     }
268.     } else {
269.         redirect('home');
```

270. }

```
271. }
272.
273. function delete_process($id_user) {
274.     if ($this->data['akses'] == 1) {
275.         if ($this->akun_model->cek_idhakakses($id_user) != 1) {
276.             if ($this->akun_model->cek_transaksi($id_user) == TRUE) {
277.                 $sukses = $this->akun_model->delete($id_user);
278.                 if ($sukses == true) {
279.                     $this->session->set_userdata('message', '1,User telah dihapus.');
```

280. redirect('akun/cobaakun/');

```
281.                 } else {
282.                     $this->session->set_userdata('message', '2,User gagal dihapus.');
```

283. redirect('akun/cobaakun/');

```
284.                 }
285.             } else {
286.                 $this->session->set_userdata('message', '2,Telah ada transaksi dengan tabel
287. lain user tidak boleh dihapus.');
```

288. redirect('akun/cobaakun/');

```
289.             }
290.         } else {
291.             $this->session->set_userdata('message', '2,User administrator tidak boleh
292. dihapus.');
```

293. redirect('akun/cobaakun/');

```
294.         }
295.     } else {
296.         redirect('home');
```

297. }

```
298. }
299.
300. function status($id_status, $id_user) {
301.     if ($this->data['akses'] == 1) {
302.         if ($this->user_model->cek_idhakakses($id_user) != 1) {
303.             $sukses = $this->user_model->change_status($id_status, $id_user);
304.             if ($sukses == true) {
305.                 $this->session->set_userdata('message', '1,Status user telah diubah.');
```

306. redirect('user/list\_user/');

```
307.             } else {
308.                 $this->session->set_userdata('message', '2,Status user gagal diubah.');
```

309. redirect('user/list\_user/');

```
310.             }
311.         } else {
312.             $this->session->set_userdata('message', '2,User administrator tidak boleh
313. dinontaktifkan.');
```

314. redirect('user/list\_user/');

```
315.         }
316.     } else {
317.         redirect('home');
```

318. }

```
319. }
320.
321. private function __rule($rule) {
322.     $this->load->library('form_validation');
```

323. \$this->form\_validation->set\_rules(\$rule);

```
324.     if ($this->form_validation->run()) {
325.         return TRUE;
326.     } else {
327.         return FALSE;
328.     }
329. }
330.
331. private function __add_rule() {
332.     $rule = array
333.     (
334.         array
335.         (
336.             'field' => 'username'
337.             , 'label' => 'Username'
338.             , 'rules' => 'required'
339.         )
340.         , array
341.         (
342.             'field' => 'password'
343.             , 'label' => 'Password'
344.             , 'rules' => 'required'
345.         )
346.         , array
347.         (
348.             'field' => 'nama_lengkap'
349.             , 'label' => 'Nama Lengkap'
350.             , 'rules' => 'required'
351.         )
352.         , array
353.         (
354.             'field' => 'jenis_kelamin'
355.             , 'label' => 'Jenis Kelamin'
356.             , 'rules' => 'required'
357.         )
358.         , array
359.         (
360.             'field' => 'alamat'
361.             , 'label' => 'Alamat'
362.             , 'rules' => 'required'
363.         )
364.         , array
365.         (
366.             'field' => 'hak_akses'
367.             , 'label' => 'Hak Akses'
368.             , 'rules' => 'required'
369.         )
370.         , array
371.         (
372.             'field' => 'no_telepon'
373.             , 'label' => 'No Telepon'
374.             , 'rules' => 'trim|required|numeric'
375.         )
376.         , array
377.         (
378.             'field' => 'email'
379.             , 'label' => 'Email'
380.             , 'rules' => 'required|email'
381.         )
382.     );
383.     return $this->__rule($rule);
384. }
385.
```

```
386. private function __edit_rule() {
387.     $rule = array
388.     (
389.         array
390.         (
391.             'field' => 'nama_lengkap'
392.             , 'label' => 'Nama Lengkap'
393.             , 'rules' => 'required'
394.         )
395.     , array
396.     (
397.         'field' => 'jenis_kelamin'
398.         , 'label' => 'Jenis Kelamin'
399.         , 'rules' => 'required'
400.     )
401.     , array
402.     (
403.         'field' => 'alamat'
404.         , 'label' => 'Alamat'
405.         , 'rules' => 'required'
406.     )
407.     , array
408.     (
409.         'field' => 'hak_akses'
410.         , 'label' => 'Hak Akses'
411.         , 'rules' => 'required'
412.     )
413.     , array
414.     (
415.         'field' => 'no_telepon'
416.         , 'label' => 'No Telepon'
417.         , 'rules' => 'trim|required|numeric'
418.     )
419.     , array
420.     (
421.         'field' => 'email'
422.         , 'label' => 'Email'
423.         , 'rules' => 'required|email'
424.     )
425. );
426. return $this->__rule($rule);
427. }
428.
429. private function __edit_akun_rule() {
430.     $rule = array
431.     (
432.         array
433.         (
434.             'field' => 'password'
435.             , 'label' => 'Old Password'
436.             , 'rules' => 'required'
437.         )
438.     , array
439.     (
440.         'field' => 'password_baru'
441.         , 'label' => 'New Password'
442.         , 'rules' => 'required'
443.     )
444.     , array
445.     (
446.         'field' => 'password_konfirmasi'
447.         , 'label' => 'Password Confirm'
```

448.	, 'rules' => 'required'
449.	)
450.	);
451.	return \$this->__rule(\$rule);
452.	}
453.	}
454.	?>

## 2. Kelas akun\_model

Penulisan kode program akun\_model dapat dilihat pada Tabel 12.

Tabel 12 Kode Program Kelas akun\_model

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Akun_Model extends CI_Model {
7.	
8.	function __construct() {
9.	parent::__construct();
10.	}
11.	
12.	function get_list_user() {
13.	\$data = \$this->db->query("select * from
14.	tb_user, tb_userHakAkses, tb_status
15.	where tb_user.id_hakakses = tb_userHakAkses.id_hakakses
16.	and tb_user.id_status = tb_status.id_status
17.	order by tb_user.id_user desc ");
18.	return \$data->result_array();
19.	}
20.	
21.	function get_user_by(\$id_user) {
22.	\$this->db->where('id_user', \$id_user);
23.	return \$this->db->get('tb_user')->row_array();
24.	}
25.	
26.	function get_user_detail(\$id_user) {
27.	\$data = \$this->db->query("
28.	select *
29.	from tb_user, tb_userHakAkses, tb_status, tb_userJenisKelamin
30.	where tb_user.id_hakakses = tb_userHakAkses.id_hakakses
31.	and tb_user.id_status = tb_status.id_status
32.	and tb_user.id_jeniskelamin = tb_userJenisKelamin.id_jeniskelamin
33.	and tb_user.id_user = \$id_user");
34.	return \$data->row_array();
35.	}
36.	
37.	function get_username_by(\$id_user) {
38.	\$data = \$this->db->query("select username from tb_user where id_user =
39.	'\$id_user'");
40.	return \$data->row_array();
41.	}
42.	
43.	function get_rows_user() {
44.	return \$this->db->get('tb_user')->num_rows();
45.	}

```
46.
47. function get_akses() {
48.     return $this->db->query('select * from tb_userHakAkses')->result_array();
49. }
50.
51. function get_new_id() {
52.     $data = $this->db->query("select max(id_user) as id from tb_user");
53.     $last = $data->row_array();
54.     if ($last['id'] == 0 || $last['id'] == NULL) {
55.         return 1;
56.     } else {
57.         return ($last['id'] + 1);
58.     }
59. }
60.
61. // cek
62. function cek_email($email){
63.     $data = $this->db->query("select * from tb_user where email = '$email'");
64.     if($data->num_rows() > 0){
65.         return FALSE;
66.     } else{
67.         return TRUE;
68.     }
69. }
70.
71. function cek_transaksi($id_user){
72.     $data1 = $this->db->query("select * from tb_rating where id_user = $id_user");
73.     $data2 = $this->db->query("select * from tb_review where id_user = $id_user");
74.     $data3 = $this->db->query("select * from tb_lalulintas where id_user = $id_user");
75.     if($data1->num_rows() > 0 || $data2->num_rows() > 0 || $data3->num_rows() > 0){
76.         return FALSE;
77.     } else{
78.         return TRUE;
79.     }
80. }
81.
82. function cek_idhakakses($id_user){
83.     $data = $this->db->query("select id_hakakses from tb_user where id_user =
84. $id_user");
85.     return $data->row()->id_hakakses;
86. }
87.
88. function __is_exist($id_user) {
89.     $this->db->where('id_user', $id_user);
90.     $data = $this->db->get('tb_user');
91.     if ($data->num_rows() > 0) {
92.         return TRUE;
93.     } else {
94.         return FALSE;
95.     }
96. }
97.
98. function __is_available($username) {
99.     $data = $this->db->query("select * from tb_user where username = '$username'");
100.     if ($data->num_rows() > 0) {
101.         return FALSE;
102.     } else {
103.         return TRUE;
104.     }
105. }
106.
107. function __is_password($id_user, $password) {
```

```
108.     $data = $this->db->query("select * from tb_user where id_user = $id_user and
109. password = '" . md5($password) . "'");
110.     if ($data->num_rows() > 0) {
111.         return TRUE;
112.     } else {
113.         return FALSE;
114.     }
115. }
116.
117. // set
118. function add($user) {
119.     if ($this->__is_available($user['username'])) {
120.         $id = $this->get_new_id();
121.         if ($user['foto'] == 1) {
122.             $this->db->query("
123.                 insert into tb_user (
124.                     id_user,
125.                     username,
126.                     password,
127.                     nama_lengkap,
128.                     alamat,
129.                     email,
130.                     no_telepon,
131.                     tgl_register,
132.                     id_jeniskelamin,
133.                     id_hakakses,
134.                     id_status
135.                 ) values (
136.                     '$id',
137.                     '" . $user['username'] . "',
138.                     '" . md5($user['password']) . "',
139.                     '" . $user['nama_lengkap'] . "',
140.                     '" . $user['alamat'] . "',
141.                     '" . $user['email'] . "',
142.                     '" . $user['no_telepon'] . "',
143.                     now(),
144.                     $user[id_jeniskelamin],
145.                     $user[id_hakakses],
146.                     2
147.                 )
148.             ");
149.         } else {
150.             $this->db->query("
151.                 insert into tb_user (
152.                     id_user,
153.                     username,
154.                     password,
155.                     nama_lengkap,
156.                     alamat,
157.                     email,
158.                     no_telepon,
159.                     tgl_register,
160.                     id_jeniskelamin,
161.                     id_hakakses,
162.                     id_status,
163.                     foto
164.                 ) values (
165.                     '$id',
166.                     '" . $user['username'] . "',
167.                     '" . md5($user['password']) . "',
168.                     '" . $user['nama_lengkap'] . "',
169.                     '" . $user['alamat'] . "',
```

```
170.         "" . $user['email'] . "" ,
171.         "" . $user['no_telepon'] . "" ,
172.         now(),
173.         $user[id_jeniskelamin],
174.         $user[id_hakakses],
175.         2,
176.         "" . $user['foto'] . ""
177.     )
178.     ");
179. }
180. return TRUE;
181. } else {
182.     return FALSE;
183. }
184. }
185.
186. function edit($user) {
187.     if ($this->__is_exist($user['id_user'])) {
188.         if ($user['foto'] == 1) {
189.             $this->db->query(" update tb_user set
190.                 nama_lengkap = '$user[nama_lengkap]',
191.                 alamat = '$user[alamat]',
192.                 email = '$user[email]',
193.                 no_telepon = '$user[no_telepon]',
194.                 id_jeniskelamin = '$user[id_jeniskelamin]',
195.                 id_hakakses = $user[id_hakakses]
196.                 where id_user = $user[id_user]
197.                 ");
198.         } else {
199.             $this->db->query(" update tb_user set
200.                 nama_lengkap = '$user[nama_lengkap]',
201.                 alamat = '$user[alamat]',
202.                 email = '$user[email]',
203.                 no_telepon = '$user[no_telepon]',
204.                 id_jeniskelamin = '$user[id_jeniskelamin]',
205.                 id_hakakses = $user[id_hakakses],
206.                 foto = '$user[foto]'
207.                 where id_user = $user[id_user]
208.                 ");
209.         }
210.         return TRUE;
211.     } else {
212.         return FALSE;
213.     }
214. }
215.
216. function edit_akun($user) {
217.     if ($this->__is_exist($user['id_user'])) {
218.         $this->db->query("update tb_user set
219.             password = "" . md5($user['password_baru']) . ""
220.             where id_user = '$user[id_user]'
221.             ");
222.         return TRUE;
223.     } else {
224.         return FALSE;
225.     }
226. }
227.
228. function delete($id_user) {
229.     if ($this->__is_exist($id_user)) {
230.         $this->db->trans_begin();
231.         $this->db->query("delete from tb_review where id_user = $id_user");
```

```

232.     $this->db->query("delete from tb_lalulintas where id_user = $id_user");
233.     $this->db->query("delete from tb_rating where id_user = $id_user");
234.     $this->db->query("delete from tb_user where id_user = $id_user");
235.     if ($this->db->trans_status() === FALSE) {
236.         $this->db->trans_rollback();
237.         return FALSE;
238.     } else {
239.         $this->db->trans_commit();
240.         return TRUE;
241.     }
242. } else {
243.     return FALSE;
244. }
245. }
246.
247. function change_status($id_status, $id_user) {
248.     $cek = $this->db->query("update tb_user set id_status = $id_status where id_user =
249. $id_user");
250.     if ($cek === TRUE) {
251.         return TRUE;
252.     } else {
253.         return FALSE;
254.     }
255. }
256.
257. function login($input) {
258.     $data = $this->db->query("select * from tb_user, tb_userHakAkses, tb_status
259.     where tb_user.id_hakAkses = tb_userHakAkses.id_hakAkses
260.     and tb_user.id_status = tb_status.id_status
261.     and tb_user.id_status = 2
262.     and username = '$input[username]'
263.     and password = '" . md5($input['password']) . "'");
264.     if ($data->num_rows() > 0) {
265.         $this->session->set_userdata('sampah_username', $data->row()->username);
266.         $this->session->set_userdata('sampah_nama', $data->row()->nama_lengkap);
267.         $this->session->set_userdata('sampah_hakakses', $data->row()->id_hakakses);
268.         $this->session->set_userdata('sampah_iduser', $data->row()->id_user);
269.         $this->db->query("update tb_user set tgl_login = now() where username =
270. '$input[username]'");
271.         return TRUE;
272.     } else {
273.         return FALSE;
274.     }
275. }
276.
277. }
278.
279. ?>

```

#### D.4 Kode Program Melihat TPS dan TPA

Kode program melihat TPS dan TPA terletak pada sampah dan sampah\_model.

##### 1. Kelas sampah

Penulisan kode program sampah dapat dilihat pada Tabel 13.



Tabel 13 Kode Program Kelas sampah

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Sampah extends CI_Controller {
7.	
8.	private \$data;
9.	
10.	function __construct() {
11.	parent::__construct();
12.	
13.	\$this->load->model('sampah_model');
14.	\$this->load->model('akun_model');
15.	
16.	\$this->data = array(
17.	'title' => NULL,
18.	'content' => NULL,
19.	'menu' => NULL,
20.	'akses' => \$this->session->userdata('sampah_hakakses'),
21.	'username' => \$this->session->userdata('sampah_username'),
22.	'nama_lengkap' => \$this->session->userdata('nama_lengkap')
23.	);
24.	}
25.	// memanggil function list sampah untuk ditampilkan di halaman awal manajemen
26.	function cobasampah(){
27.	\$this->data['menu'] = 1;
28.	\$this->data['title'] = 'Manajemen Sampah';
29.	\$this->data['content'] = \$this->load->view('sampah_list',array(
30.	'data' => \$this->sampah_model->get_list_sampah(
31.	),TRUE);
32.	\$this->load->view('template_baru',\$this->data);
33.	}
34.	
35.	
36.	// menampilkan detail ketika user mengklik icon
37.	function detail(\$id) {
38.	\$data_sampah = \$this->sampah_model->get_sampah_by(\$id);
39.	\$this->data['title'] = \$data_sampah['nama_sampah'];
40.	\$this->data['content'] = \$this->load->view(
41.	'sampah_detail', array(
42.	'id_sampah' => \$id,
43.	'data' => \$data_sampah,
44.	), TRUE);
45.	
46.	\$this->load->view('template_baru', \$this->data);
47.	}
48.	// fungsi ajax untuk menampilkan detail
49.	
50.	function detail_ajax() {
51.	\$id = \$this->input->post('id');
52.	\$data_sampah = \$this->sampah_model->get_sampah_by(\$id);
53.	if (\$data_sampah['gambar'] == "") {
54.	\$gambar = 'no-image.png';
55.	} else {
56.	\$gambar = \$data_sampah['gambar'];
57.	}
58.	
59.	echo '<img src="" . base_url() . 'images/' . \$gambar . "" style="width:150px;

```

60. height:135px; border: 2px solid #e6e6e6; border-radius: 5px; padding: 1px;"/>||';
61. echo '<h1>' . $data_sampah['nama_sampah'] . '</h1>';
62. echo 'Alamat : ' . $data_sampah['lokasi_sampah'] . '<br/>';
63. echo 'Kategori : ' . $data_sampah['nama_kategori'] . '<br/>';
64. echo 'Keterangan : ' . $data_sampah['keterangan'] . '<br/>';
65. echo '||' . $rate;
66. }
67.
68.
69.
70. //menampilkan halaman input sampah
71. function add() {
72.     if ($this->data['akses'] != NULL) {
73.         $this->data['title'] = 'Tambah Tempat Pembuangan';
74.         $this->data['content'] = $this->load->view(
75.             'sampah_add', array(
76.                 'kategori' => $this->sampah_model->get_kategori(),
77.             ), TRUE);
78.         $this->load->view('template_baru', $this->data);
79.     } else {
80.         redirect('home');
81.     }
82. }
83.
84.
85. //menampilkan halaman edit, dengan mengambil id sampah
86. function edit($id) {
87.     if ($this->data['akses'] != NULL) {
88.         $this->data['title'] = 'Edit Data Sampah';
89.         $this->data['content'] = $this->load->view('sampah_edit', array(
90.             'data' => $this->sampah_model->get_sampah_detail($id),
91.             'kategori' => $this->sampah_model->get_kategori(),
92.         ), TRUE);
93.         $this->load->view('template_baru', $this->data);
94.     } else {
95.         redirect('home');
96.     }
97. }
98. // fungsi yang dijalankan dalam proses menambah sampah
99. function add_process() {
100.     if ($this->data['akses'] != NULL) {
101.         if ($this->__add_rule()) {
102.             if ($_FILES['image']['name'] == "") {
103.                 $gambar = 1;
104.             } else {
105.                 $gambar = $_FILES['image']['name'];
106.             }
107.             $data = array(
108.                 'nama_sampah' => strtoupper($this->input->post('nama_sampah')),
109.                 'lokasi' => $this->input->post('lokasi'),
110.                 'id_kategori' => $this->input->post('kategori'),
111.                 'gambar' => $gambar,
112.                 'lat' => $this->input->post('lat'),
113.                 'long' => $this->input->post('long'),
114.                 'keterangan' => $this->input->post('keterangan'),
115.                 // 'id_icon' => $this->icon_sampah($this->input->post('kategori'))
116.                 'id_icon' => $this->input->post('kategori')
117.             );
118.             if ($data['gambar'] == 1) {
119.                 $sukses = $this->sampah_model->add_sampah($data);
120.                 if ($sukses) {
121.                     $this->session->set_userdata('message', '1,Tempat Pembuangan telah

```

```

122.      ditambahkan.);
123.          redirect('sampah/cobasampah');
124.      } else {
125.          $this->session->set_userdata('message', '2,Tempat Pembuangan
126.      ditambahkan.);
127.          redirect('sampah/add');
128.      }
129.      } else {
130.          $config_image = array(
131.              'upload_path' => './images/',
132.              'allowed_types' => 'gif|jpg|png|jpeg',
133.              'max_size' => '500000',
134.              'overwrite' => TRUE,
135.              'remove_spaces' => FALSE,
136.              'file_name' => $_FILES['image']['name']
137.          );
138.
139.          $this->load->library('upload');
140.          $this->upload->initialize($config_image);
141.
142.          if ($this->upload->do_upload('image')) {
143.              $sukses = $this->sampah_model->add_sampah($data);
144.              if ($sukses) {
145.                  $this->session->set_userdata('message', '1,Tempat Pembuangan telah
146.      ditambahkan.);
147.                  redirect('sampah/cobasampah/');
148.              } else {
149.                  $this->session->set_userdata('message', '2,Tempat Pembuangan gagal
150.      ditambahkan.);
151.                  redirect('sampah/cobasampah/');
152.              }
153.          } else {
154.              $this->session->set_userdata('message', '2,Upload gambar gagal. ');
155.              redirect('sampah/add');
156.          }
157.      }
158.      } else {
159.          $this->add();
160.      }
161.      } else {
162.          redirect('home');
163.      }
164.
165.  }
166.  // fungsi menyimpan perubahan dalam proses edit
167.  function edit_process() {
168.      if ($this->data['akses'] != NULL) {
169.          if ($this->__edit_rule()) {
170.              if ($_FILES['image']['name'] == "") {
171.                  $gambar = 1;
172.              } else {
173.                  $gambar = $_FILES['image']['name'];
174.              }
175.
176.              $data = array(
177.                  'id_sampah' => $this->input->post('id_sampah'),
178.                  'nama_sampah' => strtoupper($this->input->post('nama_sampah')),
179.                  'lokasi' => $this->input->post('lokasi'),
180.                  'id_kategori' => $this->input->post('kategori'),
181.                  'gambar' => $gambar,
182.                  'lat' => $this->input->post('lat'),
183.                  'long' => $this->input->post('long'),

```

```

184.         'keterangan' => $this->input->post('keterangan'),
185.         'id_icon' => $this->icon_sampah($this->input->post('kategori'))
186.     );
187.
188.     if ($data['gambar'] == 1) {
189.         $sukses = $this->sampah_model->edit_sampah($data);
190.         if ($sukses) {
191.             $this->session->set_userdata('message', '1, Data sampah telah
192. diperbarui. ');
193.             redirect('sampah/cobasampah/' . $this->input->post('id_sampah'));
194.         } else {
195.             $this->session->set_userdata('message', '2, Data sampah gagal
196. diperbarui. ');
197.             redirect('sampah/edit/' . $this->input->post('id_sampah'));
198.         }
199.     } else {
200.         $config_image = array(
201.             'upload_path' => './images/',
202.             'allowed_types' => 'gif|jpg|png|jpeg',
203.             'max_size' => '500000',
204.             'overwrite' => TRUE,
205.             'remove_spaces' => FALSE,
206.             'file_name' => $_FILES['image']['name']
207.         );
208.         $this->load->library('upload');
209.         $this->upload->initialize($config_image);
210.         if ($this->upload->do_upload('image')) {
211.             $sukses = $this->sampah_model->edit_sampah($data);
212.             if ($sukses) {
213.                 $this->session->set_userdata('message', '1, Data sampah telah
214. diperbarui. ');
215.                 redirect('sampah/edit/' . $this->input->post('id_sampah'));
216.             } else {
217.                 $this->session->set_userdata('message', '2, Data sampah gagal
218. diperbarui. ');
219.                 redirect('sampah/edit/' . $this->input->post('id_sampah'));
220.             }
221.         } else {
222.             $this->session->set_userdata('message', '2, Upload gambar gagal. ');
223.             redirect('sampah/edit/' . $this->input->post('id_sampah'));
224.         }
225.     }
226. } else {
227.     $this->edit($this->input->post('id_sampah'));
228. }
229. } else {
230.     redirect('home');
231. }
232. }
233. // proses mennghapus id berdasarkan sampah yang dipilih
234. function delete_process($id) {
235.     if ($this->data['akses'] != NULL) {
236.         if ($this->sampah_model->cek_transaksi($id) == TRUE) {
237.             $sukses = $this->sampah_model->delete_sampah($id);
238.             if ($sukses == true) {
239.                 $this->session->set_userdata('message', '1, Sampah telah dihapus. ');
240.                 redirect('sampah/cobasampah');
241.             } else {
242.                 $this->session->set_userdata('message', '2, Sampah gagal dihapus. ');
243.                 redirect('sampah/list_sampah/');
244.             }
245.         } else {

```

```

246.         $this->session->set_userdata('message', '2,Telah ada transaksi dengan tabel
247. lain. Sampah tidak boleh dihapus. ');
248.         redirect('sampah/list_sampah/');
249.     }
250. } else {
251.     redirect('home');
252. }
253. }
254. // fungsi dalam tambah sampah, untuk mengecek adakah nama yang sama
255. function cek_namasampah() {
256.     $validateValue = $_REQUEST['fieldValue'];
257.     $validateId = $_REQUEST['fieldId'];
258.
259.     $arrayToJs = array();
260.     $arrayToJs[0] = $validateId;
261.
262.     $cek = $this->sampah_model->cek_namasampah($validateValue);
263.     if ($cek == true) {
264.         $arrayToJs[1] = true;
265.         echo json_encode($arrayToJs);
266.     } else {
267.         for ($x = 0; $x < 1000000; $x++) {
268.             if ($x == 990000) {
269.                 $arrayToJs[1] = false;
270.                 echo json_encode($arrayToJs);
271.             }
272.         }
273.     }
274. }
275.
276.
277. function icon_sampah($input){
278.     $icon = 1;
279.     for ($i = 0; $i < count($input); $i++) {
280.         if ($input[$i] == 2) {
281.             $icon += 1;
282.         }
283.         if ($input[$i] == 4) {
284.             $icon += 2;
285.         }
286.     }
287.     return $icon;
288. }
289.
290. private function __rule($rule) {
291.     $this->load->library('form_validation');
292.     $this->form_validation->set_rules($rule);
293.     if ($this->form_validation->run()) {
294.         return TRUE;
295.     } else {
296.         return FALSE;
297.     }
298. }
299. // menambahkan rule baru ke dalam peta dan algoritma
300. private function __add_rule() {
301.     $rule = array
302.     (
303.         array
304.         (
305.             'field' => 'nama_sampah'
306.             , 'label' => 'nama sampah'
307.             , 'rules' => 'required'

```

```
308. )
309. , array
310. (
311.     'field' => 'lat'
312.     , 'label' => 'latitude'
313.     , 'rules' => 'trim|required'
314. )
315. , array
316. (
317.     'field' => 'long'
318.     , 'label' => 'longitude'
319.     , 'rules' => 'trim|required'
320. )
321. , array
322. (
323.     'field' => 'lokasi'
324.     , 'label' => 'lokasi'
325.     , 'rules' => 'trim|required'
326. )
327. );
328. return $this->__rule($rule);
329. }
330. // fungsi edit rule, ketika sampah diedit
331. private function __edit_rule() {
332.     $rule = array
333.     (
334.         array
335.         (
336.             'field' => 'nama_sampah'
337.             , 'label' => 'nama sampah'
338.             , 'rules' => 'required'
339.         )
340.     , array
341.     (
342.         'field' => 'lat'
343.         , 'label' => 'latitude'
344.         , 'rules' => 'trim|required'
345.     )
346.     , array
347.     (
348.         'field' => 'long'
349.         , 'label' => 'longitude'
350.         , 'rules' => 'trim|required'
351.     )
352.     , array
353.     (
354.         'field' => 'lokasi'
355.         , 'label' => 'lokasi'
356.         , 'rules' => 'trim|required'
357.     )
358. );
359. return $this->__rule($rule);
360. }
361. }
362. }
363.
364. ?>
365.
366.
```

## 2. Kelas sampah\_model

Penulisan kode program sampah\_model dapat dilihat pada Tabel 14.

Tabel 14 Kode Program Kelas sampah\_model

1.	<?php
2.	
3.	if (!defined('BASEPATH'))
4.	exit('No direct script access allowed');
5.	
6.	class Sampah_Model extends CI_Model {
7.	
8.	function __construct() {
9.	parent::__construct();
10.	}
11.	
12.	function category(\$id) {
13.	if (\$id != 0) {
14.	\$kategori = " and tb_sampah.id_kategori = \$id ";
15.	} else {
16.	\$kategori = "";
17.	}
18.	return \$kategori;
19.	}
20.	
21.	function jumlah(){
22.	return \$this->db->query ("select COUNT(id_sampah) as jumlah from tb_sampah
23.	where id_kategori = 1");
24.	}
25.	
26.	function get_semua(){
27.	return \$this->db->query("select * from tb_sampah")->result_array();
28.	}
29.	
30.	
31.	// fungsi untuk menampilkan pom terdekat
32.	
33.	function get_nearest_sampah(\$input) {
34.	\$kategori = \$this->category(\$input['kategori']);
35.	\$data = \$this->db->query("select distinct(tb_sampah.id_sampah), nama_sampah, x,
36.	y, tb_sampah.id_kategori, gambar, id_icon, st_distance_spheroid(geom,
37.	st_setsrid(st_makepoint(\$input[x],\$input[y]), 4326),
38.	'SPHEROID["WGS84",6378137,298.25728]') / 1000 as jarak, st_asgeojson(geom) as
39.	geojson
40.	from tb_sampah, tb_sampahKategori
41.	where tb_sampah.id_kategori = tb_sampahKategori.id_kategori
42.	and tb_sampah.id_status = 1
43.	and st_distance_spheroid(geom, st_setsrid(st_makepoint(\$input[x],\$input[y]),
44.	4326), 'SPHEROID["WGS84",6378137,298.25728]')/1000 <= \$input[radius] \$kategori
45.	\$produk \$layanan
46.	order by st_distance_spheroid(geom,
47.	st_setsrid(st_makepoint(\$input[x],\$input[y]), 4326),
48.	'SPHEROID["WGS84",6378137,298.25728]')/1000 asc");
49.	return \$data->result_array();
50.	}
51.	// menampilkan semua koordinat sampah
52.	function get_all_sampah(\$input) {
53.	\$kategori = \$this->category(\$input['kategori']);
54.	\$data = \$this->db->query("select distinct(tb_sampah.id_sampah), nama_sampah, x,
55.	y, tb_sampah.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson

```
56.         from tb_sampah, tb_sampahkategori
57.         where tb_sampah.id_kategori = tb_sampahKategori.id_kategori
58.         $kategori");
59.     return $data->result_array();
60. }
61.
62. function search($input) {
63.     $kategori = $this->category($input['kategori']);
64.
65.     $data = $this->db->query("select distinct(tb_sampah.id_sampah), nama_sampah, x,
66. y, tb_sampah.id_kategori, gambar, id_icon, st_asgeojson(geom) as geojson
67.     from tb_sampah, tb_sampahkategori
68.     where tb_sampah.id_kategori = tb_sampahKategori.id_kategori
69.     and nama_sampah ilike '%$input[keyword]%' $kategori");
70.     return $data->result_array();
71. }
72.
73. // get
74. function get_list_sampah() {
75.     $data = $this->db->query("select * from
76.     tb_sampah, tb_sampahkategori
77.     where tb_sampah.id_kategori = tb_sampahkategori.id_kategori
78.     order by nama_sampah asc");
79.     return $data->result_array();
80. }
81.
82. function get_sampah_by($id) {
83.     $data = $this->db->query("select *
84.     from tb_sampah, tb_sampahkategori
85.     where tb_sampah.id_kategori = tb_sampahkategori.id_kategori
86.     and id_sampah = $id");
87.     return $data->row_array();
88. }
89.
90.
91. function get_sampah_detail($id) {
92.     $data = $this->db->query("select *
93.     from tb_sampah, tb_sampahkategori
94.     where tb_sampah.id_kategori = tb_sampahkategori.id_kategori
95.     and id_sampah = $id");
96.     return $data->row_array();
97. }
98.
99. function get_sampah_form() {
100.     $data = $this->db->query("select * from tb_sampah");
101.     return $data->result_array();
102. }
103.
104.
105. function get_kategori() {
106.     return $this->db->query("select * from tb_sampahkategori")->result_array();
107. }
108.
109. function get_rows_sampah() {
110.     return $this->db->query("select id_sampah from tb_sampah")->num_rows();
111. }
112.
113. function get_new_id() {
114.     $data = $this->db->query("select max(id_sampah) as id from tb_sampah");
115.     $last = $data->row_array();
116.     if ($last['id'] == 0 || $last['id'] == NULL) {
117.         return 1;

```



```
118.     } else {
119.         return ($last['id'] + 1);
120.     }
121. }
122.
123. // cek
124. function cek_namasampah($nama){
125.     $data = $this->db->query("select * from tb_sampah where nama_sampah =
126.     '".strtoupper($nama)."'");
127.     if($data->num_rows() > 0){
128.         return FALSE;
129.     } else{
130.         return TRUE;
131.     }
132. }
133.
134. function cek_transaksi($id_sampah){
135.     $data1 = $this->db->query("select * from tb_rating where id_sampah =
136.     $id_sampah");
137.     $data2 = $this->db->query("select * from tb_review where id_sampah =
138.     $id_sampah");
139.     if($data1->num_rows() > 0 || $data2->num_rows() > 0){
140.         return FALSE;
141.     } else{
142.         return TRUE;
143.     }
144. }
145.
146. function __is_exist($id) {
147.     $this->db->where('id_sampah', $id);
148.     $data = $this->db->get('tb_sampah');
149.     if ($data->num_rows() > 0) {
150.         return TRUE;
151.     } else {
152.         return FALSE;
153.     }
154. }
155.
156. // set
157. function add_sampah($input) {
158.     $new_id = $this->get_new_id();
159.     $user = $this->session->userdata('sampah_iduser');
160.     $this->db->trans_begin();
161.     if ($input['gambar'] == 1) {
162.         $this->db->query("
163.         insert into tb_sampah (
164.             id_sampah,
165.             nama_sampah,
166.             lokasi_sampah,
167.             id_kategori,
168.             keterangan,
169.             id_user,
170.             id_icon,
171.             x,
172.             y,
173.             geom
174.         ) values (
175.             '" . $new_id . "',
176.             '" . $input['nama_sampah'] . "',
177.             '" . $input['lokasi'] . "',
178.             '" . $input['id_kategori'] . "',
179.             '" . $input['keterangan'] . '";
```

```

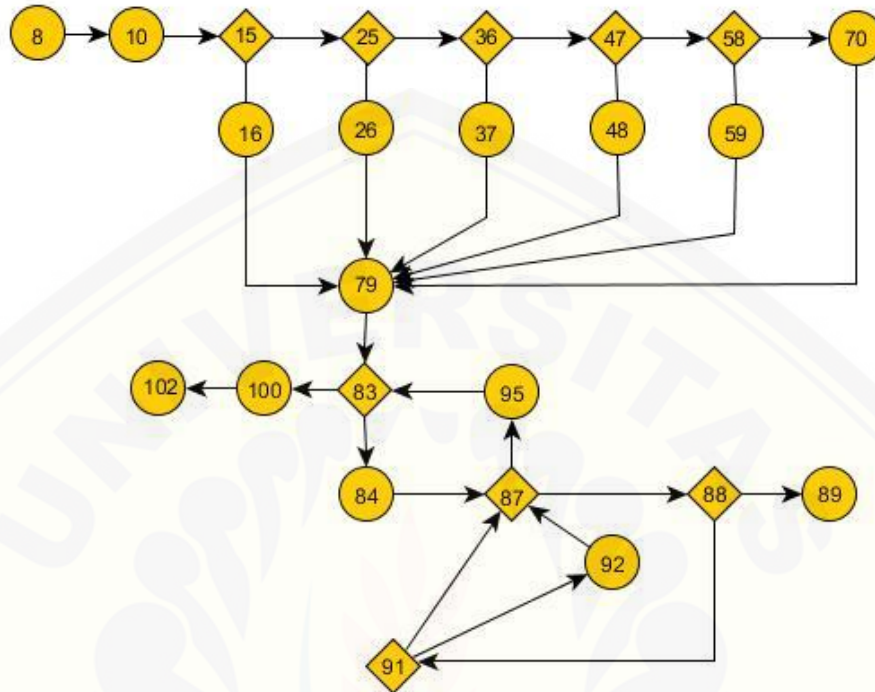
180.         ". $user . ",
181.         ". $input['id_kategori'] . ",
182.         $input[long],
183.         $input[lat],
184.         ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
185.     )
186. ");
187. } else {
188.     $this->db->query("
189.         insert into tb_sampah (
190.             id_sampah,
191.             nama_sampah,
192.             lokasi_sampah,
193.             id_kategori,
194.             keterangan,
195.             gambar,
196.             id_user,
197.             id_icon,
198.             x,
199.             y,
200.             geom
201.         ) values (
202.             "" . $new_id . "",
203.             "" . $input['nama_sampah'] . "",
204.             "" . $input['lokasi'] . "",
205.             "" . $input['id_kategori'] . "",
206.             "" . $input['keterangan'] . "",
207.             "" . $input['gambar'] . "",
208.             ". $user . ",
209.             "" . $input['id_kategori'] . "",
210.             $input[long],
211.             $input[lat],
212.             ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
213.         )
214.     ");
215. }
216.
217. if ($this->db->trans_status() === FALSE) {
218.     $this->db->trans_rollback();
219.     return FALSE;
220. } else {
221.     $this->db->trans_commit();
222.     return TRUE;
223. }
224. }
225.
226. function edit_sampah($input) {
227.     $this->db->trans_begin();
228.     if ($input['gambar'] == 1) {
229.         $this->db->query("
230.             update tb_sampah set
231.             nama_sampah = '$input[nama_sampah]',
232.             lokasi_sampah = '$input[lokasi]',
233.             id_kategori = $input[id_kategori],
234.             keterangan = '$input[keterangan]',
235.             id_icon = $input[id_icon],
236.             x = $input[long],
237.             y = $input[lat],
238.             geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
239.             where id_sampah = $input[id_sampah]
240.         ");
241.     } else {

```

```
242.     $this->db->query("
243.         update tb_sampah set
244.             nama_sampah = '$input[nama_sampah]',
245.             lokasi_sampah = '$input[lokasi]',
246.             id_kategori = $input[id_kategori],
247.             keterangan = '$input[keterangan]',
248.             gambar = '$input[gambar]',
249.             id_icon = $input[id_icon],
250.             x = $input[long],
251.             y = $input[lat],
252.             geom = ST_GeomFromText('POINT($input[long] $input[lat])', 4326)
253.             where id_sampah = $input[id_sampah]
254.     ");
255. }
256.
257. if ($this->db->trans_status() === FALSE) {
258.     $this->db->trans_rollback();
259.     return FALSE;
260. } else {
261.     $this->db->trans_commit();
262.     return TRUE;
263. }
264. }
265.
266.
267. function delete_sampah($id) {
268.     if ($this->__is_exist($id)) {
269.         $this->db->trans_begin();
270.         $this->db->query("delete from tb_sampah where id_sampah = $id");
271.         if ($this->db->trans_status() === FALSE) {
272.             $this->db->trans_rollback();
273.             return FALSE;
274.         } else {
275.             $this->db->trans_commit();
276.             return TRUE;
277.         }
278.     } else {
279.         return FALSE;
280.     }
281. }
282.
283. function latlong(){
284.     $data = $this->db->query("select x as lng, y as lat, nama_sampah from
285.     tb_sampah");
286.     return $data->result_array();
287. }
288.
289. }
290.
291. ?>
292.
```

**E. Pengujian White Box**

**E.1 Kelas Controller**



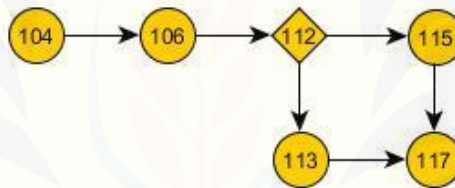
Gambar 13 Diagram Alir *Function count()*

$$CC = E - N + 2 = 31 - 24 + 2 = 9$$

Tabel 15 *Test Case Function count()*

Test case	<b>Jika ada data biasa masuk</b>
Target yang Diharapkan	Menampilkan jalur pada map
Hasil pengujian	Benar
Path/jalur	8-10-15-16-79-83-100-102
Test case	Jika data terdapat data geoJson
Target yang Diharapkan	Menampilkan jalur pada map

Hasil pengujian	Benar
Path/jalur	8-10-15-16-79-83-100-102
Test case	Jika penguraian data tidak berhasil
Target yang Diharapkan	Tidak ditampilkan jalur
Hasil pengujian	Benar
Path/jalur	8-10-79-83-84-87-95



Gambar 14 Diagram Alir *Function save()*

$$CC = E - N + 2 = 6 - 6 + 2 = 2$$

Tabel 16 *Test Case Function save()*

Test case	<b>Jika berhasil menyimpan data pembuangan</b>
Target yang Diharapkan	Menampilkan pesan sukses
Hasil pengujian	Benar
Path/jalur	104-106-112-113-117
Test case	Jika tidak berhasil menyimpan data pembuangan

Target yang Diharapkan	Menampilkan 404
Hasil pengujian	Benar
Path/jalur	104-106-112-115-117



Gambar 15 Diagram Alir *Function* escapeJsonString()

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 17 *Test Case Function* escapeJsonString ()

Test case	<b>Jika data berhasil di generate</b>
Target yang Diharapkan	Pertukaran data dalam fungsi count()
Hasil pengujian	Benar
Path/jalur	119-120-125

E.2 Kelas Model



Gambar 16 Diagram Alir *Function* get\_route\_geojson()

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 18 Test Case Function get\_route\_geojson ()

Test case	<b>Jika berhasil menambah data pembuangan</b>
Target yang Diharapkan	Menampilkan rute jalur terpendek
Hasil pengujian	Benar
Path/jalur	8-9-22



Gambar 17 Diagram Alir Function simpan()

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 19 Test Case Function simpan()

Test case	<b>Jika data pembuangan ditambahkan</b>
Target yang Diharapkan	Menyimpan data pembuangan ke database
Hasil pengujian	Benar
Path/jalur	24-25-27

**F. Pengujian Black Box**

F.1 Fitur Pengelolaan Data Akun

Tabel 20 Pengujian Black Box Pengelolaan Data Akun

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data user	Memilih menu user	Menampilkan tabel daftar user	Berhasil
2.	Menambah	Memilih	Menampilkan form	Berhasil

	data user	tombol data baru	tambah user	
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman user kembali	Berhasil
3.	Mengubah data user	Memilih tombol edit	Menampilkan form edit user	Berhasil
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman user kembali	Berhasil
4.	Menghapus data user	Memilih tombol hapus	Menghapus data dari <i>database</i>	Berhasil
5.	Detail data user	Memilih tombol detail	Menampilkan data detail user yang dipilih	Berhasil

## F.2 Fitur Mengelola Data TPS dan TPA

Tabel 21 Pengujian *Black Box* Mengelola Data TPS dan TPA

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data TPS dan TPA	Memilih menu Manajemen TPS dan TPA	Menampilkan tabel daftar TPS dan TPA	Berhasil
2.	Menambah data TPS dan TPA	Memilih tombol data baru	Menampilkan form tambah TPS dan TPA	Berhasil



		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman TPS dan TPA kembali	Berhasil
3.	Mengubah data TPS dan TPA	Memilih tombol edit	Menampilkan form edit TPS dan TPA	Berhasil
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman TPS dan TPA kembali	Berhasil
4.	Menghapus data TPS dan TPA	Memilih tombol hapus	Menghapus data dari <i>database</i>	Berhasil
5.	Detail data TPS dan TPA	Memilih tombol detail	Menampilkan data detail TPS dan TPA yang dipilih	Berhasil

### F.3 Fitur Melihat Data Pembuangan Sampah

Tabel 22 Pengujian *Black Box* Melihat Data Pembuangan Sampah

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data pembuangan sampah	Memilih menu pembuangan sampah	Menampilkan tabel daftar pembuangan sampah	Berhasil

## F.4 Fitur Melihat Data TPS dan TPA

Tabel 23 Pengujian *Black Box* Melihat data TPS dan TPA

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data TPS dan TPA	Memilih menu data TPS dan TPA	Menampilkan tabel daftar TPS dan TPA	Berhasil
2.	Detail data TPS dan TPA	Memilih tombol detail	Menampilkan data detail TPS dan TPA yang dipilih	Berhasil

## F.5 Fitur Verifikasi Data Pembuangan Sampah

Tabel 24 Pengujian *Black Box* Verifikasi Pembuangan Sampah

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data pembuangan sampah	Memilih menu verifikasi pembuangan sampah	Menampilkan tabel daftar data pembuangan sampah	Berhasil
2.	Verifikasi data pembuangan sampah	Memilih tombol verifikasi	a. Memverifikasi data pembuangan sampah yang dipilih b. Menampilkan halaman tabel pembuangan sampah	Berhasil